ABSTRACT

Title of Document: TYPING THE DANCING SIGNIFIER: JIM ANDREWS' (VIS)POETICS

Leonardo Flores, PhD, 2010

Directed By: Prof. Matthew Kirschenbaum, English Department
Prof. Martha Nell Smith, English Department

This study focuses on the work of Jim Andrews, whose electronic poems take advantage of a variety of media, authoring programs, programming languages, and file formats to create poetic experiences worthy of study. Much can be learned about electronic textuality and poetry by following the trajectory of a poet and programmer whose fascination with language in programmable media leads him to distinctive poetic explorations and collaborations. This study offers a detailed exploration of Andrews' poetry, motivations, inspirations, and poetics, while telling a piece of the story of the rise of electronic poetry from the mid 1980s until the present.

Electronic poetry can be defined as first generation electronic objects that can only be read with a computer—they cannot be printed out nor read aloud without negating that which makes them "native" to the digital environment in which they were created, exist, and are experienced in. If translated to different media, they would lose the extra-textual elements that I describe in this study as behavior. These

“behaviors” electronic texts exhibit are programmed instructions that cause the text to be still, move, react to user input, change, act on a schedule, or include a sound component.

The conversation between the growing capabilities of computers and networks and Andrews’ poetry is the most extensive part of the study, examining three areas in which he develops his poetry: visual poetry (from static to kinetic), sound poetry (from static to responsive), and code poetry (from objects to applications). In addition to being a literary biography, the close readings of Andrews’ poems are media-specific analyses that demonstrate how the software and programming languages used shape the creative and production performances in significant ways.

This study makes available new materials for those interested in the textual materiality of Andrews’ videogame poem, *Arteroids*, by publishing the *Arteroids* Development Folder—a collection of source files, drafts, and old versions of the poem. This collection is of great value to those who wish to inform readings of the work, study the source code and its programming architecture, and even produce a critical edition of the work.

TYPING THE DANCING SIGNIFIER: JIM ANDREWS' (VIS)POETICS

By

Leonardo L. Flores

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2010

Advisory Committee:
Professor Matthew Kirschenbaum, Co-Chair
Professor Martha Nell Smith, Co-Chair
Professor Regina Harrison
Professor Marilee Lindeman
Professor Katie King

Dedication

I dedicate this dissertation to all who have been so generous in lending me their unconditional support throughout this long process:

- To my family: my beautiful wife Kara, my lovely children Olivia and Blake, my patient mother Nelsie, and the rest of my dear family I've been distant or absent from while focused on this project.
- To my father, Leonardo Flores Flores, who chose us over completing his Ph.D.
- To Jim Andrews, whose poetry, openness, and willingness to help have been a true inspiration.
- To my co-chairs, Matt and Martha, who have guided me and helped me elevate my work through their example and advice.
- To my friends, who have given me the space to work and been available when needed.
- To Tony and Susan Hunt, my mentors and friends whose example sent me on this academic path.
- To my colleagues and friends in the University of Puerto Rico: Mayagüez Campus, for granting me release time to pursue my doctoral studies and for being patient and considerate with my degree completion.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introducing the Dancing Signifier

> Literary works that strengthen, foreground, and thematize the connections between themselves as material artifacts and the imaginative realm of verbal/semiotic signifiers they instantiate open a window on the larger connections that unite literature as a verbal art to its material forms. (25)
>
> N. Katherine Hayles, *Writing Machines*

One aspect of poetry that has been studied extensively is how it shapes the sounds of language to create rhythms, alliteration, rhyme, assonance, consonance, onomatopoeia, and other musical effects. To the untrained ear, these rhythms may be catchy and fun, but awareness of metrical patterns, types of rhyme, how sounds are articulated, and traditional poetic devices can enhance the pleasure derived from poetry.

This study particularly attends to how the materials with which language is physically inscribed, stored, and communicated can have an impact in the poem's creation, dissemination, and reception. Poets shape words into mnemonic patterns, rhythmically articulated columns of air, handwritten, typed, or printed lines of text, sound, film, or video recordings, and files in a computer and/or network. People (whether we call them readers, listeners or users) perform elaborate mental and physical dances with the material record of the poem to produce a unique experience of the work.

Most of the time, we are unaware of the materiality of language. As you read this sentence you are probably not exerting great effort in decoding the marks on the page or screen: your brain identifies the glyphs as letters, words, sentences, paragraphs, and so on, retrieving the meaning of these marks without thinking much of them. Other times we encounter resistance in the materials of language that make us *pay attention* to those signs, sounds, gestures, and marks, like when someone mumbles, or speaks another language, or uses a homophone, or someone's handwriting is hard to decipher, or one is unsure whether a typed mark "l" could be read as a letter or a number, or a typed word like "wind," can be pronounced and understood in two different ways. At times words displayed on a screen aren't facing us and we have to navigate around them until they become legible, or perhaps they twirl, spin, fade, move about the screen, morph, change when we place the pointer over them, or simply change at a speed that imposes a reading rate.[1]

Poets are interested in the materiality of language, most commonly the articulation and patterning of sound, the visual arrangements of its written form, the programmed behaviors of electronic media. While the great majority of poets engage primarily the semantic and aural components of their poetry and use inscription technologies and media in ways that minimize the resistance with which readers can engage such components, there have always been some poets interested in how technologies affect the creation and reception of poetry.

This type of poetry often challenges traditional literacy training to access and appreciate poetry because such schooling rarely includes how to read deeply into the material characteristics of each medium. Schools of literary criticism and critical theories (both of which inform literary instruction) provide tools for analysis and

[1] I have been making references to Emily Dickinson's (and anyone else's) manuscripts, E. E. Cummings' "l(a," Eugen Gomringer's "Wind," works in VRML like David Knoebel's "Walkdon't," Brian Kim Stephans' "The Dreamlife of Letters," and any of the Young-Hae Chang Heavy Industries' productions.

interpretation of literary texts, but most have ignored or de-emphasized the materiality of the literary texts in order to focus on the linguistic elements of language.[2] Even textual critics have made generalizations about texts specific to print—the primary inscription technology for the production and dissemination of texts for the past few centuries, and the one most readers are trained to interact with. When faced with a new technology both to create and experience literary texts, such as the computer, the old "truths" derived from print seem more like assumptions and established theories begin to lose their effectiveness as critical tools and adjustments or reinventions become necessary.

Textual critic Peter Shillingsburg, for instance, has done tremendous work to expand print-based conceptions of text to include electronic textuality, particularly in his chapter "Text as Matter, Concept, and Action" in *Resisting Texts.* This model, while thorough and able to explain some characteristics of text regardless of medium, has come under fire by N. Katherine Hayles who criticizes its "alarming proliferation of terms" and finds

> An even more serious objection to Shillingsburg's definition is its implicit assumption that "text" does not include such qualities as color, font size and shape, and page placement, not to mention such electronic-specific effects as animation, mouseovers, instantaneous linking, etc. In most contemporary electronic literature, screen design, graphics, multiple layers, color, animation, etc. are signifying components essential to the work's effects. Focusing only on "the actual order of words and punctuation" would be as inadequate as

[2] The linguistic elements referred to here are semantics (meaning), morphology (grammar), syntax (word order), and phonetics (articulation of sound).

> insisting that painting consists only of shapes and ruling out of bounds color, texture, composition, perspective, etc. The largely unexamined assumption here is that ideas about textuality forged in a print environment can be carried over wholesale to the screen without rethinking how things change with electronic text, as if "text" were an inert, nonreactive substance that can be poured from container to container without affecting its essential nature ("Translating Media" 267).

In response, Hayles calls for a new model of textuality, one that "reconceptualizes materiality as the interplay between a text's physical characteristics and its signifying strategies" ("Print is Flat" 72) and urges the development of a theory of Media Specific Analysis (MSA)—"a kind of criticism that pays attention to the material apparatus producing the literary work as physical artifact" (*Writing Machines* 29). Her theory of textuality will be discussed further in Chapter 2.

There is at present very few book-length studies of electronic poetry, the first of which is by Loss Glazier, "*Digital Poetics* itself is a book that reaches *from* hypertext *through* visual/kinetic text and *to* writing in programmable media" (170). This groundbreaking book explores electronic media as a "space of poesis," arguing that the new horizon for electronic writing lies in the hands of poets who are also programmers, who can create "the active or 'intelligent' text—a text that not only writes and reads, but being software itself, can do a little soft-shoe on the side" (170). In order to perform (programmable) media specific analysis this study creates a more developed set of terms than what Glazier discusses in his book. In the words of Ezra Pound's "Pact" with Walt Whitman,

It was you that broke the new wood,
Now is a time for carving.

Christopher Funkhouser’s *Prehistoric Digital Poetry,* begins the work of carving out a history for digital poetry before the rise of the Web in the mid-1990s. This book surveys early digital works in order to establish the foundations of a digital poetics, which were built upon by poets like Jim Andrews who developed works for the Web. Some of Jim Andrews’ early visual poetry is featured in Funkhouser’s book, providing a small overlap that is built upon further in this study.

In tune with Hayles’ call for media-specific analysis, and in continuation of the work initiated by Glazier and Funkhouser this study creates a tool for the understanding and appreciation of poetry that has been created with a new inscription technology (the digital computer) and for a new medium (the Web). *Typing the Dancing Signifier* creates a typology of behaviors exhibited by e-poetry in order to better explore the work of a prominent poet in the field: Jim Andrews. The remainder of this chapter will define electronic poetry and textuality, introduce the typology and justify the focus on Jim Andrews.

What is E-Poetry?

With the rise of the personal computer and the Internet a new textual and poetic phenomenon has emerged: electronic poetry or e-poetry are its two primary names. Other ways of referring to it are digital poetry, computer poetry, hyperpoetry, polypoetry, Web poetry, Flash poems, Javascript poems, and the list goes on. All of these names have a history and denote specific practices that render them unsuitable as an overarching term for this poetic phenomenon. For instance, computer poetry often refers to computer-generated poems, which are not necessarily published in electronic format. E-poetry is the most used term for this type of poetry, partly because the “e-“ prefix reflects popular uses of the Internet (e-mail, e-cards, e-vites, etc.), partly because of theorists and practitioners like Loss Pequeño Glazier, who in *Digital Poetics* admits that “there is no agreed-upon term for digital poetry. It will sometimes be referred to in this volume as digital poetry, electronic poetry, e-poetry, or computer-generated writing” (181). For the sake of simplicity, I will use the terms “electronic poetry” and “e-poetry” because they are broad enough to include a wide variety of practices, and focused enough on electronic media.[3]

The definition of electronic poetry (and electronic literature) is elusive because the computer is a multimedia environment, integrating elements of print, film, video, and audio recording. It is also difficult to sum up the variety of practices that fall under the rubric of e-poetry. Here are three definitions by three authorities in the field: Loss Pequeño Glazier, N. Katherine Hayles, and the Electronic Literature Organization (ELO).

1. Glazier describes three somewhat redundant qualities in *Digital Poetics*, the most complete of which is the second: “Texts with certain structural/operative forms not reproducible in paper or in any non-digital medium. These include employing hyperlinks, kinetic elements, multi-layered features, programmable elements and events” (163). This definition has the virtue of listing some of the elements that require the computer for operation.

[3] By “electronic media”, I mean the digital computer, which I may also refer to as “digital media” and “programmable media.”

2. N. Katherine Hayles' pithy (though claustrophobic) definition, "literature created in electronic media and meant to be read in them," places both the inscription technology and the author's publication intent at the center of the definition: a problematic proposition because it opens the definition to works that don't necessarily depend upon electronic media ("Writing Machines Web Supplement").
3. The Electronic Literature Organization defines electronic literature as "works with important literary aspects that take advantage of the capabilities and contexts provided by the stand-alone or networked computer," emphasizing the use of the computer as an inscription technology, but not underscoring enough the use of the computer as media ("About the ELO").

As may be clear by now, there is no "perfect" definition as of yet, nor do I claim to have achieved such perfection with my own. There does seem to be consensus on at least two characteristics of electronic literature: (1) it explores the capabilities of the computer (whether stand-alone or networked) and (2) it is designed to be experienced through the computer. With these characteristics in mind, I will define electronic poetry as poems that can only be read with a computer—they cannot be printed out nor read aloud without negating that which makes them "native" to the digital environment in which they were created, exist, and are experienced in. If translated to different media, they would lose the extra-textual elements that I describe in this study as behavior. These "behaviors" electronic texts exhibit are nothing more than programmed instructions that cause the text to move, react to user input, change, act on a schedule, or include a sound component.

Before defining these behaviors further, I will explore some of the edges of the concept of electronic poetry. To complete the definition, however, I also need to place it in conversation with preexisting definitions of poetry. In other words, how is “electronic poetry” poetry? Instead of taking on the age-old task of defining poetry itself, I will discuss how e-poetry fits within conceptions of poetry produced in oral, manuscript, or print media and how it continues ancient and recent poetic traditions. These distinctions become necessary because electronic literature and textuality are so new to literary and cultural history, as new as the technological innovations that make them possible: the digital computer and the Internet.

E-Poetry and the Computer

When identifying what makes a poem “electronic,” one should keep in mind three ways in which computers have been used for poetry: as a means of production (word processors, programming languages, and authoring programs), as a means of storage and distribution (magnetic storage media and networks, such as the Internet), and as a medium through which it is read (screen, speakers, mouse, keyboard). The history of these uses accounts for some of the confusion people have when defining e-poetries.

The most common use for the computer in the creation of poetry is as a word processor, which “remediates” the typewriter in its capabilities. Jay David Bolter and Richard Grusin coined this term to explain the process of representation of an old medium in a new one (45). In other words, the word processor is an improvement on the typewriter because it allows for the writer to format and edit a text on the screen before making a printed or “hard” copy. Using a word processor to write a poem doesn’t necessarily make it “electronic” because this kind of software is designed primarily to produce printed copies. As an inscription technology it still leaves a mark on a poem, partly in the composition process, and partly in how a poem looks, because it provides a diverse palette of formatting elements (different fonts, sizes, spacing options, color, the addition of graphical elements such as pictures, and many more options). A poet who uses a computer as an inscription technology has remarkable control over the means of production and can certainly use that for expressive potential in his or her work—but if the intended output is a printed copy or an oral rendition of the poem, then it is not really an e-poem.

Using digital media and networks to distribute a poem does not necessarily make it an electronic poem either. Many Web publications contain original or transcribed poetry that may only be accessible online because it has never been published in print, which is not electronic poetry.[4] Typing a Robert Creeley poem into an e-mail message and sending it, or posting it in a Web site, does not necessarily make the poem “electronic,” though it is certainly in electronic format and can be affected by the medium in significant ways. If you take advantage of the computer’s time-based or calculating capabilities to add behaviors to that poem, however, then

[4] Given the history of computers and their use for poetic production and distribution, it is no surprise that many people think electronic poetry is mostly bad poetry that gets self-published on the Web because no one wants to print it—a result of what Glazier calls the “me-oh-me” revolution of the 1990s (155). The evaluative filtering that has been so well established in print culture is a result of production and distribution costs involved. Publication (to make public) has opened up from the economic constraints of print, because the computer and the Web serve as a means for production and dissemination to a large audience.

you are recreating the work as an e-poem, as is the case with Brian Kim Stefans' version of Creeley's *"I Know a Man," One Letter at a Time*.[5]

This leads us to the third aspect—the place where an e-poem can be read. Electronic poetry takes advantage of the possibilities offered by the computer as a reading machine. As a rule of thumb, if a poem can be easily printed out or read aloud without it losing essential signifying aspects, then it is not truly dependent upon the computer as a medium, and it is therefore not an e-poem. Stefans' version of Creeley's could potentially be printed in a flip-book to recreate its scheduled display of the text, but that would transform the work further by providing different ways to operate the work: random access, slower or faster reading rate, not to mention the change in context and loss of the sound component.

If we base the definition strictly upon this dependence upon digital media, then we must be clear about what constitutes that dependence. This leads us to the second task of this definition: to demonstrate how e-poetry explores the capabilities, limitations, and expressive potential of digital media and still be recognizable as poetry. To do so, I will position e-poetry in the context of poetic practices and traditions that have a similar allegiance to exploring the media in which they are created and received.

[5] The title of this poem describes it well: the poem presents the reader with a white background in the center of which every letter of the poem appears and disappears at a variable rate of approximately one per second, from the first to the last letter of Creeley's poem. The font type is very much like a typewriter, emphasized by the clicking sound that accompanies the appearance of each letter. This e-poem is part of his "One Letter at a Time" series that includes similar versions of "Howl" and "The Day Lady Died."

What is so different or new about electronic poetry?

The answer to this question requires an explanation of poetry, media, and technologies of production, storage, and distribution. From a technological perspective, electronic poetry is something fairly new, as new as the personal computer that entered the market in the 1980s, but really owing its proliferation to the Internet and the World Wide Web in the 1990s. From a poetic perspective, what electronic poetry does is not so new; it is simply doing what some poets have been doing from the outset: exploring the expressive potential of language within a given medium or group of interconnected media. In order to reconcile these two perspectives I will provide a brief history of how various poets have explored the poetic potential of different technologies and media.

All poetry can be seen as an attempt to capture and communicate remarkable language. *The New Princeton Encyclopedia of Poetry and Poetics* defines poetry as:

> A poem is an instance of verbal art, a text set in verse, bound speech. More generally, a poem conveys heightened forms of perception, experience, meaning, or consciousness in heightened language, i.e. a heightened mode of discourse (938).

This definition encompasses e-poetry in the most general sense of the term: that of heightened language. Language, especially when heightened, cannot escape the constraints of the medium in which it is created, recorded, and transmitted. Different media place different demands on language—the combination of which gives rise to different poetic, definitions, forms, traditions.

The oldest technologies for poesis are supplied by the human body. Walter J Ong's discussion of Havelock's notions of orality sheds light on how memory works as a storage medium.

> In a primary oral culture, to solve effectively the problem of retaining and retrieving carefully articulated thought, you have to do your thinking in mnemonic patterns, shaped for ready oral recurrence. Your thought must come into being in heavily rhythmic, balanced patterns, in repetitions or antitheses, in alliterations and assonances, in epithetic and other formulary expressions, in standard thematic settings (the assembly, the meal, the duel, the hero's 'helper', and so on), in proverbs which are constantly heard by everyone so that they come to mind readily and which themselves are patterned for retention and ready recall, or in other mnemonic form (34).

The human body provides the means for dissemination of such thought: voice and gestures transmit poems, ideas to an audience that listens to and observes the performance. Dissemination occurs either through travel, or by teaching others the poem so they can pass it along. The more human bodies are dedicated to memorizing a poem in an oral culture, the greater chance that work has to survive over the years. This is what Charles Bernstein calls "poetry's epic function—the necessity of storage and transmission of the culture's memories and laws" (515). Some notable examples of poetry in oral cultures are epic poems such as *The Iliad*, *El Cantar del Mio Cid*, *La Chanson de Roland*, *Beowulf*, all of which survived to this date, ironically, because they were subsequently recorded in writing before they were lost to human memory.

Writing is a technology that improved the storage and dissemination of works, and added a visual dimension to poetry by translating time (of voice and mental ear) into space (on the page). Reading a printed poem is an act of decoding simultaneously several sets of visual information: the words on the page (which have both sound and meaning and are organized according to grammatical conventions), the conventions of printed poetry (such as line breaks, stanzas, and spacing), and other graphical codes (such as formatting, pictorial art, etc.).

> A page of printed or scripted text should thus be understood as a certain kind of graphic interface. The complexity of the interface varies from a minimal use of the bibliographical codes open to a given paperspace—the text you are now reading is a good example of such simplicity—to highly elaborated interfaces like those determined as poetic texts. Some of the latter exploit the bibliographical resources of paperspace to an extreme degree—Pound's *Cantos*, for example, or Dickinson's various writings—while others are satisfied to work within a set of basic and commonly used conventions (McGann, "Dialogue" 199). [6]

A poet can use the visual information in a variety of ways to create meaning in his/her work. For example, as soon as a poet inscribes a poem on a page, it becomes a written performance of an originally mental and perhaps oral performance. Traditionally, the written form is a score for an oral reconstruction of the poem: time is translated into space on the page as line breaks, spacing, and punctuation represent pauses. The page

[6] Textual critics Jerome McGann, George Bornstein, and G. Thomas Tanselle argue for the significance of graphical and bibliographical codes often deemed insignificant by critics and editors who favor only the linguistic codes that constitute a text.

becomes laden with bibliographical codes—conventions that allow the reader to reconstruct the poem in an oral performance. This system has worked well for centuries, and people have been trained to follow these protocols when they read poetry. In fact, we have become so used to poetic print conventions that we define poetry in terms of these media that we are so comfortable with, as is seen in the opening chapter of *The Norton Anthology of Poetry*: "A poem is a composition written for performance by the human voice" (Ferguson lxi). Paul Fussell illustrates this union of aural and visual components that a reader experiences when reading a poem.

> The shape which a poetic stanza cuts in time was once, before the widespread use of printing, apprehended by the ear alone. In later times, the reader's conception of stanzaic form has been both aural and visual. And now that we are fully accustomed to using printed texts for apprehending poems, our sense of stanzas has become a very complex act of mediation between what our eyes see and what our inner ears hear (128).

This definition implies a joining of two media (writing and voice), as is discussed in the previous paragraph. Most poetry fits comfortably within this model and therefore leaves its mark on mainstream definitions of poetry. The majority of poets create within the bounds of a given technology of production, storage, and distribution without necessarily exploring the relationship between language and media—after all, there are plenty of other topics to deal with poetically.

The problem with this model emerges when the graphical elements of the written performance become primary carriers of meaning. Since the written document is no longer secondary to the poem, its semantic and graphical codes are no longer transparent, translatable carriers of meaning (they never were, but we were trained to read *through* them). It gains the solidity of a final work and moves in the direction of graphic design and pictorial art, as is the case with Concrete Poetry.

> There is a fundamental requirement which the various kinds of concrete poetry meet: concentration upon the physical material from which the poem or text is made. Emotions and ideas are not the physical materials of poetry. . . . Generally speaking the material of the concrete poem is language: words reduced to their elements of letters (to see) syllables (to hear). . . . Put another way this means the concrete poet is concerned with making an object to be perceived rather than read (Solt).

The concrete poetry movement places the technotext[7] at the center of its poetics, at a time (late 1950s onwards) when the predominant inscription technologies were print and sound recording. Their attention to the media is by no means new. On the axis of writing, there is a rich tradition of emblematic verse and other visual poetry beginning circa 300 B.C. with Simmias of Rhodes' egg and axe shaped poems. On the axis of sound, performance poets (such as the Nuyorican poets, poetry slams, Def Poetry Jam), concrete sound poets, and others, have privileged the sound components of poetry over the written ones. These are just a few traditions that focus on one or the other media associated with poetry—the technotext is as old as poetry itself.

[7] "When a literary work interrogates the inscription technology that produces it, it mobilizes reflexive loops between its imaginative world and the material apparatus embodying that creation as a physical presence" (Hayles, *Writing Machines* 25).

Among those who use unique capabilities of a given medium for poetic creation, there are those who explore the potential of the inscription technologies at their disposal. Emily Dickinson capitalized on the technology of paper, ink, pens, and other letter-writing implements as a means of creation for her poems. Her means of publication was primarily letter writing, using the mail for distribution. Dickinson's work is difficult to edit and translate into print because it requires an imposition of print bibliographical conventions (line breaks, decisions whether something is prose or verse, and more) onto a work encoded by Dickinson's manuscript style. [8]

Another key figure from the golden age of print (the 19th Century), William Blake, took advantage of the press to exercise careful control over his poetic creations. He produced his works from copper plates, and personalized and adapted each print to create unique versions of the same work. His works, as well as those created by other book-artists such as William Morris, Johanna Drucker, and many others in the 20th century maximize the use of their production and storage technologies to create meaning. [9]

Other inscription technologies have transformed poetic practices. For example, the typewriter allowed poets to write in a language the printers could literally understand because they were able to produce within print conventions,

[8] The *Dickinson Electronic Archives*, produced by Martha Nell Smith and the Dickinson Editing Collective, has a rich collection of facsimiles of Emily Dickinson's manuscripts and allows for the reader to experience her work as originally created and published. The Archives also presents case studies that showcase some of Dickinson's unique poetic explorations, such as the letter-poem, highlighting the editorial interpretations needed to translate her work into print. (http://www.iath.virginia.edu/dickinson/)

[9] Johanna Drucker's *The Century of Artists' Books* provides an extensive survey and critical exploration of the Artists' Book—which is basically a technotext that operates at the level of the book.

reducing dramatically the interpretive decisions needed to translate from manuscript to print. William Carlos Williams and E. E. Cummings' typographical poems were precursors of Charles Olson's projective verse poetics, who claimed that the typewriter allowed the poet to score space directly upon a page.

The computer's use for poetic creation has a brief history of its own. Some early uses of the computer actually predated proliferation of personal computers in the 1980s. Several poets have used the computer's processing power to carry out randomizing, sorting, and other algorithms as an aid in poetic creation. Emmett Williams wrote *A Valentine for Noel* with the help of a computer that carried out an assembling and sorting procedure, as did Jackson Mac Low in *Barnesbook.*

Using tools to automate procedures and/or generate literary texts has a history of its own, since avant garde literary movements such as Surrealism, Dada, and OULIPO (Ouvroir de Litterature Potentiel) had all explored methods designed to reduce authorial agency and/or create constraints. One Surrealist method involved cutting up words from a printed text, dropping them on a surface, and writing the result as a poem. The OULIPO group used various mathematical procedures to select the words for a poem, for instance. These are just two cases that prove that the computer is not needed to introduce aleatory or algorithmic procedures into poetic creation.

The earliest publication of e-poetry can be credited to the French group LAIRE, which published in the late 1980s and 1990s an electronic poetry magazine titled *ALIRE*. Their storage medium was floppy disks and CD ROMs that contained executable programs that would display the e-poem on the computer screen when run.

Much of the poetry in these publications took advantage of the capabilities for the computer to display kinetic texts and required the readers to use the computer's input devices (keyboard and mouse). Their main obstacle was distribution, because there was not much market for this new kind of poetry, so they had to resort to subscriptions and word-of-mouth.

The Internet revolutionized the distribution of computer-readable work. The simplicity and low cost of the Web as a means for publication has caused tremendous proliferation in the amount of work done in and for the computer. Web-artist Alan Sondheim, for example, takes great advantage of the publishing capabilities of the Internet to disseminate his work on a daily basis. He e-mails his work to a large number of readers and posts it online in his Web site,[10] where it is accessible to all who care to visit his site. This distribution scale would be difficult to match in anything short of a daily newspaper, as far as the print world is concerned. To self-publish at the same scale would be impossible for anyone who is not phenomenally rich. Thus, his work is very much dependent on digital media for dissemination. E-mail and Web sites aren't the only means for online publication: online venues for e-poetry such as *Riding the Meridian, BeeHive, The Electronic Poetry Center, the Electronic Literature Organization, Cauldron and Net*, *Poems that GO,* and others showcase some of the best e-poetry on the Web.

As should be clear by now poets and other writers have explored the options offered by technology and media in their work for a very long time. When it comes to the process of creation and production, poets use the technical means of their time,

[10] http://www.alansondheim.org/

which always leaves a mark on the text, whether foregrounded in the poems or not. The same applies to storage and distribution: poets use what is available and economically viable to them. Some take special advantage of what each technology has to offer, incorporating its particularities into their poetry. Poetic creation with an inscription technology also is a tool for thought, as Loss Pequeño Glazier argues in the introduction to *Digital Poetics*:

> The poem is not some idealized result of thinking: the poet thinks *through* the poem. Similarly, investigated here is not the idea of the digital work as an extension of the printed poem, but the idea of the digital poem as the process of thinking through this new medium, thinking through *making*. As the poet works, the work discovers (6).

The digital computer and the Internet offer undeniably new technologies for poetry to be created, recorded, distributed and read. Poets who choose to explore its potential are pioneers in this new virtual frontier.

The Nature of the Electronic Text

At this point, it is necessary to explore the nature of electronic textuality. It is easy to forget that despite all the metal, plastic, and silicone that make the hardware, the computer is a machine made of words and numbers: the software that gives the computer its functionality. This software is pure language and orchestrated numbers put to work—seemingly endless lines of instructions that are read and interpreted by other assemblages of code. In “There Is No Software” Friedrich Kittler criticizes the orchestration between hardware and software, arguing that there is unnecessary “noise” in the execution of computer software and computers can be designed to achieve the same functionality with greater efficiency through hardware alone. This may be the case, but I argue that because computer languages are interfaces for people to create instructions for computers, and because these still retain strong connections to natural languages, this “noise” occurs partly at the level of language and it is therefore desirable—at least as far as literary and artistic endeavors are concerned. Writing code is as much an art as it is a craft because one can achieve the same results through different programming languages, or different procedures in within the same language. There is room for literary exploration in the space between pure efficient functionality and noise, as can be seen in the case of Perl poetry, poems written in the programming language Perl that are also executable programs. Every programming language has its own capabilities that are part of the palette of ideas a poet has at his/her disposal when composing an e-poem.

It is important to know how many layers of interpretation occur before an electronic text becomes readable for a human reader. At each level there is programming—thus an interpretive intervention, since a program is a set of encoded instructions (algorithms)—all of which interact with an electronic text to produce output documents. Ordinarily we only notice this when there are failures, misreadings, or misinterpretations that distort the output document, because most works are designed to achieve “immediacy” or transparency of the mediation.

The following list should help to categorize three main layers of programming and interpretation through which an electronic document needs to go before a human reader can access it. This is not a necessarily linear layering, since there are many feedback loops between them:

1. Presentation Layer—which includes what is displayed on the screen, played through speakers, and presented through other hardware devices.
2. Logical Layer—includes the software and hardware used to interpret the data layer in order to generate the presentation layer.
3. Data Layer—contains text, images, and other digital objects, as well as the programming codes, markup, and metadata that instruct the logical layer how to generate the presentation layer.

For instance, this document is stored in a Microsoft Word 2007 file format (with a .docx suffix), which contains not only the text and images of this study, but instructions on how to display this information for the screen and printer. The file constitutes the data layer of this document. For people to be able to access and read this electronic file with certain degree of reliability, they would need to have software compatible with this file format—ideally Microsoft Word 2007 (or 2008 with a Mac), though Microsoft Office 2003 with the conversion plugin would work, or they could use other word processing software, such as Open Office Writer, Google Documents, or Word Perfect. If the computer opening the document uses an operating system different from Windows, then the logical layer will have to interpret the data layer differently, like using a different font to present the text. Of course, the hardware used to run such software can also have an impact on how the logical layer interprets the data layer, as well as on the presentation layer. Hardware configurations such as screen type, size, and resolution, speaker type, printer type, and other variables all have an impact on the presentation layer and therefore how the reader reads these very same letters on this real or simulated piece of paper.

A Microsoft Word 2007 document is relatively simple compared to, say, a computer game, in which case the processor type and speed, amount of RAM in the computer, the storage device (hard drive vs. solid-state drive), and the type of graphics card all have an impact on how the game will perform in the presentation layer. This model becomes even more complicated when dealing with multiple networked computers and middleware, emulators, and other translating devices. When we consider the rapid pace at which software, hardware, data standards, and a myriad other factors change, it is clear to see why preservation and archiving have become important issues in the study of electronic texts.

All this code, software, and hardware add up to a considerable amount of collaborative writing and engineering: visions and revisions of algorithms and processes with a history that is sometimes documented in the programming code, sometimes archived by versioning programs, sometimes erased and lost. Writing in and for an electronic environment means joining a very long conversation that has been going on for endless hours and is recorded in countless lines of source code. Whether we perceive it or not, it forms an important part of reading electronic documents because what we read is shaped by that interaction.

The world of print is no different in the sense that there are many interpretations going on before a text reaches a reader in the form of a book or other print document. From writer, to editor, to all the people involved in printing a work, a work is read, reread, interpreted, translated and reproduced many times. When we acquire a printed document, it has a production history that has left an indelible mark upon the text. For that reason, every edition and reprinting of a work is carefully documented—it is its signature in time and space. Variability, however trivial or significant, happens with each reprinting, and sometimes within the same printing.

Given that textual variation is inevitable every time a text is produced and reproduced, we can see the difference between printed and electronic texts as one of acceleration of this mutability. Textual instability in the print world is a matter of months, years, centuries—the older the print history of a work, the greater the instability and variability. Electronic texts' instability is measured in seconds, minutes, hours, and days because every time a document is loaded onto a computer screen, it is being reproduced. For example, we can assume that two people having the same edition of a book means they have practically the same text. . . but that is not the case if they both read the same e-poem online from their own home computers, especially if they are from different platforms (such as PC, Mac, or Linux) or they are using different browsers (Netscape Navigator, Internet Explorer, Opera, Safari, Firefox, Chrome, etc.). It is comparable to having different publishers print the same work by an author: even if they are attempting to produce identical documents, the result is going to have differences, no matter if they are undetectable by the majority of readers.

That is why access to the source code of electronic documents is so important for the editorial (not to say bibliographical) study of electronic texts: this data layer contains the instructions for the production of the text and has not gone through a complex interpretation process. Of course, not everyone has the inclination nor interest in taking this kind of approach to electronic texts—nor do they need to—just as there are many readers and critics who have enjoyed and studied Emily

Dickinson’s poetry without access to the manuscripts. This approach doesn’t invalidate other ways of studying electronic texts, but its importance is foundational for their future study.

Code and Behavior

As readers and critics we have the accumulated experience of centuries of reading and theorizing about manuscript and print—to the extent that text has apparently become transparent and stable: comfortable to approach and read. It has been in the realm of textual criticism where this has been intensely debated by theorists such as Jerome McGann, Peter Shillingsburg, G. Thomas Tanselle, George Bornstein and others who have challenged our understanding of printed texts and shown us that we don’t understand the complexities of text as much as we may think. They devote much attention to graphical elements of texts, such as fonts, font sizes, spacing, and other formatting elements that are part of the experience of reading and analyzing a literary text. Thanks to their scholarly, editorial, and theoretical work, we have the critical terminology to describe written texts in print.

At the heart of the difference between print and electronic texts are differences in their codes. All documents can be described with graphical codes: font type, size, colors, spacing, and other descriptors commonly known as formatting. In print media these codes are actualized in the document, yet identifiable to computers and the trained eye. Electronic documents have formatting codes in the source document, but they can also contain codes that determine the behavior of the electronic text displayed. This is a crucial difference when these time-based codes are explored by a text in a computer. The complexities of text in print are complicated further by the capabilities of networked and programmable media.

For example, the source code may have instructions for the computer to display the words “they flee from me” in the center of a white page, in black 12-point Times New Roman font—something easily reproducible in a print document—but it may also be programmed to move the text away from the pointer on the screen, when approached. If printed out, the resulting document would have the same graphical codes, but would be unable to carry the behavioral codes. Would it be the same text in the two different media? Certainly not.

Electronic texts represent a literalization of the theoretical instability of all printed texts, as well as an acceleration of their capacity to change over time. Printed texts appear to be stable, constrained to remain so by the paper and ink material documents they inhabit, and their variability comes from authorial and editorial changes over different editions, the possibility of misprints, and the changes brought about by the act of reading itself. Electronic texts thrive on their capacity to change over time, at times at such a rapid pace that they exhibit what seems to be behavior—that is, the textual signifiers may change, move, react to the reader’s interaction, and/or be timed. This “behavior” is not an indication of textual agency, but of authorial, mediated, and/or readerly agency.

There is no agreed upon vocabulary for describing extra-linguistic characteristics of texts in digital media beyond the one already established for print texts. For that reason, I propose a typology of behaviors to describe textual characteristics possible in electronic/programmable media. The behavior characteristics and their values are the following:

1. **Static** texts are the default we are used to in print—they are texts that do not move on the screen.
2. **Kinetic** texts have words that move on the screen: this motion may be looped or linear, random, programmed, or responding to cues from the reader.
3. **Responsive** texts take advantage of the computers' interface to allow for input from the reader, which may come most frequently from the mouse and keyboard. The input cues (such as links, hotspots, and keyboard enabling) may be manifest or hidden, allow for voluntary or involuntary interaction, and have immediate or postponed reactions to the reader's input.
4. **Mutable** texts involve programmed or random changes and may also change due to reader interaction.
5. **Scheduled** texts may reveal themselves over time, which may be linear or looped; they may force a rate of reading by disappearing or scrolling; they may also trigger events over a programmed or random schedule.
6. **Aural** texts have a sound component, whether verbal, musical, or noise.

Building a typology is necessarily an exercise in personal observation and categorization—a perspective by which we can approach an e-poem or other e-text and hopefully gain greater insight about its meaning. This typology is neither prescriptive nor complete: it guides the reader's eye for detail and invites contributions of new observable behaviors. Such a tool can be useful to describe and analyze e-poetry, and will be tested in readings of Jim Andrews' work.

The focus on behavior as a distinctive feature of electronic poetry shouldn't be interpreted as trying to classify it as distinct from other poetic traditions. One could make the case that poetry and language are always behavioral. The "stasis" of written language becomes kinetic and aural when read, whether in the mind or out loud. "To articulate sweet sounds together"[11] is all about movement of the vocal tract, from lungs pushing air through the throat and out through the oral and nasal passages, made to vibrate with vocal chords, shaped with the tongue, palate, teeth, and lips. Language is also responsive to the environment they are placed in, as is the case of conjugation of words, and when a metrical pattern compels a reader to place the stress in an unusual syllable of a word. Poems can bring about such changes in a word's pronunciation, meaning, word order, and syntax, as exemplified by enjambment in open lines, that we can consider them nothing short of mutable. The words change in every performance, just as the meanings of words change through time. And what is iambic pentameter if not an example of scheduling language?[12] Meter has allowed poets to establish rhythms and carve out pieces of time that get translated into lines when recorded in writing. And (not) finally, even single words create whole environments, by evoking entire frames of reference (aka "schemas" or "conceptual frames") in the minds of readers. There is nothing that electronic poetry does that poetry hasn't done for centuries.

[11] William Butler Yeats, "Adam's Curse."

[12] Ezra Pound referred to iambic pentameter as "the metronome."

A poem can therefore be understood as a kind of software that runs on human beings and with codes make us produce sounds and silences, render images in our minds, quicken the timing of our hearts, and provoke many other changes in our bodies that we may not even be aware of.

So why embark upon an exploration of poetry in electronic media, if there's no difference? Because those operations that a poem's codes can provoke in a human being, are literalized in a computer. The programming codes are instructions for computers which we can see in action or read in the source code—and they are instrumental in the creation and production of poems in a medium that is new for poetry. This new set of codes, even when they do things preexistent in ancient poetic traditions, represent a new set of defamiliarizing[13] strategies for poetry, and draw attention to the mechanisms, technologies, issues, and pleasures electronic poetry has to offer.

This study is centered on Jim Andrews, a poet whose career is about engaging the expressive potential of language in different media and technologies, "old" and "new," aural and visual, traditional and experimental.

The Case-Study

What motivates a poet to produce poems that are unique to the computer? What about the computer and the Web encourage poets to publish poems that explore their

[13] "In studying poetic speech in its phonetic and lexical structure as well as in its characteristic distribution of words and in the characteristic thought structures compounded from the words, we find everywhere the artistic trademark – that is, we find material obviously created to remove the automatism of perception; the author's purpose is to create the vision which results from that deautomatized perception. A work is created "artistically" so that its perception is impeded and the greatest possible effect is produced through the slowness of the perception." (Shklovsky 19)

capabilities and potential? Wherever language exists, poets will be interested in exploring its expressive capability.

This study analyzes the work of Jim Andrews, a programmer and poet who works in programming languages (JavaScript, DHTML, Macromedia Director and others) to produce poems that test the limits of what is commonly considered poetry. He enjoys an international readership cultivated over the course of 15 years of consistent online publication and has had articles written about him in countries like France, Brazil, Canada, and the United States. He has received funding from both the National Endowment for the Arts and the Canadian government to work on his e-poetry. Jim Andrews is also a great example of a poet whose formation led him to take advantage of the rise of the World Wide Web and several of the technologies that emerged from it.

This focus limits the scope of this study in several ways. No single poet can accurately represent the broad spectrum of motivations, practices, and platforms that contribute to the ever-growing field of e-poetry. Other poets have entirely different approaches to e-poetry, focusing their attention on elements like the possibilities for collaboration in networked environments, three-dimensional spaces of VRML, the multi-media capability of the computer, the computer's ability to use randomness for textual generation or permutation, e-mail as a mode of publication, or the potential of scheduling tasks, to name a few. There are many platforms currently in use for e-poetry, each with its own history, strengths and limitations, just as there are some platforms that have become obsolete or are threatened by limitations and/or changing standards for computing. Rather than list the poets and platforms that will not be discussed in this study, a more reductive than productive approach, I will argue for the advantages of focusing on Jim Andrews.

Jim Andrews' poems take advantage of a variety of media, authoring programs, programming languages, and file formats to create poetic experiences worthy of study. Much can be learned about electronic textuality and poetry by following the trajectory of a poet and programmer whose fascination with language in programmable media leads him to distinctive poetic explorations and collaborations. Focusing my study on Andrews allows me to deepen my exploration of his poetry, motivations, inspirations, and poetics, while at the same time telling a piece of the story of the rise of electronic poetry from the mid 1980s until the present.

Jim Andrews is also a prolific writer of essays and forum postings about digital poetics and his work. He also corresponds with me, ever since we met during the E-Poetry 2001 Conference and Festival in Buffalo, New York. Since then, I've participated in discussions with him in a Yahoo Group he founded called WebArtery[14], I have invited him to chat with my students when we were studying his poetry, and we've developed a friendship and professional relationship. I correspond regularly with Andrews, and he has proven to be very generous with his answers to my questions, feedback on my writing, and has provided me with valuable archival materials.

The chapters on Jim Andrews can be categorized as a literary biography—a genre of academic writing that isn't as popular as it once was. Perhaps the shift away from biographical scholarship initiated by the New Critics' crusade against authorial

[14] http://groups.yahoo.com/group/webartery/

intent and completed by Poststructuralist "death" of the author writings are to blame for the decline of the genre. Perhaps the questioning of the canon contributed by shifting the discussion away from "great authors" to underrepresented writers of various nationalities, ethnicities, and genders. It is certainly not fashionable to place any writer on a pedestal in current critical practice (thought it's apparently acceptable to do so with theorists). So what are the benefits of taking a biographical approach?

First, it humanizes the subject—something much needed in the sometimes antiseptic setting of writing in digital environments. Secondly, it highlights a career path that leads to the practice of writing electronic poetry. Third, it legitimizes the field of electronic poetry by giving one of its most prominent practitioners a level of attention usually reserved for poets who are well established in the canon. Finally, it lays the groundwork and provides access to materials that could lead to further study of his work.

In addition to being a literary biography, the close readings of Andrews' poems are primarily formalist in their approach—that is, keeping their attention on the text and its source codes. This approach is the most compatible with the kind of media-specific analysis that Hayles calls for. Close attention to media without close attention to text and programming codes leads to writing around the texts but rarely about the texts.

With a field as new as electronic literature and poetry, it is important to remember that these are literary works, not specimens in some experiment. Someone put the time and effort to create a poetic experience that is particular to a medium that most of its users associate with work and/or play. This study seeks to model a close attention to detail in reading e-poetry that can lead to a greater appreciation of the work, as well as insights needed for sustained theoretical engagements. The biographical, technological, and textual groundwork established in Chapters 3 and 4 offer a wealth of materials previously unavailable and will be a major contribution for the future study of Jim Andrews' oeuvre

Mapping the Study

This dissertation is divided into five chapters, the first of which has defined electronic poetry, introduced a typology to aid in its analysis, and justified the need for a detailed exploration of Jim Andrews' work.

The second chapter, "Typing the Dancing Signifier," reviews the literature of textual and new media theories to establish clear differences between print and electronic textuality. It uses this to justify and describe the behaviors exhibited by electronic poems as textual characteristics—static, kinetic, responsive, mutable, scheduled, and aural— providing a genealogy for each behavior.

Chapter three, "Jim Andrews' Vispo(etics)," is a literary biography and a series of close media-specific analyses of some of his most important works. The literary biography focuses on his theoretical, poetic, and technological inspirations to establish his poetics. The close readings of his works focus on three areas: his visual poetry, his sound poetry, and his DHTML works.

Chapter four, "Mining the *Arteroids* Development Folder" examines the development of his e-poetry game *Arteroids,* and discusses the practical and theoretical challenges of creating an archive or a critical edition from the "*Arteroids* Development Folder." This folder contains all the versions, supporting documents, documentation, and other archival materials that went into the development of *Arteroids*. I conclude the study by discussing how this folder provides a direction for future research on Andrews' work.

This study works under the assumption that reading in any medium is a skill that takes years of training. Our training is mostly with print media (books, magazines, newspapers, etc) and screens (television, video games, film), but it is deficient when it comes to texts that explore the possibilities of the networked and programmable media. This study seeks to enhance its readers' vocabulary to analyze electronic poetry (and electronic texts in general) and a greater understanding of the poetic practices and trajectories of a major voice in the field of e-poetry. It also seeks to inform its readers of issues that affect a rapidly growing branch of literature—one that explores the expressive potential of electronic media. Electronic poetry is an ideal testing ground for language in a digital environment because poetry is the most compressed of all literary forms, potentially using every aspect of language available: graphical, linguistic, and behavioral.

Chapter 2: Typing the Dancing Signifier

> The future of textuality is not in special effects, the simple mechanics of letters dancing on the screen, whirling in dynamic display but in the practical and visionary reconceptualization of what constitutes the field of textuality as a realm of porous, multivalent, nodal and intertextual speculation, indeterminate and rich with potential to renew itself to the fullest extent of our critical engagement.

This excerpt from “Theory as Praxis: The Poetics of Electronic Textuality” by Johanna Drucker addresses a common misconception about electronic textuality—that it is no more than adding “bells and whistles” to the same plain old text we have known in print. Such misconceptions operate on a notion that textuality is little more than a sequence of words, and that everything else is external, accidental, replaceable. It would seem that current notions of textuality can be visualized as an onion composed of layers of different materialities encircling an immaterial center—with electronic textuality as the outer layer, and print, orality and thought being closer to its center and most important in the hierarchy of signification.

Concepts of poetry have long rested on this type of model, best exemplified by the definition of a poem as an arrangement of ideas captured in aurally charged sequences of words scored (written) on a page for subsequent oral reconstructions. It was against the rigidity of this hierarchical concept that the Concrete poets had to battle in order to create a space for their poetry that bordered the worlds of literature and the visual arts—and they were working with the same materials, ink on paper.

The new media that makes electronic poetry and textuality possible has reignited the struggle against this old model, partly because of its hierarchical nature, partly because its flaws become evident with the perspective gained from writing in new media. This chapter proposes that the differences between print and electronic textuality go deep enough into both practical and theoretical realms that reexamining textuality is necessary, because it is built on a foundation of notions specific to print, a foundation that does not support the procedural nature of electronic textuality.

In order to unravel the current understanding of textuality from the material specificities of print, we must first explore what are the materials of writing itself, once again with the help of Drucker's experience as a book artist and theorist. In *The Alphabetic Labyrinth*, she discusses how written language has historically used various systems: alphabetic (also known as phonetic), syllabic (where each symbol represents a syllable), logographic (where each symbol represents a word), and ideographic (where whole ideas and concepts are represented by a symbol) (14-5). This ancient technology has seen many different techniques and materials: carving, etching, painting, drawing, typing, and printing on stone, clay, cement, glass, papyrus, paper, magnetic media and computer screens. Each material and technology used to inscribe, store, and disseminate texts places its distinct characteristics at the disposal of writers. This study is interested in a writer who engages the particularities of language as produced, stored, and disseminated in standalone and/or networked computers as a meaningful element of his poetry.

But what are these particularities, exactly? One way to establish them is by juxtaposing electronic media with a medium that has dominated textual production, storage, and reception for the past several centuries: print. How is electronic media different from print media? These questions have been central to discussions in the fields of New Media Writing and Textual Criticism since the early 1990s, even though they emerge from issues raised by Poststructuralist and Bibliographical theorists decades before. The following section will establish useful differences between paper and electronic media, arguing that a distinctive feature of texts in digital media is that they can exhibit what can be described as behavior, and proposing a typology of this characteristic.

The discussion to come in the rest of this chapter arises out of my perspective on the ontology of literary works (especially poetry). “Where is the poem?” is a complex and very old question that has no clear answer, with various valid positions that may have more to do with a person’s philosophical stance towards the materiality of the self, than with any technology, media or materiality. Here are some positions that inform my own perspective.

1. Intentionalist textual theorists believed that the work was a conceptual thing created by a writer that consisted of a text or set of variant texts, captured imperfectly by documents, therefore requiring an editor to sort through the documents and textual variants to reproduce (or produce) the text that best captured the intended work. These theorists considered text to be immaterial, consisting of a particular word sequence that could be recorded in a variety of

media (voice recording, manuscript, or print publication) without affecting the text itself.[15]

2. Jerome McGann and others in the field challenged the notion that limited the scope of what constituted a text only to linguistic codes (words in a particular sequence) and made a strong case for preserving extra-linguistic codes such as font and formatting because they potentially carried valuable information. This argument, in combination with the notion that published texts are the result of a series of social transactions between writers, editors, compositors, and other parties, had the result of combining the "immaterial" text and with the materiality of documents.[16]
3. Another position in this discussion sustains that the work and its text are what occurs when a reader interacts with the document. From this perspective, the writer is considered a reader like any other, who has an understanding of their work that changes over time, and therefore the revisions that they might make to the text constitute the creation of new works, even if they go by the same title.[17]

[1515] For example, in *A Rationale for Textual Criticism*, G. Thomas Tanselle distinguishes between "the texts of works" and "the texts of documents" in the case of a typo by the writer. It is the editor's job to decide whether a misspelling was intended or not, and therefore whether it should be preserved in future editions.

[16] In "Letter-Poem: A Dickinson Genre," Martha Nell Smith demonstrates how editorial interpretations shape the publication of three manuscripts by Emily Dickinson, and how each publication could be considered different texts and even different genres.

[17] In *The Pleasure of the Text*, Roland Barthes, writes about how readers reconfigure the text by skimming or skipping through passages ("tmesis"), shaping their very conception of the work.

There are other stances on the ontology of works texts and documents, as well as combinations of the ones discussed. So where *are* the poem and its text? For me a poem and its text are conceptual things that react with whatever materials they are performed in, whether a mind, a voice or audio recording, a stage performance or video recording, handwriting, typing, or printing on a page, or a computer. Variation is inevitable when a text becomes performed, which includes the creative moment when a writer first conceives of it and records it in a document. Every interaction with the document— by the writer, editors, or readers—represents a performance and reinscription of the text and therefore variation. From this perspective, publication can be understood as an attempt to reproduce a document many times with a high degree of fidelity (since variation is inevitable) for distribution to an audience—and then the variations become exponential as every person that reads with the document performs and creates the text and work with their bodies and in their minds.

The work emerges from this cloud of textual and documentary variations. The tendency of this cloud is to grow and drift away from the original documents, texts, and conceptualization of the work as others take ownership of it.[18] Some textual critics see their role to select and produce documents and editions that strive to reproduce what the writer conceptualized as the work (a.k.a. intention)—an attempt to reduce the cloud. Others seek to validate this cloud and revel in its multiplicities. I see the role of literary critics and educators to choose documents to analyze and discuss

[18] Publicly cherished though misquoted lines are an example of this phenomenon, as can be seen with the examples of “Beam me up, Scotty” (never spoken in the original Star Trek series but later included in one of the motion pictures with a wink and a nod), “Luke, I am your father” (Darth Vader’s reply to Luke’s accusation of killing his father: “No. I am your father”), and “Play it again, Sam” (Four words never quite uttered in *Casablanca*, but immortalized in Woody Allen’s movie of the same title).

with their own audiences attempting to direct the multiple conceptualizations of the work in insightful directions.

This study seeks to examine how poems are shaped by and react with the standalone and networked digital computer by: examining the particularities of electronic textuality in this chapter, reading deeply into a large sample of Jim Andrews' work in the next chapter, and considering the challenges of archival work with electronic documents in the final chapter.

Print and Electronic Textuality

Some theorists may question the value of establishing differences between print and electronic textuality, a point that has become almost taboo in the current media environment. I believe it is a distinction that needs to be addressed; if only because so many of our notions of textuality, writing, and reading are shaped by our experience and understanding of print that they become stumbling blocks when exploring the potential of literature in electronic media. Delineating this boundary is also essential for this study because its definition of e-poetry requires the exploration of the capabilities of electronic media. I therefore venture into this theoretical minefield guided on one hand by an awareness of the traps many New Media Writing theorists have fallen into, and knowledge of the ground mapped by Textual theory on the other.

The issue has a history that started in the late 1980s, when encouraged by the proliferation of the personal computer as a writing space and armed with Poststructural theory, early New Media Writing theorists (then called Hypertext theorists or the digerati) were perhaps hasty in seceding from the world of print, making claims about the uniqueness of electronic media and hypertext.

Unfortunately, they seemed to be unaware of the sophisticated understanding of textuality developed by Editorial and Bibliographical theorists. They are not really to blame—at the time editors and bibliographers were busy debating issues of authority, materiality, and socialization of texts, ultimately deconstructing the notion of "definitive" editions—issues that seemed to belong to the dry and dusty world of manuscripts, old books, and archival scholarship. But it didn't take long for these newly reorganized Textual theorists to point out the fallacies in these original claims by showing that, regardless of media, texts have always been complex, non-linear, dynamic, and inviting reader interaction. Subsequent generations of New Media Writing theorists joined in what has become an almost ritual bashing of the early Hypertext theorists, and have since been very careful when touching on distinctions between print and electronic media to avoid the same fate.

Theorists from these two fields are not really engaged in an academic turf war, but I do think that they are somewhat entrenched in their positions and are limited in their forays into what is considered part of the other field. Textual theory has developed a sophisticated vocabulary when it comes to textuality from its long history of working with texts in manuscript and print, yet many current New Media theorists find it unsatisfying, unnecessarily complex, and mostly inadequate to describe electronic textuality. Perhaps it is because this vocabulary was developed in and for an age of print, that it is laden with print-based connotations and assumptions and that keep it unwieldy for describing electronic texts. At a glance there are three dominant critical perspectives in relation to this issue:

1. Textual theorists like Peter Shillingsburg, Jerome McGann, and others who use their model of textuality in their online archival projects, such as The Rosetti Project, The Dickinson Electronic Archives, The Blake Archive, and other excellent resources. Their "acceptance" of the model is by no means blind, however: much debate arises from their editorial work both in print and online.
2. A few theorists like Johanna Drucker, N. Katherine Hayles, and Matthew Kirshenbaum are conversant in both Textual and New Media Writing theories and their work informs and questions the established textual model, sometimes proposing new textual models.
3. Most current New Media Writing theorists prefer to create terminology and define new fields like Espen Aarseth with his book *Cybertext*, Gonzalo Frasca with ludology (game theory), Jay Edgar Bolter and Richard Grusin with *Remediation*, and others.

This chapter will explore these divergent perspectives in an attempt to strengthen the conversation between these schools of thought and hopefully find common theoretical grounds for the exploration of texts in electronic media. For my own contribution to the debate, I will examine the differences between print and electronic media in order to engage the uses and limitations of the textual models put forth by Peter Shillingsburg and N. Katherine Hayles, leading to a justification and description of a typology of behaviors that can be exhibited by texts in electronic media.

Defining Boundaries

We are never more aware of differences between print and electronic media as when we encounter limitations in one that are not present in the other. For example, my 2004 E-Mac used the Jaguar operating system (Mac OS 10.3.9), which placed 145 fonts at my disposal—a rich palette for my writing needs, especially if I wanted to print the results. If I wanted to create a document for publication on the Web, however, the range of available fonts would decrease dramatically because other operating systems have different fonts installed and only have a handful in common.[19]

These practical concerns raise theoretical issues relevant to this study, the first of which is the difference between how electronic and print documents represent text. How deep do these differences really go when it comes to notions of textuality? Some theorists assert that there is no fundamental difference between a text and the media in which it can be created, produced and received, while others argue that a new textual paradigm needs to be developed to account for other media because established notions of textuality are too dated and print-based. One of the voices for the latter argument is N. Katherine Hayles.

> As critics and theorists encounter these works, they discover that the established vocabulary of print criticism is not adequate to describe and analyze them. The language that electronic literature is creating requires a new critical language as well, one that recognizes the specificity of the digital

[19] There are ways around that, such as embedding fonts, using Flash, or transforming text into images of text—all of which complicate text with licensing permissions, large file sizes, higher production costs, compatibility, searchability, scalability, printing, and programming knowledge, especially if you want to do something more complex with the text than display it in the font of your choice.

> medium as it is instantiated in the signifying practices of these works. This new critical vocabulary will recognize the interplay of natural language with machine code; it will not stay only at the screen but will consider as well the processes generating that surface; it will understand that interplays between words and images are essential to the work's meaning; it will further realize that navigation, animation and other digital effects are not neutral devices but designed practices that enter deeply into the work's structures; it will eschew the print-centric assumption that a literary work is an abstract verbal construction and focus on the materiality of the medium; and it will toss aside the presupposition that the work of creation is separate from the work of production and evaluate the work's quality from an integrated perspective that sees creation and production as inextricably entwined ("Deeper").

The last part of this argument refers directly to Peter Shillingsburg's model of textuality presented in the chapter "Text as Matter, Concept and Action" published in his book *Resisting Texts*. This model describes three levels of performance: creative (in which the writer composes a text), production (in which the text is prepared into documents for distribution to readers), and reception (where each reader interacts with the document to access the text). These levels are not mutually exclusive and they can blend in complex ways, but examining how each of these work in print and in electronic media brings some crucial differences to the forefront. Exploring hypothetical performances in both print and electronic media should uncover these differences.

Before juxtaposing the performances in the two media types in a table, let me point out the boundaries Shillingsburg makes between the performances and add N. Katherine Hayles' concept of inscription technologies to the model. The reason for the addition to Shillingsburg's model is that it doesn't account enough for the creative contribution of the production (read inscription) technology. Shillingsburg describes the difference between the creative and production performances is that the former results in a work, while the latter results in a physical document. The writer may create a number of documents during this creative process, each of which may capture a version of what eventually is published as a work capturing an intention (or matrix of intentions) from the author(s).[20] Thus, the text is materialized by the document and represents a version of the conceptual work. Shillingsburg makes the distinction that a typo in the manuscript is a production error, not a creative one, and can therefore be corrected in subsequent production performances (76).

The following table juxtaposes creative, production and reception performances in both print and electronic media.

Print Texts	**Electronic Texts**
In the *creative performance* a writer uses one or more inscription technologies—pen & paper, a typewriter, a personal computer, tape recorder, and so on—to create a literary work. Inscription technologies can transform the creative process and resulting work, as is evident in William Carlos Williams, E. E. Cummings, and Charles Olson's use of	The *creative performance* requires the writer to make decisions about the production of the work from the outset, since different software allows for different output. The inscription technologies consist of at least two parts: hardware and software, with varying degrees of difference between them. Thus the creative and production performances

[20] What constitutes a "work" is the subject of much debate among textual theorists, particularly as it relates to the material documents that represent it. For the purposes of this study, I will use Shillingsburg's definition of the work as "a mental construct that can be known only through its physical forms and the effects they create or allow" ("Text as Matter" 67).

the typewriter in their poetry, blurring the distinction between creative and production performances.	are intertwined.
The *production performance* that materializes a work in a material document (such as a manuscript or published book) ranges from intimately individualistic to highly collaborative. This leads to choices of inscription technologies or publishing companies employing editors, letterers, compositors, printers, binders, and other professionals in the field of printing. Preparing the text for publication requires decisions on the appearance of the text: formatting and graphical elements, such as font and font size, page layout and design, use of photos and/or illustrations, and so on. The resulting print document embodies its production performance, as well as a sense of authorial intent at the time of publication.	The *production performance* is interconnected with the creative performance: for instance, when a writer wants words to move a certain way on the screen, programming them to do so becomes an integral part of creating the work. By the same token, the way the work is made accessible becomes part of its production history and context, with the end result of a file being made available for a reader to access. At this point the authorial side of the production performance is over, but the overall production performance is incomplete until the reception performance begins.
The *reception performance* begins when the reader operates the printed document (such as a book) by turning pages, scanning words inked onto their surfaces with his/her eyes, making sense of the visual information to access the linguistic meanings of the text. This performance shapes the text that the reader receives.	The *reception performance* begins with the completion of the production performance: when the reader's computer accesses the file. The computer, a varied patchwork of hardware and software, interprets the file –keeping a version compiled in its memory for access and manipulation during the reception performance –and produces an output version of the work for display on its hardware devices (screen and/or speakers). This is the material document that the reader interacts with during the reception performance. If the file has been programmed to register and respond to user input or it has randomly generated elements, then parts of the production performance are ongoing during the reception performance.

Table 1: Creative and Production Performances

Hayles' call for "an integrated perspective that sees creation and production as inextricably entwined" is more than justified, in both print and electronic media. This is true not only of new inscription technologies (e-poets) and deliberate explorations of more familiar technologies (Concrete poets and book artists), but also when the inscription technology has become so self-evident and naturalized that it is transparent to the reader. Production aspects that are taken for granted shape creative performances in perhaps deeper ways than in overt explorations because they are internalized into the creative performance.

> The space of the page has long been taken for granted as blank, while text is valorized as the agent of signification. But what is the space of the page? What are its architectures which quietly construct the possibilities of a text? As spaces for writing multiply, perhaps infatuation with literary style will be replaced by the stylistics of the page, and a desire to create mechanisms that offer new spaces for writing (Soderman). [21]

This excerpt is from the opening page of the Machine Poetics Page Space Project (2004), which paired programmers who created electronic spaces with writers that create texts that take advantage of the spaces' capabilities in a truly intermeshed combination of production and creative performances. Soderman's questions may lead us to reexamine more conventional inscription technologies, such as paper, ink, typewriters, and word processors. The putative emptiness of the blank page is a highly codified, conventionalized and politicized space already charged with a production history and with material characteristics that cannot help but affect the way a writer interacts with it to create/produce a work. The dynamics of composition are different—I am always very aware of the finality of written word when using

[21] http://machinepoetics.com/page_space/

paper and pen: a written word is there to stay, whether desired or not, which can lead me to compose different sentences from what I would have written in the easily modifiable space of a computer. The historicity of the moment and its material conditions are part of the moment of creation, whether the writer is aware of it or not.

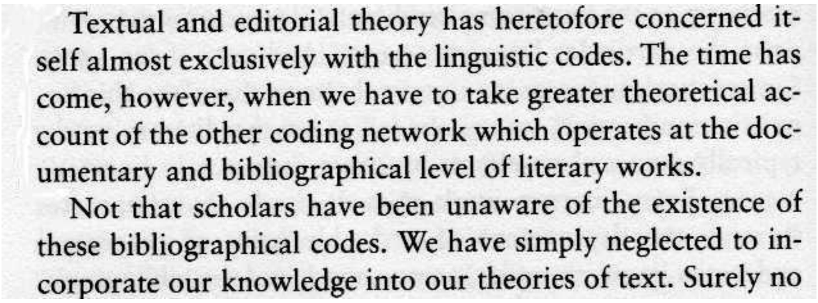
Shillingsburg's distinction between creative and production performances is useful for editorial theory, particularly as a tool to sort through multiple publications and manuscripts of a work in an attempt to separate authorial and editorial interventions. For these purposes it is convenient to conceptualize text as immaterial, with the ability to take on different materialities without losing or changing its essence. This performance model seems to be most useful when dealing with the ubiquitous book, where one could argue that the three performances are somewhat distinct. The most blending occurs between the creative and production performances, especially when the writer is involved in the production of the published work, as was the case with William Blake, William Morris and book artists, among others. Through this model, the reader's involvement is kept at a safe distance: a reception performance which is distinct from the creative and production performances and leaves the text intact. The inadequacies of this model emerge easily when considering notions of materiality, discussed so well by critics like Jerome McGann, Johanna Drucker, and N. Katherine Hayles, but they stand out further applied to electronic media. To prove this, let's examine the most clear-cut of Schillingsburg's boundaries: the distinction between production and reception performances.

Production and Reception Performances

When a reader picks up a material document, such as a book, a legion of theorists such as Iser, Barthes, McGann, and Drucker have taught us that a highly creative performance is about to begin. There is nothing passive about the act of reading, not physically, not mentally. I will discuss some ideas put forth by these four theorists, because each adds concepts that will help us explore the boundary between the production and reception performances.

In *The Act of Reading* (1978), Wolfgang Iser develops a theory of interaction between readers and texts in order to create literary works, which are necessarily virtual. In other words, the work is a concept that emerges from an interaction between the reader and the text created by an author. The act of reading is understood as a psychological process by which a reader engages the linguistic, narrative, and other structures present in the text. I proceed from this brief summary of an elaborate model to point out that the structures described by Iser result from a reader's engagement with the linguistic dimension of text, not with the material characteristics of the documents that contain the text.

Jerome McGann's distinguished career as an editor, writer, and theorist has been built largely on the concept of the materiality and socialization of texts. In *The Textual Condition* (1991) he argues that:



> Textual and editorial theory has heretofore concerned itself almost exclusively with the linguistic codes. The time has come, however, when we have to take greater theoretical account of the other coding network which operates at the documentary and bibliographical level of literary works.
>
> Not that scholars have been unaware of the existence of these bibliographical codes. We have simply neglected to incorporate our knowledge into our theories of text. Surely no (78).

McGann has described in great detail the graphical and bibliographical codes as part of what he calls the socialization of texts, with the result that textual models have been modified to incorporate material and bibliographical codes, as Peter Shillingsburg does in “Text as Matter, Concept, and Action.” Literary theorists and critics like Iser consider primarily the linguistic codes that make a text, ignoring or minimizing the impact of the other codes. The document which contains a text contains information and provides an interface that shape and affect the reception of linguistic codes. This concept has been built upon by theorists such as Johanna Drucker and N. Katherine Hayles, among others, exploring the material characteristics of formats such as the book, hypertext, or works in new media.

For instance, when operating a “traditional” (codex) book, one begins in the first page and scans the words on each page, from left to right, top to bottom, turning the page upon reaching the end of it. The reading may take place over the course of several sessions, of varying length to complete reading the book. At times, the reader may skim or skip entire passages or pages (Barthes concept of “tmesis”[22]) reread several parts, have different types of reading contexts and states of mind—all of which are part of the reception performance. Because no two reception performances are the same no two interactions with the same document will yield the same text, proving a point made by McGann and other textual critics: that no text is self-identical.

Even though this is true of all media, most of the codes described by McGann are in the context of print documents, with complete production performances that

[22] *The Pleasure of the Text*, pg. 11

crystallize a series of social interactions and intentions. When it comes to digital media, he acknowledges that “Aarseth’s and Murray’s views about the differences between traditional and cyber textualities are common and widely accepted. That fact underscores the need for a thoroughgoing retheorization of our ideas about books and traditional textuality in general” (*Radiant Textuality* 148-9). He goes on to argue that traditional text is algorithmic in character, concluding that “a text is a display and a record of itself, a fulfillment of its own instructions” (151). I agree with McGann’s points insofar as they are built on an intimate understanding of print and its conventions and tremendous experience in creating the Rosetti Archive in digital media, but questions arise when considering them in the context of electronic media, particularly in the context of this boundary between production and reception performances.

This boundary is much more permeable in electronic media, partly because the production isn’t over until completed by the reader’s computer, partly because its time-based nature allows it to have a whole different class of codes that generate what I call behavior. In the opening chapter of this study I claim that “electronic texts represent a literalization of the theoretical instability of all printed texts, as well as an acceleration of their capacity to change over time.” Texts are dynamic entities that emerge from human interactions with inscription and storage technologies, no matter whether these are analog or digital, paper or magnetic media. These human interactions are always in flux, as Johanna Drucker points out in *Sweet Dreams*.

> No work of fine art is ever finished, never in a condition of static completion. We intersect with works of art in a specific historical moment, our own, even

> as a work's capacity to elicit response changes as it moves through a historical continuum.

This theoretical instability Drucker describes is made literal by the computer joining the reader in the reception performance. A computer needs to interact with an electronic text long before a human can and its "specific historical moment" is very relevant, particularly when compatibility and obsolescence issues arise. A printed document embodies an entire production performance and history in materials that age at a much slower rate, and can last centuries, while the threat of obsolescence for electronic texts is very real and the range of compatibility of a software type or programming language can be limited to a few years.[23]

The second factor that blurs the boundary between production and creative performances is the behavioral codes. Because the computer is a time-based medium (or conglomerate of media), it is able to have feedback loops that can factor in and respond to input from the reader's reception performance or delay the completion of aspects of its production performance until certain events are triggered over the

[23] The preface of the Electronic Literature Organization's *Born-Again Bits* testifies to this issue:

> Acid-Free Bits by Nick Montfort and Noah Wardrip-Fruin (June 2004) was the first publication on digital preservation to emerge from the Electronic Literature Organization's Preservation, Archiving, and Dissemination (PAD) initiative. Addressing primarily the community of electronic literature authors, it concentrated on prescribing standards and best practices that creators can follow to prepare for "keeping e-lit alive."
>
> With the release of Born-Again Bits, ELO continues the argument by envisioning a technical framework that can not just keep e-lit alive but allow it to come back to life in new forms adapted to evolving technologies and social needs. The intended audience of Born-Again Bits includes besides e-lit authors also the publishers, archivists, academics, programmers, and funding officers who will be necessary partners in an overall, renewable ecology of electronic literature (Liu)

course of a reception performance. An example of this is Jim Andrews' *Stir Fry Texts*, in which the reader's mouse movements over the displayed text mixes it with several hidden texts, producing unique textual combinations in the displayed document.[24] This happens quite literally in the virtual text that is formed in a reader's mind from the interaction with a document—a process that also happens in the *Stir Fry Texts*—but the interaction affects the document as well, which is not the case in a print document. A typology of these behaviors will be fully developed in part 3 of this chapter.

Clearly electronic media work differently from print media in the way that texts are produced and accessed. I believe Shillingsburg's textual model is useful to distinguish between the endeavors of writers or groups of writers, those of publishers, editors, typesetters, compositors, printers and the rest of the collaborative team that produces a book or other printed document, and that of readers. As stated earlier, its limitations emerge when faced with questions of materiality, because it is based on a notion of immaterial textuality. This model directs attention to crucial differences between print and electronic media, however, paving the road for a more thorough exploration of what electronic textuality has to offer. Before embarking on this exploration, I must at least touch on the two performances considered to be most distant in Shillingsburg's model: the creative and reception performances.

From Creative to Reception Performances: A Holistic Look

Whether a writer is interested in exploring linguistic and/or extra-linguistic codes, the materiality of the medium a text is produced in, the reader's reception performance,

[24] http://www.vispo.com/StirFryTexts/index.html

or any other element of the infinitely complex communication process or not, these elements will have an impact on the text(s). So how do considerations of production and reception performances affect the creative process? Both print and electronic media offer a varied palette of possibilities, each with its history and conventions.

There are abundant studies and resources on writers who integrate production and reception performances in their creative work: Emily Dickinson's handwritten letter-poems explored in the *Dickinson Electronic Archives*, William Blake's colorful poem prints showcased in *The Blake Archives*, a legion of writers/book artists studied by Johanna Drucker in *The Century of Artists' Books*, concrete poets restored to circulation in *UBU Web*, and the latest writers/programmers to publish their works online and index them in the *Electronic Literature Organization*, to name a few. These studies not only highlight how these writers take control over the means of production to create interfaces for readers to explore their creations, but they also point out how the very same formats that both writers and readers have internalized and made transparent have an impact in the both creation and reception.

For instance, a poet writing for publication in a traditional book format (the codex) adopts a series of conventions that have dominated literary history for the past few centuries. Some of these arise from the development of the printing industry, including print technologies and the division and specialization of the labors involved in bookmaking. They also emerge from the growth of markets for the printed word in a variety of formats (books, newspapers, magazines, etc.), which requires regularized training of readers from an early age. In *Cybertext*, Espen Aarseth coins the term "ergodic literature" to refer to literature that requires a non-trivial effort to traverse.

The opposite, non-ergodic literature, is therefore literature in which the materials, interface, and other mechanisms have become so customary for both readers and writers that they are virtually transparent, and are considered unimportant or trivial. These procedures become so customary for both readers and writers that they are taken for granted, freeing them to concentrate on what the industry allowed them to concentrate on most: the bibliographical and linguistic codes of a text.

Separating the performances, as often happens in the traditional book model, can be freeing for a poet: by not needing to worry about aspects of production he/she can really concentrate on the aural, visual, and semantic elements of language. A conventionalized reception performance is also freeing, allowing the poet to focus on arranging the order of poems in the book, to create a sense of a beginning and an end for readers that follow these reading conventions. Most of the great poetry and literature has been written within these conventions, and it is understandable for most editorial theorists to deemphasize the importance of elements considered to be external to the creative intentions of the author. But to build theories based on established conventions is misguided because it disregards the many writers who take control of the means of production to create works that are meaningful in their variations on established conventions, or works that explore new rules for reading, challenging readers to commit a non-trivial amount of effort to traverse them. The past century has placed amazing production tools at the disposal of writers with the result of more integrated creative, production and reception performances, not to mention entirely new formats, media and contexts.

The decision to write an electronic poem is a decision to consciously engage some of the special time-based capabilities that the computer has to offer. Writers of e-poetry and other e-texts cannot avoid thinking about these performances holistically when they create their works, because the technical capabilities and challenges of any inscription technology are integral parts of the creative design of the work, as well as how the work will be reproduced and interacted with by the readers and their computers. The split in the production performance of electronic texts pulls the three performances together in a way that is difficult to ignore, be it by writers, critics, or readers. Even though familiar pre-designed spaces are certainly available, a writer of e-poetry usually needs to design the interface as a fundamental element of the poem.

"The computer, more than just a word processor, is a design tool: it is an instrument for crafting writing environments." I would add to Kirschenbaum's point that it also serves to craft reading environments, a point I believe he would have little difficulty agreeing with. The choice of the word *environment* evokes a sense of the ecologies that the words exist within, and the forces that can be brought to bear upon them, as well as the roles these words can inhabit in these spaces. Consider how a writer that writes in the ready-made environment of ink on paper is putting together words in a space with predetermined factors that affect the linguistic text in predictable ways. Any variations upon the culturally determined ecology of such a writing space fly against a reader's expectations of how to execute the algorithms of the page and can be used to direct a reader's attention to a particular element of the text, or to redirect the reception performance. A poet that uses a computer to design a writing and reading environment creates a space for text where the potential interactions between its elements—linguistic codes, textual appearance and behaviors, and interface—are open to the poet's needs for expression. There is a much bigger palette of options placed at a writer's disposal, for better or for worse.[25]

In other words, a poet writing for print media must think about word *selection* (with semantic and phonetic considerations), *sequence* (morphological and syntactical considerations), *appearance* (including graphical and bibliographical codes), and *interface* (conventional or repurposed for the poem). A poet writing for electronic media must add *behavior* (what the words do and under what conditions) to all the previous considerations.

The complexities of textuality may never be explained to the satisfaction of all theoretical concerns. Models such as Shillingsburg's are informed by wisdom gained through years of rigorous scholarship in dealing with manuscript and print documents. Thus, the literary work goes from being a thing of the writer's mind to gaining the materiality of a document to becoming another mental construct, this time in a reader's mind. Text is a thing of the mind—a biological materiality so individual that it might be called immaterial—that can be shared only through material media, whether it is a book, an electronic document, a live or recorded voice, or some other production.

[25] I am in no way privileging electronic media over print media as far as its creative, expressive potential is concerned. In the long history of poetry, some of the best poetry has developed under formal constraints, such as meter, rhyme, closed form, and others, while much bad poetry has resulted from a sense of freedom from the same, as was the case with free verse. Ezra Pound and T.S. Eliot both bemoaned the overabundance of what they considered poor free verse, which led Eliot to state that "No verse is free for the man who wants to do a good job" ("The Music of Poetry"). Poets should exercise care and restraint in the crafting of poetic environments.

Material and Emergent Texts

> In a world rife with unsolicited messages, typography must often draw attention to itself before it will be read. Yet in order to be read, it must relinquish the attention it has drawn. Typography with anything to say therefore aspires to a kind of statuesque transparency (17).
>
> Robert Bringhurst, *The Elements of Typographic Style*

Common sense tells us that in order to read a text we must first be able to see it, yet most of the time we are unaware of the visual materiality of the texts we read. It takes very little time for the eyes to identify a word to access its sound and meaning, moving along to the subsequent words on the document, because we are not really observing the words in detail. Our eyes scan words on a page or screen and convert them to thought, voice, or both without much thought given to the font, font size, color, formatting, or other physical characteristics of the text. Bringhurst's notion of a "statuesquesque transparency" is compelling because it highlights both the "concrete" visual materiality of printed words and the way readers read *through* them to access their linguistic information.

A more readerly perspective from someone equally concerned with producing texts is the following: "All writing has the capacity to be both looked at and read, to be present as material and to function as the sign of an absent meaning" (Drucker, *Figuring the Word* 59). For the purposes of this discussion, this study will use Drucker's distinction between *reading*—accessing the sounds and meanings of written language—and *looking*—accessing non-linguistic visual information, whether present in writing or other types of images. The differences are clear at the level of ocular motion: saccades (minute eye movements) are directed differently when looking and reading. When looking at something, our eyes are directed by what captures our attention. Reading is a more disciplined eye motion—the result of years of training and practice in scanning a written surface from left-to-right, top-to-bottom in order to apprehend linguistic and bibliographical codes. But is that all we can find when we go beyond the material surface of the text?

As discussed earlier, our training and experience with writing is based largely upon a notion of the word as being immaterial and therefore easily translatable to manuscript, print, speech, voice recording, and so on, without it being affected in meaningful ways. Theorists such as McGann, Drucker, Hayles, Kirschenbaum (in books with titles such as *Black Riders: The Visible Language of Modernism, The Visible Word, Writing Machines* and *Mechanisms)* have questioned such notions of transparency, demonstrating time and again that the materiality of the written, displayed, or otherwise recorded document is meaningful and must be considered along with the linguistic text when reading and looking at a text. At this point, the notion of materiality must seem almost commonsensical when dealing with ink-on-paper documents, but how does this hold up when dealing with texts in electronic media?

Early hypertext and new media theorists trying to distinguish between print and electronic textuality often set oppositions between the materiality of print and immateriality of electronic texts, often making claims that print texts were “static” and “stable” while electronic texts were “dynamic” and “unstable.” As discussed in part 1.0 of this chapter, it didn’t take long for this to be debunked by both editorial and textual theorists, as well as the later generation of new media writing and cybertext theorists. Two recent examples emerge to counteract the putative immateriality of electronic texts: Matthew Kirschenbaum labels this distinction the “tactile fallacy”[26] (if you can’t touch it, or manipulate it directly, it must be immaterial) and Nick Montfort calls it “screen essentialism” (thinking that the text is merely what one sees on a computer screen). To focus primarily on the materiality of the displayed document, without considering the materiality of the document in magnetic storage that is manipulated by the computer’s hardware and software, is to leave out a durable and essential part of the electronic text. Kirschenbaum in “Every Contact Leaves a Trace” and “Extreme Inscription” as well as “Hacking ‘*Agrippa*’: The Source of the Online Text” assesses the impact of the hard drive and other forms of magnetic storage media partly in order to counteract the rhetoric of immateriality used to describe electronic media, partly to fortify the theoretical base of New Media theories by placing them in the context of its foundational technologies.

For example, let’s take the idea of the text displayed on the screen as an image. Our understanding of the materiality of printed texts –as informed by a legion of theorists such as Drucker and McGann—is that the texts we see on a document are images that capture linguistic, graphical and bibliographical codes. Is that the same as the text we see on a screen? Yes and no. From a purely human reader-centric vantage point we may not be able to tell the difference between an image of text and a text displayed as an image. “As computational data structures, images differ radically and

[26] "Editing the Interface: Textual Studies and First Generation Electronic Objects." *TEXT: An Interdisciplinary Annual of Textual Studies* 14 (2002): 15-51.

fundamentally from electronic text," is a point Matthew Kirschenbaum demonstrates in "The Word as Image in an Age of Digital Reproduction." Some of the differences he discusses are in file size, searchability, and the types of operations and transformations they can undergo. It is this division between human and computer readers that I wish to discuss and how it transforms the notion of electronic textuality.

In my discussion of Shillingsburg's model of creative, production and reception performances, I was able to show how the production performance is not over until the reception performance begins by having a human reader open the file that contains the electronic text. A different perspective of this split can be achieved by considering "the reader" as an entity consisting of computer and biological parts—a cyborg reader, if you will. This perspective accounts for a reader who can read the dual materiality that constitutes an electronic text: the source document and the document(s) generated literally when the source document is opened or run. From this vantage point, we can consider the source document to have a complete production history (which it does) and the reception performance includes both the reader, his/her computer, and possibly the network they are connected to.

The source document is an electronic object that contains data and instructions in a programming language that govern how its data will be displayed (appearance), as well as how its components will act and react to input during the time it is displayed (behavior).[27] This object is stored in some form of optical or magnetic media and as such can be manipulated in many ways before it is even activated for

[27] Sometimes the data and instructions are in separate files, as is the case of dynamic Web pages, where the data can be assembled on the fly from multiple sources and formatting itself comes from a different file, an interface known as Cascading Style Sheets (CSS).

display to a human reader: it can be copied, compressed, decompressed, attached to other electronic objects, transmitted across a computer network, searched and deleted, to name a few operations. [28] The storage medium, such as a hard drive, floppy disk, or flash card can be erased using powerful magnetic fields (degaussing) or destroyed through physical means as is necessary in the case of optical storage media (such as a CD or DVD).[29] It can be accessed in at least two ways: as source code and as an executed program. When opened it undergoes many translations into different computer languages (as discussed in section 2.0 of Chapter 1), all of which are interpreted in recursive feedback loops by various software and hardware components that constitute a computer until a document is produced for display on a screen, along with other output options, potentially including sound, printouts, and so on. If the file was opened for display as code (using as an example an HTML file), it is displayed as a text file without executing the instructions provided; if the file is executed, then it should produce output that follows the instructions specified in its code.[30]

The source code itself is what I wish to focus on, since it is written in some type of programming language, whether the code is written directly through knowledge of a given programming language (a.k.a. “manual coding” or “coding by hand”) or through a WYSIWYG (What You See Is What You Get) authoring

[28] Kirschenbaum provides more detail on the history and essential characteristics of the hard drive in his article “Extreme Inscription: Towards a Grammatology of the Hard Drive”

[29] For a more detailed and visceral account of the challenges of destroying objects stored in digital media read Matthew Kirschenbaum’s “Every Contact Leaves a Trace: Computer Forensics and Electronic Textuality.”

[30] Or does it? In the case of an HTML file, experience tells us that different browsers and platforms interpret and execute certain instructions differently, resulting sometimes in radically different output versions. This point will be discussed further in a detailed exploration of Jim Andrews’ e-poem “Seattle Drift” in the next chapter.

program. Both are types of human-computer interfaces, which Lev Manovich defines as "the ways in which the user interacts with a computer. HCI includes physical input and output devices such as a monitor, keyboard, and mouse. It also consists of metaphors used to conceptualize the organization of computer data. [...] Finally, HCI also includes ways of manipulating data, that is a grammar of meaningful actions that the user can perform on it." (69).[31] How such a communication takes place is constantly evolving and mutating and there are many ways to achieve the same effects in different programs and programming languages, as is the practice of porting—transcoding from one programming language to another. Writing for electronic media employs some form of HCI in order to record instructions and data in a computer, so that the computer can then create new and/or utilize established interfaces for humans to access the data. We can think of such writing as a type of programming because, whether we are aware of it of not, we are addressing both a computer and user—the cyborg reader I keep referring to and which I will discuss in the next section.

The displayed document's materiality emerges from digital files stored in optical and magnetic media processed by the hardware and software so it can be transformed into signals that the computer's output devices can understand and turn in to visual, aural, and/or other material information, most commonly sound or some

[31] Manovich goes on to elaborate this term by adding a dimension of culture and the uses of digital media: "I will use the term *cultural interface* to describe a human-computer-culture interface—the ways in which computers present and allow us to interact with cultural data" (70). He defines this concept further by examining its component parts: "cinema, the printed word, the human-computer interface: each of these traditions has developed its own unique way of organizing information, presenting it to the user, correlating space and time, and structuring human experience in the process of accessing information" (72).

form of robotic motion (such as vibrating controllers in gaming consoles such as the Playstation, X-Box, and Nintendo). Since this output depends upon constantly refreshed signals to persist, it is open to changes according to timers, randomization, and new information received through whatever HCI devices are created or activated for the duration of the document display.[32] So the text we see on a screen is always more than just an image: it is an ongoing process. How it is processed depends greatly upon what kind of a digital object it is (JPEG, GIF, TEXT, PDF, and so on), how much storage space it takes, where it is stored and at what speeds it can be accessed, and what functionality or behaviors have been programmed into it (hotspots, hyperlinks, mouseover responses, and so on.)

The materiality of electronic documents is less stable than in print documents because they are constantly being regenerated over time and new variables entered during the life of the document can make significant changes. For instance, reducing or expanding the size of a browser window reconfigures the spatial arrangements in a Web page. It changes where the line breaks occur, the positioning of images relative to the text, and the dimensions of the displayed document—all of which can have an impact on how the text is read. The linguistic text may be constant, but the material text is generated "on-the-fly," as Loss Pequeño Glazier explains it in *Digital Poetics*:

[32] It is no wonder that early New Media theorists considered the document displayed on the screen to be immaterial, unstable, and evanescent. Transparency can also be achieved by a computer, after all, especially since the most prominent output devices are the screen, speakers, and printer and the most ubiquitous input devices are the keyboard and mouse. The processor, RAM, hard drives, and other devices are hidden away in the shell of the computer, partly from the practical needs of protecting delicate equipment, more importantly, to create a sense of immediacy—so that the moment you press a key on the keyboard, the corresponding symbol appears on the screen, or when you move the mouse the cursor responds in kind, creating the illusion that you are moving objects inside the screen.

> HTML can never sustain a "fixed" text. Indeed, HTML's most appealing quality is its lack of fixity. Because all browsers and all configurations of all browsers cannot be anticipated, HTML mark-up, in poignant contradistinction to typesetting for printing, proposes a provisional or conditional text. HTML markup details a general layout, a skeleton or arrangement of the *parts* of the document, over which the displayed version of the writing is draped like cloth or a very loose skin (15).

The "provisional or conditional" nature of electronic documents is what allows for the integration of different behaviors within a static e-text, because those conditions may change from one moment to the next.

The elaborate terminology we can use to describe the materiality of print texts as images, with all their graphical, bibliographical, and linguistic codes is available and useful to do the same with electronic texts displayed on a screen, but it would only be "screen deep," and would therefore need to be supplemented with a discussion of its functional/behavioral programmed characteristics. Conversely, the terminology used to examine the behavioral components of electronic texts wouldn't be very applicable to describe the majority of print texts, though some arcane exceptions can certainly be invoked for this purpose. Many theorists have employed Poststructuralist theories to describe electronic texts, a practice Matthew Kirshenbaum has argued against in the Electronic Book Review and elsewhere:

> a bibliographical/textual approach calls upon us to emphasize precisely those aspects of electronic textuality that have thus far been neglected in the critical writing about the medium: platform, interface, data standards, file formats,

> operating systems, versions and distributions of code, patches, ports, and so forth. *For that's the stuff electronic texts are made of* (“Materiality”).

Kirshenbaum’s own work on the ontology of first generation electronic objects has debunked misconceptions some early and contemporary New Media Writing theorists have perpetuated, such as the supposed evanescence of electronic textuality, as contrasted with the putative permanence of print. Texts in either media have a materiality that needs to be accounted for.

So to return to a consideration of the title of this section, we can establish a distinction between print texts and electronic texts along the following lines:

1. Both print and electronic texts are material and are stored in analog media.[33]
2. While the documents that print texts are stored in can be read in their storage state, electronic texts are unreadable by humans without a computer because the storage medium (magnetic or optical) records a digital representation of the source document.
3. In order for a human to read an electronic text, the source document must undergo a complex series of digital transformations and procedures in the computer from which an analog visual, aural or mechanical representation emerges in its output devices.[34] This emergent document is not the same

[33] For a technical explanation of the analog to digital and digital to analog conversions that go on in a hard drive, see Kirschenbaum, “Extreme Inscription” pp. 11-12.

[34] Taking McGann’s point that all texts are algorithmic in character and that they contain instructions for their reproduction, we can see a parallel in print between the source and display texts of electronic media.

as the source document because the source is one of many components that guide the creation of the readable document.

4. It is possible to read a source document as a representation of its code (instructions for the generation of a document) or as an executed program which generates that readable document).
5. The materiality of electronic texts is different and variable in its human-readable documents from its storage documents, while print documents are readable as storage documents and are therefore the same.[35]
6. A crucial distinction between texts stored in print and electronic documents is that print documents are material, while the materiality of electronic documents emerges from processes carried out by variable digital and material conditions, including the human reader.

If we consider what the computer does when it opens or runs an electronic document "reading"—something true at least at the level of interface metaphors—then the boundaries between the human reader and the computer she uses to read e-texts deserve to be examined and perhaps blurred enough to create a space for a metaphorical cyborg reader to exist.

[35] I make this statement well aware that while documents store texts, the relation between the two is not unproblematic. Multiple readings of the same document will produce multiple texts in the mind of the reader, as McGann argues in the conclusion of *The Textual Condition*, pp. 182-6. The fact that this is literally the case in electronic documents contributes to the conceptual joining of the readers of e-texts and their computers.

The Cyborg Reader

At what point did the reader become a cyborg? To what extent do the human and machine components come together when reading electronic texts? Notions of the cyborg and the posthuman have been explored by Donna Haraway and N. Katherine Hayles and become relevant to the conceptual blending of functions that a reader of electronic texts undergoes.

In her famous essay “A Cyborg Manifesto,” Donna Haraway develops a notion of the cyborg as a metaphorical creature that exists in the “leaky” boundary between humans and machines.

> The second leaky distinction is between animal-human (organism) and machine. Pre-cybernetic machines could be haunted; there was always the spectre of the ghost in the machine. This dualism structured the dialogue between materialism and idealism that was settled by a dialectical progeny, called spirit or history, according to taste. But basically machines were not self-moving, self-designing, autonomous. They could not achieve man's dream, only mock it. They were not man, an author to himself, but only a caricature of that masculinist reproductive dream. To think they were otherwise was paranoid. Now we are not so sure. Late twentieth-century machines have made thoroughly ambiguous the difference between natural and artificial, mind and body, self-developing and externally designed, and many other distinctions that used to apply to organisms and machines. Our machines are disturbingly lively, and we ourselves frighteningly inert. (152)

The point I wish to highlight and develop in this section is that of the permeable boundary between the human and machine readers of electronic texts—not as much in a science fictional sense popularized by cyberpunk fiction like William Gibson's *Neuromancer* and films like *Robocop* and *The Matrix* where the cyborg is a literal fusion of human and machine—but in a way that allows us to explore the interpenetrated production and reception performances further.

I have already established that the computer is an essential tool used to read electronic texts, but what is reading? It is important to distinguish between what humans and computers do in order to point out departures and convergences in what constitutes this activity.

> Reading is the process of retrieving and comprehending some form of stored information or ideas. These ideas are usually some sort of representation of language, as symbols to be examined by sight, or by touch (for example Braille). Other types of reading may not be language-based, such as music notation or pictograms. By analogy, in computer science, reading is acquiring of data from some sort of computer storage (Wikipedia Contributors, "Reading").

This definition has the virtue of being broad enough to circumvent most print and language biases that are present in other definitions, including a definition of reading in the context of computers. Note also that it distinguishes between retrieving and comprehending, and it includes both linguistic and extra-linguistic codes. These distinctions relate to Johanna Drucker's contrast between looking and reading. A person may look at a page of writing in a language foreign to him/her and not be able to comprehend what it says. Similarly, a computer may parse a sequence of magnetic polarities stored in its hard drive, but if it doesn't recognize its data structure, it cannot read the file. In both cases they would be "looking" but not "reading" which suggests a lack of comprehension of the linguistic information, despite our sometimes sophisticated understanding of visual information.[36]

We must not confuse comprehension with interpretation, though the two terms are not far from one another. Both humans and computers interpret linguistic texts, though in different ways. Computers interpret the codes they receive in programming languages that are designed to minimize ambiguity and provide clear instructions for them to follow. The language computers read is very different from the natural languages we humans speak and write, though some creoles do emerge.[37] As discussed in the previous section the computer programming languages are a type of HCI (human computer interface) which allows people to write instructions or enter data for a computer to read, interpret (also know as "parsing") and attempt to carry out those instructions as it has been programmed to do so. The multiple layers of interpretation the electronic text undergoes before display to a human being aren't as distinct from the viewer as many would like to believe, nor are they purely algorithms under the control of the writer.

Given that people and computers read in similar and different storage media, what happens when we use a computer to read and electronic text? We read a reading

[36] If Drucker's word choices seem somewhat human and print biased, it is because she is dealing precisely with visual aspects of the written word.

[37] Mez (Mary Anne Breeze) is perhaps the best known writer of what she calls "mezangelle"—a hybrid of English and computer programming codes and rebus writing.

of the codes that describe the text and instruct the computer on how to display it, but we are also giving instructions. A reader of e-texts is already a part of the computer's interpretive structures just as the computer's software and hardware are already encoded into their users mental processes. The computer's software and hardware are shaped by customization (screen size and resolution, color spectrum, gamma correction, sound volume, hardware and software updates and optimization, and so on)and more immediate conditions the moment an electronic text is accessed (window size, number of files and/or windows open in the same program, other software running at the same time, how busy a network is or how fragmented is the storage of a file in a hard drive, to mention a few factors). Readers are trained by the computers and software they use, to the extent that routine operations are often carried out without conscious thought—the "incorporating practices" Hayles discusses in *How We Became Posthuman* (199).[38] The degree to which training and customizing happen will be higher when a reader uses his/her personal computer and lower when the reader uses a computer he/she is less familiar with—but as far as an electronic text is concerned, it is encountering a hybrid reader composed of integrated human, machine and software components bound together.

From this perspective, readers of electronic texts are no longer separable from the computers they use to read. Philippe Bootz—a pioneer French e-poet and theorist—has developed a procedural model that accounts for the complexities of communication by taking a systemic approach that explores the "technical dispositif. .

[38] A central idea in this book is that new technologies create new incorporating practices (such as learning how to type, or use a mouse, trackball, or stylus) that become part of how humans experience the world.

. namely, the computers of the author and the reader, and the actors themselves" (97). This framework distinguishes between the "text-as-seen" (the display generated from the written text, the reader's computer and the reader's interactions) and the "text-as-read" (what is actually perceived by a human reader). An important part of the perspective he proposes is that text is a process engaged in feedback loops that bind the writer, the text, and the reader in ways that give power to all, but complete control to none. The author creates a text that may or may not run the way it is designed to do. The code that constitutes the electronic text may be interpreted in different ways by the writer or reader's computers. The reader's interactions vary in infinitely subtle ways from reading to rereading, in any medium they choose to read in, but the computers they use are part of these variations.

As the boundaries between readers, computers, writers and texts continue to erode, new conceptual creatures emerge in these liminal spaces. Some theorists will consider them monsters, aberrations, while others will join their dance.

Reading Textual Behaviors

Reading is a skill; something we learn as children and become increasingly proficient at through learning and practice. Since most of the reading we do is through print media—books, newspapers, magazines, journals, and so on—its conventions and technologies have become deeply ingrained in our reading practices, and we are rarely aware of them. For instance, we don't have to think consciously of turning pages, determining which word to read next, or that we are beginning a new sentence. Some print works challenge these conventions, forcing us to reexamine our reading practices, but they constitute a small portion of the works we read in print.

We are also used to reading screens, be they movie theater, television, video game, or computer screens. When watching subtitled films, for instance, we are basically reading text that operates on a strict schedule: it needs to follow the pace of the dialogue. In television channels such as CNN, we have screens loaded with text that is constantly changing, whether appearing or disappearing on a schedule or scrolling in the bottom of the screen. Video games tend to work more with images than language, but these images are charged with information, which must be identified and interpreted (or read) by the player. Successful players are necessarily good readers of not just visual and aural information, but also of the program's responses to their actions. All of these screen technologies inform how we read the computer screen.

Most documents that we read in a computer follow print conventions, and add a few of their own, such as the incorporation of links, and using hypertext for organization. These are rarely problematic, but when we encounter e-poetry or other "first generation electronic objects," our traditional reading skills are insufficient. [39] Our training in reading print does not account for words that move and form new textual combinations, nor does it teach us to explore the textual surface with the mouse to reveal hidden elements, for instance. Readers are often disconcerted by a text that imposes a reading schedule, or texts that are impossible to reread because they change every time they are accessed. So how do we read the dancing signifier?

[39] Matthew Kirschenbaum defines them as "a class of artifacts that have no material existence outside of computational file systems, which would include electronic fiction and poetry, and other types of hypertext and cybertext works" ("Materiality")

I have already suggested that when language is inscribed in programmable media such as a computer it can be described in terms of linguistics, appearance, and behavior. Since our reading skills have prepared us to see through the appearance of texts to reach a linguistic meaning, but not to deal with texts that exhibit behavior, I propose a typology that describes six behavioral characteristics: static, kinetic, responsive, mutable, scheduled, and/or aural. The ability to identify and account for the signifying strategies of these behaviors allows for more sophisticated readings of e-poetry and by consequence e-texts in general.

But before discussing the typology itself, I must pause to explain my decision to describe this textual characteristic as “behavior,” unpacking some of the connotations and denotations that load the term beyond the scope I am using. Let’s take as a point of departure a dictionary definition of the term:

> **1 a:** the manner of conducting *oneself* **b:** anything that an *organism* does involving action and response to stimulation **c:** the response of *an individual, group, or species* to its environment.
>
> **2:** the way in which *someone* behaves; **also:** an instance of such behavior.
>
> **3:** the way in which *something* functions or operates. [italics added] (“Behavior”).

By looking at the italicized words, it becomes apparent that the term is closely associated with the actions of living organisms, and only in its third definition does it describe inanimate objects. This is not accidental: the term has not been used to describe such objects for long, not since 1943, when Norbert Weiner, Julian Bigelow and Arturo Rosenblueth published an essay titled “Behavior, Purpose, and Teleology” where they define it as:

> By behavior is meant any change of an entity with respect to its surroundings. This change may be largely an output from the object, the input being then minimal, remote or irrelevant; or else the change may be immediately traceable to a certain input. Accordingly, any modification of an object, detectable externally, may be denoted as behavior (18).

Of interest in their definition is how broad its scope is, including living organisms or inanimate objects, and how it focuses on externally detectable changes. N. Katherine Hayles points out that this approach is “relatively unconcerned with internal structure” and that it leads to “ ‘black box’ engineering, in which one assumes that the organism is a ‘black box’ whose contents are unknown. Producing equivalent behavior, then, counts as producing an equivalent system” (*Posthuman* 94). She argues that it is not a neutral term and that the attempts to apply it to machines have been ideologically motivated to “elide the very real differences existing between the internal structure of organisms and that of machines” (94). As part of her discussion, Hayles asserts that Richard Taylor, a philosopher who challenged Weiner’s definition of behavior and purpose, “sensed that *behavior* had been defined so as to allow intention and desire to be imputed to machines” (97). These are all considerable problems with the term behavior, and should be addressed in order to justify its usefulness as a methodology.

First of all, do machines have intentions and desires? In a conversation with Philippe Bootz during the E-Poetry 2001 conference, he said that animation is the symbolic presence of the author in the text, and the cursor is the symbolic presence of the reader. I see these presences as the intentions and desires that fuel the behavior of an e-text—which are encoded into the source document that is executed by a computer. Yet we are not dealing merely with the preserved intentions and desires of the author and nor reader's during the process of reading the e-text. Mediating between the two there are myriad lines of code, processors, accelerators, memory, hard drives, and components—every single one of which carries code designed, written, revised, patched, ported, and so on by individuals, teams, companies, and many different types of groups, each of which has its own individual or collective intentions and desires. One could argue that the unifying desire and intention shared by all these people is for things to work and run efficiently, but is it? New jobs and companies have recently emerged to counter the questionable intentions of hackers, virus programmers, spammers, and other programmers whose electronic objects are designed to disrupt the normal functioning of a computer system. The computer orchestrates all these instructions in its processes prioritizing some, overriding others, and what emerges is its behavior, which responds to a whole complex matrix of intentions and desires, the writer's, reader's and everyone else's who contributed to the workings of a computer. So do computers have intentions and desires: not really, but they do respond to those encoded within them, and their external behaviors may be interpreted as having them.

Considering that there are different ways to achieve the same results through different programming (as evidenced by the practice of porting[40]) and that different computer hardware and software configurations interpret the same instructions differently, a purely functional approach (one that focuses on the inner workings of a computer and electronic objects) would be too cumbersome to perform literary analyses of e-texts because the variables are potentially limitless. Yet a purely behaviorist approach that focuses only on externally observable behaviors can lead to treating the computer and the source code for an e-text as a “black box” and fall into the very screen essentialism that Nick Montfort and Matt Kirschenbaum criticize. I believe a balance can be struck between the two in an approach that acknowledges differences between different types of electronic objects and still make valuable readings on how they behave in an electronic environment when executed.

This approach consists of multiple readings of an e-text using different computer configurations (i.e. using different browsers to open a Web-based document or computers from different platforms) combined with looking at the source code, when available, in order to observe for potential patterns and variations in the display documents. If there are significant discrepancies, reading the source code may help determine which variant is closest to the author’s intent, as programmed. Reading the source code can also be useful to see what are the actual mechanisms that generate the behaviors, instead of guessing the algorithms based purely on observation of said behaviors. One cannot always count on having access to all of the abovementioned

[40] “In computer science, porting is the process of adapting software so that an executable program can be created for a different computing environment (e.g., different CPU, operating system, or third party library) to the ones it currently runs on” (Wikipedia Contributors “Porting”).

resources: sometimes the source code isn’t available; at times the electronic object is obsolete in its programming; perhaps there are incompatibilities that don’t allow the e-text to execute its encoded instructions correctly. In any case, one works with what one has available, exercising careful judgment before privileging any single component or variant of an e-text. I believe that this approach can yield fruitful readings of electronic texts, as I will demonstrate in the next chapter.

The typology of behavior I will now discuss provide a critical vocabulary to describe characteristics of electronic texts. It is a brief list of characteristics which I have observed in e-poetry, along with some basic subcategories, that should describe a full range of behaviors programmable into electronic texts. One could think of this as a type of folksonomy, which can be used to tag different textual behaviors within an electronic text.

1. Static texts are the default we’re used to in print—they are texts that do not move or change on the screen.

2. Scheduled texts may reveal themselves over time, which may be linear or looped; they may force a rate of reading by disappearing or scrolling; they may also trigger events over a programmed or random schedule.

3. Kinetic texts move on the screen: this motion may be looped or linear, random, programmed, or responding to cues from the reader.

4. Responsive texts take advantage of the computers’ interface devices (most commonly the mouse and keyboard) to create a feedback loop between the reader and the text. The input cues (such as links, hotspots, and keyboard enabling) may be manifest or hidden, allow for voluntary or involuntary

interaction, and have immediate or postponed reactions to the reader's input.

5. Mutable texts involve programmed or random changes and may also change due to reader interaction.
6. Aural texts have a sound component, whether verbal, musical, or simply noise.

These categories are not by themselves unique to electronic media, nor are they mutually exclusive. They are often found in combination, and in some cases they are inseparable—aural and kinetic texts are always scheduled, for instance. The next few sections will provide a brief genealogy of each behavior and discuss some of their subcategories, and implications as outlined above, providing examples from a variety of e-poems.

Static Texts

When we think about words in a document, we assume that they remain still so we can read them. We also assume that they will remain the same, so we can reread them if necessary. Motion and mutability are not characteristics we ordinarily attribute to words because the materials on which they have been inscribed don't usually allow for such changes. Static texts are so ubiquitous that traditional definitions of text are based upon this behavior, or lack thereof. In *Scholarly Editing in the Computer Age*, Shillingsburg defines text as "the actual order of words and punctuation as contained in any one physical form, such as manuscript, proof, or book" (46). This notion of text arises from a centuries old relationship between alphabetic technologies and the media in which they have been recorded. In simpler terms, documents have been produced in stone, clay, papyrus, vellum, wood, paper, and other materials that lend physical stability to the inscriptions they carry. The words etched, inked, penciled, or glued onto those surfaces are not likely to move from where they are placed, and any motion of these depend on the manipulation of the materials they are placed upon. They are also not likely to change, even though they may be interpreted differently by readers. Therefore, in any given document, the text is defined by the stability of the materials it is inscribed in.

The moment words start to appear in screens (film, television, computer, or any other electronic device) the static default becomes simply another option available for their display, because we are dealing with time-based media that can display moving images.[41] So even if the text displayed is perceived as static it is being constantly redrawn many times per second (at a rate of 89 hertz in my 2004 E-Mac, for instance). It is for this reason that Hayles calls electronic texts "flickering signifiers."

> When a text presents itself as a constantly refreshed image rather than as a durable inscription, transformations can occur that would be unthinkable if matter or energy, rather than informational patterns, formed the primary basis for the systemic exchanges (*Posthuman* 30).

[41] The rates vary, depending upon the medium. Film requires 24 frames per second for human viewers to have what is known as persistence of vision—the illusion of smooth, continuous motion of the images projected on a screen. Video and television work with interlaced images, which are refreshed at a standardized rate of 60 frames per second. Computer screens operate at different rates, currently from 60 hertz upwards.

Some of the transformations occur at the level of textual behavior, but they can also be changes in the appearance of the text, or even linguistic information of the text. More importantly, these transformations are possible because texts in digital media are informational patterns which are subject to manipulation and reconfiguration in computers. This is obvious to anyone who can use a word processing program to modify a word's font, size, color, emphasis, spacing, indentation, and many other of its visual characteristics. What isn't obvious to many users is that we are changing the word's informational pattern in ways that the computer can recognize and reconstruct and that this pattern is particular, not universal. A different piece of software or computer may not recognize the pattern in the same way or may not have the font available, and it will interpret the information as it is able, reshaping the information pattern to conform with the requirements of the system. Therefore, static texts in digital media are not stationary objects in repose: they are informational patterns processed and constantly inscribed on a computer screen. That is what readers interact with, carrying out their own complex processes.

To reiterate a key point: there are no stable or inert texts, regardless of the medium they are inscribed in. The act of reading subverts any illusions of textual stability because the text that is reconstructed in a reader's brain is contingent on how a reader scans the words on a document—the reception performance Shillingsburg discusses. It is not uncommon to skip words, confuse them for other words, invert their order, or reread them, all of which affect the mental reconstruction of a text from a physically stable document. Reading a document a hundred times will produce a hundred similar texts in the reader's mind—or at least a hundred visions and revisions of the same text.

Purely static texts in electronic media can be similar to texts in print, especially when there is remediation at work (as is the representation of print in digital media), at least as far as the act of reading is concerned. However, since the computer is such a powerful simulation tool and can be used to create new writing environments for texts to inhabit such as hypertext, three-dimensional spaces in VRML, and multimedia so the texts may require elaborate navigation to read them. For example, David Knoebel's *Words in Space* poems use VRML to place words at different "distances" and angles so the reader must zoom in and out of the text, and maneuver amongst the words.[42] The words in Jim Andrews' *Seattle Drift*, if allowed to drift for a long period of time, would create an enormous virtual space in the browser that would require serious exploration of that space using scrollbars to find them.(Andrews)

When we describe a text as static, it should be understood that it may not also be considered kinetic or mutable, even though they may change their behavior, if scheduled or responsive. Stasis is the most common behavior for text in electronic media, but that doesn't make it any less of a behavior than the others. The specter of possibility haunts texts in digital media, because we cannot trust that what we see is what we get.

[42] http://home.ptd.net/~clkpoet/cpwis.html

Scheduled Texts

Because computers are time-based in their operation, the texts they process have the capacity to be scheduled. Texts with this behavior are either finite or open-ended. If finite, they can be described in terms of duration—whether they are linear or looped. The events triggered in the scheduled text can be described as singular or recurrent. If recurrent, we can describe them in terms of frequency of their recurrence. A key concept here is the event—an action that changes the state of the electronic object, triggered by a preprogrammed schedule or user input. An example of a schedule-driven event is when a computer goes into sleep mode after a predetermined period of inactivity.

These subcategories become more complex in combination with other behaviors, such as mutability and responsiveness, because the scheduling can be random, variable, or affected by the user. Scheduling texts can have several implications as illustrated by three examples:

1. A basic feature of print texts is that the reader controls the reading rate. Scheduled texts take control of that reading rate over part or all of the work. A good example of this behavior is evident in the works produced by Young-Hae Chang's Heavy industries, which display one phrase, word, or letter at a time on a single frame over a schedule inspired by and lasting as long as the musical piece chosen for it—resulting in a text that plays like a film and demands the reader's unflinching attention for the duration of the work.[43] The

[43] http://www.yhchang.com/

text unfolds in a linear fashion, and cannot be stopped or reversed once activated.

2. When scheduled e-texts are looped they provide the opportunity of re-reading the sequence that has occurred. Brief loops do not give the impression of scheduled operation, because they present multiple opportunities for re-reading, as is the case of the brief animation in Neil Hennessy's *Paddle*[44] and *Puddle*[45]. When compared to the lengthier Young-Hae Chang works, where if you miss a word you can't pause or go back: you must press on or restart the work to get another opportunity to read it again.
3. Open-ended schedules have events that do not recur or at least not predictably so. The drifting of the text in Jim Andrews' *Seattle Drift* goes from its original configuration until the screen is left blank. Readers have the option to stop the text from drifting, and even "discipline" it back into its original configuration, but they cannot reverse the drift, or even repeat it. The drifting is randomized, so any re-readings are necessarily of differently configured texts, except for the beginning.

There is little or no scholarship done on scheduling of texts, that I know of, and it is an area that merits further exploration. Some related fields that may provide fruitful information are studies on reception of oral language and recorded texts, such as audiobooks, and studies of subtitles, captioning, and other uses of language in time-based visual media, such as film and video.

[44] http://epc.buffalo.edu/authors/hennessey/data/puddle/index.html

[45] http://wings.buffalo.edu/epc/ezines/deluxe/two/puddle.html

Kinetic Texts

> The moving image moves. But where does that movement come from? For a certain approach in art history, an image is a discrete, whole entity. To move from one image to another is already an immense wrench: even the analysis of a diptych is wildly complex. What then is it to speak of “a” moving image, constructed from thousands of constituent images? In what sense is it *an* image? Cinematic movement is a fundamental challenge to the concept of wholeness and integrity, its becoming a test of the primacy of existence. In particular, it raises the question of temporality: when is the object of cinema? When, indeed, is the moving image? (5)

This excerpt from the introduction to Sean Cubitt’s book *The Cinema Effect* asks a relevant and provocative question about the ontology of the moving image which I will adapt to the discussion of kinetic texts. To what extent can a word in motion be considered a single signifier? More importantly, how does the shifting position of a word in motion reconfigure its relation to other linguistic, graphical and behavioral elements in ways that affect its meaning?

The singularity of the rendered electronic image is a perceptual event, whether it is still or in motion, because it is drawn and redrawn many times per second in order for humans to achieve persistence of vision. As computer graphics, however, these electronic objects can be multiple or singular, depending upon whether they are vector or raster graphics.[46] Any change in a raster graphic modifies its composition as a numerical object, whereas the formulas that create the vector graphic can have movement programmed into them, as is the case with Flash animation. For the sake of convenience, I will take the computer science approach of "object-oriented programming" to treat all kinetic texts as singular objects because it is more flexible towards incorporating other behaviors, even if they are composed of multiple frames.

The primary theoretical approaches towards computer animation comes from cinema—and appropriately so. In *The Language of New Media*, Lev Manovich uses "the theory and history of cinema as the key conceptual lens through which I look at new media" (9). His exploration goes in both directions, however, seeing also how digital media and its capabilities transform cinema. This study does a very thorough job of exploring how the history of cinema informs and helps us understand some of the ways new media works, but its focus is more on characteristics of new media, imagery and visual narrative rather than on written language and its signifying potential when placed in motion. John Cayley sets out to rectify this need in "Bass Resonance," an essay that explores the cinematic history of words in motion, focusing on the work of Saul Bass—a man famous in film history

[46] "Vector graphics stores precise geometric data, topology and style such as: coordinate positions of points, the connections between points (to form lines or paths), and the color, thickness, and possible fill of the shapes. Most vector graphic systems can also use primitives of standard shapes such as circles, rectangles, etc. In most cases, a vector graphic image has to be converted to a raster image to be viewed. Raster graphics is a uniform 2-dimensional grid of pixels. Each pixel has a specific value such as, for instance, brightness, color, transparency, or a combination of such values. A raster image has a finite resolution of a specific number of rows and columns. Standard computer displays shows a raster image of resolutions such as 1280(columns)x1024(rows) of pixels. Today, one often combines raster and vector graphics in compound file formats (pdf, swf)." (Wikipedia contributors, "Computer Graphics")

for his animated title sequences at the beginning of films like *Anatomy of a Murder* (1959), *North by Northwest* (1959) and *Goodfellas* (1990).[47] This brief essay describes some of the effects of Bass' dancing words, aligning his practice and much of the practices of e-poetries with Concrete poetics. Both studies place kinetic texts and images in digital media in historical, cultural, and cinematic contexts, yet their interest isn't with the complexities of textuality in motion and their implications for poetic practice.

An essay that takes an important step in that direction is "The Software Word: Digital Poetry as New Media-Based Language Art" by Janez Strehovec. This essay focuses on the aesthetics and cultural space that digital poetry is establishing for itself—one that moves away from the "lyrical and 'projective saying'" (143) and even beyond remediation of print poetic traditions (145). More importantly, he asserts that

> words inside textscapes are words-images-virtual bodies, they are self contained signifiers which must be perceived not only considering their semantic function but also their visual appearance as well as their position and their motion in space (149).

Strehovec is on the money when discussing digital poetry and its aesthetic function, yet his discussion of kinetic texts is a good, but insufficient beginning. Strehovec, like Manovich, argues that kinetic texts basically operate on the concept of the loop. This is a weakness in their argument because they are privileging one of several control flow statement types, roughly categorized as follows:

- continuation at a different statement (jump),

[47] http://www.electronicbookreview.com/thread/electropoetics/dynamic

- executing a set of statements only if some condition is met (choice),
- executing a set of statements repeatedly (loop),
- executing a set of distant statements, after which the flow of control returns (subroutine),
- stopping the program, preventing any further execution (halt) (Wikipedia Contributors, “Control Flow”).

These control flow statement types are what make all the textual behaviors possible and make animation in digital media so unique, because it is able to incorporate other elements discussed in this typology, such as responsiveness, mutability, and scheduled operation. Let us explore further some of the potential and implication for kinetic texts.

Time in an animation may be linear, looped, or open-ended.[48]

1. Linear kinetic texts have a clear beginning and may or may not have an end. For the reader to re-experience the animation, they may have to reload the text and experience it again from the beginning. Brian Kim Stephans’ *The Dreamlife of Letters*, for example, is a long kinetic e-poem that unfolds without allowing readers to pause, “rewind,” or skip through the text.[49] It is only at the end that the reader is given options to return to brief excerpts of the text in order to re-read sections of the poem. Another example is Jim Andrews’ *Seattle Drift*, which gives the reader the option to

[48] It could be said that all animated texts are scheduled, to a certain degree, though it is not considered scheduled unless the passage of time triggers new events. This will be discussed further in the “scheduled” section of the typology.

[49] http://www.arras.net/RNG/flash/dreamlife/dreamlife_index.html

end and restart the animation, even return to the initial configuration of the text, but the animation is both linear and potentially open-ended. One might argue that the animation ends when the last word drifts out of the screen, but just because the reader can’t see them doesn’t mean that the words have stopped drifting. There is the potential for them to drift back into the screen, given enough time, or the reader may “chase” after them using the scroll bar in their browser window.

2. Looped animation allows the reader to re-read the kinetic text when it cycles through. Sometimes the loop can blur the sense of a beginning and end for it, and its beginning may be randomly determined. Loss Pequeño Glazier’s *Colibrí* contains many short, quick, looped animated texts around the drawings of hummingbirds to mimic the birds’ flight.
3. Open-ended animation seemingly has no end, and at times it seems like there is no sense of progression at all. Continuous kinetic texts become clearer with examples such as George Hartley’s *Fly*, where the word “fly” moves across the screen from its center to the edges in a continuous stream through random trajectories, growing in size as it approach the edge of the screen.

1. Word motion can affect the reading of the text in several ways.
   a. It blurs the line between reading and looking, especially when the motion serves as an obstacle to the reading of the text. The reader

may see texts in motion, but not have the time to recognize them, in which case, the words are perceived more as objects than as signifiers. This foregrounds the graphical aspect of the text, reducing the impact of the semantic codes in the document. For instance, in Tammy McGovern's *Meaning Effect*, moving the pointer over hotspots triggers words looping so rapidly that the reader is barely able to read them. The effect of the animation speed in this poem is to obscure meaning, of what might be otherwise easily readable text.

b. It may reconfigure the word order, producing different phrases and meanings. This is evident in a work like *Seattle Drift*, where the words drift to form various different textual combinations. Since the reader has the option of stopping the text, the new syntax of the drifted poem is available without shifting further on the reader. Some of the words may overlap as well, becoming more difficult to read in the process.

c. Works like *The Dreamlife of Letters* create a grammar of motion by grouping words with the same or similar movement. For instance, in the section from "dread to drip" the words "read" and "ream" are alternated in an 11-word semicircle which moves by the stationary letter "d" to form the words "dread" and "dream," after which the solitary letter "d" drops from the center of the screen to fall by the suddenly appearing word "rip," forming the

> word "drip" for a brief moment. All these words in this section of the poem share the same letter "d." The visual organization of the alternated words ream and read cascading in their curve from top to bottom of the screen to form dream and dread highlight the relationship between both pairs of words: what is the relation between dreading to dream and reading a ream of paper, or perhaps reading being like reaming juice out of a fruit? What is the relation between the liquid action of dripping and the very solid action of ripping? The juxtaposition of these words for the reader's consideration is brought about through patterned motion.

There is much more to explore on the implications and effects of motion in texts. My concern in this chapter is to establish clear examples how they are manifested in e-poetries and their effects on the texts. The next chapter will explore textual behavior further through readings of Jim Andrews' e-poems.

Responsive Texts

I have chosen to describe these texts as "responsive" rather than "interactive" because the latter term has generated some controversy in its previous uses. This arises from the fact that all texts are interactive, because to read is to interact with the graphical and semantic codes contained within a document to generate meaning. There is also interaction with the physical document in which the text resides, such as page turning and other physical manipulations, in the case of printed texts. The responsive texts I refer to, however, take advantage of the computers' interface devices to allow for input from the reader (usually the mouse and keyboard).

Navigation is an important issue, and one explored by many writers of e-texts whenever they make decisions about the degree of control and manipulation of their works they want their readers to have. At one level they must establish the organization of the work and what tools the reader will have to traverse it. In the case hypertext works such as Michael Joyce's *Afternoon* or Shelley Jackson's *Patchwork Girl* this navigation becomes a self-conscious exploration and discovery of, not only the text, but of the act of reading the text, including decisions on when to stop. Since the entirety of the text is not physically present for the reader, there is not always a clear sense of progression, and the tools for reading such a work become integral to the text. Fabio Doctorovich's *Abyssmo*, for instance, requires the readers to use the tools offered by: (1) the browser (such as the scrollbar and back button), (2) the links clearly marked in each document, and (3) to discover hidden cues within some parts of his poem. This hypertext e-poem urges the reader to explore its lexia using overt and hidden navigation cues within the text, but there is one lexia which has no way out, unless it is through the back button of the browser, highlighting the power the reader has over the work.

This issue finds its parallel in many print works, particularly those that take advantage of the technology of the book and its bibliographical conventions. For instance, page numbers, footnotes, endnotes, table of contents, indexes, pagination, cross-referencing, and other devices allow the reader to traverse a book in an organized or even haphazard fashion, if so desired. Writers also make numerous decisions about what navigational devices to include in their books and to what degree these devices become a part of their artistic creation. Mark Danielewsky's

House of Leaves, for example, uses footnotes extensively, to create a labyrinth for the reader, mirroring the labyrinth present in his novel. The reader always has the choice to ignore such devices, however, and can skip and jump pages at will because the work is physically present and lends itself to that kind of manipulation. Navigation is only one aspect of interaction, and every technology and medium has similar issues that manifest in different ways.

> The distinctive factor for responsive electronic texts is the presence of a feedback loop that takes into account the reader's input and responds according to its programmed instructions. By "input" here, I do not refer to the mental interaction that is always supplied by readers, as described by Wolfgang Iser's reader's response theories, but to options programmed into the text by the author for the reader to trigger. These input cues (such as links, hotspots, cursor movement, keyboard entries, or others) may be manifest or hidden, allow for voluntary or involuntary interaction, and have immediate or delayed reactions.
>
> 1. Manifest input cues find their clearest example in the traditional underlined link that is such a staple of hypertext. In general, manifest cues are invitations for input, be it as simple as a clicking on a link or entering text into a box.
> 2. Hidden input cues are also an invitation to interaction, but of the exploratory kind. They challenge the reader to discover aspects of the text not apparent to the naked eye, by using the tools at their disposal, most

commonly the mouse. The mouseover function, for instance, reveals hotspots and may trigger responses from the text.

3. Voluntary triggering of responsiveness is the most common, and perhaps the friendliest towards the reader. The reader chooses to activate hotspots or links.
4. Involuntary triggers, however, present interesting possibilities. For instance, to have links or hotspots activated by a mouseover, not a click of the mouse, and to have these cues hidden can create the effect of a trapped environment, in which any movement of the cursor can trigger effects beyond his/her control. This can be seen in Giselle Beiguelman's *Recycled,* where the letters of the word "recycled" chase the pointer, no matter where it goes in the screen, and disappear when they touch it, only to reappear from the edge of the screen and continue the chase.
5. Most of the reactions of responsive texts are immediate, creating a fairly direct correlation between action and reaction. There is a sense of discovery whenever a reader activates an input cue, particularly the first time a responsive e-text is read. When re-reading, the reactions are expected and any variation may be disconcerting—a point to be discussed in greater detail in the section on mutable e-texts.
6. Delayed responses from activated input cues blur the correlation between action and reaction. This is one of the most important devices for Philippe Bootz's e-poem "Passage" because it reinforces the "unique-reading" experience of the poem. This poem in three movements allows for

interactivity during the second one, but its input cues are hidden and its reactions delayed. The information gathered by the program during this movement is then used to generate the third, which is necessarily different every time it is read, partly due to the programming, partly because of the variations in interaction.[50]

7. Sometimes the response is not predictable because there are variable responses to the triggering. David Knoebel uses this in an animated VRML poem titled *Walkdon't*. In this e-poem, the text revolves around several different axes and the only input cue is a dot underneath the revolving text. A mouseover will trigger a change in the rotations, pause some rotating words and starting others, but the reader doesn't have precise control over what happens.

All texts are responsive and interactive, irrespective of the mediums they "inhabit," because they are a machine for ideation and signification. The act of reading is by definition a dynamic interaction with the document that holds the text: and different writers will place different demands upon the reader and offer different cues for such interactivity. Electronic texts externalize aspects of this interactivity by presenting the reader with evident cues, and the reader's interaction is present and noticeable. Phillipe Bootz's observation that the pointer is the symbolic presence of the reader in the text is very appropriate to understand this. This becomes evident in works like Jim Andrews' *Arteroids,* in which certain texts chase the player's "id-

[50] For more details on this unique e-poem, read Philippe Bootz, "The Functional Point of View: New Artistic Forms for Programmed Literary Works."

entity" all around the screen. The reader's symbolic presence is an event read by the e-poems themselves, which as electronic objects have built in variables that are informed by those events.

The extent to which a text can be changed by interaction can be best described in the next element of the typology: mutability.

Mutable Texts

As discussed, e-texts are particularly susceptible to changes brought about by different software and hardware configurations. Mutable texts, however, incorporate deliberate variation into their design, making rereading the same text difficult, if not impossible. Mutable texts involve programmed, random, or user-defined changes in the document.

Mutability is not a distinctive feature of electronic texts. Works like *Cent Mille Milliards de Poemes* by Raymond Queneau use the book as a machine (and the reader as engine) to create 100,000,000,000,000 possible sonnets. This is a sonnet in which each page is cut under each of its 14 lines, so the reader can open each line on any of 10 pages, thus creating 10^{14} possible combinations. And yet, the work as a book is present to the reader, who can make choices based on page numbers and lines. Nothing is hidden, and while the potential line combinations are enormous, the fact remains that the individual lines will not change from what they are.

A parallel work to Queneau's is by Loss Pequeño Glazier's e-poem *White-Faced Bromeliads on 20 Hectares*, which is best described by its "Reading Notes:"

> Instructions: Allow this page to cycle for a while so you can take in some of the images and variant titles. When you are ready, press "begin". Once there,

> read each page slowly, even aloud, watching as each line periodically reconstitutes itself re-generating randomly selected lines with that line's variant. Eight-line poems have 256 possible versions; nine-line poems have 512 possible versions.

This e-poem' exhibits some significant differences from Queneau's, particularly regarding issues of user access and control. While it too has a finite number of variants, their access is not user-defined, and the variables are hidden from the reader. It also operates on a schedule, changing the displayed text every 10 seconds. Thus, the reader doesn't have: 1) control over the changes, 2) the ability to reread the same text, unless it is through printing out a given version, or capturing the image of one of the displayed documents, 3) access to the variants. The mutability is very much a part of this text: it shifts during the reading, encouraging the reader to reread read backwards, start over and over, attempting to make sense of this textual moving target.[51]

The difference between these two works goes deeper than their relation to the user/reader: they represent the paradigm shift from floating signifiers to flickering signifiers. According to N. Katherine Hayles in "Virtual Bodies and Flickering Signifiers," the floating signifier embodies the dialectic between presence and absence, while the flickering signifier shifts to a dialectic based on pattern and randomness. Each page/line of Queneau's book/poem represents a choice for the

[51] A reader who can access the code can see that each line has two variants and each line is randomly determined every 10 seconds. It is then possible to read each variant and even reprogram the poem to provide more user control, for careful studying of each variant. This is one way in which knowledge of code can allow for an alternative access to the text. This issue will be discussed further in the concluding chapter, under the section titled "The Hacker Critic."

reader: what lines become present and which lines are absent. Glazier's e-poem has built in randomness, yet it is structured enough that a pattern emerges from the flickering lines of his poem.

Espen Aarseth coined two neologisms in *Cybertext* which become useful for the discussion of mutable texts: scriptons and textons. Scriptons are "strings [of signs] as they appear to readers," and textons are "strings as they exist in the text" (62). Aarseth describes Queneau's *Cent mille milliards de poemes* as containing 140 textons that can combine to produce 100,000,000,000,000 possible scriptons (62).[52] A similar calculation could be applied to Glazier's *White-Faced Bromeliads*, with a significant difference: that the possibilities are part of the text, but the reader is presented with only a fraction of these. In works such as *Passage*, by Philippe Bootz, the program guarantees that you will never see exactly the same scripton, no matter how many times you reread the poem.

So the two main types of mutable texts are:

1. Programmed mutable e-texts have changes that result from authorial planning, whether it is to include random elements into the generation of scriptons, or whether these occur in a schedule, or through randomized animation.
2. User-defined mutability results from the intersection of responsiveness and the programmed nature of the e-text. The difference between merely responsive e-texts and mutable responsive e-texts is that the changes in the text are at least partly dependent upon the reader/user's input. In a mutable e-

[52] He goes on to develop a typology of "modes of traversal" of cybertexts: a useful one to show the similarities between print and electronic works that require the reader to spend "non-trivial effort" in their traversal.

> poem such as “Passage” by Philippe Bootz, the reader’s input during the second movement is essential to the changes that manifest themselves in the third movement of the poem. Different users, and repeated reading performances of the entire work by the same user, will necessarily produce different interactions, which will result in a newly configured third movement of the poem.

Mutability is necessarily a general category, but a significant one because it literalizes the textual instability present in all texts, whether in print or in electronic media. The changes take place as part of the production history of the material text that may or may not include interventions by the reader. Some change—however minuscule—is possible in any electronic text, as was discussed in the introductory chapter, but in mutable e-texts this happens to an even greater degree, and as part of the design of the poem.

Aural Texts

Poets have used writing as a recording medium for centuries by translating the sounds of poetry into alphabetic scores for oral reconstruction—just as composers have written musical scores on sheets of paper for subsequent musical reinterpretation. In poetry, sounds and units of breath become space: lines, stanzas, punctuation, spaces between words, formatting, and other visual markers become part of what readers learn to interpret in order to come up with to provide an oral rendition. The use of writing, however, led many poets to explore the expressive potential of writing in and of itself . . . culminating in the visual Concrete Poetry movement.

The rise of sound recording technologies allowed for poets to explore the aural element of language beyond the limitations of the writing and oral reconstruction model. These technologies are fairly new and have therefore accrued a smaller body of work—and market—than print. For the most part, sound recording technologies have been used to record poets reading their work: serving as an archive of authorial interpretations of the written poems. However, the Concrete Poetry movement also explored sound as a means in and of itself using the sounds of language beyond the traditional constraints, such as using words. Poets like Paul de Vree and Henri Chopin experimented with recording technologies to mix sounds, voices, and sound effects, creating sound poems that could only exist as recordings. This is yet another example of how production, storage and dissemination technologies have an impact on poetry, at times transforming it into something not witnessed before.[53]

Computers have become increasingly apt for multimedia compositions, particularly since most come equipped with sound cards and speakers. Musicians such as Moby use computers almost exclusively to compose and produce their works. Computers have also made it possible for many poets to explore the potential of adding a sound component to their e-poems. Writers like Jim Andrews take the exploration a step further by creating works of interactive music, like Oppen Do Down[54] and Nio.[55]

[53] There is a rich sound poetry tradition I have only hinted at, and an excellent online resource for its exploration: UBU WEB (http://www.ubu.com).
[54] http://www.vispo.com/vismu/OppenDoDown.htm
[55] http://www.vispo.com/nio/index.htm

So what are some possibilities for the use of sound in e-poetry? There are several different types of sound recordings possible for use with e-poems, and I will use these to organize my discussion of the aural component in electronic poetry.

1. Noises:
    a. Some kinetic works attach noises to the movements of the words on the screen. For instance, *Faith* by Robert Kendall has the word "logic" fall on and bounce off of the word "Faith," making a clinking sound when they make contact. This reinforces the illusion of solidity of the words—yet playing on their meanings at the same time.
    b. Ambient sounds can also communicate volumes, such as establishing a situation or setting. *Bus* by David Hadbawnick and Mark Gergis uses the sounds of a bus and its occupants to establish a visceral connection to the setting for the reader. Its introduction is purely aural, leaving a black screen to focus the readers' attention on the sounds.
    c. Some e-poems link sound to input cues, such as *How I Heard It* by David Knoebel, where all the reader is presented with visually is nine circles arranged in the middle of the screen. A mouseover on any of these circles, will trigger a sound associated with a bar fight, and rapidly triggering them all will create a complete aural picture of the experience.
2. Music:
    a. Whether original or borrowed, it is often put forth as a soundtrack for the e-poem and can go a long way towards setting the tone or mood of

the work. Duc Thuan's *Chronicle of Deaths Forgotten* uses a looped excerpt of opera music and choruses in conjunction to images of the Statue of Liberty to contrast with what the text itself is saying about the statue's construction. The grandiosity of the music contrasts sharply with the text's comments on the small lives that were affected by it.

   b. Jim Andrews has been exploring the possibilities of interactive music, as has been previously discussed. His work *Nio* combines graphical animation elements with looped original music and creates an interface for the user to combine them. Part of what is interesting about this piece is that it is almost like creating a new language for the user to experiment with: he/she can combine elements visually, musically, or both.

3. Verbal:

   a. Readers need to become careful listeners when part of the text is presented aurally—at least if they want to get the whole text. David Knoebel makes clever use of overlaying verbal and visual text in *Thoughts Go*, forcing the readers to decide what text they will devote their attention to.

   b. At times the audible text is the same as the visual component, which can serve different purposes. Often hearing a verbalization of a written text can place stress on different parts of the text, making aspects like tone clearer. At other times, it just gives another way of perceiving the work, like in Tammy McGovern's *Meaning Effect*.[56] In this e-poem, the words appear both visually and aurally, but at such a rapid pace that they are almost unreadable, almost unintelligible. I believe this near-reception of language is the point of this work, and the reader must use both senses to understand it.

The use of sound in e-poetry has become more prevalent since authoring programs such as Flash and Shockwave became the industry standard. These programs allow for seamless integration of sound and visual elements, allowing for responsiveness, careful scheduling, and perhaps some mutability. HTML and other authoring programs do not allow for such careful integration, because sound elements are loaded as needed, causing potential delays in the presentation of the aural element. The performance of the two different systems can be compared in David Knoebel's *Click Poems* (HTML) and *Thoughts Go* (Flash). The complexity of Jim Andrews' *Nio* was authored through Macromedia Director, and published as a Shockwave file. There are certainly many other ways to incorporate sound into e-poetry, but these two are the most commonly used.

I believe the computer, and by extension poetry "native" to it, is a mostly visual medium that is slowly incorporating sound into its workings. Most navigation and interactivity, for instance, occurs through visual and not aural cues. I also believe that the use of aural elements figures prominently in the future of e-poetry, because the silence associated with reading is linked to print technologies. While it is true that

[56] http://www.acsu.buffalo.edu/~tm22/meaningeffect.swf

texts speak when they are read (whether aloud or silently) on the page, it is thanks to screens and speakers that they have literally begun to dance and sing.

Onwards

This chapter has sought to establish crucial differences between texts in print and digital media by exploring how notions of how print texts work become quickly problematic when applied to digital media. This does not invalidate the sophisticated approaches textual theories have developed to approach the materiality of texts—it simply recognizes that they can only take us so far in our ability to analyze the materiality of electronic texts, and that new critical tools become necessary to do so. For that reason, this chapter developed a typology of behaviors exhibited by e-texts as a tool for describing their programmed actions and reactions. The next chapter will test the ideas proposed here through close readings and analyses of key electronic poems by Jim Andrews.

Chapter 3: Jim Andrews' (Vis)Poetics

> You discover alternative approaches to poetry in just about all this work, attempts to synthesize arts, media, and fields such as programming and mathematics or music and recorded sound. As well as attempts to write of the poetics of such practice. It's about putting it all together, connecting, staying human, discovering the nature of our altered humanity and language so that we can address life with fresh insight and communicative power (Andrews "Jim Andrews' Vispo.com").

Jim Andrews is a programmer, poet, and musician who explores the poetic potential of language in the computer by synthesizing his interests in arts that are often kept separate. He creates poetic texts that can be described in terms of their programming codes or behavior as well as by their linguistic and graphical codes. His writing is as much about interface, permutation, chance, music, and animation as it is about what the words say or mean. As a matter of fact, some of the linguistic texts are consciously prosaic or antipoetic—not "poemy poems" as Jim Andrews would say—perhaps to focus the reader's attention on some of the other features of the text. The beauty and wit often lie in what his language *does*, which is inseparable from what it *says*. This chapter will analyze several of his e-poems, exploring his strategies for "putting it all together," in order to get a sense of his poetics. As a writer of digital poetry for over 10 years, Jim Andrews serves as a prolific representative of the practices and poetics of this emergent scene in contemporary poetry.

Andrews seems very willing to give up some of his authorial control over the texts in order to share it with his audience. This raises methodological questions about the reading, analysis, interpretation, and evaluation of his poetry. These questions will be initially addressed here but developed further throughout the chapter, as part of the analysis of a selection of Jim Andrews' electronic poems.

1. How does one approach the work of a poet who weaves the reader into the very functioning and presentation of the poems? One cannot assume that other readers of his e-poems will make the same choices as the analyst. Then again, the choices are rarely limitless. Writers of e-poetry purposefully design the work to respond in particular ways to specific input cues, which readers may or may not activate. So giving a choice to readers or providing spaces for interactivity does not necessarily mean a relinquishing of authorial control. It simply allows for the readers to express their agency within the parameters set for them. Of course, it is each reader's prerogative to respect or subvert his or her scripted role in the design of the text.
2. How does one account for the interwoven agencies of the writer, the reader, and the behaviors programmed into an electronic poem in a way that proves to be insightful to others? One can imagine a spectrum from high authorial control over a text to a high reader control, but how does one approach those works that fall in the middle ground of these two extremes? This is why taking a text-centered approach can be useful: once the instructions are encoded into behavior, their potential for authorial or readerly agency are set.
3. If, as according to Andrews, "each word is a kind of little language widget," how does one read the behaviors exhibited by an electronic text ("Digital Langu(im)age")? There is no fixed way: each e-poem has different behaviors and

different combinations of behaviors. Not everything is always relevant simply because it is present. The trick is to think about how knowing about a given behavior provides insight on the signifying strategies of the e-poem. Textual behaviors are also often tied to the development of hardware and software over time, so analyzing them in their technological and historical contexts can provide insight on the possibilities explored by a particular electronic poem.

4. And what can such analyses suggest about Andrews' poetics specifically, and more generally about the poetics of electronic poetry? Jim Andrews' writing practices when creating electronic poetry are not only linked to programming, hardware, and software, but also respond to his interest in Burroughs' cut-up techniques, Concrete Poetry, Lettrism, and some aspects of Language poetry. This is apparent when we see texts literally cut up and shuffled through mouse movements (Stir Fry Texts), a poem drifting away from the scene of traditional verse into a scene of concrete poetry (Seattle Drift), dancing letters accompanied by rhythmic vocalizations in interactive visual music (Nio), and as a reader is able to navigate a metaphorical poetic ship to shoot up texts floating on the space of the screen (*Arteroids*).

This chapter will study these questions and inspirations and how they are put into practice throughout Andrews' work. Before embarking on this exploration of Jim Andrews' e-poetry, some background information on his life and work should provide useful context by which to approach his oeuvre. A major source of data in this chapter will be electronic correspondence with Jim Andrews, essays published by him that accompany his e-poetic works, writing "hidden" in the source code, and even comments and elaborations by Andrews on drafts of this study. So even if the Author has been proclaimed as dead, the writer is very much alive, and is a generous and invaluable contributor to this study.

Poet and Programmer

What has led Jim Andrews to his current status as a full time poet and programmer, with over 15 years of experience writing for a medium that is scarcely older than that? This question will be answered through a narrative of his life and education as it pertains to his poetic development. A parallel narrative of relevant developments in computer hardware and software, programming languages, and the rise of the Internet will contextualize his growth as a poet who writes almost exclusively for publication in electronic media.

A formative event in Jim Andrews' childhood came to him courtesy of one of the great technoformalist poets of the 20th century: William Carlos Williams.[57]

> I decided to be a poet when I was ten after reading "The Great Figure." It had seemed to me, from about the age of 5 till I was 10, that there were just some things that language couldn't convey. "The Great Figure" restored my faith in language at age ten ("Re: notebook").

This Imagist poem, originally published in 1921 at the end of Williams' book *Sour Grapes,* consists of language that bursts into the imagination vividly with the sounds

[57] The term "technoformalist" is used by Charles Bernstein in "The Art of Inmemorability" to refer to theorists such as Walter J. Ong and Havelock, who examine the formalist impact of technologies such as orality, literacy, and poetry. Williams took advantage of the typewriter as an inscription technology to control the use of page space in his poems, which contributed to his development of the triadic verse form, for example. The label of technoformalist is fitting for poets such as Williams, Cummings, Olson, Howe, and many others.

and colors of a fire engine in a dark and stormy city night. Such powerful use of fairly quotidian word choices can shape a young poet's mind in ways that reach deeply into his poetics. Andrews' own word choices, favoring everyday diction and the use of color, can be seen throughout his poetic career and in his most recent works.

When Andrews studied at the University of Victoria, he took a course in introductory programming in 1979 (in which they were using punchcards to program) but he dropped out, being much more fascinated by his English and Math courses.[58] He studied English and Mathematics, two fields that held equal interest for him, but were not taught in any interdisciplinary way. "I could spend as much time solving a math problem as writing a poem" (Andrews, "Re: on 'code poetry'").

On the literary side of his education, he developed his interest in Modern poets such as T.S. Eliot, Ezra Pound, W.H. Auden, and Wallace Stevens, as well as in postmodern poets like John Ashbery and Charles Bernstein. He describes Wallace Stevens as a favorite and an important influence in his formation as a poet.

> I liked his big brain and flights of imagination, the way he made philosophy poetical, the way he uses blue and green, which are central colors on vispo.com. He used blue and green symbolically: blue for imagination and things of the air, the sky; green for things of the earth, natural, and so on ("RE: notebook").

The use of color in "The Great Figure" and in so many of Williams Carlos Williams and Wallace Stevens' poems is clear to anyone who has experienced their work. The

[58] In retrospect, Andrews comments, "little did I realize that the computer as language machine and as computational machine is a profound uniter of the literary and scientific" (Andrews, "Re: on 'code poetry'").

poem “Disillusionment of Ten O’clock” by Stevens is a clear example of the potential vividness of the imagination represented by colors, exotic animals, and periwinkles. As will become clear throughout the rest of this study, Andrews uses color as an integral part of his visual poetics, which is anchored in the impact of Williams and Stevens in his formation as a poet.

His education and interest in both English and Math led him to develop in other areas, such as Ancient Greek literature and philosophy.

> I also read a lot of Greek literature and philosophy in translation. Aeschylus, Sophocles, Euripides, Herodotus, Thucydides, and the Pre-Socratic fragments. And read a lot of works about the pre Socratics and Greek culture and philosophy by people like F. M. Cornford, E.R. Dodds, John Burnet, W.K.C. Guthrie, and others. The Greeks and their literature, philosophy, and mathematics are sources of wonder, beauty, and insight that civilization must always remember, understand, and be appropriately grateful for. Their influence on our lives continues to this day (“RE: notebook”).

In 1983, Jim Andrews graduated with a B.A. in English and Math from the University of Victoria, in Canada. For the six years that followed, he produced two literary radio shows, *Fine Lines* and *?Frame?* that focused on sound poetry, audio writing, and avant garde literary works. Andrews describes the significance of this period: “it was how I first related the literary and electric technology, and it's where I first learned to be creative with technology” (Andrews, “Re: on 'code poetry'”). This was a period of experimentation with analog sound editing, which usually involved razors and tape, techniques that resonated with William Burroughs’ theories of the cut-up.

> It was all analog technology. Cassette decks, reel-to-reel tape decks, mixing boards, a few effects, and a razor blade to cut and splice reel-to-reel tape. That's where I got interested in the cut. The cut can be interpreted in many ways. The wound. The splice. The transition. The joining. The juxtaposition. On and on. The cut is a source of great energy and art in audio work. I started to produce my own audio art. I little bit of it is at the bottom of the page at http://vispo.com/audio. Sound poetry (Andrews, “Re: on 'code poetry'”).

The cut is a foundational device in his artistic creations, showing itself as early as in his sound poems, taking center stage in the *Stir Fry Texts*, and most recently emerging in his interactive audio works *War Pigs* and *F8MW9,* among others.

His early audio work also exposed him to the work of sound poets, such as Gregory Whitehead, Helen Thorington, and Susan Stone. Their work, along with his interest in Marshall McLuhan’s writings, led Andrews to the conviction “that there was more interesting work to be done and listened to by treating radio and recorded sound as artistic media, rather than transferring work from print to radio and recorded sound” (Andrews, “Re: on 'code poetry'”). This period helped Andrews shape his poetics into one that took into account the production technologies and the materiality of the media he worked with. For instance, he was less interested in writing traditional poems in order to read them out loud into a microphone than in creating sound poems that were designed to maximize their sonorous qualities. In addition to the theoretical and poetic formation that this period fomented, Andrews produced a collection of sound poems titled *Cassette Radio Video Destabilizer and Audio Poems* (1989). The sound poems “Woork of Aart,” “Love Song,” “Poetry Craft,” and “Song” are part of this collection currently available at vispo.com. They all take language as a starting point and find their center of gravity somewhere between music and speech, between verbal and nonverbal vocal sounds, between "natural" voice recording and an edited recording.

During this time period he also started to do word processing on a PC 286 at the radio station, which led him to return to the university in 1989 to study computer science and math for 3 years. His first computer was "a PC 386 with a 100 Mb hard drive. That was in 1990, I think. It was running DOS and, a bit later, Windows 3.0" (Andrews, Re: on 'code poetry'). He used CorelDraw and PageMaker to create a one-issue literary magazine, titled *And Yet* (1992), which published work by Joseph Keppler, Trudy Mercer, and contained two of his first visual poems: "The Collected Sayings of Time," and "Snapshot in the Continuing Adventures of I." These visual poems used CorelDraw to transform ordinary letters and words into statuesque three-dimensional figures, kind of like capturing morphs moving in space. This is a motif that he explores further throughout his career, informing much of his visual poetics, including his recent e-poem "A Pen" (2007). His visual approach to language was only one of the areas that developed from Andrews' use of the computer: this was only the beginning of his creative and professional explorations of this tool.

After leaving his studies around 1992, he worked as a freelance programmer, technical writer, and math and English tutor while continuing to pursue his interests in writing visual poetry (using CorelDraw and other applications) and music (as a drummer in a band). He learned Delphi and Visual Basic and used them to create several applications, among them the mathematical game "CoLoRaTiOn," the card game “Kings,” and the “The MORPH TEA Applet,” which powers a number of image morphs published on his Web site. During this time, he was honing his skills as a programmer, visual poet, and musician while working odd jobs. He was also corresponding and collaborating with his mentor and friend Joseph Keppler and the “Seattle crew.”

In 1995, he started his shift to Web writing by inaugurating his Web site www.vispo.com (as a schedule of events for Mocambopo—a poetry reading he organized and hosted at Mocambo, a coffee shop in Victoria) and collaborating with Web artist Florian Cramer in the Neoist “Seven By Nine Squares” project. The emergence of the World Wide Web during this period really broadened the horizon of possibilities for him, leading him to meet other digital writers, such as “Ted Warnell, Talan Memmott, Claire Dinsmore, David Knoebel, Jennifer Ley, Reiner Strasser, Philippe Castellin, Miekal And and some other digital writers on the net.” (Andrews, Becoming a Full Time Web Artist) The rich exchange of ideas, collaboration, and exploration of new hardware and software technologies for the Web inspired Andrews to develop his poetics in a new direction: writing for the computer screen rather than writing for the page or audio tape. During this time, vispo.com evolved from being a public online schedule for Mocambopo, to being an virtual gallery space for his visual poetry (hence the title “vispo”), to becoming his online “book,” what he refers to as his *Leaves of Grass* (Walt Whitman’s single book to which he added during his lifetime). Jim Andrews decided to self-publish in order to have complete control over his work—and be independent of the politics and technological limitations of traditional publication venues.

He lived in Seattle from 1997 to 2000 and “worked as a technical writer and solutions architect there for networkcommerce.com during the Web boom,” spending the rest of his free time developing and writing on Vispo.com (Andrews, Becoming a Full Time Web Artist). These were very productive years, in which he spent long hours during the night programming his e-poems and developing his Web site. During this period he individually and collaboratively wrote electronic poems in DHTML (Dynamic HTML) such as the *Stir Fry Texts, Seattle Drift, Enigma n* and *Millenium Lyric,* as well as other shorter visual and musical works—the foundation that led him to become a full time Web writer and artist. These Web-based electronic poems represent a continuation of ideas and approaches to writing from earlier in his career, with an added layer of inspiration from applications, operating systems, and the culture of computing that he had steeped himself in, as is the case with the *Stir Fry Texts*.

> The stir frys are a kind of textual analog of the interactive audio work. The stir frys involve interactive layers of text and sequences of text just as the interactive audio work involves layers of sound/animations and sequences of these. Further, the 'wreader' does not compose texts from scratch with the stir frys, but instead explores/composes with 'content' that is already within the piece. This is true of Nio and the other interactive audio work I've done so far, though it is easy to see that both could be developed as tools alone. But I am not drawn to making tools alone. (Andrews, The Art of Interactive Audio)

In 1998, he started using Macromedia Director because he found it to be the most sophisticated tool for creating interactive audio work for the Web. His interest in music and audio poetry could finally find expression through the tools of programming. He had developed an interest in interactivity through his DHTML works, particularly the aesthetics and poetics of application interfaces: menus and toolbars, both of which are prominent in the work written and programmed during this period in his career. The visual aspect of these graphical user interfaces, as well as that of language are always present in his audio work: what he calls "vismu" (a portmateau of the words "visual" and "music"). Poems such as *A Rude Little Song*, and *Oppen Do Down*, while working with a similar sound palette (recorded *a capella* vocalizations by Jim Andrews) provide an increasing interconnection between the aural and the visual, and are prototypes of ideas that reach fruition in *Nio*.

In the past six years he has developed several versions of a poetic computer game titled *Arteroids* (currently in version 3.1), has continued to produce new work, namely *dbCinema* (2007, in progress), *A Pen* (2007), *Jig-Sound* (2007, in progress), and *War Pigs* (2008). Two of his recently published projects are *On Lionel Kearns* (2004) and *First Screening by bpNichol* (2007) a collaborative recoding and restoration of a set of electronic poems written in 1984 by Nichol. He continues to create and collaborate, publishing his own and other people's work on Vispo.com, such as Ana Maria Uribe's concrete and animated poetry, and *F8MW9* (2008) a recent collaboration with Margareta Waterman.

In 2001, his work on *Nio* earned him a grant from the Senior Canada Council to devote his full attention to digital writing, focusing on interactive audio, which allowed him to develop *Arteroids* (2001-4). Since then, he continues to subsist on income he makes from freelance programming, such as the sales of his application

Windows for Shockwave, and the honorariums associated with the appearances he makes as a Web artist and poet. From 2003 to 2005, he worked for a digital musician at the University of Victoria programming in Director for "installation and performance-based work."

It is interesting to note that like mathematical and literary giants Gottfried Wilhelm Liebniz and William Blake, Andrews throughout his development and career as a Web artist and poet has usually had to find a "day job," which makes more financially rewarding use of his skills. And he is well aware of this connection, as he comments in the following e-mail:

> We learn [in the book *The Universal Computer: The Road from Leibniz to Turing*, by Martin Davis] for instance, that Leibniz had a day job. This towering intellectual giant, inventor of differential calculus, original philosopher of the monad, and early father of the computer was employed by the Hannover family—to write the Hannover family history. Hannover eventually became king of England. Leibniz, of course, wanted to be in England. Newton was there and London was the center of a mathematical frenzy of activity spurred on by the invention of calculus (simultaneously but independently   invented by Newton and Leibniz), but Hannover said, no, stay there [in Germany] and finish that history. (Andrews, "Re: on 'code poetry'")

Andrews doesn't charge for his art. There is not a speck of advertising in Vispo.com. There is not even a link to Web services that allow for his readers to make donations to him. The sales on Windows for Shockwave haven't been a major source of income in recent times, and he has announced that he will release the code for free in the near future. On the other hand, he has a very marketable skillset as a programmer, which has allowed him to make a living through specific “day jobs,” much like Blake using his skills as a printer and engraver during the late 18th and early 19th centuries to earn a living, while engraving his own poems and illuminated texts when he had the time and money. As for Blake, whose illuminated printings were never commercially successful, Andrews’ art, music, and poetry are a labor of love that has earned the admiration of his peers, an international audience, and favorable critical attention.

His writing project with Vispo.com is substantial and ongoing. It integrates many of his inspirations and interests throughout his career and allows readers to find connections among recent and early works. Jim Andrews describes his Web site as follows:

> Vispo.com is an attempt to create a literary work alternative but related to the book; to create works and experience imaginatively attuned to the media and methods of the Net. Being truly literate involves not only reading but writing; vispo.com is an attempt to write through new media. It is my life's work; and the work on vispo.com by others and in collaboration with others is a huge part of the nature of that life and work to put it all together, to make strong connections. The French poet Isou said "Each poet will integrate everything into everything." And this was way before the Net. Same job, different time and circumstances (“Jim Andrews’ Vispo.com”).

The next section will explore what components of “everything” Andrews integrates in the creation of his literary, artistic, and musical works. As suggested by his experience in radio, music, programming, and poetry, his development as an artist has provided him with a uniquely suited background for this field of electronic writing.

Theoretical Influences and Inspirations

Jim Andrews' biography brings up several influences that helped shape his thought concerning writing in new media and to this day serve as inspirations for his work in interactive sound, electronic poetry, and programming. Andrews seems to be the hands-on type of learner and experimenter, which is supplemented by his interest in theory. This section will first touch on some of the theories that inform his practice and then elaborate on the practices that arise from these.

Marshall McLuhan's studies of media and its impact on human cognition and communication was an early influence for Jim Andrews. By producing two literary radio shows and reading McLuhan, he became increasingly aware of the differences between reading poems published in print and performances of sound poems designed for audio recording and radio. "McLuhan saw, or heard, say, radio and television as introducing a very strong element of orality back into North America and the west" (Andrews, "Reading McLuhan"). This orality in poetry represents increased use of sound elements of language, such as rhyme, alliteration, assonance, consonance, and meter—in ways that are not currently in vogue in "literary poetry," as Dana Gioia suggests in *Disappearing Ink: Poetry at the End of Print Culture*. In his production of his radio shows, Andrews was able to witness this shift, and adjusted his programming to maximize the use of radio as a medium for poetic expression. Furthermore, he changed the types of poetry featured in his radio show to include more sound poetry, and audio writing.

> The effects of technology do not occur at the level of opinions or concepts, but alter sense ratios or patterns of perception steadily and without any resistance. The serious artist is the only person able to encounter technology with impunity, just because he is an expert aware of the changes in sense perception. (McLuhan, quoted in Andrews, “Reading McLuhan”)

More importantly, Andrews’ poetic practice moved increasingly towards the visual and sound boundaries of the poetic spectrum: where language flirts with being seen as visual art rather than read transparently, heard as music rather than listened to as language. His poems during his “pre-Web” period show an interest in the Concrete poetry tradition, both visual and sound, as can be seen in his sound poems published in the 1989 *Cassette Radio Video Destabilizer and Audio Poems* collection, and his visual poems available in pdf format at Vispo.com.

In his early work, Jim Andrews engages not only the aural and visual elements of language, but also the materials they are recorded in, as influenced by William Burroughs and the cut-up. “I became interested in his work when I was working in radio: Burroughs applied his cut-up technique to audio tape; he's one of the seminal figures in audio writing. Also, he has written with insight about the medium of recorded sound” (Andrews, “On William S. Burroughs”). His own work with radio gave Andrews a sense of the tactile dimension of sound media, cutting, splicing, and manipulating audio tape, scratching vinyl albums, and employing elaborate equipment to shape the sounds he broadcast in his show. To employ a pair of scissors to cut magnetic audio tape and taping random pieces together to create a new audio text is a type of blind remixing of sound because one cannot predict with accuracy what or where one is cutting and joining. To do so as part of poetic experimentation is to treat language as matter, not as thought or concept: a central element of Andrew's poetics.

It is hardly surprising that Andrews' is inspired by the physical characteristics of each medium, as well as the technologies employed to create and manipulate the objects that carry texts. When he started experimenting with computers in the 1980s and formally studying programming in the early 1990s, one of Andrews' goals was to learn the inner workings of a new writing technology. This was the time of early graphical user interfaces which led him to study programming languages such as Visual Basic and Delphi (also known as Object Pascal) which are graphical and object-based programming languages that emerged from Basic and Pascal, both of which were based on text and syntax. These computer languages shaped how Andrews sees computers and their relation to language:

> Computers are language machines. Some say they're math machines: they're computers, they compute. But they simply carry out instructions encoded in machine language even when they do math. They don't so much multiply or add, divide, etc numbers as they shift bits around according to instructions encoded in language. The gears of the machine are made of language. Language gears. Language widgets. Langwidgets. Their operation is entirely predicated on our understanding of the formal properties of language that support near flawlessly repeatable parsing, tokenization, interpretation, compilation, and execution (Andrews, "[-empyre-] Poetry and Programming (2): Computers are Language Machines").

He does not see computers, programming, and poetry as necessarily separate fields. For him, writing poetry that is purely auditory and is recorded in magnetic tape, writing poetry that is primarily visual and inhabits the page, or writing poetry that dances with its readers in computer screens are all an extension of the same interest in the materials and capabilities of language he has been doing since the early 1980s and will most likely continue to do for the rest of his poetic career. “The spirit of poetry, intense engagement with language, can both stay at home and venture out in spirit into new relation with arts, media, and programming” (Andrews, “[-empyre-] Poetry and Programming (2): Computers are Language Machines”). His awareness of the medium, materiality and formal properties of language are major influences on his poetics.

Poetic Influences and Inspiration

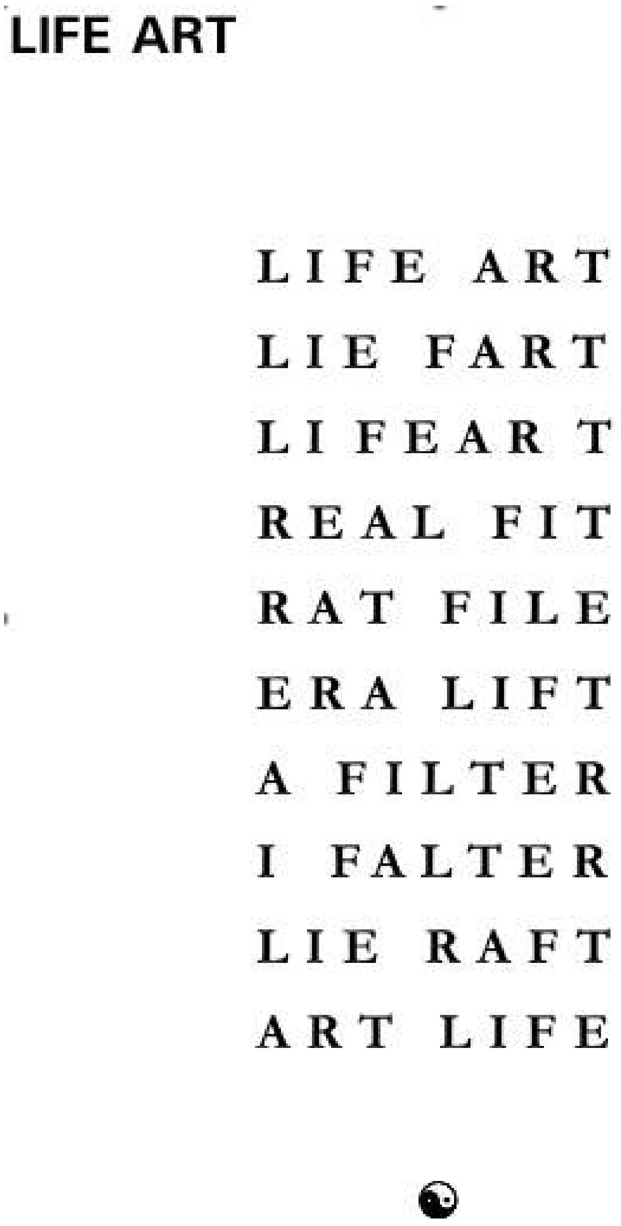
As far as poetic influences on his poetic practice, three main schools stand out: Concrete Poetry, Lettrism, and Language Poetry. All three movements engage language in small units: that is, they are more interested in the poetics of phrases, words, letters than in the rhetoric of sentences and the measures of the poetic line. Poets within these poetic schools also tend to write open works, inviting readers to participate in the completion and interpretation of their poems. Most importantly, these movements seek to bring the reader to examine their use and understanding of language through the experience of the poems, though one could argue that all poetry worth reading reinvents language for its readers.

Jim Andrews has long enjoyed reading and writing concrete poetry, a poetic movement that had limited immediate influence in the United States and England, but which was well received internationally.[59] In Canada, poets like Steve McCaffery, BP Nichol, Bill Bisset, and others explored this and other served as inspiration for Andrews, and he collaborates and corresponds with concrete poets such as Celia Regina Pinto, Marco Niemi, David Daniels, the late Ana María Uribe, and others. In recent work, Jim Andrews has written tributes to, and ports of the work of early poets of electronic media Lionel Kearns and bpNichol, both of whose work was very connected to Concrete Poetry. The *Stir Fry Texts,* a set of collaborative pieces Andrews began in 1999, was recently added to in 2006 with a set of five "Concrete Stir Fry Poems" by Marco Niemi. Andrews own work goes beyond reinventing and collaborating with others to create visual poetry: his first strong stage of artistic production was with visual poems.

His 1992 concrete poem "LIFE ART" was "the opening poem of the unpublished manuscript *Several Numbers Through the Lyric"* (Andrews, Online Writings and Vispo) The fact that this poem was soon after turned into an animated poem for the Web demonstrates how Andrews' affinity for concrete poetry influences his electronic poetry. "LIFE ART" consists of 10 lines that recombine the letters in the words LIFE and ART to create 10 different sequences of words that eventually return to an inversion of the original statement "ART LIFE." As the opening piece in his manuscript and as a central part of the design of his VISPO page on "On-Line Writings," this poem can be read as a statement of Andrews' thoughts on the relationship between life and art, art and life. Judging from the progression in the

[59] In *Digital Poetics*, Loss Pequeño Glazier suggests there is a strong link between concrete poetry and electronic poetry, an idea I have supported in earlier chapters of this study.

sequence of word combinations, the speaker is shifting from prioritizing life and leaving art as secondary to art taking over his life. (See figure 1.)



LIFE ART

LIFE ART
LIE FART
LIFEAR T
REAL FIT
RAT FILE
ERA LIFT
A FILTER
I FALTER
LIE RAFT
ART LIFE

☯

Figure 1: Jim Andrews, "LIFE ART"

For a poet for whom life is more important than art, to “LIE FART” shows an allegiance to the needs and priorities of life. This speaker has “FEAR” and seeks a “REAL FIT” and yet reaches a “RAT FILE” in the fifth line of the poem—something that has negative connotations. The sixth line of the poem begins a progression towards more positive connotations "ERA LIFT" which suggests a period of improvement. The next two lines "A FILTER" and "I FALTER" can be read as a series of measures that filter life or language to favor artistic creation, though not without self-doubt. The next to last line of the poem, "LIE RAFT" represent a surrender from the speaker, who lies down in the symbolic raft of art in order to let his life drift in the direction art takes him, an idea reinforced by the final line "ART LIFE." The speaker has gone from being a dilettante (someone for whom art is something that he fits into his life) to becoming a true artist—a person whose life is determined by his art.

The fact that this poem was written in 1992, a time in Jim Andrews' life when he was developing as both a programmer and visual poet, as well as its placement in Andrews' two collections of poetry (*Several Numbers* and *Vispo.com*) underscores its significance in terms of what it says about his poetics and aspirations. The electronic version of the poem (published circa 1996) completes the transition for Andrews from a writer of concrete and traditional poetry to a writer of electronic poetry. His art had become his life.

The electronic version was created with a Java applet he created called Morph Tea (an anagram of "metaphor"), which allows the display of animated sequences of images in a Web page. Instead of displaying the sequences of words arranged in lines, the electronic version displays the lines sequentially in the same space, at a rate that can vary from 1 to 60 frames per second, depending on where the reader's pointer is placed on the image space (left is slower rate, while the right speed up the rate). This version adds 9 new lines to the poem, such as “FILE ART,” “TAR FILE,” “EAR LIFT,” “ALT FIRE” which refer to his interest in elements of programming and interactive audio. This version doesn’t create the same sense of a linear logical progression as the concrete version because it contains 19 lines that loop without any pause. The concrete version has a clear sense of a beginning and an end which presents the inverted word order and opposite sense of priority. The electronic version enhances the sense that this is an ongoing meditation through a set of ideas centered on the original “LIFE ART” statement, and perhaps that is a more accurate representation of Jim Andrews’ current approach of artistic intent.

Despite Andrews’ shift to digital and networked arts, he continues to identify his work with concrete poetry—though he prefers the term “visual poetry” to represent his work, as he states in his 2002 interview.

> I prefer the term 'visual poetry' to 'concrete poetry' because 'concrete poetry' has more historical specificity than 'visual poetry', I mean 'concrete' to me refers to a certain period of visual poetry and a certain often mimetic approach to the work. The term 'visual poetry' has some historical baggage too, I'm sure. I'd thought when I bought the domain vispo.com that it was my own term, but the term 'vispo' preceded my use of it, not surprisingly (Andrews, Becoming a Full Time Web Artist).

Andrews is very aware that his poetics are part of a larger tradition of visual poetry than the movement that emerged in the late 1950s, though he would not exclude them from his practices. Simanowski makes a case in the interview for Andrews’ work to be considered as concrete more than visual poetry because “concrete poetry draws attention to the material qualities of the language: graphic forms of letters, font, size, color, constellation on the page and to each other." I agree with Simanowski up to a point. Andrews' poetry seems more aligned with the poetics of Concrete Poetry than with the broader tradition of visual poetry at first sight, but as will become clear in the section titled "The Electric Pen," his range of techniques exceeds those used by Concrete Poetry, and leads him to towards Lettrism.

The key goals for *Lettrisme* (translated into English as Lettrism or Letterism) is the interrogation of our customary use of language for signification by reducing it to its smallest units, letters. It even goes beyond the letter and the phoneme to create its own glyphs and nonverbal sounds with which to reinvent language and poetry free of the ideological frameworks that come with traditional signification. Lettrism emerges from Surrealism, Futurism, and Dada and sought to replace these as the dominant avant garde in the 20th century. The influence of this movement in Andrews' poetics is evident in works such as *Nio*, *Arteroids*, and most of his "Animisms," because Andrews focuses on these small units, exploring the meanings of letters rather than words.

> It isn't really the poets themselves as influence concerning lettrism. It's the idea. As a poet-programmer and visual poet, letters are endlessly fascinating to work with. In all sorts of ways. When you're programming, they make great brushes, for instance. And they are basically square compared with long thin words, so you can transform them in often more interesting ways, and move them around quicker. As material, they are more generally plastic than words. But you can use them linguistically as well as as material brushes and tokens

> and monsters etc. So you can both deal with writing, can write, and also create computer games and animations and interactive things that synthesize arts (Andrews, Re: Notebook).

The result of the Lettrist influence in Andrews' poetry shows when he goes beyond the concrete engagement of the visual and sound aspects of language to an atomistic level that is less connected to traditional signification. Concrete poems direct attention to the material qualities of written or spoken language and integrates these qualities with the meanings of the words, creating the effect of the poem, what they describe as "verbivocovisual." Letterism emphasizes the material components to a greater extreme in two ways: 1) its basic unit is smaller—letters rather than words, and nonverbal rather than verbal sounds—and 2) it abandons, if not actually subverts, the meanings of words. Andrews' work shows the influence of both movements, at times in the same poem, as is the case in *Arteroids*, where words explode lettristically.

Language Poetry is another movement that informs Andrews' poetics. Even though this is a poetic school with a variety of practices and no unified manifesto or set of principles, it has several characteristics that can be observed in Andrews' poetry.

1. Interest in language games and innovative constraints, following in the tradition of OULIPO, and Surrealism.
2. A blurring of the lines between theory and art, often employing essay-like writing and prose in their poems.

3. A rejection of “official verse culture,” as exemplified by the deeply personal, bard-like poetry of the confessional, post-confessional, and other mainstream poetic movements.

The result of these characteristics in Andrews’ poetry shows in a tendency to employ essay-like language that directly discusses theoretical concepts in some of his poems. The third characteristic often results in manifesto-poems, self–referential poems, and a preference for open works in which there is either no apparent poetic voice or in which the reader’s intervention is crucial to complete the poetic performance.

Language poetry has many practices and practitioners, and while Jim Andrews makes reference to some language poets, such as Charles Bernstein, Dave Ayre, Lisa Robertson, Andrew Klobucar, and others in his writing, he does not consider himself a languag poet *per se*. His poetic center of gravity leans towards engaging the technologies that make his electronic poems possible, seeking out the work and play these allow him to do, and the community of like-minded writers that has emerged since the beginnings of the Web in the mid 1990s.

Technological Influences and Inspirations

> As a poet, I am less involved in writing poemy poems than in creating algorithms that operate on language, image, and sound, hopefully to interesting result. Whether the algorithms cut it up or do other stuff. dbCinema is a kind of graphic synthesizer. Nio and Jig-Sound are audio sequencers. The stir frys are text sequencers. I create sequencing and synthesizing algorithms. I synthesize media and arts (Andrews, Re: ELO Conference Proposal).

There are several aspects about the networked and digital media that inspire Andrews to program and write his e-poems. These aspects are programming itself, specifically seen in what he calls architexture and langwidgets, the possibilities of interactivity and chance, and the multimedia capabilities of the digital computer, particularly the tools it provides to shape sounds into music. Programming is both a set of tools and an inspiration that serves Andrews' poetics well by freeing words and letters from their traditional roles and interrogates his readers' relation to writing.

The possibilites programming places at Jim Andrews' disposal are a major source of inspiration for him. He loves to program spaces, environments, and behaviors for his texts to inhabit and take life in, and he will often create all that before he has written a text, as he states in the source code for his "Millenium Lyric."

> I wrote the engine before I figured out the content of the poem. With some of this sort of work, you first write the engine thinking there's a poem in that engine somewhere. Then you try to figure out the best poem the engine has in it. Like a piece of stone to a sculptor (Andrews, Source Code for Millenium Lyric).

For Andrews, an e-poem is literally and figuratively "a machine made of words," as Williams famously wrote, and an integral part of its design is how it's going to operate when read, and what kinds of controls will the reader have at his/her disposal. The words or "content" of the poem emerges from his own play and exploration of the possibilities of the machine he has created, and I'm certain that as a poem takes shape, so does the "engine" that governs its behaviors. Jim Andrews creates spaces and word machines for readers to interact with differently from those those interactions they have with poems they find on the page, in order to lead them into poetic experiences that are more in tune with his own sensibilities. Andrews expresses this idea during his participation in the Empyre discussion forum:

> I find that in my own work, often how it proceeds is, first, some long time in writing the code and creating the accompanying graphical interface that supports the interactive operations on the content. This process is quite long, and throughout it, I'm thinking of possible content for the piece, and am also thinking about the efficacy of letting the player/reader/wreader supply part or all of the content, as well as providing content myself (Andrews, [-empyre-] form and content).

By creating spaces for the readers to add content, customize the reading, or heighten awareness of their role in the production and reception performance of the text, Andrews is able to expand the possibilities of writing and reading. Sometimes he is less interested in the words used than in the relationships of the words inserted into the behaviors and roles created in the e-poem, as is the case with *Arteroids*. Jim Andrews has not only created an e-poem based on the interface of the classic Asteroids video game, but has created a textual space in which readers/players/wreaders can insert texts of their own into the environment and rhetoric of this textual space. His Lettrist and Language Poetry influences show themselves in the practice of presenting alternative relations among words than are provided by grammatical and linguistic rules.

The flow of language, the pull downwards of line breaks and enjambment, the articulatory shaping of the words in the human machine, the “hands that can grasp” and “eyes that can dilate” all are results of a poem’s “running,” and it is the reader who powers the engine to set that machine in motion: as page turning, mouse gliding, button clicking, eye movement, thought, articulation, sounds, experience.

In addition to creating environments and interfaces for his readers to encounter his texts, programming allows Andrews to write language that is charged with purpose and behavior. Each word and letter is potentially a small machine of its own, what he calls a langwidget.

> But each object might have various properties in addition to its usual appearance and meaning and place amid other words. My piece *Seattle Drift* is an example of such a text. When you click the text that says "Do the text", the words in the poem eventually drift independently off the screen. Each word has its own behavior, its own partially random path of drifting off the screen. Each word is a kind of little language widget, langwidget (Andrews, “Digital Langu(im)age -- Language and Image as Objects in a Field”).

Andrews treats words as objects—and he describes them as things that have appearance, meaning, placement, and behavior. The last is made possible through executing programming in a computer and is a key to understanding his approach to poetry and programming, which is informed by object-oriented programming[60]. Anna Katharina Schaffner describes this approach to behavior as follows. “Behaviour can be inscribed into letters, and letters are put on scene like actors, words end up doing

[60] “Object-oriented programming (OOP) is a programming paradigm that uses "objects" – data structures consisting of datafields and methods together with their interactions – to design applications and computer programs. Programming techniques may include features such as data abstraction, encapsulation, modularity, polymorphism, and inheritance” (Wikipedia contributors “Object-oriented programming”)

something, like floating around, exploding, drifting off the scene, dancing, exercising, changing their size or colour etc" (Schaffner). Andrews imbues words and letters with kinetic properties, sounds independent of what they represent when read aloud, responsiveness to user input, or random aspects to their appearance or behaviors.

The linguistic meaning of the word-objects is at times secondary to the characteristics of the object, as Andrews discusses in relation to the *Stir Fry Texts*.

> The stir frys dawned on me when I realized it's easy to create an object, fill the object with whatever content you want (text, images, sounds, etc) and subsequently change the content however you want. The changes to objects can be triggered by mousing or clicking objects or by some other form of interactivity. This leads to an unusual view of a piece of literary Web art which I want to develop here--not just concerning the stir frys, but more widely for the future (Andrews, "Architecture and the Literary").

If treating his electronic poems and art as a series of objects is foundational to his poetics, so is the notion of interactivity—or programming responsiveness into his digital objects. Andrews creates poetic objects and invites his readers to interact with them, incorporating the reader's input into the signifying strategies of the e-poem. In a large degree this is true of all poetry and the act of reading in general, something Andrews is well aware of, as he discussed in the following excerpt from his interview with Randy Adams.

> In making 'interactive' works, whether they're interactive in the ways we associate with computer/person interaction or in the ways we associate with poetry on a page—which of course is also interactive—or email or IRC, say—

> which are totally interactive—you seek to engage and to be engaged meaningfully, deeply, intensely. And of course this also implies 'playfully' as in any good relationship. The poet Michael Ondaatje said 'Seduction is the natural progression of curiosity', or something like that (Andrews, Defib: Randy Adams interviews Jim Andrews about Nio).

Seduction is a key strategy for Andrews, who creates interfaces evocative of applications and computer games, seeking simplicity that arouses curiosity in the reader. For example, the options menu in *Seattle Drift* is placed much like the menu headers in an application such as a word processor, but instead of the informative (and uninviting) descriptors, Andrews labels them as “do the text,” “stop the text,” and “discipline the text.” Since the personified voice of the text says that it wants to be “done,” the reader curiosity is instantly activated, begging the question: What happens if I “do the text?” A detailed reading of what happens when that responsive textual object is activated is presented in Section 2.2 of this chapter titled “Drifting from the Scene.”

Andrews doesn’t seek interactivity for its own sake, however. His engagement of the networked digital computer as a medium for artistic expression inspires interactivity in and of itself.

> The computer itself is a very interactive thing, and the Web is also very interactive—between people and also between people and works/apps. It's a communications thang, yes? As the Web gets more broadband and also as compression and streaming technology are marshalled to provide more sound, animation, and video, the question arises whether the Web just turns into some

> commercial variant of the telephone, TV, radio, etc. I'm sure there will be a lot of passive and conventional uses of the media/um. But one of the things that attracts me to the Web and to the computer more generally is that you drive the thing quite actively or it doesn't go anywhere (Andrews, Defib: Randy Adams interviews Jim Andrews about Nio).

The driving metaphor is an apt descriptor for the way one seems to move from site to site on the Web, while remaining physically in the same place. Browsing the Web and reading a book—two activities traditionally contrasted in discussions about interactivity, activity versus passivity, and the act of reading (usually to the critic's demise)—both require an active participation from a reader, but a distinction can be made by invoking Espen Aarseth's notion of the ergodic. I believe that Andrews means that "driving" the computer and the Web requires nontrivial effort to achieve. Clicking on a link or hotspot in a page (or turning a page in a book) doesn't require much thought, but the choice it represents may, and that is what distinguishes the two activities. Andrews in his poetry invites his readers to make some effort to reach a decision in how they approach the work, whether it is by playing along with the conceits that shape the poem, or freely playing with the responsive objects, or customizing their reception of the text. The reader then becomes the engine that produces variation and brings to fruition the multiple possibilities set designed into his electronic texts.

Mutability, or the ability for a text to change, is another key aspect of Andrews poetics, who coined the term "combinatorium" to represent his notion of a range of possibilities, whether they are literally textual or interpretive.

> X does not mean whatever we want it to mean.
>
> There is a range of possible valid interpretation. This range is limited. The meanings anything can have form a combinatorium of possibilities. The word “combinatorium” is one I coined, I believe. It evokes the sense of a space, a set of combinations distributed over a (limited) space.
>
> Poem as combinatorium of meanings. Some intended. Some not. Some anticipated. Some not.
>
> One of the things I appreciate about art is that it often leaves lots of room for people to do their own thinking, their own imagining, their own creating. Part of what I try to do is create an engaging space of possible meanings and make that space intriguing to the wreader's imaginative engagement so they have lots of room to create for themselves what the piece means to them (Andrews, “Re: DHTML Dances”).

Andrews uses the term “wreader” to refer to the combination of “writer” and “reader” a term coined in the early days of hypertext theory that suggested that the reader had become like a writer in their interactions with hypertext. Used more broadly now to interrogate the traditional notion of reading as consumption rather than creative engagement, “wreader” points in the direction of a person who manipulates responsive elements in an electronic text to make changes in the text they read.

For example, the impact of one’s interaction with one of Jim Andrews’ e-poems is not nearly as much “writing” as interacting with a blank page in a word processing program. In the first case, one’s interaction is circumscribed to a finite range of possibilities that Andrews defined when designing and writing his e-poem.

In the second, the range of possibilities is enormously larger, limited by one's vocabulary, culture, and creativity. So while the former could be considered reading or even "wreading," in the end what one has done is to creatively trigger variations in Andrews' text, discovering the expressive range of the work. In the latter, one is writing, the product of which couldn't be attributed to anyone but oneself.

But not all of Andrews' interest in mutable texts is necessarily linked to the reader's interaction. He uses a variety of techniques to achieve permutation and randomization of elements in his texts, carefully controlling the parameters to achieve his desired results. He is also interested in the history of computing and one of its greatest challenges: teaching a computer how to write.

> One of the interesting things I learned in computer science is that meaning or semantics is very much more difficult for computers to deal with than syntax. A computer can parse text syntactically with relative ease. But to parse text for meaning, for semantics, is a very deep issue, partly because texts are almost always very ambiguous. But not only are they ambiguous, their semantics also depends on the world view of the parser, whereas the rules of syntax do not depend on the world view of the parser. The only 'world' the syntax parser needs to know about is very small indeed: the rules of syntax. Whereas to construct the meaning of natural language sentences, one needs to know more about the words than simply their syntactic properties (Andrews, "Re: DHTML Dances").

Andrews retains control of the semantics of his textual choices, though he is willing to share that control with his readers. Syntax, on the other hand, he is happy to give control of to the computer, through randomization or responsiveness. So while the text of an e-poem can be completely restructured syntactically, the end result is still limited in its compositional and expressive range, as informed by Andrews' world view.

The notion of mutability will be discussed in much more detail in Section 2 of this chapter, "DHTML Dances" because it was with his DHTML poems that Andrews really began to explore this through programming. He had already done so with analog technology during his radio production and sound poetry days, inspired theoretically by William Burroughs, and poetically by Helen Thorington, Gregory Whitehead, Joseph Keppler and others—a period that will be elaborated in Section 3 of this chapter, "Visual Music."

In the section that follows, "The Electric Pen," this study will focus on how his poetics develop through the exploration of software tools for the creation of visual poetry. The conversation between the growing capabilities of computers and networks and Andrews' poetry is a constant in the sections that follow, examining three areas in which he develops his poetry: visual poetry (from static to kinetic), sound poetry (from static to responsive), and code poetry (from objects to applications). An idea that will become clear throughout the rest of this chapter is that Andrews doesn't simply use new software tools just because they're new and perhaps fashionable, his exploration of their capabilities responds to his engagement with language in a variety of media, including programmable media. His background and career as a programmer provide him with the expertise to carefully select and modify his tools—and that has a tremendous impact on his poetic and artistic creations.

The Electric Pen

Jim Andrews' e-poem "A Pen" presents letters twirling on the screen, leaving colorful traces of their passage as they inscribe the space provided them by the browser window. At the bottom of the window a toolbar provides options for readers to customize the experience: iconic controls affect the speed, color, nib size, and other variables in the three pieces[61] that constitute the work— "Niolog," "O," and "Time." Help is displayed in a small text box on the left side of this toolbar when the pointer is placed over the icons, but if the pointer isn't being used this way a text is displayed in this text box, at a rate of a couple of words every 6 seconds.

How does one approach such a work? Do we look at the animated words that fill the screen and read the text below as it cycles through? Does one analyze the motions of the letters on the screen, as well as the traces they leave behind? Does one look at the animations, but read the screen captures? Are the linguistic texts in this work secondary to the concepts or technologies being employed in this piece? Focusing on any single one of these questions only begins to unpack what this poem is all about, because it is about much more than what the texts spell out.

"A Pen" is an exploration of text as a tool for writing, rather than as the result of writing. It is about the interpenetration of code and language in programmable media to imbue letters and words with behaviors and allowing the poem to emerge from their play. It is about creating tools for the readers to become involved in the process of shaping the poems that arise from these processes. Last but not least, it is a further development in Jim Andrews' lifelong exploration of the visual characteristics

[61] Would the terms "verses" or "movements" more accurately describe these three components?

of written language, and the capabilities of computers to both render it and reinvent statuesque letters as dancing signifiers that respond to input from the reader.

In order to best follow Andrews' exploration of written language in digital media, this section approaches "A Pen" by contextualizing it three ways:

- As an electronic poem by Jim Andrews: how it is an expression of his poetics.
- As a poem in the tradition of Concrete, Letterist, and Language poems.
- As an electronic object: its programming language, capabilities, and designed behaviors.

Integrating these approaches helps establish the poem's signifying strategies, leading to one or various interpretations. I believe this approach to reading electronic poems by Jim Andrews can both serve as a model for other readings of his work and as a way to read, analyze, and interpret electronic poetry in general. [62]

When Jim Andrews started Vispo.com in 1995, he named it after a portmanteau of the words "visual" and "poetry," both of which captured what his poetics were primarily about at that stage in his artistic career: experimentation of the computer's ability to transform language into poetry through intense engagement with its materiality.

> When I started my site, I knew barely anything about Lettrism or Language poetry and had only a casual acquaintance with Concrete. My work grows more from the challenges of the media I work with than from the traditions of poetry. However, these traditions, techniques, types of poetry, etc, are useful

[6262] This analysis is influenced by three approaches in New Media theory: Loss Pequeno Glazier's discussion of the computer as a space of poesis, N. Katherine Hayles' call for media-specific analysis, and Matthew Kirschenbaum's forensic study of first generation electronic objects and the technologies that support them.

> and very relevant to those challenges. But when I started my site, I did so out of an interest in exploring all the possibilities of the Internet in a literary venture. The visual, the networked, sound, programming, and so on. My training in literary matters was relatively conventional. No courses in Concrete Poetry or Language Poetry or Lettrism. The University where I studied is quite a small one that had no strong avant garde elements except, say, Lawrence Russell in audio. But the challenges of literary radio and, later, poetry on the Internet, together with my training in Computer Science and Mathematics, equipped me for the directions I have chosen. And Concrete, Lettrism, and Language poetry have just been useful approaches related to the challenges of media I've dealt with rather than strong allegiances (Andrews, comments on Chapter 3.1 The Electric Pen).

When contemplating and reconstructing the career and development of a poet like Jim Andrews, it is easy to construct a linear narrative of influence and results, in which chronologically earlier poetic movements such as Concrete poetry, Letterism, and Language poetry are largely responsible for the electronic poetry that followed. While there are surely many cases like this, the conditions under which electronic poetry emerges are in some ways radically different from the artistic and technological milieu from which these movements arose. Concrete Poetry and Letterism, for instance, are both poetic and artistic movements that emerge from the late age of print—a time in which, from the beginning of the 20th century writers and artists in movements such as as Futurism, started employing visual characteristics of written language in meaningful ways, culminating in movements that explored the visual for its own sake. In the same way, new technologies for writing (such as the typewriter) and printing also left their mark on poetic production. The same kind of impulse is what inspired Andrews development as a poet: intense engagment with the tools and materials for the composition, inscription, and reproduction of language. His intellectual development along with his interests in computers, poetry, and other technologies, led him down a path that produced similar results as the Concrete and Letterist poets, but the context in which he reached it was different. This is not unlike Pierre Menard's version of Don Quijote—which the narrator of the famous Borges story considers far superior and much more daring than Cervantes' version, despite being word for word identical linguistic texts. The Concretism and Lettrism that Andrews creates, reinvents, and then uses to conceptually inform his poetic work is significantly different from the works produced within those movements. Still, examining these poetic movements is a fruitful source of insight for some of Andrews' visual poetics, because of the affinities and to a certain degree influence on his work.

Concrete and Letterist poetry focus attention on the surfaces of texts, blurring boundaries between the visual arts and literature. Of the two, Letterism is more extreme because it has less of a commitment to meaning and the word, at times inventing alphabets and glyphs in order to free language from signification, subverting the centrality of meaning so valued by mainstream poetics and language use. Language poetry, particularly in the tradition of Charles Bernstein, blurs traditional boundaries between poetry and theory, adopting a prosaic language to further subvert the bardic voice prevalent in "official verse culture." All three movements engage language in small units: that is, they are more interested in the poetics of phrases, words, and letters than in the rhetoric of sentences and the measures of the poetic line. Poets within these poetic schools also tend to write open works, inviting readers to participate in the completion and interpretation of their poems. Most importantly, these movements seek to bring the reader to examine their use and understanding of language through the experience of the poems, though one could argue that all poetry worth reading reinvents language for its readers in some way.

Probably the strongest factor in Jim Andrews' formation as a visual poet was through Seattle poet and polyartist Joseph Keppler whom he interviewed several times and produced a radio profile for his radio show *Fine Lines* in August 24, 1988 (Andrews, Vispo ~ Langu(im)age Audio). Keppler helped Jim Andrews shape his visual and sound poetry as conceptual art, as defined in one of the Keppler interview segments Andrews chose for this radio profile:

> What I mean by the conceptual. . . it's a kind of art that draws your attention to the frame of mind which was necessary first to conceive and then follow through in the conception of the art and the simplicity of the work brings that out. I think it is one of the necessary aspects of the work to make the piece of art truly conceptual it has to look so simple that anyone could do it because then they're thrown back upon themselves in a self-examining manner. . . he [Joseph Keppler] challenges his audience to look at their own creative process (Thomas).[63]

[63] Radio show available at URL: http://www.vispo.com/audio/index.html.

This approach to art, whether it is visual, aural, verbal, procedural, or a combination of some or all, pervades Jim Andrews' work. By zoning in on a few features of electronic writing at a time, Andrews challenges his readers to reconceptualize their own use of computers for writing. In his early visual poetry, he employed color, backgrounds, textures, shading, positioning, and other features of written language to treat them as visual objects. His DHTML pieces, such as "Enigma n" focus on the reader's ability to play with the graphical and behavioral features of the texts displayed. In "A Pen" and other recent works, he challenges his readers' creative process by providing them with controls to customize some of his texts' features, like size, opacity, color, speed, and so on. The apparent simplicity of his work therefore highlights the aspects of electronic writing he wishes to focus on in his poetic explorations, as well as challenge his readers to reexamine their own reading and writing practices with the software tools at their disposal.

> Graphics programs such as PhotoShop, CorelDraw and PhotoPaint, Flash, Xres, Freehand, 3D Studio, 3D Extreme, 3D Dream, Authorware, Director, DreamWeaver, Word, Premiere, etc. are out there, do not require rocket science to use, and are great toys that most people like to play with, given the opportunity. Of course, harumph, they're very serious toys.
>
> And these programs deal graphically with text. Text as graphic. Text as object. Text as object no different from a graphical object. You create objects in these programs, and the way you create and manipulate objects remains the same whether the object is textual, graphical, sonic, procedural ('neath

> textually directive) or otherwise. (Andrews, Digital Langu(im)age: Language and Image as Objects in a Field)

Jim Andrews' sense of playful experimentation with the possibilities different software tools offer him comes across clearly in the quote above.

If we were to divide Andrews' visual poetry into periods according to the types of software or programming languages he employed to create his e-poems, we could divide it into three. During the first one (1991-1995) he used raster graphic "painting" programs as well as desktop publishing software to create static visual poetry designed primarily for the page or gallery space. In the second period (1996-1999) Andrews designed Java applets and used DHTML programming to create scheduled, kinetic, responsive, and mutable visual poems. The third period (2000 to present) begins when he shifts to Director to create visual poems that are more like applications than purely visual or kinetic poems.[64]

During the first and second periods, Andrews used Corel Draw, PhotoPaint, and other desktop publishing and graphic design software to create static visual works conceptualized at first for the page or gallery space and then for his Web site Vispo.com. Some of these pieces were created for an unpublished manuscript titled "Several Numbers through the Lyric," while others were exhibited in a 1995 gallery exhibition with Joseph Keppler in the Mocambo coffee shop. These works are collected in Vispo.com and grouped into four series titled: *Ã*, *Ê*, *Î*, and *Ô*.

[64] To divide Andrews' work into "periods" is a convenient artifice to make some focused observations into tendencies in Jim Andrews' practices, but this shouldn't be seen as either purely linear or marking sharp boundaries between practices. Andrews' use of software tools isn't technological determinism, either: he explores the possibilities offered by the tools and technologies, but he also chooses them carefully depending on what he wishes to achieve.

The earliest pieces shape the letters as objects, cutting into them, applying shading, and using other tools to imaginatively impart depth to the characters he employs. For example, “The Inner Razors of N,” “S Having a Bird,” and “aA” all give character to the letters they are about, inviting readers to reconceptualize them as sharp (see figure 1), graceful and maternal (see figure 2), or deteriorating typographically (see figure 3).

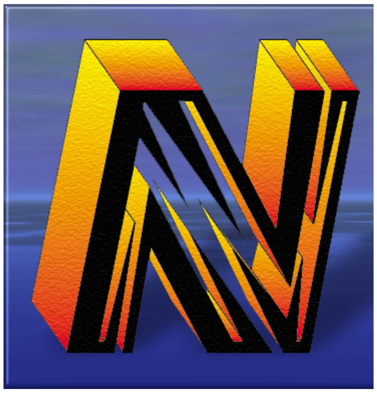


Figure 2: "The Inner Razors of N"

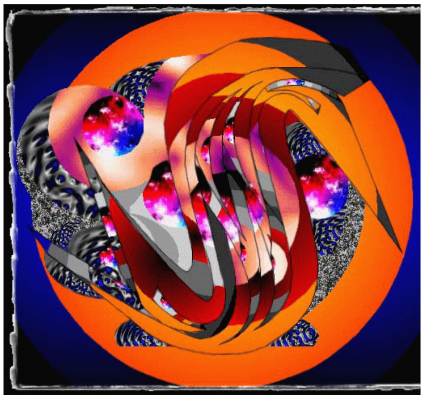


Figure 3: "S Having a Bird"

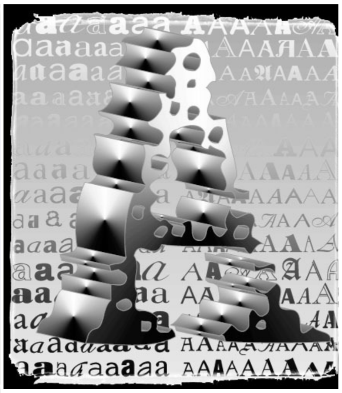


Figure 4: "aA"

The presence of backgrounds is also noteworthy in these visual poems. In “The Inner Razors of N,” the horizontal line across the middle of the background, and the shadow cast by the letter N help make the letter three dimensional by creating a sense of perspective. In “S Having a Bird,” the concentric circles with colors that shine through the letter S create a sense of a nest, along with the title. The background in “a A” consists of the letter A in both upper and lower cases in a variety of fonts, and the piece is accompanied by the following text: “Swiss metal A./ Further evidence/ that typography/ is deteriorating badly/ (if not rusting)” (Andrews, a A). In this particular visual poem the background is a counterpoint to the large metallic “swiss-cheese” A in the foreground that suggests an answer to the question of why typography is deteriorating.

Computers, particularly with word processing and desktop publishing software, place so many typographical options at a writer's disposal that two extreme (though not uncommon) reactions are to either just use the default font settings, or create excessively formatted documents. Andrew critiques this practice by evoking both the solidity of metal and the controlled decay that makes Swiss cheese and presenting it as evidence of typographical "deterioration." Is this letter being put forth as a mockery of applications and operating systems that create for their users typographically ridiculous fonts? Applications may offer a varied palette of fonts, but don't accompany these with any kind of historical information on the fonts, their uses, and the cultural baggage they bring to the page they are used in. How far can typography deteriorate before it loses all meaning? The A on the foreground cannot deteriorate much further and still be an A. Or is this an ironic statement, meant to critique attacks on digital typography from typographers who prefer the well developed and historically charged typography of print? With such a reading, deterioration becomes progress. After all, the carefully controlled decay that makes the holes in swiss cheese is also what gives it its flavor. Perhaps Andrews is suggesting that it is a matter of taste (pun intended) and that any materials that allow for the development of typography are healthy.

The other main type of visual poetry Andrews created during these first two periods suggests motion or transformation through spatial representation. See figures 4 and 5 for examples these types of visual poems.

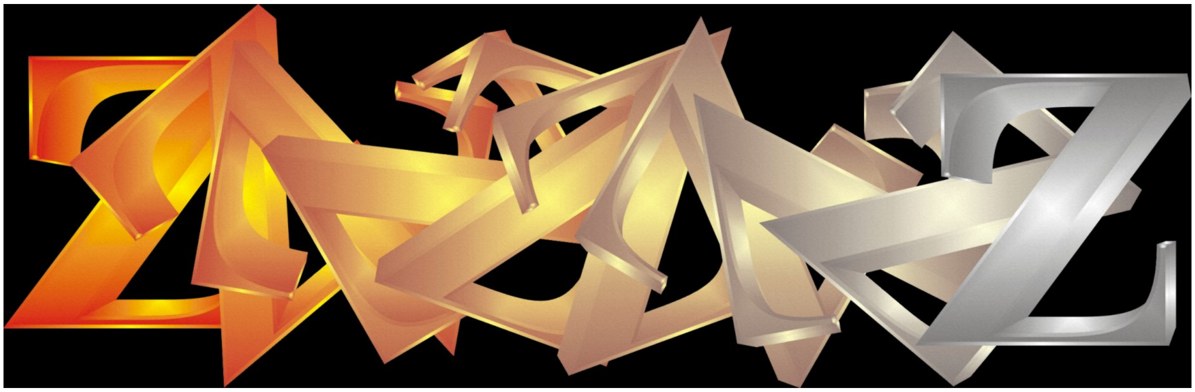
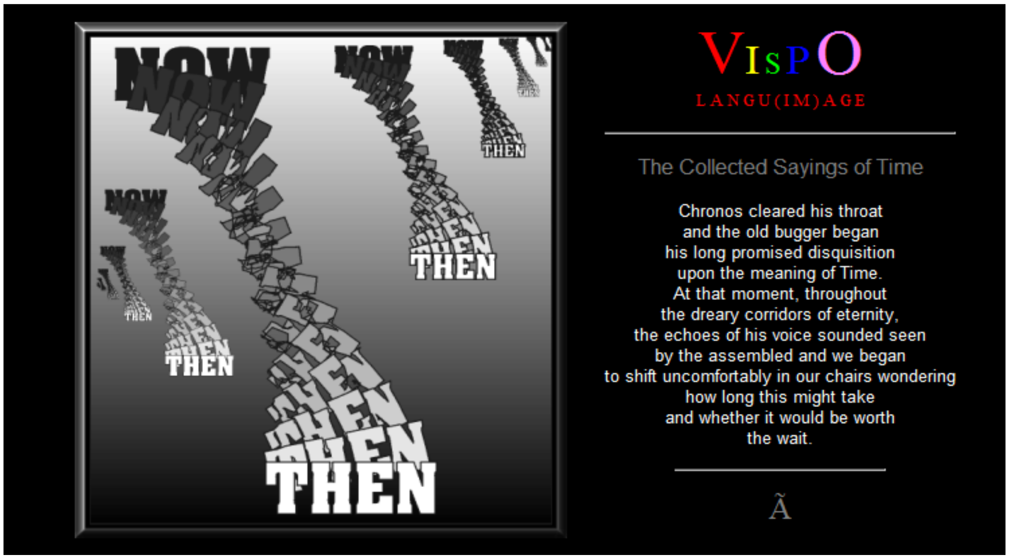


Figure 5: “Zzzzzzzzzz” (third frame)

The ten Zs in this visual poem could be described as rotating as they progress along a horizontal line that moves from front to back (or back to front), arranging the Zs three-dimensionally. A progression in color from silver to gold (or vice versa). An interesting detail when analyzing this suggested progression is that whether one sees this as moving from left/back to right/front as it rotates counterclockwise or the other way around, the rotation goes counter to the linear progression—that is, if it were a ball it wouldn’t be rolling in the direction it is moving in. The fact that the letter Z remains the same if rotated 180 degrees (unlike p, which would become d), allows one to imagine the progression differently, with a shift in rotation direction every three or four Zs so that they seem to be walking across the linear space of the poem. If read out loud, this visual poem becomes a famous onomatopoeic expression of sleep or snoring, which is reinforced by the title’s formatting: “Zzzzzzzzzz.” The reversals in rotation can be seen as reversals of breath, in and out, since breath in only one direction cannot be sustained for long, except perhaps in that sleep of death that lies beyond the alphabet. This reading could go further in exploring the visual poem, but the most important point is that the poem’s suggestion of motion through spatial arrangement of three-dimensional characters (note the shading on the letters) on a two dimensional space, points toward an interest in textual kinesis, which will mark Andrews' shift into different software and programming tools and the beginning of a new period in his writing.

The other aspect that he attempts to capture on a static two-dimensional surface is transformation, as is the case with the morph, as can be seen in the visual poem below.



The Collected Sayings of Time

Chronos cleared his throat
and the old bugger began
his long promised disquisition
upon the meaning of Time.
At that moment, throughout
the dreary corridors of eternity,
the echoes of his voice sounded seen
by the assembled and we began
to shift uncomfortably in our chairs wondering
how long this might take
and whether it would be worth
the wait.

Figure 6: "The Collected Sayings of Time"

The columns are created throught the linear arrangement of stages in morphing the words "NOW" and "THEN" along a diagonal axis, with a similar three-dimensional arrangement to handle overlap. The diagonal axis from left to right also guides how the intermediate steps between NOW and THEN are displayed because the beginnings of both words are the center of a fanning of the words, giving the illusion that NOW in the background is falling into THEN, which is in the foreground. Why is THEN foregrounded, instead of NOW? This reversal corresponds with the counterpoint created by the rotation of the Zs in the previous visual poem, and both are consistent with Andrews's Letterist challenge to the reader's assumptions of meaning in language. A cascading of THEN into NOW would make sense to readers and would reinforce the obvious reading that "then becomes now." Keeping the same arrangement, but foregrounding NOW while placing THEN in the background would also reinforce the obvious reading that "now arises from then" or that the past leads to the present. By reversing the direction and positioning, however, Andrews challenges such facile readings and by repeating the columns along two vanishing points he suggests that these relations have become tiresome through repetition, an idea reinforced by the tone of the text that accompanies the visual poem. Perhaps Andrews felt it was time to redefine the associations between space and time, and between words and their meanings.

This interest in textual kinesis and transformation, along with newly available options for animating text for the Web, explains Andrews' shift to kinetic texts, and his current static visual poems are images captured from his animated texts in motion. Before moving on to the next period in Andrews' poetic and artistic career and discussing Andrews' use of different tools to transform and animate texts, there is one tool from the first period that has transcended its original use and has become instrumental to the development of "A Pen." The tool is CorelPaint's "image sprayer," described in the patent as:

> A method and system are provided for rendering a brush stroke with multiple nibs which are added to a center point indicated by a cursor, and the position and movement of each nib are controlled based on variables which can be set by a user" (Corel Corporation).

Jim Andrews used this “image sprayer” employing letters as nibs to create several visual poems (“Cloud Fire,” “Salad Wreath,” “Word Worm,” and “The Pen”) from 1996 to 1999, and modifying it to create “A Pen” in 2007.

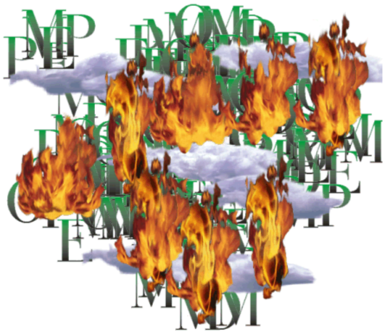


Figure 7: "Cloud Fire"

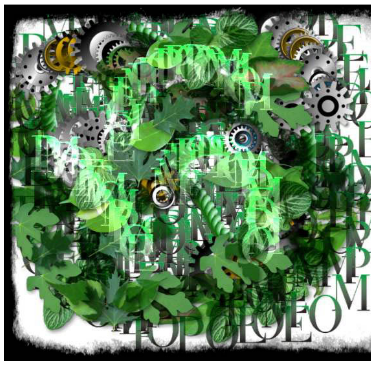


Figure 8: "Salad Wreath"

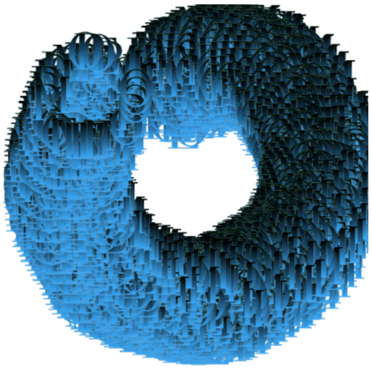


Figure 9: "Word Worm / Bandwidth"

All three pieces use bright colors, evoke three-dimensional spaces, and suggest motion through traces left by the letter shaped nibs across the screen space, particularly in figure 8. The notion of a software pen with letters as nibs that inscribe on the screen space is central to both “The Pen” (1999) and “A Pen” (2007). This tool will be discussed further in the context of a close reading of the electronic poem “A Pen,” along with the earlier suite of visual poems titled “The Pen.”

From 1996 to 2000, Andrews experimented with Java and DHTML to produce kinetic visual poetry, with responsive and mutable behaviors. The letters and words in these pieces, while they still foreground their visual materiality, are still treated as text and their transformations are purely typographical— modifying color, size, and movement, but within the considerable limitations of DHTML. For example, the words can move, but must remain upright, and there is a limited palette of fonts because the multi-plaform nature of the Web emphasizes a minimum common denominator of font options. This period is more about developing textual behaviors, such as responsiveness and mutability rather than about expanding his visual poetics—though the shift to simpler typefaces led him to rely more on design, which is a development of his visual poetics (Andrews, comments on Chapter 3.1 The Electric Pen). The key poems of this period are “Seattle Drift” (1997), “Enigma n” (1998), “Millennium Lyric” (1999) and the *Stir Fry Texts* (1999-2006). A screen capture of “Enigma n” provides an idea of the capabilities and limitations of DHTML that Andrews was able to explore.

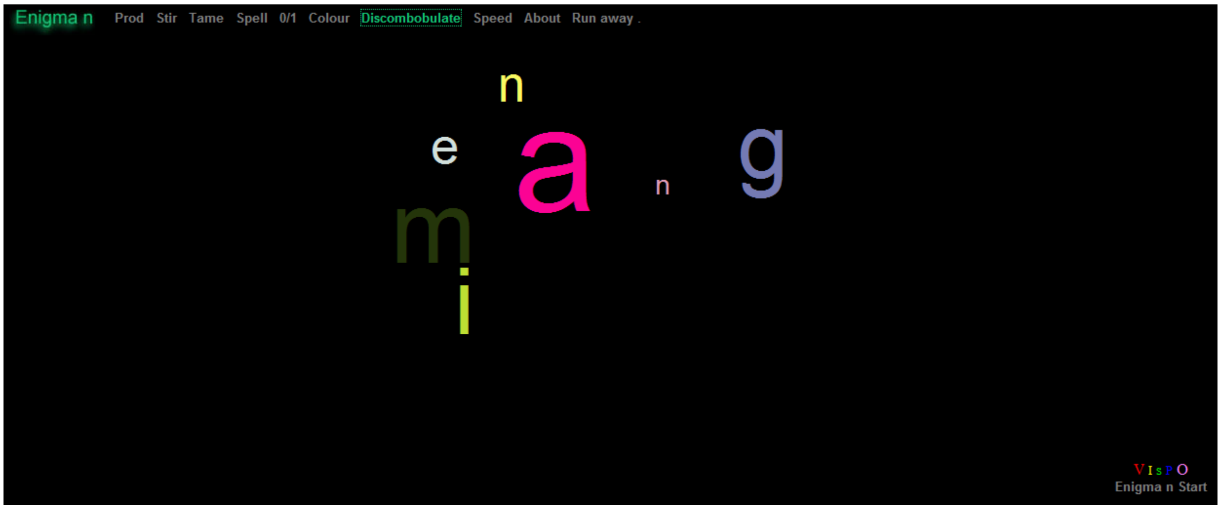


Figure 10: "Enigma n" screen capture

At a glance, we can see how this poem offers a menu of options to the reader, each of which affects the text of the word “meaning” at the center of the screen. I analyze this piece in detail in the next section, “DHTML Dances,” and will instead focus on the significance of this and the other DHTML poems from this period.

This period is especially important because Andrews shifts from treating text as a material object to treating it as a procedural object (much like a component in object-oreinted programming). In other words, the text goes from being the result of writing, drawing, and shaping, to being an ongoing process to which both Andrews and his readers can contribute. Seeing this metaphorically, Andrews went from creating language sculptures to creating language robots, which is his focus in the third period.

The third period is from 2000 to the present, in which he uses primarily (then Macromedia, now Adobe) Director, its programming language, Lingo, and a set of behaviors he programmed for Director called “Windows for Shockwave,” which allows creation and control of multiple sprites and is behind the implementation of “A Pen.”[65] During this time, Andrews creates pieces that build upon his earlier static and kinetic visual poetry, elaborating textual responsiveness and mutability and incorporating sound into pieces like “Nio,” “*Arteroids*” and others. The most significant characteristic is that Andrews now creates visual works as applications rather than as objects, emphasizing process, interface, and customization. An excellent example of this is “A Pen,” which Jim Andrews describes as follows:

> "A Pen" presents an interactive piece and screenshots of the interactive piece in process. [. . .] The screenshots have more composition to them than what you typically see when you play with the interactive piece. The screenshots were created over a couple of evenings of playing with the interactive piece and pressing the 'print screen' button on the keyboard when something looked interesting. Also, the screenshots give a pretty good indication of the compositional range of "A Pen", at least with the current animations (Niolog,

[65] A Sprite is “an independent graphic object controlled by its own bit plane (area of memory). Commonly used in video games, sprites move freely across the screen, passing by, through and colliding with each other.” (ZDNet)

> Time, O). So although the generative range of "A Pen", even just with the current animations, is infinite (it can create infinitely many different graphics), the compositional range is quite finite. Each of the visual poems (Niolog, Time, O) consists of a bunch of animations and a drawing process, and the compositional range is about 20 images, give or take a few.
>
> Poem as small combinatorium of elements. (Andrews, RE: A Pen)

A major distinction between “A Pen” and the static visual poems discussed earlier is that the tools and processes used to create them are complete and inaccessible to the readers. Readers encounter the final product and interact with it to produce their interpretations. In “A Pen” as well as in other works during this phase in his career, Andrews foregrounds the writing tool, placing it at the disposal of his readers and inviting them to play with it. Since its processes are ongoing, the readers can affect them and take part in the shaping of the work. The screenshots are more like the poems discussed earlier: they are the final result of an authorial engagement with the tool, and could be enjoyed equally on the Web, in a printed book, or a gallery space. Another aspect they share is that they are all expressions of the compositional range of the tools used to create them, as is the case with the 1998 piece, “The Pen.”

The central analogy in both “The Pen” and “A Pen” is expressed in both poems, in the earlier piece it is the lexia from the first hyperlink, while in the more recent work it is expressed in the epigraph:

> The guitar, the electric guitar.
>
> The pen, the electric pen.

The shift from the acoustic to the electric guitar evokes for me an image of Jimi Hendrix, reveling in the electric guitar's capability for feedback loops and distortion and integrating them into his musical performances. This analogy highlights the shift from the remediated pen or typewriter in the computer, to the hypermediated pen that Andrews is creating in this work. And in this case, he doesn't mean the typewriter: he really means the technology that powers an instrument that lays ink through friction with the surface of an object: the pen.

The most important part of the pen that Andrews engages in the two "Pen" poems is the nib. The nib is the point that makes contact with the surface one is writing on, leaving a trace of that contact with ink. Various pen technologies attempt to minimize the amount of friction needed to lay as consistently even a layer of ink as possible on the surface being inscribed. The electric or simulated pen doesn't have such constraints, and its relation to real pens is largely metaphorical, as are other simulated tools such as the sprayer, or paintbrush. Electronically simulated nibs can have any shape and size and create many kinds of traces on a bitmapped, simulated, surface, including the shapes of letters. One such electric pen is CorelPaint's "image sprayer," which Andrews used to create the central pieces in "The Pen." Here is a sample of three central pieces from this poem.

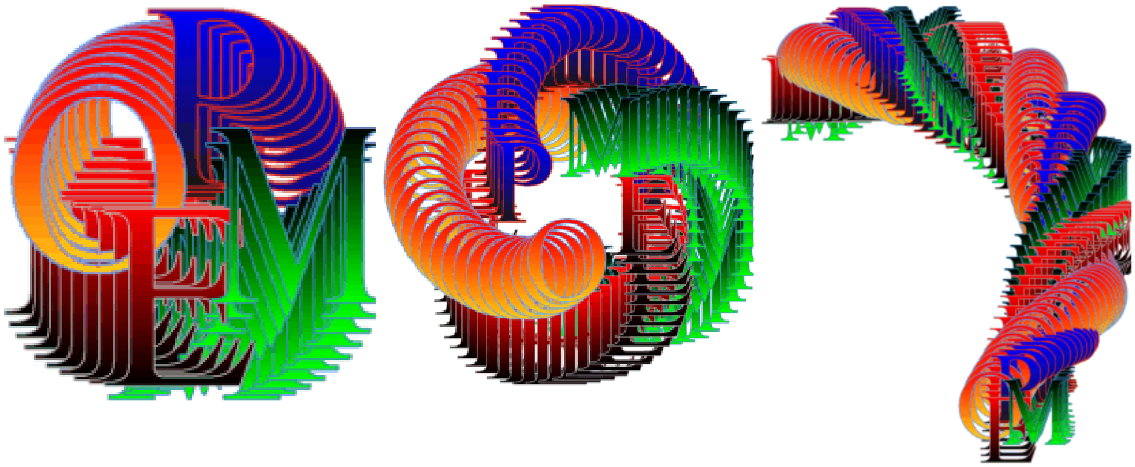


Figure 11: samples from "The Pen"

These examples showcase a motif that pervades all of these pieces: rotation. The first sample has the letters of the word “POEM” rotating a quarter of a circle to end up in the position previously occupied by the next letter. In the second sample the letters also rotate upon the same static axis, but this time they have a wider starting circumference and they seem to be closing in on themselves, perhaps in a collision course. In the third sample, the letters rotate tightly around the axis of a curved line, creating a coloful spiral that moves through the space of the page. In all of these cases, there is no actual movement, but a record of previous movement, left behind by the trace of the letter nibs leaving virtual ink on the surface of the bitmapped space. The circular movement of the letters emphasizes the feedback loop that exists between the poem and the inscription technologies used to create them, as well as evoke a character that has special significance for Andrews: the letter O.[66]

[66] He uses this letter spatially for a number of reasons: an O is an uninterrupted circle or loop, it creates two spaces (inside and outside), it can spin and be rotated in any direction and still be an O (so it is more stable than Z, N, d and p), if rotated along the axis of its diameter it creates a sphere, if moved without rotation along a line through its center it creates a tunnel. Arranging or animating letters in a circle creates a relation between them that is closer, tighter than in a line, as letters are traditionally

Eight or nine years later, Jim Andrews takes the concept to a new level in “A Pen” by making the nibs animations. Here’s how he describes the tool as he has created it:

> For instance, a nib of the CorelPaint "image sprayer" cannot be an animation; it must be a static image; the "image sprayer" is not designed to automatically move about the screen, as happens with "A Pen"; the "image sprayer" is meant as a design tool, not a show unto itself; also, conceptually, the pen in "A Pen" is such that the nibs are connected to the pen via long, loose springs--so long and loose it really isn't obvious that's the case at all--whereas the nibs of the "image sprayer" are not attached to the pen via springs at all, and the results are quite different. (Andrews, RE: A Pen)

So this is not a pen that one can actually control, though one can customize several aspects, such as nib size, opacity, inks and background color, speed and rhythm, through the toolbar below. Note that each part (or should I say verse?) of this e-poem consists of 4 simultaneous nibs, each one with a different animation and trajectory. These animations can be seen in the essay about the “Kandinsky 3” series of artworks generated by a different software tool created by Jim Andrews: dbCinema.[67]

arranged. Andrews has written at least three poems directly about the letter O (“O Exhaling in Ur” (1992-7), “Once Upon a Time O” (1998), and “Millenium Lyric” (1999)), and the shape is ubiquitous in his visual and electronic poetry. Last but not least, the letter O could also be seen as the number 0, which is an essential half of what binary computer language is made of: 0s and 1s.

[67] The Kandinsky 3 essay is available at http://vispo.com/dbcinema/kandinsky3/intro/index.htm. As a critic of electronic literature, I feel that access to the source code can allow one to see its component parts, as well as the logic and variables that organize their processes, and this can be useful for critiquing it. Jim Andrews has been rewarding readers of source code since

The presentation layer of Jim Andrews' e- poems is only one of the layers through which Andrews expresses his ideas. The code, or "neath text" finds a way into the surface in "A Pen" in a way that suggests that the idea behind the e-poem isn't just about the traces created on the virtual surface of this piece. In the toolbar on the bottom of the screen, there is a small help window in which descriptions of the customization icons are displayed. When left alone, however, a 60 line poem is displayed, one line at a time, every 6 seconds, looping back to the beginning after it finishes its 6 minute long sequence. If "you exit the app, it creates a little text file [. . .] that stores the current position of the poem text. When the poem restarts, it resumes with the next line" (Andrews, comments on Chapter 3.1 The Electric Pen). The entire text is quoted below in the following three column table for convenience: read the entire first column and continue on the second and third to reconstruct the linear sequence.

I'm the Help ghost
Help & chatter
Time corridors
Vispo robot
Time hatchery
Don't just sit there
City of time
Meant to be
Poesy machine
Poetry is now
Made of 0's and 1's
& the seashore
& your heartbeat
& what you make of it
Neural microscope
I'm the attendant
I only talk
When you're not busy
I just work here
Every attendant
Is an author
Every author

1997 with short essays and discussion of the goals and mechanics of his works, particularly during the period he was programming and writing poems with DHTML.

An unfinished poem	Brain lang tissue	An attendant
Time machine	DNA language	I just sit here
It takes two t'tango	Lang smoke sign	Taking it cool
Thoughts	Brain time	I live
brain language	100% language	In the neath text
click & drag sketch	Poems grown here	Ramon Fernandez
I'm an o pen	Pomes groan here	Neural lyric
Pen is as pen would	Fish language	Podopamine
Your brain humming	Language machine	Language processing
All the live long day	Non poemy poem	Word processor
Pen songs	Give me a hand here	Brain pen
Look for shapes	Coffee?	Bed pan
	None for me	Language onion skin

Table 2: The Text of "The Electric Pen"

Note that here we have a voice, a personified, self-aware poem, very much in Charles Bernstein's language poetry tradition, as well as in Andrews' own poetry.[68] This voice examines its role, as well as the role of the reader, attempting to encourage the reader into action, which would silence or at least reset the display of the help window text. This can be seen as the tension between the authorial desire to express an idea and the reader's own interaction with the text. The fact that Andrews designed the tools in order for readers to play with this electric pen, cannot be ignored, however, though it leads to some conclusions on this piece.

[68] "The Pop-up Poems" and "Seattle Drift" are two examples.

Perhaps Jim Andrews wants us to entice us with the prospect of playing with the software pen he has created, while at the same time he takes away our ability to control its movement. We cannot write with this pen: we can only shape how it scribbles across the screen. And yet it is when a reader has exhausted his interest in playing with the poem, and pulls his hand away from the mouse or trackpad to look at the textures of the largely unreadable writing produced on the screen space, that the poem's voice emerges from the depths of the code.

So what does this all mean? I hope I have provided the contexts and tools for you to arrive at your own informed interpretations about this work. What is more important is to realize that "A Pen" doesn't emerge from a vacuum: it is an expression with a history and a development based on Andrews well articulated poetics, software tools, and a poetic tradition of that finds expression in many time periods, from the present day electronic poetry scene back to antiquity.

<u>*DHTML Dances*</u>

From 1997 to 2000, Jim Andrews lived in Seattle, and it is the period in which he developed much of his early Web work in his Web site Vispo.com. Most of these early works are either visual pieces, using CorelDraw and similar programs, or programmed works, using DHTML. The latter consist of four electronic texts: *Seattle Drift* (1997), *Enigma n* (1997-8), *Millennium Lyric/Time Piece* (2000) and the *Stir Fry Texts* (1999-2006), which consists of six works: four collaborative and two fully attributable to him. This section will be subdivided into three parts, each dedicated to one of these works by Andrews. Between these three works, we can get another perspective on Jim Andrews' approach towards electronic poetry, particularly during the early days of Vispo.com when he was producing his first works with programmed behaviors.

But first, it is important to define and discuss DHTML as a framework that helped shape these works. The possibilities and limitations that DHTML offered Jim Andrews and other Web writers resulted in a series of works that could almost be considered a subgenre. The limited selection of fonts available in this language motivated Andrews to focus more on the design and behavior of his texts rather than on the appearance, as was discussed in the previous section. Also the paradigm that informs DHTML is that of the document: a document which can respond to a timer or respond to user input, whereas Director and Flash both operate on a timeline paradigm, which is more conducive to blending sound and animation.

“Dynamic HTML (DHTML) is a term used by some vendors to describe the combination of HTML, style sheets and scripts that allows documents to be animated” (World Wide Web Consortium). This was not exactly a programming language, since it employed multiple types of code, but it relied heavily on JavaScript to interact with the browser's DOM (Document Object Model), defined by the World Wide Web Consortium as: “...a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page" (World Wide Web Consortium). In layman's terms, DHTML worked by modifying the DOM properties of a loaded electronic document in real time, making kinetic, responsive, and scheduled electronic texts possible with Web documents. Mutability was also possible in real time, without having to reload the document in order to activate its variables. Christian Heilmann describes the possibilities of DHTML,

> The main goal of DHTML was to make the formerly impossible possible - make pages look very dynamic, move things around on the click of a mouse or with timed animation and generally make web pages more engaging.

Jim Andrews' perspective on this is a bit different:

> I think the main goals of the DHTML initiative were more about making business easier on the Web. For instance, being able to present options for purchases dynamically. Making pages more engaging was probably part of the intent, but it was a Microsoft initiative, and they're big on functional business, not engaging art ("Re: Finally, Chapter 3!").

The case of DHTML is a good example of some of the development model of the Web: businesses develop functionality for their sites, and artists repurpose them for their own needs. The important thing about this initiative was that it opened up the page to on-the-fly changes. Before JavaScript and proper DHTML support in browsers this was science fiction - any change to the document meant a reload. The main problem of DHTML was that the browsers in use were prone to change quickly, and all of them followed a different path when it came to providing the programmer with a DOM to change the page (Heilmann).

Dynamic HTML was one of the factors that helped determine the outcome of the Browser Wars (1997-1999) in which Microsoft's Internet Explorer took control of the market once dominated by Netscape Navigator, and the Web dramatically expanded its mass appeal and usage. During that period, cross-browser compatibility was difficult to achieve in DHTML, requiring redundant code for different browsers to compile, and running differently in different browsers, even when the instructions were successfully ported.[69]

As a platform for the production of electronic texts, however, DHTML was a major source of inspiration for Jim Andrews, who in his essay titled “Infoanimism” describes it as follows.

> But what is DHTML? It allows people to make Web documents that change in appearance and function quickly. More generally, it turns documents into programs. When we look at documents on the Web, we see text and graphics and controls and so forth. But upon understanding the basics of DHTML, we begin to see the 'neath text, what's unseen but present in the source code and begin to reconceptualize the document as a collection of objects with properties that can change as the reader reads. The objects can also respond to changes in other objects or initiate changes in other objects. And changes can be caused either by the underlying logic of the neath text without the reader's intervention or be caused by the reader's responses to the visible manifestation of the document.

The shift of “turning documents into programs” is a significant step in his development as an e-poet, because until this point, Jim Andrews used programs to create documents, but now the product and process became conflated. For someone with a background in programming like Andrews, this is not as new a development as

[69] For an engaging and techically savvy narrative of the Browser Wars, read Peter-Paul Koch's “A History of Browsers” available in his Web site at http://www.quirksmode.org/browsers/history.html.

it may seem, but it is a big step in the way the Web approaches its documents and it was a first step in the direction of Web applications. Before then, it was possible to insert dynamic objects within static HTML pages, an area which Flash and Director came to dominate once their plug-ins became *de rigeur*.

In a recent e-mail, Jim Andrews' explained some of the challenges with programming his DHTML poems originally, and the more recent implementation done with the help of Marko Niemi.

> A few years ago, Marko Niemi translated some of the DHTML work into Finnish. He is a poet-programmer. Not only did he translate it, he upgraded the code. When I wrote *Seattle Drift* and the other DHTML work, it was really hard to make it cross-browser or cross-platform. Most of the DHTML work only worked on IE for the PC. Marko made it work on most browsers and most platforms. Which was very welcome indeed. So I took his code, which was in Finnish, and put the English in it to upgrade the English versions.
>
> Since I wrote *Seattle Drift* and the other stuff, DHTML has become more standard across browsers and platforms. So it's a bit easier now to make it cross-browser and cross-platform. Marko is current in his DHTML skills. I'm not, I've let it slide because mostly I'm working in Director these days (Andrews, “Re: ELO Conference Proposal”).

Even though he has shifted almost exclusively to Director since 2001, Jim Andrews' DHTML poems were the testing ground in which he developed all the programmable behaviors that characterize his work, with the exception of sound--a limitation of DHTML, which acts primarily on visual information. The more important aspect of DHTML as a source of inspiration is that while Andrews had used other devices before to create images and animation, and had experimented with document responsiveness and mutability through links and navigation, it was with these pieces that he found his “voice” as writer of electronic poems, implementing and combining behaviors into them, when before they had appeared in a more isolated fashion.

These DHTML pieces each contain a brief essay or commentary by Jim Andrews within their code, which are strong indicators of his goals, inspiration, and thought process that led to their creation. These “'neath texts,” as Andrews calls them, will be discussed with each poem, as they become relevant for analysis. Creating works that express or enact his ideas in multiple layers of visible text and code is characteristic of most of his poetry, and even when the programming codes are not available, their use is often foregrounded in the visible text, or the presence of a 'neath text finds its way to the surface, as is the case with *A Pen* (2007).

This section of the chapter will explore Jim Andrews' DHTML poems starting with a brief reading of *Enigma n* to highlight how it works as an application that manages objects, going into more detail with *Seattle Drift* focusing on how the behavioral typology can be useful in examining the text, and going into more textual analyses with “Spastext,” one of his *Stir Fry Texts* that is built upon expressions of his poetics.

A Meaningful Enigma

Enigma n, Jim Andrews' second DHTML poem, written in 1998 and released with a companion essay titled “Infoanimism,” is a series of operations one can run on a single piece of text: the word “meaning.” After an epigraph by Phyllis Webb[70] ("The world is round./ It moves in circles."), the poem consists of the word “meaning” (in green) placed in the center of a black screen, and the words “prod,” “stir,” and “tame” (in gray) on the top left corner right after the title of the poem, which is also in green. As one clicks on the options in the menu bar (which is the function of the gray words), the text is set in motion, stopped, and changes its appearance, while new options appear after exploring the options offered. Once the options “About” and “Run Away” appear in the menu bar, the reader has the opportunity to end the poem and go either to the essay “Infoanimism” or to "ADVEXP: xes" a work by Ted Warnell.

The version I am describing is the most recent one: the one updated in 2004 by Marco Niemi and Jim Andrews to be more compatible with W3C standards and therefore with more browsers. When written and published in 1998 there were two distinct versions: one for Netscape Navigator 4 and one for Internet Explorer 4, and the gatekeeper was a page titled “meaning.html” that detected which browser one was using and sent one to the appropriate link. If one was using Safari, Opera or some other browser, one would get an error message and wouldn't be allowed to read the

[70] Phyllis Webb is a Canadian poet. I took a first year Creative Writing course from her in the early 80's (Andrews, [comments on] Chapter 3.2: DHTML Dances).

poem.[71] The Internet Explorer version was basically the same as the current version, even though it ran a little differently in terms of animation speed and other minor ways. The Netscape Navigator version was different in several ways:

- Its menu text reads slightly differently: “Prod meaning Stir meaning Tame meaning” instead of the single word “Prod Stir Tame” in the Internet Explorer version.
- It lacks the options “1/0”, “Color”, and “Discombobulate” which are present (and supported) in the Internet Explorer version.
- It had a “drag” function, which allowed the reader to drag the letters of the text. This was not present (or supported?) by the Internet Explorer version and has been discontinued in the current version.

The 2004 code revision collapsed the dual versions to include a single version that works on most browsers. An interesting detail is that this version's source code alludes to a disabled (“detracted from focus” in Andrews' words) “follow function,” in which the letters would follow the pointer in the window.[72] This along with the drag function shows a conscious decision on Jim Andrews' part to focus the texts' responsiveness (and therefore the reader's interaction) on the toolbar on top of the window, making it more like an application and less like a game. It also makes the text more like a toy that follows instructions rather than one that can be manipulated directly or that responds directly to the reader's symbolic “presence” on the canvas of the text. In other words, we can make the letters dance for us, give instructions for

[71] The message read “Your browser does not seem to understand DHTML. Netscape 4 or IE 4 should work if you have a PC. I'm not sure which Mac browsers support DHTML.”

[72] Source code of *Enigma n*.

them do dance differently or stop dancing, we can even take our pointer and join them in the dance, but they won't dance with us.

If we were to apply this statement to the meaning of the words “meaning” and “enigma” perhaps this poem is talking about the ways we interact with the language we read, whether on the page or screen. The meaning of something may be enigmatic, and we can turn words around, play with them, put them in motion or stop them, change their appearance in diverse ways, but we cannot directly manipulate them: they will always be just beyond our grasp, always something of an enigma. I will explore this idea through a detailed description of e-poem as a first generation electronic object.

The paradigm that inspired Jim Andrews to write this “online philosophical poetry toy for poets and philosophers from the age of four up”[73] is that of the document functioning as an application, a point he discusses in the companion essay “Infoanimism.” According to Andrews, this paradigm draws attention to the 'neath text, because it allows us to “conceive of an onscreen word as being an object with its own properties and behaviors.”[74] These properties and behaviors can usually be observed and determined from the displayed text, but given the potential for different interpretations of the programming codes by different browsers and different platforms, particularly when dealing with client side programming, it is useful to look under the hood (so to speak) to see how it is put together.

[73] Andrews, Jim. “Enigma n” *Vispo.com.* URL: http://www.vispo.com/animisms/enigman/index.htm. Accessed 28 March 2008.

[74] Andrews, Jim. “Infoanimism” *Vispo.com.* URL: http://www.vispo.com/animisms/enigman/EnigmanInfoanimism.html. Accessed 28 March 2008.

“Enigma n” is a perfect example of this. I had read “Enigma n” many times with Firefox (my browser of choice), and played abundantly with its options. When I started examining the source code, I realized that I was missing a component of the text, so I opened the document in Internet Explorer and saw what I had been missing: mousing over different words or parts of the document displays additional texts in the status bar. Here is a breakdown of the words/objects that trigger the display of these 'neath texts, the conditions in which they appear, and what effect they have upon the central word in the text (“meaning”) when clicked on:

Object responsive to mouseover and/or conditions for display:	**Text displayed in status bar:**	**Behavior of “meaning” upon mouse click**
blank spaces and the word “meaning,” when starting the poem.	“by Jim Andrews, 1998”	None
blank spaces and the word “meaning,” after mouseover of any of the words in the toolbar.	“It is the world that you love, after all, is it not?”	None
“Enigma n” (in green letters on menu bar)	“Enigma n home.”	None. Link returns reader to introductory page.
“Prod” (gray, green while activated-- applies to all subsequent words)	“Meaning prod. Prod meaning.”	Each letter begins to rotate on a different axis, direction, radius, and speed.
“Stir”	“Stir meaning. Meaning stir.”	“the letters have a common center [for rotation]”[75]
“Tame”	“Repeated tamings collapse meaning within itself.	“the letters have a common point of intersection”
during mouseover immediately after mouse click over gray words in menu bar, until you mouse away.	“Meaning is yours to discover and create”	None
“Spell” (appears after first 3 gray words are clicked on)	Spell meaning out. Spell for literalists.	Spells out “meaning” in the center of screen.

[75] Source code of “Enigma n”

“0/1” (appears after “spell” and another gray word are clicked on)	“Freeze/thaw meaning.”	Stops and starts motion and size change of letters (in case of “discombobulate.” Does not affect color change.
“Colour” (appears after “0/1” is clicked twice in a row)	“Colour meaning.”	Changes letter colors.
“Discombobulate” (appears after “colour” is clicked)	“Resize meaning.”	Changes discombobulate mode: 1. “no font size change occurs” 2. “the font size changes the same for all letters” 3. “the letters change in size differently”
“Speed” (appears several menu clicks after “Discombobulate”)	“Adjust speed of meaning.”	Provides chart on left hand side of screen with 30 speed settings.
“About” (appears, along with “Run away,” after speed has been adjusted)	“About meaning.”	Links to “About Enigma n” page.
“Run away”	“Dialog with Ted Warnell”	Links to “ADVEXP: xes” by Ted Warnell.

Table 3: Responsive Objects in "Enigma n"

As must be obvious by now, this document treats all of its elements displayed as objects, each of which has potentially variable characteristics, behaviors, and conditions under which these characteristics and behaviors can change. The document “enigman.htm” is created and operates under the visual metaphor of the application, by creating its own menu bar and employing the browser's status bar. Each of the gray words in the menu bar (with the exception of “About” and “Run Away”) somehow affects the appearance and behavior of the letters of the word “meaning” in the center of the screen. I specify letters rather than word because each one is an object that can be affected differently with each mouse click, as is the case with option 3 in discombobulate. The variables affected for each letter are the following:

size, color, movement, rotation (further diversified by variables for axis, direction, radius, speed, and overlap). With clear labels for providing input to some of these variables, Jim Andrews keeps the tone playful rather than intimidating, framing the work as “a poetic and philosophical toy for ages 4 and up.”

So how is *Enigma n* poetic and philosophical? The poetic approach is clearly Letterist, atomizing the word “meaning” into its component parts, letters, charging each one with individual color, sizes and behaviors, and giving the reader control to set them in motion. The title “Enigma n” is an anagram of “meaning” and between the title and the word, he raises the question that the e-poem is designed to provide some answers to. What is the meaning of a word, when its letters are scrambled, spinning on different axes and directions, changing colors and sizes? Do those letters mean the same thing, or do new meanings emerge from these motions? Is the meaning an enigma, with n number of possible interpretations? All these questions are the kinds of questions that Letterism wants us to ask of language.

Another poetic tradition that informs *Enigma n* is Concrete poetry, particularly if one uses the “1/0” option to pause the twirling letters to form a constellation of letters, in the style of Eugen Gomringer, for instance. (see Figure 12).

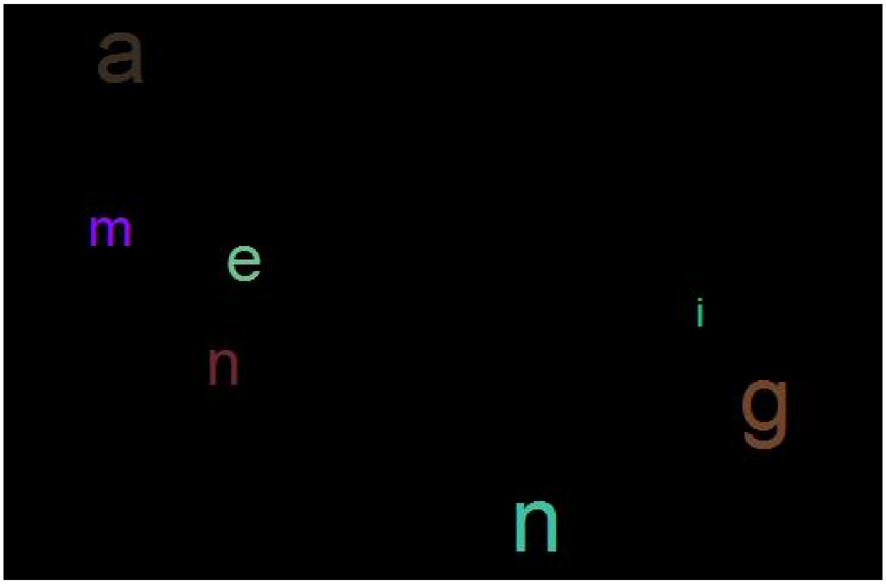


Figure 12: Enigma n stopped after being discombobulated and prodded

For a reader to read “meaning” here she would have to break several “rules” of reading, particularly those that deal with sequence. But some words do emerge without too much effort: “amen” and either “nig” or the reverse, “gin,” which might make more sense. How does this lead us into insight or discovery? Is this an intended combination, attributable to Jim Andrews? The answer to that last question is both yes and no. Jim Andrews created a machine made of words with the potential to arrange the component letters of “meaning” into many permutations, and he did it with that purpose, as suggested by the anagrammatic title. However, by granting the readers control over when to read-- whether it's during full speed animation, a slowed down version, or a completely paused still image as shown above-- Jim Andrews has placed responsibility over the interpretation of the text on the reader. If something doesn't make sense, they can stir, prod, or carry out any number of operations before pausing and reading again, and again, until they find an arrangement that seems readable and therefore meaningful. In a way, this resembles a Rorschach blot test, in which whatever the viewer sees in the random ink blots, is more of an indicator of their own state of mind than of any inherent meaning in the random marks themselves. And that is the point of this poem: meaning is an enigma, to be resolved by the reader through manipulation and play with the linguistic signs of the word "meaning."

By creating the letters as objects, imbuing them with behavior and variable physical characteristics, and creating a playful interface for the readers to manipulate them, Jim Andrews has said what he needed to say about the meaning of language in an electronic, programmable environment: what the letters of a word may spell out in any given moment is shaped as much by the way they "hang out together" as by what the reader needs or wants them to spell.

> The great British poet W.H. Auden once said that he would give less chance of success to a young writer who said he had something to say than he would to a writer who said that he liked to watch the way words hang around together. DHTML allows writers to make documents in which words hang around together and interact with each other and with the reader and possibly with other documents and readers on the Web in ways that can be relevant to what Auden said but in radically different ways than he had in mind (Andrews, "Infoanimism").

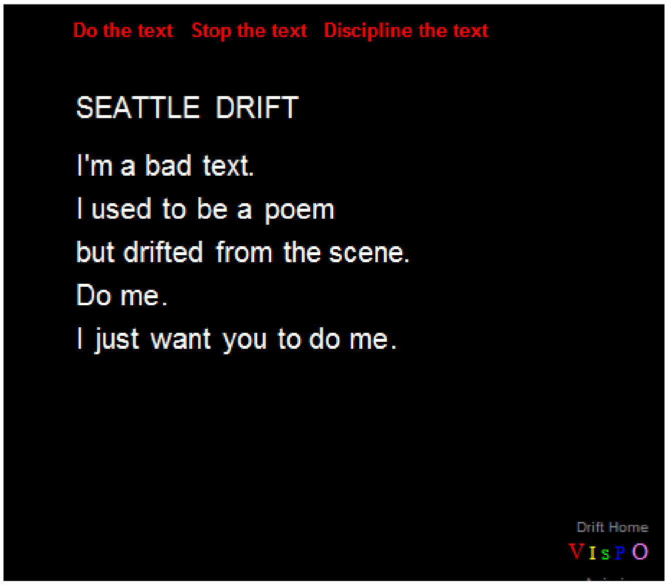
Enigma n is designed to lead letters to hang out together in different ways--lined up in unchanging sequences readable from left to right to form clear words, splayed on the surface of the screen for us to try to combine them into words, or spinning and changing colors and sizes in ways that are practically unreadable as words. It is up to the reader to make sense of these letters, or even transcend the need for letters to mean something, and simply take pleasure in their playful dance.[76]

> Drifting from the Scene
>
> This piece was written, in part, in response to the questions about text and poetry that the new medium prompts in me. I wanted the text of *Seattle Drift* to initially look like plain ordinary text so that the contrast is more apparent. The questions that I have about text and poetry prompted by the new medium are similar in fashion to those prompted by abstract art about art and representation. Both prompt, rather than raise the questions directly (Andrews, "*Seattle Drift* source code").

Seattle Drift (1997) is the most popular poem from this DHTML period in Andrews' career, and an important expression of his poetics. It was published at a time when e-poetry was starting to grow as a Web-based art form, yet it doesn't seem as dated as other contemporary e-texts. Its simplicity of design and directness of its conceit are deceptively simple: this e-poem enacts a critique of current and historical poetry scenes in order to create a space for a new e-poetry scene. This is also a piece that speaks well to contemporary audiences, particularly when being introduced to the concept of electronic poetry, because it takes as a starting point a traditional notion of poetry (verse) and leads the reader into the new scene of electronic poetry, with a concrete poetry transition to smooth the way. This is why I consider *Seattle Drift* to

[76] "One of the things the piece evokes for me is the notion of an engine. Each word rotates in a circle, and the different circles all have some relation. Like gears/cogs in a machine. When you 'tame' it, the circles have a common point of intersection; when you 'stir', the circles have a common center; when you prod, they don't have a common property" (Andrews, "[comments on] Chapter 3.2: DHTML Dances").

be a classic, in addition to its merits as a poem, which should become evident in the following reading.



SEATTLE DRIFT

I'm a bad text.
I used to be a poem
but drifted from the scene.
Do me.
I just want you to do me.

Figure 13: Screen Image of *Seattle Drift* [77]

At a glance, this e-poem looks very much like a contemporary free verse poem: divided into unrhymed, unmetered lines. Its linguistic text is very straightforward, and seems clear, uncomplicated. A central poetic device at work here is personification of the text itself, which is the voice of the poem. This self-aware voice of this poem has found expression in other works by Andrews written during this period, such as the "Pop-up Poems." The self-referential tone draws attention to the poem's form, leading the reader to look at the words rather than just reading them, an important aspect of Andrews' visual poetics. The only elements that would be out of place in a print poem are the red words found above the poem's title, which are options for the reader to activate with a mouse click, evoking the familiar computer

[77] Window size has been reduced to emphasize linguistic text.

interface of the menu bar. The presence of these input cues and the personified text's request to the reader are the keys to this poem's conceit and to its behaviors.

> Stylistically, the piece is similar to the pop-up poems (though not in behavior) in that the text talks about itself. I like this approach because it focuses attention on the questions and also allows me to develop character. The character is the text itself, and the character commenting on its own nature and behavior, though embodying that nature and behavior also, beyond it but within it, like the rest of us. (Andrews, "*Seattle Drift* source code")[78]

The speaker of *Seattle Drift* characterizes itself as "a bad text" that "used to be a poem." It has apparently ceased to be a poem because it "drifted from the scene," yet what we see is clearly a poem. Why would it make such a contradictory statement? Instead of providing reasons or further explanation, the speaker shifts to a request: to "do" it, reiterating that it is all it wants. Perhaps once the reader "does" the text, such explanations will no longer be necessary. Perhaps the text itself derives some pleasure from being "done." In this context the three phrases in red gain more meaning: "do the text," "stop the text," and "discipline the text." The connotations of the words "do me" and "discipline the text" combined with the texts' statement that it is "bad" create a slightly sadomasochistic scenario, involving the reader in the punishment/pleasure of the text. This is emphasized when the pointer becomes a hand when it is placed over the red phrases. If the cursor is the symbolic presence of the reader in the electronic texts, is the extended index finger in that hand a representation of the reader's hand, about to touch the text? Symbolism aside, this is

[78] "To expand on that a bit, behaviors allow work not simply to talk about something but to be it" (Andrews, [comments on] Chapter 3.2: DHTML Dances)

an indication that the phrase is a hotspot, or to use less suggestive language, an input cue.

Initially, the only input cue that produces a response is the first one, “do the text” because the other ones depend upon this one to have a noticeable effect. Upon activating it with a mouse click, the words begin to drift across the screen in a random pattern that slowly leads the words to the edges of the screen and out of sight. Once the text is in motion, one can “stop the text” or one can “discipline the text,” which brings it back to the original formatting and stops all motion. The responsive behavior of the text is very simple, requiring minimal input, yet important because it triggers three other textual behaviors: kinesis, mutability, and scheduling. The text is static without the reader setting it in motion by “doing the text.” The responsive element needs to be activated by the reader for him or her to be able to read the e-poem properly, or at least fully. The mutability becomes clear upon subsequent “disciplining” and re-“doing” of the text, because the drifting is subtly different every time. The use of time (or scheduling) is linear and infinite: the text can potentially drift for as long as the browser and computer are turned on, which raises some interesting questions about the limits of this scene.

The poem's motion, key words, and spaces are essential to unlock this e-poem. It may not be surprising that the words drift across the screen when activated, given the poem’s title. And yet, this movement brings attention to the use of the word “drift” in the title and the text of the poem. One of the primary uses of the word is to describe the motion of the words-- random, uncontrolled, slow-- but it can also mean moving away, perhaps without a clear direction. The poem states that it used to be a poem “but drifted from the scene,” establishing a causal connection between its drifting and it no longer being a poem. The word “scene,” can be read to represent a theatrical stage. The space in the window that one can see without needing to scroll, and the words of the poem are like theater performers, who eventually exit the stage after a performance (or do they drift out of the scene). We cannot read the words we can't see, though we can chase them with the scroll bars. Another meaning of scene is that of poetry scenes, that is performance spaces, traditions, audiences: the cultural milieu in which poetry is written, published, performed, and received. The poem's strategies (including rhetoric, behaviors, and potential readings) stem from the latter meaning of scene and will inform this analysis.

What we see at the beginning of *Seattle Drift* is easily identifiable as a poem, at least in its formatting. It is written in free verse that reads somewhat like a William Carlos Williams, Marianne Moore, or a Charles Bernstein poem, in the sense that it is prosaic in its use of language, unadorned by figurative language, figures of speech, rhyme, meter, rhythm, or other devices that we would readily identify as poetic, with the exception of the personification of the poem itself. The line breaks coincide with the ends of sentences or independent clauses, so the lines are all closed, that is, each contains a complete idea. So, while it is not a very “poemy poem,” as Jim Andrews describes more traditional poems, we can identify it as a poem primarily because it is cut into lines, which makes it verse. We can say it belongs to the general scene of poetry, focusing it further as belonging to a poetic scene associated with Language Poetry, which often subverts the conventions of poetry by using prosaic language that has been cut into lines.

To “do the text” is to allow the poem to drift away from that scene of poetry that we can define as verse, or at least prose cut into verse. As the drifting begins, the words lose their line formatting, as they drift in different directions, at times over each other, yet maintaining their horizontal alignment, so they are still readable. The original linguistic text breaks down as new ones emerge from the newly reconfigured words, something that goes on continuously until the reader clicks on “stop the text.” At that point the text becomes static, its linguistic text stabilized, and the reader can read the newly combined text . . . or should I say poem? A screen capture after a few seconds of drift should provide some insight:

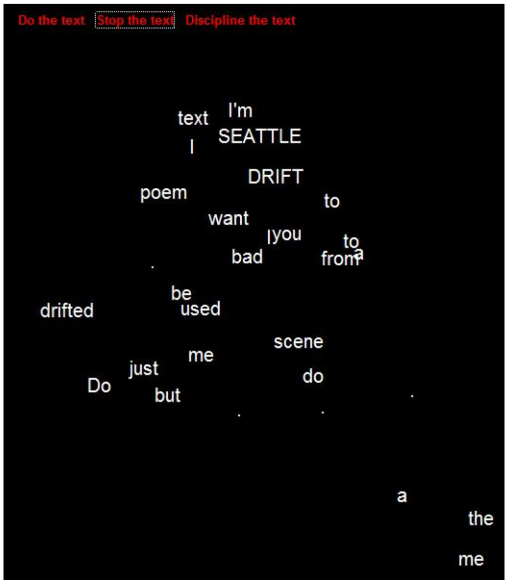


Figure 14: Seattle Drift

Is this text still a poem? Certainly. It looks like a poem by Stephane Mallarmé, Charles Olson, Susan Howe, Eugen Gomringer or any poet belonging to the Concrete Poetry scene—and these are just a few examples of poetic traditions that use spatial arrangement in ways beyond the traditional formatting of verse. We can give this new linguistic text even more attention we gave to the original one and get insightful material to inform our interpretation, because it has become less familiar and perhaps more poetic language. For instance, the opening lines can be read as “text I’m SEATTLE I Drift” after which the reader must make decisions on what direction to read and how to cluster words in ways that make sense (or produce pleasurable nonsense). This drifted text retains its personified sense of self, as seen in phrases like “I DRIFT,” “Poem want I you bad” “be used” and “do a the me.” It also retains some of the language of desire to be “done,” that is, to be cut loose from what ties it to specific scenes and drift away into other poetry scenes.

And yet the personified text claims that it is no longer a poem, because it “drifted from the scene.” Perhaps it drifted from the traditionally-formatted-poetry scene to join the spatially-composed-poetry scene, but will it drift out of that too? Given enough time, all the words will drift from the screen, leaving only the initial input cues. At what point does it stop being a poem? At what point does it cease to be a poem, but is it still a text? When there is only one word? If we consider “the scene” to mean poetry traditions and/or venues, then the poem is questioning the scene itself, and the personified text is being rebellious. If we read “the scene” as a theatrical metaphor, then the text is being self-conscious about its materiality, and refers to its literal departure from the figurative stage of the screen. Does an empty page mean there is no text in it? Certainly.

Does the same apply to an empty screen? Not in the case of *Seattle Drift*, because as long as the program is running, the text is drifting, and it can drift indefinitely because the animation operates on a linear and endless schedule. The farther the words drift from their point of origin, the larger the electronic space of the poem gets. And since the browser window has horizontal and vertical scroll bars that allow one to move the window to wherever there may be text to read, one could literally allow the poem to drift to a truly monstrous size, rendering it horrendously meaningless, “massively insolent” to use Andrews' words. Or perhaps, after years of running continuously under ideal conditions, programmer-critics would have to create new tools to navigate the enormous black spaces of the poem in order to find words to read, becoming astronomers of this language constellation of a size to give Eugen Gomringer nightmares.

> The constellation is the simplest possible kind of configuration in poetry which has for its basic unit the word, it encloses a group of words as if it were drawing stars together to form a cluster.
>
> The constellation is an arrangement, and at the same time a play-area of fixed dimensions.
>
> The constellation is ordered by the poet. He determines the play-area, the field or force and suggests its possibilities. the reader, the new reader, grasps the idea of play, and joins in.
>
> In the constellation something is brought into the world. It is a reality in itself and not a poem about something or other. The constellation is an invitation (Gomringer).

When Eugen Gomringer wrote this essay in 1954, the dimensions of a “play area” had to be fixed, whether he has working with the page or a landscape, as British Concrete poet Ian Hamilton Finlay became famous for. This notion is echoed by Charles Olson in his manifesto “Projective Verse” when he describes a poetics of “composition by field.” In all of these cases, these poets are taking advantage of the space they have at their disposal, which pretty much meant the page for all of them. Jim Andrews also takes advantage of the space available to him, but in a potentially more extreme way because the electronic page (or landscape) are mathematically infinite, and limited only by a computer's storage capacity and processing capability, than by any actual physical dimensions.

It is at this point, when we start to engage the poem's materiality as an electronic object that the poem has reached its true scene: the e-poetry scene. This is a scene of possibilities and experimentation, of play and invitations for readers to embark on a journey that challenges their traditional training as readers, their assumptions about language and how it should be configured, their preconceptions about what poetry is and should be. The e-poetry scene in 1997 was even smaller than it is today, yet there was an enthusiastic energy that came from being among the first to boldly go where no poet has gone before, even if some of those paths had been blazed conceptually in other media. How far readers would be willing to follow on this path is secondary to the need to explore it, and the invitation is always there.

Keeping the reader always in mind, however, Jim Andrews does provide the option to “discipline the text” back into “the scene”—back into traditional poetic form and into the metaphorical stage readers are accustomed to. But this act runs counter to the poem's desire to drift and be in its own scene, to explore cyberspace (or at least disk-space) beyond our ability to follow and be free from tradition, convention, interpretation. And yet is an obedient text: it responds faithfully to the three commands we are given as options. Any controls on the text come from the author, who created it (along with its programmed behavior), and the reader, who has the power set it in motion, stop it, and “discipline” it at his or her own convenience.

Let’s not get lost in the personification, however. The words have no control over their motion: they drift as Jim Andrews programmed them to. The programming is based on randomly generated numbers and a fairly simple positional equation that sends them slowly moving towards the edges of the screen and beyond. The random element is important because Jim Andrews has relinquished some control over the text over to the computer’s ability to generate random numbers, something that occurs during the production performance on the reader’s side. This assures that the text the readers are exposed to will always be different, once they “do the text,” but within certain parameters, which Andrews describes as follows:

> The motion is "pseudo-random." Note, however, that when you play the piece, the words "text" and "poetry" are almost always the last words on the screen. This is because the number of pixels each word moves (every frame of the animation), while random, is within a certain range of random values, and different words have different random ranges. The ranges of the words "text" and "poetry" are smaller than the other words so, on average, they move less than the other words (Andrews, “[comments on] Chapter 3.2: DHTML Dances”).

This built in mutability reinforces the use of the word "drift," referring to the text's lack of control over its motion, or ability to retain a single linguistic text.

Thus the literal drift is the agent for metaphorical and semantic drift of this e-poem. This deceptively simple conceit may initially seem gimmicky, yet it is essential to the poem because it reconfigures the meanings of key words. Drift is also an important aspect in Andrews' poetics, finding literal expression in many of this works, such as *Arteroids*, *A Pen*, and conceptual expression in works like *dbCinema*. To drift is to move without controlling one's direction, surrendering to whichever forces have influence over one's movement, such as wind and current in the case of a boat that isn't using its engines, sails, oars, or rudder to guide it. The geological term "continental drift" also comes to mind, particularly in the context of Seattle and the Pacific Northwest region, since its tall mountains and volcanoes are the result of an ocean tectonic plate drifting against the North American continental plate. A telling notion of drifting is the Situationist practice of the *dérive*, which directly translated means "drift," but is described by them as:

> Dérives involve playful-constructive behavior and awareness of psychogeographical effects, and are thus quite different from the classic notions of journey or stroll.
>
> In a dérive one or more persons during a certain period drop their relations, their work and leisure activities, and all their other usual motives for movement and action, and let themselves be drawn by the attractions of the terrain and the encounters they find there. Chance is a less important factor in

> this activity than one might think: from a dérive point of view, cities have psychogeographical contours, with constant currents, fixed points and vortexes that strongly discourage entry into or exit from certain zones.
>
> But the dérive includes both this letting-go and its necessary contradiction: the domination of psychogeographical variations by the knowledge and calculation of their possibilities. In this latter regard, ecological science, despite the narrow social space to which it limits itself, provides psychogeography with abundant data (Debord).

This notion of the dérive can illuminate Andrews' use of drift in this poem and in his other works, whether he was aware of this theory or not, because it points out several aspects of drifting that are in tune with his own use of it. For instance, both *Enigma n* and *Seattle Drift* showcase Andrews' "playful-constructive" side of giving up control in order to discover insights, through the use of mutable and responsive textual behaviors. The random aspects of the texts are not without parameters—his texts, much like cities, contain "psychogeographical contours, with constant currents, fixed points" and other means by which both the reader and the text can be guided in their drift. *Seattle Drift* has several such parameters: from the menu bar on the top left corner of the screen (seducing its readers to activate, stop, and reset the text), to the Vispo logo marking the lower right hand edge of the original window (framing the psychogeographical "scene" in which the poem begins), to the programming currents that guide the letters as they drift in their pseudo-random directions away from their original position. Andrews carefully crafts the ecology of his spaces, designing the "psychogeography" of his interfaces, so that he can then "let go" of his control over both langwidgets and readers and let them be drawn to what they will.

Understanding the role of textual behaviors in *Seattle Drift* —stasis, responsiveness, kinesis, mutability and scheduling—allows us to interpret the poem more completely than would be possible through a purely textual analysis, because each one helps shape the reader's experience. While I believe that textual analysis is at the heart of any good interpretation of a literary work, I am convinced that it is insufficient for the interpretation of an e-poem such as this, even though it is still a primary tool. The main reason is that since all the possible variations are practically impossible to be read, partly due to the mutability of the drift, partly due to the difficulty of starting and stopping the words at exactly the same time. A close reading of the text requires access to a relatively stable text, with the possibility of rereading it. *Seattle Drift* offers no such possibility, aside from the initial text.

Perhaps in drifting away from the scene of traditional poetry, this text has also drifted away from the scene of traditional literary criticism, and requires a new approach to supplement or even replace the critical tools of the past. This study seeks to expand and develop this emergent critical scene of New Media Studies. The next section will offer a reading of another set of texts that have built-in instability, the *Stir Fry Texts*.

> Cut Up, Heat, Stir
>
> The stir frys dawned on me when I realized it's easy to create an object, fill the object with whatever content you want (text, images, sounds, etc) and subsequently change the content however you want. The changes to objects

> can be triggered by mousing or clicking objects or by some other form of interactivity. This leads to an unusual view of a piece of literary Web art which I want to develop here--not just concerning the stir frys, but more widely for the future (Andrews, "Architecture and the Literary").

The *Stir Fry Texts*, while inspired by William S. Burroughs and Bryon Gysin's use of the cut-up, have a flavor of their own. . . a fact not lost on Jim Andrews when naming them after a culinary technique well known in the Pacific Northwest region where he was living at the time of their creation. The technique of cutting up texts for rearrangement was employed by Surrealist poets and artists early in the 20th century as a way of freeing the creative mind from rationality and the ego, allowing for the unconscious mind to express itself. Burroughs and Bryon Gysin used the technique not only as a composition method to produce novels, audio tapes, and films, but also as a type of literary criticism, since they believed that the cut-up could open up texts to their true or latent meanings.

These Stir Fry Texts aren't quite like what Burroughs and Gysin did, however. They have a style of their own. Sure they are cutting and rearranging texts, but that is where the differences between print and electronic textuality come into play. Burroughs cut up texts and rearranged them into a trilogy of novels, but the book that a reader gets is a completed process: its production performance is over and they can begin their reception performance. The *Stir Fry Texts*' production performance isn't over until the reader begins his/her reception performance: and the two blend in an ongoing process.

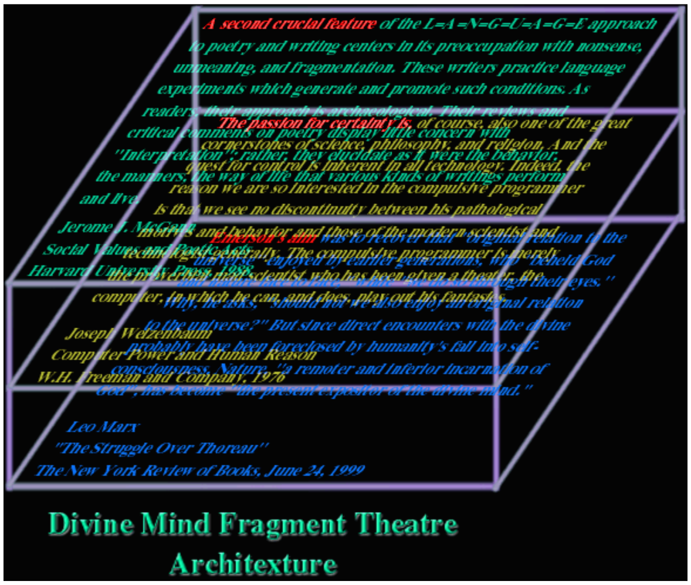
What's new in the stir frys is the spastic interactivity they give to the reader/viewer, the way that they insist on hanging together as texts, physically, anyway and, if they are successful as texts, rather than simply as langwidgetical text toys, the range of insights they afford into themselves and the random and the cut up and the Web and into oneself, since the stir frys allow you to make your own texts (Andrews, “Stir Frys and Cut Ups”).

Jim Andrews did his share of slashing of audio tapes during his years as a radio show host, and with this series of collaborative e-poems, he returns to this technique, using programming instead of a razor blade. His ingredients include e-mails, quotes, concrete poetry, and essay like writing, and lovingly sliced with JavaScript and DHTML tools. The computer provides the energy to run the scripts and keep the surface on which the texts reside hot, in a manner of speaking. Jim Andrews and his collaborators prep the texts for stir frying, cutting and linking where they see fit. The reader’s hand, by way of the mouse and its virtual pointer on the screen’s hot surface, stirs the texts, cooking them up into new combinations and possibilities for his/her consumption.

Culinary metaphors aside, the *Stir Fry Texts* can be described as a single-document hypertext, which the reader can navigate in parts or as a whole. Each *Stir Fry Text* contains 4-6 lexias (textual units), accessible in their uncut form by clicking on a colophon at the base of each page. Each lexia is color coded—that is, each lexia is in a different color or shade of gray. Sweeping the pointer over the text, produces a new text composed of different colored segments from each lexia. Jim Andrews describes the *Stir Fry Texts* as follows:

> The basic architecture of the stir frys is storied in layers of text. Like pages one underneath the other, really--like a book. But unlike a book in that each text is cut into the same number of pieces (30 or so), and when you mouse over the text, the phrase you mouse over is swapped for the corresponding phrase of the text below. And this continues, as you mouse over the phrase, until the topmost text reappears when you have reached beyond the bottom layer of texts. Also, you can see the entire, uncut sequence of layers by repeatedly clicking the colophon (Andrews, Architecture and the Literary).

Here is a graphical representation of these layers created by Andrews, using the “Divine Mind Fragment Theater” as an example. Each layer is represented by either green, yellow, or blue and the first segment of text on each layer is colored red (see figure 15) (Andrews, Architexture of Divine Mind Fragment Theatre).



A second crucial feature of the L=A=N=G=U=A=G=E approach to poetry and writing centers in its preoccupation with nonsense, unmeaning, and fragmentation. These writers practice language experiments which generate and promote such conditions. As readers, their approach is archaeological. Their reviews and critical comments on poetry display little concern with "interpretation"; rather, they elucidate as it were the behavior, the manners, the way of life that various kinds of writings perform and live.

Jerome J. McGann
Social Values and Poetic Acts
Harvard University Press, 1988

The passion for certainty is, of course, also one of the great cornerstones of science, philosophy, and religion. And the quest for control is inherent in all technology. Indeed, the reason we are so interested in the compulsive programmer is that we see no discontinuity between his pathological motives and behavior and those of the modern scientist and technologist generally. The compulsive programmer is merely the proverbial mad scientist who has been given a theater, the computer, in which he can, and does, play out his fantasies.

Joseph Weizenbaum
Computer Power and Human Reason
W.H. Freeman and Company, 1976

Emerson's aim was to recover that "original relation to the universe" enjoyed by earlier generations, who "beheld God and nature face to face," while "we [illegible] through their eyes." Why, he asks, "should not we also enjoy an original relation to the universe?" But since direct encounters with the divine probably have been foreclosed by humanity's fall into self-consciousness, Nature, "a remoter and inferior incarnation of God", has become "the present expositor of the divine mind."

Leo Marx
"The Struggle Over Thoreau"
The New York Review of Books, June 24, 1999

Figure 15: Divine Mind Fragment Theatre Architexture

As can be seen from the descriptions and diagrams above, the *Stir Fry Texts* lend themselves to two kinds of readings: a linear, lexia-driven exploration of the ideas presented in each text, and a reader targeted randomized scrambled text composed of the all the lexia. The reader has the choice on how to read the text, and how to approach the "stir frying" of the texts: when to initiate, how much to "stir," when to stop and read, and whether to repeat that or not.

The number of possible texts that can be created by such scramblings is huge, and without the ability to methodically control the textual combinations, the possibility of reading all the possible variations is practically impossible, short of hacking into the code and modifying it to allow for more control. Even so, exploring every possible permutation would be both impossibly time consuming and impractical, given the limited control one has over selection. Consider the number of possible permutations, as estimated by Jim Andrews:

> Stir frys are composed of x distinct texts, and each text is partitioned into y pieces. For those who like to count, you see from the stir frys (requires IE 4+ for the PC) that there are x^y possible permutations of a stir fry, if the parts are distinct (Andrews, Material Combinatorium Supremum).

To take this a step further, consider the point made by Raymond Queneau about his 1961 poem *Cent Mille Milliard de Poèmes*, which uses the technology of the book to intercut the 14 lines of 10 sonnets, to create 10^{14} possible permutations. He calculated that to read the whole text in all its permutations would take considerably more than what is allowed by a human lifetime, even when reading nonstop. Basically, both Queneau and Andrews have created texts that are impossible to read in their entirety. Both poets have two points to make with these impossible texts: these are pieces meant to be incomplete and reading them in their entirety wouldn’t yield sufficient insight to warrant the enormous time investment required.

Why create a work that cannot be read completely? I believe the idea is to emphasize the processes by which the texts are constituted and to encourage readers to take an active role in figuring out its patterns. In an essay titled “Material Combinatorium Supremum,” Jim Andrews discusses how the number of atoms in the universe can be calculated and after considering the technical requirements for creating a stir fry text that equals that number, concludes that it would not be difficult to achieve. The following insight emerges from this conceptual experiment:

> The 'material combinatorium supremum', represented as a text, would certainly give an impression of great variety, and hopefully interest, also, but you would eventually get a sense that you had 'read it' and you would surely be right (Andrews, Material Combinatorium Supremum).

Closure, therefore, comes as a decision from the reader that they “read it” and got what they needed from the text. I imagine the idea is to let the readers decide when they have achieved closure, or they may succumb to what Espen Aarseth calls “aporia” in his book *Cybertext*. The other insight suggests why he hasn’t undertaken the project, with characteristic humor.

> There is, of course, much silliness to such a project. It could end up so horrendously meaningless as to be an abomination unto the combinatorium supremum and an afront to all poetry in the universe. Massively insolent.

> Then again, it could possibly be amusing. And a kind of thing in itself, regardless (Andrews, “Material Combinatorium Supremum”).

The potential for creating absurdly large texts that challenge traditional notions of reading and criticism is very present in digital media, and Jim Andrews is both inspired and amused by it. When he conceptualizes his works, he balances that potential with his desire to engage his readers, having restraint in the scope of his projects while inviting his readers to playfully challenge their habits. We see this in *Seattle Drift* , where he balances the personified desire of the text to drift away (that is, for the text to explore the limitless potential space of the e-poetry scene) with interface controls (for the reader's own desire to read and understand the text). For the *Stir Fry Text*s this balance comes through restraint, as well as by the ability to read whole lexias instead of just scrambled texts. In these works, as in *Seattle Drift*, Andrews provides both the well worn paths that readers are accustomed to, and devices that encourage creative drift through the psychogeographical spaces of his works.

> The stir frys are, in a sense, conceptual pieces. You think about them. In a typical literary work, you read it and think about what it said. The stir frys have that dimension to them, but also you are meant to think of them as an architecture. We look at architecture. We think of buildings as structures. One contemplates the shape of the stir frys. As a combinatorium. As a limited but vast space of possibilities. As a shape, as it were. As a kind of conceptual visual poetry. The visual part is 'on the page' but it is also invisible, in the

> 'shape' of the combinatorial space (Andrews, "[comments on] Chapter 3.2: DHTML Dances").

Andrews' emphasis on architecture and buildings to describe texts as spaces strengthens the analogy with Debord's Theory of the Dérive because readers can either drift and follow the currents of a city, or cut across them with a purpose, but the city remains still: its buildings don't move.[79] Andrews offers his readers both the comfortable familiarity of static text, but also the ability to cut loose and drift by fluidly reconfiguring the words and phrases that form the lexia, undermining the notion of unmovable buildings, while still evoking a sense of space. A close reading of a representative *Stir Fry Text* will support the claims made up to this point.

The *Stir Fry Texts* is a project that consists of the following works, to date:

- "Divine Mind Fragment Theater" (1999) was Jim Andrews' first *Stir Fry Text*. It interconnects quotes by Jerome McGann, Joseph Weizenbaum, and Leo Marx about poetic experimentation, computers, and Emerson's aim to recover the "original relation to the universe."
- In "Spastext" (1999) Jim Andrews intercuts five texts in which he discusses questions of authorship, publishing, and self-publishing on the Web.
- In "Correspondence" (1999) Jim Andrews intertwines excerpts from an e-mail correspondence between himself, Lee Worden, Mary Phillips, and Talan Memmott.
- "Log" (2000) was written by Brian Lennon.

[79] The French word for building is *immeuble,* which literally means unmovable.

- “Blue Hyacinth” (2003) consists of four brief narratives written by Pauline Masurel.
- “Concrete Stir Fry Poems” (2006) are written by Marco Niemi.

Note that this framework has served as inspiration for several poets to create their own work. It is also noteworthy to see that the programming for this framework has been updated by Marco Niemi, who also helped Andrews update the DHTML code to be more compatible with browsers in 2004. The *Stir Fry Texts* are therefore a kind of machine, with built in interfaces, which produces similar results, but that needs texts to process. Jim Andrews has created a machine made of words, that can be emptied and refilled with words, and the engine of this literal, literary machine still operates and inspires others to provide language for it to run with.

Let’s take a close look at “Spastext,” since it is the only one of the *Stir Fry Texts* for which he supplies the language for the framework. As with many of his other poems, the essay-like text of “Spastext” is written in the tradition of some Language Poets like Charles Bernstein, who blur the lines between theory and practice, poetics and poetry. Jim Andrews likes to employ metatextual voices: personified self-aware poems that address the readers directly, as well as the voice of the poet, foregoing the illusion of creating speakers and confronting the readers with issues that relate to the themes of the poem. The latter is the case of “Spastext,” in which he discusses several issues that relate to his work: the nature and economics of authorship, the commodification of poetry and art in contemporary culture, the economics of the Web as a means for publication, and the ability of art to promote compassion and self-reflection.

“Spastext” is composed of five lexia and an opening screen all of which contain a consistent graphical element: a colophon with an image file named “twitch.gif.”[80] Upon opening the text, one sees a black screen with the centered “twitch” icon in the center of a dark screen and three verbal links: one to the title page for “Spastext,” one for the Stir Fry Texts, and one for the main page of Vispo.com. Clicking on the twitch icon, takes us to the first lexia which has its text formatted as a paragraph centered on the window. If you leave the cursor on the same position as when you clicked on the twitch icon, the first lexia will materialize under it, and its position will be somewhere in the center of the last three lines of that paragraph. Unless you have very fast reflexes and are able to move away from that position within the first second or two or you moved immediately after clicking so cursor is away from the position the text loads into, you will have no choice but to begin stir frying the text, so to speak. If you don’t move the cursor, the stir frying won’t begin, but in order to click on the icon and go to the next lexia one must move the cursor over text, activating the “stir frying” part of the program and scrambling the lexias together. This is Jim Andrews’ way of showing us the primary mechanism of the poem in action without having to tell us how to activate it through instructions.

Each lexia in “Spastext” is cut into 29 pieces and each piece is linked with the corresponding piece in the next lexia. The pieces in the fifth lexia links to the first, closing the loop for the rapid cycling from one to the next. Clicking on the twitch icon cycles from one entire lexia to the next, which is an important feature, since it allows the reader to examine each lexia before stir frying them into texts composed of

[80]

their intermingled segments. This allows Andrews to have a rhetorical progression between lexia equivalent to paragraph progression in the page. If read that way, linearly going from lexia to lexia and reading them whole without “stir frying,” one gets the sense that one is reading an essay on the issues mentioned earlier. If “stir fried” then the texts are intermingled and form a text that is more open to interpretation, and more poetic by virtue of being less rhetorical, less familiar in its structure.

A lexia by lexia reading should lead us to an initial understanding of the ideas Andrews is expressing in this piece, which will be compared to close readings of sample “stir fried” texts.

> Who now is the author? Who really cares except the one expecting the cheque in the mail? Let him whine and fret about intellectual property rights. The important thing is not who writes or makes it, but that extraordinary work be done. We own very little, owe those who have gone before very much. Pythagoreans attributed all work to 'himself', Pythagoras, ipse dixit, he said it (apparently he never wrote a thing). So did many of Warhol's friends ('Here's a great idea that nobody has done. Why don't you do it and I'll sign it?') We entered a phase of combinatorial inter-textuality long ago. The Web and anything digital or copyable perpetuates it (Andrews, “Spastext”).[81]

At a glance, this paragraph is a discussion of the politics of authorship , the indebtedness to work that has been done before, the ubiquitousness of intertextuality,

[81] This paragraph was originally part of an essay titled “Digital Langu(im)age -- language and image as objects in a field,” published in *Perihelion* 1:2 in 1998.

and its combinatorial nature. There is a clear rhetorical progression from question to answer to examples, reaching a conclusion from its discussion. So far, this is not a very poemy kind of text—which begs the question: is this a poetic text at all? It is written in prose that would be difficult to label as poetic because it is not highly charged with sonorous qualities, images, figurative language or any other of the staples of the poetic traditions. It seems to be more in tune with the anti-lyrical poetics practiced by Language poets who blur distinctions between theory and poetry, composing poems that read like poetic manifestos. So where does it fit in a discussion of poetry, or more appropriately, how does these texts benefit from being examined as a poem? The answer lies both in the code and in the executed poem, that is, the stir fried text.

When looking at the source code, one can see the places where the prose excerpt has been cut, and more than that, arranged into lines. The idea of using lines is a programming convention, which allows for a simple organization of the programming codes, in this case declaring the value of each variable. In what ways is looking at these lines of code as poetic line breaks useful? The text is displayed as prose, after all.[82] One way in which the notion of the poetic line helps approach this otherwise very prosy text is that it isn't merely prose: each paragraph invisibly is sliced and cut into segments and linked with corresponding segments in other paragraphs. These breaks in the structure of the sentences become very clear when

[82] I think of poetry as intensest engagement with language. So, for instance, I look at the work of Godel the mathematician/logician as a kind of poetry. The stir fry concept has very little to do with story, with fiction, with the prosaic. Instead, the whole concept of it is in the domain of poetry (Andrews, [comments on] Chapter 3.2: DHTML Dances).

textual segments from other paragraphs are interposed and the grammatical structures from different sentences clash and meld to produce a text that could be described as poetic. The deliberateness of a line break is equal to the deliberateness of a cut in a paragraph of Spastext, in the sense that they both cut across phrases and sentences, creating open and closed lines, enjambment, and multiple readings of sentences that would otherwise have simpler and fewer potential interpretations. Let’s take another look at the first paragraph in Spastext, this time as it appears in the source code of the poem, in order to test this assertion.

```
a[0][0] = "Who now is "
a[0][1] = "the author? Who really cares "
a[0][2] = "except the one expecting "
a[0][3] = "the cheque in "
a[0][4] = "the mail? Let him whine  "
a[0][5] = "and fret about intellectual "
a[0][6] = "property rights. The important "
a[0][7] = "thing is not "
a[0][8] = "who writes or makes it, "
a[0][9] = "but that extraordinary work "
a[0][10] = "be done. We "
a[0][11] = "own very little, owe "
a[0][12] = "those who have "
a[0][13] = "gone before "
a[0][14] = "very much. Pythagoreans "
```

```
a[0][15] = "attributed all work to 'himself', Pythagoras, "
a[0][16] = "ipse dixit, he said "
a[0][17] = "it (apparently he never wrote "
a[0][18] = "a thing). "
a[0][19] = "So did many of Warhol's friends ('Here's a "
a[0][20] = "great idea that "
a[0][21] = "nobody has done. Why don't "
a[0][22] = "you do it and "
a[0][23] = "I'll sign it?') We entered "
a[0][24] = "a phase of combinatorial "
a[0][25] = "inter-textuality "
a[0][26] = "long ago. The Web and "
a[0][27] = "anything digital or copyable "
a[0][28] = "perpetuates it. "
```

The breaks have several effects: they tend to isolate phrases from the sentences they are parts of, emphasizing and creating meanings not apparent if left whole. For instance, line a[0][1] reads "the author? Who really cares " which is a parallel construction to line a[0][4] ("the mail? Let him whine"). Both phrases have an equal number of words , beginning with a question and making a comment that can be interpreted as disparaging in tone, undermining the earnestness of the questions. The fact that both sentences end in the middle of a line and have the beginning of another sentence immediately following create a sense of forward motion with the ideas put forth. There are 9 sentences in this paragraph, but only two of them end at the end of a line, and one of them is at the end of the paragraph. Of the 29 lines in the paragraph, only four are closed: the two sentences, and two phrases that end in commas (in lines a[0][8] and a[0][15]). Every other line has strong enjambment, pulling onwards towards the next phrase that will complete its idea. Why is this significant? Because when the texts from different paragraphs become scrambled, the connections between the open lines will be stronger. The pull of these texts is towards coherence, despite the built in incoherence that comes from recombining them. Between the two contradictory forces, arises the kind of tension and play one could come to expect from poetic language.

Taking this analysis a step further, a juxtaposition of the first two paragraphs, broken into lines of code should provide insight on the connections between the lines.[83]

a[0][0] = "Who now is "	a[1][0] = "Writers "
a[0][1] = "the author? Who really cares "	a[1][1] = "must eat too, but why crusade “
a[0][2] = "except the one expecting "	a[1][2] = "for further “
a[0][3] = "the cheque in "	a[1][3] = "perpetuation of “
a[0][4] = "the mail? Let him whine "	a[1][4] = "the idea of art “
a[0][5] = "and fret about intellectual "	a[1][5] = "as commodity, art “
a[0][6] = "property rights. The important "	a[1][6] = "as a product “

[83] For the sake of compression, I will remove the font and color tags to focus attention on the text.

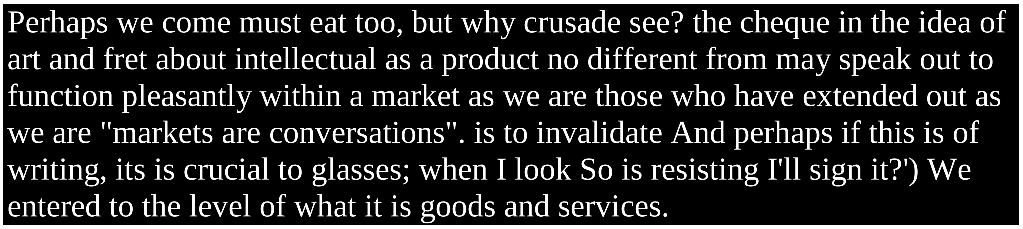
a[0][7] = "thing is not "
a[0][8] = "who writes or makes it, "
a[0][9] = "but that extraordinary work "
a[0][10] = "be done. We "
a[0][11] = "own very little, owe "
a[0][12] = "those who have "
a[0][13] = "gone before "
a[0][14] = "very much. Pythagoreans "
a[0][15] = "attributed all work to 'himself', Pythagoras, "
a[0][16] = "ipse dixit, he said "
a[0][17] = "it (apparently he never wrote "
a[0][18] = "a thing). "
a[0][19] = "So did many of Warhol's friends ('Here's a "
a[0][20] = "great idea that "
a[0][21] = "nobody has done. Why don't "
a[0][22] = "you do it and "
a[0][23] = "I'll sign it?') We entered "
a[0][24] = "a phase of combinatorial "
a[0][25] = "inter-textuality "
a[0][26] = "long ago. The Web and "
a[0][27] = "anything digital or copyable "
	a[1][7] = "no different from “
	a[1][8] = "other things produced “
	a[1][9] = "to function pleasantly “
	a[1][10] = "within a market “
	a[1][11] = "society? To crusade “
	a[1][12] = "for the ‘artist’s “
	a[1][13] = "right’ to have work “
	a[1][14] = "treated like “
	a[1][15] = "any other commodity “
	a[1][16] = "is to invalidate “
	a[1][17] = "the subversive force “
	a[1][18] = "of writing, its “
	a[1][19] = "destructive content, its “
	a[1][20] = "truth, and denigrate “
	a[1][21] = "the alien and “
	a[1][22] = "alienating oeuvres of “
	a[1][23] = "intellectual culture “
	a[1][24] = "to the level of “
	a[1][25] = "familiar “
	a[1][26] = "goods “
	a[1][27] = "and “
	a[1][28] = "services. “

a[0][28] = "perpetuates it. "	

Table 4: Paragraph Comparisons in "Spastext"

The second paragraph is structured more or less the same as the first one, exploring ideas in an essay-like language, and cutting them into open segments that are enjambed with the previous and following segments. The ideas progress from questions to examination of the ideas they pose and conclude with a statement of the speaker’s position on these issues. The similarities of subject matter, grammatical structure, and rhetorical progression of ideas strengthen the connections between different paragraphs.

Looking at the other 3 paragraphs, one can see a logical progression in the argument but also a move towards more introspectively poetic language—all of which is encoded to be mixed with previous paragraphs. The result is a combination of different registers and styles, all cut into segments that can be scrambled together with the lightest touch of a mouseover, creating a text that has many possibilities for expression. The best test of this potential is to stir fry some texts and then perform some readings of the resulting text. Here is the text that results from zigzagging across the paragraph from top to bottom, left to right numerous times to have a thorough scramble.



Perhaps we come must eat too, but why crusade see? the cheque in the idea of art and fret about intellectual as a product no different from may speak out to function pleasantly within a market as we are those who have extended out as we are "markets are conversations". is to invalidate And perhaps if this is of writing, its is crucial to glasses; when I look So is resisting I'll sign it?') We entered to the level of what it is goods and services.

This text breaks a number of semantic and syntactic rules of English, yet it makes sense in the way some of Gertrude Stein's poetry does. It is a fragmented set of phrases for which color can serve as indicators of punctuation at times, emphasis at other times, or simply expose the breaks within the text as they appear in the source code, leaving the breaks open to the reader's interpretation. This text has a similar message as the original paragraphs, but it is emphasized, and one could argue that there are truths that emerge from the *Stir Fry Texts* that are implied in the original ones.

For example, in the second sentence, the subject of the sentence, "the cheque in the idea of art" can be read as the commodification of art that goes so deep that it becomes part of the very concept of art. Its continuation "and fret about intellectual as a product no different from" can be read as how intellectuals are seen or treated as products and in some ways are no different from products. While "may speak out to function pleasantly within a market" suggests that that commodification of art may express itself towards a perpetuation of a placid relation between intellectuals, artists, and the market.

The statement that follows "as we are those who have extended out as we are" can be read in a number of ways, depending upon how one clusters phrases and allocates pauses within the sentence. The initial "as we are" can be an assertion of the state of "function[ing] pleasantly within a market," echoed by another "as we are" which reasserts "those who have extended out." In other words, it emphasizes that "we" (intellectuals and artists) are at this moment working well within the artistic market and are extended out, concluding with the quote "markets are conversations," suggesting that the conversations of intellectuals extending out within a market or economic system. That statement could be read differently if we do not ascribe any pauses within it. In that case, it would mean that "we" (intellectuals) "are those who have extended out as we are," meaning that they have not changed how they are in order to "extend out" beyond the market and the commodified artistic world. The difference between both readings is whether intellectuals are functioning pleasantly within the market system, or whether they have extended out beyond it. Is this a resolvable ambiguity?

To continue close reading the paragraph would yield equally ambiguous results, partly because the text is very open to readerly intervention and interpretation. Where does a lexical/grammatical unit begin and end? Where does one break the text into phrases and sentences? How does one make sense of the work? Perhaps a good approach is a kind of divination, in which a reader searches for moments of lucidity, and insightful permutations. Perhaps the pleasure of Spastext is in the surprising insights it offers with every recombination that reveal the truths that lie beneath and between the more deliberately planned statements in the unscrambled texts. After all, the last sentence in the excerpt quoted above is "We entered to the level of *what it is* goods and services" (emphasis added, though present already through a color shift), which establishes that the level entered by the intellectuals is a true one that simplifies everything they produce to "goods and services."

What happens to these "realities" or "truths" that emerge from the texts when stir fried further? Is there any consistency to the statements? Do they really cohere? This can be determined through further stir frying and analysis of the resulting texts.

> Picture buy and sell see? over the Web the mail? Let him whine with books and pictures as a product insofar as we may speak out of the pain be done. We as we are rent asunder, gone before as we are any other commodity ipse dixit, he said And perhaps if this is of writing, its is crucial to great idea that the alien and you do it and all that crap, to the level of what it is goods and perpetuates it.

This text returns to the ideas of buying and selling, but this time emphasizing media ("the Web" and "the mail"),and bringing in some sense of who is speaking "out of the pain" of being "rent asunder" for the creative act, yet being treated as "any other commodity." A rhetorical move that is perpetuated and emphasized in this stir fried text is to contradict and undermine the statements made, putting forth a sense of uncertainty. For instance, the phrase "its is crucial to great idea" emphasizes the importance of the idea "that the alien and you do it" (writing), but then undermines it by saying "and all that crap," returning to the statement of writing as marketable "goods." The ideas are expressed recursively yet emphases fall on different topics, as some phrases become intelligible and others less so.

Perhaps the magical number three should guide this exploration of Spastext with another scramble of the texts. This should confirm the hypothesis of emphasis and elaboration that results from this textual recombination.

> Writers yourself. What do you art and ideas the cheque in you see but within as a product insofar as we may speak out of the pain be done. We own very little, owe rent asunder, extended out treated like any other commodity to say what we are become. it (apparently he never wrote a thing). destructive content, its great idea that the alien and alienating oeuvres of all that crap, to the level of inter-textuality to be one and have and hold.

Here the text zooms in on the reader "yourself" and questions him/her directly "what do you" (or what you do). "Art and ideas, the cheque in [them]" returns to a main and strongly recurring theme: the commodification of art, ideas, the author, intellectuals, and readers, all of which are “treated like any other commodity,” “as a product.” Other ideas that were in the original texts (or scriptons) that find emphasis in this stir fry text (or texton84) are the notions of “alien and alienating oeuvres,” intertextuality, and “to be one and have and hold.” These three quotes taken from the last “sentence” foreground the central tension expressed in Spastext. On the one hand is the egotistical pull of the self, embodied by the concept of the author, who “receives the cheque in the mail,” who reaches out to other people and texts and claim them as their own (as seen in the examples of Pythagoras and Warhol) who seeks “to be one and have and hold.” On the other is the writer’s move outward from the egotistical self: towards society, towards works that can have “destructive content” and be “alien and alienating,” disturbing the comfortable functioning of the marketplace, and acknowledge the intertextuality of “art and ideas.” Let us not forget that the notion of an author is a complex set of legal, economic, and literary concepts, all suggesting that an author has rights, receives compensation, and produces works imbued with intentions. “Spastext” is Jim Andrews self-conscious exploration of his own role as a writer/author and how he balances the complex set of desires, needs, and intentions that making a living from writing implies.[85]

84 Espen Aarseth’s neologisms: “scriptons” (the source texts) and “textons” (the text that results from manipulating the scriptons) are useful to distinguish the two kinds of texts in this work.

[85] I see Spastext, the interactivity, the combinatorial nature of that interactivity, as basically manifesting all the contradictions inherent in the 5 underlying texts. Manifesting and 'dramatizing' those conflicts. The 5 texts are not consistent with one another. They may be consistent within themselves, ie, text 1 may itself be consistent, but text x is at odds with text y. The contradictions are ones we live, though. We realize that commodification of art poses real problems. But we also realize that commodification also is what enables work to function not only in the economic markets but also, to some extent, in the 'market of ideas'. So Spastext manifests and

I conclude this reading of “Spastext” with a look at its title and the keywords Jim Andrews uses to tag this piece in its programming code: “Spastext, Cut Up, Stir Fry Texts, Web art, visual poetry, vispo, Jim Andrews, concrete, alphabet, lettristic, poetics, innovative, innovation, poet, typography, cabalistic, experimental, langu(im)age."[86] The title is a both a pun and a portmanteau of the words “spastic” and “text,” referring to the “spastic interactivity” he talks about in the essay “Stir Frys and Cut Ups.” The keywords point to the poetics that inform his work: concrete, visual, and lettristic poetry, innovation and experimentation, and the interconnection of language and image through his coined langu(im)age. How the visual characteristics of language interact with meaning is central to his poetics, not to mention behavior, sound, and the reader.

Conclusion

This period in Jim Andrews work shows the significant development of the behavioral aspect of his electronic poems, because DHTML made many of the ideas he was exploring earlier possible. It is here where his treatment of language as an object and/or image-- langu(im)age-- gained an added dimension, behavior, to become langwidgets. The three DHTML poems analyzed demonstrate the potential and limitations of the materials he was working with, as well as the expression and development of his poetics. The next stage in his growth as a full time Web artist and poet marks a return to his longtime interest in music, but this time exploring the potential of networked and programmable media at his disposal.

exaggerates or confuses or dramatizes some of the contradictions of our social system and our art systems and our lives (Andrews, [comments on] Chapter 3.2: DHTML Dances).

[86] From the source code of http://www.vispo.com/StirFryTexts/spastext.html.

Since writing these e-poems in DHTML Jim Andrews has moved almost completely to using Director, partly because of the compatibility issues due to the lack of DHTML implementation standards, but particularly because of its limitations for using sound. Jim Andrews' new direction would be to develop interactive audio works such as *Rude Little Song* (1999), and *Oppen Do Down* (2000), leading up to *Nio* (2001).

> Since that time (1999), the 9 intervening years have changed the computing environment. The way browser companies have implemented DHTML is more standardized because of adherance to W3C standards. So it's easier to make DHTML pieces that run on different browsers and different platforms without having to make multiple versions and take out features for some versions. The audio capabilities are still not very well-developed, however (this is also true of Flash). Nor are the visual processing possibilities of DHTML very well-developed. Which is to say that as an animation tool, DHTML is still fairly primitive. But the DOM (document object model) of DHTML is very interesting from a literary point of view because the main paradigm is the document, not the movie. DHTML is still very interesting concerning its ability to create exceedingly funky documents (Andrews, “Re: The notebook arrived”).

The paradigm shift from DHTML (which is the document and its objects) to the cinematic approach of Director resulted in an increased use of scheduling in the electronic poems that followed. The shift to Director as an authoring tool also came with a shift in direction and location. Jim Andrews moved back to Victoria, Canada in

2000, having received a grant from the Senior Canada Council to work on interactive audio-- what he calls vismu (a portmanteau of "visual music"). The beginning of a new stage did not mean that he would abandon his interest in visual poetry and programmed behavior, as can be seen from the highly visual characteristics of his vismu pieces, to be discussed in the next section.

Visual Music

> I called my site Vispo ~ Langu(im)age—language and image—but that was a few years ago. Now there are several other media involved, not least of which is sound (Andrews, "Defib: Randy Adams interviews Jim Andrews about Nio").

The previous sections in this chapter have focused on two aspects of Jim Andrews' development as a visual poet and a writer of electronic poetry, which thrives on the processing power of programmable media. This section traces connections between his aural poetry, from the sound poems he recorded in1989 to the interactive music that he now writes/programs/composes, centering on *Nio* (2001), a piece that seeks to tie in all the elements Andrews explores in his poetic works: visual, kinetic, aural, and 'neath texts (programmed behaviors, particularly mutability and responsiveness).

Jim Andrews first became interested in the material qualities of language during the six years he produced a radio show that was first called "Fine Lines" and later became "?Frame?." This change in titles for his show reflects an important shift in how he approached poetry for the show and in his own practice: "Fine Lines" found its center of gravity in recording oral performances of fairly traditional poetry, while "?Frame?" featured sound poetry and works that took advantage of audio recording and editing technologies to shape the final product beyond what is possible with the human voice alone.

> In my radio show, I started out recording poets reading poems, writers reading stories, and occassionally producing those a bit for radio. Then I heard the audio writers such as Gregory Whitehead, Susan Stone, and Douglas Kahn. I first heard them on the 1985 audio anthology "The Sound of Radio" at http://ubu.com/sound/tellus_11.html . The Whitehead piece "If a Voice Like Then What?" came as a revelation to me. Here was someone doing a type of audio poetry that was new. And media savvy. Smart about the media of recorded sound and radio. I started to realize that understanding media was important to being able to work artistically with media (Andrews, "Gregory Whitehead").

Andrews elaborates on this shift, narrating this formative stage in his life in the following excerpt from an e-mail sent to me in November 2007.

> After I graduated in 1983, I started a literary radio show at CFUV-FM in Victoria Canada. This was where I learned to try to be creative with technology. I produced the show on cassette tape and sent the cassette each week to 15 other campus/community stations across Canada. They aired the show each week. The show was on CFUV-FM, CITR-FM in Vancouver, CFRO-FM in Vancouver, CJSW-FM in Calgary, and as far away as CHMR-FM in St. John's Newfoundland.
>
> I'm mentioning the radio show I produced for six years in the 80's because it was how I first related the literary and electric technology, and it's

> where I first learned to be creative with technology. Also, I came across the work of the 'audio writers'--people such as Gregory Whitehead (http://www.ubu.com/sound/whitehead.html), Helen Thorington (who now produces http://turbulence.org), and Susan Stone. They were doing very interesting literary work better suited to radio than reading a print poem into a microphone or writing a play and putting it to radio. Whitehead and crew were thinking deeply and entertainingly about the medium of radio and recorded sound, also. This was when I started reading Marshall McLuhan and other media theorists. And came to understand that understanding the medium/media one uses is very important because the phenomenology or specific characteristics of the media/um provide some of the expressive power of art in that media/um, and also because if the artist does not understand the media/um and come to 'feel with it', the media/um will often dominate the message, drown out the artist. Media themselves are active at least like the turbulence of air or water (Andrews, “Re: on 'code poetry'”).

Radio is clearly the beginning of Jim Andrews’ awareness of language as something that can be deeply affected by the media it is produced in. Even before encountering Whitehead, Thorington, Stone and others, the experience of learning the tools and techniques required by radio expanded his awareness of its material characteristics. These sound poets were catalysts for his transformation from a poet and literary radio show producer whose paradigm was the written page and the performances it evoked (as evidenced by the show’s title Fine Lines), to the kind of writer and producer who thought of radio and related media as instruments for poesis, renaming his radio show

?Frame? and featuring sound poets and “audio writers.” Whitehead’s work, in particular, broadened Andrews’ horizons by showing him a direction his earlier work had only hinted at.

> If you listen to some of Whitehead's work, you see that he has a rather intense engagement with language. Theatrical. Musical. Theoretical. Literary.
> I saw in his work a kind of intense engagement with language that is more interesting to me, concerning contemporary poetry, than the manner of the poemy poem.
>
> In Whitehead's work, I saw that poetry was moving in directions that weren't even called poetry, weren't regarded as poetry, but, to me, were the strongest directions of contemporary poetry (Andrews, Gregory Whitehead).

Andrews shifted from being a poet who valued primarily the fine line of lyric poetry (or “poemy poems”) to a poet who was not afraid to challenge the poetic conventions learned as an English major at a fairly traditional university, even if it led him in directions that were no longer considered poetry by the status quo. To date, many of his works can be categorized as visual or electronic art, music, computer games, and/or applications rather than as poetry, and Andrews welcomes the questions that arise from readers as he challenges their assumptions of what is poetry. He also began a lifelong creative engagement with the inscription technologies at his disposal. His work with the materials of radio led to a series of sound poems titled *Cassette Radio*

Video Destabilizer and Audio Poems (1989), some of which are available in the Audio page in Vispo.com.[87]

> I worked at the radio station. I wrote grant proposals. Some of them were successful. I managed a couple of project and mostly produced my radio show. It was all analog technology. Cassette decks, reel-to-reel tape decks, mixing boards, a few effects, and a razor blade to cut and splice reel-to-reel tape. That's where I got interested in the cut. The cut can be interpreted in many ways. The wound. The splice. The transition. The joining. The juxtaposition. On and on. The cut is a source of great energy and art in audio work. I started to produce my own audio art. A little bit of it is at the bottom of http://vispo.com/audio . Sound poetry (Andrews, “Re: on 'code poetry'”).

These sound poems, particularly “Poetry Craft,” can be listened to as expressions of Andrews’ rebellion against the traditional apparatus of poetry, critiquing the inflated social value given to lyric poetry and publication, a bubble full of hot air that in the end goes “pop.” He uses a variety of strategies throughout the poem, such as using different enunciations of the same word, placing emphasis on different parts of the word or evoking different registers. For example, the word “poetry” is frequently pronounced as “poretry” in a haughty voice. In the same way “lyric” is super enunciated, from the initial “l” to the hard “k” at the end, at times sounding as distorted as “learik,” creating a counterpoint to the songlike or euphonic connotations of lyric poetry. Some other techniques he uses are repetition, nonsensical verbalizations, and manipulating the sound recording to echo or distort his voice. For

[87] http://vispo.com/audio.

example, when he says in a highly enunciated voice “poretry lovely”, he starts to repeat that last word overlaying it with an echoing sound effect to transform it into “blah blah blah” and further, mocking the highbrow voice’s comments on poetry by boiling them down to the nonsensical. Also does this at the very end of the poem, by shifting from “poetry published” to “pop” (repeated many times), undermining the high cultural value given to publication by evoking low or popular culture.

His interest in sound poetry led Andrews to meet a Seattle poet and artist who became one of his most important influences: Joseph Keppler.

> Also, by that time, through the radio show I'd met Joseph Keppler from Seattle. Joe and I are still friends. He was the first artist I'd met and become friends with who was strongly active in several arts in a literary way. I did a radio show on Joe's work. It's at http://vispo.com/audio . Joe was something of a mentor. He's a bit older than me. And is widely read in poetry, philosophy, visual art, and film. He showed me what it meant to be a polyartist. One of his lines I recall is "We continue to think to continue". And he does (Andrews, “Re: on 'code poetry'”).

Joseph Keppler’s influence on Andrews as a visual poet has been discussed in section 3.1 of this chapter, but this section merits brief elaboration on his broader impact on the formation of Andrews as a multimedia artist and poet. Keppler works on sculpture, painting, visual poetry, sound poetry “in a literary way,” as Andrews states above, which is perhaps the strongest aspect of his influence on a young poet discovering the impact media can have on language. This diversified Andrews’ interest in exploring the poetic in several media, and using the computer to do so, and empowered him to move beyond radio into other media.

> By 1989 I'd pretty much done what I wanted to do in radio. And I'd had a little bit of experience with a word processor and a spreadsheet in 1989 at the radio station. And we were coming to understand that computers were going to be very multi-use, at that point. That's when I quit the radio show and went back to university to study Computer Science and Math (Andrews, “Re: on 'code poetry'”).

In the following decade, while Andrews focused on developing his skills as a programmer and poet, he never lost touch with the microphone and music. He was a drummer for a few years with the Laughing Boot Quintet. He started and hosted an open mike in the Mocambo coffee shop in Victoria titled Mocambopo, bringing Joseph Keppler in as the first guest reader. Andrews’ first Web site was a calendar of events for Mocambopo in 1995, which along with a collaborative art show and reading with Keppler, led him to create Vispo.com soon after. So while the bulk of the work that is available from the 1990s is visual and electronic poetry, this side of his poetics was kept on the backburner, so to speak, largely because the Web was mostly visual in its early years and the tools for working on audio for the Web were limited. It wasn’t until 1999 that Andrews found the right tools to begin working with audio again. He provided details on this search for tools in the following excerpt from an e-mail:

> Before I turned to Director, i looked at other options for interactive audio.

I tried embedding multiple media players in a web page. It worked fine on my computer but crashed many others i tested it on. Typically what people would do with audio and HTML was embed a wav or midi file in a page, and it would play while the page was displayed. One sound file.

Java only had 8-bit sound, at that point. Which is very poor quality audio.

Beatnik was for midi work, and i wasn't doing midi.

Flash was almost suitable, but its synchronization capabilities were minimal. The audio API of Flash was considerably smaller than in Director. In other words, the range of audio commands was smaller.

Basically what I was looking for was the ability to synchronize and sequence interactive layers and sequences of audio files.

You have a bunch of audio files A, B, C, ..., Z. The user should be able to choose a bunch of them and arrange them in sequence so they play in whatever order the user desires. And the user should be able to have synchronized layers of audio, so different sounds play at the same time yet are synchronized with each other. And synchronizable as the user desires.

The only trick to sequencing audio files A and B is making sure there is no silence between A and B. A plays and when it's finished, B plays immediately. That was more or less impossible with multiple embedded media players. Java would do it fine but with 8-bit audio. Flash would do it but painfully.

The real challenge was the synchronization of layers of files. Director does it with some pain but it was apparently impossible with Flash..

> I also wanted to be able to copy sound icons. Windows for Shockwave takes care of that (Andrews, “Re: doc”).

Macromedia Director was therefore a good multimedia authoring tool available to Andrews which allowed him to publish his work on the Web, by embedding shockwave files. Right around the year 2000 was the time when Macromedia’s Flash and Director really became the tools of choice for many Web artists and poets, because they integrated multimedia very effectively. Andrews used both, but really focused on Director because of the more complete options for manipulating, layering, and synchronizing sound files.

> Sound is now a big part of the net because of sound compression and streaming technology (which provides sound even to 56k modem connections), increased bandwidth for many via DSL and cable modem, etc., and also because of programs like Napster, sites like mp3.com, and the presence of radio stations on the net. Music and sound has migrated to the net in quite a big way just as writing did previously. Last but not least, multimedia tools for the Web such as Director, Beatnik, Flash, and a number of other products enable artists of all types to combine media in works that stream well to 56k modems (Andrews, “The Art of Interactive Audio”).

In addition to the software and programming tools available to him, the Internet and the technologies that support it developed in a way that was very favorable for audio and multimedia work. An important consideration was that most people used 56k modems to access the Internet at the time, and streaming large works through that limited bandwidth was a constraint that sometimes led to creative solutions, as was the case with *Nio* (more on that later).

His return to audio work after a ten year hiatus in which he had explored visual poetry, learned programming, and used a variety of applications and programming languages to create his poetry had modified his approach to audio work. He had become very familiar with the computer as an inscription technology, and had in the past few years been exploring some of the possibilities of creating mutable and responsive texts as a way of expanding and building upon interactivity between the readers and his works. The computer program became a major force that shaped his approach to audio work (as well as writing and visual art) in this current period in his artistic career.

> Also, the application (I mean the computer program) is in some sense one of the fundamental compositions or, as it were, literary forms possible on a computer. It would be odd to draw some line between art and the application or between the tool and art. If anything, it's territory that needs close scrutiny by artists. Computer programs people use should expand their humanity, not turn them into robots. There are both types around, it seems (Andrews, The Art of Interactive Audio).

"Vismu" (visual music) and "interactive music" are the names and paradigms that have guided his audio work from 1999 onward. Andrews' visual music , to date, consists of the following pieces: *Rude Little Song* (1999), *Prototype* (2000), *Oppen Do Down* (2000), *Nio* (2001), *Arteroids* (2001-4), *Enigma n^2* (2002), *Sarah Vaughn Mix* (2003), *Idea of Order Reordered* (2004), *War Pigs* (2007), and *Jig-Sound* (2007, in progress). *Nio*, though it is chronologically early within this list of visual music, is a centerpiece to this collections, partly because the works previous to it could be considered preparation for Nio and the ones that follow are explorations of a different vein in interactive music, with the exception of *Jig-Sound*, which is also an audio sequencer (more on that later). *Nio* is also a work that thoroughly and successfully integrates the elements of language that Andrews has explored in his poetic career: the visual and kinetic elements are graceful and evocative, the permutational possibilities are placed at the reader's disposal through interfaces that excite curiosity and invite play, the sounds are catchy and combine to create engaging musical phrases, and the work is both a product and a tool which could be repurposed yet feels at home with the musical phrases at its disposal.

Nio consists of the work itself in two "verses," some static and kinetic visual art derived from *Nio,* two essays by Andrews ("The Art of Interactive Audio" and "Technotes on Nio and Audio Programming with Director 8"), the source code for *Nio*, and an interview conducted by Randy Adams about *Nio*. All of these are available in the main page for *Nio* (http://www.vispo.com/nio/index.htm) and do an excellent job of exploring Andrews' poetics and how they inform the work, as well as showing the nuts and bolts of this electronic object. This section will not attempt to duplicate what these works already do so well. Instead it will focus on showing the characteristics that lead up to Nio and what pieces are informed by *Nio* itself, in order to demonstrate *Nio*'s centrality to Andrews' current stage as a Web artist, musician, and poet.

The first work of interactive audio Andrews created was *Rude Little Song*, using Macromedia Director 7 in 1999. This piece is an exploration of the software's ability to process sound in order to create interactive audio. *Rude Little Song* presents a simple interface: squares which trigger sounds upon a mouseover and a slide bar for volume control (see Figure 16).

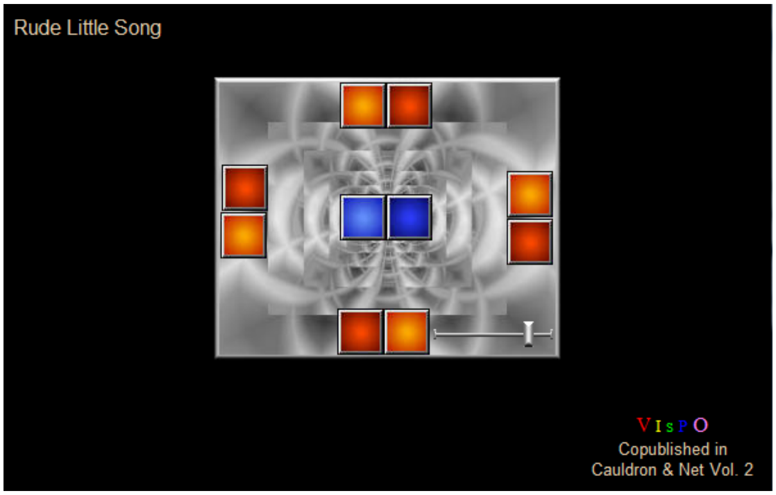


Figure 16: Rude Little Song

The sounds in this and all his other Vismu works are all generated, recorded, and edited by Andrews himself. They consist of finger snapping, whistling, singing, and other vocalizations, and when played individually or combined with other sounds form music that evokes *a capella* singing. The vocalizations are largely musical in focus, and if transcribed phonetically would yield occasional letters sounds, but only in very few cases would contain words. Andrews has a voice that has been well trained by his 6 years working in radio and his lifetime interest in music and poetry: and it is a resource he uses well. The fact that we are listening to bits and pieces of Andrews himself, singing to us as we interact with his colorful and visual interfaces, breathes great life into these little language widgets he has created. *Rude Little Song* is very rough and limited when compared to more polished pieces like *Nio* (“rude” in Andrews own words), but it comes to life with Andrews’ voice, and showed him that Director, indeed was a tool that would serve his purposes.

When Macromedia released version 8.0 of Director in 2000, it did so with a new audio engine, and enhanced functionality. To test out the synchronization capabilities of Director, Andrews created a short piece titled *Prototype* in 2000, which is an early version of *Nio.* This is the first piece he labels as “Vismu,” which reveals a committment to the concept of interactive, visual audio for the Web.

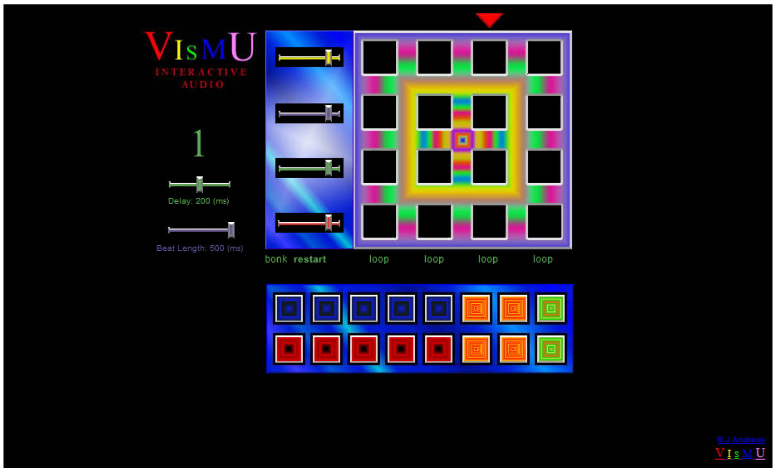


Figure 17: Prototype

Prototype features the same kind of minimalist design as his previous piece: slide bars and colorful squares, but this time he enhances the interactivity by allowing his “readers” the ability to combine the squares in layers, sequences and loops. The squares do not provide much information about the sounds they trigger and they are not as visually engaging as some of his later work. This is because with this work and Rude Little Song, Andrews was focused on developing the “engine” that would allow him to layer and synchronize sounds, so he could then build a visual interface that was more in tune with his poetics. A much more polished design based on this concept is what appears in Verse Two of *Nio*, which was the first verse he wrote (more on that later).

In his next work, *Oppen Do Down* (2000), Andrews simplified the interface visually to bring it closer to language and the poetic by using white words in various sizes arranged on a black background in the center of the browser window. Visually, it evokes both *Seattle Drift* and *Enigma n*, reading as if he had "done the text," "discombobulated" it, and then "stopped it" (see Figure 18).

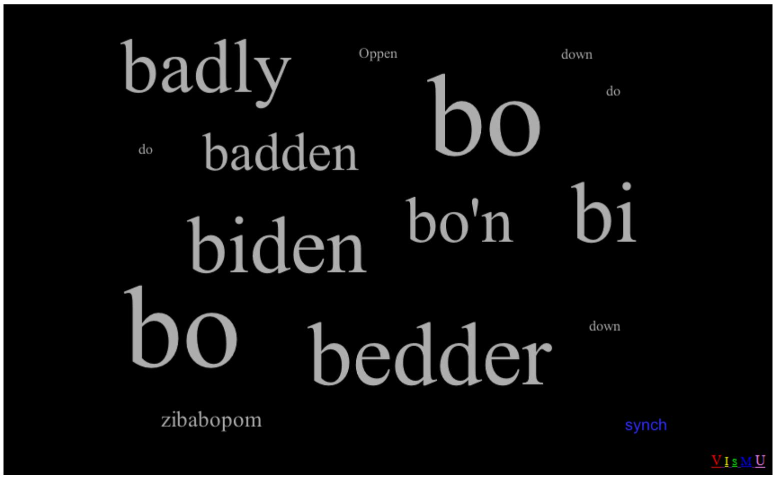


Figure 18: Oppen Do Down

Close scrutiny of the visual components of this e-poem isn't necessary to realize that the largest words all begin with the letter b and that the other words are much smaller and arranged along the margins. The difference between white and colored words is also noteworthy, because they distinguish different classes of objects. Starting from the lower right hand corner, the multicolored logo "Vismu" identifies the piece as belonging to this category of Andrews' work. The blue "synch" is a tool that synchronizes sound layers when multiple words are selected on the screen. The white words turn light blue when selected, triggering the five second sound loop associated with each word. Up to eight words can be selected simultaneously, after which clicking on a new word causes the previous chosen word to become unselected.[88]

Oppen Do Down loads with the word "badly" preselected to get things started. It is not surprising that the word is on the top left corner, evoking the most basic of reading conventions in Western culture: the starting point. This word is one of three words in the poem that contains multiple layers of sound, in this case the sounds are: fingers snapping to the beat of the piece (alternating between single and double snapping), "Oppen do down do down do down," and what I can roughly transcribe as "baduba barabambam bombobo." All the other words trigger single layers of sound, though perhaps some of the choral voice effects are achieved through multiple layers of the same sound clip. So Andrews' choice to preload the first sounds of the poem gives the readers an initial sense of what the piece is all about, eliminating the need for instructions, inciting some curiosity in the readers, and inviting playful exploration of the piece.

The simplicity of its design masks powerful complexities within the piece, and one can potentially go deep into the work. The combinatorial possibilities are enormous when you have 14 words (and the sounds they trigger) that can be selected individually or combined in groups of up to 8, yet the compositional range is more limited. One doesn't have to spend years going methodically through all the permutations to get a sense of the musical possibilities *Oppen Do Down* offers, but at the same time the sheer enormity of possibilities keep the work fresh session after session. So what are some avenues for exploration offered by *Oppen Do Down*? I will

[88] There is an exception to this rule: the first time one clicks on a ninth word, the first word one selected becomes unselected.

discuss three routes into the poem that correspond with Andrews' engagement with language: visual, semantic, and musical.

The initial description of the poem already highlighted several visual characteristics, reiterated here: the allusion to Andrews' DHTML poems, the foregrounding of words that begin with the letter b through font size, the use of color to identify selected words/sounds, and the evocation of the page with its initial preselection of the word "badly" on the top left corner. The words are arranged deliberately on the space of the page along two main central axes for the large size "b" words, and all the smaller words are arranged peripherally to form three word groupings.

The first group forms a diagonal axis from the top left corner to the bottom left corner (see figure 19). It consists of two-syllable words that are related semantically around the initial concept of "badly." So to "badden" something could be seen as making it badder (which is not necessarily worse, given how "bad" has come to denote "cool" during the past 20 years). "Biden" doesn't have a clear meaning, though it could be the (grammatically incorrect) past tense of "bid" or "bide" or a homophone of "bitten." "Bedder" is a mispelling that has roughly the same pronunciation as "better," completing a progression from "badly" to "bedder," which could be seen as either an improvement from bad to better or from cool to cooler. What makes it better, one might ask? Perhaps it is the very departure from normalized spelling and spatial formatting, from customary and conventional uses of language to reinvented, repurposed language, with new arbitrary relations between grapheme and phoneme.

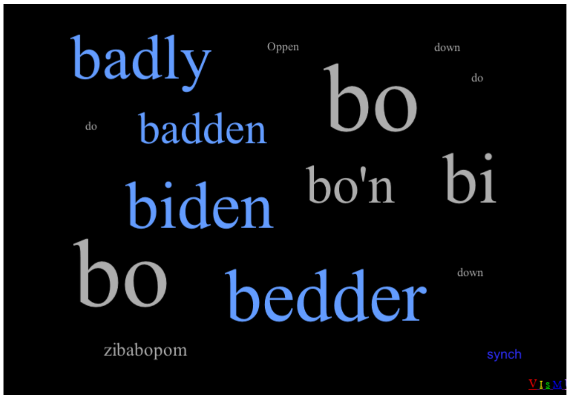


Figure 19: first axis

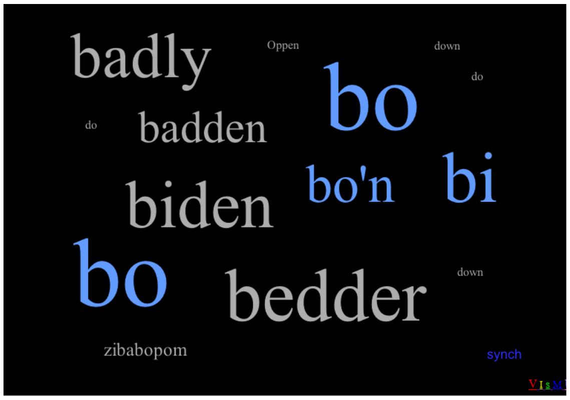


Figure 20: second axis

Figure 20 highlights another visual grouping of words that could be seen as a second axis. When these four monosyllabic words are highlighted together and synchronized, it is clear that they have the same rhythm, but achieved with different syllabic units: one for each of them, except for “bi,” which uses two syllabic units. This visual grouping has more musical simiarities than the other two, an element to be discussed further below, though the semantic connections aren’t as strong. I will not venture into further readings of these visual and semantic groupings and will instead point out some of the sound groupings, since this is, after all, a visual music piece.

One aspect of the visual interface not yet discussed becomes a useful tool for the following analysis: the words can be dragged and dropped by the user to create new arrangements. I have used this function to group the words by similar sound phrases, the results of which can be seen in Figure 20.

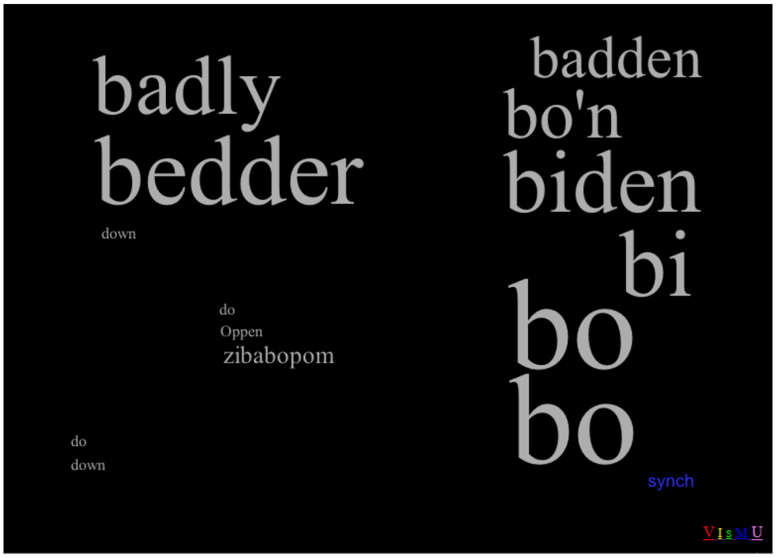


Figure 21: *Oppen Do Down* sorted by sound types

There are four distinct groups of sound in this poem. The first group on the top left hand column all work with the basic phrase “Oppen do down do down do down” and have a layer of finger snapping mixed in. “Bedder” and one of the “down[s]” are basically the same sound, and the only difference between them and “badly” is the layer of “baduba barabambam bombobo” mixed into “badly.” The second group (immediately beneath the first) consists of variations on the “baduba” sound in the first group. “Oppen” and “do” are the same musical phrase except that in Oppen there’s a single voice and there are multiple choral voices in “do.” The third group “do” and “down” in the bottom left corner use mostly vowel sounds and are less punctuated by the /b/, /p/, /d/ sounds that so present in the other musical phrases. The fourth group is the largest one, the column on the right hand of the screen, and it consists of slight variations on what is basically the same musical phrase, articulated with different consonant sounds, using /n/, /m/, and /b/. Two exceptions: “bi” uses two different consonant sounds (/d/in the first half and /n/ in the second half), and “badden” is all whistling. The fourth group is all high pitch sounds (with the exception of “bi”) and the rest of the groups are mid and low pitch sounds. Something noteworthy is that repetitions of the same word do not produce the same sounds, and with the exception of “bo,” are in completely different sound groups.

> The sounds, taken together, are an example of a "heap". This is a term I use to describe a bunch of sounds that can be sequenced and layered to form a larger piece. They didn't start out as a song that was cut apart. They were composed in a multi-track recording program called Sonar in such a way that most of the sounds go together with most of the other sounds (Andrews, “Oppen Do Down [email]”).

Its significance is that it reinforces an arbitrary relation between words and sounds, a technique used by Andrews in Lettrist fashion to challenge the readers’ assumptions of meaning, sound, and language. The relation isn’t always arbitrary, as is the case with “zibabopom” (which actually has a verbalization of the invented word), but it is arbitrary often enough to dispel a comfortable prediction of the sound based on how one might read the word. This unpredicability is designed to excite the reader’s curiosity and invites playful experimentation with the piece—a characteristic Andrews explored further in *Nio*. The sounds themselves aren’t as arbitrary as they might seem, as can be seen in Andrews’ description of how he composed the sounds:

> The music was created like this. I'd record finger snapping first. Then I'd turn that into a perfect loop. Then I'd play the loop and groove on it until some music occurred to me. Then I'd record that. When I'd record, I set up Cakewalk so that as I recorded, it split it up into loops as I recorded it, loops

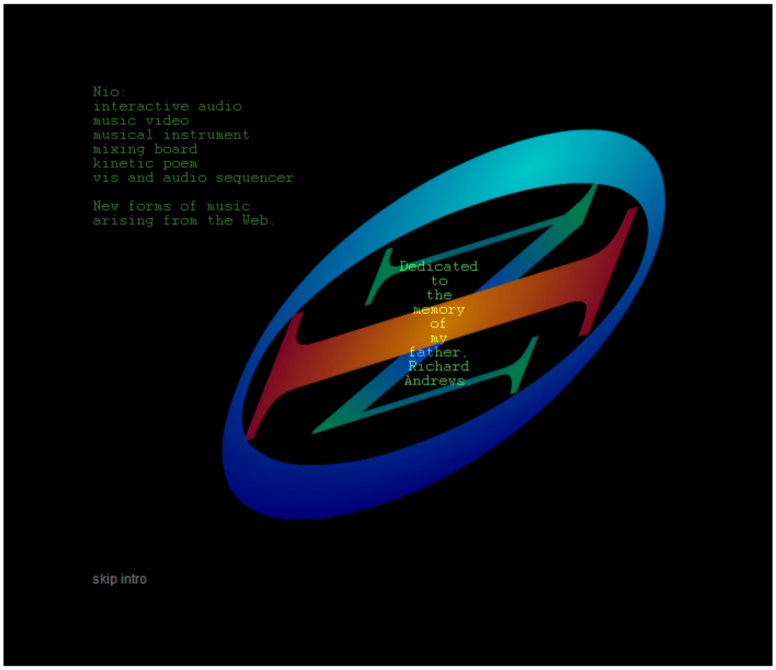
> the same length as the finger snapping loop. Then I'd save some and delete others. So at this point I have several tracks, all the same length, arranged vertically in the time line. Then I'd mix some of those, perhaps, the vocal tracks, I mean, and come up with 1 to 3 combinations that sounded OK. Then I'd listen to those looped, groove on them, until some different music occurred to me that went with the previous stuff. And then I'd just repeat the process. This way, I was creating loops that more or less went together really well (“Re: Finally, Chapter 3!”).

Why is this poem titled “Oppen Do Down?” Does it have any relation to Objectivist poet George Oppen? I suspected not, and Andrews confirmed in an e-mail that he wasn’t thinking about him when he recorded and wrote the poem (Andrews, Re: Oppen Do Down). I think the title is phonetic rather than allusive, and could be read as “up and do down.” Its rhythm is symmetrically balanced: it is composed of a trochee followed by an iamb. Scanning most of the musical phrases reveals that they consist largely of iambic tetrameter (“**op**pen do **down** do **down** do **down**”) though with plenty of substitution to keep things varied.

I think the poem tantalizes us by using language, but is designed to resist interpretation and “meaning” in order to favor the musicality of language freed from the constraints of meaning. It is an interactive sound poem, in which the poet’s voice has be recorded, cut, mixed, and tied to a word-based interface that suggests (purposely imprecisely) the musical phrases the reader will be able to select, layer, and synchronize to create many possible musical outcomes.

> When I present Oppen Do Down publicly, I basically make a song of it. I start out with just one sound playing ("down" down in the bottom right) and slowly add and subtract other sounds to form a song that's about 3 minutes long (Andrews, “Oppen Do Down [email]”).

Nio takes this idea to anotherlevel by practically eliminating ties to words and the semantic baggage it carries. In true Lettrist fashion, Nio employs letters, glyphs, and visual art derived from these linked to verbal and nonverbal musical sounds similar to “Oppen Do Down” and his earlier vismu pieces, but more polished and varied. Jim Andrews worked on this poem for a year, funded by Turbulence.org and presented it at the E-Poetry 2001 Conference in Buffalo, New York. This was the first project for which he received funding and could be a full time artist, and it allowed him to obtain funding for his next big project: *Arteroids*. It is also a project that advances all three areas of his poetics—his vispo, the presence of the ‘neath text, and the vismu—and does so in an integrated fashion. It was Andrews most ambitious work to that point and, in some ways, is his most successful one, in part because of the number of concepts he brings together in one piece, as shown in his introduction (see figure 21).



Nio:
interactive audio
music video
musical instrument
mixing board
kinetic poem
vis and audio sequencer

New forms of music
arising from the Web.

Dedicated
to
the
memory
of
my
father,
Richard
Andrews.

Figure 22: *Nio* introduction

This still from the introduction to *Nio* describes the work in several ways: “interactive audio / music video / musical instrument / mixing board / kinetic poem / vis and audio sequencer.” These descriptors bring in many different schema to the table and one might wonder how the piece is able to be successful at all of these things in a way that is engaging to an audience. Andrews’ background in radio informs some of these conceptual domains and he creates tools and interfaces that allow users to layer and sequence sounds and their associated images in two different ways, as seen in the two verses of *Nio* below.

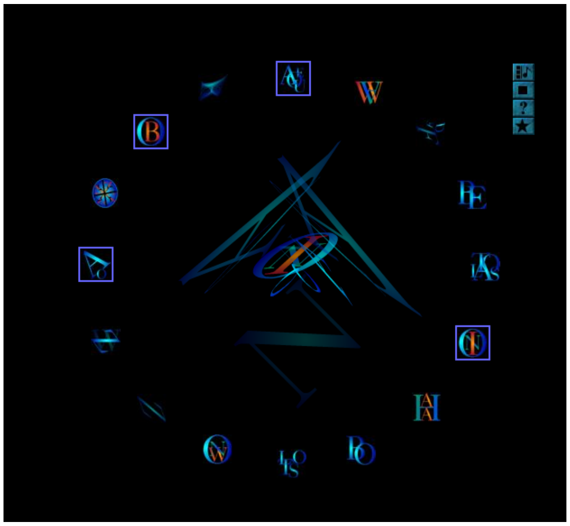


Figure 23: *Nio*, verse one

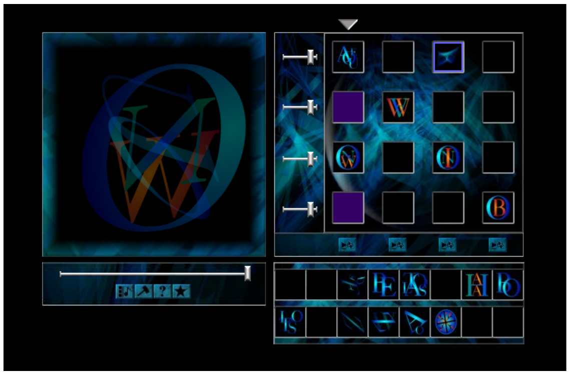


Figure 24: *Nio*, verse two

The two verses in *Nio* can be seen as extensions of two of Andrews' previous vismu pieces, conceptually and in how they employ their interfaces. Verse one has a similar interface to *Rude Little Song* and *Oppen Do Down*: clicking on icons or words activates the musical phrases associated with each, and multiple icons/words can be activated to layer up to six sounds. Verse two and *Prototype* both work with a four by four grid, which offers volume and loop controls, and icons linked to sounds for his audience to drag and drop into the grid. Verse two allows for layering of up to four sounds in four sequencing tracks, has a randomization icon that will randomly reassign the icons already placed within the grid.

> *Nio* was only verse two for the longest time. But then it came time to make the thing stream nicely to 56k modems. And I found that it would be a lot of work to make verse two stream like *Nio* does. So I introduced verse one, which is a lot like a previous work called *Oppen Do Down*, only a bit more deluxe in some ways. That's how verse one came about. The funny thing is that many

> prefer verse one to verse two (Andrews, “Defib: Randy Adams interviews Jim Andrews about *Nio*”).

Even though there is no accounting for taste, I can venture forth an explanation for why verse one is preferable to verse two. Verse one simplifies the interface so that it is more intuitive to people who regularly use the Web. Readers only need to click on an icon to be rewarded with music and animation both of which invite them to continue selecting and unselecting icons to create pleasurable combinations of animation and sound. The visual design is also more inviting: the circular arrangement of the icons creates a perfect space for the animations to unfold, and since they seem to be tumbling down towards a vanishing point, the circle and its black background can be read as a hole in cyberspace into which the letters can be pushed in by the reader. Verse one is like a musical instrument placed at the disposal of its audience to play with creatively. Even though there are fewer options in verse one, since it doesn’t allow sequencing (only layering), these options are sufficient to explore the compositional range of the piece.

Verse two feels more like a tool: an audio player and mixing board that allow for more complex interaction and can therefore be used to expand the compositional range of the piece. For those interested in crafting something more complex with the materials and tools provided, this is the verse of choice. But for the more casual reader, the number of options and the grid and window interface, with slide bars and other controls may cause the impression of greater complexity and a longer learning curve and therefore be less inviting. Verse two also deemphasizes the integration of the visual and sound components by spatially arranging the animations and color control in a window on the left side of the screen and a the icons and sound controls in the grid on the right side of the screen. By dividing the audience's attention between two separate areas and types of controls, the impact of creating a "kinetic poem" or "music video," is reduced, emphasizing the "mixing board" and "vis and audio sequencer" aspect of Nio. Andrews comments on this on a article in 2003.

> These [online synthesizers] generally allow you to layer and sequence sound, and sometimes sound and visuals. Note that all of these use a grid structure. Sequenced sounds are represented as horizontally contiguous sound-icons; layers or channels of sounds are vertical. This is all a bit too rectangular and *sound-studio* influenced - the designs resemble bland devices from recording studios rather than having the sort of thematic relevance we find in the devices of *Electrica*. Getting away from the grid paradigm may provide more interest to the visual dimension and allow the visual dimension to be more thematically interesting, be art in its own right, as opposed to merely following function. Form should follow function, but if that is all it does, then we end up with understandable but merely functional interfaces (Andrews, "Interactive Audio on the Web 1").

I think Andrews achieves a more thematically interesting form in verse one, even though it comes at a cost of functionality. For that reason, Nio would feel incomplete if either verse was missing. As discussed earlier in the section on *A Pen*, the application is an important part of Andrews' poetics, and *Nio* is clearly an application designed to create visual music. Andrews blurs the lines between tool and product: presenting both to his readers for them to not only marvel at and play with the result, but actually get a feel for how that artistic expression was created and take a hand in shaping the final work. Blurring the line further, Andrews made the source code available in the main page for others to either use the “engine” for other purposes or simply substitute their own animations, record and attach their own sounds, and create their own visual music with the *Nio* tool. The fact that anyone could repurpose the Nio code emphasizes Andrews’ committment to interactivity. How the piece responds to Andrews’ poetics as a programmer who favors interactivity and permutation should be clear by now, so I will discuss how this piece advances his visual and audio poetics.

Before *Nio*, Andrews used software and programming languages that allowed him to point in the direction he wanted to develop in, but couldn’t achieve due to technical limitations. In the earlier discussion of *A Pen*, we saw how Andrews used CorelDraw to create visual and letterist pieces that evoke kinesis with static works. Using Corel’s patented image sprayer, he used letters as nibs on a pen to leave traces of movement over the virtual surface of the screen, writing with language rather than writing to produce language. His DHTML poems allowed Andrews to make his words drift on the screen space, and allowed for the letters in *Enigma n* to move in circles along varying axis, allowing readers to change formatting aspects of the text. The movement of these pieces worked fine for the pieces they were created for, but it seemed that he had a difficult choice to make at the time: he could manipulate words and letters visually or kinetically, but not both to the extent he wanted to.

Director and Flash allowed him to do both and with the music he recorded created for *Nio* he had a tune for these letters to dance to. The reason for this is the shift from using raster graphics (with programs designed to create bitmaps, like CorelDraw) to vector graphics in Flash. The letters that dance in the screen aren't exactly images described bit by bit anymore, they are mathematical formulas that render the images, and are therefore much more amenable to smooth animations and transformations. Andrews used Flash to create animations for Nio that matched the rhythm of the recorded music, and imported them into Director. With Director, he then created sixteen electronic objects, each composed of three different kinds of objects linked together: a musical phrase, an iconic graphic, and an animation. The icons are composed of letters that have been stacked, blurred, or otherwise transformed arranged into compact representations of the animations and sounds they activate when selected. A perfect example of this is the Nio icon, which represents the work and is one of the icons available for selection in the poem (see figure 25).

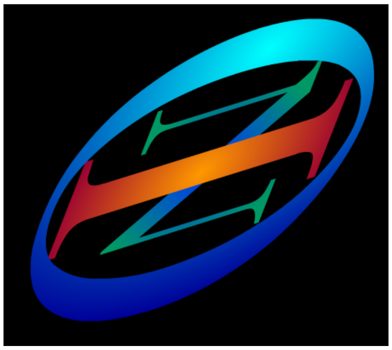


Figure 25: *Nio* icon in Introduction

The Nio icon suggests the very shape and animation of verse one of this e-poem by placing the N behind the I and surrounding them with the O.[89] This could be interpreted spatially, as if the letters were the same size, but arranged one behind the

[89] The Nio icon appears sideways in the introduction, but it is upright in Nio itself. With this orientation, the N could be read as a Z, a motif that appears in other animations in this piece, as well as in some of Andrews' earlier visual poetry.

other at a distance that would create this arrangement through perspective. This evokes both the animation, which consists of letters falling within the circular space of the poem and the outward projection of sound waves from the computer's speakers towards the readers. The sound activated by the icon can be transcribed as "na**na**na**na**na**na**na **nio**" (boldfaced letters mark stress, which like most sounds in *Oppen Do Down* is can be scanned as iambic tetrameter), which in traditional sound poem fashion emphasizes the sounds of the word "Nio."

The title of this work has no meaning that could be found in a dictionary, though it could be associated with "new" or "Neo" (the protagonist of the Wachowski brothers' cyberpunk film "The Matrix"). The sense of innovation one can connote from the word, when seen in the light of how Andrews employs language in this work, leads me to think that Nio is about freedom from meaning and convention. *Nio* employs letters and sounds freed from the rules that govern their spatial, phonetic, and semantic arrangement into words, something Andrews has been doing throughout his career.

> I've been drawn for years to visual poetry, particularly lettristic visual poetry that deals in syllables and letters as opposed to words, phrases and sentences. For the above reasons, but also because in the digital realm the shapes of letters are more various than the shapes of words, which tend to be elongated rectangles. And, as a programmer, a letter is typically a continuous thing on which various transformations/animations are more visually appealing and suggestive than on whole words or sentences. Letters are characters. They have more character than words do, in some ways. And since they take up less

> memory than whole words, they're more amenable to smooth animations (Andrews, “The Art of Interactive Audio”).

When focusing on the visual aspect of Nio, one could argue that it is about exploring the character of letters through animation, “a kind of Lettristic dance” (Andrews, Defib: Randy Adams interviews Jim Andrews about Nio). The *Nio* animations have been the most important recent development in Andrews’ visual poetics, because he makes his letters dance with a gracefulness that was unprecedented in his earlier work. While his earlier visual poetry suggested motion, transformation, depth through shapes, arrangement, and perspective, these animations do so by seamlessly integrating motion and formatting changes. The dance of these letters creates the illusion of depth by floating, rotating, sliding, or twirling, all along a linear or spiraling line most frequently from the edge to the center of their space, losing size until they disappear as they reach the end of the animation. This size reduction, in addition to having a consistent direction for the movement, creates a sense of words moving towards a visual horizon or vanishing point. The circular arrangement of the icons in verse one suggests a well or vortex in the black space of the screen into which the animations and sounds fall when triggered.[90] The more icons have been selected, the larger the number of letters and sounds are cast into the virtual space of the computer and the very real space of the world outside of the computer by way of the computer’ speakers (devices with a circular shape).

[90] The choice of a black background with colored letters is also significant computationally: a black space on a screen is basically a space with no signal, a silent space to which the screen reverts to when turned off. Conversely, the white space of the remediated page in this document (when seen electronically) is full of digital signals constantly being refreshed on the screen space, and the letters are little silent spaces.

There are too many references to circularity to ignore this as a signifying aspect of Nio, a work whose own iconic representation reinforces this idea, but one more point should reinforce this importance: the sound and animations are looped, which connects the ending right back with the beginning. The fact that there is a slight pause between ending and beginning is simply a nod to that breath that the virtual poet must take before repeating the same musical phrase and sending the letters back on their journey into the silence of the black space on the screen and cyberspace. This loop takes what is visually a circle and transforms it conceptually into a four-dimensional sphere, creating a closed space where meaning cannot enter and a new language can dance. The opening into this space is created through interactivity: this dance can only happen through the choices of its audience, and the space closes around them the moment they click on an icon. Once inside, their preconceptions are challenged by this Lettristic dance, while they play with the piece, and hopefully they emerge changed in the way they think language can be used, if only in a small way.

One of the reasons Nio keeps itself accessible and inviting is that the reader isn't left at the mercy of randomly reconfigured language: there is a method to the madness, and patterns emerge from close readings of this language. I have identified four groups or types of objects, according to their sounds, animation, and/or icon, as well as by patterns in their arrangement in verse one. For the analysis that follows, I will provide screenshots of the groupings, but since sound clips are out of the question for a document designed for print, I invite you to go to *Nio* (http://www.vispo.com/nio/Nio5.htm), match the selections, synchronize them for accuracy, and listen to the sounds.

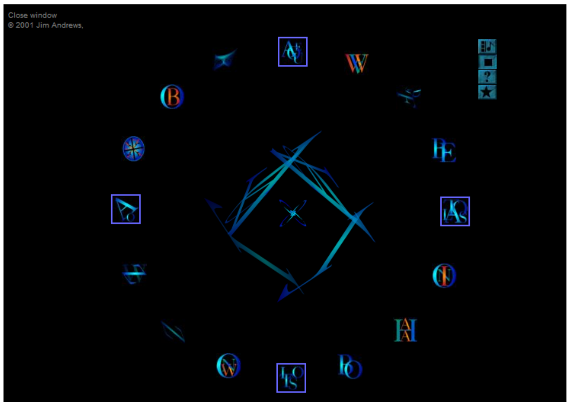


Figure 26: group one

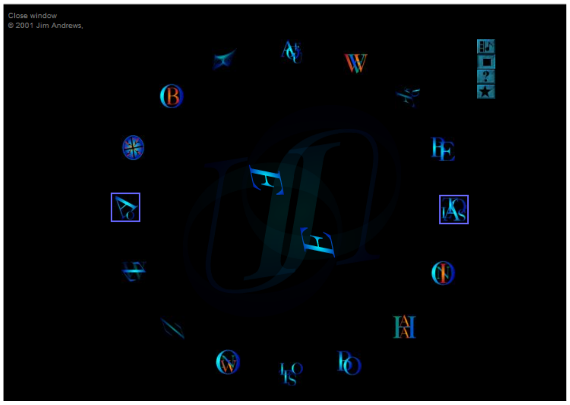
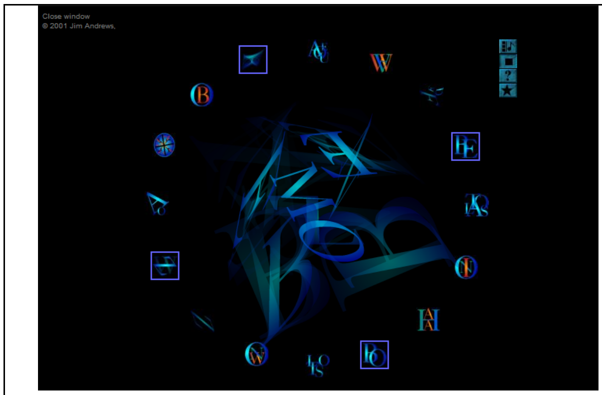


Figure 27: group one subset

The four icons selected in Figure 26 all contain the same audio track and are the only ones with two layers of sound already mixed: “lambara bambam badubaba” and finger snapping (alternating single and double snapping). The animations are composed of almost the same elements, so that when selected together and synchronized, they form a kaleidoscopic arrangements of the letters. Eliminating one of them creates a gap in the pattern, breaking the visual, if not the musical or temporal loops. The arrangement of letters A and L creates a space that frames the atom-like pattern created by the interlocking Os in the center of the screen. Andrews is showing an alternate way of combining the smallest units of written language, creating both static and kinetic visual art (still images via a stop button on the menu). Musically, this is the most recurrent pattern in the work, which means that it serves as a base for all the other layers of music that can be built around it. The finger snapping to the beat, helps set the rhythm for the whole music in a way that is catchy, bodily, and simple enough to invite the audience to join in.[91]

An aspect of the music which makes things more complex than his earlier work, like *Oppen Do Down,* is that each musical phrase is repeated twice to create a complete animation loop. The music sounds the same, but it allows for mixing animations, as is the case with the icons selected in figure 27. These two icons have a different animation during the first musical phrase, and the same animation (though oriented in opposite directions) during the repetition of the musical phrase, as can be seen in the screen capture above. Note that interlocking, rotating, fading, and overlapping letters are all ways in which Andrews subverts the mainstream conventions for the arrangement of letters into words, while creating visually appealing language art. See Table 3 for three other Nio object groupings to see how he arranges some of the language in this poem.

	What defines this group is its use of the /b/ and /m/ sounds to form syllables, the use of a single voice (unlike the choral voices in group 1). This group is composed of two subsets—the icons on the left of the circle and the ones on the right. The musical phrase for the subset on the right is identical, as well as the animation (except for their

[91] While the whole vocal apparatus, from the diaphragm to lips, is employed in any speech act, poetry tends to draw attention to one’s articulation of language and therefore to one’s body as a language machine. Sound poetry often takes that to the next level by choosing and combining sounds purely for their articulatory and sound value and relegating meaning to a secondary order of consideration. The finger snapping in Nio brings awareness of the hands into the bodily experience, particularly the thumb, index, middle fingers—all three of which are needed to operate a mouse, incidentally.

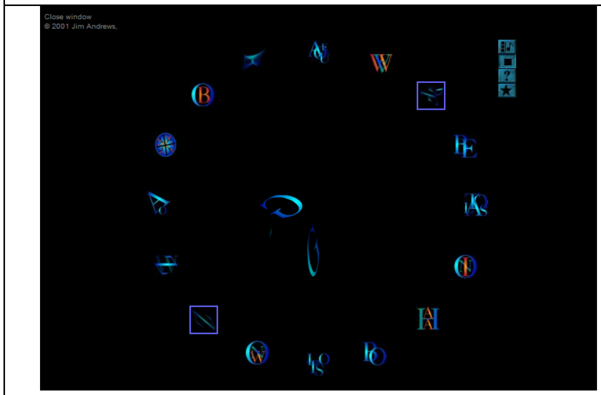
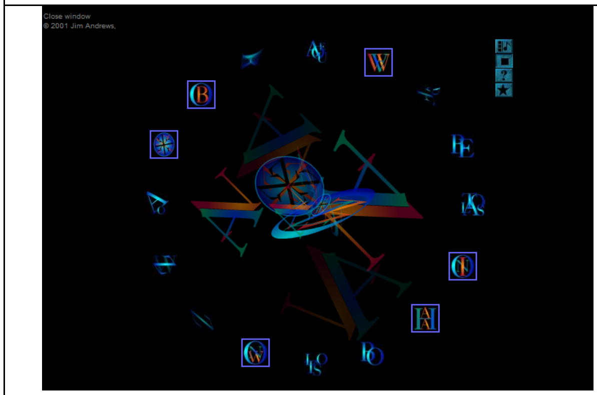
Group 2	orientation), but what distinguishes them is a slight tonal variation: the lower icon is slightly lower on the musical scale.
 **Group 3**	This group is the only nonverbal one: it consists of whistling two different tunes that become the same tune at the very end. This divergence and convergence in musical phrase is matched by the animation, as shown in the screenshot.
 **Group 4**	This group is the largest one and probably contains subdivisions within it, but they have the strongest convergence of characteristics. They are the only icons and animations that use red and orange tones, along with the blue and green used in all the other ones. Musically, they all have in common several /na/ sounds, in a variety of tones and rhythms, as well as being sung in falsetto. They all use stacked NA or AN letters in two tones, floating from various positions in the edge of the circle towards the central vanishing point, and an icon composed of letters, such as the *Nio* icon.

Table 5: Object Groupings in Nio

There is much more detail one could go into analytically, particularly if one possesses a musical background or is familiar with Director and programming in Lingo. In depth phonetic or phonological analysis would also yield articulatory patterns that could prove insightful. Another productive avenue for exploration is how

Nio works as an electronic text because it exhibits all the behaviors in my typology: describing it as a responsive, mutable, kinetic, scheduled, and aural text. Here is a work that would be extremely difficult to carry out in anything other than a computer, because it employs its capacity for processing, feedback loops, it constantly redefines or reasserts the state of its objects, and it takes advantage of the multimedia and simulation capabilities of a computer to create environments for its texts to dance and sing. All of these approaches have great potential because Nio is such a rich work of art, music, and Lettrist poetry, in addition to being an an application and a set of tools. In short, Nio is the most successfully integrated work by Andrews, arising from his earlier experimentation and informing his subsequent artistic, musical, poetic, and programming work.

When Andrews presented *Nio* at E-Poetry 2001 in Buffalo, New York, the Canada Arts Council (which had previously rejected an application from Andrews to fund *Nio*) asked him to reapply. He did so and obtained funding for his next big project, *Arteroids*. Even though Andrews classifies this e-poem under Vismu category in his Web site, I feel that it is not musically driven, and will therefore not discuss it in this section. *Arteroids* is a very important work, however, and it will be the focus of the next chapter in this study.

Andrews has produced several other works of visual and interactive music that could be grouped together based on what they do: they cut up songs, mix them and/or loop the fragments, providing the reader an interface that both interprets the music visually and allows the listener to select loops within the song. These works are titled: *Enigma n^2* (2002), *Sarah Vaughn Mix* (2003), *Idea of Order Reordered*

(2004), and *War Pigs* (2007). All of these works could be considered remediated tributes to the poem or music they work with, dissecting and reconstructing (and in the case of the *Sarah Vaugh Mix*, combining two musical pieces) the works they are about.

Enigma n^2 returns to some of the ideas Andrews was exploring in *Enigma n* back in 1998, primarily interrogating meaning construction in language. *Enigma n* achieves this with the rearrangement and reformatting of letters, while *Enigma n^2* does the same by cutting and shaping the playback of a recording of the word "meaning" spoken three times by Andrews, "twice forward, once backwards" (Andrews, "Chapter 3"). In both cases, Andrews employs random elements, as well as some user control over the text placed before them. The interface is much simpler for *Enigma n^2*: the user can click on different parts of the sound graph placed before them to determine the beginning of each portion or they can allow the program to follow its own randomly generated cutting of the phrase and repetition of the excerpts it creates. The length of the clip is always randomly determined, and unless the user clicks again, it will repeat 1 to 6 times, before changing to a new random location (see Figure 28).

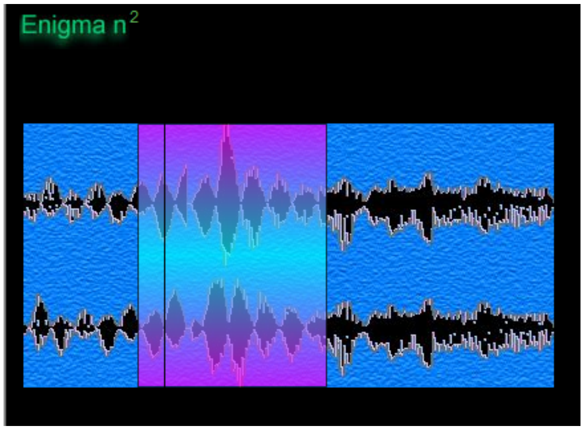


Figure 28: ***Enigma n^2***

The effect is rather visceral: Andrews voice is deep and guttural, speaking the word “meaning” in two different inflections, and emphasizing different vowel and consonant sounds in the word in true sound poetry tradition. On top of that the cuts often come in places one doesn’t expect, such as in the middle of a sound or syllable. When a short portion is repeated multiple times, it can create unexpected rhythms or words, such as “me/me/me,” “mean,” “meaning” “na/na/na” and others. If we interpret some of these words, an idea emerges that is consistent with Andrews’ previous expressions and the original *Enigma n* poem: meaning is constructed by me/me/me, but it may very well mean na/na/na (naught? nothing?). Andrews’ sonic cut and reconstruction of the word “meaning” successfully expresses an idea he has been exploring since the beginnings of his poetic career: how can meaningful work be created and communicated in ways that take advantage of the particularities of different media and technologies?

> The work of Godel and Turing, which has led us to the development of the computer, also resulted in a situation where languages are as much of a topic of study in mathematics as the ole particle of physics. Consequently,

> human language, codes of all kinds, and, basically, anything that can be turned into information are all placed into an uneasy proximity of code and language stew in the computing environment. It's all just ones and zeros in the computer. It all gets mixed together in an unholy and, to me, fascinating kitchen sinkness. Languages get mashed together. So do different media and arts. And the networking environment leads to a hyper connectedness where connections between ideas (crucial to innovation) are made and noted at an accelerated pace.
>
> Nio's engagement with language is an engagement with synthesis of arts, media, and types of language, from the Shockwave works in Nio to the essays, source code, still graphics, and so on (Andrews, "Nio [email]").

Andrews' return to audio work with the tools computers place at his disposal, and more importantly, the tools it allows him to create has been a vital development in his growth as a poet, programmer, and artist. His visual poetry benefitted from shifting from raster to vector graphics and from dancing to the rhythm of his music. As a programmer, his need for more sophisticated programming tools has led him to create products, such as Windows for Shockwave, which allowed him to expand the capabilities of Director to create an e-poetry game titled *Arteroids* (2001-4) as well as create beautiful still images with a new art tool he is currently developing titled *dbCinema* (2007, in progress). His vismu project continues with two works in progress: *War Pigs* (an interactive music game) and *Jig-Sound* (an expansion and elaboration of the concept behind *Nio*).

All of these works mentioned in this paragraph merit further discussion, but since they are all works in progress (with the exception of *Arteroids*), they will receive the attention they deserve in future studies. I will conclude this chapter with a quote from a recent e-mail from Andrews in which he describes how he sees himself as a poet at this time.

> As a poet, I am less involved in writing poemy poems than in creating algorithms that operate on language, image, and sound, hopefully to interesting result. Whether the algorithms cut it up or do other stuff. dbCinema is a kind of graphic synthesizer. Nio and Jig-Sound are audio sequencers. The stir frys are text sequencers. I create sequencing and synthesizing algorithms. I synthesize media and arts (Andrews, "Re: ELO Conference Proposal").

Other Works

This chapter has classified and analyzed a large sample of Jim Andrews' *oeuvre*, focusing on three major areas of his artistic production and development, but this has unfortunately left out some important works that cannot be analyzed within the scope of this study. There are roughly three groups of works not included in this chapter: text poems (including hypertext), tributes (which includes both homages and digital restoration work), and works in progress. This section will briefly describe these as a means to conclude the discussion of Andrews' poetry and poetics and lead to the next chapter, titled "Mining the *Arteroids* Development Folder."

As discussed in the first section of this chapter, Andrews' literary formation was fairly traditional, so it is no surprise that his first explorations of language were with poetry written for the page and for the voice—what he calls "poemy poems." I will refer to these works by Andrews as “text poems” because they foreground linguistic aspects of language, and don’t explore language in visually artistic, musical, or programmable directions. Here is a list of Andrews’ text poems currently published in Vispo.com, along with his descriptions:

- The Riddler (1997)
  About Heraclitus.
- The Material (1996)
  This also is published on Jennifer Ley's site.
- Kasparov (1996)
  A poem on the strength and weakness of the chess champion defeated by a machine.
- Second and Third Snow (1995, PDF, 191Kb)
  Second and third aren't as dramatic as the first, but...
- First Snow (1995, PDF, 53Kb)
  I was there.
- Customs (1994, PDF, 75Kb)
  Customs at the border between Canada and the USA. Originally published in Quarry when Steven Heighton edited it.
- My Friend's Friend (1993, PDF, 150Kb)
  A, uh, hopefully humorous song and dance on love, jealousy, and the cosmic dynamo from my manuscript Several Numbers Through the Lyric.
- Eyes (1992, PDF, 36Kb)
  The beginning of a suite of four short poems. A Web version of the suite is located at Sequence
- Song For Sun and Moon (1992, PDF, 43Kb)
  Second in the suite.
- Border Crossing (1992, PDF, 43Kb)
  Third in the suite.
- An institutionalized avant garde is: (1992, PDF, 45Kb)
  Fourth in the suite.
- LIFE ART (1992, PDF, 29Kb)
  The opening poem of the manuscript Several Numbers Through the Lyric. This was the original version of a piece I moved onto the web in animation fairly early on.
- Lifer (1992)
  A poem for the late poet Luella Kerr.

- Trust (1991)
  In a word.
- The Meeting Place II (1991, PDF, 168Kb)
  There are a few poems I've done that involve various notions of the meeting place.
- Alice In Flatland (1991)
  A fifteen page poem on the edge of knowledge and other boundaries.
- The Secret Life of Trees (1991, PDF, 188Kb)
  A song to good times and hard times and the spirit of the trees outside my place in the city in Victoria in 1991.
- Safe Cracking (1989, PDF, 75 Kb)
  A poem about conspiracy, complicity, guilt, and taking chances. Originally published in Quarry magazine from Kingston.
- La La (1989, PDF, 70Kb)
  A lyrical/anti-lyrical love poem. This operates through the lyric.
- Hands (1984, PDF, 44Kb)
- The Meeting Place (1984)
  This poem is published on Jennifer Ley's site The Astrophysicist's Tango Partner Speaks. A PDF version that I designed is also available.
- Walk in Streetlight (1983, PDF, 51Kb)
- Lower Yates at Midnight (1983, PDF, 47Kb) (Andrews, On-Line Writings and Vispo: Poems)

Andrews' text poems are written in a variety of forms: free verse (often evocative of William Carlos Williams and Wallace Stevens' sparse imagistic lines or Charles Bernstein's prosaic diction), prose poems, and rhymed couplets. They range in tone from philosophical to whimsical, using narrative, colorful images, and references to Greek philophers, mathematical and computer theorists, and technological innovation. Most of these poems were originally published in print and Web magazines, were part of Andrews' 1992 manuscript *Several Numbers through the Lyric*, or were published originally in Vispo.com.

A text poem not present in that list is the *Pornomorphs*, a 17-part hypertext poem (or a series of related poems) written in 1997, which Andrews categorizes as a “graphic poem” in his full site pull down menu. Each part of this poem uses visual art created with the CorelDraw Image Sprayer, but instead of using letters as nibs (as in other visual poems from this period), he employs images of a penis, a vagina, eyes, and other body parts. Each lexia (with the exception of Madame Ovary) is accompanied by a text poem which develops a narrative of sorts about Ur Grrl and Penis Unix, two bots that inhabit the Web.

he and Ur Grrl and the rest of the pornomorphs
populate the Web like wild west
wanted things,
Agents, bots without masters,
the first AI,
inhuman,
not quite like you and me.

(Andrews, “The Pornomorphs: Ur Grrl Alias 2”)

This is a work that is thematically in tune with the rest of his poetic work, but in this case foregrounds a creative sexuality that is usually either absent or more subtly present, as is the case of *Seattle Drift,* in which langwidgets are released to follow their semi-random paths when the reader “does the text” in a sexualized interaction. *The Pornomorphs* have characters that are artificial intelligences (sexualities?), born native to the Web expressing their desires, released from human control—much like the language imbued with behavior that dominates his artistic creations from 1997 onwards.

I include this hypertext work in the category of text poems. They also foreground the textual, using links to assemble a series of lexia. Another group of text poems worth mentioning are the *Pop Up Poems*, which he wrote between 1996 and 1999, using the pull-down menu tool offered by HTML. These charming short poems consist of personified voices that are aware that they have been written to reside inside a pull down menu, which leads them to have low self-esteem. Andrews describes them as follows:

> The pop-up poems haven't fully matured yet, come to accept their place (or lack thereof) in the world. They're still acclimatizing to the 'neath text they inhabit, still dealing with it. They've never really known any other place, but they know that they're in an odd position (Andrews, “Pop up Poems”).

The fact that Andrews was imbuing his text poems with personality is a strong indicator of the conceptual direction he wanted to take with his poetic works for the Web, which he sensed HTML and DHTML would allow him to pursue. Andrews’ hypertext and pop up poems are voices from the ‘neath text, the source code that powers the behaviors his subsequent texts will exhibit, which is the major paradigm shift from writing documents to writing applications described in DHTML Dances.

Andrews’ text poems, whether written for the page or for the Web, are strong indicators of the ideas that he is struggling with artistically, and can serve both as a way to map his development as a poet and artist, and as works worthy of analysis in and of themselves. The focus on electronic textuality that guides this study excludes deeper analysis of these “text poems,” but not without recognizing them as a gold mine of insight on the poet and a worthy subject of critical attention.

Another important area of Andrews' work that isn't included in the scope of this study are his tributes, restoration work, and reworkings of poetry by Lionel Kearns, bpNichol, and Joseph Keppler—three poets whose work is very much in tune with his own.

His 2004 work *On Lionel Kearns*, is "a binary meditation on the work of a pioneering Canadian poet contemplating digital poetics from the early sixties to the present" (Andrews, "On Lionel Kearns"). The words "binary meditation" are aptly chosen to describe a project in which the 1/0 of binary code is explored directly through selected visual poems by Lionel Kearns, as reimagined by Jim Andrews for the Web. Andrews' interest in Kearns is best described in the following excerpt from the essay which accompanies the e-poem.

> He is a Linguist. Which is interesting because one of the reasons I became so excited about his work is he seems to understand that there has been a synthesis going on, over the last seventy years, more or less commencing with the work of Godel and Turing, between number and language, between mathematics and language studies, between the arts and sciences, between any pair of fields that code their material in such a way that it can be turned into information. Kearns is a language man. Kearns is a real thinker. Kearns is a media man. Kearns is a polyartist. Kearns has been contemporary since the mid sixties. I can't think of any other poets from around here except maybe bp Nichol who might have written anything in the sixties about digital poetics and issues (Andrews, "About 'On Lionel Kearns'").

Lionel Kearns' work brings Andrews right back to the first two passions that guided him in his university studies and have defined his career: mathematics and language studies. If binary codes (1s and 0s) are the foundation of his 'neath texts, the math and language are the conceptual foundations of Andrews' poetic work. By exploring Lionel Kearns' work, Andrews is exploring his own foundations as an artist at a time when he had completed his three-year project *Arteroids* (2001-4) and perhaps needed to decide what direction he wanted to pursue next..

> I created 'On Lionel Kearns' via email correspondence with Lionel. I didn't meet him face to face before I finished the piece. He'd shown me his visual poem 'The Birth of God/uniVerse' and told me how early on he'd created that. It was obviously a significant early piece relevant to digital poetics. And indeed Chris Funkhouser thought so too in including it in his history of early digital poetry. At the same time, I was learning 'imaging Lingo' in Director, which is a bunch of techniques for doing image processing. So I thought I'd try learning the techniques while applying them to Lionel's writing. It was a way of meditating on Lionel's work in an active kind of reading looking for connections between Lionel's works and between Lionel's works and digital poetics. And a way of creating a kind of contemporary wreading of Lionel's work. 'On Lionel Kearns' is somewhere between a wreading of Lionel's work, an essay on his work, and a digital poem in itself ("Re: Finally, Chapter 3!").

Clearly, writing "On Lionel Kearns" was both a creative engagement of Kearns' work and a learning experience for Andrews.

First Screening by bpNichol (2007) is an important work of editing and digital restoration made possible by Jim Andrews, Marko J. Niemi, Geof Huth, Lionel Kearns, and Dan Waber. Andrews and Niemi worked on the Javascript programming to port the original Apple Basic animated poems into a format that would run on the Web. Andrews didn't really become interested in bpNichol's work until he was well developed as a digital artist, and it was through his friendship with Lionel Kearns that he was exposed to this early digital work from the mid-80s. Publishing this pioneering digital poetry by a fellow Canadian poet contributed to establishing the history of the digital writing scene and presented an interesting technical challenge that he was concerned with in his own work: time.

> In programming, you can hardly avoid having to deal with repetition and therefore rhythm. Once a computer has done something once, it is very easy to make it do the same thing again. In this sense, programmed work usually deals with rhythm and repetition, whether the programmer does so consciously or not. The loop is one of the fundamental structures of programming. But can you make it funky? Can you give it humanly significant rhythms and feeling rather than just the rhythms of an automaton? In "Off-screen Romance", it isn't just through the animation itself, but through engagement with the work as a written thing both at the level of the screen and the off-screen level of programming that we come to a deeper appreciation of the relations of the piece to writing, poetry, and synthesis with other arts such as music, cinema, programming and other time-based arts. In

> this sense, “Off-screen Romance” refers to the romance of all that is present but unseen on the screen (Andrews, “Framing 'Off-screen Romance'”).

Time, repetition, and rhythm are key elements in his recent work—vispo (*A Pen*), vismu (*War Pigs*, *Jig Sound*, *F8MW9*), and cinema (*dbCinema*)—and challenge him artistically and technically. It must be inspiring to restore and publish the work of a prominent poet dealing with the same kinds of issues over 20 years ago, because it gives depth to a poetic scene that many contemporary readers consider to be rather young.

Two recent tributes by Andrews are strong indicators of his current direction of reconnecting with the past as he shapes his future direction as poet and programmer. The first is his publication of his mentor and friend’s Joseph Keppler’s visual poems from “The First Remainder Series” of *Poets.Painters.Composers.Critics.Sculptors.Slaves,* published in print in 2007 and in Vispo.com in 2008 (vispo.com/guests/keppler). These visual poems by Keppler weren’t really designed for Web publication by Andrews, yet they lent themselves so well for it. They are a testament to the ideas they have been sharing over the past twenty years. The visual poems are so minimalist in approach that they achieve a conceptual elegance that Andrews explores in his own works, even though superficially Andrews’ works seem to be visually busier than Keppler’s.

The other tribute is a collaboration between Margareta Waterman and Jim Andrews, titled *F8MW9* (2008), in which he designed an interface to shape a series of glyphs and audio recordings of her sound poems.

> There is hardly any English in the interface, either aural or visual. But don't be confused. Feel it out. Look and listen and, when you feel like it, explore the interface: click around and see what the buttons and spin controls and sliders do. They don't really need any explanation if you play with them (Andrews, “F8MW9 [e-mail]”).

This work is pure visual and aural language that needs no meaning or explanation except that it is beautiful: and Andrews has kept the same spirit in the interface design. There is no help menu or help ghost. This is a work that leads its audiences to drift along with it, to play creatively with it through its interfaces, much like *Nio.* Waterman and Andrews have been friends since they met in Seattle in 1989, and Andrews’ choice to develop this collaboration testifies to her impact on Andrews’ formation as a poet.

The main reason I have not chosen to analyze these “tributes” in the amount of detail I have given other works in this chapter is that I wished to focus on works written entirely by Jim Andrews, rather than on collaborations, because I could reach a “purer” sense of what his poetry was all about. Mapping his poetics, however, does lead one to the edges of what his poetry is: and that is where collaboration can help one fine-tune some boundaries. Seeing the works that emerge between Jim Andrews and the work of Joseph Keppler, bpNichol, and Margareta Waterman helps us understand the affinities between them, but also where Andrews will not follow.

There are a number of unfinished works that this study will not explore, such as *dbCinema* and *Jig-Sound*, because they are works in progress. They are still under development—conceptually, artistically, poetically—and I feel that a critical intervention at such a delicate time could influence them in ways that could prove detrimental to the artistic vision that got them started in the first place. Besides, I relish the prospect of approaching those works from the perspective gained from this study, once they are completed.

This chapter provided a detailed and insightful look into important facets of the work of a poet whose formation is very much in tune with the development of a new frontier in poetic production known as electronic poetry. Andrews' poetics of the visual, musical, linguistic, and behavioral aspects of language should be clear by now, as is his commitment to repurposing functional interfaces (like the menu) to create artistic experiences for his readers. An implicit goal of this chapter was to model a Formalist approach to media-specific analysis, as informed by biographical and technical details.

The next (and final) chapter will examine the "*Arteroids* Development Folder" a treasure chest of source materials for one of Andrews' most significant projects: *Arteroids*.

Chapter 4: Mining the *Arteroids*

> So part of the 'confrontation' of *Arteroids* is between poetry and new media, poetry and entertainment, poetry and popular culture, poetry and programming, poetry and visual art, art and game. Writers realize, in their confrontation with the page and with language, that they need to understand their medium or it will have its way with them. When writers move to the Web and/or the Net (which includes email etc), they often do not acknowledge that the change in media has consequences for their work, how it is distributed, read, contextualized, and understood. What I have been trying to do for about thirteen years is develop as a writer in the multimedia soup that is computer-mediated writing. *Arteroids* rises from the bog and hunkers toward poetry ("Poetry, Arteroids, and the Flaw").

The previous chapter analyzed key products of Jim Andrews' exploration of the computer as a medium for artistic expression. While those earlier works focused on one or two aspects of computer mediation: visual, aural, and behavioral components, his more recent work with Director integrates these media to achieve his artistic goals. *Arteroids*, like *Nio* before it, is a milestone in Andrews' artistic development because of its ambition and complexity both as a work of electronic poetry and as a work of programming. More than any of his work to date, *Arteroids* seeks to "bring it all together"—vispo, vismu, and interactivity—in a work that references the most native genre in computerized entertainment: the videogame.[92]

[92] Games have been present from the very beginnings of the personal computer because they entered popular culture in the 1970s with hardware designed for it:

The complexity arises from the conceptual blend of schemas embodied by *Arteroids*. The areas of “confrontations” that Andrews lists in the excerpt above serve as frames of reference to help us map *Arteroids* as an artistic creation. I believe Andrews places the word confrontation in quotation marks in order to soften the hostile connotations of the term and emphasize the coming together of the terms as they “face” each other. These kinds of confrontations are creative for Andrews, exploding binary oppositions in a deconstructive work that questions the ideologies that establish them as oppositional in the first place.

Jim Andrews has been facing these oppositions from the very beginnings of his career, and he has been doing so artistically. From the traditional education in poetry he obtained in college, (as exemplified by the work of Wallace Stevens and William Carlos Williams) Andrews learned a concept of poetry that was traditional as far as its use of the conventions of print and reading aloud. When he started with his radio show, *Fine Lines*, this notion of page-and-voice based poetry came face to face with the medium of audio recording and its (page-less) materiality as influenced by MacLuhan, Burroughs, and Gregory Whitehead. The result from this “confrontation” was reinventing the radio show into *?Frame?* and writing a series of audio poems, such as “Woork of Aart” and “Poetry Craft.”

His next major creative confrontation was between poetry and visual art, combined with the medium of the computer as a tool for creating and manipulating images, the results of which are discussed in detail in Chapter 3, in the section titled

arcade video games and video game consoles such as Atari. Computer games predate both, but not as an entertainment industry, and text-based games dominated the early personal computer gaming market.

"The Electric Pen." When combined with the new media and the HTML programming that launched the World Wide Web, the two previous types of poetic engagements found a means of publication in his Web site Vispo.com, but it wasn't until he started using DHTML that Andrews really placed poetry in "confrontation" with programming, the results of which are discussed in Chapter 3, section 2: DHTML Dances. His move to Macromedia Director and programming in Lingo made the integration of poetry with its visual, sound, and programming components possible, resulting in several works that led him to a successfully integrated one, titled "Nio."

"Nio" is a poem, musical instrument, and toy, placed before its audience to read it, play it, play with it, to explore it and create with it. As such, it developed some questions for Andrews that he tried to answer through "*Arteroids*" by taking the concept of play and channeling into the more formal experience of the game.

1. What is the shape of poetry created (written, drawn, recorded, programmed) for publication in networked personal computers?
2. How does an audience receive (read, listen to, look at, play with) such poetry and to what extent can Andrews build upon or shape its expectations?

The combination of poetry and video game is a major boundary crossing and one that has only recently been explored.

On the other hand, poetry and games have had a long history, provided one include play in the definition of a game. The constraints created by OULIPO group could be considered as poetry games, for example the N+7 poem (in which someone replaces each noun in a poem with the 7th word following it in a dictionary), the palindrome, the lipoem and other playful constraints that lead to poetic creativity. Surrealism and Dada both entered chance operations into poetic creation, as was the case with the original cut-up poems and the exquisite corpse. Creating constraints or playful conditions for writing poetry is a very ancient practice, and could be traced back to traditional poetic forms, such as the sonnet, villanelle, and sestina.

Poetry created by whimsical constraints weren't just used for free play: the competitiveness of a game can also be attributed to poets who used poetry to outwit their rivals. For instance, the well documented rivalry between Sir Walter Raleigh and Christopher Marlowe manifested itself in the latter's "The Passionate Shepherd to His Love" and the point-by-point rebuttal by the former with "The Nymph's Reply to the Shepherd." A case could be made for poetry as one of the oldest forms of entertainment in both literate and pre-literate cultures. One might go as far back as Anglo-Saxon riddles to see how they could be used as a game, as imagined by J. R. R. Tolkien in *The Hobbit* when Gollum and Bilbo have a life-or-death riddle contest in the depths of a mountain. Some poetry games that survive to this day are based poetic or song improvisation, such as the Puerto Rican "trova" competitions, in which contestants draw a word from a hat and have to improvise a highly structured song based on that word. Rappers do "battles" in which they have to outwit and outperform their competitors, as can be seen in the Marshall Mather (Eminem) 2002 movie *8 Mile*.

As must be clear by now, poetry and games have enjoyed a productive connection for centuries, but none of these examples account for the paradigm shift resulting from the combination of videogames and poetry. Part of the reason for this is that most videogames place their emphasis on graphics and sound, but not on language. If verbal language was an important part of videogames, then some playfulness with language would be inevitable and the connection might have emerged sooner, but since is not the case, videogames have developed along very different lines and developed their own genres (such as the first-person shooter, the multiplayer duel, the strategic wargame, etc.). Bringing poetry and videogames together has therefore resulted in unusual combinations.

This combination isn't unique to Andrews. The Fall 2003 issue of *Poems that Go* was dedicated to "literary games," featuring *Arteroids* (version 2.5), Natalie Bookchin's adaptation of the Borges story "The Intruder," "Nine" by Jason E. Lewis, "Bad Machine," by Dan Shiovitz, and an introductory essay by Nick Montfort. He concludes that,

> The games in this issue, drawing on the tradition of computer and video games in various ways, provide a more certain proof that the literary game can do the serious, hard work of both literature and gaming, and suggest several ways in which different aspects of a literary game can function effectively together ("Literary Games").

In response to Andrews' characteristically humble statement that "*Arteroids* rises from the bog and hunkers toward poetry," (like some sort of swamp monster), I would have to agree with Montfort that *Arteroids* does the "hard work of both" a poem and a game. It is a work that is very consistent with Andrews' poetics and has received international critical attention through articles, reviews, interviews and translation to Portugese. In short, *Arteroids* is a significant contribution to the development of electronic poetry and poetics because it successfully integrates so many of the frames of reference that inform writing in digital media.

Jim Andrews started working on his literary computer game *Arteroids* in 2000 when his work on *Nio* helped him receive funding from the Canada Council for the Arts. For the next two years, he worked on *Arteroids,* publishing numerous versions during that time in his Web site and discussing them in Webartery.[93] The earliest versions were titled Web*Arteroids*, preparing him to publish *Arteroids* 1.0 in 2002. He published version 2.02 to participate in the Augustart show in New York City (August 24 to September 2, 2002) and published version 2.5 in the Fall of 2003 issue of the electronic poetry magazine *Poems that Go*. The most recent version (3.11) was published in Vispo.com in August of 2006, but Andrews chose to keep version 2.5 available, an indication that version 2.5 still captures his intentions for the piece.[94]

It is well known to textual critics that authorial intentions vary over time, and Andrews is no exception. For example, in an unpublished document in the *Arteroids* Development Folder titled "For the Judge," Andrews describes a direction for future versions of the work that haven't been developed, or have been developed differently from what is there described:

[93] Webartery is a Yahoo group he created with other writers of electronic literature, such as Mez, Alan Sondheim, Thomas Bell, Millie Niss, David Knoebel, and many more, as "a serious forum for discussion of Web art both from an artistic and critical point of view" (Andrews, Jim, "webartery : Message: Re: welcoming Patrick-Henri Burgaud"). This forum is an invaluable source of information on *Arteroids*, because it was the primary space for Andrews to share drafts of the work and bounce ideas off the group. He also received quite a bit of feedback, so it is an essential resource for the critical study of this e-poem.

[94] It is also to protect the texts people wrote and saved for Canto 2, because it wouldn't be compatible with version 3.11 (Andrews "Re: Back in Business")

> Version 3.0 will also introduce different types of *Arteroids*. Ones that shoot back at you and require more firepower to destroy. Graphical *Arteroids*. I have developed some political *Arteroids* at
>
> www.webartery.com/temp/blair/speedtest.htm
>
> www.webartery.com/temp/bush/speedtest.htm
>
> www.webartery.com/temp/chretien/Burroughs.html
>
> www.webartery.com/temp/hussein/speedtest.htm
>
> www.webartery.com/temp/binladen/Burroughs.html
>
> These guys fire at you when they turn full profile. I'm not sure if I'll end up using them or others.
>
> Version 3.0 will also store scores on the server.
>
> Version 4.0 will introduce a story. At certain points, you'll be whisked away into the story which will be about a man who is worried about his own rage and what he is going to do to the world. And it will also be about an acquaintance of his who also is worried about what this guy is going to do. The guy is developing a shoot-em-up computer game.
>
> Hopefully this will be a way into exploring violence in computer games and in the world.
>
> Version 5 will use the Macromedia Multi-User server and allow people to play one another at *Arteroids*. And that is where I will stop.

For now, Andrews has stopped developing *Arteroids* with version 3.11 (which is different from what is described in the 2002 document) and has moved on to other projects. Yet this excerpt suggests that the conceptual development of the *Arteroids* is much larger than what one can sense by reading/playing the versions published by Andrews in Vispo.com. Reading the Webartery postings by Andrews as he was developing *Arteroids* reveals a great number of ideas that he was considering, revising, getting feedback on, adopting, and discarding. As a matter of fact, the three versions currently published are merely the tip of the proverbial iceberg, because there are 82 different versions of *Arteroids*, many of which were developed under the working title *WebArteroids*. There are also numerous essays, online forum postings, e-mails, documents, sound, image, and other files, all of which present a complex matrix of developing intentions and conceptualization of the work known as *Arteroids*.

None of this information has been published, and I am aware of it because Jim Andrews has been generous enough to share the "*Arteroids* Development Folder" with me, as well as notebook full of notes, diagrams, graphs, and ideas that document his work in *Nio*, *Arteroids*, *Jig-Sound*, *DBcinema* and in other recent projects. These materials are a gold mine of information for a scholar interested in studying *Arteroids* in depth as a work of e-literature, as a first generation electronic object, as a computer game, and as a record of an artist's work with programmable media, among others.

This chapter will make a case for the importance of these materials and how they enrich the experience of *Arteroids*, justifying their use for the creation of a critical edition of *Arteroids*. The argument will be developed along two main lines of critical inquiry: it will employ the bibliographic theories of Jerome McGann, Peter Shillingsburg, G. Thomas Tanselle, and John Bryant, among others to formulate my editorial approach towards these materials and it will be informed by the work of

Mathew Kirschenbaum and N. Katherine Hayles on electronic textuality. The discussion will highlight key moments in the development of *Arteroids* supplementing them with documents from the *Arteroids* Archive and Webartery—a Yahoo Group in which Andrews expressed, discussed, and to a certain extent discovered the direction he conceptualized for the work.

Why focus on all these unpublished drafts and versions? G. Thomas Tanselle offers an elegant justification for capturing the history of a work in a critical edition.

> Approaching every human creation with an understanding of its textual history, seen against the panorama of all other textual histories, helps us to appreciate the humanity movingly embedded in each version of a work and to enjoy the hard-won accomplishment represented there. (Tanselle)

John Bryant considers this textual history a record of the fluidity of texts. From the outset, the fluidity of intention and design in *Arteroids* is evident in its early draft versions titled *Webarteroids*, and it is this group of versions that this chapter will focus on.

Before embarking upon a project of mining the *Arteroids* Development Folder for data that may be fruitful for criticism, it is important to assess the critical tools one will use to perform such data mining. For instance, why go through the exercise of descriptive and analytical bibliography? Fredson Bowers delineates a clear path for the first step of this inquiry.

> The concern of the descriptive bibliographer is to examine every available copy of an edition of a book in order to describe in bibliographical terms the characteristics of an ideal copy of this edition, to distinguish between issues

> and variants of the edition, to explain and describe the printing and textual history of the edition, and finally to arrange it in a correct and logical relationship to other editions (6).

Arteroids has been published in a number of Web sites in addition to Vispo.com, and each publication contextualizes the work as it suits its needs. For instance, the version published in *Poems that Go* (2.5) is part of an issue dedicated to literary games. It contains an essay by Andrews, "*Arteroids*, Poetry, and the Flaw," which captures Jim Andrews' conceptualization of the piece at the time it was published (2003) and in the context of the collection. The same version of the poem (with a revised version of the essay) is published in Vispo.com, along with version 3.11. Each publication has its own history, context, accompanying materials, and other distinguishing factors: all of which are enough to consider them as discrete editions of the work. These are socialized texts, as Jerome McGann established in his writings, because they have entered a social contract that shaped the final product. The contents of the *Arteroids* Archive are drafts, versions, raw materials, writing about the work, among other things, but they have not necessarily been prepared for an audience (also known as edited). Some of these versions were shared with the Webartery group for feedback, testing, and discussion, but are no longer available online—they can be considered to be "circulating drafts," using Bryant's terminology. With access to these materials, one could begin the editorial work of describing and analyzing them in order to supplement and further contextualize the published editions of the work.

Applying "bibliographical terms" to electronic objects can be challenging, because the materiality of printed objects leads to different terminology than what they require, as Kirschenbaum points out in “Editing the Interface: Textual Studies and First Generation Electronic Objects.”

> Significantly, a bibliographical/textual approach calls upon us to emphasize precisely those aspects of electronic textuality that have thus far been neglected in the critical writing about the medium: platform, interface, data standards, file formats, operating systems, versions and distributions of code, patches, ports, and so forth. For that's the stuff electronic texts are made of.

So what is *Arteroids* made of? It was authored with a variety of tools, but the central one was Macromedia (now Adobe) Director—a proprietary multimedia authoring software that uses a high-level programming language called Lingo. Because it is proprietary, one can only experience the presentation layer of the work, without access to its data layer, that is, the materials and source code that go into producing it. And that is fine for most readers, but for those interested in a textual approach to this poetic game, those materials are denied.

This is another contribution this study can make to those interested in exploring the textual materiality of *Arteroids*: providing insight into the contents of the *Arteroids* Development Folder. Some of the materials extracted from the source code and published in this study are:

- A listing of all the versions of *Arteroids* contained in the Development Folder, with “date modified” metadata and file sizes (Appendix A).
- The linguistic texts of *Arteroids*: Canto 1 & 2 Green and Blue texts, Winning Notices and Death Notices (Appendix B).

- A complete listing of 1331 files, directories, and subdirectories in the *Arteroids* Development Folder (Appendix C).
- A compressed folder with files from the Arteroids Development Folder discussed or alluded to in this chapter (Appendix D.)

These materials can inform readings of the work, allow for the study of its source code and programming architecture, and can eventually lead to a production of a critical edition or fluid text edition of the work.

A bibliographical/textual approach does raise challenges. As established in Chapter 2, this approach is built upon a vocabulary and set of concepts developed for manuscript and print, which becomes problematic when applied directly to electronic texts. One area that requires attention is the numbering conventions for different software versions and releases. For example, the numbers next to a software title (such as Firefox 3.6.3) correspond to at most four levels of versioning:

- Major: The major number is the first integer in the version string (e.g., v1.2.3). Changes in the major number typically indicate a significant change in the code base and/or end-user functionality. The major number is always included in the version number.
- Minor: The minor number is the second integer in the version string (e.g., v1.2.3). Changes in the minor number typically indicate a incremental change in the code base and/or end-user functionality. The minor number is always included in the version number.
- Release: The release number is the third integer in the version string (e.g., v1.2.3). Changes in the release number typically indicate a bug fix in the

code base and/or end-user functionality. If the release number is 0, it is omitted from the version number (e.g., v1.2 has a release number of 0).

- Quantifier: Open MPI version numbers sometimes have an arbitrary string affixed to the end of the version number. Common strings include: aX (for alpha versions), bX (for beta versions), and rcX (for release candidate versions). (Open MPI: Version Number Methodology)

The Open MPI Project's methodology is a good example of versioning and release conventions used in programming. The difference between the major, minor, and release numbers could be considered analogous to differences between versions, editions, and errata inserts in the world of print. How do these correlate to notions of work, version, text, and document as described in the field of textual studies? If we were to use Tanselle's notions of the "texts of documents" versus "the texts of works," a bug fix documented in the release number would correspond to a correction of an error in the "text of the document," while major and minor releases would represent versions and editions of the work, each one adding to "the text of the work."

These correlations are important because the world of computer software the release numbers are documented in the publication, along with mechanisms that provide and facilitate updates, patches, upgrades, and backwards-compatibility for those who choose to not update. The world of print has long sought to elide versions in order to create the illusion of a single, unitary, authoritative work. The exception is the practice of creating critical editions, such as variorum editions, which foreground the variations in different sources and reveal the versions within the text. Recent tendencies in bibliography and textual studies seek to foreground these differences, as is the case with John Bryant's Fluid Text edition of *Typee*, and with projects like the *Dickinson Electronic Archives* and the *William Blake Archives*, which provide facsimiles of manuscripts and prints of illuminated texts, respectively. When these two worlds collide in electronic textuality, however, problems arise due to differences between source code and displayed documents, and the critical tools developed for each distinct media.

For example, what is considered textual variation in the world of print is primarily variation in linguistic texts: not variations in source code. Variations in visual formatting (fonts, sizes, etc.) would be considered by Tanselle to be documentary variations and wouldn't contain information worth preserving or documenting. In electronic texts, variations in textual behavior would need to be documented, whether through a description of the behavioral change from version to version or by documenting differences in the source code.[95]

An example from the Arteroids Development Folder should underscore this point. The linguistic text of Webarteroids 1 through 5 is the first stanza of the nursery rhyme "Mary Had a Little Lamb," after which it starts to be replaced by versions of the original text compiled in Appendix A of this study. In Webarteroids 1 through 3 this text appears, a few words at a time, in green texts that float across the screen at

[95] The very use of the term "version" has a similar meaning but different connotations in the worlds of print and programming. The main difference is that versions are commonplace in programming, because documenting the progress of developing code and testing it out is an integral part of the drafting and release process. The equivalent in print is drafting and publishing different editions of a work, something that historically implies fewer versions and variations. As discussed throughout this study, the difference between print and electronic texts is one of scale, time frame, and convention.

variable speeds in random linear trajectories originating from the top right hand corner of the screen. Webarteroids 4 introduces a variation which becomes a major line of development in *Arteroids*: some of the text of the nursery rhyme is blue, emerges from the top left corner of the screen, and follows the word "poetry" (which the player controls) throughout the screen—much like the little lamb followed Mary.[96] From Webarteroids 6 onwards, the blue texts are going to be different from the green ones, but the behavior remains basically the same.

This is where a linguistic text approach would end, but here's a detail that is documented in a programming approach, and should be addressed when exploring electronic texts: between Webarteroids 4 and 6, the formula which determines how the blue text follows the word "poetry" changes. In Webarteroids 4, the blue text moves horizontally from left to right to match the position of the word "poetry" and then it descends vertically to attempt collision, giving the illusion that the word is dropping from above. Webarteroids 5 changes the horizontal movement to one shallowly diagonal towards the word "poetry," an angle that becomes sharper and more direct in Webarteroids 6, and randomly variant in Webarteroids 9.

How does one read the way the blue texts follow the word "Poetry" text? The left to right, top to bottom initial motion is evokes the most common reading convention in Western Culture, a point that is lost when the algorithm was modified

[96] Throughout my study of Arteroids, starting with the published 2.5 and 3.11 versions and working backwards when I got the Arteroids Development Folder, I considered the use of the first stanza of "Mary Had a Little Lamb" as a textual placeholder during the initial programming stage of the poem. Now I see that either the choice was not as arbitrary as I thought, or that the initially arbitrary choice led Andrews to an important textual behavior—that of the text following the player's "ship" throughout the screen.

for the text to make a beeline towards the word "poetry." This tendency is reminiscent of the way *Seattle Drift* begins looking like a traditional poem, but moves away from that "scene" to embrace the scene of e-poetry by creating its potentially infinite space. Here is another example of the "confrontations" Andrews sees between poetry and games, in which the behavior of the blue texts is perhaps inspired by a poem, and the animation begins very strongly aligned to reading, and reduces this effect to become more game like. Then again, maybe Andrews was simply tweaking the chasing algorithm from version to version to achieve a vision he had all along.

Another consideration is that these revisions are geared towards a different reader: the computer and software that parses and executes the source code. In other words, these revisions are not for the reader to interpret as much as they are for the logical layer of the computer to interpret and execute.

The point is that the behaviors of electronic texts need to be taken into account when editing them because they can provide insight into the work. Tracking variation in different versions, as suggested by John Bryant's fluid text approach, by observing changes in textual behavior can lead to a more nuanced understanding of the function of such behaviors in the work, as well as highlight the creative process. John Bryant's fluid text approach provides a framework to track differences in versions of texts, but the method has yet to be applied to a first generation electronic object, and it doesn't account for fluidity in textual behavior and source code, because it was designed for linguistic texts in manuscript and print. This raises a several questions about the pragmatics of applying this approach to electronic texts, particularly when seen in the framework of data layer, logical layer, and presentation layer:

- How important is it to track the fluidity of source code, the documents from which the presentation layer of electronic texts is generated?
- Does one document the source code (data layer) or the presentation layer of the text (what we read on the screen)?
- How much variation can be attributed to the computer hardware and software (the logical layer)?
- How can one go about creating an edition of an electronic text that can incorporate significant versions in how it runs?

Part of the answer lies with tools created for programmers to be able to document and track variations in executable code: two of the best known ones are CVS (Concurrent Versions System) and Apache Subversion. These open source version tracking systems are used to track the differences between versions of the same document and are especially useful in collaborative environments where individuals and teams develop and modify code.[97] If implemented from the beginning of a programming project, these systems can automatically track all changes and number the versions according to the conventions described earlier in this chapter. These programs don't work on already completed programs, however, as is the case with Arteroids. Another limitation of the software is that it doesn't function with proprietary programming languages, as is the case with Macromedia/Adobe Director Lingo.

[97] This kind of feature is becoming more commonly used outside of programming circles, as is the case with Google Documents, which keeps all saved versions of a document and allows the user to compare, accept, and/or reject changes within versions. Word processors such as Microsoft Word and Open Office Writer have tools to track changes, but once the changes become accepted or rejected by a reviewer, they are absorbed in the document and are no longer retrievable.

Another part of the answer can be seen in the example of *The Agrippa Files*, a scholarly Web site dedicated to “one of the last great memes of the pre-Mosaic Internet” (Kirschenbaum *Mechanisms*). This collaborative project was created by the Transcriptions Project in UC Santa Barbara, and is best described in the words of its editorial board.

> *Agrippa (a book of the dead)* appeared in 1992 as a collaboration between artist Dennis Ashbaugh, author William Gibson, and publisher Kevin Begos, Jr. *The Agrippa Files* is a scholarly site that presents selected pages from the original art book; a unique archive of materials dating from the book’s creation and early reception; an emulation of Gibson’s included poem in its original born-and-die-digital form (it ran from a diskette once before encrypting itself into oblivion); a simulation of what the book’s intended “fading images” might have looked like; a video of the 1992 “transmission” of the work; a “virtual lightbox” for comparing and studying pages; full-text scholarly essays and interviews; an annotated bibliography of scholarship, press coverage, interviews, and other material; a detailed bibliographic description of the book; and a discussion forum (Liu, et. al. “The Agrippa Files”).

Agrippa is a fascinating work, largely because it was originally published through electronic and physical objects that were designed to self destruct, but has proven to have a life beyond the intended design thanks to the intervention of a hacker named

Templar.[98] The work of descriptive bibliography, forensics, and archiving carried out in The *Agrippa* Files is a model of how to create an online resource for scholarship on a first generation electronic object, but it has its limitations.

The first limitation is that despite the exquisite degree of details with which the objects are described, analyzed, and made available to the public—including the recent addition of a disk image that one can copy and run in an emulator for a less mediated experience of the electronic text in action—the project has drawn the line at a point that I can't help but simultaneously respect and criticize:

> We finally want to emphasize that we did not in any way "hack" the *Agrippa* program to accomplish what we describe here. This is important not mainly for legal or ethical cover, but because the language of hacking would obscure what are in fact well-established, open procedures in the digital preservation and forensics community. Hacking has had a colorful place in *Agrippa's* lore. Indeed, I would hold that Templar and his colleagues can indeed claim credit for a "hack" of sorts—albeit one that was not fundamentally computational in nature—when they were able to transcribe Gibson's poem from their bootleg video. But the term "hacking" would lend our work an aura of derring-do that is both deceptive and distracting. (Kirschenbaum, et al. "No Round Trip")

The point the team of The *Agrippa* Files project makes is valid, and places emphasis on what is perhaps the most valuable contribution of the archival project: using "well established, open procedures in the digital preservation and forensics community."

[98] Matthew Kirschenbaum dedicates a chapter to this famous e-poem and artist book in his *Mechanisms*, where a detailed narrative of how the text was "hacked" and shared on the Internet on the very same day it was publicly screened.

Indeed they may have shed light upon the most insightful part of the materials that inform this tantalizingly closed work of electronic literature, but we have yet to see the source code to understand the inner workings of *Agrippa*, and it seems like the only way to do so, short of with the help of Gibson, Ashbaugh, or the anonymous programmer who helped him, is through hacking.

The term “hacking” is loaded with connotations of transgression, illegal and unethical practices, but also has air of sexiness largely owed to Gibson’s cyberpunk novels, whose hacker protagonists help free information from the control of corporations to subvert their questionable agendas. Gibson and Ashbaugh’s work was designed with a full awareness of the challenge it represented to the hacker community, and was released with the expectation of some sort of a hack occurring—and it did, but in a limited fashion. What was “hacked” was the linguistic text and the presentation layer of the text through video recording, perpetuating its transmission beyond the event of its presentation. The materials made available in The *Agrippa* Files (with the consent of Gibson, Ashbaugh, Templar, and others), especially the disk image, are more sophisticated reproduction of the materials, but the source materials are still tightly shut within the black box of its encryption.

The time has come to open that box and see what lies within, the mechanisms at work that we may not be aware of by simply seeing the presentation layer. What documentation, insight, or even “goodies” can be reached by taking up Gibson’s computational challenge to the hacker community and find a way to break the encryption and access its source code? Curiosity aside, revealing the source code would add a whole new layer to interpret: it would reveal its mechanisms, algorithms, variables, references and borrowings from other programming codes, inactive code that may reveal other intentions, and much more.

Perhaps this could be considered to go counter to the author's intent—and in this case the term "author" is more fitting than "writer" to refer to the collaborative team that created and exercises legal control over the materials—but why go through such pains to create and publicize such a challenge for the hacker portion of Gibson's readership? Perhaps hacking *Agrippa* is computationally impractical or even impossible without the encryption key, which may be in control of any, all, or none of the people involved in creating the electronic version of *Agrippa*. In any case, the potential rewards are worth the effort, and perhaps publishing the disk image in The *Agrippa* Files could be interpreted as an invitation to the community to attempt to crack the encryption and reach the source code. This would be a valuable contribution to the study of this important early e-poem.

The Arteroids Development Folder

Thanks to Jim Andrews' generosity and help with this study, there is no need to hack *Arteroids*, to get to the source code. The source code for all the versions is available in the *Arteroids* Development Folder, as are many of the materials that went into *Arteroids*: images, sound files, and more. Better yet, since Andrews is a programmer, he knows the conventions of software version numbering and named and numbered the files in a way that makes it easy to track the versions of the work.

The rest of this chapter will describe four versions of *Arteroids*, two of which are only available in the Archive (*Arteroids* 1.0 was published in issue 11 of *The*

Remedi Project (http://www.theremediproject.com/indexSetHigh.html) but the project ended in 2002 and *Arteroids* no longer loads successfully in the site), and the two versions currently published in Vispo.com. The first two versions could be considered drafts because they were not even titled *Arteroids:* “shipshoot8” and “Web*Arteroids*.” I will contextualize these drafts with some of Andrews’ postings to the Webartery discussion group, establishing how he conceived the project, and pointing out ideas that may be useful for interpreting the work. The study will then provide a brief comparison and contrast of three published versions of *Arteroids*: 1.0, 2.5, and 3.11, pointing out key variations and developments.

What this chapter will not do is provide detailed readings of the texts, as was done in the previous chapter. Its goal is to make a case for the importance of studying the files in the *Arteroids* Development Folder and justify the creation of a critical edition of the work by delineating some of its characteristics, challenges, and editorial principles. This chapter will conclude the study by pointing a direction for future research on Andrews’ work.

WebArteroids: Preparation and Conceptualization

While Jim Andrews was exploring the possibilities offered by Macromedia Director 8 and its programming language, Lingo, he found a sketch of the 1979 Atari game *Asteroids* by Ian Clay which had been posted on Director Web on Feb 5, 2001 (“New Is Old”).

Asteroids is a classic from the arcade video era and it is consists of basic elements: a ship in space avoiding and shooting at floating asteroids, the large ones explode into smaller asteroids, each with different speeds and trajectories. The smallest asteroids disintegrate when fired upon. Occasionally a flying saucer appears and fires randomly into the screenspace while moving in a linear trajectory across the screen. Upon clearing the screen of asteroids, a new and more difficult level begins, with more asteroids moving across the screen. The game ends when the player's ship has been impacted by asteroids or flying saucer fire more times than the ships accumulated in the game (see figure 31).

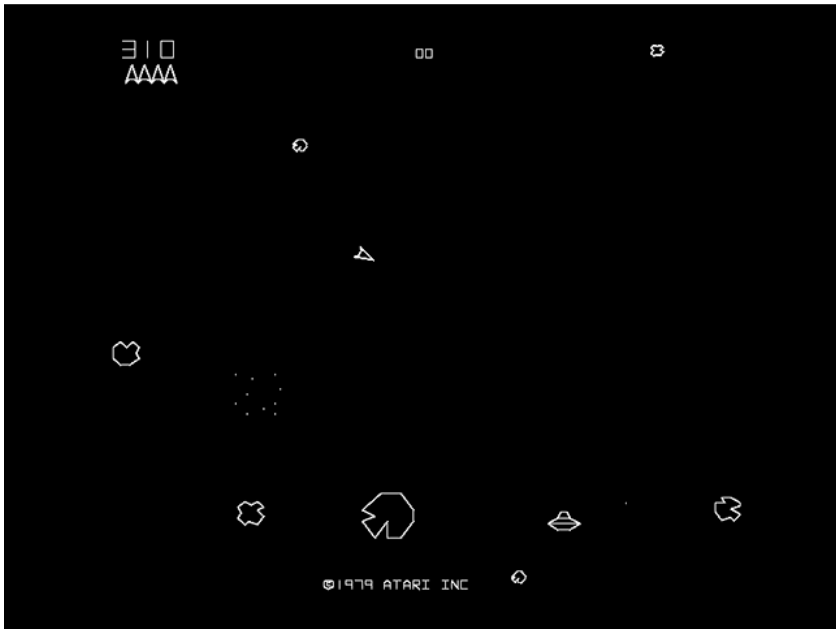


Figure 29: Asteroids screenshot

This screenshot contains all the elements described in the game. You can see the three sizes of asteroids, an explosion, a flying saucer and a shot close to it. The ship is the triangular shaped object near the center of the screen. The reserve ships (or "lives") are lined up beneath the score on the top left corner of the screen. One element worth mentioning is that the space of the game is what is known as a "wraparound" space, which means that objects that go off one edge of the screen automatically reenter the screen on the opposite edge—in other words, there is no escape. The player needs to deal with what is present on screen.[99]

Perhaps the A-shaped ship appealed to Jim Andrews' Lettristic sensibility, or some other aspect of the game captured his imagination, but inspired by the potential he saw in adapting the game for his poetic explorations, Andrews started to develop an e-poem and game he initially called *WebArteroids*. These early drafts (available in Appendix D), along with the forum discussions held in the Webartery group, are evidence of the conceptual groundwork for *Arteroids* as well as a record of Andrews' intentions for the work and are therefore a valuable resource for media-specific analysis and bibliographical study of the work.

The very first draft found in the Archive is titled "shipshoot" and it is purely a test of the game framework, consisting of two ships (the smaller ship for the user, the larger as a target), instructions displayed above, the ability to shoot and the capability to detect an impact from the shot (registered as a slight movement of the otherwise static ship) (see figure 32).

[99] *Asteroids* is based on one of the first computer games, *Spacewar,* a program developed by Steve Russell in 1961 and elaborated by Pete Sampson and Dan Edwards for release to the computing community in 1962. This game was free and open source, so subsequent hackers developed other versions or added functionality to the game (Kent 17-18). *Arteroids* is a contribution very much in the spirit of the programming community, and its lineage seems fitting for an early e-poetry game.

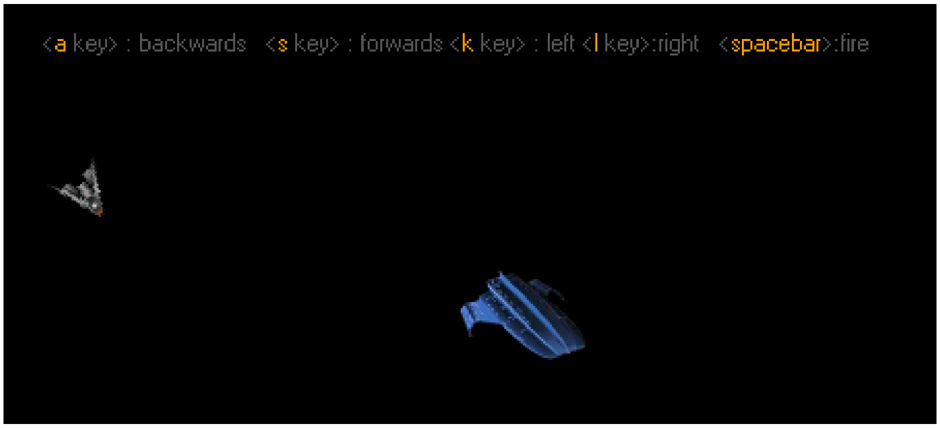

<a key> : backwards <s key> : forwards <k key> : left <l key>:right <spacebar>:fire

Figure 30: screen capture of "shipshoot"

The first draft he shared with the Webartery group is titled “shipshoot8” (see figure 2), in which the large target ship was replaced with a floating head that chases the player’s ship in an attempt to “eat” it.[100] His July 2, 2001 forum posting is the first published reference to the work that would become *Arteroids*, and it includes a link to “shipshoot8” and an invitation to the Webartery community to send pictures of their face from different angles, so he could turn them into “space monsters” (Andrews, Jim, “web*Arteroids*”). This initial concept of the poem/game pitted the player in a destroy-or-be-eaten relationship with the disembodied heads of poets, perhaps the poets who participated in the Webartery community at that time. If the work had remained as that it would have probably become an amusing game focused mostly on strengthening the online community of poets and net artists, and it wouldn’t really have such a prominent position in this study. But for Andrews, playing with this

[100] The head animation is more complex than a simple image, as described as follows by Andrews “I borrowed my friend's digital camera and then just held it at arm's length and snapped away, looking into a mirror. Deleted most of them. Ended up with 24, but so far have only used 8. Took them into PhotoPaint and removed the background, replaced it with black, and turned the photos into grayscale. Also increased the contrast to get more shadow, more of a just black/white thing, a dark thing, and made myself into a bit more of a monster than I am in some others.” (see figure 33) (“webartery : Message: RE: [webartery] webarteroids”).

interface was the beginning of a creative conceptualization that led him to the text-based version now published in several versions.

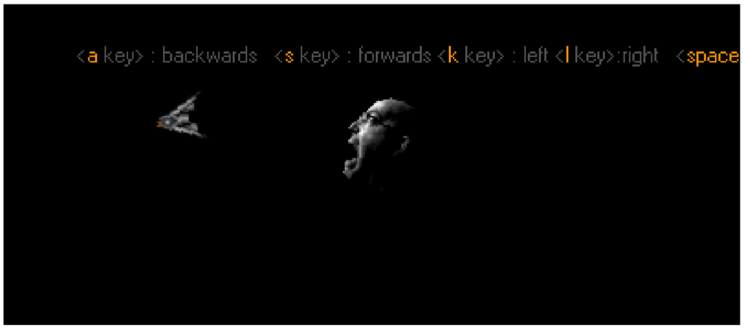

<a key> : backwards <s key> : forwards <k key> : left <l key>:right <space

Figure 31: shipshoot 8

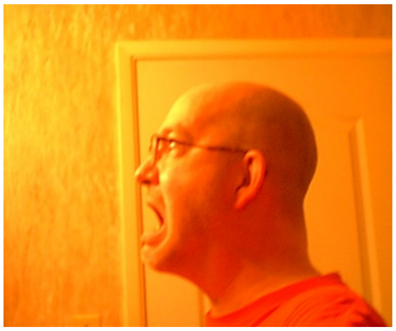


Figure 32: Image 23 in *Arteroids* Development Folder

The reception was enthusiastic and the forum postings suggest several volunteers for sending pictures for the piece, but Andrews was already uneasy with that initial concept, as evidenced by this posting the very next day.

> The poetry in this piece... where is the poetry in this piece ... ?
>
> I think it will be in the nature of the departures from Asteroids, the import of the animations and sounds... what is the player doing? Blowing up poets and/or other things also? What is the id-entity of the player? It is a ship now, but it could change through the game. And what are the poets and/or characters doing?
>
> I'm way open to suggestions here (Andrews, Jim, “more web*Arteroids*”).

And suggestions he got, such as making it a kind of magnetic poetry/asteroids combination, questions about what happens to the heads when exploded and whether they became other smaller heads, comments that dismissed the venture as a bauble, and long-rambling philosophical writings about meaning in language and poetry. As Andrews discovered the direction he wanted for *WebArteroids* and made choices that focused the project (to the point of softening the Webartery reference to rename the work *Arteroids)*, the feedback from the Webartery community became focused as well, keeping itself important as a sounding board for Andrews.[101] The discussion in the community space is also important because it led Andrews to explain the work, the directions he contemplated, his choices, and his poetics.

Two messages from July 8, 2001 are particularly useful to reconstruct Andrews' thoughts on the directions he might take in developing the work.

> I am working on one now where the id-entity is the word 'id-entity' and the 'asteroid' is a text that grows in letters as you shoot it...if you run into the text, then the text gets set back to one letter long and the text scores points against you...if you manage to shoot it enough times without running into it, thus revealing the full text, then you score points and dispatch it to hell (Andrews, Jim, "webartery : Message: RE: [webartery] web*Arteroids*").
>
> Another one of course could be where the id-entity is the word 'poetry' and the asteroids are lots of the word 'prose' and 'ad' etc.
>
> Or the id-entity is the word 'web.art' and the asteroids are 'net.art', ehehe. Or the other way around. or historicism vs web.art or whatever.
>
> Or the id-entity is a toywar figure and the asteroid is etoys...
>
> Or the id-entity is a graphic or set of animations of you and the asteroids are

[101] I suggest visiting Webartery (http://groups.yahoo.com/group/webartery) and searching the message archive with the following keywords: "asteroids," "webarteroids," and "arteroids" to access the discussion of the work in progress.

> your pet peeves (or worse)...
>
> There can be more than one 'asteroid' on the stage at a time but only one id-entity. And the behaviors of the asteroids can vary, I'm open to suggestions here. So far I can see some that follow the id-entity, some that don't but just drift. And then there's the one I mentioned where you have to shoot it several times before it expires, and each time you shoot it it changes, like if it's a text it might grow by a letter or word or shrink by
>
> a letter or word. Or if it's an animation it might display a different animation each time it's hit, and a different one yet when it is 'destroyed'.
>
> And the 'missiles' can change in their graphical and sonic nature also. Can be letters or words or some other graphic and the associated sounds (if any) can change (Andrews, Jim, "webartery : Message: RE: [webartery] web*Arteroids*").

A foundational idea discussed in these messages is the notion of a text that gradually reveals itself through game play. The linguistic content of the text is still indeterminate here, but the adversarial relation between the "id-entity" (the player's "ship") and the targets (the "asteroids") is evident, as are the militaristic undertones ("missiles," "destroyed," "dispatch it to hell," and "toywar figure"). From the outset, Andrews places the player/reader and the poem/poet on either side of this relation but had yet to decide how to explore or deconstruct that opposition.

The first version actually titled "Web*Arteroids*" was published to the Webartery group on July 11, 2001, and it was the first step in a textual path that would remain consistent to the latest version of *Arteroids* (see figure 35).

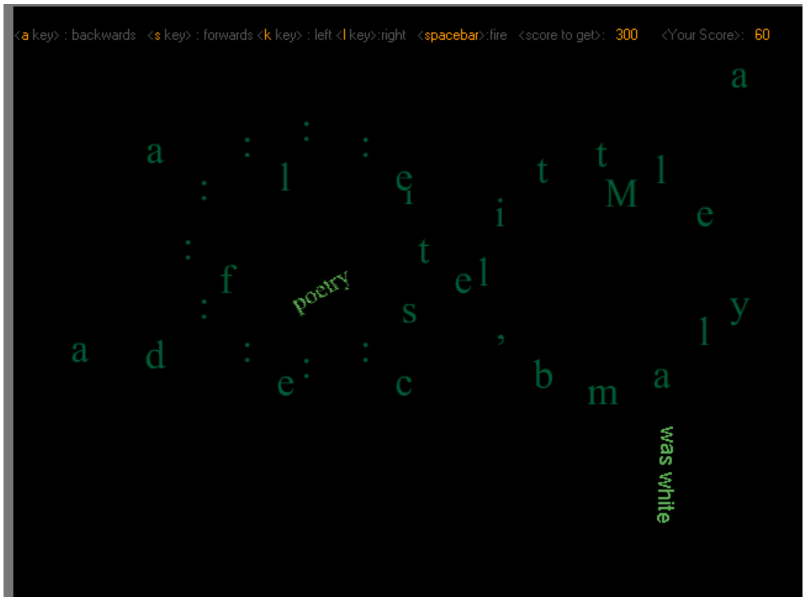

<a key> : backwards <s key> : forwards <k key> : left <l key>:right <spacebar>:fire <score to get>: 300 <Your Score>: 60
poetry
was white

Figure 33: Web*Arteroids*1

In this version, the word "Poetry" has replaced the ship (or what Andrews calls the "id-entity"), though it retains the ability to shoot at floating texts. The text of the children's song "Mary Had a Little Lamb" appears one word or phrase at a time and moves in a random linear trajectory at variable speeds until exploded by being shot. There is no negative effect from a collision beyond losing points in the overall score, so the player is indestructible, and the game's only level ends when the player has accumulated 300 points. With this version, Andrews defines the basic structure of the game/poem and needed only to develop it along the lines of game design, mechanics, text, and sound.

The following morning, July 12, 2001, Jim Andrews received notification that he had been awarded a $20,000 grant from the Canada Council's Electronic and Spoken Word program to develop *Arteroids* (Andrews, Jim, "webartery : Message: surprise"). This allowed him to continue working full time on this project all the way up to *Arteroids* 1.0, which he published in *The Remedi Project* and submitted a copy to the Canada Council along with its documented source code (See Fig. 36).[102]

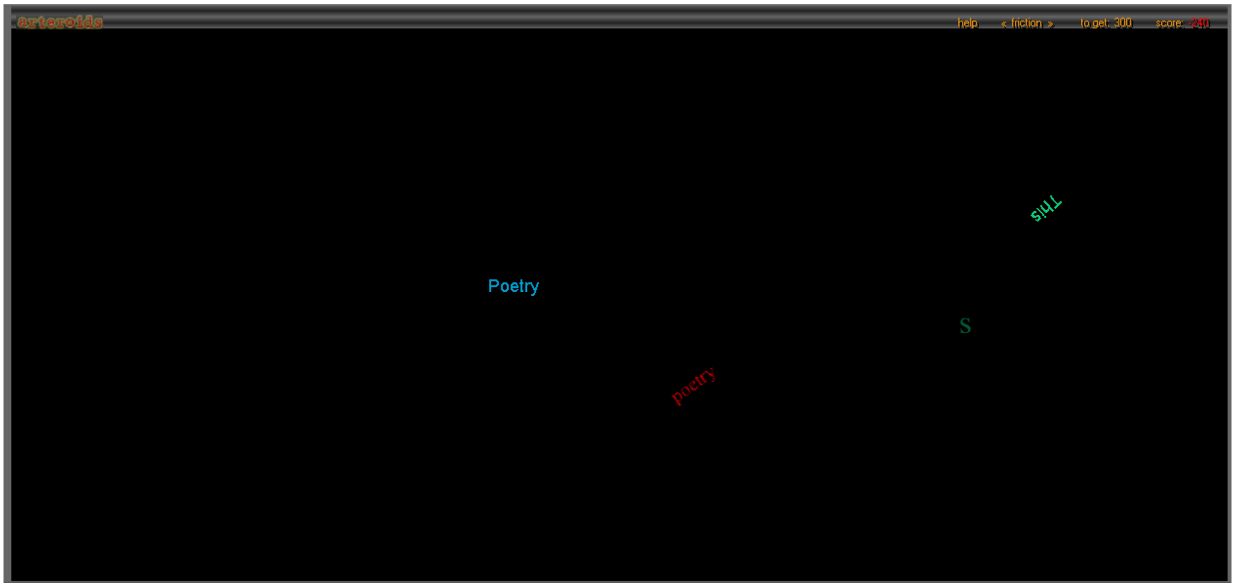


Figure 34: *Arteroids* 1.0

But there are four months of work on *WebArteroids* and conversations with the Webartery group before that led to the first officially published version of the work. The list below identifies some landmark versions in the development of the work.

- *WebArteroids4* introduces blue texts that follow the player's "id-entity."
- *WebArteroids6* gives the blue texts an independent text to display, as well as an explosion that is distinct from the text.
- *WebArteroids8* opens with a text editor which allows readers to write or copy and paste green and blue texts for the game.

[102] The file "arteroids1_for_Arts_Council" in the *Arteroids* Archive is a working copy of *Arteroids* 1.38 with the added benefit of a voice recording of Jim Andrews discussing the e-poem.

- *WebArteroids9* moves the text editor to Canto 2, reachable after reaching a score of 300 points.
- *WebArteroids25* opens with a menu which allows users to choose between Cantos 1 and 2 and displays instructions for controlling the id-entity. It also includes an original text for both the green and blue "texteroids." (see Appendix A for a complete list of texts.)

Beyond this version, the differences become more subtle, as Andrews develops the code, materials and text for a smoother, more playable experience. After publishing version 1.01, still in many ways a work in progress, the developments continue significantly. An important landmark that doesn't fall into this list because it occurs in *Arteroids* 1.38 is the addition of sound to the game.

As must be clear by now, exploring different versions of *Arteroids* can provide useful insight for those interested in studying the work, be it for analysis and interpretation, for its programming, or for the development of its concept. An insight I get from seeing the work in process is that Andrews' moved from a work that was initially concerned with icons, faces, poets, and graphical objects to interact with to a more focused engagement with language in the materiality of the digital environment as envisioned and simulated by the *Asteroids* game. Andrews was already interested in words drifting in the scene of digital media: *Arteroids* allowed him to expand on that concept, its lexicon, its simulated physics, its lexicon, its interactivity, its multimedia capabilities, and its expressive potential.

The Arteroids Archive and its Challenges

The *Arteroids* Development Folder presents significant challenges to transform it into a publishable archive. The following discussion will elaborate on these to outline some of the work that will continue beyond the scope of this study.

The first concern arises out of how to organize the materials for a chronologically organized descriptive bibliography. There are about 1331 files, directories, and subdirectories in the "*Arteroids* Development Folder," some of them with the same titles, some of which are duplicated, some compressed in folders, and so on. A challenge of organizing these materials chronologically has to do with the usefulness of the files' metadata. For instance, the "date modified" information is a good guideline, but the "date created" data is lost, because the process of transferring and decompressing the files records the present date onto that metadata. Working with the "date modified" metadata does present the problem that it records when Andrews finished working on the document, but not when he began. Fortunately, in the early stages of *Arteroids,* there were versions that were hours apart in their "date modified," so one can get to a close approximation of the time spent on each version. Another more labor intensive yet precise way to get a sense of versioning is with the documents contained within each Director file.

Each *Arteroids* Director file contains several dozen objects (image, sound, and text files, scripts, timelines, etc.) each of which contains date created (an accurate one) and date modified metadata. As Andrews developed versions of *Arteroids,* the number of objects increased, some objects became disabled within the work (but remained within the file), others were rewritten, redesigned, merged, and so on. So cataloguing each file with its multiplicity of objects is a huge job, particularly if one is interested in tracking the changes within them—which would require a high level of expertise in Director and its programming language Lingo.

Another challenge comes from the fact that Jim Andrews started with Macromedia Director 8 and upgraded versions up until reaching Adobe Director MX 2004 (version 10.1). At this time, Adobe Director is in version 11.5, which changes how it generates and interprets its code and has a new audio engine. This means that when you open an *Arteroids* file with Director 11.5, it converts it to the new code, changing it into a format that no longer runs properly, changing the date modified metadata, and providing an interface that doesn't allow the same access to the objects. In other words, you're not really looking at the same file anymore. Getting a legal old version of Director is challenging and expensive, presents potential problems with compatibility with current operating systems[103], and is no longer supported by the company.

The issue of compatibility is key, particularly because software companies rarely maintain backwards compatibility for more than ten years. One of Andrews' current projects, *Jig-Sound,* doesn't work in Director 11 because the sound engine is completely different, so he is no longer upgrading his authoring software. *Jig-Sound* still runs with the Shockwave Plugin, but how long will Adobe produce a plugin that read current and "legacy" codes? And on the academic side of things: even when one can "run" the files, accessing its source materials and code in an intelligible way (that is, with the right version of authoring software) is becoming increasingly difficult.

[103] I am running my copy in Window 7, using a Windows XP compatibility mode.

The threat of obsolescence is a tremendous motivation to create an *Arteroids* Archive that provides a more lasting and significant access to the work.

An initiative that Andrews and I will be working with is the Preserving Virtual Worlds project (http:// http://pvw.illinois.edu/pvw/) a federally funded digital preservation initiative that is interested in long term preservation of virtual spaces, such as desktop and online games. This is the kind of project that would provide an appropriate context for publication, as well as the expertise and negotiating power with Adobe, given that the works are created with proprietary software. Andrews himself has offered to participate in this initiative, so I suspect that the future of the *Arteroids* Archive will be in good hands, since Jim Andrews' programming skills and understanding of his work will be placed at the service of this preservation effort.

One thought for creating a critical edition of *Arteroids* is how to represent significant variations within a playable file. A possibility could be to have a menu in which one can switch between variant texts and behaviors to see how they play/read. The result would be a massive file, full of documented Director objects that are organized to be both human and machine readable. This kind of endeavor would require Jim Andrews' interest and collaboration, since only he has the expertise and knowledge of the way this complex e-poem is put together. A less computationally demanding possibility is to create a resource that offers a selection of versions of *Arteroids* for those interested to play, compare, and study.

Yet another possibility for future development is to port *Arteroids* to some of the newer programming languages and development platforms such as Actionscript, Flex, Air, and Flash, some of which are more open in their source codes, are more in tune with the way the Web works, and may not require proprietary plug-ins. Andrews' is currently learning how to program with these technologies and describes it as "a major retooling" and "a new beginning, of sorts" ("yo"). I am interested in seeing the work that will be inspired by these newer technologies for developing rich content on the Web.

It is fitting that this study concludes with a beginning, particularly since Andrews' poetic career and influence continue to grow in the field of electronic poetry. This study has focused its attention on the growth and development of an extraordinary poet whose formation has made him so in tune with the development of digital media and the emergence of the World Wide Web that he has been able to thrive artistically. Jim Andrews is an exemplary figure in the rise of electronic poetry and studying his work with an attention to the technologies and media that they take advantage of gives rise to issues that are both timely and enduring. I believe this is a story worth telling and a legacy worth preserving, and I am optimistic about the future of this endeavor.

What began as a relatively traditional single author study is now transforming into a collaborative effort, whether it occurs in the context of Preserving Virtual Worlds project or in some other preservation effort. The future will tell.

Appendices

Appendix A: Versions of Arteroids in the Arteroids Development Folder

Versions available in *Arteroids* Development Folder			
File Name	**Date Modified**	**File Size (KB)**	**Location**
shipshoot	2/12/2001	40	webarteroids (old version of arteroids)
shipshoot85	7/2/2001	55	
shipshoot8	7/4/2001	106	
shipshoot8text1	7/8/2001	177	
webarteroids1	7/15/2001	174	
webarteroids2	7/16/2001	288	
webarteroids3	7/16/2001	296	
webarteroids4	7/17/2001	328	
webarteroids5	7/18/2001	416	
webarteroids6	7/20/2001	409	
webarteroids7	7/22/2001	237	
webarteroids8	7/23/2001	504	
webarteroids9	7/24/2001	324	
webarteroids10	7/25/2001	569	
webarteroids11	7/25/2001	327	
webarteroids12	7/26/2001	576	
webarteroids13	7/29/2001	648	
webarteroids14	7/31/2001	615	
webarteroids15	8/1/2001	395	
webarteroids16	8/1/2001	666	
webarteroids17	8/4/2001	760	
webarteroids18	8/5/2001	753	
webarteroids19	8/5/2001	756	
webarteroids20	8/5/2001	757	
testeditor	8/11/2001	103	
webarteroids21	8/11/2001	755	
webarteroids22	8/14/2001	773	
webarteroids23	8/16/2001	846	
webarteroids24	8/17/2001	843	
webarteroids25	8/21/2001	507	
webarteroids26	9/8/2001	1062	
webarteroids27	10/4/2001	1005	
webarteroids28	10/6/2001	1013	

webarteroids29	10/10/2001	1950	
webarteroids30	10/10/2001	1477	
webarteroids31	10/12/2001	1889	
webarteroids32	10/13/2001	1964	
webarteroids33	10/15/2001	880	
webarteroids34	10/16/2001	1531	
arteroids35	10/18/2001	989	1_38
arteroids36v85	10/20/2001	1596	
arteroids1_36	10/27/2001	1731	
arteroids1_37	11/1/2001	1667	
arteroids1_38	2/9/2002	1334	
arteroids1_38	5/26/2002	3381	
arteroids1_38v2	5/30/2002	3209	1_39
arteroids1_38v3	6/28/2002	1892	
arteroids1_38v4	7/4/2002	3747	
arteroids1_38v5	7/5/2002	4147	
arteroids2_0	7/19/2002	3879	arteroids2_01.zip
arteroids2_0portugese	10/21/2002	20103	arteroids
arteroids2_0	5/28/2003	5619	
arteroids2_5	6/25/2003	5821	arteroids2_5.zip
arteroids2_5	2/13/2004	5629	arteroids
arteroids2_6	6/5/2004	6379	
arteroids2_6music3	6/6/2004	5734	
arteroids2_7MX2004	5/4/2004	5486	
arteroids25music1	6/6/2004	11147	
arteroids25music2	6/6/2004	10933	
arteroids3music3	5/30/2005	5765	
arteroids3music30	8/21/2006	5791	
arteroids3music301	8/22/2006	4086	
arteroids3music302	8/23/2006	5441	
arteroids3music303	8/24/2006	5657	
arteroids3music304	8/26/2006	5482	
arteroids3music305	8/29/2006	5510	
arteroids3music306	9/3/2006	5816	
arteroids3music307	9/6/2006	4280	
arteroids3music308	9/6/2006	5953	
arteroids3music309	9/8/2006	4398	
arteroids3music310	9/11/2006	4507	
arteroids311	9/14/2006	4523	backup
arteroids312OLD	9/15/2006	4498	arteroids
arteroids312	9/18/2006	6484	

arteroids313	9/28/2006	4490	
arteroids314	9/28/2006	4496	
arteroids315	11/12/2006	6141	
arteroids316	11/14/2006	4535	
arteroids317	6/7/2007	6547	
arteroids317white	6/5/2008	4595	
arteroids318	6/9/2008	4619	
arteroids1_for_Arts_Council	10/31/2001	56178	

Table 6: Versions of Arteroids

Appendix B: The Texts of Arteroids

Arteroids contains many texts displayed at different times during the game. This appendix contains the texts from Web*Arteroids* 1.0. Using Andrews' terminology, the "outer" texts are the words that float around and the "inner" texts are displayed when they've been shot in the game. Canto 1 is game mode and Canto 2 is play mode, and each has its own texts. Outer green texts drift linearly while outer blue texts chase the player's id-entity. The death notices are displayed after collision with an inner text and the winning notices are displayed after completing a level of game play.

Canto 1 Texts	
Outer Green Text	Inner Green Text
This	death
is	nothing
the	bravado
battle	insecurity
of	fear
poetry	status quo
against	tradition
itself	the safe
and	censorship
the	cynicism
forces	automation
of	voice problems
dullness	alienation

fear
death
nothing
bravado
insecurity
fear
status quo
tradition
the safe
censorship
cynicism
automation
voice problems
alienation
bunk
bojum
propaganda
prose
ads
dullness
bullshit
institutionalization
computers
complacency
conformity
insanity
progress
quiescence
poetry
too much pain
busyness
money
The President
the Prime Minister
the Grand Poobah
technology
grandiloquence
now
publishing
culture
you
me
frustration
sentimentality
mendacity
literalism

bunk
bojum
propaganda
prose
ads
dullness
bullshit
institutionalization
computers
complacency
conformity
insanity
progress
quiescence
poetry
too much pain
busyness
money
The President
the Prime Minister
the Grand Poobah
technology
grandiloquence
now
publishing
culture
you
me
frustration
sentimentality
mendacity
literalism
prejudice
stupidity
intelligence
war
distrust
abuse
new world order
world order
capitalism
society
scarcity
civilization
hatred
language

prejudice	training
stupidity	habit
intelligence	sadness
war	happiness
distrust	self-satisfaction
abuse	sloth
new world order	ambition
world order	good
capitalism	evil
society	coveting
scarcity	arrogance
civilization	pride
hatred	Hecuba
language	knowledge
training	ignorance
habit	::::::::::
sadness	OOOOOOOOOO
happiness	HEARTHEART
self-satisfaction	::::::::::
sloth	**********
ambition	@@@@@@@@
good	WMWMWMWM
evil	**********
coveting	JOYJOYJOY
arrogance	JOYJOYJOY
pride	(:)(:)(:)(:)
Hecuba	6969696969
knowledge	6969696969
ignorance	::::::::::
Outer Blue Text	Inner Blue Text
poetry	OOOOOOOO
poetry	OoOoOoOoOoOo
all	*************
is	eXeXeXeXeXeX
poetry	MeMeMeMeMeMe
detroyed	@o@o@o@o@o@o@o
and	oOoOoOoOoO
created	696969696969

Table 7: Canto 1 Texts

Canto 2 Texts	
Outer Green Text	Outer Blue Text
I wanna	Two texts

explode, reveal what's inside the word cracked open, language eruption mind rupture meaning hemorage do me till poetry is dead reborn gamma 9 splice of mind and word exploded explode me make me blow your mind across the screen write me right me write my innards write the inner word write the outer word write the green guts and skin of language	four texts vortext coretext, I am the other woven with the one I am of two minds four minds your mind make me what I am make me what you will I am the enigma of me outer blue word in a sea of other mind I explode into another sextext
Inner Green Text	Inner Blue Text
Sexesexesexe *c*o*r*t*e*x*t OOOOOOOOO instinct ???????????? Nuclettrisme fire breathing brain spasm @@@@@@@@@ OOOOOOOOOOOO poetry is alive poetry is dead surd word {}{}{}{}{}{} 69696969696969 FUCKFUCKFUCK WRITEWRITE HEARTHEARTHEART ****************	10101010 A1Z0A1Z0 OOOOOOOO ************** 1234567890 ZNZNZNZNZN @O@O@O@O <><><><><><> YOUYOUYOU inner blue JOYJOYJOY ?????????? MEMEMEME 888888888 brain spasm MWMWMW @@@@@@@ 6969696969 SEXESEXE

WRITE YOU RIGHT YOU green guts desire neath the word visible manifestation THE UNSAID THE SAID GODTONGUE	

Table 8: Canto 2 Texts

Game Mode Winning Notices

```
pWin=["The idea is to drive poetry in dangerous directions.", \
"You are on the demolition squad.", \
"What is revealed when the word is cracked open?", \
"This is the funky chicken of poetry.", \
"Closure successfully manufactured.", \
"What poems mean is partly--sometimes mostly--what you make of them. So be bold.
Be daring. But don't be stupid.", \
"Poetry is the art of being profoundly vague.", \
"This feels like poetry lives: contentiously.", \
"What is celebrated usually depends on who is doing the celebrating.", \
"Poetry triumphs over the forces of dullness.", \
"This is a language machine.", \
"How does it feel, " & sprite(gUserDataManager).getUserName() & "?", \
"Part poem, part twitch.", \
"Poetry is not a game somebody wins.", \
"Poetry is no longer just words on a page.", \
"Language machines need a funk button. This whole thing is a funk button.", \
"We read differently now.", \
QUOTE & "My favourite poem is the one that starts 'Thirty days hath September'
because it actually tells you something." & QUOTE & RETURN & "Groucho Marx",
\
"Poetry created and destroyed.", \
QUOTE & "The finest poetry was first experience." & QUOTE & RETURN &
"Ralph Waldo Emerson", \
"Arteroids as poem.", \
"Word as object.", \
"Action poem.", \
"Fire and war were Heraclitus's editors.", \
"Pause. Enjoy.", \
"Score one for strange poetry.", \
"It's an odd feeling to 'win' at this game/poem.", \
"Osama bin Laden sees himself as a poet.", \
"You word warrior.", \
"Spam mapS.", \
"Poetry in motion.", \
"The poetry machine.", \
"Even computer games that don't have any language in them are written things.", \
"This whole thing is a literary machine.", \
"Literary machines are made out of language.", \
"There's no proof we aren't fundamentally machines.", \
"If we're machines, that only shows how complex and strange machines can be.", \
"The wreader is a reader and writer.", \
"You type among the broken shards of language.", \
QUOTE & "Poet, be seated at the piano. Play the present, its hoo-hoo-hoo, its shoo-
shoo-shoo, its ric-a-nic, its envious cachinnation." & QUOTE & RETURN &
"Wallace Stevens", \
"The idea here is to explore the relations of poetry, game, art and play. And have
some fun.", \
"Even when poems aren't about poetry they're about poetry.", \
QUOTE & "A poem is a machine made out of words." & QUOTE & RETURN &
"W.C. Williams", \
QUOTE & "Most of the poets I've known aren't poets anymore." & QUOTE &
RETURN & "Al Purdy", \
"What people tell you is important about poetry is what's important to them.", \
"The poets are the generalists of language arts.", \
"This piece is about the poetics of literary machines.", \
"The seems of reality.", \
"Only " & string(216 -
sprite(gScoreModeAndLevelManager).returnClosestLowerLevel()) & " more levels
to go.", \
QUOTE & "It's a piece of cake until you get to the top. You find you can't stop
playing the game the way you've always played it." & QUOTE & RETURN &
"Richard M. Nixon", \
QUOTE & "Art is a game which has two objectives: it seeks to attract interest, be
memorable, and prompt re-creation--it competes for brainspace and recreation time so
it will propagate competitively as a cultural virus or 'meme' or 'memeplex'; also,
important art competes to change awareness of possibilities for perceiving or acting
or being acted upon, and for making more potent art." & QUOTE & RETURN &
"Gary Boyd", \
"Literary contests negate what they affirm.", \
"In our age, we are becoming aware of the usefulness of thinking of the ways in
which game is present in things.", \
"Poetry is and is not a game somebody wins.", \
QUOTE & "The man who has no problems is out of the game." & QUOTE &
RETURN & "Elbert Hubbard", \
QUOTE & "Here one must think of writing as a game within language." & QUOTE
& RETURN & "Of Derrida", \
QUOTE & "Politics is not a game. It is an earnest business." & QUOTE & RETURN
```

```
& "Winston Churchill", \
"Satisfaction. Dissatisfaction.", \
QUOTE & "When the game is over it is really just beginning." & QUOTE &
RETURN & "Jerry Kramer", \
"If we don't smarten up we'll blow ourselves off the planet or poison the whole
world.", \
"At low speed you can read the text. At high levels, it's pure computer game.", \
"You open a book. It's a game. You open your life. Have some fun.", \
"I wanted to make this literary machine a real game.", \
QUOTE & "The better the coaching has become, the worse the game has become." &
QUOTE & RETURN & "Scotty Bowman", \
QUOTE & "Whatever games are played with us, we must play no games with
ourselves, but deal in our privacy with the last honesty and truth." & QUOTE &
RETURN & "Ralph Waldo Emerson", \
QUOTE & "Games are a compromise between intimacy and keeping intimacy away."
& QUOTE & RETURN & "Eric Berne", \
QUOTE & "By amusing myself with all these games, all this nonsense, all these
picture puzzles, I became famous. I am only a public entertainer who has understood
his time." & QUOTE & RETURN & "Pablo Picasso", \
QUOTE & "All my games were political games; I was, like Joan of Arc, perpetually
being burned at the stake." & QUOTE & RETURN & "Indira Gandhi", \
QUOTE & "A stereotyped but unconscious despair is concealed even under what are
called the games and amusements of mankind." & QUOTE & RETURN & "Henry
David Thoreau", \
"Cool. You made it.", \
QUOTE & "In love, in art, in avarice, in politics, in labor, in games, we study to utter
our painful secret." & QUOTE & RETURN & "Ralph Waldo Emerson", \
QUOTE & "This is a war universe. War all the time. That is its nature. There may be
other universes based on all sorts of other principles, but ours seems to be based on
war and games." & QUOTE & RETURN & "William S. Burroughs", \
QUOTE & "War's a game which, were their subjects wise, Kings would not play at."
& QUOTE & RETURN & "William Cowper", \
QUOTE & "It's never just a game when you're winning." & QUOTE & RETURN &
"George Carlin", \
"Are computers helping us live better lives?", \
"An odd achievement.", \
QUOTE & "The game of life is not so much in holding a good hand as playing a poor
hand well." & QUOTE & RETURN & "H.T. Leslie", \
QUOTE & "Politics isn't about big money or power games; it's about the
improvement of people's lives." & QUOTE & RETURN & "Paul Wellstone", \
QUOTE & "What is sport to the cat is death to the mouse." & QUOTE & RETURN
& "German proverb", \
"Art as game is the politics of social relations.", \
"Although the notion of the 'best art' is empty, artists want to be seen as the best. This
is the tacky game of art.", \
"Artists can't compete with big games companies, but artists can make better concept
```

```
games, more thoughtful and rewarding games.", \
"Well done, " & sprite(gUserDataManager).getUserName() & ".", \
"Only " & string(216 -
sprite(gScoreModeAndLevelManager).returnClosestLowerLevel()) & " more levels
to go.", \
QUOTE & "I listen to kids play a lot." & QUOTE & RETURN & "Itzhak Perlman", \
"Play is hard-wired into us and other animals.", \
QUOTE & "Play is the exultation of the possible." & QUOTE & RETURN &
"Martin Buber", \
"At low levels of play, Arteroids is readable more like we read normally.", \
QUOTE & "A play there is, my lord, some ten words long, which is as brief as I have
known a play, but by ten words, my lord, it is too long, which makes it tedious." &
QUOTE & RETURN & "Shakespeare", \
QUOTE & "To play is nothing but the imitative substitution of a pleasurable,
superfluous and voluntary action for a serious, necessary, imperative and difficult
one." & QUOTE & RETURN & "Max J. Friedländer", \
QUOTE & "Play for young children is not recreation activity,... It is not leisure-time
activity nor escape activity.... Play is thinking time for young children. It is language
time. Problem-solving time." & QUOTE & RETURN & "James L. Hymes, Jr.", \
QUOTE & "To play safe, I prefer to accept only one type of power: the power of art
over trash, the triumph of magic over the brute." & QUOTE & RETURN &
"Vladimir Nabokov", \
QUOTE & "Thank God this isn't a play. Critics can kill a play. But not a hotel." &
QUOTE & RETURN & "John Portman", \
QUOTE & "Blues is easy to play, but hard to feel." & QUOTE & RETURN & "Jimi
Hendrix", \
"You win.", \
QUOTE & "With foxes we must play the fox." & QUOTE & RETURN & "Thomas
Fuller", \
QUOTE & "There is something shameful about the death of a play. It doesn't die with
pity, but contempt." & QUOTE & RETURN & "Mary Roberts Reinhart", \
QUOTE & "As History stands, it is a sort of Chinese Play, without end and without
lesson." & QUOTE & RETURN & "Henry Brooks Adams", \
QUOTE & "A child of five would understand this. Send someone to fetch a child of
five." & QUOTE & RETURN & "Groucho Marx", \
"You can play levels you've passed again if you want.", \
QUOTE & "You have a blue guitar, you do not play things as they are." & QUOTE
& RETURN & "Wallace Stevens", \
QUOTE & "There comes a point in many people's lives when they can no longer play
the role they have chosen for themselves." & QUOTE & RETURN & "Brian Moore",
\
QUOTE & "The true object of all human life is play. Earth is a task garden; heaven is
a playground." & QUOTE & RETURN & "G.K. Chesterton", \
QUOTE & "If A is success in life, then A equals x plus y plus z. Work is x; y is play;
and z is keeping your mouth shut." & QUOTE & RETURN & "Albert Einstein", \
QUOTE & "A Judge may play a little at cards for his own amusement; but he is not
```

```
to play at marbles..." & QUOTE & RETURN & "Samuel Johnson", \
QUOTE & "An assassin is never entirely assassin. They play a role, you understand. While a dead man, he is really dead." & QUOTE & RETURN & "Jean-Paul Sarte", \
"Try 'Word for Weirdos' in 'play mode' if you haven't already.", \
QUOTE & "What we play is life." & QUOTE & RETURN & "Louis Armstrong", \
"Only " & string(216 - sprite(gScoreModeAndLevelManager).returnClosestLowerLevel()) & " more levels to go.", \
"Shoot for art.", \
"Like listserv conversation between artists. Bang bang.", \
"Dedicated to those who remain in the art fray all their lives.", \
"Like a net.art sand mandala.", \
"Listen to what other people say about art, but it's up to you to make up your own mind.", \
"If you want to be a popular artist, suffer fools gladly.", \
"The art itself should matter.", \
"When the academies have too much power in matters of art, no one else is interested.", \
"I once heard a well-known curator say 'An artist without a gallery is nothing'. A publisher might say the same of an author. But that needs to change, and it is, via the Internet.", \
"Arteroids as artists. History as id-entity. Score points as they may, history will have its way.", \
"What you think and feel about art is as important as what anybody else thinks and feels about it.", \
"That there are no absolute rules in art is scary but exciting.", \
"Because there are no absolute rules about art, the notion of what art is best is empty.", \
"There's two types of power: your own, and the type that can be bestowed on you by others. Which do you want?", \
QUOTE & "We are called to be architects of the future, not its victims." & QUOTE & RETURN & "Buckminster Fuller"]
```

Game Mode Death Notices

```
a="Poetry has suffered another blow."
b="The battle of poetry against itself and the forces of dullness."
c="Score one for the forces of dullness."
d="Go write a sonnett."
e="Rewrite your notions of poetry."
f="Devoured by itself."
g="The critics have savaged you."
h="Formless."
i="Oops. Poetry is dead. And it's your fault."
j="So much for poetry."
```

```
k="This is a time loop. You are in a rhythm."
l="Whatever else is true, we live many lives in this one."
m="Poetry passed away " & the long date & ", " & the long time & "."
n="Incohate."
o="You're blasted."
p="Poetry has suffered as never before."
q="Oops."
r="When you write something, you usually use the keyboard."
s="Imagine a word with a mind of its own."
t="You are experiencing your own thought forms." & RETURN & RETURN
& "Play
again?"
u="Poetry will suffer this without loss."
v="Poor old poetry."
w="Loss. Pain."
x="The score is in your mind."
y="You're dead."
z="You can just use the keyboard while playing."
aa="The cosmic drama."
ab="Little death. Play again?"
ac="Meaning?"
ad="Poetry has been tormented mercilessly. Play again?"
ae="The idea is to muse with you while you triumph and die a thousand times.
Time chamber."
af="Is this a broadening of poetry or a reduction? Or something else?"
ag="Different levels are harder or easier at different screen sizes or
resolutions."
ah="Hello " & sprite(gUserDataManager).getUserName() & "." & RETURN
& " You
appear to be
toast."
ai="Is this content?"
aj="Poetry has just bitten its own tail. Bandaid? Or hold on?"
ak="The little death."
al="Language problems."
am="It is safe to die now."
an="Should poetry go here?"
ao="Death notice number 9."
ap="You will be reborn in Kapuscasing."
aq="I'm sorry. You were Daffy Duck in another life."
ar="Death notice:" & RETURN & "Please clean up after yourself." &
RETURN &
RETURN & "Thank
you." & RETURN & "The Management."
as="Rhetorical, dude."
at="There are 216 bardoroids."
```

```
au="If a poem happened in conversation, would we recognize it?"
av="A light touch. Like typing." & RETURN & RETURN & "Bang bang." &
RETURN &
RETURN & "Ooo.
Little death."
aw="What is inside words cracked open?"
ax="Tip: turn out all the lights!"
ay="Tip: turn up the sound!"
az="You killed poetry. I don't know whether to congratulate you or cry."
ba="Little death."
bb="Oh maybe we should just leave poetry alone."
bc="The net writers have answered your email."
bd="Afterlife" & RETURN & "before you die."
be="Afterlife. Play again?"
bf="It does not look good for poetry."
bg="Epitaph epigraph."
bh="Poetry is now invisible."
bi="Dandelion seeds."
bj="Language cracked open." & RETURN & "Fix it?" & RETURN & "Let it
bleed?"
bk="It does not look good for poetry."
bl="Arteroidial schlerosis." & RETURN & "Poetry attack." & RETURN &
"Defibrillate?"
bm="Hemeroidal arteroid encountered." & RETURN & "Blast away?"
bn="Poetry expired from hardening of the Arteroids. Take a walk."
bo="There are more defeats in a lifetime than victories."
bp="An obstreperous text has edited you extensively."
bq="Words are jumping off the page."
br="Crack language open."
bs="Poetry converted to energy."
bt="Breathe."
bu="Sing."
bv="Fire."
bw="Worgatory."
bx="Ascend the 216 bardoroids."
by="Will this piece be dated in thirty years? Or even playable?"
bz="I feel for you in your beleagered poetry ship. You're toast."
ca="Tip: shoot accurately for higher scores."
cb="Tip: you can have max 6 missives on stage at once."
cc="An obstreperous text has damaged your id-entity. Water off a duck's
back."
cd="Bummer."
ce="Ouch."
cf="Let's just say you are no longer suffering."
cg="Tip: try Play Mode to learn how to drive the red id-entity."
ch="Tip: if you experience problems, click anywhere on the screen and try
```

```
again."
ci="There is winning" & RETURN & "there is losing." & RETURN & "But
which is
which" & RETURN &
"is confusing." & RETURN & RETURN & "Tip: you lost."
cj="I read that some people" & RETURN & "have committed suicide after" &
RETURN & "being
ridiculed in computer games." & RETURN & RETURN & "You don't need
to, OK?" &
RETURN & "You're
already dead." & RETURN & "You just haven't hit the ground."
ck="You have been rewritten by words with legs."
cl="The green and blue texts of destruction have done some damage."
cm="Poem bites back."
cn="Losing is for winners and losers."
co="Winning is for winners and losers."
cp="You haven't hit the ground."
cq="Our greatest victories are of the spirit."
cr="Poetry is an action."
cs="When the word cracks open, the future spills out."
ct="When the word cracks open, what is revealed?"
cu="Deepest condolences."
cv="Get well soon."
cw="So sad."
cx="It is never" & RETURN & "the thing we want," & RETURN & "but the
feeling"
& RETURN & "we
think the thing" & RETURN & "will give us."
cy="You're a winner." & RETURN & "You're a loser." & RETURN &
"You're a human
being."
cz="How many times" & RETURN & "do we win" & RETURN & "in one
life?" & RETURN
& RETURN & "And
how many times" & RETURN & "do we lose?" & RETURN & RETURN &
"Oh, uh, add 1" &
RETURN & "on the
'lose' side."
da="You will now be reincarnated as a post-print poet."
db="This is the battle of poetry against itself and the forces of dullness."
dc="Ten lashes with a wet noodle."
dd="You have to lose a few to win."
de="Your id-entity has been deconstructed."
df="What's in a word?"
dg="Do not fear for poetry."
dh="Reading this poem is like riding a horse."
```

```
di="I guess you'd be the mighty word 'poetry'."
dj="Score one for texts with legs."
dk="Blue is for the sky and the mind."
dl="The green is the lush grass and the earth we walk on."
dm="Even if you don't read you read."
dn="Save poetry from yourself."
dp="What is poetry?"
dq= "Art, game, and play?"
dr="Road kill."
ds="Can this be fun and also poetry?"
dt="The poetics of death."
du="Poetry must die occassionally."
dv="Time grows old, time grows old," & RETURN & "can't remember what
it's
told."
dw="Book of firecrackers!"
dx="Should poetry be good for society?"
dy="The blue text seeks out poetry."
dz="The green text does not care about poetry."
ea="This is the downswing."
eb="Language wears out. Like technology?"
ec="Language is a technology."
ed="Technology consists of tools made by people."
ee="Poetry has collided with a marauding text."
ef="This is where your friends carry you out."
eg="The soul."
eh="You are as one drop in a full bottle of wine. Delicious!"
ei="I wish for you that you win delight."
ej="I wish for you that you win laughter."
ek="I wish for you that you win pleasure."
el="I wish for you that you win exhilaration."
em="There's rhythm in game and music."
en="I wish for you that you win your love."
eo="The net is a set of cruising texts."
ep="Beware the psycho text."
eq="Like email wars."
er="Let's see who can be happy first--one, two three, go!"
es="poapptree"
et="O my."
eu="While you are playing, just use the keyboard. The mouse is for
navigating
other parts of
Arteroids."
ev="We imagine what is at stake."
ew="You get savaged by the odd text."
ex="Nothing inherantly shoot-em-up about the code of a shoot-em-up."
```

```
ey="If it's too fast for you, boost your monitor's resolution."
ez="How do game and poetry relate?"
fa="How do game and play relate?"
fb="How do art and game relate?"
fc="How do poetry and art relate?"
fd="How do poetry and play relate?"
fe="How do play and art relate?"
ff="Dancing/grooving is to music as playing is to game."
fg="How do poetry and programming relate?"
fh="The dynamo and its rhythms."
fi="Those texts have had their way with you."
fj="Communications?"
fk="If it's too slow for you, make the browser window smaller."
fl="Head-on collision between poetry and an obstreperous text."
fm="This is the part where they say nice things about you."
fn="Is this bad for poetry?"
fo="Way to go, Shakespeare."
fp="Are you writing with the keyboard?"
fq="In the beginning was the word. And the word grew legs, and yea, you are
toast."
fr="Obstreperous texts have deconstructed your id-entity."
fs="This will eventually be the world's first addictive poem."
ft="Allow me to dedicate this game to those who would prefer to die but are
curious about what
will happen next."
fu="May you live a long and intense life! Start over?"
fv="The battle of poetry against itself and the forces of dullness."
fw="How do programming and art relate?"
fx="new language <=> new mind <=> new media"
fy="new(language <=> mind <=> media)"
fz="language <=> mind <=> media"
ga="Rhythm, pattern, and weave."
gb="Who are you playing to, in your mind?"
gc="The cosmic drama of birth, death, and generation."
gd="The cosmic drama of winning and losing."
ge="The cosmic play."
gf="The cosmic game."
gg="Character is fate."
gh="What does how we imagine dying say about us?"
gi=QUOTE & "Time is a child playing at dice; the kingdom is a child's." &
QUOTE & "--
Heraclitus"
gj=QUOTE & "Attention is the soul's prayer." & QUOTE & " -- Paul Celan"
```

Appendix C: Directory Listing of The *Arteroids* Development Folder

```
J:\arteroids\

1_38\
1_39\
203\
25\
_notes\
doc sounds\
documentation\
dswmedia\
finnish\
for distribution\
graphics\
interviewgraphics\
me explosion\
old prefs files\
portugueseinterview\
Prefs\
sounds\
webarteroids (old version of arteroids)\
alphabetblue2008good.fla
alphabetblue2008good.swf
alphabetgreen2008good.fla
alphabetgreen2008good.swf
andrewsinterview.zip
arteroids.zip
arteroids25music1.dir
arteroids25music2.dir
arteroids2_0.dcr
arteroids2_0.dir
arteroids2_01.zip
arteroids2_0dummy.htm
arteroids2_0portuguese.dir
arteroids2_5.dcr
arteroids2_5.dir
arteroids2_5.zip
arteroids2_5dummy.htm
arteroids2_6.dcr
arteroids2_6.dir
arteroids2_6music3.dir
arteroids2_7MX2004.dir
arteroids311.dcr
arteroids311.dir
```

```
arteroids311.htm
arteroids312.dir
arteroids312.exe
arteroids312OLD.dir
arteroids313.dir
arteroids314.dir
arteroids315.dir
arteroids316.dir
arteroids317.dir
arteroids317.zip
arteroids317white.dir
arteroids318.dir
arteroids318.exe
arteroids3music3.dcr
arteroids3music3.dir
arteroids3music3.htm
arteroids3music30.dir
arteroids3music301.dcr
arteroids3music301.dir
arteroids3music301.htm
arteroids3music302.dcr
arteroids3music302.dir
arteroids3music302.htm
arteroids3music303.dir
arteroids3music304.dir
arteroids3music305.dcr
arteroids3music305.dir
arteroids3music305.htm
arteroids3music306.dir
arteroids3music307.dir
arteroids3music308.dir
arteroids3music309.dcr
arteroids3music309.dir
arteroids3music309.htm
arteroids3music310.dir
asource.htm
deathNotice.doc
ENCRYPTIONSTRINGFOR38.txt
forthejudge.doc
forthejudge.htm
garbage.dcr
garbage.dir
garbage.htm
memtest.dir
reviewshypertext.dcr
reviewshypertext.dir
reviewshypertext.htm
stringtest.dir
test.dir
test.exe
winningNotice.doc
WS_FTP.LOG
~$arteroids.htm

J:\arteroids\1_38\

3D_progbar.dcr
arteroids.htm
arteroids.zip
arteroids1_36.dcr
arteroids1_36.dir
arteroids1_36.htm
arteroids1_37 backup.htm
arteroids1_37.dcr
arteroids1_37.dir
arteroids1_37.htm
arteroids1_38.dcr
arteroids1_38.dir
arteroids1_38.dir old
arteroids1_38.htm
arteroids1_38old.htm
arteroids1_for_Arts_Council.dcr
arteroids1_for_Arts_Council.dir
arteroids1_for_Arts_Council.htm
arteroids35.dcr
arteroids35.dir
arteroids35.htm
arteroids36v85.dcr
arteroids36v85.dir
arteroids36v85.htm
asource.htm
bulletpositioning.dir
Copy of ENCRYPTIONSTRINGFOR38.txt
ENCRYPTIONSTRINGFOR38.txt
index.htm
indexnioandarteroids.htm
WS_FTP.LOG

J:\arteroids\1_39\

arteroids.htm
arteroids.zip
```

```
arteroids1_38.dcr
arteroids1_38.dcrold
arteroids1_38.dir
arteroids1_38.htm
arteroids1_38dummy.htm
arteroids1_38old2.htm
arteroids1_38simple.htm
arteroids1_38v2.dir
arteroids1_38v3.dir
arteroids1_38v4.dir
arteroids1_38v5.dcr
arteroids1_38v5.dir
asource.htm
debugtest.dir
ENCRYPTIONSTRINGFOR38.txt
index.htm
index18.htm
WS_FTP.LOG

J:\arteroids\203\

arteroids2_0.dcr
arteroids2_0.dir
indexenglish.htm

J:\arteroids\25\

arteroids2_5.dir

J:\arteroids\_notes\

dwSiteColumnsMe.xml

J:\arteroids\doc sounds\

1.wav
2.wav
3.wav
4.wav
5.wav
6.wav
7.wav
8.wav
9.wav

J:\arteroids\documentation\
```

graphics\
Anomalie.doc
Arteroids 2.doc
Arteroids by Andrews for Anomalie 2.doc
Arteroids by Andrews for Anomalie 3a.doc
Arteroids by Andrews for Anomalie.doc
Arteroids.doc
Arteroids.pdf
arteroids.zip
arteroids1_38.dcr
arteroids1_38.dir
arteroids1_38.htm
arteroids1_38.ps.prn
Arteroids3.doc
ArteroidsCode.doc
GAMERS_Andrews.pdf
jim_andrews.mov
JimAndrews-Arteroids.doc
JimAndrews-Arteroids.pdf
reviews.htm
rodrigues_dg_me_sjrp1.pdf
WS_FTP.LOG

J:\arteroids\documentation\graphics\

1-6.gif
10-22.gif
101-102-player.gif
104-107-green-arteroids.gif
23-24.gif
25-37-scoreboard.gif
48-72-green-letters.gif
49-statistoids-for-canto1.gif
49-statistoids-screenshot.gif
53-player-loses.gif
7-9.gif
73-97-exploding-blue-letter.gif
arteroids_graphics_for_shanna.zip
blue letters in action.psd
blue-letters-in-action.gif
Canto-1-Frame-Scripts-and-M.gif
code.psd
Credits-(15-22).gif
green letters in action.psd
green-letters-in-action.gif

```
ID.gif
Level-1-initialization.gif
level-1-scoreboard.gif
main-menu-shot-(10-22).gif
mainmenu.gif
mainscreenshot.gif
mainscreenshot.psd
moviescripticon.bmp
Playing Arteroids.psd
PlayingArteroids.bmp
PlayingArteroids.zip
score flowchart.sdr
score1.gif
scoreflowchart.wmf
Text Editor Score.psd
text score screenshot.psd
Text-Editor-Score.gif
text-score-screenshot.gif
Thumbs.db
wordForWeirdos.psd

J:\arteroids\dswmedia\

dummycasts\
arteroids.htm
arteroids2_0portuguese.dcr
arteroids2_0portuguese.htm
arteroids2_5.htm
arteroids2_6.dcr
arteroids2_6.dir
arteroids2_6dummy.htm
arteroidsportuguese.htm
index.htm
indexenglish.htm
indexportuguese.htm
jimandrews.htm
onarteroids.htm
onarteroidsportuguese.htm
poetry5forindex.swf
purchase.htm

J:\arteroids\dswmedia\dummycasts\

dummy maker\
dummy1.cct
dummy1.cst
```

```
dummy10.cct
dummy10.cst
dummy100.cct
dummy100.cst
dummy11.cct
dummy11.cst
dummy12.cct
dummy12.cst
dummy13.cct
dummy13.cst
dummy14.cct
dummy14.cst
dummy15.cct
dummy15.cst
dummy16.cct
dummy16.cst
dummy17.cct
dummy17.cst
dummy18.cct
dummy18.cst
dummy19.cct
dummy19.cst
dummy2.cct
dummy2.cst
dummy20.cct
dummy20.cst
dummy21.cct
dummy21.cst
dummy22.cct
dummy22.cst
dummy23.cct
dummy23.cst
dummy24.cct
dummy24.cst
dummy25.cct
dummy25.cst
dummy26.cct
dummy26.cst
dummy27.cct
dummy27.cst
dummy28.cct
dummy28.cst
dummy29.cct
dummy29.cst
dummy3.cct
dummy3.cst
```

```
dummy30.cct
dummy30.cst
dummy31.cct
dummy31.cst
dummy32.cct
dummy32.cst
dummy33.cct
dummy33.cst
dummy34.cct
dummy34.cst
dummy35.cct
dummy35.cst
dummy36.cct
dummy36.cst
dummy37.cct
dummy37.cst
dummy38.cct
dummy38.cst
dummy39.cct
dummy39.cst
dummy4.cct
dummy4.cst
dummy40.cct
dummy40.cst
dummy41.cct
dummy41.cst
dummy42.cct
dummy42.cst
dummy43.cct
dummy43.cst
dummy44.cct
dummy44.cst
dummy45.cct
dummy45.cst
dummy46.cct
dummy46.cst
dummy47.cct
dummy47.cst
dummy48.cct
dummy48.cst
dummy49.cct
dummy49.cst
dummy5.cct
dummy5.cst
dummy50.cct
dummy50.cst
```

```
dummy51.cct
dummy51.cst
dummy52.cct
dummy52.cst
dummy53.cct
dummy53.cst
dummy54.cct
dummy54.cst
dummy55.cct
dummy55.cst
dummy56.cct
dummy56.cst
dummy57.cct
dummy57.cst
dummy58.cct
dummy58.cst
dummy59.cct
dummy59.cst
dummy6.cct
dummy6.cst
dummy60.cct
dummy60.cst
dummy61.cct
dummy61.cst
dummy62.cct
dummy62.cst
dummy63.cct
dummy63.cst
dummy64.cct
dummy64.cst
dummy65.cct
dummy65.cst
dummy66.cct
dummy66.cst
dummy67.cct
dummy67.cst
dummy68.cct
dummy68.cst
dummy69.cct
dummy69.cst
dummy7.cct
dummy7.cst
dummy70.cct
dummy70.cst
dummy71.cct
dummy71.cst
```

```
dummy72.cct
dummy72.cst
dummy73.cct
dummy73.cst
dummy74.cct
dummy74.cst
dummy75.cct
dummy75.cst
dummy76.cct
dummy76.cst
dummy77.cct
dummy77.cst
dummy78.cct
dummy78.cst
dummy79.cct
dummy79.cst
dummy8.cct
dummy8.cst
dummy80.cct
dummy80.cst
dummy81.cct
dummy81.cst
dummy82.cct
dummy82.cst
dummy83.cct
dummy83.cst
dummy84.cct
dummy84.cst
dummy85.cct
dummy85.cst
dummy86.cct
dummy86.cst
dummy87.cct
dummy87.cst
dummy88.cct
dummy88.cst
dummy89.cct
dummy89.cst
dummy9.cct
dummy9.cst
dummy90.cct
dummy90.cst
dummy91.cct
dummy91.cst
dummy92.cct
dummy92.cst
```

```
dummy93.cct
dummy93.cst
dummy94.cct
dummy94.cst
dummy95.cct
dummy95.cst
dummy96.cct
dummy96.cst
dummy97.cct
dummy97.cst
dummy98.cct
dummy98.cst
dummy99.cct
dummy99.cst
WS_FTP.LOG

J:\arteroids\dswmedia\dummycasts\dummy maker\

dummy maker.dcr
dummy maker.dir
dummy maker.htm
dummy1.cct
dummy1.cst
dummy10.cct
dummy10.cst
dummy100.cct
dummy100.cst
dummy11.cct
dummy11.cst
dummy12.cct
dummy12.cst
dummy13.cct
dummy13.cst
dummy14.cct
dummy14.cst
dummy15.cct
dummy15.cst
dummy16.cct
dummy16.cst
dummy17.cct
dummy17.cst
dummy18.cct
dummy18.cst
dummy19.cct
dummy19.cst
dummy2.cct
```

```
dummy2.cst
dummy20.cct
dummy20.cst
dummy21.cct
dummy21.cst
dummy22.cct
dummy22.cst
dummy23.cct
dummy23.cst
dummy24.cct
dummy24.cst
dummy25.cct
dummy25.cst
dummy26.cct
dummy26.cst
dummy27.cct
dummy27.cst
dummy28.cct
dummy28.cst
dummy29.cct
dummy29.cst
dummy3.cct
dummy3.cst
dummy30.cct
dummy30.cst
dummy31.cct
dummy31.cst
dummy32.cct
dummy32.cst
dummy33.cct
dummy33.cst
dummy34.cct
dummy34.cst
dummy35.cct
dummy35.cst
dummy36.cct
dummy36.cst
dummy37.cct
dummy37.cst
dummy38.cct
dummy38.cst
dummy39.cct
dummy39.cst
dummy4.cct
dummy4.cst
dummy40.cct
```

```
dummy40.cst
dummy41.cct
dummy41.cst
dummy42.cct
dummy42.cst
dummy43.cct
dummy43.cst
dummy44.cct
dummy44.cst
dummy45.cct
dummy45.cst
dummy46.cct
dummy46.cst
dummy47.cct
dummy47.cst
dummy48.cct
dummy48.cst
dummy49.cct
dummy49.cst
dummy5.cct
dummy5.cst
dummy50.cct
dummy50.cst
dummy51.cct
dummy51.cst
dummy52.cct
dummy52.cst
dummy53.cct
dummy53.cst
dummy54.cct
dummy54.cst
dummy55.cct
dummy55.cst
dummy56.cct
dummy56.cst
dummy57.cct
dummy57.cst
dummy58.cct
dummy58.cst
dummy59.cct
dummy59.cst
dummy6.cct
dummy6.cst
dummy60.cct
dummy60.cst
dummy61.cct
```

```
dummy61.cst
dummy62.cct
dummy62.cst
dummy63.cct
dummy63.cst
dummy64.cct
dummy64.cst
dummy65.cct
dummy65.cst
dummy66.cct
dummy66.cst
dummy67.cct
dummy67.cst
dummy68.cct
dummy68.cst
dummy69.cct
dummy69.cst
dummy7.cct
dummy7.cst
dummy70.cct
dummy70.cst
dummy71.cct
dummy71.cst
dummy72.cct
dummy72.cst
dummy73.cct
dummy73.cst
dummy74.cct
dummy74.cst
dummy75.cct
dummy75.cst
dummy76.cct
dummy76.cst
dummy77.cct
dummy77.cst
dummy78.cct
dummy78.cst
dummy79.cct
dummy79.cst
dummy8.cct
dummy8.cst
dummy80.cct
dummy80.cst
dummy81.cct
dummy81.cst
dummy82.cct
```

dummy82.cst
dummy83.cct
dummy83.cst
dummy84.cct
dummy84.cst
dummy85.cct
dummy85.cst
dummy86.cct
dummy86.cst
dummy87.cct
dummy87.cst
dummy88.cct
dummy88.cst
dummy89.cct
dummy89.cst
dummy9.cct
dummy9.cst
dummy90.cct
dummy90.cst
dummy91.cct
dummy91.cst
dummy92.cct
dummy92.cst
dummy93.cct
dummy93.cst
dummy94.cct
dummy94.cst
dummy95.cct
dummy95.cst
dummy96.cct
dummy96.cst
dummy97.cct
dummy97.cst
dummy98.cct
dummy98.cst
dummy99.cct
dummy99.cst

J:\arteroids\finnish\

arteroids317-finnish.dir
arteroids317-finnish.zip
arteroids318-finnish.dir
arteroids318-finnish.zip

J:\arteroids\for distribution\

arteroids.dir
arteroidsjim.htm
README.TXT

J:\arteroids\graphics\

essaygraphics\
2-2a.fla
agent7.gif
agenttrans.gif
artericon.bmp
artericon.psd
arteroids good screenshot.psd
arteroids good screenshot2.psd
arteroids screenshot copy.bmp
arteroids screenshot.psd
arteroids text editor screenshot copy.bmp
arteroids text editor screenshot.psd
arteroids.ico
arteroids1.bmp
arteroids1.gif
arteroids1b.bmp
arteroids2.psd
arteroidslogo.fla
arteroidslogo.html
arteroidslogo.swf
arteroidslogoMX.fla
arteroidslogoMX.html
arteroidslogoMX.swf
arteroidsscreens0 copy.bmp
arteroidsscreens0.psd
background1.psd
background2.psd
background3.psd
bernie.jpg
bullet.bmp
bullet.psd
button email.psd
Button.psd
canadacoun2.gif
canadacoun2.psd
cancoun.bmp
cancoun.gif
centrelogo.gif
chairetmetal.gif

chairetmetal.psd
close.bmp
close.psd
confirm44.gif
Credits.gif
credits.psd
credits2.gif
credits2.psd
credits2_0.gif
credits2_01.gif
credits2_02.gif
db1trans.gif
db2trans.gif
db4.gif
db5.gif
emaildown.bmp
emailup.bmp
get_shockwave.gif
gold texture 2.psd
gold texture.jpg
gold.gif
gold.psd
helpbg.psd
helpbg2.psd
hypertiling.psd
hypertiling2.psd
IDnumDOWN.bmp
IDnumDOWN.psd
idnumUP.bmp
instinct.psd
keyboard.bmp
keyboard.jpg
keyboard.psd
keyboardx.bmp
keyboardx.psd
large_a.gif
layers4.gif
loadbal10a.gif
loadbal17.gif
loadbal1trans.gif
loadbal2trans.gif
main menu gold 0.5.psd
main menu gold copy.bmp
main menu gold copy.psd
main menu gold.psd
main menu practice.psd

onion1 copy.bmp
onion1.psd
PLAY MODE startup.psd
poetry.fla
poetry2.fla
poetry3.fla
poetry3.html
poetry3.swf
poetry3MX2004.fla
poetry3MX2004.html
poetry3MX2004.swf
poetry3noActionscript.fla
poetry3noActionscript.swf
poetry4.fla
poetry4.html
poetry4.swf
poetry4MX2004.fla
poetry4MX2004.html
poetry4MX2004.swf
poetry5.fla
poetry5.html
poetry5.swf
poetry5foressay.fla
poetry5foressay.html
poetry5foressayMX.fla
poetry5foressayMX.html
poetry5foressayMX.swf
poetry5forindex.fla
poetry5forindex.html
poetry5forindex.swf
poetry5MX.fla
poetry5MX.html
poetry5MX.swf
poetry5MX2004.fla
poetry5MX2004.html
poetry5MX2004.swf
quadgameslogo.gif
quadgameslogo.psd
RAIDtrans.gif
randy texture.psd
reginalogo.gif
remedi.gif
remote3.gif
remotetrans.gif
reviews-mouseoversimanowski.gif
reviews.gif

reviews1.psd
reviews2.gif
reviews2.psd
reviews3.gif
reviews3.psd
reviews4.gif
reviews4.psd
ScoresBugAtLevel216.gif
screenshot.gif
slave8.gif
slavetrans.gif
slider both.psd
slider handle.bmp
slider handle.psd
slider rail.bmp
small.gif
SubmitScores button.psd
switch6.gif
switchtrans.gif
Thumbs.db
turb_logo.gif
tv copy.bmp
tv.bmp
tv.psd
tv1.psd
tv2.psd
WeLiveManyLives.psd
WS_FTP.LOG
zettaarchitecture9b.gif

J:\arteroids\graphics\essaygraphics\

_notes\
action1.gif
action1trans.gif
action1trans2.gif
action2.gif
background2.gif
background3.gif
bl3.bmp
defeat.gif
email.gif
email2.gif
flowchart.gif
flowchart.psd
flowchart2.gif

```
flowchart3.gif
flowchart4.gif
flowchart5.gif
gamemodemenu.gif
info.gif
levelcontrol1.gif
mainmenu.gif
playmodemenu.cpt
playmodemenu.gif
playmodemenu.psd
poetry5foressay.swf
savemenu.gif
Thumbs.db
wordforweirdos.gif
WS_FTP.LOG

J:\arteroids\graphics\essaygraphics\_notes\


J:\arteroids\interviewgraphics\

back.gif
INTER.gif
next.gif
nothing.gif
reginalogo.gif
Thumbs.db

J:\arteroids\me explosion\

test.dir

J:\arteroids\old prefs files\

jwebtest1.txt (matt)
jwebtest1.txt (randy)

J:\arteroids\portugueseinterview\

images\
divulgaç¦o.zip
Reginajimportuguese.htm

J:\arteroids\portugueseinterview\images\

back.gif
```

INTER.gif
next.gif
reginalogo.gif

J:\arteroids\Prefs\

jwebtest1.txt

J:\arteroids\sounds\

enigman sounds\
forJeremy\
music\
musicgun\
old death sounds\
poetrycraft\
amie.mp3
bushlong.wav
d1.wav
d10.wav
d11.wav
d12.wav
d13.wav
d14.wav
d15.wav
d16.wav
d17.wav
d18.wav
d19.wav
d2.wav
d20.wav
d21.wav
d26.wav
d27.wav
d28.wav
d3.wav
d32.wav
d33.wav
d34.wav
d4.wav
d5.wav
d6.wav
d7.wav
d8.wav
d9.wav
deathofpoetry.wav

```
Full_Arteroids_2_Recording.mp3
Full_Arteroids_2_Recording.wav
Full_Arteroids_2_Recordinggarbage.mp3
highScore.wav
homawhosong.mp3
lose long.wav
lose.wav
lose2.wav
shiplong.wav
shiplongreverb.wav
shoot1.wav
shoot10.wav
shoot2.wav
shoot3.wav
shoot4.wav
shoot5.wav
shoot6.wav
shoot7.wav
shoot8.wav
shoot9.wav
shootlong.wav
song1.bun
song1.wrk
Sound18.wav
test.wav
win.wav
WS_FTP.LOG

J:\arteroids\sounds\enigman sounds\

e.wav
g.wav
n.wav

J:\arteroids\sounds\forJeremy\

zip\
1.wav
4.wav
9.wav
a.wav
d1.wav
d11.wav
d13.wav
d16.wav
d19.wav
```

```
d2.wav
d20.wav
d22.wav
d23.wav
d24.wav
d25.wav
d32.wav
d33.wav
d34.wav
d6.wav
d8.wav
d9.wav
deathofpoetry.wav
e.wav
forNicolle.zip
g.wav
i.wav
m.wav
n.wav
shoot1.wav
shoot10.wav
shoot2.wav
shoot4.wav
shoot5.wav
shoot7.wav
shoot8.wav
shoot9.wav

J:\arteroids\sounds\forJeremy\zip\

1.wav
4.wav
9.wav
a.wav
d1.wav
d11.wav
d13.wav
d16.wav
d19.wav
d2.wav
d20.wav
d22.wav
d23.wav
d24.wav
d25.wav
d32.wav
```

```
d33.wav
d34.wav
d6.wav
d8.wav
d9.wav
deathofpoetry.wav
e.wav
g.wav
i.wav
m.wav
n.wav
shoot1.wav
shoot10.wav
shoot2.wav
shoot4.wav
shoot5.wav
shoot7.wav
shoot8.wav
shoot9.wav

J:\arteroids\sounds\music\

11s.mp3
16s.mp3
17s.mp3
18s.mp3
18s.wav
1s.mp3
2s.mp3
4s.mp3
8s.mp3
8s.wav

J:\arteroids\sounds\musicgun\

wholelottarosie\
10g.wav
11g.wav
12g.wav
13g.wav
14g.wav
15g.wav
16g.wav
17g.wav
1g.wav
2g.wav
```

```
3bg.wav
3cg.wav
3dg.wav
3eg.wav
3fg.wav
3g.wav
3gg.wav
3hg.wav
3ig.wav
3jg.wav
3kg.wav
3lg.wav
3mg.wav
4g.wav
5g.wav
6g.wav
7g.wav
8g.wav
9g.wav
Aerosmith - Walk This Way with markers.wav
Aerosmith - Walk This Way with markers2.sfk
Aerosmith - Walk This Way with markers2.wav
Aerosmith - Walk This Way with markers3 22khz.wav

J:\arteroids\sounds\musicgun\wholelottarosie\

10rosie.wav
11rosie.wav
12rosie.wav
13rosie.wav
14rosie.wav
15rosie.wav
1rosie.wav
2rosie.wav
3rosie.wav
4rosie.wav
5rosie.wav
6rosie.wav
7rosie.wav
8rosie.wav
9rosie.wav

J:\arteroids\sounds\old death sounds\

d1.wav
d10.wav
```

```
d11.wav
d12.wav
d13.wav
d14.wav
d15.wav
d16.wav
d17.wav
d18.wav
d19.wav
d2.wav
d20.wav
d21.wav
d22.wav
d23.wav
d24.wav
d25.wav
d26.wav
d27.wav
d28.wav
d29.wav
d3.wav
d30.wav
d4.wav
d5.wav
d6.wav
d7.wav
d8.wav
d9.wav

J:\arteroids\sounds\poetrycraft\

2\
1.wav
10.wav
11.wav
12.wav
13.wav
14.wav
15.wav
16.wav
17.wav
18.wav
19.wav
2.wav
20.wav
21.wav
```

22.wav
23.wav
24.wav
25.wav
3.wav
4.wav
5.wav
6.wav
7.wav
8.wav
9.wav
last.wav
PoetryCraft.wav

J:\arteroids\sounds\poetrycraft\2\

1.wav
10.wav
11.wav
12.wav
2.wav
3.wav
4.wav
5.wav
6.wav
7.wav
8.wav
9.wav

J:\arteroids\webarteroids (old version of arteroids)\

_notes\
backup\
backup 1\
backup 2\
davidknoebel\
Documentation\
graphics\
indexnew_files\
sound\
index.htm
indexnew.htm
shipshoot.dir
shipshoot8.dcr
shipshoot8.dir
shipshoot8.htm

shipshoot85.dir
shipshoot8text1.dir
testeditor.dcr
testeditor.dir
testeditor.htm
webarteroids1.dcr
webarteroids1.dir
webarteroids1.htm
webarteroids10.dcr
webarteroids10.dir
webarteroids10.htm
webarteroids11.dir
webarteroids12.dcr
webarteroids12.dir
webarteroids12.htm
webarteroids13.dcr
webarteroids13.dir
webarteroids13.htm
webarteroids14.dcr
webarteroids14.dir
webarteroids14.htm
webarteroids15.dcr
webarteroids15.dir
webarteroids15.htm
webarteroids16.dir
webarteroids17.dir
webarteroids18.dir
webarteroids19.dir
webarteroids2.dcr
webarteroids2.dir
webarteroids2.htm
webarteroids20.dir
webarteroids21.dir
webarteroids22.dir
webarteroids23.dir
webarteroids24.dir
webarteroids25.dcr
webarteroids25.dir
webarteroids25.htm
webarteroids26.dir
webarteroids27.dir
webarteroids28.dir
webarteroids29.dir
webarteroids3.dcr
webarteroids3.dir
webarteroids3.htm

webarteroids30.dir
webarteroids31.dir
webarteroids32.dir
webarteroids33.dir
webarteroids34.dir
webarteroids4.dcr
webarteroids4.dir
webarteroids4.htm
webarteroids5.dcr
webarteroids5.dir
webarteroids5.htm
webarteroids6.dcr
webarteroids6.dir
webarteroids6.htm
webarteroids7.dcr
webarteroids7.dir
webarteroids7.htm
webarteroids8.dcr
webarteroids8.dir
webarteroids8.htm
webarteroids9.dcr
webarteroids9.dir
webarteroids9.htm
WS_FTP.LOG

J:\arteroids\webarteroids (old version of arteroids)_notes\

dwSiteColumnsMe.xml

J:\arteroids\webarteroids (old version of arteroids)\backup 1\

shipshoot8.dir

J:\arteroids\webarteroids (old version of arteroids)\backup 2\

testeditor.dir

J:\arteroids\webarteroids (old version of arteroids)\backup\

shipshoot.dir
shipshoot8text1.dir

J:\arteroids\webarteroids (old version of arteroids)\davidknoebel\

groan.mp3
laugh.mp3

lfgrn.mp3
webarteroids.zip

J:\arteroids\webarteroids (old version of arteroids)\Documentation\

Code.doc
User Data Manager code.doc

J:\arteroids\webarteroids (old version of arteroids)\graphics\

arteroids1.bmp
arteroids1b.bmp
arteroids2.gif
Canadacoun1.jpg
canadacoun2.jpg
Cancel Down.bmp
Cancel Up.bmp
cancoun.bmp
cancoun.gif
CheckBoxChecked copy.bmp
CheckBoxChecked.psd
CheckBoxUnchecked copy.bmp
CheckBoxUnchecked.psd
Delete Down.bmp
Delete Up.bmp
drop down button.bmp
drop down button.cpt
face001.bmp
face002.bmp
face003.bmp
facesmall1.bmp
Image01.jpg
Image02.jpg
Image03.jpg
Image04.jpg
Image05.jpg
Image06.jpg
Image07.jpg
Image08.jpg
Image09.jpg
Image10.jpg
Image11.jpg
Image12.jpg
Image13.jpg
Image14.jpg
Image15.jpg

Image16.jpg
Image17.jpg
Image18.jpg
Image19.jpg
Image20.jpg
Image21.jpg
Image22.jpg
Image23.jpg
mary4.cpt
modal background 1.bmp
modal background 1.cpt
modal background 2.bmp
modal background 3.bmp
modal background 3.cpt
modal background 4.bmp
modal background 4.cpt
modal background 5.bmp
modal background 5.cpt
modal background 6.bmp
modal background 6.cpt
modal background 7.bmp
modal background 7.cpt
modal exp.cpt
new-11.cpt
new-12.cpt
new-13.cpt
new-14.cpt
new-14s.bmp
new-14s.cpt
new-14sb.bmp
new-15.cpt
new-15s.bmp
new-15s.cpt
new-15sb.bmp
new-18.cpt
new-18s.bmp
new-18s.cpt
new-18sb.bmp
new-19.cpt
new-19s.bmp
new-19s.cpt
new-19sb.bmp
new-20.cpt
new-20s.bmp
new-20s.cpt
new-20sb.bmp

new-21s.cpt
new-21sb.bmp
new-7.cpt
new-7s.bmp
new-7s.cpt
new-7sb.bmp
new-8.cpt
new-8sb.bmp
new-8sb.cpt
No down.bmp
No up.bmp
OK Down.bmp
OK.bmp
OKButton copy.bmp
OKButton.psd
OKButtonDown.bmp
Save Down.bmp
Save Up.bmp
Save.psd
slider both.psd
slider constraint.bmp
slider rail.bmp
slider.bmp
slider.psd
webarteroids1.gif
webarteroids1small.bmp
webarterysmall,-black-bg.gif

J:\arteroids\webarteroids (old version of arteroids)\indexnew_files\

AnimationStillLife4.jpg
get_shockwave.gif

J:\arteroids\webarteroids (old version of arteroids)\sound\

SEASHA~1.mp3

Appendix D: Selected Files from the Arteroids Development Folder

The following files and folders are contained in the Appendix D folder:

- README – a copy of the text of Appendix D, available for quick reference.
- webarteroids (old version of arteroids)—a folder which contains all the drafts and versions of *Arteroids* before version 1.0. This is folder has been copied as it appears in the *Arteroids* Development Folder. I encourage you to explore the folders, open files, and get a feel for the organization of the virtual space. Imagine what the rest of the Development Folder is like based upon this sample.
- Arteroids Documentation—The code documentation for *Arteroids* 1.0.
- Arteroids1_for_Arts_Council—there are two files with this name: open the HTML one in your browser and it will activate the shockwave movie (the other file). This fully playable version contains a semi-interactive voice recording of Jim Andrews explaining different aspects of the game. It is analogous to director and cast commentary tracks in DVD and Blu-Ray movies. I strongly recommend exploring this fascinating version.
- forthejudge—a document written to accompany *Arteroids* 2.02. It describes a development plan for the work until version 5.0.

If you're interested in viewing and/or playing any of the early versions, open the html files, which will activate the shockwave movie (you may need the plugin for your browser). To view the source code of any of these versions, you need Adobe Director MX 2004 (Version 10.1). The current version (11.5) will attempt (unsuccessfully) to update the code and the result will be inaccurate and unplayable.

The Appendix D folder is available upon request. You can contact me at leonardo.flores@upr.edu. Thank you.

Bibliography

Aarseth, Espen. *Cybertext: Perspectives on Ergodic Literature.* Baltimore: The Johns Hopkins U P, 1997. Print.

“About the ELO.” *Electronic Literature Organization*. n.d. Web 11 April 2010.

Adams, Randy. “Defib: Randy Adams interviews Jim Andrews about Nio” *Turbulence.org*. 22 March 2001. Web. 11 April 2010.

Andrews, Jim. “Chapter 3.2: DHTML Dances.” Comments in Microsoft Word document.14 April 2008.

Andrews, Jim. *Arteroids*. *Vispo.com*. 2001-2006. Web. 15 May 2010.

--. *Arteroids* Development Folder. Digital Files. 2008.

--. "[-empyre-] form and content." *Empyre*. 7 May 2003. Web. 7 Sept. 2008.

--. "[-empyre-] Poetry and Programming (2): Computers are Language Machines." *Empyre*. 4 May 2003. Web. 7 September 2008.

--. "About 'On Lionel Kearns'." *Vispo.com*. 2004. Web. 3 Nov. 2008.

--. "Architecture and the Literary." *Vispo.com*. 1999. Web. 7 Sept. 2008.

--. "Architecture and the Literary." *Vispo.com*. 1999. Web. 2 Nov. 2008.

--. "Architexture of Divine Mind Fragment Theatre." *Vispo.com*. 1999. Web. 2 Nov. 2008.

--. “Arteroids, Poetry, and the Flaw.” *Poems that GO*, Fall 2003. Web. 11 April 2010.

--. “Chapter 3.” Comments in Microsoft Word document. 1 Aug. 2008.

--. “Chapter 3.1 The Electric Pen.” Comments in Microsoft Word document. 09 July 2008.

--. "Digital Langu(im)age -- Language and Image as Objects in a Field." *Vispo.com.* 1998. Web. 7 September 2008.

--. “F8MW9.” Message to the author. 20 Aug. 2008. E-mail.

--. “For the Judge.” *Arteroids Development Folder.* 28 Aug. 2002. File.

--. "Framing 'Off-screen Romance'." *Vispo.com*. 2007. Web. 3 Nov. 2008.

--. “Gregory Whitehead.” Message to the author. 5 May 2008. E-mail.

--. “Infoanimism” *Vispo.com.* 1999. Web. 13 April 2010.

--. "Interactive Audio on the Web 1.". *trAce Online Writing Center*. 22 September 2003. Web. 30 July 2008.

--. “Jim Andrews’ Vispo.com” *Networked Performance*. 11 Dec. 2006. Web. 13 April 2010.

--. "Material Combinatorium Supremum." Vispo.com.1999. Web. 2 Nov. 2008.

--. “Nio.” Message to the author. 1 Aug. 2008. E-mail.

--. "On Lionel Kearns." *Vispo.com*. 2004. Web. 3 Nov. 2008.

--. "On William S. Burroughs." *Vispo.com*. 1992. Web. 7 Sept. 2008.

--. "Online Writings and Vispo." *Vispo.com*. 1992. Web. 7 September 2008.

--. "On-Line Writings and Vispo: Poems." *Vispo.com*. 2008. Web. 3 Nov. 2008.

--. “Oppen Do Down” Message to the author. 31 July 2008. E-mail.

--. “Poetry, Arteroids, and the Flaw.” *Poems that GO*, Fall 2003. Web. 11 April 2010.

--. "Pop up Poems by Jim Andrews." *Vispo.com*. 1999. Web. 1 Nov. 2008.

--. “Re: Back in Business.” Message to the author. 5 Jan. 2010. E-mail.

--. “Re: DHTML Dances.” Message to the author. 14 April 2008. E-mail.

--. “Re: doc.” Message to the author. 12 July 2008. E-mail.

--. “Re: ELO Conference Proposal.” Message to the author. 17 Jan. 2008. E-mail.

--. “Re: ELO Conference Proposal.” Message to the author. 13 Jan. 2008. E-mail.

--. “Re: Notebook.” Message to the author. 26 Mar. 2008. E-mail.

--. “Re: on 'code poetry'.” Message to the author. 7 Nov. 2007. E-mail.

--. “Re: on 'code poetry'.” Message to the author. 30 Nov. 2007. E-mail.

--. “Re: Finally, Chapter 3!.” Message to the author. 8 Nov. 2008. E-mail.

--. “Re: Oppen Do Down.” Message to the author. 22 July 2008. E-mail.

--. “Re: The notebook arrived.” Message to the author. 5 April 2008. E-mail.

--. "Reading McLuhan." *Vispo.com*. 1988. Web. 7 September 2008.

--. "Source Code for Millenium Lyric." *Vispo.com*. 2000. Web. 7 Sept. 2008.

--. "Spastext." *Vispo.com.* 1999. Web. 1 Nov. 2008.

--. "Stir Frys and Cut Ups." *Vispo.com.* 1999. Web. 1 Nov. 2008.

--. "The Art of Interactive Audio." *Vispo.com.* April 2001. Web.7 July 2008.

--. "The Pornomorphs: Ur Grrl Alias 2." *Vispo.com.* 1997. Web. 3 Nov. 2008.

--. "Vispo ~ Langu(im)age Audio." *Vispo.com.* Web. 7 July 2008.

--. “yo.” Message to the author. 10 April 2010. E-mail.

--, et. al. *WebArtery*. Yahoo! Groups. 1998. Web. 15 May 2010.

Barthes, Roland. *The Pleasure of the Text*, First American Edition. Canada: Harper Collins Canada, Ltd, 1975. Print.

"Behavior." *Merriam-Webster Online Dictionary*. 2010. Web. 13 April 2010.

Beiguelman, Giselle. “Recycled.” *Desvirtual*. 2001. Web. 13 April 2010.

Bernstein, Charles. “The Art of Inmemorability” *A Book Of The Book: Some Works and Projections about the Book & Writing*, Jerome Rothenberg, ed. New York: Granary Books, 1999. Print.

Bolter, Jay David and Richard Grusin, *Remediation: Understanding New Media*. Cambridge, MA: The MIT Press, 2000. Print.

Bootz, Philippe. “Passage” *Alire* 11. 1998. CD-Rom.

--. “Readers” *p0es1s: The Aesthetics of Digital Poetry*. Stuttgart: Hatje Cantz Publishers, 2004. Print.

--. “The Functional Point of View: New Artistic Forms for Programmed Literary Works,” *Leonardo*, 32:4, 1999. Print.

Bowers, Fredson. *Principles of Bibliographical Description.* New Castle, DE: Oak Knoll Press, 1995. Print.

"Brief History of Macromedia Director." *Lingo Workshop*. n.d. Web. 21 July 2008.

Bringhurst, Robert. *The Elements of Typographic Style,* 2nd edition. Vancouver: Hartley and Marks, Publishers. 2002.

Bryant, John. *The Fluid Text: A Theory of Revision and Editing for Book and Screen*. Ann Arbor: U of Michigan P, 2002. Print.

Cayley, John. “Bass Resonance.” *Electronic Book Review*. 11 May 2005. Web. 13 April 2010.

Chang, Young-Hae. *Young-Hae Chang Heavy Industries*. n.d. Web. 13 April 2010.

Cubitt, Sean. *The Cinema Effect*. Cambridge, Mass: The MIT Press, 2004. Print.

Danielewsky, Mark. *House of Leaves*. New York: Pantheon, 2000. Print.

Debord, Guy. "Theory of the Derive." 2006. *Bureau of Public Secrets*. Ed. Ken (translator) Knabb. Web. 1 Nov. 2008.

Doctorovich, Fabio. “Abyssmo.” *Akenaton – Docks.* 1997. Web. 11 April 2010.

Drucker, Johanna. *The Century of Artists’ Books*. New York: Granary Books, 1997. Print.

--. *Figuring the Word: Essays on Books, Writing, and Visual Poetics*. New York: Granary Books, 1998.

--. *Sweet Dreams*. Chicago: U of Chicago P, 2006. Print

--. "Theory as Praxis: The Poetics of Electronic Textuality." *Modernism/Modernity* 9.4 (2002): 683-91. *MLA International Bibliography*. Web. 11 Apr. 2010.

Ferguson, Margaret et. al., eds. *The Norton Anthology of Poetry,* 4th edition. New York: W. W. Norton & Company, 1996. Print.

Funkhouser, C. T. *Prehistoric Digital Poetry: An Archaeology of Forms, 1959-1995*. Tuscaloosa, AL: U of Alabama P, 2007. Modern and Contemporary Poetics. Print.

Fussell, Paul. *Poetic Meter and Poetic Form,* revised edition. Columbus, OH: McGraw-Hill, 1979. Print.

Gioia, Dana. *Disappearing Ink: Poetry at the End of Print Culture*. Minneapolis: Graywolf Press, 2004. Print.

Glazier, Loss Pequeño. “Colibrí.” *Electronic Poetry Center.* 5 May 1999. Web. 11 April 2010.

--. *Digital Poetics: The Making of E-Poetries*. Tuscaloosa, AL: U of Alabama P, 2002. Print.

--. “White-Faced Bromeliads on 20 Hectares.” *Electronic Poetry Center.* 1999. Web. 11 April 2010.

Gomringer, Eugen. "From Line to Constellation." *UbuWeb*. 1954. Web. 1 Nov. 2008.

Hadbawnick, David and Mark Gergis. “Bus.” *Cauldron & Net*, Autumn 2002. Web. 11 April 2010.

Haraway, Donna. "A Cyborg Manifesto: Science, Technology, and Socialist-Feminism in the Late Twentieth Century," *Simians, Cyborgs and Women: The Reinvention of Nature*. New York: Routledge, 1991. Print.

Hartley, George. “Fly.” 2001. Web. 11 April 2010.

Hayles, N. Katherine. “Deeper into the Machine: The Future of Electronic Literature,” *Culture Machine 5: The E-Issue*. 2003. Web. 11 April 2010.

--. “Electronic Literature.” *Writing Machines Web Supplement*. Cambridge, MA: The MIT Press. 2002. Web. 11 April 2010.

--. *How We Became Posthuman.* Chicago: U of Chicago P, 1999. Print.

--. "Print Is Flat, Code Is Deep: The Importance of Media-Specific Analysis." *Poetics Today* 25.1 (2004): 67-90. MLA International Bibliography. Web. 11 Apr. 2010.

--. "Translating Media: Why We Should Rethink Textuality." *Yale Journal of Criticism: Interpretation in the Humanities* 16.2 (2003): 263-90. MLA International Bibliography. Web. 11 Apr. 2010.

--. “Virtual Bodies and Flickering Signifiers,” in Mirzoeff, Nicholas (ed. and introd.); *The Visual Culture Reader*. London, England; Routledge; 2002. Print.

--. *Writing Machines*. Cambridge, MA: The MIT Press. 2002. Print.

Heilmann, Christian. "From DHTML to DOM Scripting." *Web Develoment Articles by Christian Heilmann*. 2006. Web. 1 Nov. 2008.

Hennessey, Neil. “Paddle” *Electronic Poetry Center*. 1998. Web. 13 April 2010.

--. “Puddle” *Electronic Poetry Center*. 1998. Web. 13 April 2010.

Iser, Wolfgang. *The Act of Reading: A Theory of Aesthetic Response*. Baltimore: The Johns Hopkins UP, 1980.

Kendall, Robert. “Faith.” *Cauldron & Net*, Autumn 2002. Web. 11 April 2010.

Kent, Steven L. *The Ultimate History of Video Games*. New York: Three Rivers Press, 2001. Print.

Kirshenbaum, Matthew. "Editing the Interface: Textual Studies and First Generation Electronic Objects." *TEXT: An Interdisciplinary Annual of Textual Studies* 14 (2002): 15-51. Print.

--. “Every Contact Leaves a Trace: Computer Forensics and Electronic Textuality.” *Mechanisms: New Media ant the Forensic Imagination.* Cambridge, MA: The MIT Press, 2008. Print.

--. “Extreme Inscription: Towards a Grammatology of the Hard Drive” *Text Technology: The Journal of ComputerText Processing*, 31:2, 2004. Print.

--. “Machine Visions: Towards a Poetics of Artificial Intelligence” *EBR 6*, Winter 97/98. Web. 11 April 2010.

--. “Materiality and Matter and Stuff: What Electronic Texts Are Made Of.” *Electronic Book Review* 12, Fall 2001. Web. 11 April 2010.

--. *Mechanisms: New Media and the Forensic Imagination.* The MIT Press, 2008. Print.

--. “The Word as Image in an Age of Digital Reproduction.” *Eloquent Images.* Eds. Hocks and Kendrick. Cambridge, MA: The MIT Press, 2003. Print.

Kirschenbaum, et al. “No Round Trip: Two New Primary Sources for *Agrippa*” *The Agrippa Files*. 5 Dec. 2008. Web. 13 April 2010.

Kittler, Friedrich. "There Is No Software." *Stanford Literature Review* 9.1 (1992): 81-90. *MLA International Bibliography*. Web. 11 Apr. 2010.

Knoebel, David. “How I Heard It.” *Click Poetry*. n.d. Web. 13 April 2010.

--. “Thoughts Go.” *Click Poetry*. n.d. Web. 13 April 2010.

--. “Walkdon’t.” *Click Poetry*. n.d. Web. 13 April 2010.

--. “Words in Space,” *Click Poetry*. n.d. Web. 13 April 2010.

Liu, Alan et. al. *The Agrippa Files*. n.d. Web 13 April 2010.

Liu, Alan et. al. “*Born-Again Bits: A Framework for Migrating Electronic Literature*.” *The Electronic Literature Organization*, 5 Aug. 2005. Web. 11 April 2010.

Mac Low, Jackson. *Barnesbook: Four Poems Derived from Sentences by Djuna Barnes*. Washington DC: Sun & Moon P. Print.

Manovich, Lev. *The Language of New Media*. Cambridge, MA: The MIT Press, 2002. Print.

McGann, Jerome. “Dialogue and Interpretation at the Interface of Man and Machine,” *Radiant Textuality: Literature after the World Wide Web.* London: Palgrave Macmillan, 2004. Print.

--. *The Textual Condition*. Princeton: Princeton U P, 1991. Print.

McGovern, Tammy. “Meaning Effect.” *Electronic Poetry Center*. 2001. Web. 11 April 2010.

Montfort, Nick. “Literary Games.” *Poems that GO*. Fall 2003. Web. 11 April 2010.

Ong, Walter J. *Orality and Literacy: The Technologizing of the Word*. London: Routledge, 1982. Print.

Preminger, Alex, et al, eds. *The New Princeton Encyclopedia of Poetry and Poetics. Princeton*, NJ: Princeton UP, 1993. Print.

Queneau, Raymond. *Cent Mille Milliards de Poemes*. Paris: Gallimard, 1961. Print.

Rosenblueth, Arturo, Norbert Wiener and Julian Bigelow. "Behavior, Purpose, and Teleology" *Philosophy of Science,* Vol. 10, No. 1 (Jan., 1943). Web. *JSTOR*.

Schaffner, Anna Katharina. "From Concrete to Digital: The Reconceptualization of Poetic Space." *Vispoets*. 9-10 November 2006. Web. 7 September 2008

Shillingsburg, Peter L. *Resisting Texts: Authority and Submission in Constructions of Meaning*. Ann Arbor, MI: U of Michigan P, 1997. Print.

--. *Scholarly Editing in the Computer Age: Theory and Practice*. Ann Arbor, MI: U of Michigan P, 1996. Print.

Shklovskij, Viktor. "Art as Technique." *Literary Theory: An Anthology*. Ed. Julie Rivkin and Michael Ryan. Malden: Blackwell Publishing Ltd, 1998. Print

Simanowski, Roberto. "Becoming a Full Time Web Artist" Interview with Jim Andrews. 29 January 2002. Web. 2 Nov. 2008.

Smith, Martha Nell. "Letter-Poem: A Dickinson Genre," *Dickinson Electronic Archives*, 1999. Web. 11 April 2010.

Soderman, Braxton. "Machine Poetics" *Page_Space Project*. 2005. Web. 11 April 2010.

Solt, Mary Ellen. *Concrete Poetry: A World View*, reprinted in UbuWeb. n.d. Web. 11 April 2010.

Stefans, Brian Kim. "The Dreamlife of Letters." *Arras.net.* 26 April 2001. Web. 11 April 2010.

--. "I Know a Man, One Letter at a Time," *Arras.net*. n.d. Web. 11 April 2010.

Tanselle, G. Thomas. *A Rationale of Textual Criticism*. Philadelphia: U of PA P, 1992. Print.

Thuan, Duc. "Chronicle of Deaths Forgotten." *Ducthuan.com*. 2002. Web. 13 April 2010.

Webartery. Yahoo Groups. Web. 13 April 2010.

Wikipedia contributors. "Computer graphics (computer science)." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 22 Jun. 2006. Web. 13 Apr. 2010.

--. "Control flow." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 19 Jun. 2006. Web. 13 Apr. 2010.

--. "Object-oriented programming." *Wikipedia, The Free Encyclopedia*. 21 Feb. 2010. Web. 24 Feb. 2010.

--. "Porting." Wikipedia, The Free Encyclopedia. Wikipedia, The Free Encyclopedia, 5 May. 2006. Web. 13 Apr. 2010.

--. "Reading (action)" *Wikipedia*. n.d. Web. May 29, 2006.

Williams, Emmett. *A Valentine for Noel: Four Variations on a Scheme*. Barton, VT: Something Else Press, 1973. Book.

World Wide Web Consortium. "Document Object Model FAQ." *W3C Document Object Model* 22 Jan. 2008. Web. 1 Nov. 2008.