

## ABSTRACT

Title of thesis: **THE ROLE OF 3D SPATIOTEMPORAL  
TELEMETRY ANALYSIS IN COMBAT FLIGHT  
SIMULATION**

Sourabh Mane, Master of Science, 2024

Thesis directed by: **Dr. Niklas Elmqvist  
College of Information Studies**

Analyzing 3D telemetry data collected from competitive video games on the internet can support players in improving performance as well as spectators in viewing data-driven narratives of the gameplay. In this thesis, we conduct an in-depth qualitative study on the use of telemetry analysis by embedding over several weeks in a virtual F-14A Tomcat squadron in the multiplayer combat flight simulator DCS World (DCS) (2008). Based on formative interviews with DCS pilots, we design a web-based game analytics framework for rendering 3D telemetry from the flight simulator in a live 3D player, incorporating many of the data displays and visualizations requested by the participants. We then evaluate the framework with real mission data from several air-to-air engagements involving the virtual squadron. Our findings highlight the key role of 3D telemetry playback in competitive multiplayer gaming.

THE ROLE OF 3D SPATIOTEMPORAL TELEMETRY ANALYSIS IN  
COMBAT FLIGHT SIMULATION

by

Sourabh Mane

Thesis 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  
Master of Science  
2024

Advisory Committee:  
Dr. Niklas Elmqvist, Chair/Advisor  
Dr. Joel Chan  
Dr. Caro Williams-Pierce

© Copyright by  
Sourabh Mane  
2024

## Acknowledgments

This thesis was made possible by timely help, encouragement, and support from faculty, staff, my HCIM seniors, friends, Discord buddies, my family, and my cat.

First and foremost, I'd like to thank Dr. Niklas Elmqvist for giving me the opportunity to work on this project. I would not have the niche expertise that I possess today if not for this research. You have supported and mentored me through various roles—wearing multiple hats as a professor, my supervisor, a domain expert, and now as my advisor. Thank you for fostering a collaborative research atmosphere through the data visualization lab; I learned a lot from each senior.

I am thankful to Dr. Joel Chan and Dr. Caro William Pierce for kindly agreeing to be on my examining committee. I'm grateful for their patience and valuable feedback from my first brown bag lunch at HCIL to my thesis defense. I'd also like to thank Dustin, Tatyana, Carol, and the rest of the staff and faculty at the iSchool for their support since the beginning of my journey.

I would like to acknowledge the financial support from the UMD College of Information Studies received through the Research Improvement Grant. Additionally, I would like to thank Susan Winter for her support during the process.

My special acknowledgment goes to Dr. Wayne Lutters, Professor Alex Leitch, and AJ Rudd, who *talked me into* pursuing a master's thesis. All of you were right—it is difficult, and it is definitely worth it. Thank you to the VF-1 Wolfpack squadron for being welcoming, and to all the participants who took part in this study.

Finally, I would like to extend my gratitude to my friends and family. My mother, Dr. Shailaja, for teaching me when to push hard, and my father, Dr. Vijaykumar, for showing me the value in knowing when not to. Thanks to my uncles, aunts, and siblings for being supportive and taking an interest in my research. Much appreciation to Sakham Verma for providing different perspectives on this project throughout its entirety.

## Table of Contents

Acknowledgements	ii
1 Introduction	1
1.1 Thesis Statement	2
1.2 Approach	2
1.3 Outcomes	3
2 Background	4
2.1 Game Telemetry	4
2.2 Visualization and Games	6
2.3 Spatiotemporal Visualization	8
2.4 Related work and tools	10
2.4.1 Visualizing Real-Time Strategy Games	10
2.4.2 Visualeague II Tool	11
2.4.3 StratMapper Tool	12
2.4.4 Spatio-Temporal Player Profiling in Open-World Games	14
2.5 Summary	14
3 Case Study: Virtual Squadrons in DCS World	16
3.1 Multiplayer Support	17
3.2 Telemetry Collection and Display	18
3.3 Virtual Squadrons	19
3.4 VF-1 Wolfpack	20
4 Formative Study	24
4.1 Participants	24
4.2 Method	25
4.3 Results	25
4.4 Implications	28

5	TacReel: 3D Telemetry Visualizer	29
5.1	Use Cases . . . . .	30
5.2	Constraints . . . . .	31
5.3	Data Model: 3D Telemetry Persistence . . . . .	32
5.4	3D Telemetry Player . . . . .	33
5.5	Event Timeline . . . . .	34
5.6	Reel Director . . . . .	37
5.7	System Architecture . . . . .	39
5.7.1	Backend . . . . .	39
5.7.2	Frontend . . . . .	41
5.8	Performance . . . . .	41
5.8.1	Browser Compatibility . . . . .	42
6	Summative Study	44
6.1	Participants . . . . .	44
6.2	Method . . . . .	44
6.3	Results . . . . .	47
7	Discussion	52
7.1	Explaining the Results . . . . .	53
7.2	Generalizing the Results . . . . .	55
7.3	Limitations . . . . .	56
7.4	Future Work . . . . .	57
8	Conclusion	59
A	Formative interview questions	60
B	Summative Evaluation Protocol	61
	Bibliography	62

## List of Figures

2.1	<b>Screenshot of Star Craft 2 visualization system [27]</b> . . . . .	10
2.2	<b>Screenshot of Visualeague II visualization tool prototype [1]</b> . . . . .	11
2.3	<b>Screenshot of StratMapper interface [2]</b> . . . . .	12
2.4	The trails of Just Cause 2 [5] . . . . .	13
2.5	<b>Game object movement across plane</b> Game object P moves from P1 to P2. Path a takes route that is above the plane, Path b along the plane, Path c below the plane . . . . .	15
3.1	<b>Grumman F-14A Tomcat.</b> A VF-1 Tomcat fighter entering the pattern for a CASE I recovery on CVN-71, the <i>U.S.S. Theodore Roosevelt</i> , while deployed in the Persian Gulf. (Image by P04 from <i>DCS World</i> .) . . . . .	20
5.1	<b>TacReel player.</b> . . . . .	29
5.2	<b>Mission replay using TacReel.</b> . . . . .	31
5.3	<b>Data visualization pipeline. [12]</b> . . . . .	31
5.4	<b>Target Pairing.</b> Pink line shows the distance between player and target in nautical miles. . . . .	34
5.5	<b>Spatial event generation.</b> . . . . .	36
5.6	<b>Director tool with event timeline.</b> The timeline is organized with units on the vertical (Y axis) and time on the horizontal (X axis). Colored boxes represent events detected by the rule engine. Users can create custom events by brushing the timeline. . . . .	38
5.7	<b>Highlight reel.</b> TacReel player showing a highlight reel of multiple clips organized into a sequence. . . . .	39
5.8	<b>TacReel system architecture.</b> . . . . .	40
5.9	<b>Update function optimization</b> . . . . .	42
5.10	TacReel inside Apple Vision Pro’s Safari browser. . . . .	43
6.1	<b>Summative evaluation missions.</b> Our team was embedded during two missions: a fighter sweep in the Caucasus map (left), and a carrier qualification event on the <i>U.S.S. Theodore Roosevelt</i> . . . . .	45
7.1	<b>XR Continuum. [36]</b> . . . . .	58

7.2	Steps followed in breaking the spatial arena of a game into semantic zones.	
[32]	.....	58

## Chapter 1: Introduction

Video games are becoming increasingly complex and accordingly require increasingly sophisticated skills and tactics. This is doubly true for online multiplayer games, particularly those that pit players against other players in so-called Player vs. Player—or “PvP”—combat because the opponent is not an artificial but a real intelligence. For example, in the multiplayer game *Battlefield V* (2018), the players take the role of World War II soldiers from both Axis and Allied sides and battle it out on historic battlegrounds; in *Fortnite* (2017) the players fight solo or in teams against a hundred other people in a shrinking area in the format known as “Battle Royale” (after the 1999 namesake novel written by Koushun Takami); and in *World of Warcraft* (2004), players become characters of fantasy classes such as paladins, wizards, and archers to fight in either open-world or arena-based PvP. In all of these examples, as well as in countless others on the internet, there is an opportunity for collecting, processing, and analyzing potentially vast amounts of 3D telemetry data to help players improve their skills and coordination. In fact, in the burgeoning esports industry, where players compete professionally in organized multiplayer competitions, such *game analytics* [13–15] may be the difference between a win and a loss, which could amount to thousands of dollars in prize money. Similarly, as gameplay—professional and casual alike—is increasingly streamed online on Twitch and Youtube, the boundary between electronic and traditional sports is becoming blurred and the demand for robust data

collection and display is increasing.

In this thesis, we study 3D real-time telemetry for post-game analysis in the context of the combat flight simulator *DCS World* (Digital Combat Simulator) (2008).

## 1.1 Thesis Statement

Studying the needs of a flight simulator community. Building and evaluating a 3D telemetry data exploration tool to support these requirements.

## 1.2 Approach

Embedding in a DCS F-14A Tomcat virtual squadron over a period of several weeks, we conducted an in-depth observational and interview study to understand the data collection and analysis practices of a highly competitive group of flight simulator players. While our approach is applicable to any multiplayer 3D game, combat flight simulation is particularly amenable to this kind of telemetry data analysis because extensive debriefings tend to be routine for many so-called “milsim” (military simulation) communities. During the debriefings, squadron members pore endlessly over individual maneuvers to give feedback and help each other improve. In response to a formative evaluation with DCS players, we derived requirements for 3D telemetry analysis and propose a suite of tools called TACREEL for collecting, visualizing, and analyzing this kind data. TacReel caters both to players analyzing their performance by replaying a game session as well as to spectators viewing a data-driven narrative summarizing the session. In the playback mode, TacReel provides situated visualizations to help the players orient themselves and analyze the telemetry data. In the summary mode, TacReel uses author-specified rules to

automatically detect events and generate compelling highlight reels summarizing a long session. We report on a summative evaluation involving members of the virtual squadron and collect their feedback on using TacReel for their mission data.

### 1.3 Outcomes

In this thesis, we first give a background on game analytics and its use of visualization and visual analytics. We then discuss the multiplayer gaming problem domain and its data management, analytics, and visualization needs. We describe DCS and VF-1 Wolfpack, the virtual DCS World squadron flying the Grumman F-14A Tomcat fleet defense interceptor that we primarily embedded in for our work. We present results from formative interview sessions with DCS players, including one who is also a professional combat pilot. Based on these findings, we then present the TacReel tool, its visualization components, and its deployment in DCS World. Finally, we present findings from a summative evaluation where squadron members used the tool for their training debriefings. We close by deriving implications and discussing directions for future research.

## Chapter 2: Background

*Video game analytics* [13–15, 33] refers to the process of collecting, processing, and analyzing data generated by player interactions within video games. This data encompasses a wide range of information such as player behavior, in-game events, performance metrics, in-game purchases, and player statistics. For this reason, game analytics is often a data science problem [15]. In this section, we review the background on video game analytics, focusing in particular on 3D telemetry data. We begin by discussing the larger area of game analytics, which covers many different topics and audiences, including game developers, designers, and marketers, but also the players themselves and the spectators who enjoy watching them play. We then discuss the emerging role of visualization for game analytics. Finally, we present the specific visualization techniques that are relevant to real-time 3D telemetry data: spatiotemporal visualization.

### 2.1 Game Telemetry

*Telemetry* can loosely be thought of as *remote measurement*: the automatic collection and transmission of data from a remote point of interest [10], such as a space, airborne, or underwater vehicle. In the context of video games, telemetry can be used to refer to any data collection from a running game [13, 34], but for the purposes of this article we will use it in the spirit of classic vehicle telemetry: real-time player behavior in the form of

quantitative spatiotemporal metrics such as position, orientation, and velocity, as well as discrete events such as shooting, reloading, changing posture, etc.

Many video games capture real-time player behavior at high precision. It is particularly true for first-person shooters (FPS) such as *Counterstrike: Global Offensive* (CS:GO) (2012), *Halo: Infinite* (2021), or *Call of Duty: Modern Warfare II* (2022), but also for real-time strategy (RTS) games and Multiplayer Online Battle Arena (MOBAs) games such as *Starcraft II* (2010), *Defense of the Ancients 2* (DotA 2) (2013), and *League of Legends* (LoL) (2009). FPS telemetry goes back to the original *Doom* (1993) by id Software, which used a special “lump” file format (.lmp) that was essentially a time-stamped stream of player inputs,<sup>1</sup> allowing a game session to be perfectly recreated with little storage requirements.

There are several legitimate and well as illegitimate uses of real-time 3D telemetry. Game developers use aggregated telemetry data to identify concerns and balance gameplay, such as areas where the polygon count is too high; bugs and glitches in logic, levels, or 3D models; or places in a level where the player is too vulnerable or too protected [6]. During esports events broadcast to an in-person or virtual audience, telemetry can be used to generate a spectator view that shows more information than what the individual players see, such as the name, team, and location of all combatants in the level. Also, in-game replay theaters and online replay aggregators such as GameReplays use telemetry to recreate post-mortem recordings of a match, often with the added benefit that the player can freely navigate both in time as well as in the 3D space of the level. However, the ability to extract exact telemetry for all enemy players can also be used by cheaters, for example by making the level’s walls semi-transparent (wallhack) or even automatically aiming and shooting at

---

<sup>1</sup>[https://doom.fandom.com/wiki/Demo#Technical\\_information](https://doom.fandom.com/wiki/Demo#Technical_information)

the enemy with inhuman speed and accuracy (aimbot). For this reason, anti-cheat software such as Easy Anti-Cheat, BattlEye, and Denuvo use—among other things—telemetry to detect tampering and cheating.

Flight simulators are a special case because of the desires for many simulator pilots to study their performance after a flight to understand mistakes, how to address them, and how to improve. Thus, many flight simulators, such as *Microsoft Flight Simulator* (first released in 1982; most recent release in 2020), *X-Plane* (first released in 1995; most recently in 2022), *Falcon BMS* (first released as *F-16 Fighting Falcon* in 1987; BMS from 2012), *IL-2 Sturmovik* (first released in 2001; most recent release in 2013), and *DCS World* (2008), include built-in telemetry flight recording and replay functionality.

In addition, the third-party tool *Tacview*<sup>2</sup> (first released in 2006) uses a universal telemetry file format called ACMI,<sup>3</sup> and comes with exporters for most popular simulators (the ones listed above among others). Tacview is a native Windows application based around a 3D map view of the world. Once installed, the tool will automatically record telemetry for any flight the player takes on that computer. It is extensively used by DCS players in particular.

## 2.2 Visualization and Games

Several comprehensive surveys have been presented on the state of data visualization in video games. Medler and Magerko [35] conducted a study encompassing a wide array of currently available games on the market. Building upon the foundation of casual information visualization [37], they introduced the concept of “playful visualization,” which involves using data visualization to enhance and encourage gameplay. Bowman et al. [6]

---

<sup>2</sup><https://www.tacview.net/>

<sup>3</sup><https://www.tacview.net/documentation/acmi/en/>

presented a design framework that explores visualization research problems as they apply to video games. This framework presents five categories that can be used to classify specific visualization technique found in video games:

- **Primary purpose:** Intended use;
- **Target audience:** Intended audience;
- **Temporal usage:** Timing of use;
- **Visual complexity:** Level of visual sophistication of the visualization in the game;
- **Immersion/integration:** Whether the visualization is in spirit with the game and whether it is inside the game or not;

The authors derive several common design patterns from an extensive survey of games, such as *Heads-up Display*, *Replay Theatre*, *Progression Tree*, *Data in Space*, *Data in Time*, etc. From these patterns “data in space, data in time & replay theatre” are precisely what our proposed visualization intervention in this thesis supports.

Finally, Wallner and Kriglstein [39] review the literature on visualization-based analysis of game metric data, providing an overview of the current state in this field, categorizing visualization techniques, highlighting their characteristics, and identifying open research challenges for the future.

While visualization has made significant inroads into the games industry, the contributions to the visualization scientific community are still relatively few. Halper and Masuch [18] concentrate on the extraction of noteworthy highlights from real-time first-person shooter footage for the benefit of spectators. Data Cracker [34] is a web-based information dashboard tool designed for the multiplayer variant of *Dead Space 2* (2010). Li et al. [29]

present a multi-view dashboard showing event sequences in a MOBA over time, using it to explain comeback behavior.

The burgeoning esports genre particularly benefits from data-driven displays and visualizations [26]. Charleer et al. [8] design information dashboards for both *LoL* and *CS:GO* esports events, finding that they improve spectator insight and experience but increase the viewing complexity and the resulting cognitive load. Ahmad et al.’s interactive behavior analysis method [2] combines a map with an event sequence visualization to help esports players improve. Kleinman et al. [24] build on this to present a discrete event timeline that, when coupled with an interactive map, can help externalize teamwork in multiplayer games such as *DotA 2*. Finally, Wallner et al. [41] present an in-depth study of player needs for post-game visualizations; the guidelines particularly relevant to our work include careful data curation, integrating many data sources, highlighting high-level gameplay concepts, and summarizing in-game events.

### 2.3 Spatiotemporal Visualization

Real-time 3D telemetry is best visualized using spatiotemporal visualization techniques [4]. Accordingly, there has been a significant adoption of spatiotemporal visualization in many video games, online as well as off. For example, the original *Sim City* and *Populous*, independently released by legendary game designers Will Wright and Peter Molyneux in 1989, both incorporate forms of in-game bird’s eye views and heatmaps to aid the player in their decision-making.

Some of these spatiotemporal visualizations have made the leap to academic research, either adopting techniques for real-time game telemetry, or even using the game as a springboard for innovation. Hoobler et al. [21] employ localized visual elements such as player

symbols, paths, and traces in space to depict team-oriented competitive gameplay within multiplayer first-person shooter games. G-Player [7] uses heatmaps and event plots to facilitate the understanding of player movement in time and space. Drawing inspiration from classic battle maps from military history, Wallner [38] present an approach to create a static visual summary of engagements in the online game *World of Tanks* (2010). Kuan et al.'s visualization system [28] uses a similar approach to summarize battles in *StarCraft II* (2010) as static map overviews with arrows showing movement and troop concentrations as colored fields. VisuaLeague II [1] enables the analysis of *LoL* game data through dynamic spatiotemporal visualization, comparing its performance to the *LoL* replay system as well as the online website OPGG. Building on the popularity of hexagonal maps in board games such as *Gettysburg* (1958) and *Catan* (1995), which later was adopted in video games such as *Steel Panthers* (1995) and *Civilization V* (2010), hexbin maps [40] are a spatial binning technique for aggregating spatial data and then visualizing it using color-coded glyphs. CS:Show [22] is a match analysis tool for *CS:GO* that uses dynamic heatmap representations of game levels to support esports coaching. Finally, ggViz [42] presents visual representations of player tracking data in *CS:GO* to enable users to query a vast esports dataset through game state sketches, addressing the gap in data-driven analysis for esports.

Building on many of these spatiotemporal gameplay visualizations, Kleinman et al. [25] presents results from a qualitative user study with experienced *DotA 2* players, yielding an early version of a taxonomy for user interaction with spatiotemporal game data that consists of seven activities organized into three categories: data interaction, sensemaking, and validation. Our proposed TacReel tool supports actions drawn from each of these three categories and seven activities, specifically - . However, to our knowledge, no existing

work targets flight simulators and competitive online air combat.

## 2.4 Related work and tools

### 2.4.1 Visualizing Real-Time Strategy Games

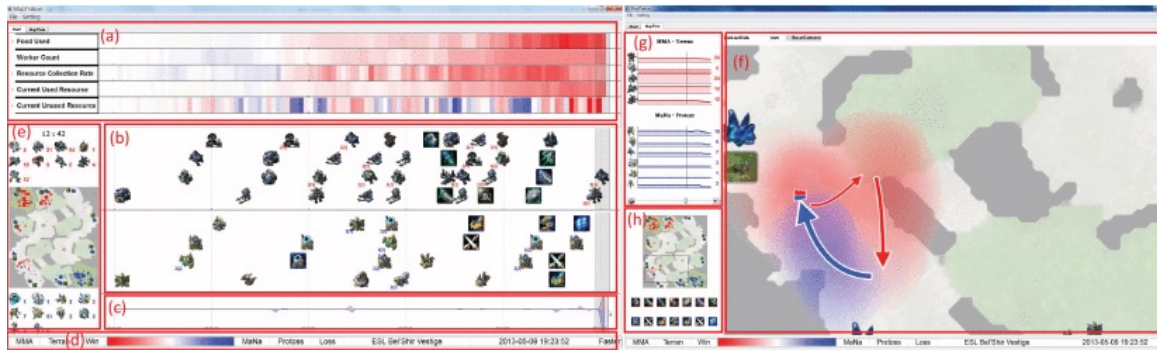


Figure 2.1: Screenshot of Star Craft 2 visualization system [27]

Previous work by Yen-Ting Kuan Et al [27] presents a visualization system for users to analyze real-time strategy games. The system visualizes movements of army units belonging to respective players on a map. The system allows exploration through multiple views. In the *global view*, battles are showing by using curved arrows as marks. Through these marks and the height map of the terrain, users can make sense of offensive and defensive strategies. In the *local view*, each unit of different types represented on the map, and their movements are shown through animation. The system also renders attack line between pairs of units which help users analyze the micro and macro strategies. In *status view* users can view build order, and unit deaths to seek to interesting events. Lastly the *battle view* shows a map to describe the process of a battle. Users pointed out in the study that they had to manually tune the level of details of armies' movements and the arrow size, related to the zoom level, they expected the parameters to be automatically determined. This caused

occlusions of curved arrows on the map as sometimes the arrows are too large or bent to avoid passing through obstacles.

## 2.4.2 Visualeague II Tool



Figure 2.2: Screenshot of Visualeague II visualization tool prototype [1]

Visualeague II [1] is a visualization tool for analysis of League of Legends (LoL) game's statistical and spatio-temporal data for players, teams, single and matches. The VisuLeague II tool focuses on the *animated map* visualization technique, which covers 80% of the screen, showing players positions, pathing/trajectory progress and events like players deaths, towers destruction etc. The remaining screen has displays for static and statistical data which includes - *Filter Panel, Timeline, Mini Map & an Information Panel*. Their user study performed showed that the participants prefer to visualize spatio-temporal data dynamically, i.e., using animated maps or the replay system. Furthermore, it also showed different ways in which the same information is analysed by casual players and

professionals, which suggests importance to study the possible design implications these may bring.

### 2.4.3 StratMapper Tool

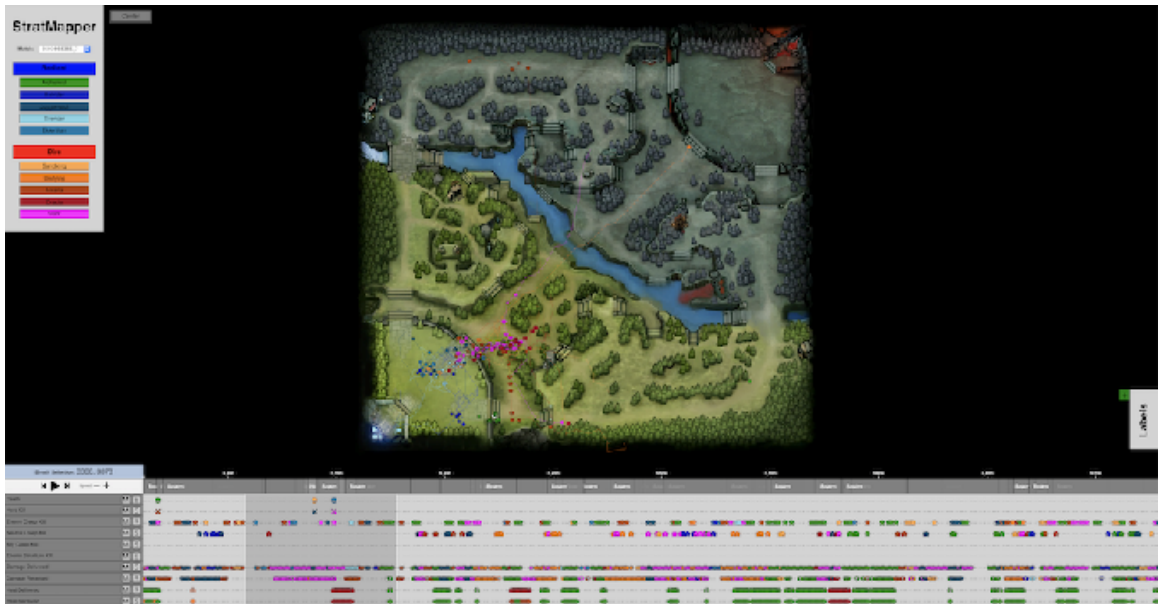
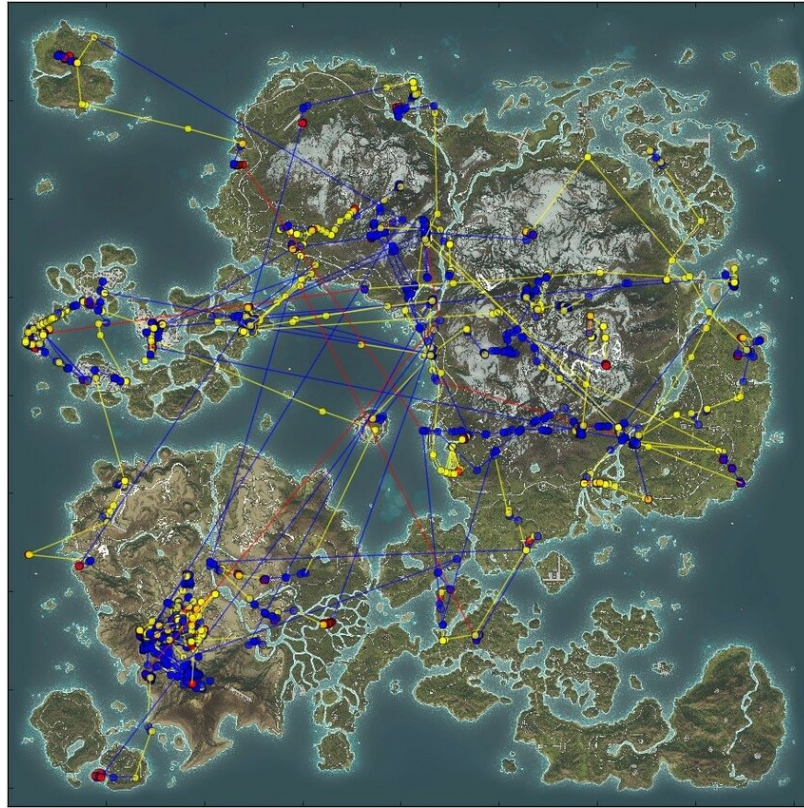
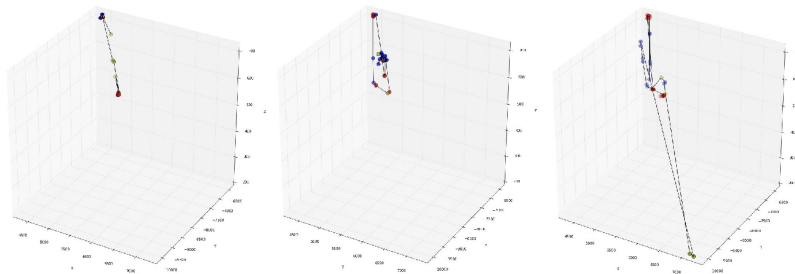


Figure 2.3: Screenshot of StratMapper interface [2]

Stratmapper [2] is a spatio-temporal visualization system developed in the GUII lab. StratMapper, similar to the other systems described earlier, visualizes players' movement as well as game variables over time and over spatial maps focusing on two games Boom-Town and DOTA 2. However, Stratmapper goes further and allows space-time constraint modifiers that allow users to apply spatial and temporal constraint on queries, limiting the amount of data visualized to a small set. In the *timeline* users can select a temporal period of interest with the brush component. Events listed on this timeline can also be filtered based on the event type. The paper also presents a new methodology, called Interactive Behavior Analytics (IBA), comprised of two visualization systems, a labeling mechanism, and



(a) Example of a typical player trajectory from DEDICOM-generated cluster



(b) Player trajectories of different behavioral profiles for the Mountain Rescue mission in Just Cause 2

Figure 2.4: The trails of Just Cause 2 [5]

abstraction algorithms packaged in a seamless interface to facilitate knowledge discovery from game data.

#### 2.4.4 Spatio-Temporal Player Profiling in Open-World Games

This study by Myat Aung Et al.[5] focuses on behavioral profiling of players in digital games analytics, specifically in Open-World Games. In Open-World Games incorporating the spatial and temporal dimensions of player behavior is necessary when profiling behavior, as these dimensions are important to the playing experience. The study analyses the behavior of more than 5,000 players of the major Open-World game - Just Cause 2 . It further presents analyses that apply cluster analysis and the DEDICOM decompositional model to profile the spatio-temporal behavior of players shown in Figure 2.4. Through this is displays the applicability of spatio-temporal profiling to condense player behavior across large sample sizes. The method presented in this study provides a means to build profiles of player activity in game environments with high degrees of freedom. The results of this study may be specific to one game, but they provide a means to build profiles based on player activity in unrestricted open-world game environments. For any problem space that contains a large range of spatial, temporal and varied data, the combinative methods used in this study are suitable, allowing player profiling to be adapted to a variety of situations, with examples including detailed analysis of the performance and positional tactics of teams.

#### 2.5 Summary

In all the previous mentioned work the tools focus on games that are three dimensional, however their third dimension is used only to support the aesthetics [23] of the underlying gameplay. When the dimensionality is reduced and the gameplay represented in the respective tool's visualization, the dimension of height/altitude/Z-axis/ZY plane/ZX plane

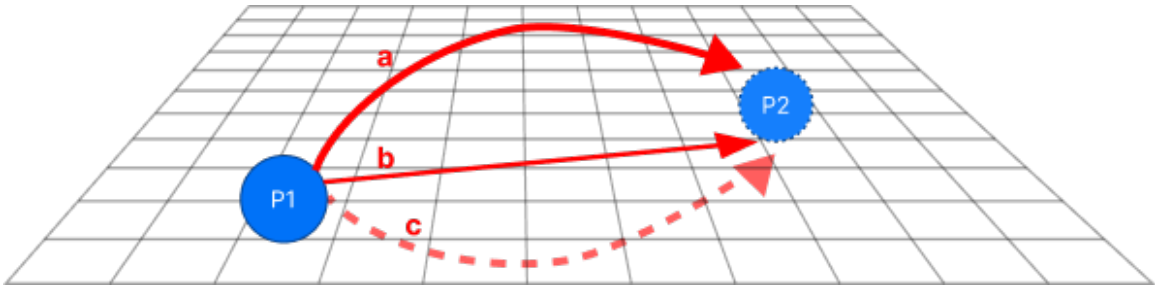


Figure 2.5: **Game object movement across plane** Game object P moves from P1 to P2. Path a takes route that is above the plane, Path b along the plane, Path c below the plane

is lost. In no above mentioned research does this impact the user's understanding of the ongoing scenario. This is explained using figure 2.5, the information about the path ( a, b, c) that any game object or player object ( P1 ) takes is of little to no importance compared to the objective destination ( P2) it arrives over time. Suggesting that when the information about the spatiotemporal path needs to be explored, these tools can not support the third dimension of movement. Building upon the previous research and user studies of various tools, this thesis aims to build a data visualization tool that can support 3D spatiotemporal telemetry data.

### Chapter 3: Case Study: Virtual Squadrons in DCS World

*Digital Combat Simulator World* (2008), often known merely as DCS, is a comprehensive combat simulator for PC focused on high realism. The game is developed by Eagle Dynamics, a Russian/Swiss game developer. Initially introduced as *Lock On: Modern Air Combat* in the early 2000s, it has since expanded to cover various historical eras, offering a wide range of combat aircraft and immersive environments. DCS has gained popularity among aviation enthusiasts and military professionals, and its user community continues to grow. As a digital battlefield software, the core DCS game is free to play and includes two free aircraft—the Su-25 Frogfoot (a Soviet-era attack jet) and the TF-51 Mustang (an unarmed trainer version of the famous U.S. WWII fighter). However, additional full-fidelity aircraft, known as “modules,” are generally paid and cost around \$80 a piece, featuring realistic flight models, fully clickable cockpits, and painstakingly modeled weapons and sensor systems. Some of the modern-day aircraft include the F/A-18C Hornet, the AJ 37 Viggen, the F-16 Fighting Falcon (known as the “Viper” among its pilots), the F-15E Strike Eagle (the “Mudhen”), and the mighty F-14 Tomcat (Figure 3.1). DCS also includes a number of rotary-wing aircraft, like the AH-64D Apache, the Mi-8 Hip, and the venerable UH-1 Huey, as well as World War II “warbirds,” such as the de Havilland Mosquito light bomber, the Focke-Wulf Fw 190A “Anton,” and, of course, the iconic Supermarine Spitfire.

The simulator is considered “study-level,” which means that players are expected to

have to read detailed manuals to perform even simple procedures, ranging from merely starting up or shutting down the aircraft, to taking off and landing as well as engaging weapons. Accordingly, many players purchase expensive computer peripherals to best mimic the real-world experience of flight, including flight sticks and throttles, multi-function displays and rudder pedals, and even seat vibrators and full-motion platforms. Hardcore enthusiasts sometimes go so far as to build 1:1 replica cockpits. Many players use VR to find themselves inside a fully immersive 3D cockpit. Those that do not have access to VR at least use head tracking (using a webcam or dedicated hardware such as TrackIR) with a regular monitor to control their viewpoint merely by moving their head (although the screen obviously does not move).

### 3.1 Multiplayer Support

While most DCS players use the simulator in single player mode with computer-controlled allies and enemies, DCS also has full support for online servers where multiple players—sometimes as many as 64 at a time—can connect to fly together or fight. At the time of writing (February 2024), there are currently more than 1,500 public DCS servers. The basic DCS server setup has two factions: a red (often Soviet or Russian bloc) and a blue (NATO and/or Western countries). Each server has its own server rules and parameters; some support only player vs. environment (PvE) combat against a computer-controlled faction, whereas others enable full PvP combat against other players. There are even servers dedicated to air acrobatics. Another dimension is the level of military verisimilitude; for example, many public PvP servers essentially facilitate “air quake” deathmatches using combat aircraft, whereas others focus on realism.

Key for multiplayer coordination in DCS is communication. This is mostly facilitated

using SimpleRadio Standalone DCS (SRS-DCS), a free third-party tool that interfaces with the in-game radios of each aircraft to enable voice-over-IP communication between players. SRS models radio parameters such as line of sight, range, and frequencies to provide an immersive communication environment. Furthermore, not all players on a multiplayer DCS server fly aircraft. Players can also take the role of GCI/AWACS (Ground-Controlled Interception/Airborne Warning And Combat System) operators, directing friendly fighter to enemy air threats or commanding a strike, or as ground-based JTACs (Joint Terminal Attack Controllers) directing the close-air support (CAS) aircraft attacking ground targets such as tanks or troops. When no human GCI or JTAC is around, these roles are often managed by voice-controlled third-party software that interface with the game using SRS. Many servers also provide ATC (Air Traffic Control) services for administrative flight operations, such as coordinating take-offs and landings around airfields and aircraft carriers.

Multiplayer functionality has also given rise to a small but determined esports scene within DCS. The SATAL series of events pit teams of players in multi-aircraft engagements under specific rules of engagement. However, because of the in-depth nature of the simulator, DCS lacks the broad appeal that more popular esports enjoy. Analogously, on the other end of the spectrum is the DCS Academy, a group of experienced players dedicated to teaching fundamental aviation and military tactics through an in-depth, hardcore training program spanning several weeks.

### 3.2 Telemetry Collection and Display

The ability to collect and view real-time 3D telemetry is key for many training activities in DCS. While Eagle Dynamics does not provide any specialized data collection and visualization functionality inside the game beyond a simple mission replay, this mechanism is

instead provided by Tacview.

The Tacview tool itself is basically a 3D map viewer that can visualize real-time telemetry of many objects—aircraft, ships, and ground units—over time in time and space. The tool has support for many different unit types (icons representing both NATO and Soviet/Russian aircraft), view settings (first-person, top-down, 3D orbiting camera, tactical, etc), as well as data displays (including speed, altitude, G-force, aspect ratios, distance, etc). It also has native support for screen-sharing so that the presenter can stream a clutter-free display during debriefings.

### 3.3 Virtual Squadrons

For players who want to lean into the social aspect of multiplayer gaming, there are DCS “virtual squadrons” that represent stable groups of players flying together under common goals and values. Virtual squadrons are focused on specific missions, such as aerial acrobatics, ground attack, air-to-air-combat, or even transport missions (such as with the C-130 Hercules or any of the available transport helicopters such as the UH-1 or the UH-60 Blackhawk). Often these squadrons fly a common airframe under a semi-formal order of battle and with various training and certification protocols. Many take the name of a real or fictional squadron drawn from air forces or navies around the world.

Like many social gaming organizations, DCS virtual squadrons typically span many different forms of internet media. Discord tends to be the main chat and voice communication platform, which is consistent with many gaming communities. SRS-DCS tends to be used for in-game radio communication. GCI, JTAC, and ATC operators often use the DCS World client itself in the F10 “map mode” together with SRS to communicate with their assigned aircraft, but the specialized LotAtc tool provides a dedicated GCI/ATC interface

for this purpose. Some squadrons also have websites, discussion forums, and Facebook groups. Because of the prevalence of study-level documentation in the DCS world, many squadrons use Google Docs or other filesharing services to distribute written materials to their pilots.



Figure 3.1: **Grumman F-14A Tomcat.** A VF-1 Tomcat fighter entering the pattern for a CASE I recovery on CVN-71, the *U.S.S. Theodore Roosevelt*, while deployed in the Persian Gulf. (Image by P04 from *DCS World*.)

### 3.4 VF-1 Wolfpack

VF-1,<sup>1</sup> the *Wolfpack* (callsign Wichita), is a DCS World virtual squadron. The squadron considers itself “semi-milsim” (military simulation), which means that its procedures are inspired by—but do not strictly follow—real military protocol. In the case of VF-1, this means that the squadron attempts to emulate actual U.S. Navy combat tactics, but take a relaxed approach to other aspects of Navy life, such as ranks, roles, and duties. VF-1 was selected for this study because one of the authors has been an active member in the squadron for two years.

---

<sup>1</sup>Names and designations for the units and pilots described in this thesis have been changed to preserve the anonymity of the participants.

**Aircraft.** The squadron currently flies the Grumman F-14A Tomcat (Figure 3.1), an all-weather interceptor and air superiority fighter operated by the U.S. Navy from 1970 to 2006. Made famous by the movie *Top Gun* (1986), the Tomcat is one of the most iconic combat aircraft of all time, characterized by its dual engines, twin vertical stabilizers, variable-sweep wings, and two-man crew: a pilot (e.g. Maverick) and a radar intercept operator (RIO; e.g. Goose). The DCS module for the F-14 (both A and B variants) is designed by Heatblur Simulations, a third party vendor, and is widely regarded as one of the highest quality and most accurate aircraft modules in DCS World.

**Organization.** As a Tomcat squadron, VF-1 has a full complement of 13 aircrews, each one consisting of a pilot and a RIO. The squadron is commanded by a commanding officer (CO), supported by an executive officer (XO). At the time of this research, the squadron had 9 active aircrews and 5 replacement/reserve/training ones. The squadron itself is part of CVW-1, a virtual Carrier Air Wing formed by several virtual squadrons based on the same carrier. CVW-1 is stationed on the U.S.S. *Theodore Roosevelt* (CVN-71), a *Nimitz*-class supercarrier, and also includes two F/A-18C Hornet strike fighter squadrons. (The F/A-18 is a DCS aircraft module designed by Eagle Dynamics themselves.)

**Missions.** The Tomcat's—and thus also VF-1's—primary mission is *fleet defense*: the protection of U.S. naval assets—such as aircraft carriers, ships, and other aircraft—from potential threats in order to maintain maritime superiority. This mission includes employing a variety of defensive strategies, technologies, and tactics to counteract threats such as enemy ships, submarines, aircraft, and missiles. In the context of the F-14 Tomcat and VF-1, this primarily entails air-to-air combat operations against hostile air assets using the powerful AIM-54 Phoenix, a long-range active radar-guided air-to-air missile capable of

engaging enemy bombers at ranges in excess of 100 nautical miles (180 kilometers). To support the Phoenix missile, the F-14 is equipped with the AWG-9 radar, which at the time of its introduction in 1970 was one of the most powerful airborne radars ever placed on a fighter aircraft, and the primary reason for a second crew member serving as a dedicated radar operator. The F-14 also carries AIM-7 Sparrow (semi-active radar guided) and AIM-9 Sidewinder (heatseeking) missiles for medium and short range engagements, respectively.

Since the introduction of the AN/AAQ-14 LANTIRN targeting pod, the F-14 Tomcat is also capable of laser-guided precision bombing, making the F-14 a true multirole strike fighter. The DCS F-14 module has full support for the LANTIRN pod. Thus, a secondary mission for VF-1 is ground attack.

**Procedures.** VF-1 officially meets as a squadron for training three times week on the VF-1 or CVW-1 private DCS servers. Squadron members often fly together on the server outside the official training nights as well. Furthermore, when deployed in a theater of operations, the squadron may also be engaged in missions beyond or in lieu of the training flights (often Saturdays). Training flights are typically two to three hours in duration, and start with a briefing where the section (2 aircraft, a lead and a wingman) and division (4 aircraft; 2 sections) leads decide on the evening's mission, discuss tactics and areas to train, and determine the flight roster. This is followed by the flight itself, which usually starts and ends on the squadron's carrier (the *Roosevelt*) with a CASE I departure and arrival (clear flight conditions), unless the squadron is deployed for shore operations on an airfield. Because of VF-1's status as a carrier-capable Navy squadron, the squadron engages in monthly carrier qualifications (CQs) as well as regular CASE III (no visibility or night) launch and recovery operations. After recovery, each training is followed by a debriefing session, where one member will share their screen of the Tacview tool loaded

with the .acmi telemetry trace file for the session.

## Chapter 4: Formative Study

We conducted several interviews with dedicated DCS players to better understand the domain and their visualization and analytics needs. These interviews were conducted both in groups or as 1-on-1 sessions, online as well as in-person, and included observing a PvE session with briefing, game session and debriefing. Below we describe our study in detail. Note that this study as well as the summative study (Section 6) were approved by our institution's IRB.

### 4.1 Participants

We engaged a total of seven individuals (P1-P7) in interviews: five 1-on-1 sessions (one in-person, the rest remote), and two in a group session on Discord voice. All participants were screened to be 18 years or older, native or fluent English speakers, and experienced DCS players. P1 was a professional combat pilot. P2, P3, P4, P5, and P7 were part of VF-1. P5 was ranked in the top percentile in another popular multiplayer strategy game (*Dota 2*). P6 was a flight test engineer who collaborated with pilots and had an understanding of flight simulators. All participants were male.

## 4.2 Method

Sessions started with participants providing informed consent, during which time they were given the chance to ask questions about the study. We followed a semi-structured interview protocol; the list of questions can be found in our supplemental material. The format was relaxed and questions were asked in no particular order, but rather as dictated by the flow of conversation. Sessions lasted between 30 minutes up to an hour (the group discussion). The experimenters collected notes and observations, but none of the sessions were recorded (thus the lack of direct quotes).

## 4.3 Results

Here we summarize the main themes from our conversations.

- **Flight simulators tend to be a life-long interest.** All participants expressed an affinity towards flight simulators that go back many years, often to each participant's very first video gaming experiences. While several participants said that they played other video games, they all indicated that DCS is a game they keep coming back. For example, P3 had been part of DCS since the Black Shark module release in 2008; P4 since at least 5 to 6 years, but had been active flying flight simulators since their early childhood.
- **Virtual squadron authenticity.** All participants expressed in various ways how authenticity and military simulation is a big part of what attracts them to DCS and their virtual squadron. The attraction appeared not restricted to merely employing weapons against enemies, but to all aspects of modern air combat, including ad-

ministrative flight duties such as communicating with ground control, ATC, carrier operations, etc. Participants expressed the feeling that many DCS players are enthusiasts who wanted to but never had the opportunity to fly in a military air force. Several of the current VF-1 members are civilian pilots. Some have a military past, although only P1 was a current combat pilot.

- **On the importance of the debrief.** All participants emphasized the importance of a debriefing following a mission, either the same day or the following one. They felt that debriefings are what puts organized DCS players apart from other online games. Another reason for the debrief is that it is a routine occurrence in real fighter squadrons, and thus should be simulated also in the virtual squadron.
- **Egos and lack thereof.** Both P3 and P4 remarked on the lack of egos and posturing during debriefing session, which is unlike their experiences in other multiplayer games (especially first-person shooters such as *Fortnite* or *Call of Duty*), which they felt were rife with toxicity. The participants speculated that this may be because DCS virtual squadrons such as VF-1 aspire to emulate real combat pilots, who are said to be equally non-defensive during debriefs. P3 expressed the sentiment that a pilot should be the first to offer self-criticism on their own mistakes; failure to do so is a red flag. However, sometimes errors are not obvious, which leads to the next point...
- **Data is king.** All participants expressed the sentiment that constructive criticism and reflection is not possible without telemetry data serving as evidence. Most players as well as the server itself runs Tacview, thereby capturing the real-time telemetry of each flight, and one player would routinely open up the tool to stream their screen during debriefings. During the debrief, the different aircrews will take turns going

through the recorded data to discuss what went right, what went wrong, and offer constructive criticism to themselves and others. In the real world, all combat aircraft are equipped with video recorders that capture different aspects of the flight.

- **Mind the voice.** Voice communications—both the intercom between the pilot and RIO within the aircrew and the inter-flight VHF/UHF and administrative VHF radios in the Tomcat—are often of critical importance. During intercept operations, for example, the RIO will issue directive commentary to the pilot to control the fighter’s heading, speed, and altitude. Similarly, during tactical formation flight—which is employed during air-to-air engagements—the flight lead will issue standardized maneuvering commands. Administrative flight operations require checking in and checking out of different channels, communicating with ATC, ground, and tower control, and managing air intercept, strike lead, and tactical communications. For all of these situations, the participants expressed the need to capture radio communications in addition to telemetry.
- **Tacview woes.** While participants generally agreed that Tacview is a powerful piece of software that fills an important role in the DCS scene, they also had feedback on some of its shortcomings and gaps. Radio communication, for example, is not routinely captured, likely because Tacview is not integrated with SRS. P3 did note that Tacview can link telemetry to radio communications, and related an anecdote where this was used to great effect in a sequence of training missions. Furthermore, both P3 and P4 remarked on Tacview labels sometimes being so large and full of data that they obscure the ongoing action. Another concern is that Tacview data files can be very long, and it can be challenging to navigate through a sequence to find a specific event or repetition of interest. This is particularly challenging for

large-scale missions, where hundreds of moving 3D entities may be involved over a duration of several hours. It also means that viewing everything that happens in such a large-scale mission is cumbersome and time-consuming. Participants also had difficulty in recalling and then pinpointing game events on the Tacview timeline. P3 also mentioned feeling friction in the entire process of collaborating during the debrief which involved uploading the replay file, waiting for everyone to download it, and then screen sharing using another software such as Discord.

#### 4.4 Implications

We take several teachings from our semi-structured interviews. The primary implication is that the DCS community is highly interested in data-driven telemetry analysis and has the motivation and time to engage in it—or at least virtual squadrons do. At the same time, while the Tacview tool serves this need in many ways, it has several shortcomings:

- I1 - Tacview is a native Windows application, which makes sharing telemetry and specific highlights cumbersome.
- I2 - There is no easy and scalable approach to sharing clips from specific missions to illustrate important concepts.
- I3 - While the tool supports several data displays, such as aspect ratio, distance, speed, altitude, and G-forces, these displays sometimes obscure the action, can be cumbersome to display, and do not aggregate data over time.
- I4 - The Tacview tool is designed for viewing individual sequences, but does not fare well when displaying data from large-scale missions spanning many units, a long time period, and a large volume of space.

## Chapter 5: TacReel: 3D Telemetry Visualizer

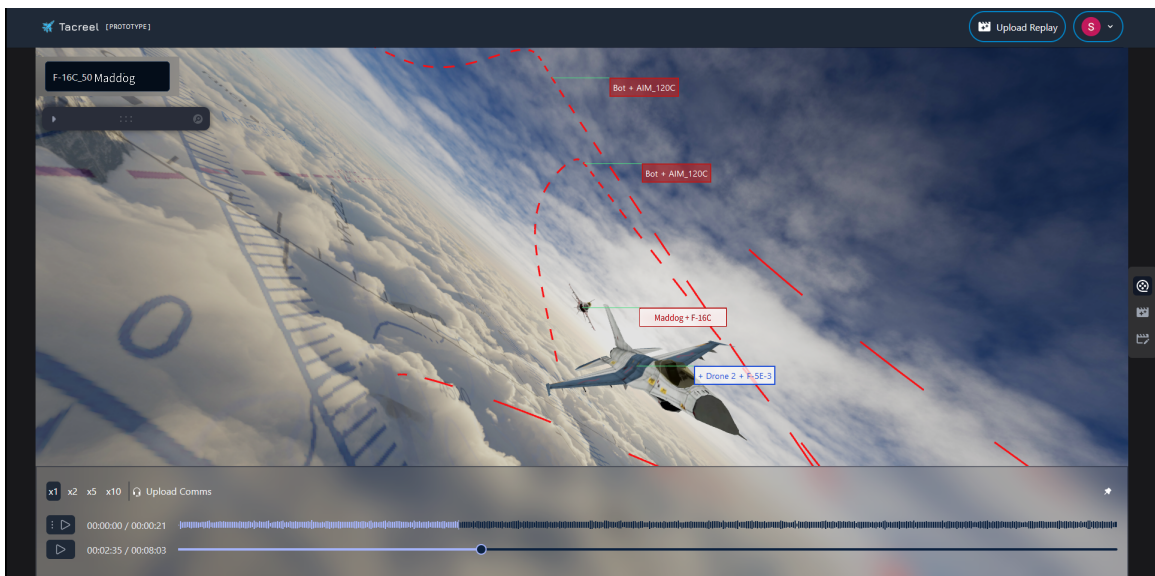


Figure 5.1: TacReel player.

To address the shortcomings of existing tools for 3D telemetry analysis, we propose TACREEL, a web-based 3D visualization tool that allows virtual pilots to explore, analyze, and share combat flight simulator telemetry data. The TacReel player is a 3D telemetry visualization tool that enables the user to navigate freely in both time and space for telemetry playback. Users can control the unit to focus on 5.1) (dropdown menu in the upper left). Labels (whose size, position, and visibility can be controlled) show the units being tracked. Trails show unit movement in both time and space. Based on user requirements, the tool also has support for uploading and synchronizing audio, such as radio communi-

cations stored via SRS. The interface is very similar to a regular web-based video player, such as Youtube TacReel consists of four components: (1) a *telemetry persistence* backend that enables uploading and storing ACMI and other telemetry files; (2) a *telemetry player* that enables the playback and analysis of 3D telemetry data; (3) a *reel director* for editing and constructing highlight reels that summarize long missions; and (4) an *event timeline* to automatically detect important simulator-specific events in a telemetry reel.

## 5.1 Use Cases

Drawing on our formative evaluation, we derive the following uses cases for the new tool:

- **Players** want to recall specific events during a flight, share them with the squadron, and compare the execution to their actual plans;
- **Scenario/Level creators** are interested in knowing did the scenario play out as intended and which parts could be more entertaining;
- **Streamers/Broadcasters** need to tell a story of the game play session through interesting events, possibly through different perspectives and cameras; and
- **Spectators** watch for entertainment and seek interesting moments; expert spectators want to explore, go into details and potentially learn through observation.

Based on these use cases, we designed TacReel to help users (1) recall and locate specific events by providing rich context, (2) analyze telemetry data in both time and space, (3) identify important events based on simulator-specific or player-defined rules, and (4) construct reels using storytelling techniques such as views, clips, and camera angles.

Below we discuss each of these components in turn.

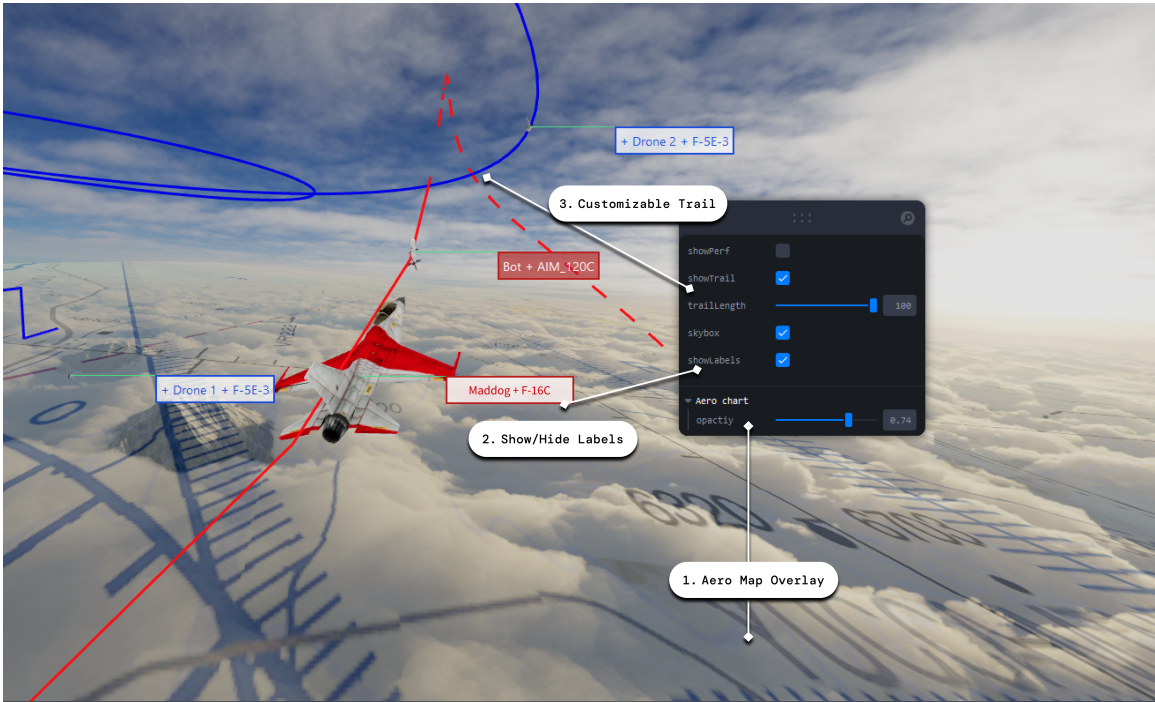


Figure 5.2: Mission replay using TacReel.

## 5.2 Constraints

The design goal for TacReel was to keep interactions fluid. This meant system should support and feel natural along every step of the visualization pipeline [12] shown in figure 5.3 .

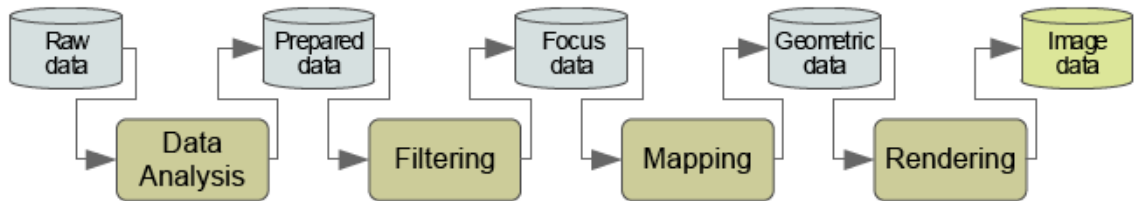


Figure 5.3: Data visualization pipeline. [12]

To achieve this goal TacReel had to be built around two important constraints -

- **Responsiveness:** According to Zhicheng Liu Et al. Interactive visualizations that

have delay of 500ms incurs significant costs, decreasing user activity and exploration [30]. This further leads to reduced observations, generalizations, and hypotheses. Therefore all the interactions in TacReel had to have less than 500ms response time.

- **Frame rate:** Frame rate has impact on both user's performance and enjoyment while resolution of the system has little impact on performance and some impact on enjoyment [9]. Therefore wherever necessary TacReel prioritizes frame rate of the 3D telemetry player over its resolution and graphical fidelity. TacReel aims for at least 30 frames per second and 60+ frames per second in most use cases.
- **Occlusion:** Occlusion is a common usability problem in 3D visualizations. The occlusion effect in 3D environments negatively impacts tasks involving discovery, access, and spatial relation of objects in a 3D visualization. [16] To mitigate occlusion, TacReel's representation had to allow users to zoom and rotate freely. Additionally, the system should allow filtering to remove clutter. [31]

### 5.3 Data Model: 3D Telemetry Persistence

Once a player has created an account in the TacReel tool, they can upload telemetry data as ACMI files into the system. The ACMI file is converted into an internal time-stamped data stream format consisting of entities with their position and rotation over time. Positions are specified using longitude and latitude offsets from a reference starting point. Times are specified down to milliseconds from a reference starting time.

Because different telemetry exporters may use different sampling rates, data is resampled to a common refresh rate to save memory and disk usage. The resulting files are then indexed into a database for quick access and stored as flat files. The persistence layer en-

ables accessing data values, such as X, Y, Z or rotation variables, for any unit at any point in time; the persistence layer will automatically perform linear or spherical interpolation between sampling values.

The TacReel player supports advanced spatial reasoning using trails, labels, and spatial metrics (such as distance, aspect, bearing, etc). The toolbar dialog enables full control of these parameters. 5.2)

## 5.4 3D Telemetry Player

The TacReel 3D telemetry player (Figure 5.1 and 5.2) is akin to the Youtube player, but instead of showing 2D image sequences (i.e., videos), it shows 3D telemetry data captured over time. This means that in addition to moving the action in time, the player can freely move the camera in 3D space. Users can orbit the camera around a specific unit of interest in the telemetry data using mouse dragging or tapping. The tool exposes several features:

- **Switch camera focus:** Navigate in 3D space by changing the camera focus to a new intended target of interest.
- **Trail length:** Control the time over which object's positional trail is displayed.
- **Overlay map:** Flight maps can be overlaid on the existing world to increase spatial cues, such as no-fly (exclusion) zones, ATC maps, territorial boundaries, etc.
- **Audio communications:** Players can upload their audio communication recorded during the game (often from SRS), sync it to the replay data, and play it together with the telemetry replay.
- **Pin, enable, disable object labels:** Visual cluttering due to labels was a common

pain point from our formative study. TacReel allows users to pin individual labels or disable them when required.

- **Target Pairing:** In the formative study we observed that users paired their plane with the target plane such that the camera always contains both the player and target. It also showed the details such as distance, velocity, altitude difference between the two. TacReel includes preliminary form of this feature as seen in the figure 5.4

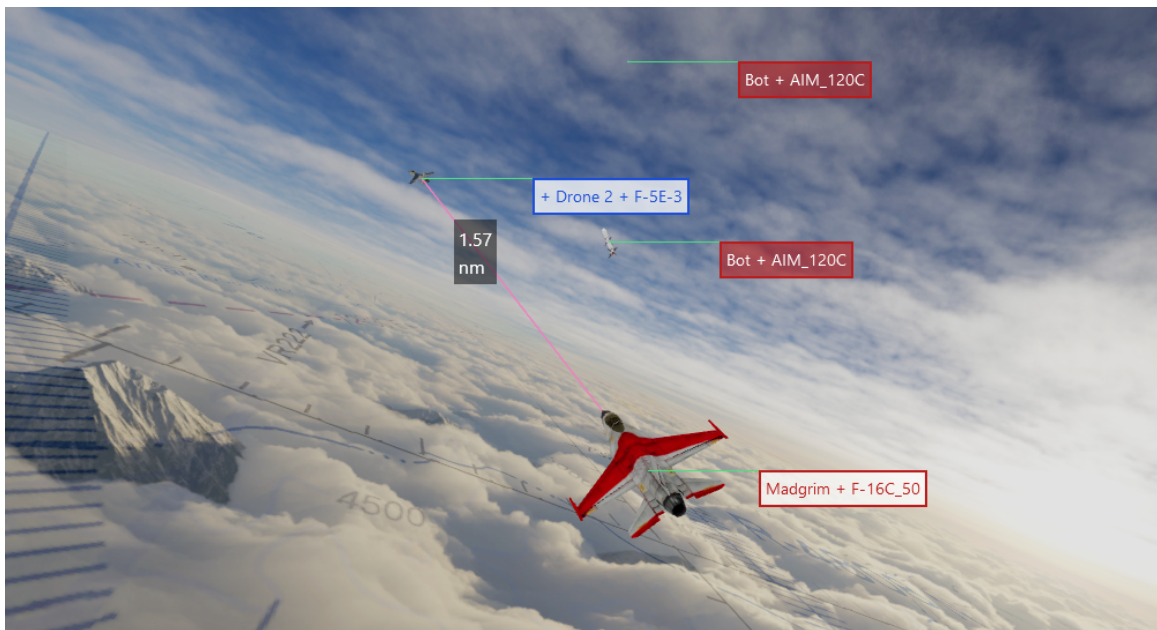


Figure 5.4: **Target Pairing.** Pink line shows the distance between player and target in nautical miles.

## 5.5 Event Timeline

The event timeline displays an event chart with game objects on the Y-axis and timeline on the X-axis. Events are detected by using an event rule engine based on a pre-determined set of game-specific rules. Events can be filtered based on types and object parameters such

as object type, pilot name, collation, etc. User can then use the timeline view in the director mode to select and determine order of the events in the final reel.

We define an *event generation rule*  $R_e(t)$  such that, when applied to a full time period  $t \in T$  will generate a set of events  $E$  for a time period  $T$  such that these events exceed an interest threshold. Each rule can use the telemetry data for all units, usually in a window around the time  $t$ , to determine the level of interest for that time. Below we list a set of event generation rules that are applicable to VF-1 (this list is extensible and non-exhaustive):

- **Takeoff/Launch:** A friendly (player) unit is taking off from a shore airfield or launching from a carrier (CASE II or III departures will have higher interest ratings than a routine CASE I departure).
- **Landing/Recovery:** A friendly (player) unit is landing on a shore airfield or recovering on a carrier (CASE II or III arrivals will have higher interest ratings than a routine CASE I arrival).
- **Merged:** Two opposing aircraft approach within 1 mile, at which point their radar contacts merge (thus this is known as “the merge” by fighter pilots). This usually results in a dogfight, which is of high interest to viewers, and is normally resolved by short-range heatseeking missiles or guns.
- **Strike Comms:** Tactical communications on strike, intercept, or bullseye/BRAA<sup>1</sup> calls are often important.
- **Danger Close:** Situations when air assets are deploying ground attack weapons close to friendly ground assets. This situation is often particularly interesting because of

---

<sup>1</sup>Bearing, range, altitude, and aspect; the four pieces of intercept information that AWACS/GCI issues to committed air assets during an active intercept or BVR—Beyond Visual Range—timeline.

the high risks associated with such action.

- **Radar Contact:** During intercept and BVR (Beyond Visual Range) operations, acquiring an enemy air asset on fighter-borne radar is a significant event that precedes weapon deployment.
- **Weapons Launch:** Firing any form of weapon will be of high interest to viewers.<sup>2</sup> This event is separate from weapon hit because modern stand-off air-launched weaponry often are often launched from 50 miles or more.
- **Weapon Hit:** Whenever a missile, bomb, or bullet hits a target, an event of high interest is generated.
- **Cranking:** When deploying air-to-air missiles, VF-1 standard procedure is to “crank”; to turn away from the target and drop altitude in an effort to reduce closure and prepare for defensive maneuvers while still allowing the fighter’s radar to track the enemy and guide missiles (modern fighter radars are forward facing).



Figure 5.5: Spatial event generation.

<sup>2</sup>The NATO brevity radio call for firing an active radar missile is Fox 3. Fox 2 signifies a heatseeking (often short range) missile, whereas Fox 1 is a semi-active radar missile guided by the fighter itself.

Beyond semantic event rules, we also provide user-controlled spatial event rules using *spatial brushes*. Rather than generating events based on unit-specific actions, these events are generated whenever a unit enters a spatial brush placed by the user. Such brushes can be placed, for example, around an enemy airfield serving as a strike target. Events will then be automatically generated whenever an aircraft enters the brush, allowing the user to create a reel showing a concerted combined arms attack on the airfield. Figure 5.5 shows an example of a spatial brush and its controls. Spatial brush (green box; other shapes are possible) used to generate events around a specific geographic location on the map. TacReel allows users to generate and place any number of brushes, as well as control which units it affects (e.g., only blue, only ground-based, or only airborne units). Affected units entering the brush volume generate an event.

## 5.6 Reel Director

The TacReel director tool allows users to author sequences—*reels*—consisting of multiple *clips*: snippets of telemetry data limited in time and space (Figure 5.7). This enables the user to author condensed, summary, or highlight reels that decrease the full length of a telemetry stream consisting of hours of data and hundreds of units into a smaller and more manageable sequence that can be viewed in the TacReel player. Even if the director allows the user to control the camera angle and distance, the telemetry player retains the possibility for the user to navigate in 3D space as well.

The director tool integrates the event timeline visualization (Figure 5.6; see the following subsection for a detailed description of this advanced functionality). The user can either select pre-defined events for inclusion in the clip, or define new clips by selecting directly one or more units and a time period on the timeline. They can also arrange the order of



Figure 5.6: **Director tool with event timeline.** The timeline is organized with units on the vertical (Y axis) and time on the horizontal (X axis). Colored boxes represent events detected by the rule engine. Users can create custom events by brushing the timeline.

the clips as well as the camera angle and distance used in each. Director view provides a flexible way of supporting non-linear creation process and allowing users to iteratively go back to different phases of the process which are important consideration authoring data videos [3].

After applying above mentioned operations on the canvas, the main toolbar has three timelines stacked and aligned horizontally (Figure 5.7). From top down they are, Highlight events timeline, audio communications timeline & highlights player timeline. The highlight events timeline shows labelled rectangles for each event selected from the director view.

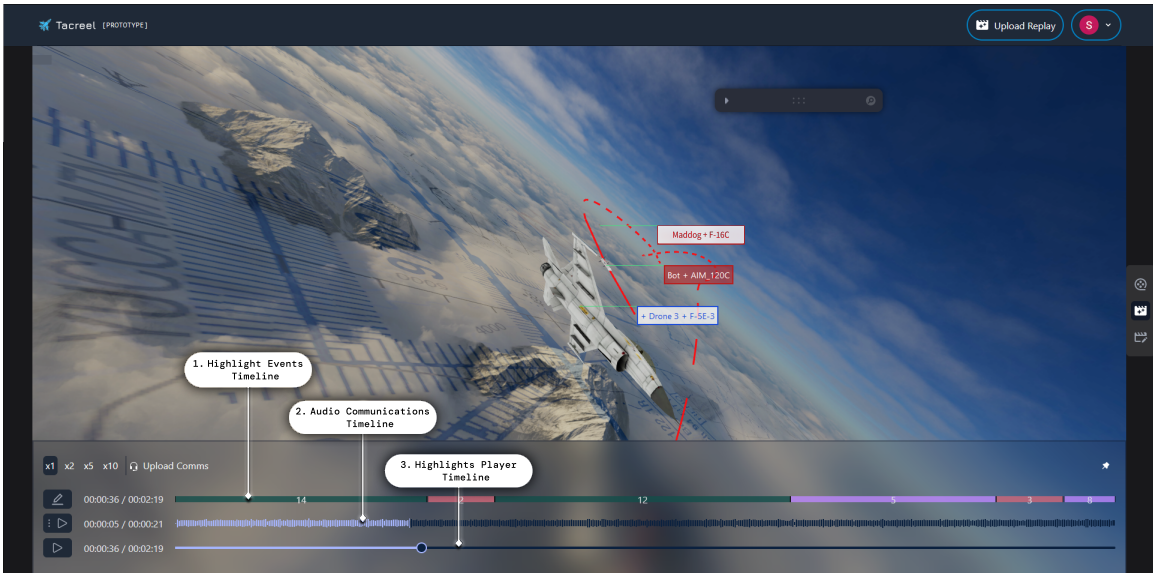


Figure 5.7: **Highlight reel.** TacReel player showing a highlight reel of multiple clips organized into a sequence.

## 5.7 System Architecture

Figure 5.8 shows an overview of the TacReel system architecture. TacReel uses React for the frontend and Node.js for the backend.

### 5.7.1 Backend

When users upload their ACMI replay file it is processed and modified before it can be streamed back to the user in TacReel's telemetry player. The processing involves following steps -

- **Step 1. Raw data to JSON:** The ACMI file is converted to a parseable JSON object using the `jsacmi` library. This virtual database object holds the data for every object in the replay for each 250th millisecond of the replay time.
- **Step 2. Rule Engine:** The replay database is processed under various rule sets to

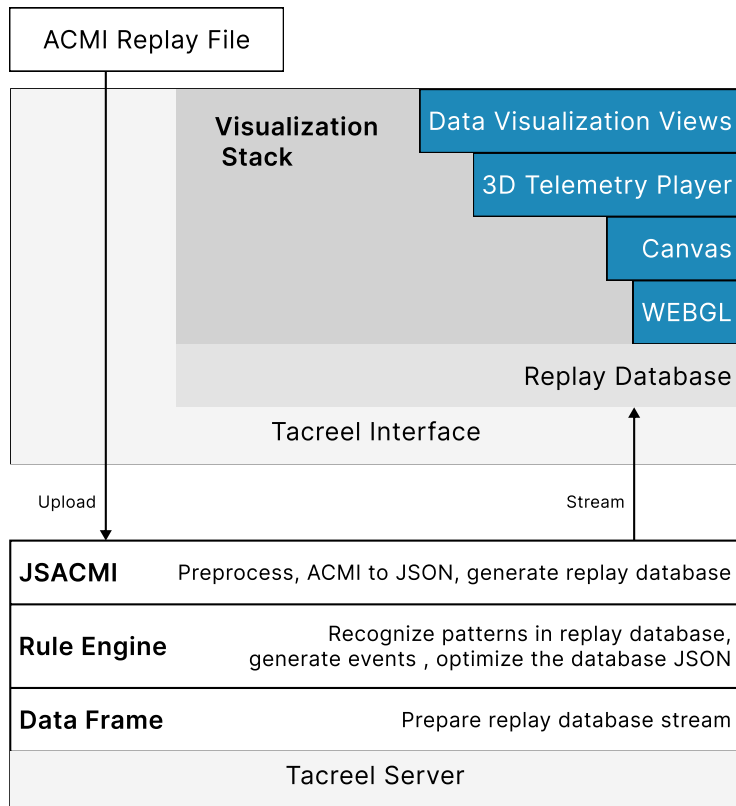


Figure 5.8: **TacReel system architecture.**

auto-detect events such as player death, missile launch etc. In some ways it works similar to StratMapper [2] for event detection and generation. The object database is checked for static objects and given a static positional value over the entirety of the replay in order to reduce the size significantly.

- **Step 3. Data Frame:** Lastly, the object database is made *streamable* by fitting to a data frame that is compatible with TacReel’s telemetry player. This is an important step which allows TacReel to potentially handle telemetry data beyond DCS replay files by converting the respective telemetry data to fit TacReel’s data frame.

## 5.7.2 Frontend

The front-end uses the Three.js library and several free open source 3D assets for recreating the simulation data that is controllable through our custom built 3D player. The player and event timeline use D3.js for data visualization functionality. Internally, the 3D player and the event system is designed to integrate with Theater.js<sup>3</sup>, a motion library to help create cinematic scenes in Three.js.

## 5.8 Performance

TacReel player's performance depends on the user's browser, RAM and CPU usage. Across multiple hardwares and browsers we observed that the Google Chrome browser had the best performance in terms of frame per second (FPS). TacReel averaged on stable 68 FPS across all types of replays. Using WebGL's instancing [20] TacReel also manages to maintain less than 100 draw calls resulting in lower CPU usage. The above performance was achieved through various optimization techniques and algorithms that are beyond the scope of this thesis. Most notable out of them was optimization to how the telemetry data gets updated every frame. Initially we were under the assumption that longer replay times will result in longer update times per fetch per frame, however when we graphed the performance as shown in 5.9 we found out the update times correlate to the number of objects inside replay and not the replay length. This required us to rewrite significant part of how data is handled by our telemetry player resulting in significant improvements to the performance.

---

<sup>3</sup><https://www.theatrejs.com/>

Performance of telemetry update functions w.r.t replay size and no. of game objects

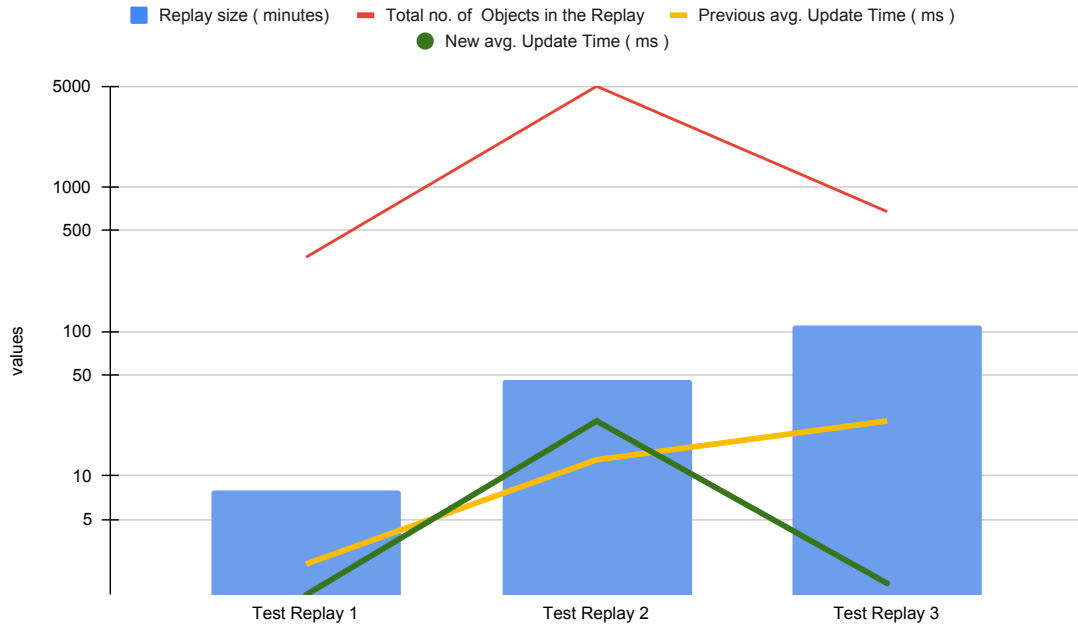


Figure 5.9: Update function optimization

### 5.8.1 Browser Compatibility

Since TacReel is built using web-based framework ( React and Node.js ) and further more the client reel player is implemented using the Three.js on top of HTML Canvas and WebGL, it can run on any browser including desktop, mobile and VR browsers Figure 5.10. This gives opportunity for users to experiment with unique personalized workflows.



(a) Playback mode.



(b) Playback mode synced to gameplay video.

Figure 5.10: TacReel inside Apple Vision Pro's Safari browser.

## Chapter 6: Summative Study

We used the needs, requirements, and current gaps in practice from our formative evaluation (Section 4) to design the TacReel tool (Section 5). Then we conducted a summative evaluation of TacReel by deploying the tool with VF-1 and collecting their feedback over a period of two weeks. The purpose was to understand whether the new tool addresses the gaps we identified in current practice, and how the tool may change their practice going forward. Finally, we used the evaluation to derive generalizable findings on post-flight telemetry visualization for combat flight simulation.

### 6.1 Participants

We recruited five participants for this study; P2 through P5 as well as P7 from the original formative study. All participants were part of the VF-1 virtual squadron we were embedded with.

### 6.2 Method

We deployed the TacReel tool with VF-1 over a period of two weeks. During this time, we participated in two separate squadron missions as embedded observers. We also discussed the tool and its functionality with participants between the missions on several occasions. All data collection was performed on the VF-1 Discord; this included both



(a) Aerial refueling for VF-1 Tomcats during Caucasus mission. (b) Carrier qualifications (CQ) on *U.S.S. Theodore Roosevelt*.

Figure 6.1: **Summative evaluation missions.** Our team was embedded during two missions: a fighter sweep in the Caucasus map (left), and a carrier qualification event on the *U.S.S. Theodore Roosevelt*.

text chat, voice conversations, and screen sharing (“streaming”). Screen-shared content included Google Slide briefing slides, in-game DCS streaming from the F-14 cockpit (pilot and RIO seats), and both Tacview and TacReel for debriefings. Note that in VF-1, while players choose their own callsigns, their real names are also painted on their jets. They do not, however, use webcam video during voice communication even if Discord supports this feature.

Participating in a mission involved listening in on the briefing, watching the stream from one or several of the participants during the mission, and then participating in the debriefing after the flight. During the debriefing, we uploaded the ACMI track file recorded by participants into the TacReel tool and shared the URL to the tactical reel with the participants. We then asked them to use the tool to debrief the mission. Participants were also encouraged to use Tacview for the same purpose to allow for comparing and contrasting the tools. During the debriefing sessions, participants were asked questions about their decision making and to think-out-loud while operating the analysis tool.

We selected two missions for our deployment that were very different in scope: a “real” VF-1 campaign mission, and a routine training night. The rationale was to better understand

whether our novel TacReel was better suited for a real or training mission. The two live missions that we participated in were the following (Figure 6.1):

- **Low-level in the Caucasus:** This mission was put together by one of the sister Navy F/A-18 strike fighter squadrons and involved the VF-1 Tomcats preparing the way for a ground strike package in a “fighter sweep”: an offensive mission designed to seek out and destroy enemy air assets in a designated area. The mission took place in the Caucasus map, originating from the squadron’s carrier and entering hostile territory where the squadron encountered enemy MiG-23s and MiG-29s attempting to intercept the ground attack mission. A total of three virtual squadrons participated: two F/A-18 squadrons and VF-1, the sole F-14 squadron. The mission was a success, with no blue force losses and the ground targets destroyed. The mission lasted for a total of 3 hours: 30 minutes briefing, 90 minutes of game session, 1 hour of debriefing.
- **Tuesday training in the Persian Gulf:** The Tuesday flight night is designed for basic training, the topic depending on the week in the month: (1) basic aircraft handling and shore operations, (2) carrier operations, (3) within visual range (WVR) combat, including basic flight maneuvers (BFM), air combat maneuvers (ACM), and short-range missile deployment, and (4) beyond-visual-range (BVR) combat, including tactical formation, intercept geometry, and BVR timelines. The Tuesday we attended was the second in the month, which meant that the focus on the training was on CV operations, including carrier communication, CASE I, II, and III launch and recovery, and emergency procedures. The training night lasted for a total of 2 hours and 45 minutes: 15 minutes of briefing, 2 hours of game session, and 15 minutes of debriefing.

Data collection during the sessions included screenshots and live notes. At the end of the session, we collected the ACMI trace files with the permission of the participants. We did not record audio or video during the sessions.

### 6.3 Results

Here we report on the major findings from both mission deployments of TacReel and its comparison to Tacview.

**Interface.** Participants felt that Tacview’s interface “gets the job done” and praised its information displays and rich features. However, several participants noted that there was a steep learning curve. P6 felt that Tacview provides an impressive amount of details, but the interface is not intuitive. In contrast, all participants found the TacReel interface visually pleasing. Some of them also liked the more realistic 3D models rather than just having plain unit markers.

**Timeline navigation.** During debriefing participants sometimes had difficulty using Tacview to quickly identify and locate the events they wanted to talk about. This was much easier with the TacReel tool.

**Communication.** Even with 40+ years of combined experience with flight simulators, mission participants sometimes had confusion about communications during the game sessions. AIC/AWACS (air-intercept control) calls were particularly confusing: examples included bullseye calls (“*bullseye 340 for 26*” means bearing 340 and 26 nautical miles from an agreed-upon common marker, the so-called “bullseye” point), reciprocal and magnetic bearings (the reciprocal bearing is the magnetic bearing plus or minus 180°), picture calls

(an overall description of the air “picture”; all radar contacts seen by AWACS), commit calls (fighter-initiated calls for attacking a target), etc. This was brought up in debriefing where participants claimed they heard and interpreted something differently than the communicator had intended. All participants agreed that having an audio overlay of communication from SRS on top of the 3D telemetry replay would be very beneficial to resolve such confusing conflicts and for continued training purposes. This is addressed in TacReel.

**Summary reels.** All participants appreciated the summary reels, especially the control they had on the events. Some mentioned that it would take them time to get used to the new workflow since they are used to verbalizing events during debriefing rather than visualize them. Furthermore, participants noted that the director mode and event timeline would require customized filters to correctly identify the events the squadron tends to be interested in.

P1 noted that it would be useful to have event rules that show deviation from realistic flight. For example, P1 noted that freely moving your head in a real jet under 6Gs and above is extremely difficult, which is not the case in DCS. He also noted that the human body cannot hold above 9Gs for more than few seconds, but this is not true in the simulation, where some people routinely do it for long periods. P1 wanted to find such events and judge the gameplay based on it.

Finally, a summary reel can also be used as a quick “recap” to share with others, either outward-facing to people outside the squadron, or internally to squadron members who had missed a particular training night or mission. These highlight reels would be a good way to generate interest and capture takeaways or major events.

**Addressing the Design Implications.** Here we review how each of our identified design implications were addressed by TacReel:

- I1 **TacReel is a platform-agnostic web application.** It addresses *I1* by allowing users to incorporate TacReel within their personal workflow as long as they have access to a web browser. Users can upload the file to TacReel and allow entire squadron to have access to it.
- I2 **Director view and Summary view facilitate creation, viewing and sharing of clips.** Users can effectively observe distinct, specific events at a glance and then choose to generate a highlight reel that helps to illustrate important concepts.
- I3 **Finer control over display elements.** On screen elements such as but not limited to object trails, maps, sky boxes, data displays, distance can be altered visually by controlling their opacity, length, expansion, level of detail and more to reduce on-screen clutter and reveal the action to an extent (Section 7.3).
- I4 **Temporal and spatial constraints on querying.** On timeline users can select a time period of interest or create their own events in director view with the brush component. Further, spatial brush filters for events that occurred within the specific volume of the cube.

**Anecdotes: Low Level in the Caucasus Mountains.** As every other DCS mission, we observed this session as well though three phases: briefing and planning, mission execution and finally debriefing.

Briefing and planning phase started out with all the participants and a scenario planner joining the squadron's designated Discord channel. The scenario planner explained the

modification he had made since the last run—implying that the squadron had done this mission multiple times before. Therefore it was brought to our notice after the session by one of the participants that this briefing was shorter and more concise than what they usually do. The squadron members dispersed into different groups preplanned in the briefing by joining respective Discord channels in the squadron server.

As the mission execution commenced, pilots joined the game server and the SRS-DCS channel. As observers we were provided with the respective communication channels to listen in and the participant that we were observing shared their screen with us on Discord. We observed that pilots frequently switched communication channels to listen-in or broadcast information. They used SRS exclusively while still staying in Discord voice chat to help fix technical problems with the entire game server and game communications setup. The Discord voice was still used by the relatively inexperienced pilots to ask in-situ questions about navigation and planning. This ensured that the immersion of the other pilots communicating on SRS did not break as well as helped pilots distinguish communication that was about the in-game mission from the talk that was about the overall game knowledge. TacReel allows users to upload the recorded .mp3 files and stack such voice communications. The mission lasted for about 50 minutes before everyone landed safely back on their rendezvous runways.

Lastly, to start the debriefing one of the players uploaded the mission replay file to Discord. The mission planner took the lead by downloading and opening the replay in Tacview and then sharing their screen to the entire squadron. At the same time we asked our participants through private message to open the replay file in TacReel. Our participants quickly scrolled on the timeline to check if the replay file was working correctly. They then navigated to the director view with timeline events visualization, where they filtered out

the non-playable game objects like anti-air infantry, SAM sites, and other grounded planes. They then selected their own plane, their wingman's planes and then rest of the squadron's planes. When asked if they found any deeper insights into the game play, the participants said it feels convenient to observe the mission in this format however since they have already practiced this mission many times before they already knew all the insights to this particular mission. At the end of the debriefing we reached out to the scenario planner for their thoughts on TacReels Director View. The scenario planner mentioned that they would definitely use such feature to quickly compile a reel to *show and tell* while allowing the rest of the participants to do the same.

**General feedback.** P6 and P3 stated that storing, sorting, and sharing of replays with the squadron was cumbersome using Tacview. Many of the participants also appreciated the ease of opening a browser-based tool that was accessible to everyone immediately instead of a native application. Some also mentioned that they liked the ability to overlay custom flight maps on top of the 3D player world.

## Chapter 7: Discussion

Overall, we can summarize our high-level findings as follows:

- Semi-milsim virtual pilots are very interested in telemetry analysis for the purposes of improving their skills.
- Data visualization is a natural fit for 3D real-time telemetry, and a basic 3D replay tool can easily serve as a platform for more advanced visual data displays.
- Telemetry data can be useful both for discussing mistakes and demonstrating correct procedure, but also for summarizing a mission as a highlight reel, both for internal and external audiences.
- Online multiplayer games are (much) more than just the game; the communities span many different media, and third party tools are necessary to make up for gaps that the developers do not fill.

Below we explain our findings in detail, generalize them to wider contexts, and discuss our limitations. Overall, we think that our work in this thesis may be of value for both the specific context of virtual squadrons such as VF-1 within flight simulation community as well as the broader simulation and multiplayer games community.

## 7.1 Explaining the Results

We found the virtual squadron community to be different than a traditional online video games community. Learning to operate a simulator takes an enormous commitment of time along with monetary investment in form of computer peripherals and hardware. Furthermore, simulator players must spend time studying—both inside and outside the game—to operate their aircraft. This means overall that flight simulator pilots in general, and DCS players in particular, tend to have a high tolerance for reading, analyzing data, and spending time outside the game. In particular, it explains the dedication to detail exhibited by our players and the amount of time they were willing to invest in our work.

Embedding inside VF-1 gave us a glimpse of how DCS virtual squadrons operate and how they—to some degree—mimic real-world military units. Overall, the squadron is a tight-knit group and several pilots have played together for many years. Membership is tightly controlled and joining requires both knowledge and pilot skill entry test, recruiting is limited, and there is also a limited turnover of pilots. While the squadron has a perennial challenge in attracting RIOs (which is an arguably less glorious position in the jet, even if the RIO often assumes command of the aircraft during BVR engagements), VF-1 have more applicants than they can accept.

We also observed an internal friendly rivalry within the squadron as participants often mentioned other players who were good at specific aspects of the game: “*Maverick is really good with angles and energy management during dogfights*” and “*Iceman’s Phoenixes always hit but we can never get our own to hit anything.*” The squadron maintained a special channel on their Discord server for After-Action Reports, which were written by flight leaders after a mission or engagement. In this way, the squadron records history and

perpetuates hard-won knowledge within the ranks.

Finally, VF-1 also has a dedicated “heritage room” channel on their Discord. Written like a fictional historical account of the squadron’s exploits with narrative text and illustrative images, the channel serves as annals for the squadron.

Fighter pilots, both in real life as well as on the internet, always want to improve. During debriefing, players would try to understand reasoning behind each other’s decision and context in which they were made. Analyzing others’ play and data for learning further increases the need for understanding context and decision making at each step. Overall, this is typical of serious or professional players: they often turn to observing other players’ gameplay to learn new skills, or compare it against their own to identify areas for improvement [41].

The deployment of TacReel prototype with VF-1 revealed its potential benefits but also its gaps. TacReel provides an improved context that is not just an overview of the game, but helps player recall events better through visual cues. Participants liked the control over visual storytelling through events. The director mode also aids in communication and sharing. P4, who is a non-native English speaker, said “*it may support my need to often show rather than to tell.*”

The semi-automatic labeling and creation of events saves time and efforts by reducing laborious task of going back and forth on the timeline to find the required moment. To take the automation even one step ahead, we asked a probing question that whether participants would prefer to have text input like ChatGPT for directing and generating a highlight reel. To which P1 & P7 mentioned that they don’t want to type during debrief after a long gaming session, but are fine with speaking on microphone if it can turn voice into text. They also noted that it depends on whether the input accepts flying language or movie-making terms

(dolly, boom, truck, pan, tilt, roll etc). Lastly, TacReel, by virtue of being browser-based, provides easier sharing and collaboration.

## 7.2 Generalizing the Results

In conducting this work, we opted for embedding with a single virtual squadron rather than broadly engage with the DCS or even entire flight simulator community. This means that our findings may be specific to flight simulators, to DCS, or even to the semi-milsim virtual squadron community in particular. Nevertheless, while all our participants indicated that DCS and VF-1 were their main gaming affinity, they were no strangers to other flight simulators as well as multiplayer online games. All participants mentioned that they had been exposed to simulator games (not limited to flight simulators) early in their lives. Further, all of them played at least one other simulation game other than DCS, such as *Microsoft Flight Simulator* or *X-Plane* as well as racing simulators such as *iRacing*, *Assetto Corsa*, or *Dirt Rally 2*. Several were also active within first-person shooters: both tactical first-person simulator shooters such as *Ready Or Not* (2021), *ArmA 3* (2013), and *Escape from Tarkov* (2017), as well as more traditional first-person shooters such as *Battlefield*, *Call of Duty*, *Counter Strike* and *Valorant*. In other words, while the VF-1 players surveyed in this thesis represent a specialized segment of the entire online gaming population, we do not think that they are too different from the rest.

For this reason, while our study was focused on VF-1, the challenges and feedback from this squadron might mirror those of other virtual squadrons or online communities that engage in cooperative virtual missions or games. For example, the reliance on tools (both TacReel and Tacview) and other supporting tools mentioned in the study (SRS and other) verify the importance of specialized software in enhancing the experience and learning of

virtual squadron members. Overall, that games are increasingly dependent on third-party tools that exist *outside* the game is consistent with prior findings [41]. Modern multiplayer games tend to have a much longer shelf-life than traditional single player games; for example, *Ultima Online* (1997) and *Dark Age of Camelot* (2001) are both well past their two-decade mark, but are still active. Sometimes games are abandoned by their developers but find new life in the player community; *Falcon BMS* is one example. Nevertheless, this means that as games branch out and communities grow, third-party tools are only going to become more important in the future.

### 7.3 Limitations

Our study involved five participants from the same virtual squadron (VF-1). This sample may not be representative of the broader combat flight simulation community. Furthermore, our perspective in this work was solely focused on the players, which leaves a gap in understanding how level creators, broadcasters, and spectators may perceive post-game telemetry visualization. We had planned a longer study where users and virtual squadrons can use TacReel over weeks and provide us detailed feedback. We received 35 waitlists and 3 big communities interested in trying out the tool within their workflow. However, the way TacReel's current backend is built it cannot reliably stream concurrently to multiple users and will need a rework for scaling. In future, we plan to deploy the tool to multiple squadrons and for a longer period of time.

TacReel's current version has two limitations that affect the visuals of the 3D telemetry player. For the elevation of the globe i.e height map, we use NASA Shuttle Radar Topography Mission's [17] height data. This height map in some cases does not match the DCS' exported map data, resulting in game objects visually appearing above or below sea levels

depending on the elevation of that area on the globe. Secondly, the replay player uses Euler angles instead of quaternions angles for yaw-pitch-roll of objects. This, in rare cases causes gimbal lock positions resulting in out-of-place rotations of the objects [19].

## 7.4 Future Work

Our comparison between TacReel and Tacview, while informative, may have been influenced by users' prior familiarity with Tacview. Their feedback on TacReel could have been shaped by existing habits and preferences. Furthermore, TacReel is in its prototype stage, and as such represents only a starting point for summarizing and storytelling spatio-temporal game data. Future versions could also integrate real-time collaboration, and enhanced communication features such as annotations, camera views, camera movements, and transitions between events for better storytelling. Extended reality has long been utilized in the games industry and is emergent for pilot training in the military and commercial airline sectors, as mentioned in an extensive literature review by Jamie Cross Et al. while virtual reality may not replace traditional flight simulators, the technology can supplement classroom training activities and some aspects of simulator procedure training with promising learning outcomes [11]. Tools like TacReel can certainly fill these needs. Furthermore the spatial information in games & simulators can be studied beyond mere pixel values, spatial data of this kind can be abstracted and used to analyze player behavior patterns and decisions [32].

In mixed reality workflows (Section 7.1), tools like TacReel can certainly fill in the gaps by supplementing classroom training activities and some aspects of simulator procedure training [11].

Identifying player strategies and decision patterns that give insights into player behavior

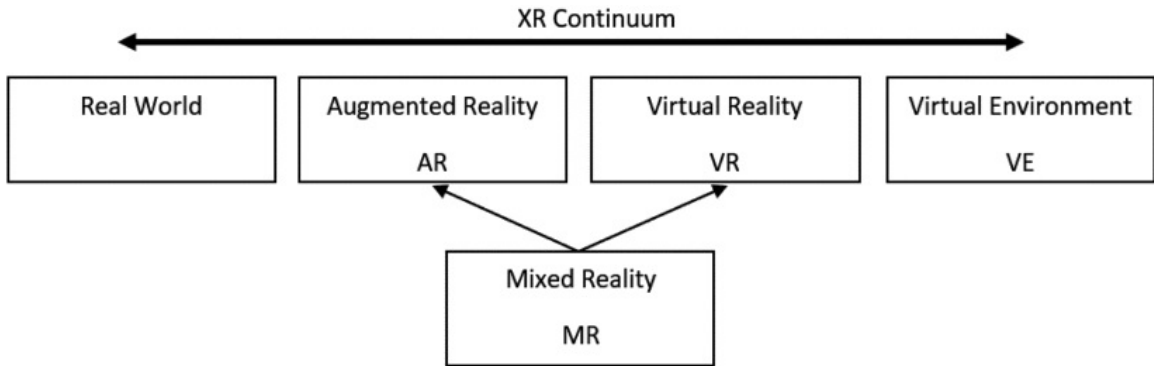


Figure 7.1: **XR Continuum.** [36]

is one of the most difficult tasks for games and sports analytics. The spatial information in games & simulators can be studied beyond their pixel values. Spatial data of the kind described in this thesis can be abstracted and used to analyze player behavior patterns and decisions [32]. Future work into tools like TacReel can play a part in visualization of such spatial analysis (Section 7.2).

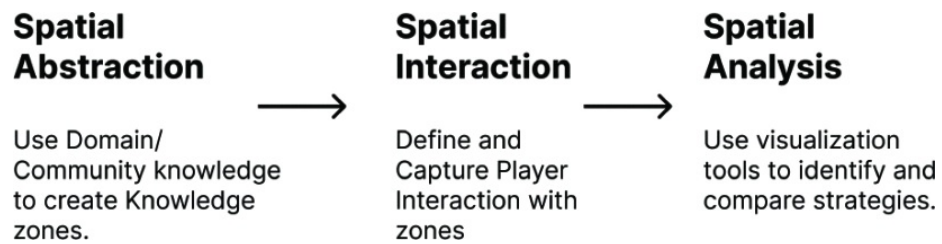


Figure 7.2: Steps followed in breaking the spatial arena of a game into semantic zones. [32]

The core features, especially the 3D player, is built for any telemetry data and could be extended to other simulation environments, such as naval or ground combat simulations or even arcade simulations.

## Chapter 8: Conclusion

We report on findings from an extensive qualitative study embedded in a virtual squadron for the combat flight simulator *DCS World*. During several interviews, group sessions, mission briefings, combat flights, and debriefings, we collected rich qualitative data about the telemetry analysis needs for these players. Based on these findings, we designed and developed a web-based telemetry visualization framework that addresses the current gaps and shortcomings the squadron faces. We then deployed this tool with the squadron and evaluated its use in a summative study. Our findings underline the flight simulator community's dedication to data-driven analysis and its willingness to devote time and resources to this pursuit. This area is ripe for deployment of advanced data analytics and visualization tools, and the player community is enthusiastic about adopting them.

## Appendix A: Formative interview questions

### Formative Interview Questions

Title: "Wichita 1-1, Fox Three" – The Role of 3D Telemetry Analysis in Combat Flight Simulation

#### Questions for Players:

##### 1. Background & Experience:

- How long have you been playing on flight simulators?
- How did you get into flight simulators?
- Which flight simulators do you commonly use?
- Are there any specific aircraft or scenarios you prefer?

##### 2. Usage Patterns:

- How often do you play or practice with the simulator?
- Do you follow real-world flight plans or scenarios?
- How do you prepare for a session? (e.g., debriefing, route planning, weather selection)

##### 3. Goals & Objectives:

- What are your main objectives when you use the simulator? (e.g., leisure, training, exploring new aircraft, competition)
- Do you set personal goals or challenges? If so, what are they?
- How do you measure your success or improvement?

##### 4. Analysis & Feedback:

- After a flight session, what kind of data or feedback do you wish you had?
- Do you currently use any tools or methods to review your sessions?
- What aspects of your flying do you commonly review or analyze? (e.g., landings, flight path, fuel usage, energy levels, war games)

##### 5. Pain Points:

- What challenges or frustrations do you face when trying to analyze your flight data?
- Are there any features missing in current analysis tools that you wish existed?
- Have you ever abandoned/quit an analysis tool? If so, why?

##### 6. Future Desires:

- How would you like to see flight simulator analysis tools evolve?
- Are there any specific tools or feedback mechanisms you'd love to have?
- Would you like integration with any other software or platforms?

## Appendix B: Summative Evaluation Protocol

### Summative Evaluation Questions

*Title: "Wichita 1-1, Fox Three" – The Role of 3D Telemetry Analysis in Combat Flight Simulation*

#### Questions for Players:

##### 1. Open questions:

- How did your last flight session go ?
- Can you recall any interesting events ?

##### 2. Goals & Objectives:

- How would you explore this event in either/both of the tools ?
- Do you think there were other major events taking place in the game during this ?
- Would it help exploring these events ?

##### 3. Analysis & Feedback:

- What kind of data or feedback do you wish you had?
- Do you give/take feedback on other's performance ? How do/will you convey this information ?
- Did either/both analysis tools provide the information and insights you were looking for?

##### 4. Pain Points:

- What challenges or frustrations do you face when trying to analyze your flight data?
- Are there any more features missing in current analysis tools that you wish existed?

##### 5. Future Desires:

- How would you like to see flight simulator analysis tools evolve?
- Are there any specific tools or feedback mechanisms you'd love to have?
- Would you like integration with any other software or platforms?

## Bibliography

- [1] Ana Paula Afonso, Maria Beatriz Carmo, and Tiago Moucho. Comparison of visualization tools for matches analysis of a MOBA game. In *Proceedings of the International Conference on Information Visualisation*, pages 118–126, Los Alamos, CA, USA, 2019. IEEE Computer Society.
- [2] Sabbir Ahmad, Andy Bryant, Erica Kleinman, Zhaoqing Teng, Truong-Huy D. Nguyen, and Magy Seif El-Nasr. Modeling individual and team behavior through spatio-temporal analysis. In *Proceedings of the ACM Symposium on Computer-Human Interaction in Play*, pages 601–612, New York, NY, USA, 2019. ACM.
- [3] Fereshteh Amini, Nathalie Henry Riche, Bongshin Lee, Christophe Hurter, and Pourang Irani. Understanding data videos: Looking at narrative visualization through the cinematography lens. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems, CHI '15*, page 1459–1468, New York, NY, USA, 2015. Association for Computing Machinery.
- [4] Natalia Andrienko and Gennady Andrienko. *Exploratory Analysis of Spatial and Temporal Data: A Systematic Approach*. Springer, New York, NY, USA, 2006.
- [5] Myat Aung, Simon Demediuk, Yuan Sun, Ye Tu, Yu Ang, Siva Nekkanti, Shantanu Raghav, Diego Klabjan, Rafet Sifa, and Anders Drachen. The trails of just cause 2: spatio-temporal player profiling in open-world games. In *Proceedings of the 14th International Conference on the Foundations of Digital Games, FDG '19*, New York, NY, USA, 2019. Association for Computing Machinery.
- [6] Brian Bowman, Niklas Elmquist, and T. J. Jankun-Kelly. Toward visualization for games: Theory, design space, and patterns. *IEEE Transactions on Visualization and Computer Graphics*, 18(11):1956–1968, 2012.
- [7] Alessandro Canossa, Truong-Huy D. Nguyen, and Magy Seif El-Nasr. G-Player: Exploratory visual analytics for accessible knowledge discovery. In *Proceedings of the International Conference of Digital Games Research Association and Foundation of Digital Games*, Dundee, Scotland, 2016. Digital Games Research Association/Society for the Advancement of the Science of Digital Games.

- [8] Sven Charleer, Kathrin Gerling, Francisco Gutiérrez, Hans Cauwenbergh, Bram Luycx, and Katrien Verbert. Real-time dashboards to support esports spectating. In *Proceedings of the ACM Symposium on Computer-Human Interaction in Play*, pages 59–71, New York, NY, USA, 2018. ACM.
- [9] Mark Claypool, Kajal T. Claypool, and Feissal Damaa. The effects of frame rate and resolution on users playing first person shooter games. In *Electronic imaging*, 2006.
- [10] Consultative Committee for Space Data Systems. Telemetry: Summary of concept and rationale. Technical Report 100.0-G-1, CCSDS, December 1987.
- [11] Jamie Cross, Christine Boag-Hodgson, Tim Ryley, Timothy J Mavin, and Leigh Ellen Potter. Using extended reality in flight simulators: A literature review. *IEEE Transactions on Visualization and Computer Graphics*, 29(9):3961–3975, 2023.
- [12] Selan dos Santos and Ken Brodlie. Gaining understanding of multivariate and multidimensional data through visualization. *Computers & Graphics*, 28(3):311–325, 2004.
- [13] Anders Drachen, Pejman Mirza-Babaei, and Lennart Nacke, editors. *Games User Research*. Oxford University Press, Oxford, United Kingdom, 2018.
- [14] Magy Seif El-Nasr, Anders Drachen, and Alessandro Canossa, editors. *Game Analytics: Maximizing the Value of Player Data*. Springer, New York, NY, USA, 2013.
- [15] Magy Seif El-Nasr, Truong-Huy D. Nguyen, Alessandro Canossa, and Anders Drachen. *Game Data Science*. Oxford University Press, Oxford, United Kingdom, 2021.
- [16] Niklas Elmqvist and Philippas Tsigas. A taxonomy of 3d occlusion management techniques. In *2007 IEEE Virtual Reality Conference*, pages 51–58, 2007.
- [17] Tom Farr, Paul Rosen, Edward Caro, Robert Crippen, Riley Duren, Scott Hensley, Michael Kobrick, Mimi Paller, Ernesto Rodriguez, Ladislav Roth, David Seal, Scott Shaffer, Joanne Shimada, Jeffrey Umland, Marian Werner, Michael Oskin, Douglas Burbank, and Douglas Alsdorf. The shuttle radar topography mission. *Rev. Geophys.*, 45, 06 2007.
- [18] Nick Halper and Maic Masuch. Action summary for computer games: Extracting action for spectator modes and summaries. In *Proceedings of the International Conference on Application and Development of Computer Games*, pages 124–132, Hong Kong, 2003. City University of Hong Kong.
- [19] Mouton HD. Comparison of body rotations using euler angles and quaternions. *Rev. Geophys.*, 01 2021.

- [20] Xavier Ho, Juan Miguel de Joya, and Neil Trevett. State-of-the-art webgl 2.0. In *SIGGRAPH Asia 2017 Courses*, pages 1–51. 2017.
- [21] Nate Hoobler, Greg Humphreys, and Maneesh Agrawala. Visualizing competitive behaviors in multi-user virtual environments. In *Proceedings of the IEEE Conference on Visualization*, pages 163–170, Los Alamos, CA, USA, 2004. IEEE Computer Society.
- [22] Robin Horst, Stefan Manuel Zander, and Ralf Dörner. CS: show - an interactive visual analysis tool for first-person shooter esports match data. In *Proceedings of the IFIP Conference on Entertainment Computing*, volume 13056 of *Lecture Notes in Computer Science*, pages 15–27, New York, NY, USA, 2021. Springer.
- [23] Robin Hunicke, Marc Leblanc, and Robert Zubek. Mda: A formal approach to game design and game research. *AAAI Workshop - Technical Report*, 1, 01 2004.
- [24] Erica Kleinman, Sabbir Ahmad, Zhaoqing Teng, Andy Bryant, Truong-Huy D. Nguyen, Casper Hartevelde, and Magy Seif El-Nasr. ”and then they died”: Using action sequences for data driven, context aware gameplay analysis. In *Proceedings of the ACM Conference on the Foundations of Digital Games*, pages 63:1–63:12, New York, NY, USA, 2020. ACM.
- [25] Erica Kleinman, Nikitha Preetham, Zhaoqing Teng, Andy Bryant, and Magy Seif El-Nasr. ”what happened here!?” A taxonomy for user interaction with spatio-temporal game data visualization. *Proceedings of the ACM on Human-Computer Interaction*, 5(CHI PLAY):235:1–235:19, 2021.
- [26] Athanasios Vasileios Kokkinakis, Simon Demediuk, Isabelle Nölle, Oluseyi Olarewaju, Sagarika Patra, Justus Robertson, Peter York, Alan Pedrassoli Chitayat, Alistair Coates, Daniel Slawson, Peter Hughes, Nicolas Hardie, Ben Kirman, Jonathan Hook, Anders Drachen, Marian Florin Ursu, and Florian Block. DAX: data-driven audience experiences in esports. In *Proceedings of the ACM Conference on Interactive Media Experiences*, pages 94–105, New York, NY, USA, 2020. ACM.
- [27] Yen-Ting Kuan, Yu-Shuen Wang, and Jung-Hong Chuang. Visualizing real-time strategy games: The example of starcraft ii. In *2017 IEEE Conference on Visual Analytics Science and Technology (VAST)*, pages 71–80, Oct 2017.
- [28] Yen-Ting Kuan, Yu-Shuen Wang, and Jung-Hong Chuang. Visualizing real-time strategy games: The example of starcraft II. In *Proceedings of the IEEE Conference on Visual Analytics Science and Technology*, pages 71–80, Los Alamos, CA, USA, 2017. IEEE Computer Society.
- [29] Quan Li, Peng Xu, Yeukyin Chan, Yun Wang, Zhipeng Wang, Huamin Qu, and Xiaojuan Ma. A visual analytics approach for understanding reasons behind snowballing and comeback in MOBA games. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):211–220, 2017.

- [30] Zhicheng Liu and Jeffrey Heer. The effects of interactive latency on exploratory visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2122–2131, 2014.
- [31] Aristides Mairena, Carl Gutwin, and Andy Cockburn. Which emphasis technique to use? perception of emphasis techniques with varying distractors, backgrounds, and visualization types. *Information Visualization*, 21:147387162110453, 09 2021.
- [32] Sai Siddhartha Maram, Johannes Pfau, Jennifer Villareale, Zhaoqing Teng, Jichen Zhu, and Magy Seif El-Nasr. Mining player behavior patterns from domain-based spatial abstraction in games. In *2023 IEEE Conference on Games (CoG)*, pages 1–8, 2023.
- [33] Ben Medler. Visual game analytics. In *Game Analytics, Maximizing the Value of Player Data*, pages 403–433. Springer, New York, NY, USA, 2013.
- [34] Ben Medler, Michael John, and Jeff Lane. Data Cracker: Developing a visual game analytic tool for analyzing online gameplay. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 2365–2374, New York, NY, USA, 2011. ACM.
- [35] Ben Medler and Brian Magerko. Analytics of play: Using information visualization and gameplay practices for visualizing video game data. *Parsons Journal for Information Mapping*, III(1):1–12, 2011.
- [36] Paul Milgram and Fumio Kishino. A taxonomy of mixed reality visual displays. *IEICE Trans. Information Systems*, vol. E77-D, no. 12:1321–1329, 12 1994.
- [37] Zachary Pousman, John T. Stasko, and Michael Mateas. Casual information visualization: Depictions of data in everyday life. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1145–1152, 2007.
- [38] Guenter Wallner. Automatic generation of battle maps from replay data. *Information Visualization*, 17(3):239–256, 2018.
- [39] Günter Wallner and Simone Kriglstein. Visualization-based analysis of gameplay data - A review of literature. *Entertainment Computing*, 4(3):143–155, 2013.
- [40] Günter Wallner and Simone Kriglstein. Multivariate visualization of game metrics: An evaluation of hexbin maps. In *Proceedings of the ACM Symposium on Computer-Human Interaction in Play*, pages 572–584, New York, NY, USA, 2020. ACM.
- [41] Günter Wallner, Marnix van Wijland, Regina Bernhaupt, and Simone Kriglstein. What players want: Information needs of players on post-game visualizations. In *Proceedings of the ACM Conference on Human Factors in Computing Systems*, pages 330:1–330:13, New York, NY, USA, 2021. ACM.

- [42] Peter Xenopoulos, João Rulff, and Cláudio T. Silva. ggviz: Accelerating large-scale esports game analysis. *Proceedings of the ACM on Human-Computer Interaction*, 6(CHI):1–22, 2022.