Viewer Interface for First Person Shooter Streaming

by

Eva M. Yeung

S.B., Computer Science and Engineering, MIT, 2015

Submitted to the Department of Electrical Engineering and Computer Science in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science at the Massachusetts Institute of Technology

May 23, 2016

Copyright 2016 Eva M. Yeung. All rights reserved.

The author hereby grants to M.I.T. permission to reproduce and to distribute publicly paper and electronic copies of this thesis document in whole and in part in any medium now known or hereafter created.

Author: 

Signature redacted

Richard Eberhardt, MIT Game Lab Studio Manager, Thesis Supervisor
May 23, 2016

Certified by: 

Signature redacted

Dr. Christopher J. Teitelbaum, Chairman, Masters of Engineering Thesis Committee

Accepted by: 

Signature redacted
DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

The images contained in this document are of the best quality available.
Viewer Interface for First Person Shooter Streaming

by

Eva M. Yeung

Submitted to the Department of Electrical Engineering and Computer Science
May 23, 2016
In Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Electrical Engineering and Computer Science

Abstract

Although eSports has stepped into the realm of mainstream entertainment, the first person shooters still lag behind other types of games in terms of viewer consumption. This is partly due to the genre’s fast paced nature, which makes it difficult for new users to pick up or for experienced users to explain effectively during a stream; thereby limiting its accessibility to the general public. This paper presents an implementation of an interface, based on Epic’s Unreal Tournament’s spectator mode, which helps mitigate these problems by providing stream casters a way to guide their audience better through a more informative HUD display and predictive notifications for where potential action would take place. This work serves as a proof of concept for how to make first person shooters a more viable genre in the general eSports arena.
Contents

Title Page and Abstract

Contents

1 Introduction .................................................................................................................... 7
  1.1 Background and Motivation ................................................................................... 7
  1.2 The Need for a Better Interface ........................................................................... 8
    1.2.1 The Status Quo ............................................................................................ 8
    1.2.1 FPS Specifics ............................................................................................. 10

2 Research and Preliminary Work ................................................................................. 13
  2.1 User Interface Design Principles and Application to eSports ............................ 13
  2.2 The Audience ..................................................................................................... 15
  2.3 Use of Unreal Tournament .................................................................................... 15
    2.3.1 Why Use Unreal Tournament ..................................................................... 16
    2.3.2 Evaluating Spectator Mode ....................................................................... 16
  2.4 Survey of Needs .................................................................................................. 18

3 Implementation ............................................................................................................. 21
  3.1 Showing Current State ......................................................................................... 22
    3.1.1 Minimap ....................................................................................................... 22
      3.1.1.1 How It Works ....................................................................................... 22
      3.1.1.2 What is Its Significance ....................................................................... 23
    3.1.2 Player Statistics ............................................................................................ 24
      3.1.1.1 How It Works ....................................................................................... 24
      3.1.1.2 What is Its Significance ....................................................................... 24
  3.2 Predicting Future Conflict ..................................................................................... 25
    3.2.1 Detecting Conflict ....................................................................................... 25
    3.2.2 Map Notification ......................................................................................... 27
    3.2.3 Camera Jumping ........................................................................................... 28
    3.2.4 Remaining Issues ......................................................................................... 29
4 Reflection ................................................................................................................... 31
  4.1 Postmortem ........................................................................................................ 31
  4.2 Further Work .................................................................................................... 32
  4.3 Conclusion ....................................................................................................... 33

References
1 Introduction

1.1 Background and Motivation

Traditionally, most people associate video games with mouse clicks, keystrokes, and little to no athletic abilities. Because of that initial impression, eSports—short for electronic sport, the term for organized competitive multiplayer gaming—had long languished as a sort of niche entertainment. However, to become a professional gamer, it actually often takes a great deal of mental and physical dexterity as well as the experience and deep understanding of game mechanics and strategy required of a professional chess player. In recent years, it seems that an increasing portion of the general population has started appreciating the skill it takes to be a professional gamer. As can be seen by Blizzard Entertainment’s partnership with TESPA in developing its eSports community division, the advent of Twitch.tv, the establishment of Major League Gaming (the MLG), and the ESPN broadcasting tournaments, starting with the 2014 finals of the International Dota 2 (PC, OS X, and Linux, 2013) Championships, eSports has solidified itself as a source of mainstream entertainment and as a multimillion dollar industry.

Despite the gain in popularity, not all genres are as popular on the eSports scene as others. In the last half of 2015, eSports took up 21.3% of all hours watched on Twitch.tv, the world’s largest game streaming service. [11] 58% percent of those hours were taken up by MOBA, massive online battle arena, games, spear headed by League of Legends (LoL) (PC and OS X, 2009) and followed by Dota 2. On the other hand FPS, first person shooters, only took up 27% of eSports viewership with Counter-Strike, one of the oldest mainstay franchise and the only FPS on Twitch’s Top 10 Games on the eSports scene which also has a similar game duration to LoL, taking up the vast majority of those hours. Even when the hours are broken down by game, Counter-Strike: Global Offensive (CS:GO) (PC,
OS X, Linux, Xbox 360, and PS3, 2012), the second most popular game on Twitch, garnered slightly less hours of viewership than the less popular DOTA2.

This difference is even further emphasized during competition season. League of Legends 2015 World Championship broke records with 360 million hours viewed live between 36 million unique viewers. [7] On the other hand, Counter-Strike's ESL One Cologne 2015 Tournament, while still beating records in the game's own history, only captured 34 million hours and 27 million unique viewers. [8] This means that there are more people viewing the LoL tournament for much longer even though LoL barely has 1.5 times the viewers CS:GO has on Twitch. [15]

1.2 The Need for a Better Interface

A large contributor for this disparity between overall popularity and eSports viewership may be explained by the fact that eSports events draws in many non regular viewers. FPS are notorious for being hard to grasp for beginners so a casual viewer quickly exiting a stream due to confusion would not be unusual. The remedy for this would be to build a better interface so that any viewer can understand a shooter game.

1.2.1 The Status Quo

For the most part, eSports streams visually take after real life sports broadcast, incorporating many of the features that have proven useful in traditional broadcasts. Figure 1 shows a typical screenshot of a MOBA stream and a traditional American football broadcast, and they have analogous components which helps the viewer absorb necessary game state information at a glance and helps casters and announcers orient an unfamiliar viewer so that they can better explain the game. Specific examples of these features include the scoreboards which provide insight for the tide of the game and the mini-map on the video game view coupled with 1st & Ten line on the football screen shot as they both help with spatial orientation. Because of player information is integral to
gameplay, the eSports shot even provides player statistics on the main screen while these numbers are usually relegated to cutaways on the typical football program.

FIGURE 1. TOP: “First Blood” notice from League of Legends 2015 World Championship Finals [5]; BOTTOM: Example of typical football game, note scoreboard and 1st & Ten Line markers. [10]
Despite its adaptation of all these traditional methods, the eSports scene still does not support many of the predictive and visual game analysis tools that modern sportscasters often use. In between plays, sports commentators can make use of these tools to direct the audience's attention to certain plays or analyze potential strategies by highlighting or overlaying diagrams on the field. This sort of predict and analyze pattern is present when streaming, but it usually takes the form of commentators talking on the side and having little ways visually designate the specific objects or area they are speaking about. Still, this works well enough when most of the necessary information is already on the screen, like in a MOBA.

That is not to say that eSports have not made any innovations on their own. Because the exaggerated action on screen in most games, important status messages, like in Figure 1 had become the norm. This clues the viewer into events that they should be paying attention to, and it helps the commentators predict and analyze the trajectory of the game.

1.2.1 FPS Specifics

Although these traditional features work well in most top-down view third person games, like MOBAs, first person shooters have a their own issues to be addressed because of the vast difference in their gameplay styles. To start off, MOBAs competitions work well with a port of the traditional sports interface since MOBAs and regular sports games are already visually very similar: both feature very clear views of the surroundings near where the camera is focused and they usually wait until there is high conflict between players to zoom in on the action. This sort of “anticipate then focus” nature of these two types of broadcast allows more time for the audience and the casters’ attentions to wander and take in the vast amount of information on their screens.

As the name indicates, FPS is a focus on the nuances and experiences that first person perspective gameplay can provide, but that brings it in conflict in with some of the established ways
of broadcasting a competitive sports event. Especially since a broadcasted FPS game is usually experienced by having the focus shift between the first person views between players, it is hard for the casual viewer to reorient themselves and take in the new surroundings without an expert’s guide. In addition, because of the more limited field of view, an uncluttered screen and minimal HUD, (head-up display) is the norm in order to maximize and concentrate on visual feedback from the arena, but that is not always possible during a broadcast as information about all the players need to be shown, as can be seen from Figure 2. Therefore, both the commentator and the viewer would have to spend more active time parsing through what is happening on the screen.

However, because of the fast paced nature of an FPS, high action moments, even in more deliberate and slower games within the genre like Counter-Strike, does not often allow the audience time for information parsing. Expert players move quickly, which causes casual viewers much confusion, especially when the commentators have little time to slow down and explain the situation. This is compounded with the fact that field of view is limited. The decrease in information on the surroundings and other player positions makes it harder to predict what strategies are being put into motion, so commentators have a harder time preempting the situation and easing the constraints on the explanation timeframe.

All this means that the FPS genre is not compatible with the traditional methods of sports broadcasting because they do not provide enough information for the casual viewer to understand.
the game, even with the commentator as a guide. Experienced players and other people already entrenched in the genre would have little problem analyzing the game from with the current set of features or less, but with the increase in eSport popularity, this is no longer the only group whose needs have to be considered. In order to bridge the discrepancy between the popularity of the FPS genre and its standing in the eSports area, pushes need to be made to improve the interface and make it more inclusive and consumable to the casual and novice viewers.
2 Research and Preliminary Work

This chapter details the research and design decisions that went into the final product. It starts with establishing the principles behind the design plan, narrowing down the target audience, researching the technology, and a more in depth look at the grievances of the current state of interface.

2.1 User Interface Design Principles and Application to eSports

The basic principles of designing a good UI are often quoted from Molich and Nielson's paper on “Improving a Human-Computer Dialogue” [9]. For the improved interface, the most important principles to keep in mind are those that pertain to information architecture as we want to streamline and simplify the information being presented to the viewer in an eSport match. Therefore, when designing the interface, it is important to adhere to the following rules:

- **Simple and Natural Dialogue** - keep only relevant information and display them in an easy-to-navigate fashion. Again, the focus is on the action; explicit information is meant to augment the viewer's understanding.

- **Speak the User's Language** - make sure the user understands what is being presented and avoid using overly technical terms. This is clearly violated if the interface is a straight stream from a screen like the left image of Figure 2 because extremely game specific terms and symbols are used.

- **Minimize the User's Memory Load** - keep often used information on screen and cut out as much calculations and memorization as possible. Good examples of this from Figure 2 are the mini-map for navigation and health point gauge.
- Provide Feedback - the system should keep the user updated about its current status. This can take the form of notification messages or feedback when the viewer actively manipulates the display.

- Provide Shortcuts - It is likely that experienced users do not need as much explicit information as new viewers, so it would be beneficial to provide them with more streamlined methods of interaction and information, thus allowing them to focus on the action taking place.

Simplifying these principles to the more generic and popular terms provided by Nielson in the more contemporary article "Improving a Human-Computer Dialogue," these principles fit mostly under learnability (how easy it is to learn to use the interface and encompassing simple and natural dialog, speaking the user's language, and providing feedback) and efficiency (how efficiently the interface can be used, encompassing minimizing memory load and providing shortcuts), terms that will be used for the rest of this paper. [12]

Although these principles were initially designed for more static interfaces, these sorts of information architecture rules can be applied very nicely in dynamic game settings. In fact, in Schutter and Tan's talk about building eSport interfaces, Tan specifically states that most of the features, such as notification popups for completed upgrades and battle casualties, implemented in Gameheart and Overseer for StarCraft 2 (PC and OS X, 2010), a game with a fairly mature eSport scene, and similar viewing overlays, revolve around the idea of "timely information... calling on important information only when it is appropriate." [13] These popup features generally follow all the guidelines provided above — giving useful information and feedback without being too disruptive to the overall viewing experience.

More important points that Schutter and Tan bring up in their talk is the ability to see the field, supporting commentators, and customization of interfaces. According to Schutter, it is necessary for viewers to be able to see more of the map than the players who are familiar with the game since it gives the audience a "better idea of what the players see in their minds." Therefore, the
information displayed to the spectator is potential strategies. For less experienced players, these insights can be relayed to them through commentators. In addition, as with the shortcut principle, not everyone needs all that information. As pointed out in the video, even interfaces in athletic major sports toggle between whether or not they are displaying in-depth information on screen.

2.2 The Audience

There is an interesting case with the target audience of this project. Although the end consumers are the viewers of an eSports game, it is the commentator, people who know the game well, that end up controlling it. They cannot be separated as the commentators are, in a sense, part of the experience of the broadcast interface for the audience because of the information and guidance they are responsible for.

Therefore it bears the two-fold responsibility of making the interface usable by both parties. In such a case, learnability is the focus for the viewers while efficiency is the case for the commentators. A balance needs to be struck between these two so that the unfamiliar viewer gets all the information they need, even if the commentator does not need that information, and the commentator can use the interface without being bogged down by too many extra information widgets on the screen.

2.3 Use of Unreal Tournament

*Unreal Tournament*, the newest title in Epic Games' *Unreal Tournament* franchise, is a first person shooter game developed through crowdsourcing. It is currently in pre-Alpha stages for PC, OS X, and Linux. This project was created as prototype add-on to its Capture the Flag spectator mode.
2.3.1 Why Use Unreal Tournament

Because of the inherent difficulties of creating a multiplayer FPS from scratch, *Unreal Tournament* was used as a base since it already had the necessary architecture in place.

In addition to the ease of starting a project in *Unreal Tournament*, it also includes their signature “Capture the Flag” game mode. As the name says, this is a mode focused on defending a player’s base and retrieving the opponent’s flag rather than the outright death matches many other FPS are structured around. Because of the goals of this mode, teamwork and strategy are central to winning, so it provides a more sportsman-like approach to the genre. It forces players to move around much more rather than relying on “camping,” hiding and waiting for opponents, so it keeps the pace of the game quite high. This is the ideal circumstance for the project to be tested under. This mode also already has a pre-Alpha season bracket, so it will likely be one of *Unreal Tournament*’s main competitive eSports modes.

2.3.2 Evaluating Spectator Mode

In April of 2015, the development team for Unreal Tournament unveiled their work on their Enhanced Spectator Support (ESS) mode [16]. This mode allows players who have the game client but do not want to participate in the game enter the arena and spectate either a playback or a live
match. As this project was conceived around then, add-ons to the interface were determined at that time. Updates to this interface were periodically pushed by Epic throughout the year, and components added to or were being made for this project were replaced by the official versions when they came out since both implementations were similar in appearance and purpose, and the official releases were much more stable.

This section evaluates ESS mode in its release state in the spring of 2015 because that was the starting point of the project. ESS features include a spectator operated camera which allows the spectator to follow a player or projectile, or the spectator can roam around free in third person view as in Figure 4. The player is identified on the bottom left of the screen in first person mode, as seen in Figure 3. Beacons marked the current locations of the flags, and the user had the ability to see through walls at outlines of all the players. Most importantly, large notifications appear whenever the flags change hands.

Although all these features helped facilitate the understanding of the current status of the game, most seemed beneficial to only experienced viewers. It relied on knowledge that may not be automatically or intuitively available to a new viewer. As illustrated in Figure 3, there was no indication of where the camera is looking on a map, and the only point of reference and apparent point of interest is a constantly moving flag. Despite having outlines of other players, it was hard to tell where everyone is without swiveling the camera around or without the use of a map. Also, there was a severe lack of player statistics and indicators as to who was on the teams.

Player statistics and a mini-map were part of later updates to the interface, and they were part of the widgets that worked on and then replaced in the project.
2.4 Survey of Needs

There have been multiple threads on Unreal Tournament’s forums about what people want to see from a spectator interface. [2, 3, 4] In particular, the following few features were repeatedly mentioned¹:

- Camera controls
- Need of a mini-map
- Show player statistics
- Countdown timers for power-ups

The last three items were expected because of the bare bones look of the spectator interface at the time this project started, but it should be noted that in these threads, there were multiple followups on the subject of camera controls. Free moving and preset camera views were mentioned, but multiple people also brought up the desire for an automated camera that followed the action. There were even threads that debated the idea of what smarter cameras could do, such as zooming in and following players in high action areas, and some proposed that these cameras would be helpful in shout casting. [2] However, that type of camera automation would mostly benefit spectators who were already familiar with the game, as they would know what they were seeing. For a casual viewer, that experience probably seems similar to following an aggressive player, so they would likely still need a commentator to set up the scenario to understand the significance of the play despite being able to view the important scenes at the right time.

Despite the popularity of the idea of automated cameras, Twitch streamer and eSports caster, Josh Sutherland, or AskJoshy as he is known online, expresses a different opinion about the use of cameras from a commentator point of view. In an interview about casting Overwatch (PC, Xbox One, and PS4, 2016) AskJoshy says “Third-person camera I think is still really good for setting up

¹ Forums posted pre-ESS release and updates
the exposition -- like 'the defence is here, the snipers are here, tanks are back here' whatever... You basically have to get into a spot where it overlooks the whole scene and you're fine." [1] This sort of simplistic approach to the camera seems to fit the goals of this project well: it is a tool that the commentator uses to convey a bigger picture to the viewers.

That is not to say that the cameras cannot be somewhat smarter to make it easier for the commentator to "get into a spot." It would increase the commentator's efficiency if they only had to concentrate on talking about the game and not worry if the viewers have a good view on what is being commented on. Therefore, rather than an automated dynamic camera system, there is a use for a system that automatically moves a static third person camera to an area of high conflict.

With these grievances and previous interface evaluation in mind, the main issues that this project needed to address were narrowed down to three main issues separated into two categories:

- **Current Game State Information/HUD add ons**
  - Player statistics — a display of important, player specific metrics to gameplay, such as health point, ammo, shielding, etc. Power up timers will be part of this.
  - Mini-map — a representation of the arena that displays the positions of important figures, such as players, goals, and flags.

- **Predicting Future Game State**
  - Conflict Exposition Camera — a system that will predict potential conflict and bring the commentator within view of that area before the action begins.
3 Implementation

This chapter presents the current iteration of the project. It gives a more in depth description of the HUD elements and explains how the new camera system works and the design decisions behind it.

FIGURE 5. Screenshots of the current build. TOP: Camera is attached to Player Samael in third person view. BOTTOM: Camera is detached and is located at the yellow, person-shaped marker on map.
3.1 Showing Current State

Improvements to the HUD provide information about the current state of the game. This information is important since it gives the viewer an insight as to where the player thinks they stand, and these snapshots are all factored into larger strategies should players decide to execute them. Although information given by these features is concurrent with the game, they ultimately affect predicting the future of the match quite heavily. This section describes how the mini-map and player statistics work and the importance of those two widgets for the viewer.

3.1.1 Minimap

The mini-map is drawn on the top right hand corner. It gives the location of all players, the bases, and the flags at any given moment. Figure 5 shows the mini-map in both player following and free roaming modes.

3.1.1.1 How It Works

![FIGURE 6. Close up of mini map. Viewer following the blue mark with the red flag, as indicated by the small yellow circle around it.](image)

![FIGURE 7. Close up of mini map. Viewer camera is free roaming, and position indicated by yellow, person shaped mark.](image)

The mini-map flattens the 3D area into a 2D representation. Most maps in Unreal Tournament do not have too many instances of multi-leveled structures, and player movement largely
occurs in the xy-plane. Therefore, the z-axis can be eliminated since it gives little important information about the location of the tracked objects, and this increases the efficiency of the map.

The red and blue circular markers designate player position and what teams they are on. Adhering to learnability principles, the color of the markers are consistent with the colors of the characters' suits, so that the teams can easily be distinguished. The markers also have a pointed tip, indicating which direction the player is looking at.

Yellow markers show the viewer’s camera position. When the camera is following a specific player, a small, yellow circle appears around that character’s marker, as can be seen in the enlarged mini map in Figure 6. When the viewer detaches the camera and controls it themselves, a yellow person appears to mark their position, as shown in Figure 7. Surprisingly, implementing the free camera marker was not part of the official release, so it had to be reinserted into the project.

3.1.1.2 What is Its Significance

As stated in 2.1, Schutter regards an informative mini-map as a way to see into a player’s mind. Viewers can either draw conclusions from the maps by themselves, or they can learn from how the caster analyzes it. Because the map shows the entire arena, casters like AskJoshy could potentially give exposition about certain parts of the map even if the camera is following another player away from the point of interest.

The yellow markers are very important in terms of orienting the viewer. The bright pop of yellow draws attention to itself, so the viewer has a good idea of where they are looking at all times. This is especially so in the case of the camera being detached from the player since the inexperienced viewer always needs feedback for where they are in the game. In the case the caster wants to follow a player again, the viewer has a good idea where they were before, and would hopefully be less disoriented when they lock onto someone all the way across the map. It is a similar
reason why other games have permanent indicators of where the camera is at all times, like the white box in the mini-map on the top image of Figure 1.

3.1.2 Player Statistics

Simple player statistics, such as player names, health points, defense, and current weapon, are located on the left side of the interface. Any player that is not being followed has a transparent box with a health bar and defense indicator floating above their character, as shown in the bottom image of Figure 5. The statistics for the followed player has changed locations. As can be seen in the top image of Figure 5, it is now a cluster of three semi-transparent polygons in the center of the screen, detailing the player's defense, health, and ammo count.

3.1.1.1 How It Works

Player statistics are color coded by their team on the overview. Once again, the consistency makes it easy to match to the character to their statistic. The health bar on the hover overs are an efficient way for the viewer to gauge relative health quickly without having to parse a number in their heads.

The followed player's key stats are now placed in a conspicuous location. Because of their transparency and muted color, they should not be too distracting to the viewer or commentator despite being in the center of the screen.

3.1.1.2 What is Its Significance

The purpose of player statistics gives an idea how a player thinks they would fare against their opponents. Therefore, it is makes it easier to predict whether they would run towards their teammates and home base for protection when they are low on health, or if they think they are strong enough to stay on the offensive against the opposing team. The caster helps guide the viewers by analyzing these numbers and giving them a good benchmark of when those different tactics would be used.
3.2 Predicting Future Conflict

The tide of an FPS game can change within five to thirty seconds of play. Because of that fast pace and how quickly strategies have to change in order to adapt to new situations, there is too much variability to predict where action would happen too far ahead. Also because of the small size of these maps, if there is a strategic area on the map, it can be assumed that the caster would already know about it, and would not have to be specifically brought over to it to examine it in the middle of the game. Therefore, the implemented camera system predicts where potential high action areas are before the conflict starts, which gives the commentator the option of whether or not they want to jump to the closest camera to that area. That way, the commentator can deliver information about the action to viewers at the appropriate time.

3.2.1 Detecting Conflict

![Diagram showing conflict level detection](image)

**FIGURE 8.** Player checked has dotted outline. Teammates are white circles. Opponent is black circle. LEFT: No conflict detected. RIGHT: Opponent in radius and conflict detected. Red is weight 1, orange is 0.5, and yellow is 0.25.

The heuristic used to determine if an area has the potential for high action is called the “conflict level.” The conflict level is calculated by checking for enemies within close distances to each other in an area. Conflict is determined solely based on proximity rather than weapons firing or
other metrics as the castor may want to be in the area as a team is setting up and coordinating for an
attack. Also, because every player holds a ranged weapon, conflict likely happens once opponents are
in range of each other.

The xy-plane of the arena is divided into grids. Each time a frame is drawn, the predicted
future position for each player is calculated based on their velocity. Using these predicted positions,
the players are assigned to a square in the grid. The grids around each player are then checked for
opponents, as illustrated in Figure 8. If there are none, then it is decided that there are no conflicts
around that player. If an enemy is detected, then each character within a two tile radius of the player
is assigned a “conflict weight.” Those in the same square of the checked player get a weight of 1,
those one tile out get a weight of 0.5, and those who are two away are assigned 0.25. Any flag that
gets detected has its weight multiplied by 1.5 as conflict more likely erupts when a team is
competing for a flag. These weights are then added together to get the conflict value. The grid with
the highest conflict is considered the area of most action in that frame.

There are some special cases to consider before confirming a square as the most conflicted
area. First of all, there might be multiple grids that have the same level of conflict. In that case, unless
one of the squares in the tie was the previous most conflicted area, one of the squares is randomly
chosen. In case of the mentioned exception, the previous area wins the contest for the sake of
consistency. The second case where the previous area remains the chosen square is if the new grid is
one away from it and the difference in conflict level is less than 0.5. The small amount of movement
is ignored since many skirmishes have a good amount of back and forth movement although they
remain in the same general area.
3.2.2 Map Notification

Once the highest action area is determined, a notification is set on the map. A yellow double circle is drawn at the epicenter of the predicted conflict, as seen in Figure 9. Automatic camera traveling was not implemented since the commentator is guiding the viewer.

Although completely automated cameras would ease the duties of the commentator and allow them to focus more on talking about the game, there are times when a commentator would not want to be lured. Take for example the classic decoy strategy. One group can lure their opponents into a distracting skirmish while a smaller group sneaks the flag out or sets up a trap. If the camera was allowed to automatically travel, it would focus on that large skirmish rather than the whole picture, which the commentator would want to follow. Because the commentator’s job is to analyze and explain the game as a whole, it is left up to their discretion whether they think that the area of potential highest action is worth traveling to. In such a case, the “Go to Conflict” button on the left hand side in Figure 6 immediate hooks the camera view onto the nearest preset camera.
position and angle so that the commentator has an efficient shortcut to get the audience into position for the action.

### 3.2.3 Camera Jumping

As stated earlier, the camera jumps to the preset closest to the area of conflict. Other options for where to send the viewer to see the conflict included jumping into the view of one of the players at the scene and calculating the position to put the camera to best see the skirmish. The latter was attempted, but a satisfying solution could not be implemented within the timeframe of the project. Therefore, both were eliminated for the following reasons:

**Player View**

- A good way to determine which player the camera would attach to could not be found. It would obviously be someone detected an opponent breaching their conflict space, but there is no guarantee they would be on the team the commentator would want to follow at that moment, let alone the right person. Also, because the notification was designed to preempt new skirmishes, there was no guarantee how much the player the camera attached to would participate in the action.

- If there is a large skirmish, dropping the audience into first person view may be disorienting, even with the map. It also limits what the audience can see of the fight, especially if it cannot be guaranteed that the observed player leaves the area before the battle is finished.

**Calculated Position**

- It is easy to get the camera within the range of the conflict; it is hard to determine where it should be placed and how it should be angled to get a decent view of action and everyone participating in it.

- There may be nothing to view if the action epicenter occurs in a narrow corridor. The camera often get stuck in walls or end up in an adjacent area separated by a wall.
Although positioning these preset camera position puts more work on the map creator and commentators's sides, it works well within AskJoshy's opinion that it would be best to find a good place to overlook the action in order to make clear comments. These cameras cannot cover all areas, but, as stated earlier in this section, there are strategic places where interesting skirmishes are likely to happen, so a well positioned, preset camera there would help the caster illustrate their points much better.

Another benefit of this type of third person camera positioning is that it makes it the commentator's responsibility to make adjustments to what the viewer sees. Although it is not optimal, it is little more work for the caster to walk the free roam camera to a good position to see the conflict if the preset was close enough to the destination. This would be much harder if the camera is locked onto a player as it adds two extra levels of work: detaching the camera from the player, getting out of the skirmish and assessing its area of effect, and then finding somewhere to view it from. In addition, if the commentator already has a view of most, if not all, of the players in the area, they can choose whose first person view they want to see and attach the camera by clicking on the target's name on the left hand player status section.

3.2.4 Remaining Issues

Although the project accomplished the goals it set up in section 2.4, there are still issues to be addressed that would put a layer of polish on the prototype.

There are some minor tweaks in the HUD elements that would make it much easier to read. It would be ideal to reduce the use of some of the numbers and replace or add health bars or other graphics so that those statistics can be understood at a glance. Color coding the numbers could also be implemented so that the audience could immediately pick out who is low on health, and it would draw more attention when a player is being attacked and losing health points.
As for conflict prediction, more work needs to be done to make the conflict prediction more consistent. Because of how quickly trajectories or plans can change, the area of conflict is prone to jumping around the map in response to those changes. This may be an issue of having the system predict positions further into the future more accurately, but it also has to be balanced with keeping the system flexible to redirection because of the quick gameplay of FPS.
4 Reflection

4.1 Postmortem

This project had its fair share of setbacks. The ones that were the impedes workflow the most but also where I learned the most from were not being able to find a testing group and difficulties building *Unreal Tournament* on OS X.

The situation with the first issue may have been improved if I had a clearer direction for my project at the beginning and had been more persistent about finding test subjects. The initial scope of my project had been very broad. It included making the interface more usable for someone just spectating a match at home rather than focusing on the eSports side of streams. By the time I had the project well scoped out and my audience well defined, it became clear the the caster and viewer combination group would be hard to find, especially if I could not meet them in person for testing. This has reinforced the importance of getting a project clearly scoped out early on. It was also the first time I had a target audience that was not easily accessible within MIT, so I had been naive to how difficult it is to organize tests in real life. Had I been able to gather testing groups, I would have gotten valuable data to evaluate the success or failure of my project.

The most time consuming issue I ran into was probably trying to build *Unreal Tournament* on OS X. I am not sure if it is because most people who build *Unreal Tournament* from source do not do it on their Macs, because most game development does not happen on OS X so there is a lack of expert advice and trouble shooting, or because I just had specific problems that no one had run into before, but after some updates to the source code and an operating system upgrade early in the development process, *Unreal Tournament* kept on failing to build on my computer with many different errors, and I found little to no information pertaining to most them online or through
official Epic resources. In the end, I ended up going over the code in build process and applied ad hoc fixes where necessary. I definitely learned a lot more about how Unreal Engine works and some of the difficulties of developing for multiple platforms.

### 4.2 Further Work

Just as how traditional sports broadcasts have evolved over the years, research into effective eSports viewer interfaces definitely has places to go. For this project specifically, besides the polish issues mentioned in 3.2.4, there is still much of room for improvement and expansion.

In terms of improving the interface, the first thing that needs to be done is to run some user testing. Testing and evaluation would bring up important places where the interface can be changed for the better and confirm which points I correctly implemented.

As for expansion, there are points brought up in my initial plan for the project that would help the commentator build their narrative for the the game if implemented. One of such ideas is the possibility for implementing an overlay where the commentator can draw out plays or strategies on the current view or in a larger map, similar to some ways analysis is sometimes given in traditional sports programs. In addition, the automated cameras discussed in 2.4 has great potential in the replay setting where the commentator can get down to specifics about the gameplay that just occurred.

Because there is currently only one major FPS game in the eSports realm, so there is still time and room to try and research different ways to implement the viewer interface before its look is set in stone.
4.3 Conclusion

As the eSports business continues to grow in popularity, first person shooters will have to find a way to appeal themselves to various kinds of viewers in order to keep up with other competitive games. This thesis attempted to find a way to appeal FPS to casual eSports viewers by creating a prototype of an interface that those viewers could easily understand and help commentators better relay information to them. The end result confirmed some aspects of general spectator interfaces, such as the mini-map and player statistics, that viewers cannot do without and created a new system where commentators would receive alerts as to where conflict would soon happen so that they could lead viewers a place where they could fully view and analyze the strategies that are being put into play. If proven to work, this could help diversify the interests of the consumer base of eSports.
References


