The eSports Statistics Bot

Kiran Ramjisingh

HO 496 Reflective Essay

Honors School

Monmouth University

December 9, 2022

Dr. Ling Zheng, First Reader

This project has both great meaning to me and a wider audience of avid eSports viewers. In a broad context, my project aims to provide detailed information about eSports players and teams in a manner that is convenient. By targeting users of a widely used messaging platform, groups of people can discuss eSports with relevant information available with a simple, autocompleting command.

Unlike athletic sports with easily accessible information, eSports fans often scour the web to find information. Websites such as Liquipedia offer in-depth information for all popular eSports titles but are very crowded, so finding specific stats can be troublesome. Other options include mobile applications and websites, each dedicated to a specific game. While these are good options, they tend to offer less information that Liquipedia does and only cover one game. The ***eSports Statistics Bot*** works to make access to this information easy and fast. Allowing fans to message one another about eSports without leaving their messaging platform, Discord.

The idea for this project came about from my recent interest in eSports. I had never been into athletic sports but had always played video games. The media landscape today is filled with people streaming rudimentary gameplay on platforms such as Twitch and YouTube. I decided that rather than watch casual gamers, I would watch the best of the best. This is a common trend among eSports fans today, and is what pushed me to create this discord bot.

The eSports Statistics Bot has gone through many changes over the past year. The idea was to provide detailed statistics about tournaments, the teams playing in them, and the players making up each team for two games. By focusing on only two games, the amount of information provided would be unparalleled, pulling information from multiple sources and making cards from the data.

To decide which games the bot would cover I created two criteria that the candidates needed to meet. First, the games had to be ones I enjoy. Aside from a personal bias, picking games I am familiar with would help me determine what information is worth displaying. For example, the global ranking of Valorant teams' is much more important than their previous year's win rates. This is only something I can decide on if I have prior knowledge about the game. The second criteria was for the game to have a large following. The bot would not be very useful if it covered games that only a few hundred were interested in. The original two games planned for this project, League of Legends and Valorant, have some of the largest followings in the entire gaming industry.

Through my research it became apparent that detailed information for my selected games was scarce. The first game, League of Legends, is one of the oldest eSports titles and does have tons of data available but at a high price. Detailed, real-time statistics were only offered by a few sources at upwards of $200 per month. The second game, Valorant, is a relatively new game to the eSports scene, being released in 2020. Because of this there was little to no statistical information available. I was able to find a website, PandaScore, that offered a free-to-use API covering both my selected games. The free tier of their service provides basic information about teams, players, and tournaments for each game. Data such as players' real and online names, nationality, age, birthdays, and role in a team. While this information is not as detailed as I had originally hoped, it was enough to provide some insight.

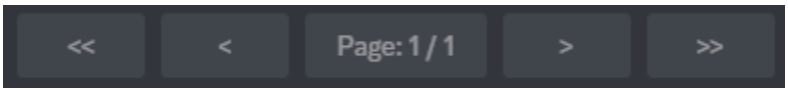
After coding a working demo of the bot, I realized that the scope of the project was too small. Because I was lacking the in-depth information I had intended for, it only seemed right to broaden the scope of the bot to include other eSports titles. Following this, I expanded the bot to cover four more games: Overwatch, Counter Strike: Global Offensive, Rocket League, and

Rainbow Six Siege. After compiling and displaying as much information about each game as provided, it was time to start working on quality-of-life elements. The simplest change was color coding the output of the bot based on the game that was searched. This would give users a visual que about the information being presented, especially helpful to those joining in late to an ongoing conversation.

The next update involved adapting the bot to coincide with Discord's new guidelines. Part of programming anything is thinking about the future of the product and how it will be used today, as well as a decade later. Discord has been making the push for "slash commands" recently, forcing me to redesign my code. The typical way bots were used was with a key character, followed by a command, and then an optional argument. An example of this syntax is "$valorant_player asuna" where '$' is the key, 'valorant_player' is the command to search for a professional Valorant player, and 'asuna' was the name of the player I was searching for. The issue with this is that the bot must read every message and send it to the server to look for the '$' character. In an attempt to move away from this, Discord introduced slash commands. Now users type a slash and a list of commands appear. Typing "/valorant" will give the user three options to search for a team, player, or tournament for the game Valorant. The bot does not need to read all messages anymore, as Discord recognizes the slash character and passes along only this message to the bot.

The last element I wanted to add was buttons to allow players to see search results in one information card, called an embed. Previously if a user's search had multiple results, the bot would flood the messaging channel with multiple embeds, each one for a different search result. I had originally limited the bot to only sending eight embeds for any given search, limiting the clutter in the channel. This could accidentally cut out the result the user was searching for, so a

better solution was needed. Along with slash commands Discord was also pushing the use of buttons, an official way for an embed to be updated through user input. The bot has four buttons under each card it sends. The first and last buttons update the card to the first and last result respectively. The second and third buttons move back and forth through the results. In the middle is a page indicator so that users can keep track of which result they are viewing.



The original plan for this project was a highly statistics-based bot that covered two games in great depth. This plan slowly changed to contain much more topical information, but over a wider range of games. In the original plan, during the research phase, I discovered the developer of League of Legends and Valorant offered their own official API for both. Each had competitive sections with information like win rates and regional rank. What I came to realize in the development phase was that this information was for all players, not just professionals. Furthermore, many accounts were private by default, meaning I could not access their information. This pushed me to search for another API with information more aligned with my original plan. This led me to PandaScore, a service with information on a wide range of games. The catch was that while general information was freely available for all games, the statistical information was locked behind a paywall. These challenges are what caused drastic changes to the project.

There are a few points throughout the project that I can distinctly remember due to their personal significance. The first was when I finished the roadmap for the project's development. I had started this project last semester with the purpose of teaching myself how a real-world

software-development project would work, and completing the roadmap was the first point where I felt that my project was taking form.

Another important moment for me was receiving the first response from the bot. I had stopped the planning phase early during HO 396 because I found the documentation process tedious. I began working on the bot and became obsessed with getting it to work. I ended up creating a rough but working bot that not only replied to my messages but sent information about professional players for the game Valorant. It was at this point I felt that the project could become a reality. The next milestone was a sad one, as it was when I realized my bot would not have the functionality I had originally intended for it to have. The paywall on the API I used for my information was a huge hurdle blocking development. To get detailed, statical information about eSports I would need serious funding, and given the limited time, I had to pivot my idea. I shifted from detailed information about two games, to general information for six games.

My first reader took a hands-off approach with this project. This was by request, as it let me make the project my own and gave me an even bigger sense of accomplishment upon completing it. Dr. Zheng often offered her opinion on the design and function of the bot. This was helpful because she is unfamiliar with eSports, meaning that following her input would make the bot more widely accessible. Her ideas contributed heavily to the way the data is laid out on the embeds. From including the searched term to creating page indicators, these ideas all helped to make the bot as easy to use as it is. She also gave me ideas about some of the functionality the bot should have. Her functionality ideas consisted of possible statistical analysis that could be computed from provided information. While these ideas would have been great additions, they were too large in scope to be completed properly in the given timeframe. As I will discuss in a later part of this essay, I plan to include these ideas in future updates.

The submission of this project does not mark the completion of the bot, it is more of 1.0 release. I have many plans for the eSports Statistics Bot that I plan to carry out over the coming months. The first step is to publish my bot on Discord's public page. This is like a marketplace where users can add bots to their servers. The process for approval takes some time, so it has not yet been published for the public to view. After this, I plan to increase the bot's functionality to align with my original plan. Without the worry of deadlines I can take my time in determining which sources have the information I need as well as spend time incorporating numerous sources into the bot. When pulling information from different places the format it is given in can vary widely. Now, I can spend time picking apart each source and mixing everything into the embeds cleanly. I will also put Dr. Zheng's ideas into effect, doing my own statistical analysis for information that cannot be found easily online. Finally, I want to add more games to the bot as they come out. I limited this version of the bot to six games to keep confusion while showcasing to a minimum. I had originally only planned for two games to keep the scope of the project reasonable but ended up managing six without issue. To make the bot useful to more people, I want to broaden its range. Hopefully the games a user wants information about will be included on my bot in the near future.