Matthew Parnham

2287511

CPSC 408

05/17/20

Final Paper

`A lot of my free time is spent playing videogames. I have thousands of hours invested in competitive shooters like Counter Strike. I am an extremely competitive player and as such, I put a lot of effort into improving my game, learning from mistakes, and monitoring my performance and statistics. In Counter Strike, this is supremely easy. The developers, Valve, have built a fantastic system that serializes entire matches into files called “demo’s”. These can be replayed frame by frame. They can be paused, fast-forwarded, rewinded, and viewed from the perspective of any player or even in a “free” camera mode. I have spent countless hours re-watching demos to improve. Because of this, the game also has fantastic retention policies, storing basic details for every match you’ve ever played indefinitely, and storing the demo to download locally at any time, allowing you to store all your matches, for as long as you would like. Valve also has opened up an API that allows your individual player statistics to be pulled via webhooks to use in your own applications. This, overall, is a very robust way to track improvement and see how small changes can have more broad impacts across your gameplay.

In April of this year, Riot Games, developers of League of Legends, released a new title named “Valorant” to compete directly with Counter Strike and other highly competitive tactical shooters. While it addresses a lot of the gameplay issues plaguing Counter Strike and competitive integrity, it is very new. The game is currently in a closed beta and does not have reporting or auditing features for any stats or old match data. Valorant allows you to access a considerable amount of match data right after the game is finished. You can also go back and view your ten most recent matches. After a match becomes 11 matches old, however, it becomes inaccessible. Since the game features no overall player stats, it would be nice to be able to export this match data to generate my own player stats. Unfortunately, the data is baked into the game’s UI and cannot be extracted easily. As a cherry on top, there is also currently no public API at this time. In summation, my issue is that I want to maintain this high level of auditing functionality in Valorant but am unable to.

I decided to do some research into the issue to see if any solutions already existed before I began mine. I also wanted to get my feet wet a little before I started to swim. I first checked out Riot Games’ developer site. They have an API for League of Legends but have yet to integrate Valorant’s servers into this API. It would seem that this is definitely in the works. Some developers have talked about it, and many popular game stats websites have “Coming Soon” pages regarding Valorant stats. This does not cut it for me, however. I want my data now, and I came up with the following solution.

A screenshot of a video game

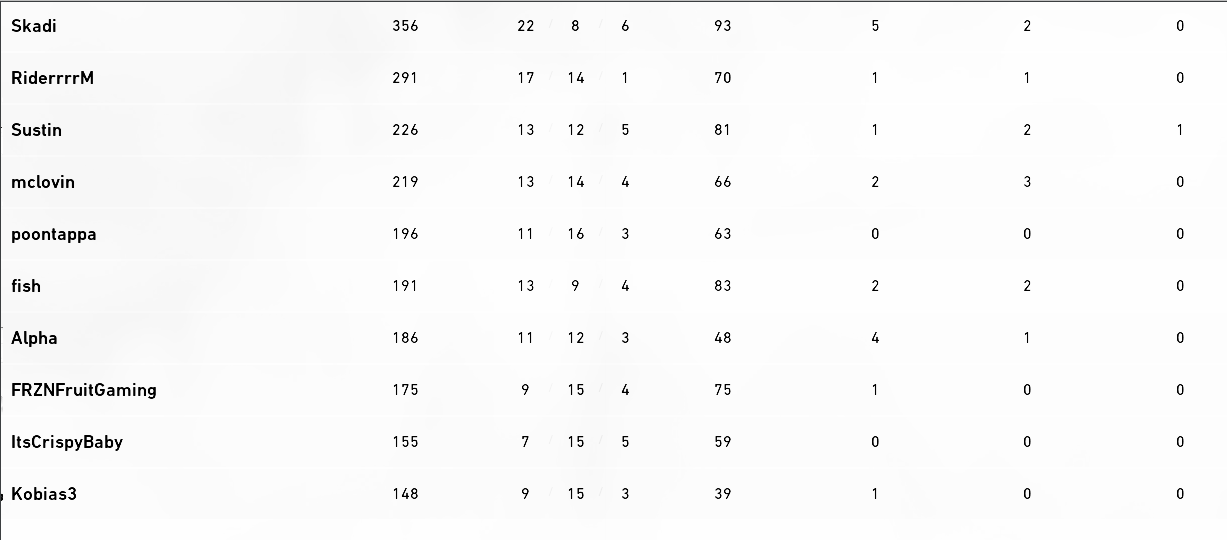
Description automatically generated I sought to find a way to export match data from Valorant into a usable, text format. I would then use a DBMS to store my data for retention and statistical purposes. The easy part was setting up the database. It consists of 7 tables used to store all the match details that Valorant provides. The seven tables consist of a main table that stores the match information, three tables that store lots of supplementary data for the matches, and 3 tables containing game specific data. There is an extra eighth table used to store practice information as well.

After setting up the tables, I attempted to manually insert a match. This process took around an hour, and I quickly realized it was not feasible to do in the long term. Nonetheless, after generating some fake data to populate my database, I was able to actually return some meaningful statistics that the game could not provide, so I knew I was on the right track.

At this point, I shifted gears and began work on exporting data from the game into my database. I settled on the approach off doing OCR on screenshots from the game to try to parse out match data. After some research, I decided to splice, transform, and preprocess the images using OpenCV and then try to do some text recognition and classification using Tesseract, specifically the python connector pyTesseract.

A picture containing colorful, black, tiled, white

Description automatically generated My first challenge ended up not even being OCR but instead being image classification. While I considered using a CNN to classify the images, I realized the overhead was super unnecessary. The images that needed to be classified would be the same every time they appeared, and there existed only 10 of them. Overfitting a CNN to 10 training images was not the correct approach. Instead, I wrote my own classification algorithm. At first, it simply compared the test image with each of the training images to see if all the pixels matched. This worked most of the time, but there were slight variations in the color of the background that broke it. I added a threshold that the pixels simply needed to be similar enough. This helped but did not fix the issue. Finally, by adding a buffer around the edge of the images and only looking at pixels near the center, I negated the discolored borders entirely and was left with only the centers.

As for classifying the text using OCR with Tesseract, I ran into issues as well. A username like “Skadi” was being recognized as “סס؇Ԫ( (“. I consulted a professional, Dr. Linstead, for guidance. He suggested making the text black, as it was currently white. Thankfully, OpenCV can do this easily. Inverting the colors did the trick, and I was rolling. I even blurred the text a bit and thresholded it to make it stand out even more. I had a hard time with single digit numbers for some reason, but I was able to classify a good amount of the data. 

Unfortunately, however, the rapid changes in Valorant, which is currently in closed beta, broke everything. The font, placement of text on the screen, color of text, layout of pages, and even character sprite images all changed. They have been changing almost daily for a while now, every ay bringing a new iteration. This is simply too much to keep up with, and I decided to halt work on this part of the project until the UI in game stabilizes a bit as the game approaches launch. I definitely got the opportunity to learn a lot, and I got some cool, flashy results from it.

As for the rest of the project, I built a Windows form. This was not super challenging since Visual Studio makes the creation of WinForms as easy as drag and drop. The most interesting thing I learned was how to handle various events that occur within the form. I stored my database queries in what were essentially miniature excel spreadsheets. I wanted different update and delete SQL commands to run as I edited, added, and deleted rows from this spreadsheet. It wasn’t difficult but learning how to do that was an aspect of front-end development I have never dealt with before.

My end result was an application that gave me granular control over all my past matches and provided some extra insight into my own performance and abilities. It was super cool to get to put together a larger scale application than most in class assignments call for. Building front ends is not my forte but being able to see my progression from the kind of things I was building 2 years ago compared to now is pretty incredible. In the future, I hope to finish the OCR so that I can easily import my match records into the application. I also have plans to create a user system to allow my friends to use the app. This will allow us to aggregate more data and find more interesting trends.