Project Title

Esportism

Project Summary

Esportism is a resource for aspiring esports players created by University of Illinois students that allows users to reference player and match statistics scraped from game APIs so they can do things like make comparisons of pro players, see average match performances at a given rank, analyze team compositions and roles, or compare themselves against pro players.

For this first (project scope) build of the site, we have taken a dataset containing several relevant data topics from the competitive game Rocket League. What the project allows users to do is be able to filter and search for data points relating to a specific search function that is already defined as a stored procedure. For example, users can search for results from a specific player id, or search for the ids of teams of a specific region, as an example.

There will also be a page for users of the site, where they can tie a username/userID to an in-game playerID and more easily access data points relevant to their own progress. There may be other additions, such as adding favored teams or a region, but at the very least, users should be able to tie to a playerID.

Description of Application

Trying to make improvements in esports is often a very unclear and daunting task. Often, surface level information such as individual match performance, rank, and win rate are some of the only information points available for players to see. However, the games often keep track of more information that could be relevant for player development. What our project aims to do is make this information more accessible and readable by allowing players to narrow down this massive data set into something they want to look for. Maybe they want to see average scoring among players in the next rank, or compare themselves to other players in the region. By providing information in ways that can be customized, filtered, and read out, we aim to provide a resource for players to use to reference stats and to help aim for their own paths to improvement.

Beyond the ability to improve as a player in the realm of esports, we also want to allow for players that are not in the esports scene to read this data and compare themselves to those that are currently in the professional field. Most data sources for esports and competitive games in general are focused on gameplay at the highest level. We want to make useful data available that can be used by players at all levels, but specifically for players at lower pro and non-professional levels since data isn't readily available at most stat websites for these players. This can serve multiple purposes, such as highlighting the gap between different levels of play, which can serve as an illustration of the improvements that need to be made or just to admire a higher level of play, for example.

Creative Component

The creative component of our application will be a feature that allows users to compare their previous match game data against the ranked games from high ranked players. Users will be able to filter and customize their comparisons based on specific criteria such as rank, region, and player role. There will be visual representations, including bar charts and line graphs, to highlight key performance metrics such as goals, assists, and touches. This will make it easy for players to identify which areas they excel in or if they need improvement on. Octane.gg's API will be used to access the Rocket League datasets.

We can try to help make this happen by filtering down games by teams and selecting certain popular teams or popular player names/IDs to show. Putting this into an easy-to-select search can make this task quick and easy. By providing these detailed comparisons and customization options, our application aims to provide a resource for players looking to improve their gameplay. This tool will help players make informed decisions about their games and play styles and help track their progress over time.

Usefulness

Our application will display player and team game data from Rocket League for highly skilled players in the Esports competitive scene. This will enable the user to sort through player earnings and statistics to decide what game they would want to competitively play. Additionally, the application will allow the user to understand what area of the game they should be focusing on to be more competitive. The simple features of our application will give the users the ability to scroll through data and see the mechanics that are more valued within the game. The complex features of our application will allow users to compare their own match data against other players and highlight the parts of their game that could be improved.

In terms of similarities, some comparisons we could make are located at the sites pandascore.co and e-sportsstats.com, but these are very loose comparisons. Pandascore hosts an api that holds in depth match information at the professional level, but it can be hard for the average player to access and view that data, mainly because that data is behind both a paywall and having to set up the API. e-sportsscore is very user friendly, but lacks a lot of key data for players to use. Another thing for both of these sources is that the collection of stats is very professional oriented, focusing on games at the top level, while our proposed implementation allows searching through non-professional games when the data source allows.

Realness

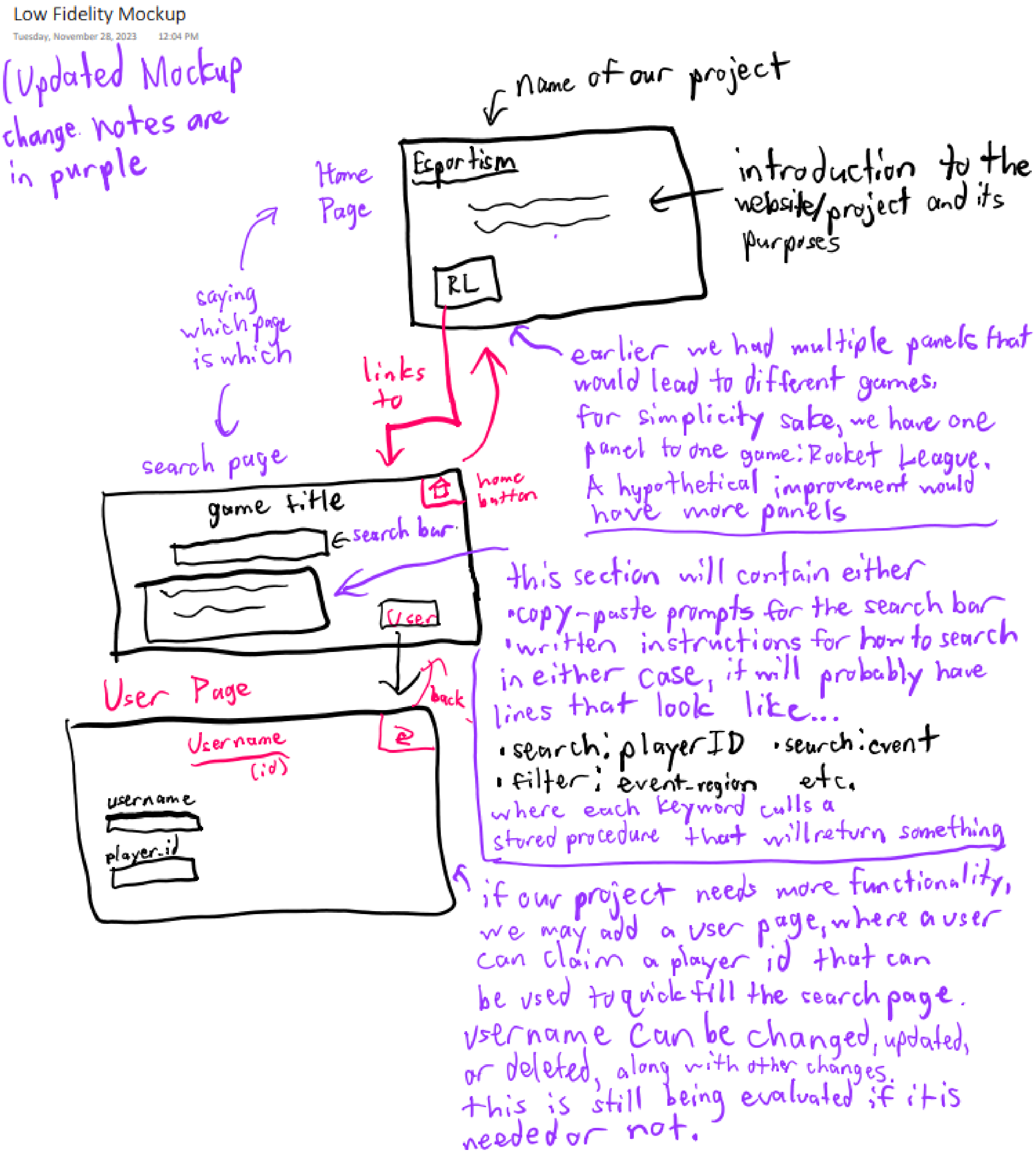
The data for the Rocket League Championship Series is obtained from a Kaggle dataset in which data has been collected using tools from Rocket League replays. These tools involve the octane.gg team collecting data from ballchasing.com using the ‘carball’ library. Additionally, the scope of the data collection is very vast, the cardinality of each dataset is very high (10,000-100,000 unique rows and 30-100 unique columns), and the datasets are mostly sorted in a binary relationship. For example, the datasets are sorted based on games-players, games-teams, matches-players, and matches-teams. Next, regarding the data captured by each dataset, the usual, main stats are present: shots on goal, goals, score, saves, assists. Additionally, the less common stats are also present: boost time, movement time, positioning percentages. Lastly, the format of the data is csv.

Detailed Functionality Description

For a detailed description of the functionality that our website offers, we want to deliver a way to sift through the statistics offered through our datasets. In this manner, we will find ways to organize the data depending on the role of each player in a game. We can compare the different player types in Rocket League. For example, we can allow users to organize the data by strikers or passers.

In addition to methods for organizing the data, we also want to allow users to search for specific players or teams through our website. In this manner, we want to allow users to use our proposed creative component in conjunction with this ability to search for specific players or teams to compare their own stats with specific players. We will not allow users to delete that is not theirs from our website, but we will provide a way for discrepancies in data to be reported in the event of inaccuracies in the backend.

UI Mockup


Low Fidelity Mockup

Tuesday, November 28, 2023 12:04 PM

(Updated Mockup change notes are in purple

earlier we had multiple panels that would lead to different games, for simplicity sake, we have one panel to one game: Rocket League. A hypothetical improvement would have more panels

this section will contain either

- copy-paste prompts for the search bar
- written instructions for how to search

in either case, it will probably have lines that look like...

- search: playerID
- search: event
- filter: event_region etc.

where each keyword calls a stored procedure that will return something

if our project needs more functionality, we may add a user page, where a user can claim a player id that can be used to quick fill the search page. username can be changed, updated, or deleted, along with other changes. this is still being evaluated if it is needed or not.

(Something that was missed making the diagram is that the search bar and the boxes on the User page are all text entry fields, while the user, RL, home, and back buttons are just buttons/panels to link to other pages)

Task distribution:

We will divide the project into 3 categories: Front end, Back end, Data handling

- Front end tasks:
  - Website design
  - Data charts
- Back end tasks:
  - Javascript functionality to website
  - Connecting database with the front end
  - User input
  - API Fetch
- Data handling
  - Merging
  - Filtering
  - Upkeep of database application
  - Retrieval of data

Since all of us have backend experience, each of us will handle one back end task. For the database and data handling, Ethan will be leading the tasks along with Zach. The front end will be worked on by Suyash and Ivan.