Stat-SMITE

# Synopsis

My project will provide statistics for the players of the MMOBA, SMITE. Its intention will be to provide statistics similar to that of websites like “smite.guru”. The difference being that it will be run locally on a machine so that it will only need internet access for API calls. This will provide the end user with online and offline statistics on their recent matches and their progress in mastering characters. The sorting functionality of the product will allow players to see where they need to improve or spend more time playing. For example, A support main will be able to note that their win rates are higher with healers than with engagers. The main goal is to provide the information that players need to improve their weak points and take note of their strengths. Prior to this software there was a lack of offline tools for SMITE players. Along with offline statistics the project will allow users to see how they stack up against their friends.

# Functionality

* A user can see all users stored in the database along with some basic statistics for each
* A user can see a list of all gods in the game along with some basic information about each
* A user can get a sorted list of the average damage for every god
* A user can add a single player to the database
* A user can reset the database and pull-down information for all that player’s friends
* A user can see the matches for any given user in the database
* A user can see total worshippers for every role for a specified user in the database
* A user can find friends who are better than a specified player with a specified god
* The user interface will only be in terminal, but output will be formatted in a dynamic manner

# Database

The ER diagram for this project should have been submitted with this document.

The general idea with this database was to accurately model the database used at HiRez to store the associated data for SMITE. There is a lot of information that isn’t being used in the project but is being stored to protect against the future.

Ability: This table stores a list of all abilities in the game along with important variables and ability descriptions

God: This table stores a list of all gods currently in the game along with important statistics.

Item: This table stores a list of all items currently in the game along with important statistics for each item. Note that ItemDescription is a table for storing the set-valued attribute for ItemDescription

Player: Stores players along with basic information and high-level statistics

PlayerGod: Stores information for how players performed with different gods

PlayerMatch: Stores matches that players performed in

# Stakeholders

The users of this program will be players of the game SMITE who are interested in tactfully improving their gameplay by targeting their weak points. It will mostly be used by core groups of players who wish to track their performance against their friends.

# Technological Requirements

I wrote my whole project in Python 3, a language I previously had no exposure to. There is a database connector for python called ‘sqlite3’. This connector allows for easy queries and simple access to the responses. I initially chose python because there appeared to be a wrapper class for the SMITE API, but this project was abandoned. I stuck with python for a learning experience. The GUI for this project was a sort of pipe dream. I spent a lot of time trying to learn how to develop a website and how to make a simple python GUI. When it came down to it, I have no experience or passion when it comes to front-end development.

# Wrapper Class

I initially selected python for a language because I thought it had a complete wrapper class for the SMITE API. To complete the project, I had to develop my own fully functional wrapper class for the SMITE API. After getting it up and running I realized just how useful to the public it could be. I am now in the process of trying to publish it as the new python 3 wrapper for the SMITE API.

# Dynamic Data

SMITE is a game that is ever changing with not only the match statistics, but also the god/item balance. When brainstorming for this project, I never wanted static data because it wouldn’t seem realistic or practical. In the end of the project, I utilized my wrapper class as well as clever response parsing to be able to modify data on the fly.