Final Project Report

Nathan Zaks

CPE551_WS Python

Mukund Iyengar

link to demo video: youtube.com/

Pygame Pokemon

Introduction

My initial foray in programming was a simple, text-based Pokemon copy in C++. This project allowed a user to pick one of six Pokemon for themselves and their opponent. These two pokemon, each with hard-coded stat values, attack each other until one's hp falls below zero. The attempt for this project was to come back to this recreation in Python.

Updates

This iteration of the project uses several new tools in order to improve upon the previous version. Pygame is a set of modules used to create video games with Python. Requests is an HTTP library which can be used to make API requests. Pokeapi is a service I found which provides API endpoints to retrieve every pokemon statistic used in the game series. Using these tools, PygamePokemon is a video game based on the same video game logic fro he inital game with several new features.

How to use

PygamePokemon, following the instructions in the readme, is compiled using Python3. Upon running the script, the user is greeted with another window which prommpts the user to enter a name. Upon entering a name, the user is then prompted to enter a pokemon name for the user and for the enemy to use. The user can then choose to attack to progress the game until one of the two pokemon is unable to battle (their hp falls below zero).

Behind the scenes

Each pokemon, the user's and enemy's, are encapsulated in a class instance. This class is contains various variables and functions. Some of the variables hold it's name, all of the pokemon stats, and the sprite, among other things. The string entered by the uuser is taken and used as part of the request to the pokemon api. This request returns json data which is then used to set the pokemon stats and sprite. These sprites are then displayed to the screen on one of the available background options. The sprites move across the screen when attacking and trigger damage to the other pokemon. Each pokemon's name and hp is displayed on the battle screen.

Future Work

I would like to add the ability to handle specific moves and interactions between pokemon and move typings. Additionally, I would like to implement more of the story line, not just a battle simulator.

Project Considerations

I approached this project with the idea of lightening the size of the program by using http requests to an API instead of storing and reading data from a file. One such file is seen in the Data folder. This csv file consists of all pokemon names, types, and stats. Instead of using an HTTP request, the pokemon could be initialized using several files such as this to set the stats. Using this approach, the stats, sprites, and all assets needed in the game would need to be shipped with the code as well. Another option, instead of csv files, was to use a MySQL database to store data. mysqlTest in OldScripts shows a test to retrieve data from the pokemon api and storing it in a database. Again, this would result in a relatively large database which would be needed to run the code. This option is much heavier than the current approach.