Pokemon Battle Simulator:

Team Members - Drew Sheppard, Tharid Uralwong, Alexander Pena

Overview:

Our final project will be a Pokemon battle simulator based on the web game Pokemon Showdown. It will contain two modes - a multiplayer mode in which two players with preset teams battle each other like in a Pokemon battle, as well as a single-player mode where the player can battle against a basic AI opponent. Match-making in multiplayer mode will be decided randomly by pairing the two players who have been waiting in a match-making queue the longest. As a stretch goal, we will aim to implement the ability to customize one's team by adding Pokemon from a predetermined list.

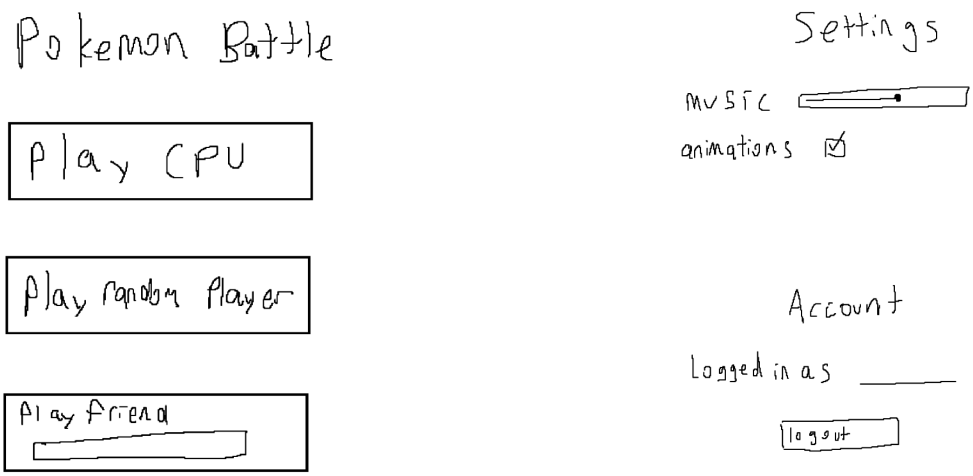
Pokemon in the battle will be displayed as 2D sprites that perform some animation when doing and receiving moves. The game will also have random backgrounds, music, and sound effects for the moves. As in a normal game of Pokemon, the winner will be the first player to knock out all

Pokemon on the opponent's team, after which the game will end and the player will be sent back to the main menu where they may

Front End:

The front end of our project starts with a menu that allows the user to select between single-player or multiplayer games and select different sound/visual settings. However, the main part of our project will be implementing our battle UI where the user can interact with their Pokemon and opponent and do things like, Attack an opponent, swap Pokemon, and chat with their opponent. We will also implement Pokemon sprites and have small movements to help bring the game to life. The game will also be able to change the battle environment the player is in to help make it feel like it is a different battle.

Main Menu Page Layout



Pokemon Battle Page Layout

Backend

The backend will be implemented using Node.js with the Express framework to manage API endpoints and routing. It will be the real application, with the front-end mainly being a way to manipulate the back-end. MongoDB will be used as the database to store information about Pokemon, moves, users, and battles. As mentioned, we anticipate using npm modules mongoose for storing data and express for creating routes. Socket.io and socket.io client will be used to create a websocket that allows for real time updates. For user information, there will be routes to register, login, and get the information about a user. Battles will have routes for getting pokemon information, starting a battle, getting battle state, doing a move in the battle, and ending the battle.

Routes:

POST /register

POST /login

GET /trainer/:id

GET /pokemon/:id

POST /battle/start

GET /battle/:id

POST /battle/:id/action

POST /battle/:id/end

Database structure:

Pokemon schema: name, type, stats, moves

Move schema: name, type, power

Trainer schema: username

Battle schema: players, pokemon, gamestatus, winner

Relfections:

This project was much more complex than we originally thought. To implement this project, we had to learn some socket.io and other interfaces, which we had never discussed in class. We had a month to do the project but I think we could have worked on this all semester to polish our game correctly and it felt like we didn't have enough time with other classes and things going on. We implemented the spec like we described but it's not as polished as we thought. Digital Ocean is also a big headache and I wish we spent more time going over how to implement our project with Digital Ocean in mind.