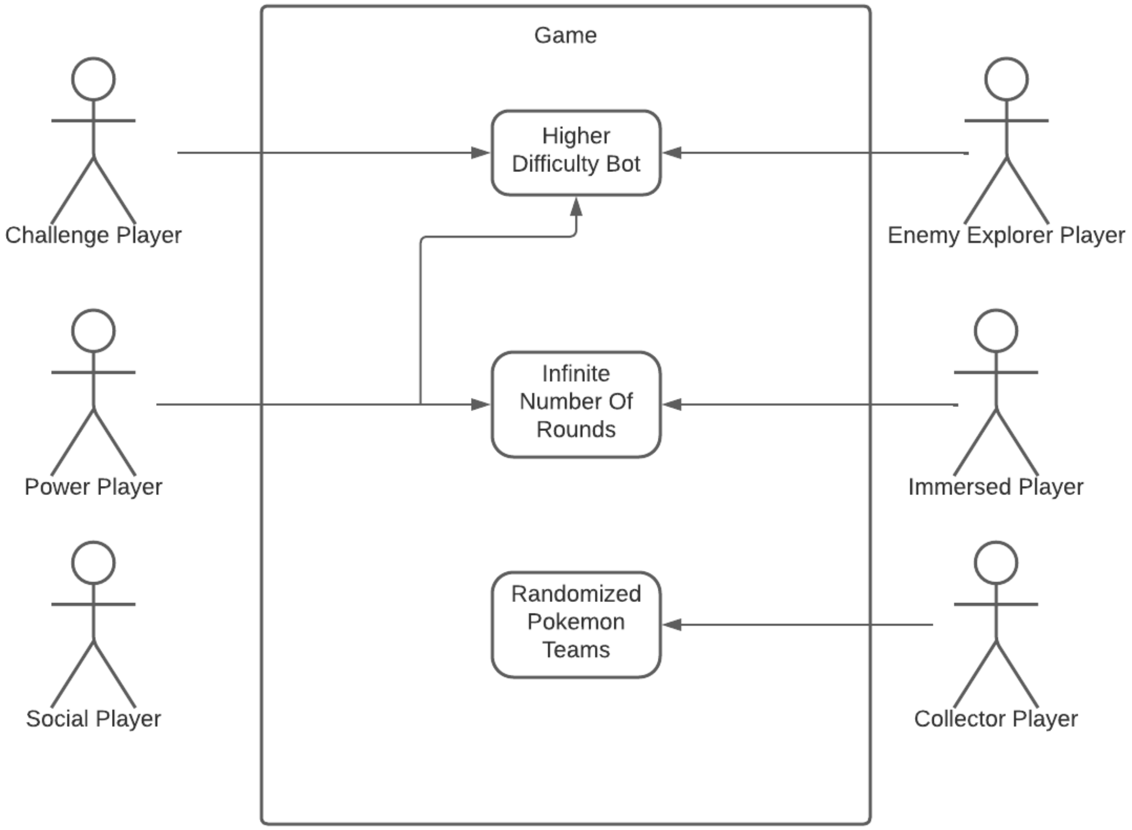
General Problem Statement

In this project, we developed a Pokemon Battle Factory battle simulator based on the Pokemon from Generation 3. Users can choose 3 of their favorite Pokemon from 6 randomly selected Pokemon to form their team and battle against system-generated bot teams through moves and switches. At the end of the game, feedback will be given based on the user's combat record.

Introduction and Background

There are two main aspects in solving this battle simulator problem. First is the main game model which keeps track of what state the game is in, whose turn it is, how much damage should a particular move with types from a Pokemon with specific stats and types do against the bot's current Pokemon who also has specific stats and types, which Pokemon should attack first based on their Speed stat, and so forth. Generally, we started from the bottom up in terms of how the game model worked. Before designing the game, we took into consideration the different types of players that we would likely be tailoring our game towards, which are known as user stories. We initially started with a set of six user stories: Immersed Player, Challenger Player, Collector Player, Social Player, and Enemy Explorer Player. We tried our best to take into consideration all of these different types of players, but under our development constraints we had to neglect some of these players' needs. For instance, we did not provide any social aspect to our game; our game is a single player game (not including the bot). Additionally, there is no collecting Pokemon aspect; at best, the player can consider seeing different random Pokemon as collecting Pokemon. The below UML Use Case Diagram is an illustration of how we considered our different users into our design:



The second main problem is configuring the JavaFX and creating a playable graphical user interface that allows the player to interact with the behind-the-scenes game model. Ultimately, we ran these two in parallel with some instances in which one model had to wait for the other model to update. There were a lot of timing issues, and ultimately we were forced to add non-significant delays in computations in our program.

Motivation

Nate Ahearn, one of our team members, is a really big fan of the Pokemon franchise. The other members of the team really were touched by his passion for the game, and thus we took his ideas to fruition.

Instructions

As long as the user is able to see the GUI, we have designed our game to be very user friendly. The buttons are clear, and the flow of the game is pretty straightforward for anyone who has played a video game before. However, let us make the assumption that the user has never played a video game before. We will walk through how to play the game using our GUI. First, on the main screen there are four buttons that can be clicked. The first to the right is the Rules button which allows the player to read a short description of how the game works if they are not familiar with how Pokemon BattleFactory works. In that screen, the player only has the option to back out and return to the start screen. The next two buttons to the right of the Rules button are the two difficulty start game buttons: Normal and Hard; as the names imply, these will start the game with corresponding bot difficulty level and prompt the user with selecting three out of six randomly selected Pokemon from our pool of 64 Pokemon created for this game. Back on the starting screen, the last button is simply an Exit button which quits the game. When the user clicks the Start Game button, they will see an intuitive interface in how to select Pokemon; in addition, we have added some helpful hints on how to learn more about the Pokemon that they are presented with to allow them to make a better decision in selecting the Pokemon for their team. After selecting three Pokemon, the user will see that the game automatically switches them into the Battle Scene in which they will be first prompted with three choices: Attack, Switch, and Forfeit. The player will see these three buttons in the first half of their turn for every round that they play. Then, the second half of the round depends on what button the player picks. If the player chooses to attack, they will then be prompted with the four moves of their current Pokemon to use. Afterwards, the round will play out (both the player’s current Pokemon and the bot’s current Pokemon will conduct their moves in the order of which Pokemon has the higher Speed stat), then the player will be prompted with the new buttons for the first half of the next round (Attack, Switch, Forfeit). If the player chooses Switch, they will then be prompted by their other remaining alive Pokemon to switch with their current Pokemon. If the player chooses to Forfeit, then they will be prompted again to see if they really want to forfeit and end the current game or play again. If the player wants to play again (whether they forfeited, they won, or they lose), the player will be given a new set of six randomly generated Pokemon to choose three from and start the process all over again. The player’s accumulated stats (wins, losses, rounds played, win rate) will all be shown to the player at the end when the player chooses to not play again.