**Project Documentation**

**C(++)’s Get Degrees**

**Frontend**

In the frontend, the game begins with a home screen that displays the Welcome to Pokemon screen. The user is able to choose between two buttons: Play Pokemon and About Our Team. The About Our Team button leads to a new page that says some information about the team of programmers and course. The Play Pokemon button will lead to a page where the user is able to choose their first Pokemon. The user is able to choose their initial Pokemon from a predetermined list. The available Pokemon to choose from are Bulbasaur, Pikachu, Charmander, and Squirtle. The rationale behind this is that the actual Pokemon game allows users to first begin a game by choosing a strong pokemon. Once both Player 1 and Player 2 have chosen their starting Pokemon, the new Pokemon are then added to the respective player rosters to be accessed in the Battle gui. The user will then confirm that those are the Pokemon that they would like to have in their rosters and then will allow the user to continue to catch pokemon screen. The catch pokemon screen allows the user to enter a field of pokemon. The user can click on the tall grass screens and catch Pokemon that are hidden. This will work by switching the turn between the two Pokemon and again, will update to the respective rosters. Once the rosters are full, a button will appear that allows the user to progress to the battle arena.

In the battle arena, the user will see a screen that dynamically receives text for the Pokemon name, health, and combat power. Each Pokemon will also have an image displayed based on the current Pokemon as well as which players turn it is. The Pokemon will then continue to battle using the battle multiplier as is defined in the back end of the function. When a Pokemon’s health reaches 0, they are removed from the roster, and the next Pokemon’s information is updated. The number of Pokémon left is also updated as well. When a players number of Pokemon left in the roster reaches 0, they lose the game and the user can go back to the home screen.

**Backend**

The backend performs all of the game functionality: creation of Pokemon, the players, and the battling. Pokemon is a class that contains Pokemon name, type, health, attack, defense, and an ID number. There are many member functions as well, but most are getters for the private member variables. We wanted to involve strategy with our gameplay, so the three other functions deal with battling: damage, attUp, and defUp. Damage calculates how much health is subtracted from the Pokemon due to an attack from an opponent with the following equation:

damage = (100.0 - this.defense)/100 \* damage;

AttUp increases a Pokemon’s attack value; an attack value is a member variable that is set, but if a player wanted to make their Pokemon stronger (to deal more damage), they have the option with this method. DefUp increases a Pokemon’s defense value; this makes an opponent’s attack weaker (see above equation). The player class is used to store information about each player, such as their roster and the list of possible pokemon. The roster is made using an ArrayList and has various functions to manipulate it, like adding to or removing from it.

In the Battle class, the fight starts by looking at the two types of Pokemon - type 1 from Pokemon 1 and type 2 from Pokemon 2. As with the real Pokemon games, types matter greatly: an effective type has a multiplied effect when attacking. We created a CSV file that has the types and the multipliers between each type pair. For example, if the two Pokemon are Electric and Water, Electric’s attack is twice as powerful and Water’s attack is half as powerful. Once the damage multiplier is found by reading through the CSV file, we begin the battle. Player 1 chooses first between “Attack,” “Shield,” and “Raise Attack.” Each option then calls the respective method from Pokemon class. For example, if Player 1 chooses to raise defense, Pokemon 1’s DefUp function is called. If Player 1 chooses to attack, Pokemon 2’s damage function is called (since it’s health is decreasing). The Battle class iterates through each Player’s roster, so that it can automatically call the next Pokemon in the list once the previous one dies. Battle also returns the current Pokemon for both players, so that this can be easily accessed for the front end to display the correct and updated information.