Pokémon Battler

Created By: Mitchell Miller, using properties of TPC/Nintendo

The most accurate, inaccurate Pokémon battle simulator.

1. Proposal
   1. Overall description  
      I intend to create a (somewhat) accurate recreation of a Pokémon battle using the Java programming language.
      1. Overview  
         This game will contain several classes, many array-lists and multiple code files. It will be a turn-based combat game, with all 905 Pokémon, and all possible moves for each one. This will be the only Pokémon battle simulator where you can have two magikarps hyper beam each other.
      2. Objective  
         I intend to learn more about the Java Programming language, and how it can connect to external resources such as either the internet, or an external data file to load in large amounts of data. I also intend to translate some skills from C# and C into Java, and utilise them efficiently
      3. Connections  
         This project is related to CS 3120 – Object Oriented programming. This is because I’m using Java, an Object-oriented language.  
         This project is also related to CSE2120 – Data structures, since I intend on accessing external files and reading them to get data.
2. Planning

This project will contain at least 4 classes: mainClass, Trainer, Pokémon, and Move. More will be added as they are required. The mainClass contains the core game loop, and the main functionality of the game. Trainer represents a player and contains an Array List of 6 Pokémon that is the player’s team to use in battle. Pokémon represents, well, a Pokémon, and contains all the stats and attributes of that given species. Move represents a Move that a Pokémon can use.

The main function works as follows:

1. Import all Pokémon, Moves, and other required data from the .txt files in the directory with the game.
2. Ask each player, one at a time, what 6 Pokémon they want to select, and what Moves they would like to give them. There are a few restrictions on what moves a given Pokémon can learn, but it’s a lot more open than other battle simulators. Wanna see a Magikarp use hyper beam? Sure.
3. Once each player has created their team, the battle begins with each player selecting which Pokémon to bring out first.
4. We now begin the while loop that lasts as long as a player hasn’t had all their Pokémon knocked out. I have written this in 5 different phases of the combat.
5. Phase 1: Player 1’s turn  
   This phase is where the player enters their input, and selects what move to do.
6. Phase 2: Player 1 Damage  
   This phase is where damage is calculated for the other Pokémon.
7. Phase 3: Mid Turn  
   This phase is where the game checks to see if there is a winner or a loser.
8. Phase 4: Player 2’s turn  
   This phase is where player 2 selects what move to use, or to switch Pokémon.
9. Phase 5: Player 2 Damage  
   This phase is where damage is calculated for the other Pokémon
10. If no player has won, loop back to step 4.
11. If a player has won, congratulate them, and ask if they want to play again.
12. If the players want to play again with the same teams, all Pokémon HP is restored to full, and go back to step 4.
13. If the players wan to play again, but with different teams

There is also 2 global array lists that contain all 905 Pokémon, and another one for all 825 moves.

1. Development log is in the project folder.
2. Reflection
   * 1. I didn’t need to deviate from my original project at all, and I was able to add all the Pokémon, and moves without any major issues.
     2. I probably would have made the project a little larger in scale and added more gimmicks and features.
     3. The plan helped in that I already knew where everything went, and how it worked, and allowed me to finish this project really fast
     4. I’m quite proud of the super effective system, and the fact that I have all of the Pokémon, and (almost) all the moves.
     5. I would use overridable functions, and add abilities and moves that actually have effects on the game, other than just damaging the opponent. Things to do next year!
     6. I haven’t noticed any large bugs, but I think there was a small one that occurred when I was downloading the Pokémon with my python script, some of the stat calculations didn’t work due to some data from the database being weird. I haven’t fully figured out why this happened though.
     7. I learned more about Java’s syntax, forgot a bunch of semicolons, and learned more about the object structure in Java. Also reading in data from a file, and that it’s a little more complicated than C or C# or even Python.