Project Information

Name(s)	Lynda Flygare
Project Title	PokeBattle
Project Summary	turn based Pokemon battle sim using a CLI where players pick creatures and moves and then battle via turn-based rounds. It will use RNG for damage/crit/status, sort by Speed for turn order, persist saves/logs to files, and parse simple config files using regex for moves/creatures
Project Goals	1- build a modular, bug free battle engine that simulates turn-based combat with creatures, moves, stats, and win/loss conditions 2- provide a user friendly CLI interface that allows team selection, move choices, and readable battle feedback, with logs and saves for persistence 3- demonstrate variables, loops, sorting, RNG, I/O, and regex through damage calculation, turn order, and battle logging 4- get better acquainted with Github and ship a clean repo with docs for an overall well documented project

Grading

Project Organization: (10 Pts) Solo/Group Notification Project Title Project Summary Project Goals/Objectives	10
Technical Requirements: (10 Pts) Upload to GitHub	10
Project Contents/Code: (50 Pts) Utilizing topics from semester Proper commenting Importing module for each author Code executes without error	50
Demonstration: (20 Pts) Concise explanation of project Demonstration of code Reflection on lessons/difficulties (Optional) Next steps / Version 2.0	20
Post-Project Report: (10 Pts)	10
	Total Score: 100 / 100

Comments

- First of all, my “random” Pokemon with “random” moves was Wigglytuff which had zero damaging moves. I demand a refund.

```
Player: #040 Wigglytuff (Type: Normal/Fairy)
  HP:140  ATK:70  DEF:45  SP.ATK:85  SP.DEF:50  SPD:45
  Moves for this battle:
1. Sing  [Normal/Physical]  Pow:-  Acc:55  PP:15
2. Disable  [Normal/Physical]  Pow:-  Acc:100  PP:20
3. Defense Curl  [Normal/Physical]  Pow:-  Acc:-  PP:40
```

- In all seriousness, this is a great implementation of your objectives and a great exhibition of the topics from the first half of the semester and some from the 2nd half.
- You went above and beyond with the packaging of the code and inclusion of a README file.

- Unlike with the homework assignments where I talk about simplifying and not over-encapsulating, this was a great use of classes and functions.
- Next Steps: These are possible ideas for expansion if you choose to continue the topic, not a requirement or even a suggestion. Simply for idea generation if you don't know where to go from here
    - I would love to see this integrated with calls to a site like https://pokeapi.co/ to pull additional moves/sprites/etc.
    - As you mention in your Post-Project Report, this is ripe for a GUI or deployment on a web framework to make the game more accessible and remove a lot of the limitations imposed by a CLI.
    - You are keeping logs, you can use that data to generate visualizations of the analysis of the data, such as most-winning pokemon, most utilized moves, etc.
    - I see a possibility to implement recursion to handle status conditions such that the effect will persist for X number of rounds then ultimately fade.