Benjamin Carson (bkc52), Nicholas Rutledge (njr48), Timothy Eng (te76)

CS 3110

M. Clarkson

MS1: Progress Report

**Vision:** Our current system is a version of Pokemon where you can battle wild Pokemon as well as trainers, catch Pokemon, use items, and travel from town to town after collecting the gym badges. We are currently planning on using ascii art for graphics. From our proposal, our project has evolved in how we are implementing battles, in which we are now typing in “attack [move name]” to execute an attack. We currently plan to streamline attacking in the future, as typing “attack pulverizing pancake” takes more time than selecting a move should take.

**Summary of progress:**In this sprint, we accomplished moving between towns by using Nick’s A3 code as a framework. This resulted in removing >75% of the code (and as a note once moving between towns is done, will look nothing like A3 besides a few command names and similarities in main method). The rest of our progress has all been towards the battle state, which is currently completely separate from the game state with moving between towns. We implemented the ability to parse through a json containing a type effectiveness matrix (currently 6 types only) to 1) create a 2d array containing the modifiers and 2) generate a hashing function to convert element type to row/column index. This can be extended to all future types by updating the json file. We implemented a pokemon module that can create pokemon by parsing a json file, and functions to increase/decrease health and get their stats. We have yet another module for parsing a json file to grab information about given moves, which are then stored for a given module.

We have also implemented a rudimentary battle system, where the player and computer take turns attacking each other. Once a pokemon faints (as currently it is only 1v1) the game ends, displaying the appropriate exit message. The player can pick from a list of their current pokemon’s moves, and can also use a command to show all the information (attack, accuracy, type, description).

**Activity breakdown:**We worked together on almost all of the project so far. We have met several times and implemented most of the project together, so it is impossible to give a detailed list of individual responsibilities and activities. The two exceptions to this are that Nick created the type system/hashing matrix individually, and Tim created the JSONS containing all of the Pokemon moves, pokemon themselves, format, etc. Otherwise, we used LiveShare to simultaneously work on the project at the same time whenever we met.

**Productivity Analysis:** As a team, we were fairly productive. We did almost all of our work over short meetings scattered throughout the weeks, and during each meeting we were focused and got a good amount done. Occasionally, a bug would slow us down, but generally we could plow through the work. We accomplished what we planned, as we have a solid battle state working and a good framework for the rest of the project. We accomplished the satisfactory, good, and excellent scope for the first sprint. Our estimations of what we could accomplish were reasonable, as we completed everything just before the end of the sprint. There was a slight sense of urgency throughout the sprint, but we never felt overwhelmed by the amount of work we had to get done. Also, we never felt that we were running out of things to do or that we were accomplishing our goals too quickly.

**Scope grade:** Excellent. We were able to complete all aspects of our satisfactory, good, and excellent scopes: We can move between towns, have created our functions to create both pokemons, moves, and type effectiveness, and have a working battle state. In fact, our current battle state incorporates more aspects than we originally thought we had time for, including damage scaling, randomness for CPU, and ability to pick from multiple moves (as well as pick multiple moves outside of game, which will soon be implemented in game on leveling up).

The one thing we did not quite get as much done as we originally intended was in our satisfactory scope: moving around. Since we realized moving around between empty towns does not make for an interesting build, we have punted this until next scope when we integrate the battle and moving states in order to focus on the battle state for a more interesting demo. This is also because we are not implementing routes until next sprint, and thus there was even less to add/show on this topic.    
**Goals for next sprint:**

Our first goal for the next sprint, corresponding to the satisfactory scope, is to integrate the overworld movement aspect of our game with the battle aspect. Currently, these two parts are separate, and our game heavily relies on a seamless transition between the two.

Our second goal, corresponding to the good scope, is to integrate routes between the towns in our game. Each route will be fairly complex, with wild encounters and trainer battles occurring. Also, the player will also be able to capture new Pokémon to add to their party on the routes, but this may be too much to do in this sprint and thus may need to be implemented in the last sprint.

Our last goal for the next sprint, corresponding to the excellent scope, is to implement several small changes and upgrades to different aspects of the game to make it an overall better experience for the player. These include adding more types of moves (such as status moves and stat affecting moves), adding items that can be purchased in pokecenters in each town, adding more types for the Pokémon and the moves, and implementing some enhancements to battles such as being able to swap Pokémon and use items.