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

CS 3110

M. Clarkson

MS0: Proposal

**Regular status meeting:**

Outside of our twice a week discussion, which is where we will bring up any issues that we are having with our TAs, we plan to meet Tuesday around 3:30, and have not yet figured out another day – probably Friday afternoons, but will see as the project progresses. In addition, we have a couple other communication/project management systems set up – besides the obligatory Cornell GitHub repo, we also have a group chat for quick issues that don’t need us to be in person to discuss, and an Azure DevOps board, which is an Agile/Scrum project management tool that (te76) has used at work and found helpful for user story tracking, epic/issue/task management, division of labor, sprint tracking, etc.

**A very short statement of the core vision or key idea for our system:**

We will implement a small version of a Pokemon game that will be completely text based. Features will include routes, trainers, wild pokemon, gyms, battles, etc. The player will be able to play the game similar to the originals (not Showdown style).

**A short bulleted list of the key features of the system:**

* Moving from route to route and into gyms, etc.
* Catching pokemon
* Battling trainers/wild pokemon
* Buying items
* Winning (beating the Champion aka Clarkson)

**A narrative description of the system we intend to build:**

This will be a simplified pokemon game (not showdown or lets go) in which there is a starting town where you pick your starter, and then a route to the next town. This route will give a certain number of wild pokemon that you can fight and catch. There will also be trainer battles which behave very similarly to that of the wild encounters, (except cannot catch the pokemon, and allows for several pokemon on computer’s side). In every other town, the player will have the option to go back to the previous town, go to a PokeMart/PokeCenter, Gym, and next route (only accessible after beating the town gym leader). On each route, after a battle, you have the option to retreat back to town, use an item, or keep going. If all held pokemon have feinted, you immediately retreat to PokeCenter, and if this happens when you are battling a trainer, you also lose money. Winning a fight will increase the XP of all pokemon in team, giving a bonus to the one(s) used. On a level up, stats for each pokemon will increase by a set amount, the pokemon may learn a new move (and thus maybe forget one as well), and will evolve at a certain level.

**Roadmap:**

Sprint 1: Alpha

* Satisfactory: Move around from location to location, basically A3 stuff but for our game.
* Good: ability to build pokemon and type effectiveness chart from JSON file (able to hash type of attack and type(s) of defender to get attack modifier.
* Excellent: working battle state: 1 pokemon vs 1 pokemon, picking moves, feinting when hp = 0 followed by announcing the winner and stopping the battle. Decrease # uses of a move for that pokemon when used, have accuracy for determining if move hits and speed for determining which pokemon attacks first (the more ambitious aspects of this may not be implemented until later)

Sprint 2: Beta

* Satisfactory: working battle state (if not already done), change states between overworld and battle, random encounters part 1 (pokemon shows up)
* Good: leveling up after battle & exp., random encounters part 2 (capture, run away), routes/gyms layout
* Excellent: updating pokemon stats and swap party during battle, three fully formed towns (first being starter) and all gyms and routes for them.

Sprint 3: Release

* Satisfactory: Change move set/Evolution & other TBD enhancements from the original games
* Good: Battle enhancements not already done (status effects beyond stat buffs/debuffs, finalization of all towns, routes, and gyms.
* Excellent: other TBD enhancements from the original games, perhaps a choice to quickly battle a wild pokemon, maybe audio

**Preliminary design sketch questions:**

**What are the important modules that will be implemented? What is the purpose of each module?**

* modules for player, pokemon, trainers, and gyms

**What data will your system maintain? What formats will be used for storage or communication?**

Data our system will maintain:

* Player’s pokemon, items, money(?)
* Pokemon stats - HP, XP, ATK, DEF, SPD
* Trainer/gym leader state (if defeated, what pokemon they have)
* Hash table for moves

Format for storage and communication

* Store main game data in JSON
* Use console for showing info
* Commands include GO, BUY, ITEMS, ROSTER, TGM (toggle God Mode for devs testing)

**What data structures do you expect to use as part of your implementation?**

Data structures we will need:

* Lookup table for type effectiveness
* Variants for types
* Json files
* Arrays, lists

**What third-party libraries (if any) will you use?**

We will use JSON files to store move sets for different pokemon and will use the Yojson library to do this, similar to A2/A3, and audio libraries if we get there. Other than that, everything else will be using built in features.

**How will you test your system throughout development? What kinds of unit tests will you write?**

We will use some unit tests to check if pokemon created properly, characters, moves, etc. Also check for illegal moves, win condition etc. Large amount of this will involve playtesting like with A3, as it is a game where we can check for bugs more easily by playing. May also try to find a way to use Bisect if possible/if time.

**How will you, as a team, commit to following your testing plan and holding each other accountable for writing correct code?**

We will use DevOps board to assign tasks and keep track of timeline, bugs, etc. We will also check in with each other in meetings/in discussion and discuss issues with TAs. We will assign tasks to each other as well.