

Project Vulpix

Weekly Progress Report #4

2/13/19 - 2/19/19

This weekly summary report for the fourth week of Project Vulpix will describe the progress of the team as well of each individual member. It will also describe the problems encountered last week and the solutions the team agreed on, as well as what is planned for next week.

Last week we worked on having the player takes turns. Implementing the actions, a player can take on each turn is still being developed this week.

This week we have built upon the work we did last week. We started working on the item and energy cards and what a player can do with them. Each item card can do different actions so each needs to be classified into base action types. Examples of these types of actions are drawing a card, moving energy cards from one Pokemon to another, searching the players deck for a certain type of card, or discarding an energy from an opponent's Pokemon.

Energy cards are attached to Pokemon and they power their attacks. This attaching is achieved via an array of card objects that Pokemon cards have in their class structure. There are many different energy types: water, grass, fire, dark, dragon, psychic, fairy, and fighting.

We implemented retreating Pokemon. Retreating is when a player chooses to move a their active Pokemon onto their bench. This makes it so that the opposing player cannot attack this Pokemon directly. In order to do retreat a Pokemon has a retreat cost. This is an amount of

energy that must be discarded from the Pokemon to retreat. This requires removing energy card objects from the array mentioned in the previous paragraph. They are then appended to the discard pile array. Once this happens the Pokemon is removed from the active Pokemon array and appended to the player's bench array.

We have also created a graphical user interface to serve as a debugging tool. We found that when debugging we were displaying too much information to the console when using print statements. We designed and coded a GUI that can display the information and show that our code is doing what it is supposed to do. We are still debating if this GUI should make its way into the final project. Currently, it was not intended to be anything but a debugging tool. If we finish early or get ahead it might be redesigned and incorporated into the final project.

Next week we will be finishing up the game engine. There will likely still be individual cards to program, but we intend to have the functions to perform all game actions finished by the end of next week. Specifically attacking is the main concern for the next week. This is proving to be one of the more challenging tasks and is why it has been pushed to the end of the engine building phase. Attacks deal damage in addition to performing other game actions. This makes attacks more tedious to program than other game functions. We also plan to start the AI portion of our project next week. We won't have any code for the AI next week, but we should have the design of the AI started next week.