Jason Corriveau
Eric Marshall
Ben Matase
Alexander Murph

UserManual

The program that we have created is designed to allow users to experience the game of Pokemon with our own Bucknell twist on it. The idea behind the game is that users are able to both select their Pokemon and have Pokemon given to them randomly, and then they have the choice of fighting preset or random trainers. Providing these options to users allows both new and seasoned Pokemon players to enjoy the game. Any experienced player can select all of their favorite Pokemon to form a team a fight the challenging preset trainers, while new players can have a team set for them randomly and play against random trainers, or the preset trainers.

Pokemon, the game we are recreating through this project, is a role-playing game in which a user acquires up to six Pokemon. Pokemon have specific types (water, fire, grass, etc) and can learn up to four moves, which also have a type. To play the game, the user trainer will fight enemy trainers, select the move their Pokemon will use, and then the game handles how much damage is dealt to the Pokemon being attacked. If either the user or enemy Pokemon's health is lowered to zero, they must select a new Pokemon to send out from their team, and if they are out of healthy Pokemon, they lose. The motivation behind our project is that we all played Pokemon in our childhood, and wanted a way to have players quickly redesign their favorite team, or fight with a random one. Furthermore, we have customized the game by selecting computer science professors and ourselves as preset trainers to fight against to give the game a unique feel. This is why we set up the game such that users battle each of eight computer science professors, then the elite four (featuring the design team for the game) in order to win.

In order to allow players to fully experience this game, we came up with several user stories. The stories in general all point towards two main goals for the program: allow the player to choose Pokemon and moves for each Pokemon, and let the user battle enemy trainers. These main goals are demonstrated in the UML Use Case diagram, seen in figure one. The first main user story, choosing Pokemon and moves, has five major actors: the player, the Pokemon and

move databases, the Pokemon Model, and the Pokemon Creator Panel. The databases store the data for the Pokemon and the moves, the Pokemon Model stores user selections, the Pokemon Creator Panel is needed to display options for the user, and the player then selects their choices. As seen in figure three, the game provides a user-friendly display, allowing users to either choose each Pokemon individually, or simply click the random button to have a team randomly generated for them.

The second goal of this game, letting the user battle enemy trainers, is split up into battling random trainers and battling the preset trainers. Either way, as seen in figure two, more detail is provided for the use case diagram, as this user story is more complex. For both battling preset trainers and random trainers, the AI Utility is needed to select moves for the enemy, the battle simulator has to handle to logic of each turn, and the Battle State displays the outcome of the user and enemy moves. If the battle is with a random trainer, the random trainer utility is needed to generate a random trainer. Otherwise the Trainer database is accessed to acquire the data of the preset enemy trainer.

As our program has implemented these important players in the system to fulfill key user stories, a user is able to play through Pokemon. A user will begin at the main menu, and can proceed by clicking the mouse, or pressing the spacebar or enter to get to the team chooser state. Once at the team chooser state, seen in figure three, a user can navigate through the slots they have for Pokemon through the arrow keys and press the spacebar to choose and Pokemon and their moves. Or, a user can select a slot to fill with a Pokemon using the mouse. Once a slot has been selected a set of drop-down menus come up, as seen in figure eight, which allows a user to pick a pokemon and up to four moves for it. A user can do this for each slot, or can click the random button to randomize their team. Finally, once a user is content with their team, they can either press enter or click “Done” to proceed to the main menu of the game.

The main menu, as seen in figure five, can be navigated with both the arrow keys and the mouse. A user can select “”Battle Random Trainer” to fight a random enemy, or the other battle option to fight the preset trainer. Lastly, the “Restart” option can be selected to bring the user back to the team chooser state. Once a battle is begun, a user is greeted with a menu of four options, as seen in figure four. As we did not have time to get to aspects of the game featuring

the “Bag” or “Run” options, selecting these moves provides a user with a string that explains the option is cannot be used.

As this menu can once again be navigated with either the mouse or arrow keys, users can select the “Fight” option, which brings up another menu, as seen in figure seven, to select moves. Once a move is selected, the game display will read the results of each move, update the health of the Pokemon, and indicate if a Pokemon died. If the user’s Pokemon died, they will be brought to the Pokemon swap screen, as seen in figure six. This screen has the same layout as if the user selects the “Pokemon” menu option on the battle menu, and the Pokemon selected (through mouse or arrow keys), will update the user’s current Pokemon. If either the user or enemy trainer is defeated, the battle sequence ends, the user’s Pokemon are healed, and the battle menu is displayed once again. If a preset enemy trainer is defeated, the main menu screen will be updated to indicate that the user can fight the next enemy trainer.

The game is “won” once a user battles through and defeats all of the computer science professors and the Elite Four with their team of Pokemon. To keep the game balanced, if a user decides that they want to change their team through the “Restart” menu selection, they will have to begin back at the first preset trainer. In general, menus can be navigated through with both the mouse and the arrow keys and the use of the spacebar and enter key.

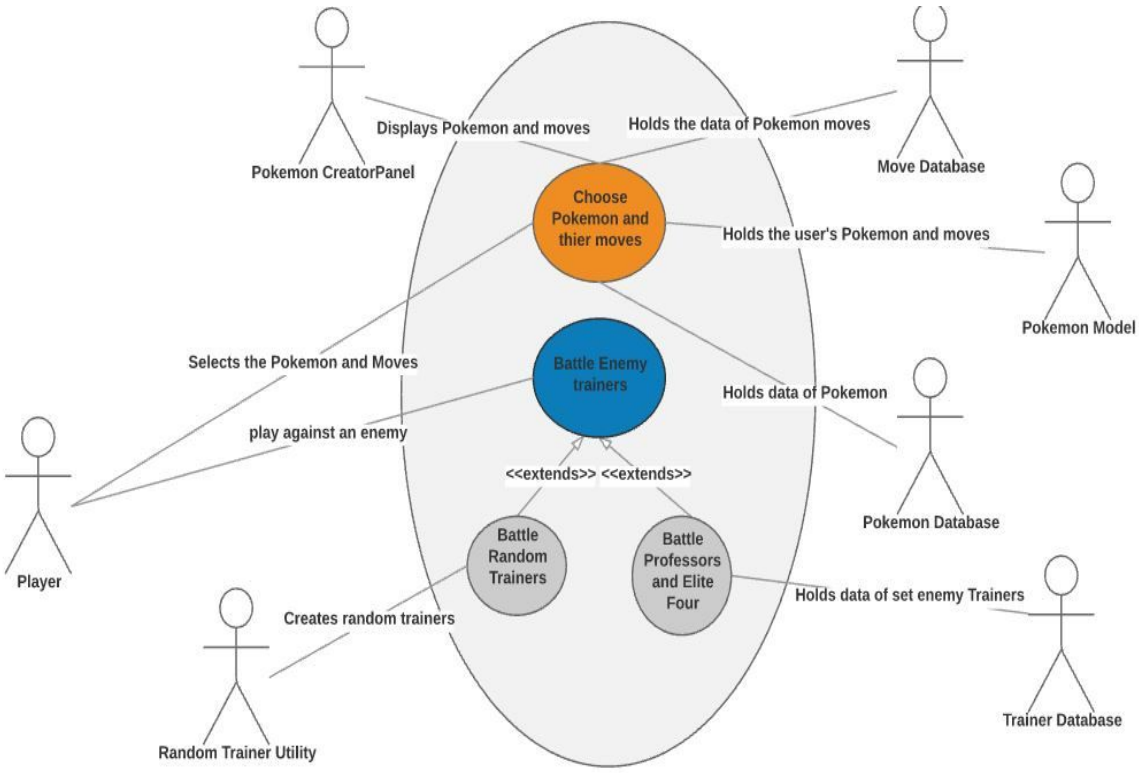

Figure One: General Use Case Diagram

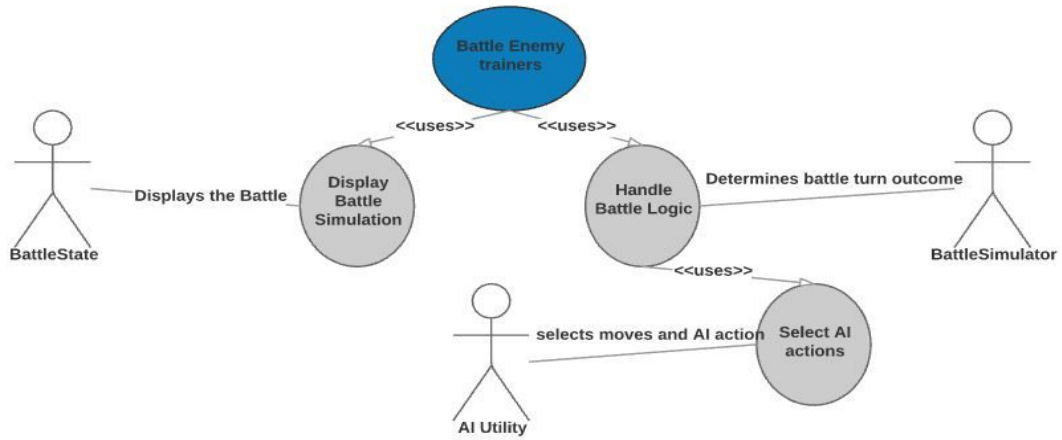

Figure Two: Additional detail for Battle Enemy Trainers

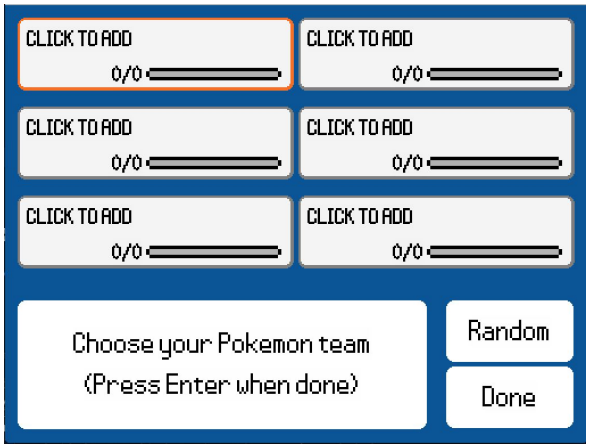

Figure Three: Team Picker

Figure Four: Battle Menu

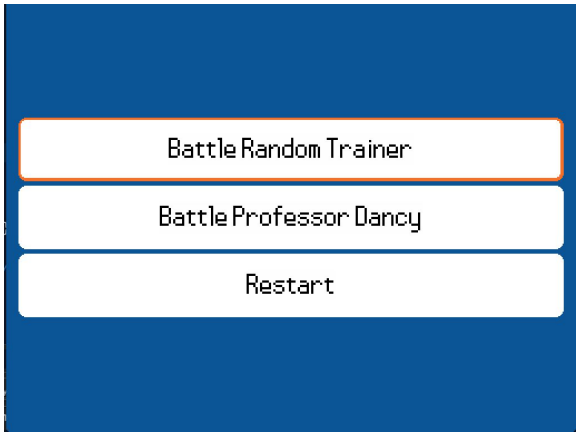

Figure Five: Battle Selector Menu

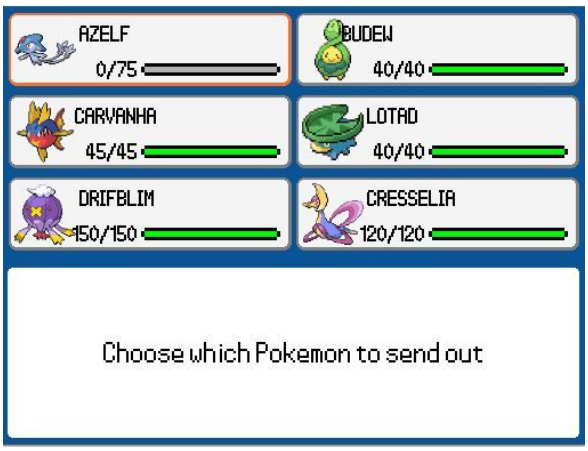

Figure Six: Choosing new Pokemon in

Battle

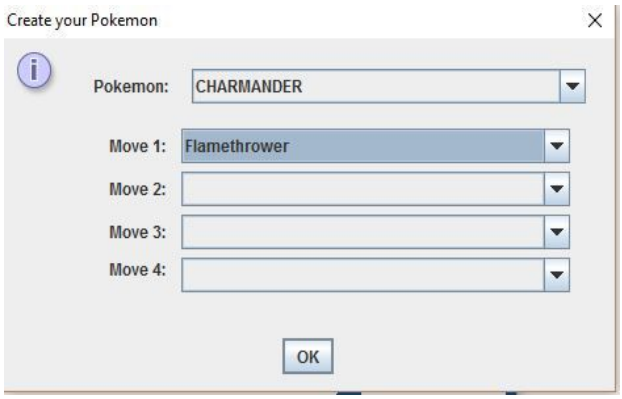

Figure Seven: Move Selection chooser

Figure Eight: Pokemon & Move