EckerdQuest Progress Report

By: Jacob Kula, Zachary Smith, Rachel Halepeska

EckerdQuest is a text-based RPG (Role-Playing Game) based around Eckerd College. At this point in the creation of the game we have been able to complete the tutorial, create the player class, create a functioning map, and create the social battle system.

Jacob: So far, I have worked on the Player class. This class contains the information about the player stats. The player stats are important to EckerdQuest because they allow the player to battle and to create an adventure that they choose. Along with the stats, the Player class contains information about increasing these stats. The main stats that are used are HP or Hit Points/Health Points, sta or Stamina, Social, Wisdom, and Literacy. These stats change given the players choices, for example, The Wisdom stat will increase when a player goes to class and the Library. We are showing that it is very important to prioritize the classroom and the Library, because these stats are a must in order to win (academic) battles.

I also helped add methods to the locations that the player can visit. The locations are important because they perform different situations when they are visited. As I said before when the player visits the Classroom, their stats for Wisdom and Literacy increase. The methods contain the information to increase the stats, and provide some commentary. The methods also show your stats so that you can see what has changed in your stats after visiting the location.

Zack: The map is important for the game to function because it allows for the player to complete various tasks and challenges that have been placed throughout the game. It also adds a visual aspect, which helps with keeping the user engaged while playing. To allow for a properly functioning map there were a few pieces that needed to come together. These pieces include: a method for prompting the user to input the location they wanted to go was created, a method for displaying a photo of the map, methods for displaying a photo for each different location.

To create the tutorial we first spent time, as a group, coming up with the dialog that would be displayed to the user. The dialog that's displayed mostly comes from a fictional roommate, Alex, who gives instructions on how to use and display the map, information about location and what can be done or occur when there, and instructions on how to win a battle.

Rachel: The battle system so far consists of social battles against other Eckerd students. The tutorial battle is a scripted battle, so it will be exactly the same each time. The player has four moves to choose from: run, communicate, joke, and tease. The student type determines which of these is effective. The types are friendly, antagonistic, and clueless. The type for this tutorial battle is friendly, so the effective move is joke, and the worst option is communicate. The correct move will lower the enemy student’s hp. Once hp is at 0, the player wins the battle. Currently, this battle system is separate from the rest of the code. It still needs to be integrated with the player class and main class.

I also coded the tutorial method which mostly prints out the dialogue from our dialogue document, as well as checking that the player inputs the correct commands.

EckerdQuest Progress Report Part 2

Jacob: There are 2 new methods that have been added to the Player class, they are the printStats method and the increaseHP method. The printStats method is helpful for the player so that they can see what their stats are and is a command that the player can use wherever they want after the tutorial. The IncreaseHP method is used during the battle to give the player a Heart piece. Once the player obtains 3 heart pieces from battle it will increase their maximum HP.

I also added some dialogue to the tutorial methods for the player to learn more about the social battle that occurs in the tutorial. I added a method for players to input a help command into. This allows them to input the command ‘help’ and returns a bunch of helpful information to the player that they can input.

Rachel: The battle system now includes three types of battles: the previously mentioned Social Battles, and two new types, Wild Battles and Academic Battles. The corresponding enemies for the two battle types are Animals and Quizzes. The player’s moves for these battles are: feed, fight, picture, run for Wild Battles and remember, common sense, guess, give up for Academic Battles. There is also a randomized social battle now, and the other two battle types are modeled after this battle. For example, the Wild Battle also has a randomly chosen Enemy type, which affects which answer the user should choose in order to win. These types are fish, squirrel, and dolphin. These battles work like a puzzle, where the player must figure out the correct sequence of moves to win. There are some random chance elements that determine whether or not the player’s moves are successful as well. This is similar to how the Academic Battles work. The player must get 3 out of 5 of the questions correct in these battles in order to win, but the player’s success is partially chance based. The other determining factor is the player’s literacy and wisdom stats. The higher these stats are, the better chance of success for the player.

Zack:

The main now includes a new method called gopavillion. The method checks the day counter to make sure the user can access this location. If it is day 6, 7, 13, or 14 the player is then presented with dialogue that explains which event is currently occurring and then calls the Social Battle class 3 times. Also, the map was updated to give the player a better idea of what each location is. Instead of just using numbers to represent the locations, an icon that represents the location was added. There were also two smaller improvements to the program. The first was switching user input to the map method from int to string to make sure the program doesn’t crash from incorrect user input. Another improvement was adding icons to the map to give the player a better idea of what each location is on the map.