Chad Hodson

Prof. Arias

Software Dev 1

4/6/17

Project Milestone

The project that I chose to do was a text based adventure game. The player starts stranded on a beach and must move around and explore an island in order to find a way to escape. You are given more information about what to do as the game goes on and you explore more areas. There are significant items and parts that will help the player complete the game as well as riddles and other elements. After completing the exploration section of the game the player must then battle a snake and defend himself until help arrives. Nothing from the adventure part carries over it was just another feature I added later in the development of my game. The snake is given two different attacks and the player is given two attacks and a dodge. Once you defeat the snake the game officially ends. The goal of the game is to get off the island and make it back home.

I am enrolled in the game design major and I wanted to do something game related so I figured this would be my best option. I am a big fan of puzzle type games where there is a lot of thinking required in order to advance. I also feel that I could be more creative with a text based game since I would be able to control the story and what happens instead of just programming a standard game like checkers. To me trying to think about what items and areas to incorporate an sort of build my own small world was a really fun part of the project.

A majority of the situation is presenting you with information and the player must choose what to do with what they are presented for example. Every time you enter a new area it will give you information about that area and possible hints to let the player know what to do. For example there is a dark cave that the player cannot see down normally. They must find an item to light up the area before they can advance. The idea behind this was to try and immerse the player as much as possible and There are also dangerous wild animals that they player must either choose to run or attack. The game will prompt your decision.

The main way the user interacts with the program is they are prompted to enter a list of commands. The player can enter “north”, “south”, “east”, or “west” to choose a direction and they can also enter, “examine”, “take”, or “drop” to interact with items. There are also a few miscellaneous commands that include, “quit”, “help”, “escape” and then there commands to activate items such as “map.” The program then outputs information based upon what the user has entered.

For the second part of the game, the program chooses what attack the snake is going to perform based upon a random number the program outputs the decision the snake has made and then prompts the user what they would like to do. The program then outputs whether you or the snake got hit depending on what the player chose to do. The program ends when you or the snake die.

I tried to keep the game very small in scale because I kept running into little bugs and errors in my coding which really brought down my motivation and sanity. One of the reasons I ran into so many errors was because of the way I coded my project and stupid mistakes that I just wasn’t catching. So much of my time was sorting through my code only to find that I was missing a bracket or had put a single equals sign instead of double equals signs.

My main class is essentially one big if else statement inside a while loop. I had started this project very early on using loops and by the time I realized I had made a terrible decision it was too late to go back and redo what I had done. In hindsight I never want to do something that messy again it was long, inefficient and just painful to go though and make changes. Again at a certain point I realized what I was doing was a very dumb idea but I had to keep going with and and I made the if else statements longer. One positive that came of this was my code was easy to read (for me atleast) and it was also coded in a very simple fashion so when I ran into an error most of the time I could fix it. It wasn't necessarily trying to fix the error in my code but trying to find the error was the hard part.

Going more in depth with that just one slight mistake was having one string variable prompt everything. I could have made separate variables to prompt certain actions in specific locations. For example there is a code that is need to be entered into a safe which I should have turned into a separate variable but I left it under cmd. This would have cleaned up my code slightly and would have made life just a tad bit easier.

When I had the idea to add a second feature into my project I made sure to take a more organized approach to it. I immediately started writing down potential function names that I knew I wanted my game to perform such as getEnemyAttack and playersMove(). It made my life so much easier and also solidified how much of a bad idea one big if else statement was. After defining the functions i started adding to them and testing the code. I figured out that I had made a few useless ones and just joined them with other functions.

The goal of the program is just to provide the user with some entertainment for a few minutes. The game is not very hard to figure out and there not a lot to it. It's just suppose to be a simple quick game to enjoy. The only goal the program really accomplishes is entertaining the person who plays it.

RealCombat()
int health int snakeHealth int fastAttackDamage int strongAttackDamage int snakeFastAttack int snakeStrongAttack int currentHealth String playerDec boolean strongOrFast
+playersMove() +getEnemyAttack() +chooseEnemyAttack +enemyStrongHit +enemyStrongMiss +enemyFastHit +enemyStrongAttackPrompt +enemyFastAttackPrompt +promptUser