Tawan Scott

Professor Arias

CMPT_220L_201

4 May 2018

Final Writeup

Abstract

This project is a text-based adventure game that lets the player advance through the game at their own pace and order. This means that there will be different options and paths for the player to freely choose from, which will affect how they play the game as well as bringing in an element of the RPG genre.

Introduction

The reason that I chose to make a text-based adventure game for my final project is that I've always been a big fan of RPG games ever since I got my first computer as a child and it is also one of the reasons that sparked my interested in computer science. Therefore, I think that there is nothing more fitting than to make an RPG game for my final Software Development project.

This paper will cover the process of the coding that was done to make the game, which will include in depth descriptions of how the program works, how the user can interact with the game and a UML diagram. There will also be requirements for the game, literature survey, user manual, and a conclusion that will reflect and summarize the process of making this game.

System Description

There are four different paths that the player can explore and these paths have their own purposes. The Temple is where the player can pay the priest to be fully healed, costing them 20 gold coins. The player will start off with 60 gold coins and they will have the opportunity to obtain more coins by killing the monsters in the game. Forest is the location where the player can answer three riddles in order to win over a powerful weapon, which will help in their adventure. Braavos' Gate is where the player can meet or attack the guard and this is where the game ends depending on the player's action. Monster's cave contains three different monsters that the player can fight. As for game mechanics, randomize and set calculations are used in combat such as player's health, enemies' health, enemies' damage, and player's weapon damage. Each enemies will have different amounts of health as well as the damage they can do. I coded the attack so that each attacks are calculated randomly and the player's weapon will also play a factor in these calculations as better weapons will have a higher ceiling of damage numbers that it can be calculated too. This done through the use of java.util.Random.nextInt(n) where the number of n will be determined by the weapon being used. This will encourage the player to explore in order to find better a weapon. In combat, the player will have two options, which are to attack or to run. These options will be selected by pressing either 1 to attack or 2 to run. Most of the coding are comprised of If and Else statements.

Each location will lead to other locations in the game world. For example, if the player is at forest() and they input 2 and this will send the player to riddle(). This is how most of the different locations interact with one another and the user's input will determine which location they go to next.

The end goal of this game is to enter a town called Braavos, which is sealed away by guards from travellers such as the player's character. So to enter this town, the player will have to take on a quest given by the guard and that will be the main objective or they can choose to fight the guard. The pace and ways in which this is done is entirely up to the player.

UML Diagram

Game
-riddleDone : boolean -monster : boolean -monsterTwo : boolean -allMonsters : Boolean -coin : int -playerHealth : int -monsterHealth : int -monsterHealthTwo : int -monsterHealthThree : int -guardHealth : int -choice : int -globinHead : int -playerName : String -playerWeapon : String
+newGame() : void +braavosGate() : void +guardCombat() : void +guardEngagement() : void +crossroad() : void +temple() : void +forest() : void +riddle() : void +riddleTwo() : void +riddleThree() : void +monsterCave() : void +giantRatEngagement() : void +combat() : void +wolfEngagement() : void +combatTwo() : void +globinEngagement() : void +combatThree() : void +death() : void +victory() : void +victoryTwo() : void +victoryThree() : void +gameEnd() : void +gameEndTwo() : void

Requirements

Any computer with Java will be able to easily run this programme.

Literature Survey

There are many other games out there that have similar mechanics to this game. A lot of these games are under the category of dungeons and dragons as these games involve a lot of text-based actions to play.

User Manual

To start the game, you must run the Game.java file. First, you will be asked to enter a name for your character. This will be the only time in which the input you enter are letters. After this, the options that you can select will be listed in numbers. For example 1. Temple, 2. Forest, 3. Braavos' Gate, and 4. Monster's Cave. If you want to go to the temple, then all you have to do is type 1 and press enter. The same goes for combat (1. Attack and 2. Run). The rest of the game will be played in this manner. There are two different endings that the player can choose from and these endings will depend on the player's actions and choices.

Conclusion

This project has been a great learning experience for me as I used the skills that I've learned during my freshmen year of programming. It was also a long process that has taught me time management as well as problem solving skills. There were many occasions where I had to go back to fix my codes in order to make the game flow nicely. Overall, it was a very good experience for me and I'm very proud of what've created.

References

The game that inspired me the most during this project was the Baldur's Gate series. It was the first RPG that I played and the riddles that I included in my project came from the series.