Gerard Andrews

4/1/2017

Prof. Rivas

Software Development 1

Project 2 Milestone

For a brief reminder of what I am doing for project two, I am current making an application project that features a text based game. I have made large amount of progress since my proposal and I may be able to step up and make more additions that make the program more efficient. As someone who is aspiring to be a game designer it’s important for to understand basic game loops and how the process goes. This will help in making more complex games in the future.

For the detailed system description, this will be going over the class and how it interacts with everything that makes the whole game. First, the program needs variables that every method needs to access, so while this was in development the best way to make this possible to me is creating variables under the class itself using public static int. In total, there are four variables: an int for the player’s current location, a list containing all the Booleans for whether the player has visited a location, a list that contains all the descriptions when the player enters a location, and an in for the player’s score. These variables are used throughout different methods, specifically the current location variable.

After that the main method was established and the scanner was ready to go. I started by setting up initializing the player’s location, this meant setting up his current location, setting up input so the player can put in his name and setting the backstory. Once this is done the game loop that is the core part of this program can be started. This is done using a while loop.

The game loop consists of three functions, prompting the user for input, updating the game state, then rendering a scene. For prompting the user for input I setup, a variable called cmd (command) that has specific intake for what is put in. The commands for this input give a response are, directional input like north (n), south (s), east (e), west(h), help (h), quit (q), points (p). If none of this is inputted the user will be informed that the input was not valid and will be asked to input a command again until he types a valid input.

For the directional inputs the command depends on where the player is, for example if the player types “north” or “n”, the game will check for one of the if statements regarding north if there is a statement that complies with that situation then the player’s current location variable will move to the corresponding location and then it can move on to step 2 for the game loop. If none there is no if statement that complies with that location for that command, the layer is prompted that there is nothing in that direction and will be asked to enter another command. The help command will print statements regard what are the basic commands of the game to help this is also prompted when the user is first asked to enter a command. The points command will the display the player’s score which is given based on how many locations he visited for the first time. The quit command will break out of the game loop and end the program.

Once an input is done and it complies with something, specifically the directional input the second step can begin, updating the game state. This consists of multiple things, first there’s moving the current location which is specifically changing the variable, then the game needs to determine if the player has already visited this location or not. If it was not visited at all the variable for whether the location was visible or not is switched to true and the player is a rewarded with points for his score variable. If it’s already visited the variable will be kept true and the loop can move to the third step.

The third and final step is rendering a scene this is essentially telling the player where he is in terms of locations. For this calls for the list of descriptions back in the class. The list of descriptions is based on the number of locations this means one of the locations such as loc1 will have the same variable number as description1. By doing this I can tell the game to load a description based on the player’s current location. The current location variable is also shared by the location numbers as well. Once the description is displayed to the player the program will loop all the way back to the player being prompted to enter an input.

There are a few other methods that help with the basic game loop. These are all specifically for helping with the basic game loop. For example, one method displays all the information after input the help command. Another displays a message for when the user inputs the quit command. Future methods are all going be for helping with the basic game loop usually.

Text based games are generally easier compared to other types of games. Some programs do handle the job better though. Specifically, python handles it better because it’s generally less picky with formatting and it has the use of actual global variables that allow the player to be more flexible with variables. I’m not a fan of using public static int with java because of the dangers it brings but I wanted some way to use variables without needing to pass them in for every method. There are also game engines such Unity and Unreal that sport the use of Java, Python, and C#. C# is generally used on this engine because it’s much friendlier to game developers who are coding and building for specific systems.

This text based game has a similar style to ZORK where the player must enter an input then the game state is updated and a scene is displayed. This basic game loop is extremely common in many text based games and there is hardly room for major change. Future addons to my game such as taking items which can be used to enter different locations are also common to see in these games.

For the user manual, this system is specifically used for entertainment purpose, it does not have any other function and requires the player's participation always. The user must at all times have an input until the player is informed that the game is over. In terms of physical requirements, the main requirement is that the user must always put in input. There is no other physical requirement that the user must do.

The last thing to really look into doing, I want to try and make my code more efficient. What I want to do specifically is create matrix that stores locations, descriptions, and Booleans into one specific list. For many this is known as a navigations matrix, otherwise it's a list of lists. This can be done with the use of a multidimensional array that stores a string for descriptions, the name of the place, the Boolean to check if locations are visited and the number of items in the list correspond to the current location variable.

In conclusion, the basic game loop currently function exactly as it's supposed to, albeit with the use of public static variables in the class. The project is making leaps and bounds because it's my first time making such a project on java and knowing how to code games on java is good future reference on the mobile game industry although it is better to make on coding languages such python and C#.