THE WONDELAND SAGA: The Final Showdown

Emil Paul: 3698580

RESERCH DOCUMENT: ASSIGNMENT 4

This Assignment was the most challenging one of all the Assignments. The creation of this game required a lot of research outside the curriculum and a lot of time watching tutorials. To develop the Wonderland text-based game, I began studying classic text-based adventure games I could find online, like Zork and Colossal Cave Adventure. These games provided valuable lessons regarding key gameplay, command-based navigation, item handling, etc. This was probably the foundation that led to building my own game. Apart from that I delved into some Java programming techniques mostly focusing on multiple-file handling. I learned how to load data from text files into objects which was the first crucial step. This was pretty challenging, I had to decide what methods in Java I should use for it. Ultimately I find this common method enabled me to develop and edit the game logic without actually updating any of my code logic. With the help of Object-oriented programming principles, I designed classes and ensured clear understanding and separation between them. I made sure I used variable and method names that could be easily understandable by a new user who was reading my code.

Designing and organizing the code was the most challenging part, I decided to create separate classes for Location, Item, Character, Control, and the main game class itself. For example, all data related to a specific area in the game is handled in the Location class, which includes an inner class called ‘exit’ that handles directional exits which ensures proper navigation.

Loading data from text files externally was tricky as well, I was initially confused about what information I should include in it and what format I should choose, so I could create a general method to read through them. I created a technique that gives a detailed description of a location when arrives the first time, and a shorter one when revisited with the help of adding them into ArrayLists once visited. I created a total of 5 text files with location, items, characters, action, and exits. Additionally, I wanted to create a storyline that follows, for that I rewatched the original ‘Alice in Wonderland’ movie and took notes on important characters and items. I also introduced a lot of personal childhood favorite characters and mythological items in my game to make it more personal.

During development, I encountered several challenges, particularly with file handling and data parsing. Ensuring the data loaded from text files was correctly loaded and handled was challenging, especially when dealing with missing data. To overcome this, I implemented error-checking mechanisms in the file-loading process. For instance, for loading all my exits, I decided to handle missing locations, I made sure to check that ‘fromLocation’ and ‘toLocation’ were successfully found, if either location was ‘null’, meaning the location name in the file did not match any location, a warning message is printed. And if both locations are found the code adds exit to ‘exitData’ a direction from ‘fromLocation’ that leads to ‘toLocation’

Another challenge I faced was creating a method to keep track of player's progress, visited locations, and inventory management. I was stuck between 2 methods for handling visited locations, initially, I decided to use the Boolean method to check if the player had been in a location, but was a bit complicated, so I went with array lists. So the first time the user enters a location, it gets added to an array list and gives a detailed description, and the second time he goes there it checks the ArrayList to provide a short description. As for inventory management, I used ArrayList again, allowing items to be added and removed as the player interacts with the game world. My 'Control' class manages all the user commands and re-routes them to do tasks, which was quite stressful because apart from all commands, I also included special cases, scenarios, and game rules. All my character interaction dialogues are also present here. I created commands for 'go',' take',' inventory', and 'exit', I also decided to create a simple command for 'help', which gives a text message describing all controls in the game to help the user, this can be commanded any time of the game. I tried creating the game as friendly as possible, for instance, when a user enters a wrong/ invalid command user gets notified about the invalid command and suggests using the 'help' command. Also, during riddles, when the user gets it wrong the first time, the user gets a friendly hint to help.

Testing was a critical part of the game. Initially, I simulated player interactions by feeding various commands into the game and observing the output, which helped identify some edge cases where the game could behave unexpectedly, such as trying to move to a location with no exit. I did a full playthrough of the game with all the possible things you could do, including locations, items, characters, and actions. During testing I encountered a number of issues like some commands behaving unexpectedly, to overcome that I implemented formatting the user command in the 'Control' class by converting it into lowercase and trimming whitespace, which ensured commands were recognized regardless of player input format. I adjusted and added a narrative flow to the game, to make users understand what exactly is happening when an interaction occurs, A number of feedbacks were added to ensure the game challenge was fair and clear.

As I created and finalized my game, I found some aspects that could use improvement in the future, and things I could have done differently. While adding characters, items, and exits I made sure locations exits, which can handle some error handling, I think I could have added more robust error handling techniques in the game, like enhancing error messages with possible solutions or corrected data would be great. Another thing, I created the character interaction likely static, I could have introduced more dynamic character interactions. I also could have introduced more puzzles and quests to make the game more challenging, I was thinking about something more complex and multistep quests that require the player to gather multiple clues from different locations to progress. One other is to include a 'use' command to use items in inventory, at the moment my game pretty much uses an item as long as the item is in inventory and gives feedback, introducing a 'use' command can make the game more interesting and vibrant. Even though, I gave narratives and small storylines, implementing a detailed narrative with backstory, hidden legends or even multiple ends would be amazing. I also could have enhanced the 'help' command with progression tips. Finally, I find once the game is completed, the replay might feel repetitive, adding elements that enhance replayability like different starting

locations, random item and character placements, alternative paths, and easter eggs. This would have encouraged the user to replay the game. I also would love to introduce a graphical interface and gameplay on my current storyline, I am planning on developing it in the future.

Overall, this assignment was a great project to use all the Java techniques I have learned so far and develop something of my own. I was able to practice and solidify my knowledge of Java concepts and reuse it. However, I am very happy and proud of my creation and hope to expand and upgrade it in the future once I learn more.

Here are some links that helped me in my journey,

https://youtu.be/lHFlAYaNfdo

https://youtu.be/70qy6_gw1Hc

https://youtu.be/KhaXszafHWQ