WORLD OF ZUUL

Heman Seegolam

K21003628@kcl.ac.uk

20/11/2021

INTRODUCTION

The user is immediately placed in a foreign, dark place covered for the most part in deep fog. Upon looking around they make out the silhouette of a building on their left past the fog and so they are asked whether they would like to proceed. Later they will find out that they are trapped inside a nightmare and have to escape by trying to awake themselves.

THE HOUSE

Once the user selects to proceed, they are introduced to a house in which they are able to move between a main foyer, a living room and a kitchen. In the main foyer there are remains of stairs that have been burnt which had led to an upstairs area. Upon exploration, the user will come across many items, notably a ladder in the kitchen which they are able to use to travel upstairs. They will thereby have access to two bedrooms in which they can look through the window to make out a building across from the house. This then unlocks this building for the user to travel to. They will also find a hidden riddle in the kitchen's pot of coffee and a key in the pot of sugar. They can also find a pen in the living room which they can use to write on a notepad which they find in the main foyer. They will find a computer in the living room which when switched on presents another riddle which takes in a correct answer in return for a code which will be needed later on to escape. There is also a locket in an upstairs bedroom stored within a jewellery box which the user can open and will see a date which represents a second code. There is another key in a second upstairs bedroom to unlock a later room. Bathroom/Garden doors, etc. will be presented as locked to the user.

THE WAREHOUSE

By seeing a building through the window of the house, the user will have now unlocked access to the warehouse, which they can get to by going outside again and having a new option to go east. Inside, the user will be able to interact with various items but crucially, they will meet a new character that is unconscious on the warehouse floor, which the user will be told they vaguely recognise. The user will then be able to interact with this character (try to wake them, search them, etc.) and will find a note in their pocket. Upon reading the note, the user will then look up to see the new character to have woken up and running away.

THE WAREHOUSE BACKDOOR

If the user attempts to follow the new character to the backdoor of the warehouse, they will find out that when they attempt to go through this door, they will be transported randomly to any of the rooms mentioned. The character will continue to move randomly through the game

THE WAREHOUSE DUNGEON

Under the warehouse is the dungeon. When the user proceeds into the dungeon, they find a locked cell which can be unlocked by using a key found in an upstairs bedroom from the abandoned house. When unlocked, the user meets a prisoner who escapes and moves randomly throughout the game. In the cell, the user will have access to a trap door. The user must then collect two keys and two codes to escape and win the game.

TECHNICAL DETAILS

The user will be able to carry items to a total weight of 20 kilograms.

Character descriptions should always start with a / an followed by character name.

TASKS

Base Tasks -

• The game has several locations which the player can move between.

- Using a hash map, each room has associated string directions to room objects, which replace the ‘current room’ of a user.

• There are items in some rooms, some of which can be picked up by the player.

- Using a boolean value to store whether an item is collectable, when the user types in ‘take’ followed by an item, it is added to the user’s inventory.

• The player can carry some items with him. Every item has a weight. The player can carry items only up to a certain total weight (20).

- Using a maximum weight integer value, every time the user attempts to take an item, the remaining weight space in the inventory is checked against the weight of the item.

• The player can win. There has to be some situation that is recognised as the end of the game where the player is informed that they have won.

- When the user ‘opens’ the trapdoor, they are prompted for two codes and relevant keys are automatically taken. If the user enters the correct two codes and contains both keys, they win.

• Implement a command “back” that takes you back to the last room you’ve been in.

- Using an array list, the trail of the user is recorded by adding rooms in order of which they were entered. If a duplicate is found in the list, i.e. the user returns to a room they were at before, all rooms after the first instance of the duplication are removed. This allows the user to repeatedly go back in the correct order (starting from outside).

• Add at least four new commands (in addition to those that were present in the code you got from us).

- Open Inventory and remove items from inventory.

- Interacting with items…

    - Switch on light-switch and computer.

    - Open window (to unlock warehouse).

    - Open and write in notepad.

    - Open kitchen pots (to get key from sugar and note from coffee).

    - Open jewellery box (to find and open locket).

    - Searching the coat.

    - Open trapdoor (to assess whether user has the required two keys and codes for them to win).

    - Attempt to wake, search and talk to character from warehouse.

Challenge Tasks -

• Add characters to your game that can move around by themselves.

    - By making the player an object of the character class, they are able to imitate the user by making use of the same functions.

• Extend the parser to recognise three - five word commands.

    - Using the parser to identify the next word(s) in user input.

• Add a magic transporter room (the warehouse backdoor) – every time you enter it you are transported to a random room in your game.

    - Using the random function to randomly assign an index to a list of rooms that the user can travel to.

• Have several locations that are ‘locked’ and have to be first unlocked by the user for them to access.

- Ladder to go upstairs in abandoned house.

- Key to enter cell in warehouse dungeon.

    - By creating class variables that monitored whether a user has access to a room and modifying the access when they use the appropriate items.

Implementations of:

Cohesion & Maintainability–

In the processCommand function, each command word is marked against one function that processes the command entirely in a separate function and then further directs the user to a separate function (primarily in the relevant item, room or character class) to carry out tasks. For example, for a ‘go’ instruction, this has been separated into multiple functions, one that first determines whether the user is going to a room or an item / character. If going to a room this brings the user to the goRoom function in the character class which brings the user to a room. If going to an item or character, there is another function to make this difference, again redirecting the user to separate functions to carry out the appropriate task, based on the command. By restricting functions to no more than one essential task, it helps to pinpoint errors in the program and therefore improves maintainability.

Coupling, Responsibility-Driven Design & Maintainability –

By defining all methods strictly related to an item only in the item class, those for characters in the character class and those for rooms in the room class, along with having all associated class variables private, it prevented having a strong dependence on the game class to carry out certain functions. Instead, the user in the game class is simply redirected to the relevant class to perform a task rather than attempting to perform it all in the game class. This helped to prevent errors when modifying or extending functions in the other classes. Whilst there is still a link between these classes, they are very loose-coupled since a class is alone responsible for managing most of its data, and so are not very reliant on each other, thus reducing opportunity for error and increasing maintainability.

Walk through –

Go west / go north / go to ladder / take ladder / go back / use ladder / go up / go to pot of coffee / open pot of coffee / open note / go back / go south / go to computer / switch on computer / ‘locket’ / go north / use ladder / go up / go west / open jewellery box / open locket / go east / go north / go to window / open window / go south / go down / go east / go east / go to person / search person / go west / go west / go north / open pot of sugar / take key / go south / go east / go east / go to cell key / take cell key / go down / go to coat / search coat / take key / use cell key / go east / go to trapdoor / open trapdoor / ‘3164’ / ‘0802’ --> You win.

Known bugs:

- Whilst the go back function works the majority of the time, there are some occasions where it fails, primarily when going back outside and when going to a random room through the warehouse backdoor.