

## **First Report**

Paulina Traub

Programming through Python

24-11-2022

University of Sussex

---

Index:

- 1- Problem Statement
- 2- Class Diagram
- 3- Changes in the code example
- 4- Problems on the road

### **1-problem statement:**

In the game what is required of you is to find the master key that is hidden in one of the rooms, what you need to do is move forward without getting caught by one of the ghosts. There is a specific piece (Dolls Room) that can lead you to lose the game. If you press west, it will take you to the Cursed Cemetery. If you press east, you will reach the Red Door Room piece where you can free yourself from the ghosts that haunt you. What you can do in the game is to choose where to go, get Items and use Items (this option was not fully developed, but I've put it there anyway in the meantime.

The path is as follows, you wake up in My Room, and choose to open the door to go to the corridor, then you have 3 options, open the door to the basement, the door to the lake or stay in your room. The best option is to go to the lake because there you will find the key that will open the door to the room with the red door. Once at the lake, you have the option to go down the path to the greenhouse, go back to the corridor, go to the forest or teleport. The idea is that at this point you teleport and get to the doll piece. Where you have to choose EAST to win the game and be able to get to the red door piece and get rid of the ghosts, otherwise, if you choose SOUTH it will take you to the cursed graveyard.

**First Part Map**

North

Green House



lake



woods



My Room



corridor



basement

**Second Part Map**

Lobby



Dolls Room



Red Door Room



Cursed Cementery

South

EAST

WEST

## 2-Class diagram:

Class: AdventureWorldStarterCode.Game. <b>Game</b>
f) create_rooms() f) game f) self.current_room = self.myroom f) self.textUI = TextUI() f) self.bookshelf
M) __init__(self): create_rooms(self) play(self) print_welcome(self) show_command_words(self) Seeltems(self) keys(self) opendoor(self,) print_help(self) do_go_command(self, second_word): def main()

Class: AdventureWorldStarterCode.Room. <b>Room</b>
f) self.description = description f)self.doorposition = doorposition f) self.exits = {}
M) __init__(self): m) set_exit(self, direction, neighbour) m) get_short_description(self) m) get_long_description(self) m) get_exits(self) m) get_exit(self, direction) m) door_position(self)

Class: AdventureWorldStarterCode. <b>Items</b>
self.name self.description
M) __init__(self) m) getName(self) m) getDescription(self)

Class: AdventureWorldStarterCode. <b>bookshelf</b>
f) item_list = []
M) __init__(self) m) setup_bookshelf(self) m) get_item_list(self) get_item(self, name)
Class: AdventureWorldStarterCode. <b>TextUI</b>
M) __init__(self) m) get_command(self) m) print_to_textUI(self, text)

### 3-Changes in the Original code

- The initial program has been changed in that I have added more pieces in the game, these compartments also have different items and keys inside.
- Regarding the code, I added new commands, like BAG, OBTAIN, OPENDOOR, in order to make it more interactive.
- I made a new class called bookshelf where the items are stored, and I also created the Item class to store them.
- I added the commands with the upper Word function so that Python could read the names without any problem.

### 4-Problems on the road

At first, I felt overwhelmed because I didn't understand what the game was about, I started from the beginning reading the code, and I realised that the structure was already done and that I only had to add a couple of things to sweeten the game.

When I tried to make the story, I found it difficult to put together the context of each of the pieces and define the objective of the game.

One of the problems I had was the fact that I didn't know how to start the game, if I had to say a particular word like "start" and I typed that and I got an error.

I think that sometimes the program, because it was so sensitive, didn't recognise words in upper or lower case.