**ADVENTURE GAME**

INTRODUCTION :-

The theme of our text-based adventure game is to traverse from room to room, and search for and retrieve data from a hard drive placed in one of the rooms of the map chosen by random. The layout of the rooms is shown at the beginning and is updated according to the directions chosen. The keys ‘n’,’e’,’w’,’s’ are used to travel north, east, west and south respectively.

The objective to win the game is to find the correct room where the hard drive is, and return back to the starting point within a limited number of moves available. There is an exit option when you reject the mission or don’t want to repeat it.

To start the game, run the program and press ‘y’ to accept the mission. This project uses many basic and advanced features of python language.

SOFTWARE USED :-

The software used are: Sublime Text, IDLE 3.7 and Anaconda 3

Various modules such as pygame, random, pyttsx3, PIL were installed and imported to access their in-built functions such as mixer and random.

Sublime Text was used to find and fix any indentation error in the code.

IDLE and Anaconda were used to write python script and run it.

RESULTS :-

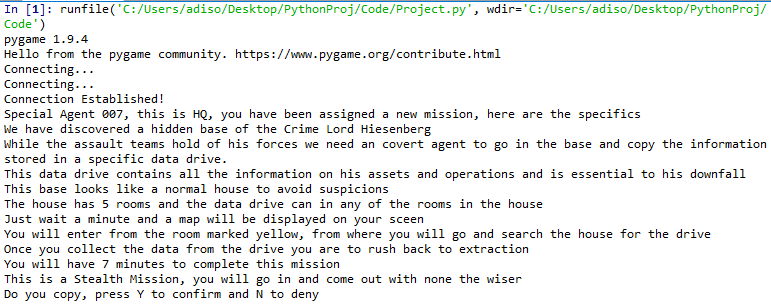
When the program is run, the player is first briefed about the mission to find the hard drive, copy it’s contents and find your way out in the least amount of steps (One step is one direction).

Them, the program waits for confirmation from the user, and exits the program if the user does not press ‘y’. If the user inputs ‘y’, the game begins and the map is displayed in the form of an image.

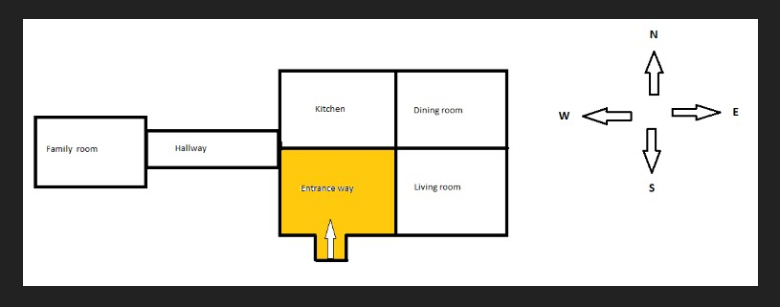
After this, the keys used for going in the four directions are: ‘n’, ’e’, ’w’, ’s’. Any other input is considered a wrong step and the number of moves is decremented. If the room the user steps into is the room where the hard drive is kept, the player is displayed the appropriate message and asked to head back to the extraction (starting) point as soon as possible before the number of moves run out. Running out of moves is also a way to end the game in failure.

The game is won by reaching the starting point before the time runs out.

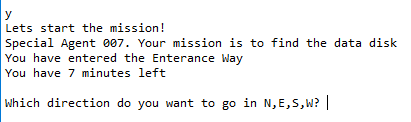
SNAPSHOTS :-



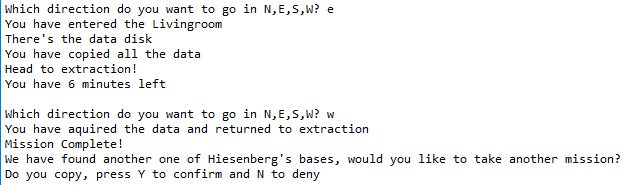
*Fig 1 : The start of the output and explanation when the program is run.*



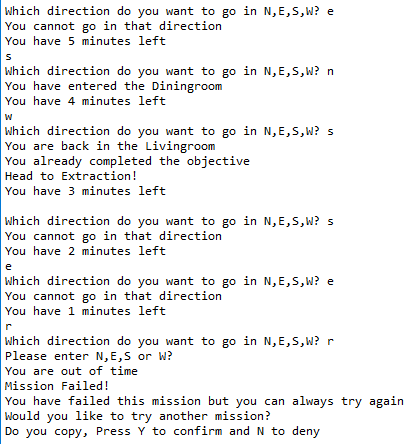
*Fig 2: The layout shown at the beginning with the yellow room being the current room*



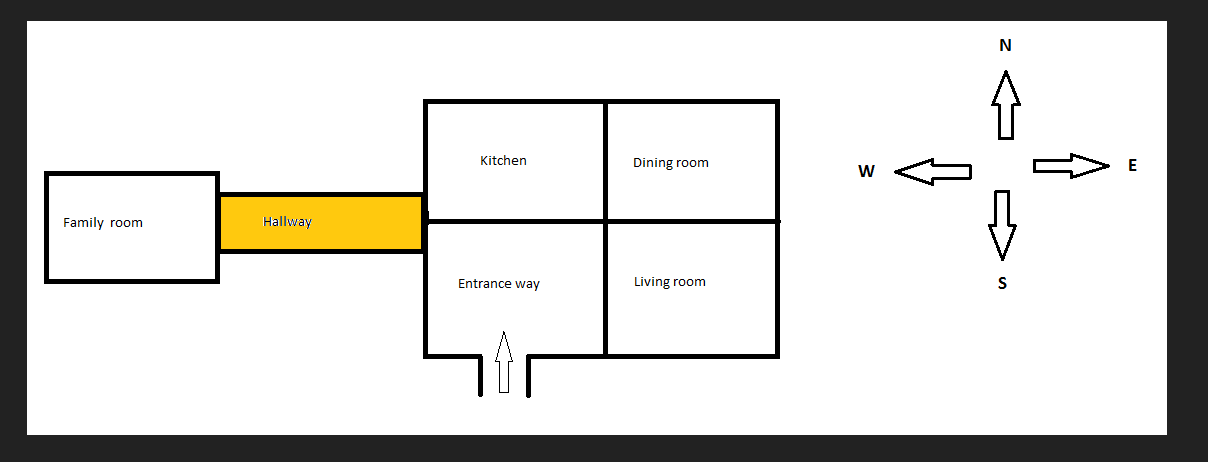
*Fig 3 : The start of the game once the mission is accepted.*



*Fig 4 : Example of winning the game.*



*Fig 5: Example of play and losing the game.*



*Fig 6 : Example of map being updated according to the current room the player is in.*

CONCLUSION :-

Python language is flexible, easy to use and has a lot of in-built modules. We used many basic, as well as advanced concepts of the Python programming language. We have used data structures unique to python such as sets, lists, tuples and dictionaries. These data sstructures are used to identify the current room, entered direction and the next room accordingly. Without sets and tuples, the code would be too messy and involve a large number of conditional statements.

We also used concepts of python like importing packages, conditional and looping statements, logical operators, etc.

Apart from these, other features of python incorporated in this project are:

1. *Pygame* : The pygame module makes it easy to add features such as sound and background music in the game. This is how sound is played when the player wins the game.

Functionalities used are: pygame.mixer.init() , pygame.mixer.Sound(“sound.wav”), sound.play()

1. *Random* : The random module is used to choose a random room from a list of all the rooms except the starting room. Functions used: random.choice(rooms)
2. *Pyttsx3* : The pyttsx3 module is used to convert the text to speech. It is used in this project everywhere to speak out instructions about the mission. Functions used: pyttsx3.init(), engine.say(“Insert text here”), engine.runAndWait().
3. *Time* : The time module
4. *Image from PIL* : This module is used to open and show an image file in the program. Functions used: img=Image.open(“image\_name.png”), img.show()

FUTURE SCOPE :-

This game is a very basic version of a Role Playing Game (RPG). It can be heavily improved in both complexity and graphics to make a commercial applicable game. Similar code can be used to make games like Ludo. This game is a very basic building block and many such blocks can be interlinked to make a market appropriate RPG with many variables and scenarios with market level graphics.

REFERENCES :-

* “Core Python Programming” by R. Nageswara Rao
* James Payne, ”Beginning Python: Using Python 2.6 and Python 3.1”, Wrox Publication.
* <https://knightlab.northwestern.edu/2014/06/05/five-mini-programming-projects-for-the-python-beginner/>
* <https://www.python.org/>
* [https://www.programiz.com/python-programming/dictionary#methods](https://www.programiz.com/python-programming/dictionary" \l "methods)s

ALGORITHM :-

1. Required Modules are imported and number of moves is set to 8.
2. Various functions like the Main, Map, Get Choice, Msg, Restart are defined.
3. The objective of the game is explained.
4. User input is asked to start the game where;
5. If the input is N the program ends
6. If the input is Y go to main function.
7. With the help of random module, the object is placed in one of the available rooms.
8. A map is opened for the starting point and the user’s choice is asked for the direction in which the user wants to head in.
9. After receiving the direction the program refers the input to the get\_choice function and the user is transferred to the appropriate room and the number of turns left and the room’s message is displayed. Also the map of the respective is room is displayed.
10. The user is again asked for directions and if the room travelled to is the room with the object the user is notified and asked to return to the starting room before the turn ends.
11. If the user returns to the starting room successfully before the turns end the user wins the program asks for a replay in which if the input is Y the turns is set to 8 and the main function is called again, if the input is N the program ends.
12. If the user is unable to find the object or is unable to reach the starting room before the turns end the program notifies the user of their loss and asks the user for a replay for which the restart function is called.
13. The restart function takes user input and if the input is Y the turns is set to 8 and the main function is called again, if the input is N the program ends.