

“Maze Game”

Anthony Wayman, Harrison Gilbert

CS-150

12/12/2023

Abstract

We will be constructing a game that consists of a player character, “P”, a world map that is randomly generated based on the player's movements, and an exit the player must find before the timer runs out. We are using the Python programming language to code this game in a Jupyter notebook. For this project we will be utilizing everything we have learned in this class to produce a complexly designed yet simple game.

Introduction

Our game, “Maze Game”, will display a start screen menu where the user is prompted to start a new game or exit the game (upon winning/losing). The player is trapped in a randomly generated maze of rooms. A timer begins counting down and the player must find the exit before it is too late. When the user starts a new game the clock function begins running and the user is tasked with using the w, a, s, and d keys to navigate the map and find the Exit, “E”. Each time the player moves to the edge of the map, a new map is generated for the player to navigate. Within the game there is a very rare resting area the player can find, “O” to reset the clock. The entire collection of maps is called the world map and exists so the player can navigate the pre-existing maze.

Methodology

For this project we used functions, for loops, if statements, formatted strings, and lists to create “Maze Game”. At the beginning of our code we imported time to implement the clock feature. Then, our code consists of a start menu with the two user inputs to respond to “start game” and “exit”. Just under the start game menu we have identified all the possible room configurations, these rooms are created in lists with the walls being made of (#) symbols. The proceeding functions control character movement, up, down, left, and right. The map is updated upon the expand functions which tell the code to pick an appropriate room based on our players movements. Near the end we have a while loop that allows the user to input the (w, a, s, d) keys, as well as what to do when the clock runs out.

Design and Implementation

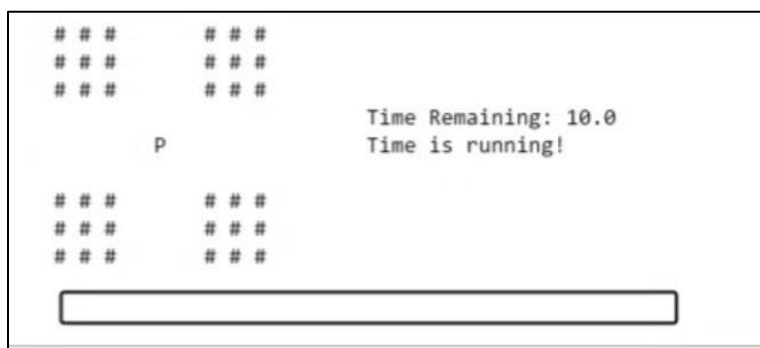
The beginning of our game code we have defined a function, “printScreenAndPlayer” to update the display for the user. This also controls the clock being displayed and the message, “Time is running!”. Each available section of the maze is defined in a list at the top of the code. The expand functions work in conjunction with the world map to coherently paste the walls of the maze together. This is to make sure the player is able to walk from one room to another without interference. The next few functions define the player movement. These functions ensure the player cannot walk through walls. It also changes the player index based on movement and changes it based on the ex

Challenges and Solutions

We encountered several challenges during the making of the game. The most prominent of which included a complete re-haul of the game. We originally had the rooms set up as a large array. We found trouble navigating them, and got several errors. We eventually had more luck setting everything in lists. Another problem we encountered was with the “O” character. It was supposed to reset the clock. We eventually figured out we had an error in the way we placed our code and re-wrote it so upon hitting the “O” the clock variable “start” was reset.

Discussion

In the end, we were left with a successful project that does everything we set out to achieve. If we were to make changes, we would like to possibly add enemies that also spawn in the map and chase our character. We would also like to implement more room designs or features in certain rooms to make the game better.



Conclusion

In conclusion, the “To-Do” List is a daily task manager, built to create lists in which users can input specific tasks along with the time to complete those tasks. Users can edit, or remove these

tasks as well as mark them complete. We used a variety of coding principles in order to create this project such as functions, if statements, and lists. This list will ultimately help manage daily tasks that otherwise would go unfinished.

References

For this project, we used our knowledge gained from in-class notebooks.