Vasudha Srinivasan

15-112 TP1 Project Proposal

Project Description:

My term project will be a game titled “Escape!”, where the player will travel through multiple randomly generated 3-D, 1st person view mazes to reach their final destination.

Competitive Analysis:

There were a few similar projects I have seen online. One example of this is a term project from Spring 2019 that is called “Maze”. One similarity between this project and my own is that in “Maze”, the designer created an almost 3D version of their maze through which the player could move, something I plan to do as well. I also plan to execute this using Tkinter, just like in “Maze”. However, the biggest difference between “Maze” and my game is that while the designer of maze allowed the user to place down blocks that would later turn into the maze that the player could travel through, I will randomly generate the mazes through use of multiple algorithms and incorporate a storyline with the main motive to solve the maze to escape.

Another project similar to my plan is a term project from Fall 2017 called “Maze Crawler”. This project is similar to my idea in that it uses multiple maze generating and solving algorithms to form its mazes. However, it is different in that it acts as a multiplayer game and is only in 2D, while I plan to make a 3D version of my maze and only display the 2D version if the uses needs help. I also plan to use Tkinter, while “Maze Crawler” uses Pygame.

Lastly, “TiltMazeAR” is another project I found from Spring 2018. This project is similar to my plan in that randomly generates mazes and incorporates graphics that are 3D like with Tkinter. However, it is different in that it uses OpenCV to display and move the maze using a board, while my plan is to make the player look like they are in the maze and travelling through it solely with keyboard commands.

Structural Plan:

My project will consist of randomly generating mazes using multiple different maze generation algorithms and drawing them in 3D for the player to walk through as if they were doing it in person. To execute this project, I plan to create a class MyModalApp under which my game will be subclassed into different modes (start screen, levels, etc.). I will be using multiple algorithms to randomly generate different mazes, and each algorithm will be under its own class. These classes will be called in other classes in order to draw the 2D maze and then eventually transfer it to a 3D one.

Algorithmic Plan:

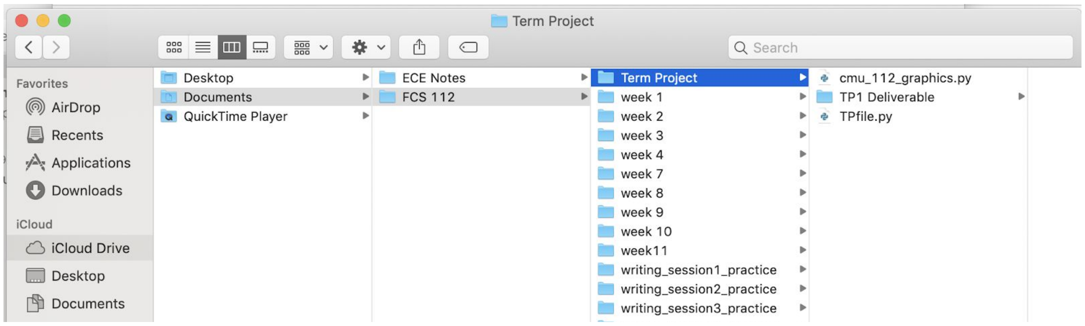
The main algorithms for my project will be in the generation of random mazes. I will be using multiple algorithms to do this, increasing complexity and making sure that the mazes themselves are not predictable and vary more based on level of difficulty. All of these methods will end up involving recursion to generate the maze. One algorithm I will be using is based off of the concept of depth-first search. For this method, I will create a grid, and starting from the top left corner, will randomly progress in random generations adding cells to a path until the path reaches a dead end, in which it will use backtracking to find the next open path while still storing the dead end to ultimately create a maze. Another algorithm for the maze will be based off of Kruskal’s algorithm. To do this, I will assign a tuple to each cell, randomly pick a pair of cells next to each other, and “connect” them by adding the cell’s tuple to the other. This will go on, connecting pairs as long as they have not been connected beforehand until pairs cannot be connected anymore. This will create a single branching path with dead ends as well as a solution to be used as a maze. I plan to also use the concept of Prim's algorithm, which consists of picking a random cell, picking a random neighbor, and connecting them to form a new "cell" and repeat the process. Lastly, if I have time, I plan to incorporate recursive division as another way to generate a maze. After implementing all these algorithms, I plan to use backtracking to simultaneously store the solution for each generated maze so that it can be provided to the user if needed.

Timeline Plan:

I started working on my project by first focusing on the main algorithmic part, the random generation of the mazes. My goal is to finish coding up the 2D versions of my randomly generated mazes by this Saturday, November 20th, with the code for the different types of maze generating algorithms hopefully done by the end of Thursday. I then plan to start working on the 3D graph conversion. I hope to figure this set up by TP2 so that I am ready for MVP. After this, I plan to add some extra features such as generating different levels based on previous rounds, showing the player the solution and adding mini games and music by TP3.

Version Control Plan:

I plan to back up my code in multiple sources, including the cloud as well as google drive to ensure that I do not lose my progress if something happens to my computer. I have set up my computer so that any file I open and edit updates in a file on the cloud on a regular basis. As shown in the picture below, in addition to my TP files being stored in my computer, it is also in the iCloud Drive.



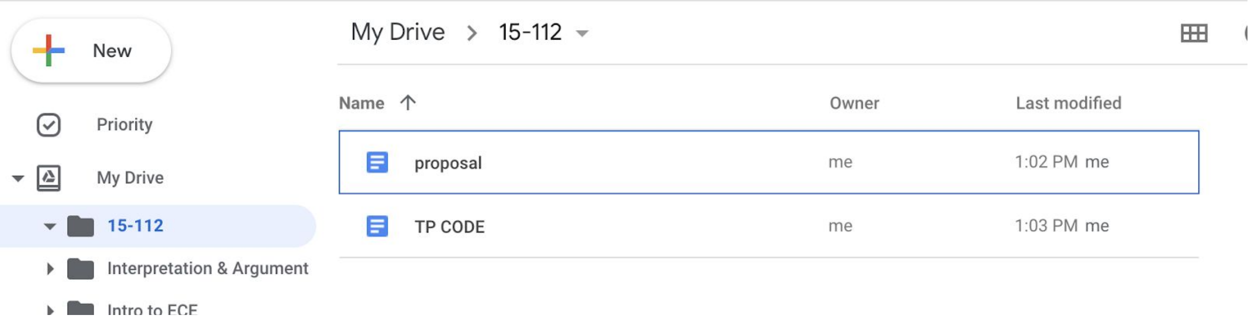
Desktop
Documents
QuickTime Player

ECE Notes
FCS 112

Term Project
week 1
week 2
week 3
week 4
week 7
week 8
week 9
week 10
week11
writing_session1_practice
writing_session2_practice
writing_session3_practice

cmu_112_graphics.py
TP1 Deliverable
TPfile.py

In addition to storing my files on iCloud, I will store them in Google Drive on a regular basis. To avoid taking up too much storage on my drive, I will be putting temporary updates of my code in a document (only uploaded files take up space in Google Drive), and then finally uploading the whole python file at the end of the day so that my code is still saved even if I don't have the python file.



Name ↑	Owner	Last modified
proposal	me	1:02 PM me
TP CODE	me	1:03 PM me

Module List:

So far into my project, I am not planning to use any external modules, hardwares, or technologies.

TP2 Update-

Timeline Plan:

Unlike my original plan, I was only able to create the 2D version of my maze by the TP2 deadline. However, I still plan to execute my original plan design of having my ultimate product to be a 3D maze that the player can go through by the final deadline.

TP3 Update-

Project Description:

My term project is still called "Escape!" and the player still travels through randomly generated mazes to reach the end. However, instead of making all modes of my game in 3D, the basic levels are in in 2D with an option to go through a 3D level if the player chooses the mode.

Structural Plan:

As previously mentioned, instead of making all the mazes in 3D and then only switching to 2D when the player is stuck, the player has different options for modes, with the basic levels in 2D and then extra modes for 3D and randomly selecting a level. Also, the mazes used for the 3D mode are significantly easier than in 2D as solving the maze in 3D seemed to be much more difficult for players that they did not always want to go through completely.

Algorithmic Plan:

I based my random maze generation based off of multiple algorithms such as Prim's algorithm and Kruskal's algorithm, but did not completely use recursion as I said I would (I instead used loops at points where it seemed easier and more efficient to do so).