15-112 Term Project   Title: MAZE   Karina Shethia (Kshethia)

Project Description

A maze game. You can choose from 3 modes (each based on a philosophical theory) and 3 levels of difficulty. For all modes:

- The arrow keys are used to move the player (a dot).
- Randomly placed dot marks the goal. When reached, score increases by 1 and new maze generated

Mode → *Difficulty* **↓**	**Absolutism** (Timer, 3 enemies)	**Eternalism** (No time limit)	**Idealism** (Timer, 3 enemies, as time runs out player moves slower)
Easy	30 seconds to complete each maze	No enemies	30 seconds to complete each maze
Medium	20 seconds to complete each maze	One enemy	20 seconds to complete each maze
Hard	10 seconds to complete each maze	3 enemies	10 seconds to complete each maze

Structural Plan

Final project will be organized in one file in separate functions, segmented using comments and app.mode:

	Section separated using block comment	**App mode**
1	Main/Overall Function • appStarted, appReset, getCellBounds	
2	Choosing the Mode Functions • keyPressed, parameters for the different modes, redrawAll	chooseMode
3	Choosing the Difficulty Level Functions • keyPressed , parameters for the different difficulties, redrawAll	chooseDifficulty
4	Game Functions • timerFired / doStep, checkForWin, keyPressed • redrawAll (maze, player, goal, enemies, score, timer)	game
5	Maze Generation (using Prim's algorithm) & Maze Class	game
6	Player & Goal Classes (including move and valid move methods in player class)	game
8	Enemy Functions (including move and valid move methods)	game
9	Happiness Level Functions (display and calculations, for Idealism mode)	game
10	Game Over Functions • redrawAll (displays 'Game Over' & final score), keyPressed (e.g., restart)	gameOver
11	Running the App (i.e. runApp)	

Algorithmic Plan

Prim's algorithm : Generating the maze

1. Cell coordinates stored in a 1D list. Each coordinate represented as a tuple (e.g., top left is (0,0))
2. Using Prim's algorithm & the 1D list to create a dictionary mapping each cell to the adjacent cells it connects to (i.e., doesn't share a wall with, therefore the player can move through)
3. Go through each key-value pair in the dictionary. Creates a new instance in the Maze class
4. In redrawAll, go through each instance and draw lines according to the Boolean values of each cell's boundaries (e.g., if rightBound == True, draws a line on the right side of the cell)

Dijkstra's algorithm : Adding enemies that track and move towards the player

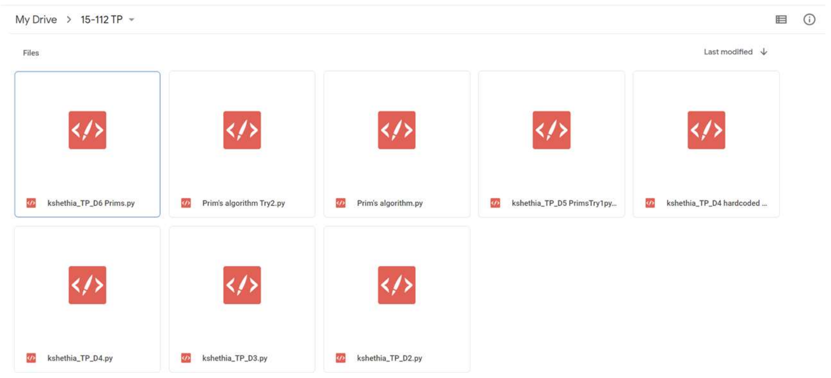
Still figuring out how to implement this one, but here's what I am thinking so far:

1. Cell coordinates stored in a 1D list again.

2. Keys in dictionary will again be each tuple coordinate
3. Instead of the values being the tuple coordinates, will probably contain the coordinate that it connects to form the shortest path
4. To find overall path enemy would take, follow the keys -> value of that key becomes the new key etc. till hit end (i.e., same coordinate tuple as player)

Version Control Plan

- Saving new versions on my laptop whenever I implement a big change / tackle a segment of my plan
- Uploading each version onto Google Drive



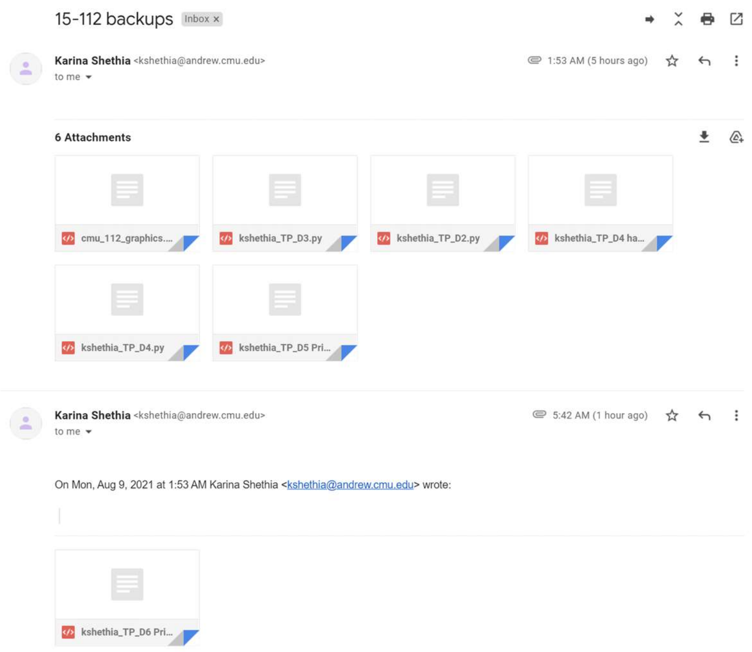
My Drive > 15-112 TP

Files

Last modified

kshethia_TP_D6 Prims.py
Prim's algorithm Try2.py
Prim's algorithm.py
kshethia_TP_D5 PrimsTry1py...
kshethia_TP_D4 hardcoded ...
kshethia_TP_D4.py
kshethia_TP_D3.py
kshethia_TP_D2.py

- Emailing each version to myself via Andrew email



15-112 backups Inbox ×

Karina Shethia <kshethia@andrew.cmu.edu>
to me
1:53 AM (5 hours ago)

6 Attachments

cmu_112_graphics....
kshethia_TP_D3.py
kshethia_TP_D2.py
kshethia_TP_D4 ha...
kshethia_TP_D4.py
kshethia_TP_D5 Pri...

Karina Shethia <kshethia@andrew.cmu.edu>
to me
5:42 AM (1 hour ago)

On Mon, Aug 9, 2021 at 1:53 AM Karina Shethia <kshethia@andrew.cmu.edu> wrote:

kshethia_TP_D6 Pri...

TP2 Update

Removed	Added
• In 'Idealism' mode, player no longer moves slower as time decreases	• Instead, as timer decreases, player's happiness level decreases. • As happiness decreases, the no. of enemies multiplies sporadically. • There are hearts throughout the maze that the player can pick up. Collecting a heart increases happiness and also gets rid of an enemy .
• Dijkstra's not being used for enemy movement	• Instead, I have implemented enemies that move randomly throughout the grid (the game is over if the player collides with them).
	• Separate high scores are stored for each mode and difficulty level (i.e., 9 different high scores total). They are saved using file IO (in the file highScores.txt) and the current high score for the current mode and difficulty level is displayed on the Game Over screen
	• There will be a collision animation when player collides with enemy and the game ends. This animation will be dynamically sized according to how large the player piece and enemy is

TP3 Update

On the Game Over screen, instead of using keyPressed to restart / change mode etc., these choices appear as buttons (with a visual hover effect using keyMoved).