

# The Maze Game

CS108: Software Systems Lab Project

Indian Institute of Technology, Bombay

23B1016, Harshil Solanki

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Problem Statement . . . . .	2
<b>2</b>	<b>Modules Used</b>	<b>2</b>
<b>3</b>	<b>Directory Structure</b>	<b>2</b>
<b>4</b>	<b>Playing Instructions</b>	<b>3</b>
4.1	GUI Instructions . . . . .	3
4.2	Extra Instructions . . . . .	4
<b>5</b>	<b>Features</b>	<b>4</b>
5.1	Menu Screen . . . . .	4
5.2	Game Screen . . . . .	4
5.3	End Screen . . . . .	5
<b>6</b>	<b>Project Journey</b>	<b>5</b>
6.1	The Book . . . . .	5
6.2	The Maze matrix . . . . .	5
6.3	The Map . . . . .	5
6.4	The Camera and The Pegin . . . . .	5
<b>7</b>	<b>My Algorithm</b>	<b>6</b>
<b>8</b>	<b>Learning Outcomes</b>	<b>7</b>
<b>9</b>	<b>Maze Generation Algorithms Referred</b>	<b>8</b>
<b>10</b>	<b>Git Log</b>	<b>11</b>

# 1 Introduction

I'm Harshil Solanki, roll number 23B1016, and here's my project report for 'The Maze Game' Project under TA Guramrit Singh for course CS108: Software Systems Lab.

## 1.1 Problem Statement

The aim of this project is to design and implement a variant of a 2D maze game using Pygame, a popular Python library for game development. The game will consist of a single player-controlled character navigating through a maze with a restrictive 2D top view from the starting point to the end point while avoiding obstacles and traps.

# 2 Modules Used

- pygame: The main module in creating this game, it provided all methods required to set up the display and the objects and their attributes which we can use to modify the object as per our choice
- numpy: This module gives base to this game by allowing it to use an ndarray object that store all the maze information in matrix form
- math: Used in settings module, to compute the score as an exponential function linearly scaled to give us output on given input, that is the distance between the player and the end point
- os: os.path function used in settings module, to access directory at a lower level than the directory given file is in
- sys: used to exit the game whenever required
- random: used to select a random element out of a given sequence
- datetime: used to save the score with a timestamp of when the game ended

# 3 Directory Structure

---

```

notes.md # Store TODOs and progress and literally anything I wanted to note down for time being
game.py # Running this file starts the game! Contains Initializations and Steps of what to do next based
        on previous output
path.txt # File containing the Directions to be taken Sequentially to Reach to the end and complete the
        level
last_maze.txt # File to store the matrix of the maze that was created in the last game
|
\-\modules/
    |--__init__.py # Formality
    |--player.py # Class
    |--settings.py # Class
    |--button.py # Class
    |--camera.py # Class
    |--sprites.py # Class
    |--timer.py # Class
    |--game_functions.py # Functions
    |--menu.py # Functions
    |--end_screen.py # Functions
    |--game_screen.py # Functions
    |
    |--maze_logic/
        |---maze.py # Class
        |---builder.py # Functions
        |---hunt_and_kill.py # Function
        |---random_walk.py # Function
    |

```

```

\-\images/
  |--player_{left/right}.png # Peginion as our Player :)
  |--ducky{left/right}.png # Evil duckies in our blind pursuit
  |--blocks[i].jpeg # Flexibility to choose
  |--sky.jpg # Sky is not the limit
  |--nest.png # The Nest is the Goal
  |--heart.png # Image used to display lives
  |--mute.png # Image used to covey the mute state of game
  |--sound.png # Image used to covey the unmute state of game
  |--WASTED.png # Well, Just Nostalgia. Die in the game deliberately and see yourself!
|
\-\data/
  |--saved_data.csv # File to store data (scores and time taken) of previous games
|
\-\audio/
  |--caproni.mp3 # The Background music played during the game
  |--counter-strike-jingle-cs-radio-ok-lets-go.mp3 # Another Nostalgia to breed
  |--ducky-toy-sound.mp3 # Our peginion's 'thud' sound
  |--super-mario-beedoo_F3cwLoe.mp3 # Nostalgic Game Over
  |--gta-v-death-sound-effect-102.mp3 # Too much of Nostalgia ;)
  |--happy-happy-happy-song.mp3 # Winning = Happyness
  |--punch-gaming-sound-effect-hd_Rz1G1GE.mp3 # Losing = :
|
\-\the_latex_project/
  |--report.tex # Parent of the pdf you're reading
  |--references.bib # All the references I used, noted

```

---

## 4 Playing Instructions

Before proceeding to the run this game, you must be aware of the miserable situation of our *dear peginion* who has been caught into a MAZE laid by the *Evil magician* who conspires to take the home of this little peginion away from it, essentially levitating it in open air in midst of a vast calm sea (See the paradox, calm sea and evil villain, duh). The Evil has constructed a maze around nest *articulating in front of the world (you mean the world to him ;)) his immense intelligence* and **challenging you in front of the world (yourself, hehe)** to break his code, solve the maze and reach to your nest. By watching the peginion rest in peace in it's nest, the evil will *burn to ashes* and so burns away the maze with it (peginion and nest are fire-resistant by nature) and you prove your intelligence against the world (now it means something).

To your surprise, he's added Evil Duckies to the Maze on hitting whom the Ducky dies and your peginion loses a third of it's total health and that implies that with zero health the peginion is dead and the game is over. The Duckies are blind in daylight and they move around randomly where they can. If you pass a ducky fast enough, it might save one life of yours!

Run the file `game.py` with `python3` and help the peginion (**by controlling it with up, down, right and left arrows**) reach it's nest!

Beware, you must be considerate of your own intelligence, choose the level wisely. Otherwise, you'll have to face depression as you'll be graded relatively to the players who've played if before! But if you're my type, and grades don't matter much, enjoy the peginion, the flight and the music!

### 4.1 GUI Instructions

- The Menu Screen gives you an option to mute the Music and Sound Effects of the game with a button on the topleft corner of the screen.
- The Game Screen shows you your Score and the Remaining on topleft corner of the screen and the lives remaining on the topright corner of the screen.
- It's a fast game so tighten your seat-belts before starting!

## 4.2 Extra Instructions

- In case the game is stuck and not loading (a random walk error maybe due to overconsumption of memory) interrupt the terminal and re-run the game.
- There's also a map version of the game, if you're curious you can see the whole maze in the screen by tweaking the value of `self.mode` in `modules.settings.py` as "map". You might want to disable the enemies using the `settings.py` module, set `self.enable_enemies` to False.
- In case you want pegin to move at a faster speed, set `self.move_fast` to True in `settings.py`.

## 5 Features

Not much of a feature, but you can exit the game anytime you want by pressing ESC :)

### Score

The Score is calculated at a point of time in the maze as follows:

$$score = 100 \left( \frac{w_1 \left( e^{\frac{-2d}{D}} - e^2 \right) + w_2 \left( \frac{t}{T} \right)}{w_1 + w_2} \right) \quad (1)$$

where

$d$  = distance between the player and the end point,

$D$  = maximum distance between the player and the end point possible in given maze,

$t$  = time remaining when the game was finished/over,

$T$  = total timeout allowed in the game,

$w_1$  = weight of distance parameter (here 70),

$w_2$  = weight of time parameter (here 30)

### 5.1 Menu Screen

- The game has three levels to select from
- Top 3 scores from all of game history are showed in the menu along with the time that game was played
- There's a button to mute/unmute all of game's music and sound effects

### 5.2 Game Screen

- A reverse timer is show on the top of screen showing how much time is left with you to complete the game
- Current score is shown below the timer
- The number of lives remaining is shown on the top right corner of the screen
- There are evil duckies chasing you blindly. A hit kills the ducky and you lose one life
- Background Music is being played to motivate the player to complete the game
- Collision detection is applied producing ducky-toy squash sound effect on hitting a wall (we have a pegin obsessed with ducks)
- Fading music when Game ends

### 5.3 End Screen

- Interesting music is being played in each outcome of game
- Option of returning to Menu is given

## 6 Project Journey

A brief overview of my journey could be given by my commits and code frequency on GitHub (Fig.(1a) and Fig.(1b)).

Figure 1



(a) Commit history in git

(b) Code frequency over history in git

### 6.1 The Book

I started my project by solely referring to the book *Python Crash Course* [9] provided as a reference by the TA. It gave the fundamental basis to my game and showed me the power of Classes in building big projects. I created the Settings class and Player class.

Then I implemented the menu in the game, learning first time about pygame Rects and Surfaces and how to blit them as Interactive buttons [4] and [6]. Created the Buttons class. I also stared write notes in file `README.md`[1].

### 6.2 The Maze matrix

After making the menu I started to Research over the different Maze generation algorithms available. I came across [this answer](#) on StackExchange [13] and have added the Algorithms I studied in Maze Generation Algorithms section (Section 9). I wasted a few hours figuring out how to start the walk but my code didn't work. Next day I started with a sample code from repl.it [12] and based on it built my random walk fine-tuning it and adding additional parameters such as the range of steps the walk should consist of. Here are the Screenshots of my working out the random walk, Fig. 2a and Fig. 2b

Getting Inspiration from all these Algorithms and understanding how randomness is used in each algorithm, I created my own Algorithm here [7] Then I made the game dynamic by allowing the player to move and deploying the camera (while keeping the complete map layout as it is, so it can also be accessed by tweaking the code in settings)

### 6.3 The Map

Before the Camera class was created I was blitting object directly to the screen as a map, here are screenshots of that, Fig. 3a and Fig. 3b

Final map view looked like this, Fig. 4a, Fig. 4b and Fig. 4c

### 6.4 The Camera and The Pexion

For the Camera to function efficiently [18], I needed the “*things*”, i.e. the player and wall and path Rects to be Sprites (children of class `pygame.sprite.Sprite`) hence I redefined the classes a bit and learnt more about Sprites

Figure 2

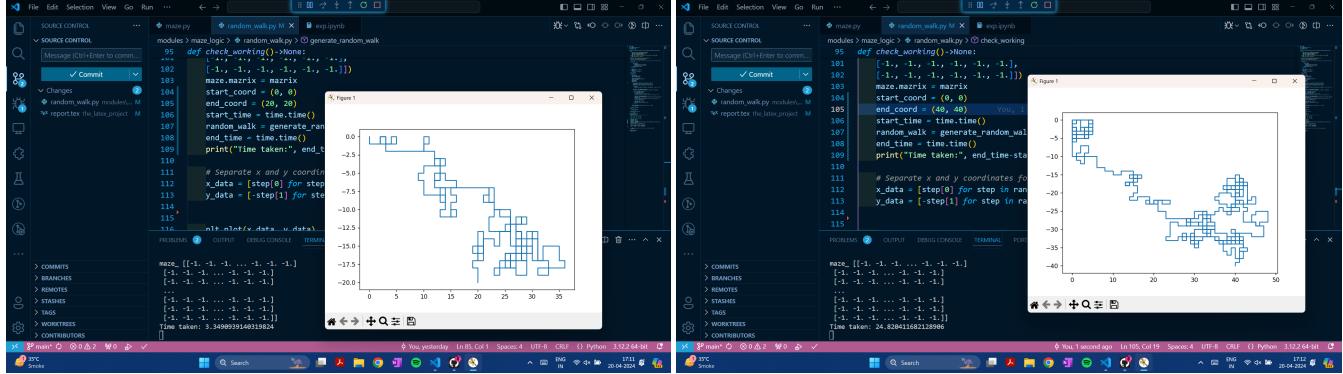
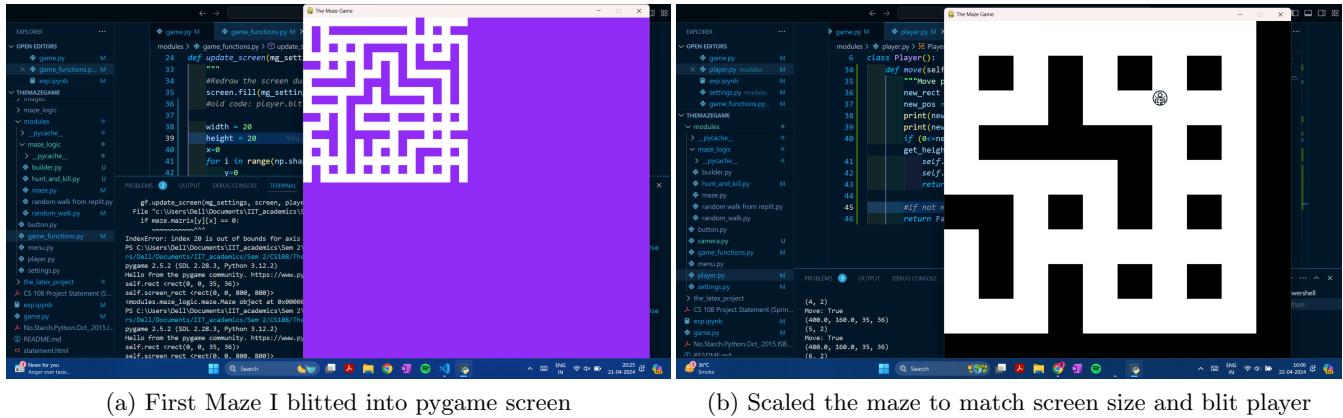


Figure 3



and Groups in pygame [5]. I also used groups to create animation effect in End-Screen [11]. After implementing the camera, the game looked like, Fig. 5a and end-screen like Fig. 5b.

Then I improved the wall sprites to take on random block images as Surfaces. Then, I added comments and descriptions to functions and methods and organise references and report. Finally, I created a child class of Player, Enemy, which spawns randomly moving Evil Ducks on screen and you lose a life if you hit one. Here are the screenshots of the final Game, Figures 6a, 6b, 6c and 6d

## 7 My Algorithm

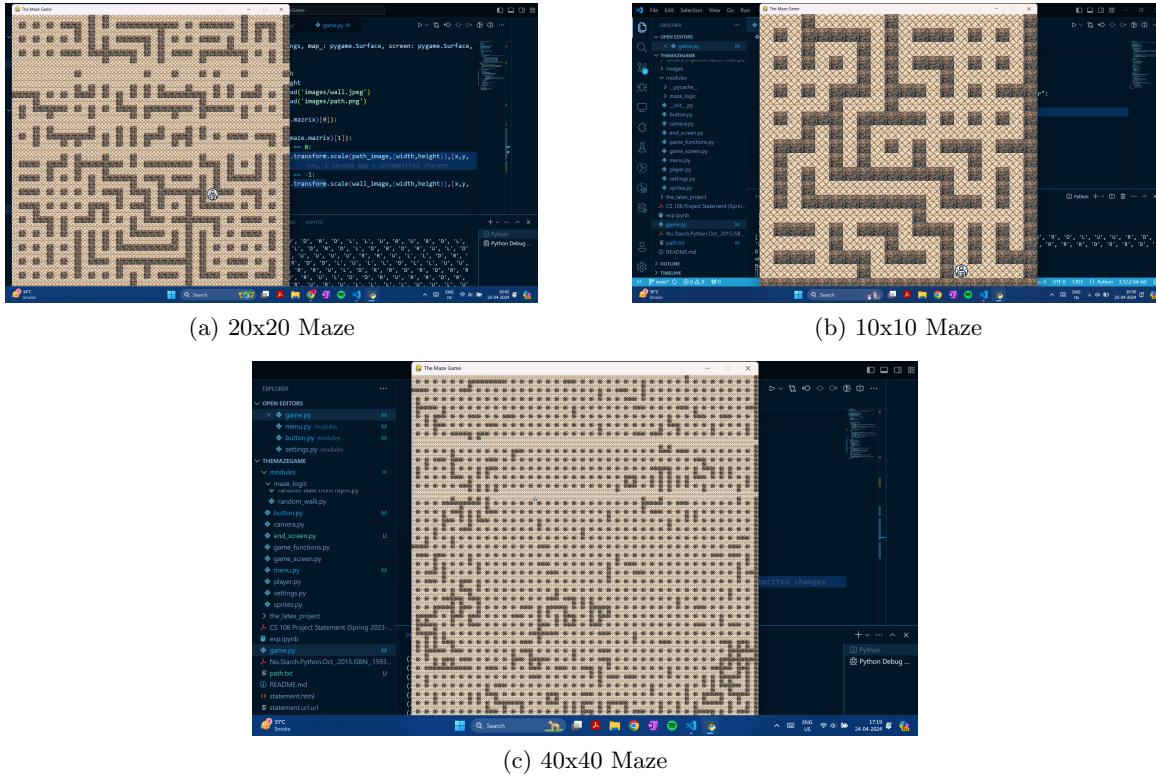
Taking inspiration from the algorithms in 9, I implemented my algorithm in this way:

1. Make a random path from start-point to the end-point with a constraint on how much steps the path should consist of.
2. Choose two arbitrary points in this path and block the initial path we made between them and generate another random path.
3. Use Hunt and Kill algorithm 9 to verify no cell is unvisited. Fine tune it to ensure that other paths are not broken down to points.

### Differentiator Between Game Levels

- First level has  $9 \times 9$  Maze with 10s Timeout and no breaking up of initially made solution path. Start Point is Top Left corner and End Point is Bottom Right corner. It has 5 enemies.

Figure 4



- Second level has  $19 \times 19$  Maze with 20s Timeout and we modify the initially generated solution path by breaking it at random place and generating another random path to increase its complexity. Start Point is Top Left corner and End Point is any of other three corners. It has 10 enemies.
- Third level has  $39 \times 39$  Maze with 30s Timeout and we modify the initially generated solution path by breaking it at random place and generating another random path to increase its complexity. Start Point is Top Left corner and End Point is any random point not in the first quarter of maze. It has 15 enemies.

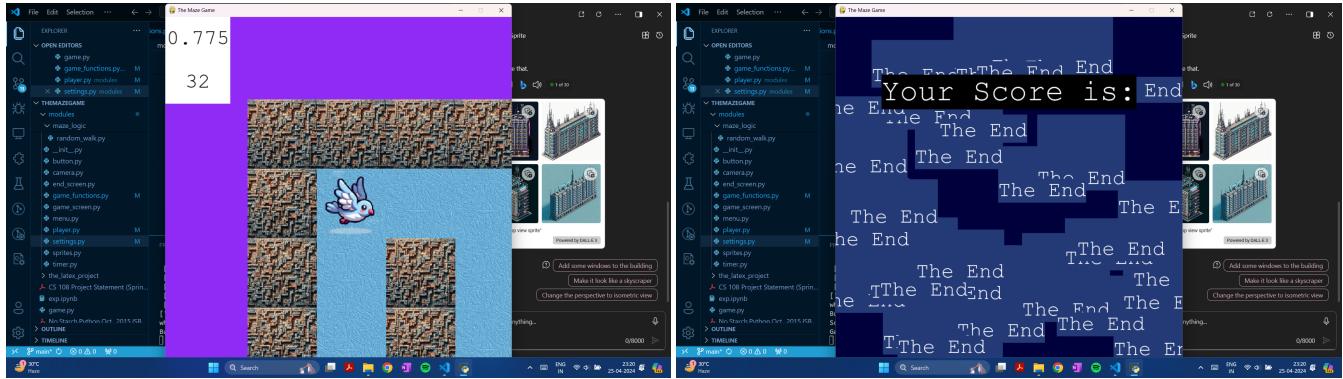
## 8 Learning Outcomes

Of all I've learnt, the foremost should be that starting everything from scratch requires so much of patience and hardwork, also it consumes a lot of your time and you may not be able to implement all the features that you wanted and that you could in given time.

However, building things from scratch cleared my concepts about Classes, Child Classes, how they're used to write code efficiently and perform huge operations easily. This project introduced me to the Sprites class of pygame, which is its inbuilt class which allows users to automate tasks without writing explicit codes for every element in game. I learnt how innovating and implementing an algorithm can be so tough that it might exhaust you of all due to some side cases you missed and that single mistake messes up all of your game. And, anyways, you eventually end up in having some bugs in your game which you couldn't find and fix out. In my case, a freeze error where maybe the memory is all used up and some coordinates are printed in the terminal unexpectedly. The only option is to KeyboardInterrupt and run a new instance of game.

In the end, it has been great fun to work for this project under deadline pressure and hectic Endsem schedule.

Figure 5



(a) Starting Camera screen where pigeon is deployed

(b) Animated End-Screen, incomplete

## 9 Maze Generation Algorithms Referred

### Kruskal's Algorithm

[\[Source\]](#)

Kruskal's algorithm is a method for producing a minimal spanning tree from a weighted graph. It works something like this:

1. Throw all of the edges in the graph into a big burlap sack. (Or, you know, a set or something.)
2. Pull out the edge with the lowest weight. If the edge connects two disjoint trees, join the trees. Otherwise, throw that edge away.
3. Repeat until there are no more edges left.

The *Randomized Kruskal's algorithm* just changes the second step, so that instead of pulling out the edge with the lowest weight, you remove an edge from the bag at random. Making that change, the algorithm now produces a fairly convincing maze.

### Prism's Algorithm

[\[Source\]](#)

The standard version of the algorithm works something like this:

1. Choose an arbitrary vertex from  $G$  (the graph), and add it to some (initially empty) set  $V$ .
2. Choose the edge with the smallest weight from  $G$ , that connects a vertex in  $V$  with another vertex not in  $V$ .
3. Add that edge to the minimal spanning tree, and the edge's other vertex to  $V$ .
4. Repeat steps 2 and 3 until  $V$  includes every vertex in  $G$ .

And the result is a minimal spanning tree of  $G$ . For maze generation, in the second step, we select a random edge instead of edge with lowest weight.

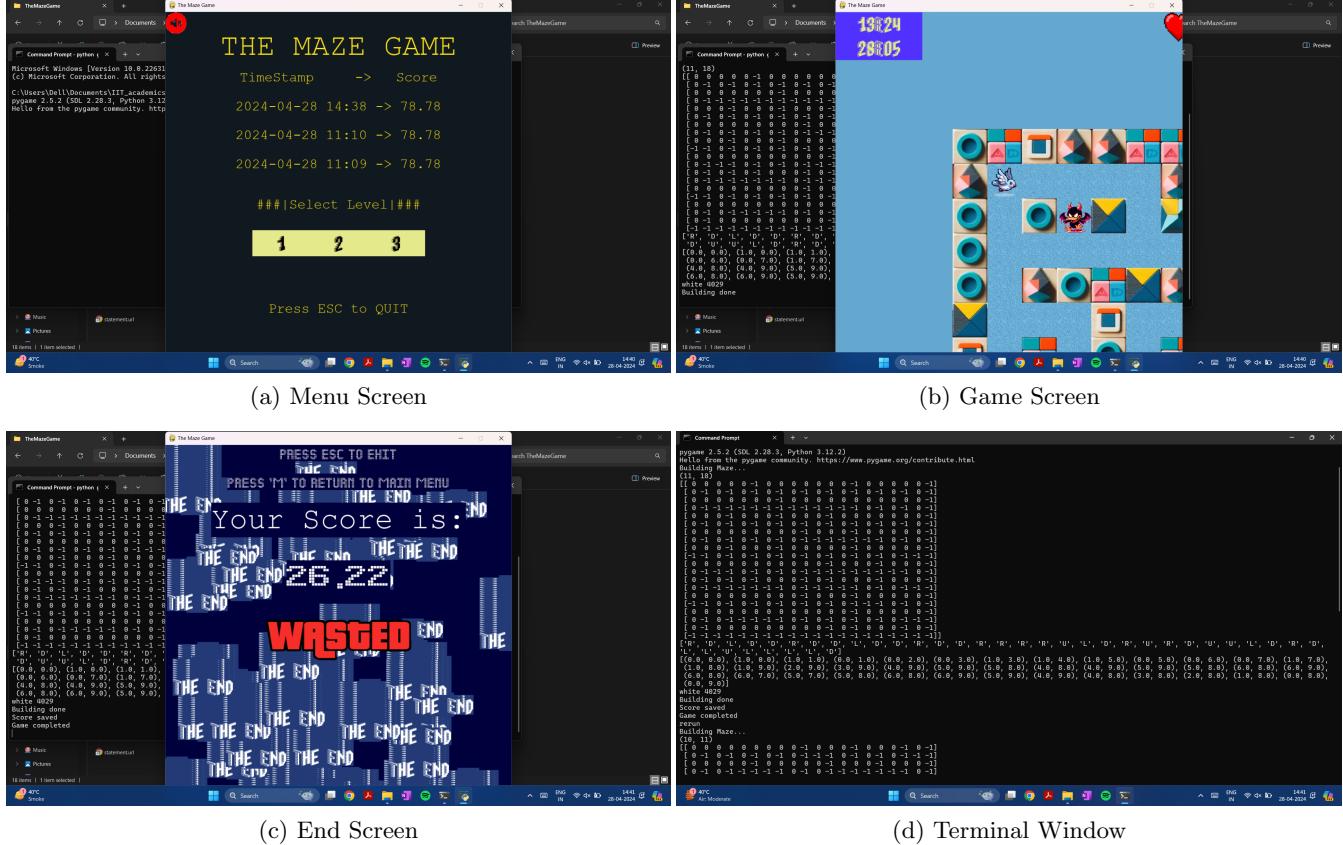
### Recursive Backtracing

[\[Source\]](#)

Here's the mile-high view of recursive backtracking:

1. Choose a starting point in the field.
2. Randomly choose a wall at that point and carve a passage through to the adjacent cell, but only if the adjacent cell has not been visited yet. This becomes the new current cell.

Figure 6



3. If all adjacent cells have been visited, back up to the last cell that has uncarved walls and repeat.
4. The algorithm ends when the process has backed all the way up to the starting point.

Seems simple enough.

## Aldous-Broder Algorithm

[\[Source\]](#)

Aldous and Broder were researching these uniform spanning trees, and independently arrived at the following algorithm:

1. Choose a vertex. Any vertex.
2. Choose a connected neighbor of the vertex and travel to it. If the neighbor has not yet been visited, add the traveled edge to the spanning tree.
3. Repeat step 2 until all vertexes have been visited.

Note: this algorithm is notable in that it selects from all possible spanning trees (i.e. mazes) of a given graph (i.e. field) with equal probability. The other algorithms shown don't have this property.

## Wilson's Algorithm

[\[Source\]](#)

Note: a spanning tree is a tree that connects all the vertices of a graph. A uniform spanning tree (UST) is any one of the possible spanning trees of a graph, selected randomly and with equal probability. The algorithm goes something like this:

1. Choose any vertex at random and add it to the UST.
2. Select any vertex that is not already in the UST and perform a random walk until you encounter a vertex that is in the UST.
3. Add the vertices and edges touched in the random walk to the UST.
4. Repeat 2 and 3 until all vertices have been added to the UST.

So, it's still doing the random walk, but this algorithm converges much more rapidly than Aldous-Broder.

## Hunt and Kill Algorithm

[\[Source\]](#)

Toda's algorithm is the "hunt-and-kill algorithm". Sounds violent, doesn't it? It's actually quite tame. In a nutshell, it works like this:

1. Choose a starting location.
2. Perform a random walk, carving passages to unvisited neighbors, until the current cell has no unvisited neighbors.
3. Enter "hunt" mode, where you scan the grid looking for an unvisited cell that is adjacent to a visited cell. If found, carve a passage between the two and let the formerly unvisited cell be the new starting location.
4. Repeat steps 2 and 3 until the hunt mode scans the entire grid and finds no unvisited cells.

## Growing Tree Algorithm

[\[Source\]](#)

A slick algorithm. Here's how it works:

1. Let C be a list of cells, initially empty. Add one cell to C, at random.
2. Choose a cell from C, and carve a passage to any unvisited neighbor of that cell, adding that neighbor to C as well. If there are no unvisited neighbors, remove the cell from C.
3. Repeat step 2 until C is empty.

Pretty straight-forward, really. But the fun lies in how you choose the cells from C, in step 2. If you always choose the newest cell (the one most recently added), you'll get the recursive backtracker. If you always choose a cell at random, you get Prim's. It's remarkably fun to experiment with other ways to choose cells from C.

## Eller's Algorithm

[\[Source\]](#)

It does this by building the maze one row at a time, using sets to keep track of which columns are ultimately connected. But it never needs to look at more than a single row, and when it finishes, it always produces a perfect maze.

Like the recursive backtracking algorithm, here's the "mile-high" overview of Eller's algorithm:

1. Initialize the cells of the first row to each exist in their own set.
2. Now, randomly join adjacent cells, but only if they are not in the same set. When joining adjacent cells, merge the cells of both sets into a single set, indicating that all cells in both sets are now connected (there is a path that connects any two cells in the set).
3. For each set, randomly create vertical connections downward to the next row. Each remaining set must have at least one vertical connection. The cells in the next row thus connected must share the set of the cell above them.
4. Flesh out the next row by putting any remaining cells into their own sets.
5. Repeat until the last row is reached.
6. For the last row, join all adjacent cells that do not share a set, and omit the vertical connections, and you're done!

## 10 Git Log

Here's the output of command `git log --graph --oneline --stat --color` with changes to .pyc files and changes to the another Game project [7] I referred to for building up camera in my game, removed.

```
* 552fbf3 finalise game
| audio/ducky-toy-sound.mp3 | Bin 17059 -> 11258 bytes
| audio/ducky-toy-sound2.mp3 | Bin 0 -> 17059 bytes
| data/saved_data.csv | 6 +-+
| game.py | 25 +++++-
| images/WASTED.png | Bin 0 -> 3248 bytes
| ...395c5f90-17de-480f-b710-36c0ea1de012.jpeg | Bin 224957 -> 0 bytes
| ...c385f6fc-cfe2-4f33-ba2a-2552487c9500.jpeg | Bin 94816 -> 0 bytes
| images/mute.png | Bin 0 -> 45358 bytes
| images/player - Copy.bmp | Bin 7738 -> 0 bytes
| images/player.bmp | Bin 3942 -> 0 bytes
| images/player_old.png | Bin 3002 -> 0 bytes
| images/sound.png | Bin 0 -> 37212 bytes
| images/wall.jpeg | Bin 398914 -> 0 bytes
| last_maze.txt | 60 ++++++-----
| modules/button.py | 21 +++++-
| modules/end_screen.py | 4 +-+
| modules/game_functions.py | 15 +---+
| modules/game_screen.py | 16 +---+
| modules/menu.py | 36 ++++++-----
| modules/settings.py | 2 +
| path.txt | 2 +-+
| 27 files changed, 110 insertions(+), 77 deletions(-)

* 511a3f7 enemies ready
| data/saved_data.csv | 37 +-----+
| fonts/Arcade.ttf | Bin 0 -> 27700 bytes
| fonts/Blox2.ttf | Bin 0 -> 108460 bytes
| fonts/led.ttf | Bin 0 -> 367124 bytes
| game.py | 15 +---+
| images/ducky_left.png | Bin 0 -> 797358 bytes
| images/ducky_right.png | Bin 0 -> 796872 bytes
| images/heart.png | Bin 0 -> 3093 bytes
| last_maze.txt | 60 ++++++-----
| modules/button.py | 13 +---+
| modules/end_screen.py | 8 +-+
| modules/game_screen.py | 42 ++++++-----
| modules/menu.py | 26 +----+
| modules/player.py | 41 ++++++-----
| modules/settings.py | 31 ++++++-----
| path.txt | 2 +-+
| 22 files changed, 167 insertions(+), 108 deletions(-)

* dc5fbfb Add enemies #piecing into walls
| data/saved_data.csv | 79 ++++++-----
| game.py | 16 +---+
| images/player_left.png | Bin 0 -> 12411 bytes
| images/{player.png => player_right.png} | Bin
| last_maze.txt | 38 +-----+
| modules/end_screen.py | 7 +-+
| modules/game_functions.py | 25 +++++-
| modules/game_screen.py | 45 ++++++-----
| modules/menu.py | 15 +---+
```

```

| modules/player.py | 25 +++++-
| modules/settings.py | 14 ++-
| path.txt | 2 +-
| the_latex_project/report.tex | 4 ++
| 19 files changed, 167 insertions(+), 103 deletions(-)
* 3bb7b5f Update report
| .../ltex.hiddenFalsePositives.en-US.txt | 1 +
| the_latex_project/log.txt | Bin 0 -> 98050 bytes
| the_latex_project/references.bib | 8 +-
| the_latex_project/report.aux | 96 ++-
| the_latex_project/report.bbl | 103 ===
| the_latex_project/report.bbl-SAVE-ERROR | 113 +++
| the_latex_project/report.bcf | 8 +
| the_latex_project/report.blg | 24 +
| the_latex_project/report.fdb_latexmk | 45 +-
| the_latex_project/report.fls | 107 ===
| the_latex_project/report.log | 318 ++++++---+
| the_latex_project/report.out | 38 +-+
| the_latex_project/report.pdf | Bin 187644 -> 4069284 bytes
| the_latex_project/report.synctex.gz | Bin 53120 -> 129393 bytes
| the_latex_project/report.tex | 559 ++++++-----+
| the_latex_project/report.toc | 38 +-+
| .../screenshots/Screenshot (164).png | Bin 0 -> 360483 bytes
| .../screenshots/Screenshot (166).png | Bin 0 -> 361422 bytes
| .../screenshots/Screenshot (168).png | Bin 0 -> 355008 bytes
| .../screenshots/Screenshot (169).png | Bin 0 -> 283114 bytes
| .../screenshots/Screenshot (170).png | Bin 0 -> 474084 bytes
| .../screenshots/Screenshot (171).png | Bin 0 -> 448526 bytes
| .../screenshots/Screenshot (172).png | Bin 0 -> 661720 bytes
| .../screenshots/Screenshot (174).png | Bin 0 -> 1145991 bytes
| .../screenshots/Screenshot (175).png | Bin 0 -> 591723 bytes
| .../screenshots/Screenshot (176).png | Bin 0 -> 1059786 bytes
| .../screenshots/Screenshot (177).png | Bin 0 -> 476293 bytes
| .../screenshots/Screenshot (178).png | Bin 0 -> 333411 bytes
| the_latex_project/screenshots/code_freq.png | Bin 0 -> 97736 bytes
| the_latex_project/screenshots/commits.png | Bin 0 -> 78707 bytes
| 30 files changed, 1292 insertions(+), 166 deletions(-)
* 8c85451 remove unwanted things, update problem statement, work on report
| ...Spring 2023-24) _ Maze Game in Python.pdf | Bin 3332129 -> 3343895 bytes
| data/saved_data.csv | 12 +-
| exp.ipynb | 3 -
| last_maze.txt | 20 +
| modules/button.py | 25 +-
| modules/camera.py | 27 +-
| modules/end_screen.py | 19 +-
| modules/game_functions.py | 14 +-
| modules/maze_logic/builder.py | 6 +-
| modules/maze_logic/hunt_and_kill.py | 22 +-
| modules/maze_logic/maze.py | 54 +-
| .../maze_logic/random walk from replit.py | 36 -
| modules/maze_logic/random_walk.py | 41 +-
| modules/menu.py | 3 +-
| modules/settings.py | 10 +-
| modules/sprites.py | 4 +
| modules/timer.py | 25 +-
| notes.md | 24 +-

```

```

| path.txt | 2 +-  

| the_latex_project/report.aux | 25 +-  

| the_latex_project/report.bcf-SAVE-ERROR | 2364 -----  

| the_latex_project/report.blg | 17 -  

| the_latex_project/report.fdb_latexmk | 32 +-  

| the_latex_project/report.flst | 44 +-  

| the_latex_project/report.log | 202 +-  

| the_latex_project/report.out | 23 +-  

| the_latex_project/report.pdf | Bin 136846 -> 187644 bytes  

| the_latex_project/report.synctex.gz | Bin 28646 -> 53120 bytes  

| the_latex_project/report.tex | 98 +-  

| the_latex_project/report.toc | 23 +-  

| 107 files changed, 528 insertions(+), 4344 deletions(-)  

* ddcff22 rename readme  

| README.md => notes.md | 0  

| 1 file changed, 0 insertions(+), 0 deletions(-)  

* 5de56d0 organise and comment things  

| README.md | 59 ++++++-----  

| data/saved_data.csv | 5 +-  

| game.py | 32 +++++-  

| images/player.png | Bin 3002 -> 12946 bytes  

| images/player3.png | Bin 12946 -> 0 bytes  

| images/player_old.png | Bin 0 -> 3002 bytes  

| modules/game_functions.py | 3 +-  

| modules/game_screen.py | 11 +++  

| modules/player.py | 27 +----  

| modules/settings.py | 75 ++++++-----  

| path.txt | 2 +-  

| the_latex_project/references.bib | 2 +-  

| the_latex_project/report.aux | 5 +-  

| the_latex_project/report.bbl | 13 +++  

| the_latex_project/report.bcf | 1 +  

| the_latex_project/report.blg | 28 +---  

| the_latex_project/report.fdb_latexmk | 27 +---  

| the_latex_project/report.flst | 6 ++  

| the_latex_project/report.log | 28 +---  

| the_latex_project/report.pdf | Bin 112827 -> 136846 bytes  

| the_latex_project/report.synctex.gz | Bin 28106 -> 28646 bytes  

| the_latex_project/report.tex | 11 +++  

| 26 files changed, 270 insertions(+), 65 deletions(-)  

* cce1681 Improve references.bib  

| ...toy-sound.mp3 => duck-toy-sound_copy.mp3} | Bin  

| audio/ducky-toy-sound.mp3 | Bin 0 -> 17059 bytes  

| audio/ducky-toy-sound.mp4 | Bin 0 -> 24907 bytes  

| data/saved_data.csv | 5 +-  

| modules/game_functions.py | 2 +-  

| modules/game_screen.py | 4 +-  

| path.txt | 2 +-  

| the_latex_project/references.bib | 96 +----  

| 10 files changed, 28 insertions(+), 81 deletions(-)  

* 80cff37 basic requirements fulfilled  

| .../angry-birds-theme-song-audiotrimmer.mp3 | Bin 0 -> 129913 bytes  

| ...ter-strike-jingle-cs-radio-ok-lets-go.mp3 | Bin 0 -> 26677 bytes  

| audio/duck-toy-sound.mp3 | Bin 0 -> 30765 bytes  

| audio/gta-v-death-sound-effect-102.mp3 | Bin 0 -> 124804 bytes  

| audio/happy-happy-happy-song.mp3 | Bin 0 -> 303291 bytes

```

```

| ./punch-gaming-sound-effect-hd_Rz1G1GE.mp3 | Bin 0 -> 25423 bytes
| audio/super-mario-beedoo_F3cwLoe.mp3       | Bin 0 -> 133870 bytes
| data/saved_data.csv                         | 28 ++++++-----+
| game.py                                     | 32 ++++++-----+
| modules/end_screen.py                      | 19 ++++++---+
| modules/game_functions.py                  | 19 +++++---+
| modules/game_screen.py                     | 2 +-+
| path.txt                                    | 2 +-+
| the_latex_project/references.bib          | 10 ++++++
| 19 files changed, 89 insertions(+), 23 deletions(-)
* f475a3f Complete End Screen and start adding sound
| audio/caproni.mp3                          | Bin 0 -> 4276122 bytes
| audio/wind.wav                            | Bin 0 -> 1059244 bytes
| data/saved_data.csv                        | 76 ++++++-----+
| game.py                                    | 5 +-+
| images/nest.png                           | Bin 0 -> 426146 bytes
| images/nest2.png                          | Bin 0 -> 578484 bytes
| modules/camera.py                         | 1 +
| modules/end_screen.py                     | 28 ++++++
| modules/game_functions.py                 | 16 +--+-
| modules/game_screen.py                   | 15 +--+-
| modules/maze_logic/builder.py            | 15 +--+-
| modules/maze_logic/hunt_and_kill.py     | 7 ++
| modules/maze_logic/random_walk.py       | 2 +-+
| modules/player.py                         | 3 +-+
| modules/settings.py                      | 18 +--+-
| modules/timer.py                          | 3 +-+
| path.txt                                  | 2 +-+
| 27 files changed, 119 insertions(+), 72 deletions(-)
* 70ec88f improve sprites
| data/saved_data.csv                      | 40 ++++++-----+
| game.py                                    | 2 +
| images/block1.jpeg                         | Bin 0 -> 7486 bytes
| images/block2.jpeg                         | Bin 0 -> 6086 bytes
| images/block3.jpeg                         | Bin 0 -> 7165 bytes
| images/block4.jpeg                         | Bin 0 -> 8485 bytes
| images/block5.jpeg                         | Bin 0 -> 8212 bytes
| images/block6.jpeg                         | Bin 0 -> 6637 bytes
| images/block7.jpeg                         | Bin 0 -> 7161 bytes
| images/blocks.jpeg                        | Bin 0 -> 140051 bytes
| images/blockscopy.jpeg                    | Bin 0 -> 169321 bytes
| images/path2.png                          | Bin 0 -> 411409 bytes
| images/player.jpeg                        | Bin 0 -> 109513 bytes
| images/player2.png                         | Bin 0 -> 13552 bytes
| images/player3.png                         | Bin 0 -> 12946 bytes
| images/sky.jpg                            | Bin 0 -> 5818 bytes
| modules/game_functions.py                | 21 ++++++---+
| modules/player.py                         | 2 +-+
| modules/settings.py                      | 16 +++++---+
| path.txt                                  | 2 +-+
| 24 files changed, 71 insertions(+), 12 deletions(-)
* aefb038 add scoreboard
| data/saved_data.csv                      | 9 +++
| game.py                                    | 59 ++++++-----+
| modules/end_screen.py                    | 1 -
| modules/game_functions.py               | 18 +---+

```

```
| modules/game_screen.py | 39 ++++++---  
| modules/maze_logic/builder.py | 8 +-  
| modules/settings.py | 50 ++++++++---  
| modules/timer.py | 29 +++++++  
| path.txt | 2 +-  
| 15 files changed, 180 insertions(+), 35 deletions(-)  
* b46fc77 implement end-screen  
| game.py | 53 ++++++-----  
| modules/__init__.py | 0  
| modules/button.py | 15 +----  
| modules/end_screen.py | 43 ++++++++---  
| modules/game_functions.py | 1 -  
| modules/settings.py | 9 +++  
| path.txt | 2 +-  
| the_latex_project/references.bib | 10 +++  
| 13 files changed, 106 insertions(+), 27 deletions(-)  
* f82232f Add ending screen  
| game.py | 41 ++++++-----  
| modules/button.py | 2 +-  
| modules/end_screen.py | 3 ++  
| modules/game_functions.py | 10 +--  
| modules/game_screen.py | 12 +--  
| modules/menu.py | 28 ++++++---  
| path.txt | 1 +  
| 12 files changed, 80 insertions(+), 17 deletions(-)  
* e06e1a1 initialise game completeion code  
| game.py | 29 ++++++---  
| modules/game_functions.py | 44 ++++++-----  
| modules/game_screen.py | 14 +++++  
| modules/maze_logic/builder.py | 30 ++++++---  
| modules/maze_logic/maze.py | 18 +----  
| modules/maze_logic/random_walk.py | 23 ++++++---  
| modules/menu.py | 1 -  
| modules/settings.py | 5 +-  
| 15 files changed, 114 insertions(+), 50 deletions(-)  
* 81e8ef3 updating camera failed  
| game.py | 6 +-  
| maze_logic/alg1.py | 23 -----  
| modules/camera.py | 55 ++++++-----  
| modules/game_functions.py | 13 +--  
| modules/player.py | 3 +-  
| modules.sprites.py | 1 -  
| 12 files changed, 50 insertions(+), 51 deletions(-)  
* 692a00d update todo  
| README.md | 7 +----  
| 1 file changed, 6 insertions(+), 1 deletion(-)  
* 6d07e35 Camera working  
| game.py | 9 +-  
| modules/camera.py | 28 +-  
| modules/game_functions.py | 73 +----  
| modules/player.py | 8 +-  
| modules.sprites.py | 10 +  
| the_latex_project/references.bib | 30 ++  
| 74 files changed, 1827 insertions(+), 28 deletions(-)  
* 45a9ab7 Worked on camera  
| README.md | 5 ++
```

```

| game.py | 7 +-+
| images/path.png | Bin 0 -> 1347603 bytes
| images/wall.jpeg | Bin 0 -> 398914 bytes
| modules/camera.py | 28 ++++++-----
| modules/game_functions.py | 47 ++++++-----+
| modules/player.py | 2 +-+
| the_latex_project/references.bib | 10 +***+
| 11 files changed, 65 insertions(+), 34 deletions(-)
* 1d42932 Added player move capacity
| exp.ipynb | 144 ++++++-----+
| game.py | 11 +-+
| modules/camera.py | 19 +**+
| modules/game_functions.py | 31 +**-
| modules/maze_logic/builder.py | 17 +-+
| modules/maze_logic/hunt_and_kill.py | 5 +-+
| modules/maze_logic/maze.py | 28 +**-
| modules/maze_logic/random_walk.py | 4 +-+
| modules/player.py | 25 ++-
| modules/settings.py | 6 +
| 17 files changed, 270 insertions(+), 20 deletions(-)
* 271096b Completed primary maze matrix
| exp.ipynb | 104 ++++++-----+
| game.py | 5 +-+
| modules/maze_logic/builder.py | 20 +***+
| modules/maze_logic/hunt_and_kill.py | 99 ++++++-----+
| modules/maze_logic/maze.py | 8 +-+
| modules/maze_logic/random_walk.py | 39 +****-
| the_latex_project/report.blg | 26 ++--+
| the_latex_project/report.fdb_latexmk | 22 ++--+
| the_latex_project/report.log | 12 +-+
| the_latex_project/report.pdf | Bin 112982 -> 112827 bytes
| the_latex_project/report.synctex.gz | Bin 28438 -> 28106 bytes
| 17 files changed, 308 insertions(+), 58 deletions(-)
* 07b796a modifications
| modules/maze_logic/random_walk.py | 21 ++++++-----+
| the_latex_project/report.tex | 3 +-+
| 2 files changed, 12 insertions(+), 12 deletions(-)
* 906971e implement walls and complete random walk
| modules/maze_logic/maze.py | 11 ++
| modules/maze_logic/random_walk.py | 75 ++++++-----+
| 3 files changed, 64 insertions(+), 22 deletions(-)
* 9eca23d successfully applied random walk to maze without walls
| exp.ipynb | 53 +***-
| modules/maze_logic/maze.py | 14 +-+
| ./maze_logic/random_walk from repl.it.py | 36 +***+
| modules/maze_logic/random_walk.py | 173 ++++++-----+
| the_latex_project/references.bib | 10 +
| 6 files changed, 174 insertions(+), 112 deletions(-)
* d2e23dd trying out my algorithm
| exp.ipynb | 67 +
| modules/maze_logic/maze.py | 14 +
| modules/maze_logic/random_walk.py | 125 ++
| the_latex_project/report.bcf-SAVE-ERROR | 2364 ++++++-----+
| the_latex_project/report.blg | 26 +-+
| the_latex_project/report.fdb_latexmk | 20 +-+
| the_latex_project/report.log | 14 +-+

```

```

| the_latex_project/report.pdf | Bin 111688 -> 112982 bytes
| the_latex_project/report.synctex.gz | Bin 26699 -> 28438 bytes
| the_latex_project/report.tex | 14 +-
10 files changed, 2612 insertions(+), 32 deletions(-)
* 86e74bf le tex
| noob.py => game.py | 0
| .../{refs.bib => references.bib} | 2 +-
| the_latex_project/report.aux | 24 +-
| the_latex_project/report.bbl | 49 +
| the_latex_project/report.bcf | 2405 ++++++
| the_latex_project/report.blg | 15 +
| the_latex_project/report.fdb_latexmk | 38 +-
| the_latex_project/report.flst | 47 +
| the_latex_project/report.log | 340 ++-
| the_latex_project/report.out | 19 +-
| the_latex_project/report.pdf | Bin 108698 -> 111688 bytes
| the_latex_project/report.run.xml | 85 +
| the_latex_project/report.synctex.gz | Bin 25871 -> 26699 bytes
| the_latex_project/report.tex | 12 +-
| the_latex_project/report.toc | 19 +-
15 files changed, 2971 insertions(+), 84 deletions(-)
* e684e1a Research for Maze generation algorithms out there
| .vscode/settings.json | 4 +
| maze_logic/alg1.py | 23 +++
| modules/player.py | 2 +-
| the_latex_project/presentation.aux | 5 -
| the_latex_project/presentation.out | 0
| the_latex_project/presentation.pdf | Bin 40246 -> 0 bytes
| the_latex_project/presentation.synctex.gz | Bin 1371 -> 0 bytes
| the_latex_project/presentation.toc | 0
| the_latex_project/refs.bib | 10 ++
| the_latex_project/report.aux | 14 ===
| ...tation.fdb_latexmk => report.fdb_latexmk} | 27 +--+-
| .../{presentation.flst => report.flst} | 51 +-----+
| .../{presentation.log => report.log} | 106 ++++++-----+
| the_latex_project/report.out | 9 ++
| the_latex_project/report.pdf | Bin 0 -> 108698 bytes
| the_latex_project/report.synctex.gz | Bin 0 -> 25871 bytes
| the_latex_project/report.tex | 100 ++++++
| the_latex_project/report.toc | 9 ++
20 files changed, 274 insertions(+), 86 deletions(-)
* adf76d8 modiy main game file acc to menu
| modules/game_functions.py | 2 +-
| modules/menu.py | 4 +-+-
| noob.py | 6 +----+
5 files changed, 7 insertions(+), 5 deletions(-)
* 144de1c Primitive menu made
| modules/menu.py | 30 ++++++-----+
| modules/settings.py | 14 ++++++-+
| the_latex_project/refs.bib | 12 +++++-
5 files changed, 42 insertions(+), 14 deletions(-)
* 2e6f8f3 Buttons enhanced
| README.md | 12 ++++++-+
| modules/button.py | 20 ++++++-----+
| modules/menu.py | 23 ++++++-----+
| modules/settings.py | 4 +-+-

```

```

| noob.py | 2 --
| the_latex_project/refs.bib | 16 ++++++
| 9 files changed, 60 insertions(+), 17 deletions(-)
* 2d634af Add button Menu creation fail with subsurface
| modules/button.py | 42 ++++++
| modules/game_functions.py | 20 ++++++-
| modules/menu.py | 19 ++++++-_
| noob.py | 5 +-_
| 7 files changed, 79 insertions(+), 7 deletions(-)
* 26339ee create menu and plan
| README.md | 26 ++++++
| modules/menu.py | 17 ++++++
| noob.py | 2 ++
| the_latex_project/refs.bib | 20 ++++++
| .../{presentation.tex => report.tex} | 0
| 5 files changed, 64 insertions(+), 1 deletion(-)
* 2b8c49e start latex work
| ...Spring 2023-24) _ Maze Game in Python.pdf | Bin 0 -> 3332129 bytes
| the_latex_project/presentation.aux | 5 +
| the_latex_project/presentation.fdb_latexmk | 70 +++++
| the_latex_project/presentation.flst | 128 ++++++
| the_latex_project/presentation.log | 237 ++++++
| the_latex_project/presentation.out | 0
| the_latex_project/presentation.pdf | Bin 0 -> 40246 bytes
| the_latex_project/presentation.synctex.gz | Bin 0 -> 1371 bytes
| the_latex_project/presentation.tex | 33 +++
| the_latex_project/presentation.toc | 0
| 10 files changed, 473 insertions(+)
* 0b2f33f statement update to google sheets link
| ...Spring 2023-24) _ Maze Game in Python.pdf | Bin 973566 -> 0 bytes
| statement.html | 8 ++++++
| statement.url.url | 5 +++++
| 3 files changed, 13 insertions(+)
* 39e1d33 most of static part done
| ...395c5f90-17de-480f-b710-36c0ea1de012.jpeg | Bin 0 -> 224957 bytes
| ...c385f6fc-cfe2-4f33-ba2a-2552487c9500.jpeg | Bin 0 -> 94816 bytes
| images/player - Copy.bmp | Bin 0 -> 7738 bytes
| images/player.bmp | Bin 0 -> 3942 bytes
| images/player.png | Bin 0 -> 3002 bytes
| modules/game_functions.py | 19 ++++++
| modules/player.py | 23 ++++++
| modules/settings.py | 10 +++++
| noob.py | 30 ++++++
| 12 files changed, 82 insertions(+)
* db150e5 Pdfs add
| ...Spring 2023-24) _ Maze Game in Python.pdf | Bin 0 -> 973566 bytes
| ...rch.Python.Oct_.2015.ISBN_.1593276036.pdf | Bin 0 -> 5641524 bytes
| 2 files changed, 0 insertions(+), 0 deletions(-)
* 3d0154a first commit
README.md | 1 +
1 file changed, 1 insertion(+)

```

## References

- [1] *Basic md formatting.* URL: <https://docs.github.com/en/get-started/writing-on-github/getting-started-with-writing-and-formatting-on-github/basic-writing-and-formatting-syntax>.
- [2] *Colors selection.* URL: <https://webflow.com/blog/best-color-combinations>.
- [3] *Fonts downloaded from here.* URL: <https://www.1001freefonts.com/bitmap-pixel-fonts.php>.
- [4] *geeksforgeeks guide for creating start menu in pygame.* URL: <https://www.geeksforgeeks.org/creating-start-menu-in-pygame/>.
- [5] *geeksforgeeks guide to creating sprites.* URL: <https://www.geeksforgeeks.org/pygame-creating-sprites/>.
- [6] *geeksforgeeks guide to rendering text.* URL: <https://www.geeksforgeeks.org/python-display-text-to-pygame-window/#:~:text=Create%20a%20Font%20object%20using,of%20pygame%20text%20surface%20object..>
- [7] *Github repo of the game used in Youtube camera tutorial.* URL: <https://github.com/HUCORP-admin/Pygame-Camera/tree/5aac706677e9f20bbf30002b15744cd17e64657>.
- [8] *Heart Image.* URL: <https://www.onlygfx.com/wp-content/uploads/2021/02/6-pixel-heart-4.png>.
- [9] Eric Matthes. *Python Crash Course.* William Pollock, 2016.
- [10] *Pygame Documentation.* URL: <https://www.pygame.org/docs/>.
- [11] *Pygame Sprites and Groups Explaination.* URL: <https://www.youtube.com/watch?v=4TfZjhw0J-8>.
- [12] *Replit - AI assistant for coding.* URL: <https://replit.com/@harshilsolank10/Python>.
- [13] *StackExchange avaialable algorithms for maze generation.* URL: <https://stackoverflow.com/questions/38502/whats-a-good-algorithm-to-generate-a-maze>.
- [14] *Wasted Font.* URL: <https://textcraft.net/style/PedroBlocks24/gta-wasted>.
- [15] *Website from which game theme(Caproni (An Aeronautical Designer's Dream)) is downloaded.* URL: <https://downloads.khinsider.com/game-soundtracks/album/the-wind-rises/03%2520Caproni%2520%2528An%2520Aeronautical%2520Designer%2527s%2520Dream%2529.mp3>.
- [16] *Website used for adding sound effects to game.* URL: <https://www.myinstants.com/en/categories/games/>.
- [17] *Youtube another camera tutorial.* URL: <https://www.youtube.com/watch?v=u7LPRqrzry8&t=925s>.
- [18] *Youtube Scrolling Maps Cameras tutorial.* URL: <https://www.youtube.com/watch?v=FDJU8lI0bVE>.