**15-112 FINAL TERM PROJECT DESIGN**

**Project:**

Generate random maze board using Kruskal’s algorithm or Prim’s algorithm. Showcases how complex algorithms originally meant to be used as searching algorithms and highly explored in graph theory, can also be used to generate mazes. There are two specific features these algorithms had that made them well-suited for this:

1. Both generate a path in a way that there is always a path between two nodes (cells)
2. They also generate paths in such a way that there is only one such path between any two given nodes (cells)

**My Code:**

1. user\_main.py handles the user interactivity.
2. Maze.py handles drawing the mazes and interactivity with the player module, by checking for collisions while navigating through maze and moving it accordingly
3. kruskal.py and prim.py handle the implementation of their respective algorithms, and return the set of walls necessary to draw the maze.
4. player.py handles drawing and moving the player

**User Interface:**

My user interface is simple and easy to use. The focus is not on playing or solving the mazes themselves, though that is a fun element added in. Rather, it is on the actual implementation of the algorithms and the heavy-lifting required by the code to do so. The background and the buttons were chosen and designed with care. The features look simple, and are also equally simple to use. The keys and buttons are hopefully clear. Instructions are always available. There is always a hint of fun involved, with the easy, informal language used throughout my design. This enables more closeness with the user, and allows them to feel amused as well as interested in the larger picture this project is looking at. Hopefully, a more interesting way to look at algorithms (which can be fun!).

**Previous Software:**

Most of what I had seen on the internet related to my original storyboard, but with my updated storyboard, I looked mainly at the examples of the code and videos, which can be found at:

1. <https://en.wikipedia.org/wiki/File:KruskalGeneratedMaze.webm>
2. <https://en.wikipedia.org/wiki/File:MAZE_30x20_Prim.ogv>

Both had similar aesthetic sense, with the formal, but pleasing black-and-white lines which generated mazes in real-time. However, I found this minimalistic approach forbidding. Also, the code was very unfamiliar to me, as it was written in Java or C, using vectors, and even the code written in Python, used modules other than tkinter, random and math.

I was looking for simple, straight-forward code, with fun graphics. The implementation might have also been done beautifully, but it was done using principles of Physics, and unfamiliar syntax. Therefore, I wanted to implement these algorithms without using more complicated reasoning, or modules, than I needed to. I wish I could have made mine a live-rendering one as well on the basis of these two, but it was a problem I could not tackle with this term project.

**Other Blurbs:**

I have chosen to document my failures here as well after being inspired by [this](https://www.princeton.edu/~joha/Johannes_Haushofer_CV_of_Failures.pdf). My reasoning is that this Term Project is this beautiful culmination of hard work, but it is so easy for others looking at this months ahead, to miss out on the failures, and the bugs. Therefore, I have decided to also include in my design document the pictures from the times I was standing on a chair in Gates, writing on whiteboard walls, trying to figure my problems out. I am going to show my thought process, because that was the really challenging part of what I have done. I am going to show the bugs in my code that had occurred previously, which made me want to tear my hair off at some times, and just laugh at other times.