Mouse Maze Report

My project is an interactive maze inspired by the common children’s puzzle. My goal with this project was to take what seemed like the most primate graphic type, drawn lines, and create a difficult, engaging, game. I used three types of lines to accomplish this design. Initially, a maze of pre-drawn green lines is displayed, made to resemble the walls of a hedge maze. The players can solve the maze by holding down the mouse to draw a red line through the correct path from start to finish. The complication is that the maze grows, and a new green line, representing growing hedges, will begin to fill the map to block the player. If the player does not work quickly, the new lines will block them from getting to the finish.

I used HTML and JavaScript to implement the project and hosted it on Tesla. My simple HTML sets up a canvas and generates text to explain the rules. My JavaScript is broken up into three sections, one for each line type used in my design.

The first line type coded is the red player line that allows interaction with the maze; this line is what transforms the project from an image to a game. I add event listeners for mouse actions such as mouse down, mouse up, and mouse move. Next, the code adds functions to draw the line when the mouse is button is down and moving, along with the mouse style such as color, shape, and thickness. Finally, the code uses moveTo and lineTo to draw the line. I do allow the player to mouse up to stop drawing as it feels more natural, though the rules of the game advise the player not to use this when actually playing.

The next line type is the growing maze walls. Originally, I was going to have this line be a spotlight that the players had to avoid. During testing, this slowed the game down without adding any real challenge as the player simply had to stop and wait for the, originally white, spotlight to pass. After testing, I decided on a green line to represent growing hedges. This had the opposite effect of speeding the game up as players now need to race against the growth or else risk being cut off from reaching the finish. The code in this section simply draws a line with a strategically pre-seeded start position and velocity. If this was randomized, it would be possible to put the player in unwinnable scenarios early in the game. Finally, code is provided to cause the maze growth to bounce off the canvas limits to keep the obstacle in play. Since motion is used, the project could be extended here in the future to change the path or speed of the growth line.

The final line type is the static maze walls. These are the simplest lines and are predefined to create a winnable path given the growth line. Though this section of the code is several hundred lines, the pattern is simple of spanning across the rows left to right and drawing lines where there should be barriers. Once one row is complete, the code moves to the next row down. The grid is based on a unit system that functions like a constant. Future extensions of the project could adjust the unit size to scale the map up or down.

One challenge I faced was the original design choice to have the moving line be a spotlight that the player needed to avoid. Pivoting to the maze growth line greatly improved the pace of the game. Another challenge, that I was not able to solve, was a bug in the growth line. When the page starts, the growth line is thick; however, when the player starts up the red line, the growth line shrinks. I believe this is related to the line style being shaped between the growth and player line. This bug is purely aesthetic and does not change the play of the game, so I left it as is. There are several areas I could extend this project in the future, as mentioned above. I could change the maze walls, their scale, and the path of the growth line. I could also add more UI elements such as a start button which would likely be preferred to starting the game on the page load. I enjoyed and learned a lot from this project and would be interested in improving and extending it in the future.