

Dhruv Patel (NetID: dnp78)
Professor Aanjaneya
CS:428
7 March 2021

Homework 2

For my implementation of this assignment of maze generation and including interactivity, I have included an HTML, CSS, and JavaScript file to generate a maze based on the specifications provided by the user and then enabling the user to guide a rat figure, which is an arrow in my case, through the maze. To run my program, open up the index.html file in your browser or run the same file with live-server. When it came to drawing the lines of the maze, I had implemented a scaling factor to ensure that maze dimensions would fit in the canvas size and inputs from 1 to 100 should work completely fine; however, as the maze becomes larger, it becomes slightly more difficult to see the mouse figure when controlling it through the maze.

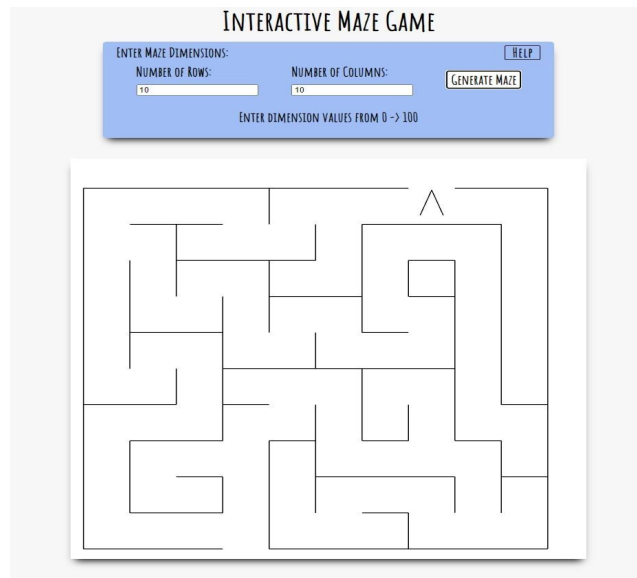
The approach I used was using 2 2d arrays of cells that represent the walls of each cell by default as an array ([1,1,1,1]) and boolean false values to determine if the cells were visited or not when randomly choosing a direction where there is a valid path. I randomly begin somewhere in the 2d array of cells, and then randomly choose what direction to go to based on what cells were unvisited and remove walls from those cells to connect them. After this is complete and all cells are visited, I was left with a maze and just had to randomly remove 2 walls from the perimeter to create an entrance and exit and save these positions globally to use later. Then, I created a function to draw the maze and included my scaling factor to shrink the maze down so that it's completely visible in the canvas window.

Then, I created all the functions for drawing the mouse positions whether it was up, down, left, right and adding the interactivity allowing the mouse to move based on its cell position and the positions of possible walls to disable the mouse from being able to move through the walls. I added the interactivity or controls of the mouse movement by adding an onkeydown event listener and moved the mouse accordingly based on what keys were pressed. Additionally, I added some styling to make the web page a little more attractive.

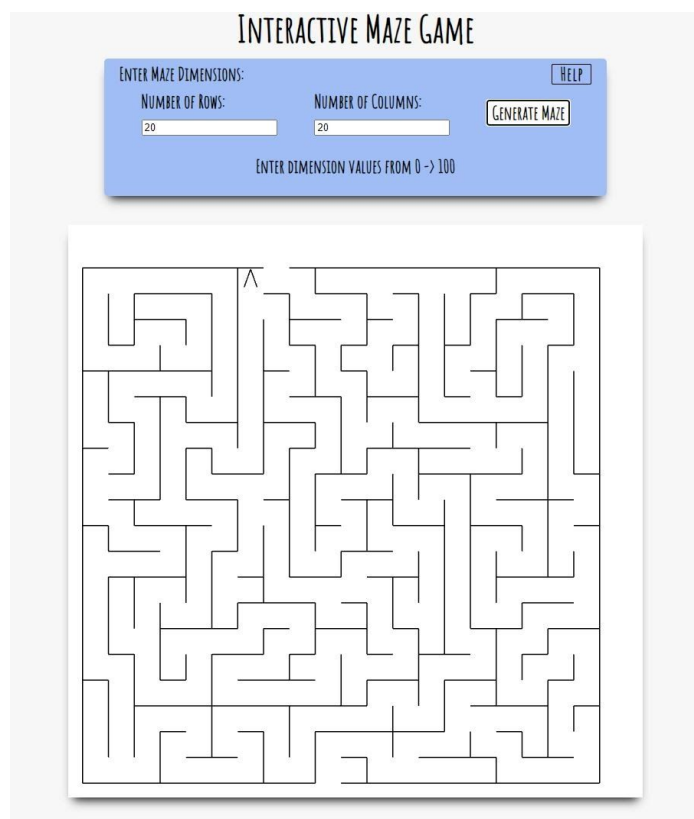
Once the user completes the maze and reaches the end cell, an alert will prompt them that they have won and the window will reload after 5 seconds to reset the web page. By default, the mouse will always be generated facing up.

Examples)

10 x 10



20 x 20



100 x 100

INTERACTIVE MAZE GAME

ENTER MAZE DIMENSIONS:

HELP

NUMBER OF ROWS:

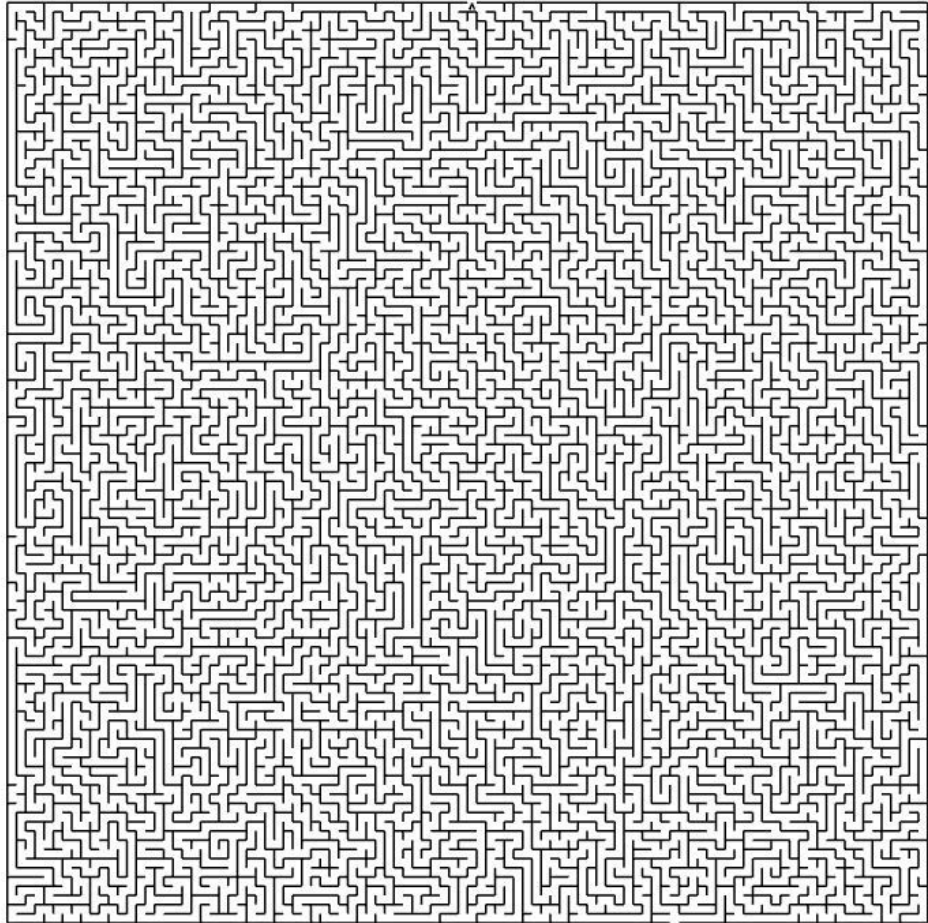
NUMBER OF COLUMNS:

GENERATE MAZE

100

100

ENTER DIMENSION VALUES FROM 0 -> 100



A large, square maze grid with a complex, winding path. The maze is composed of black lines forming a grid of paths and dead ends. The overall shape is a square, and the internal structure is highly intricate, with many small loops and dead ends. The maze is displayed on a white background with a subtle drop shadow.

PS: If you input 100 x 100 and nothing loads, please refresh the page and try again as sometimes it throws an error of Maximum call stack size exceeded as my maze generation works recursively. Other than that, my program should run completely fine to the exact specifications of the assignment.