Problem Statement

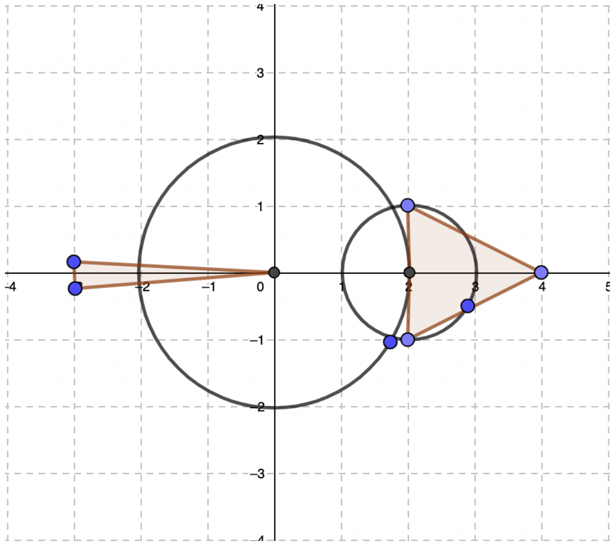
1. Take input Width and Height for maze dimension
2. Create a Maze with only one solveable path
3. Add a graphical rat to traverse the maze
4. Use Up-Left-Down-Right keys to move the rat
5. Rat should point to the direction of the arrow button

Implementation

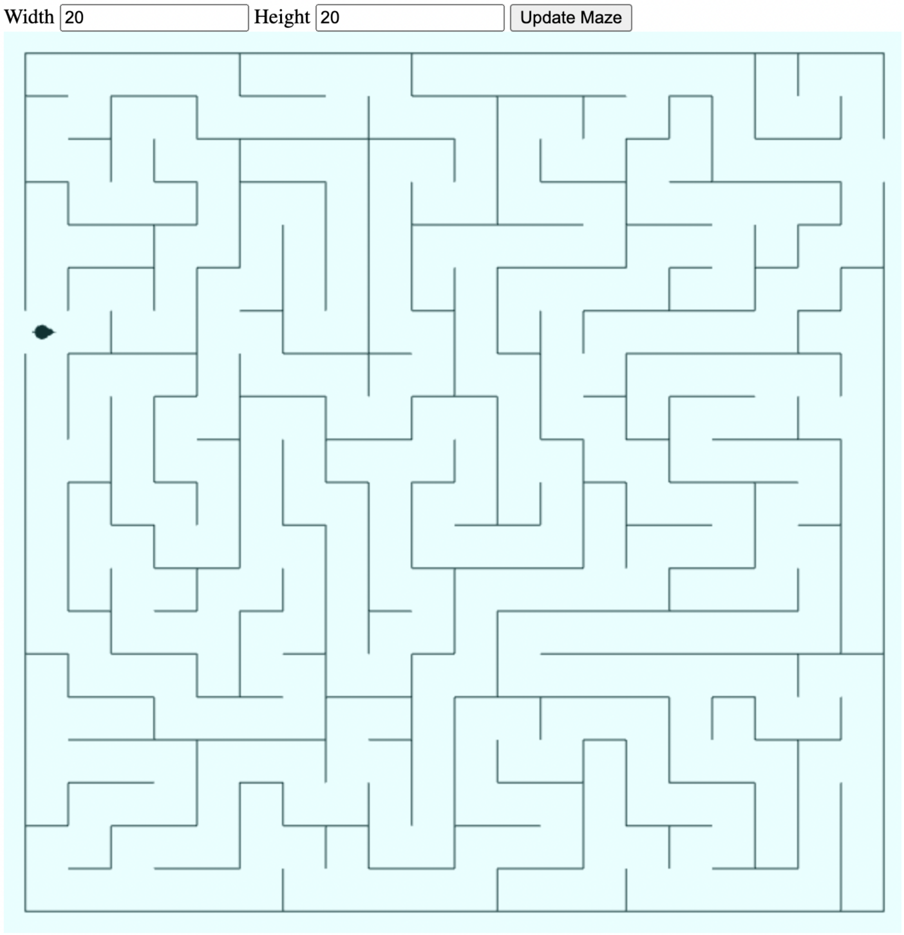
I used Graph data structure to store the data. I created 2 graphs, one for the vertices (intersection points), and the other for the cells (rat position). For every maze, we create new graphs with all the edges connected. Then, a true path and other dummy paths are created by erasing the edges using backtracking. This is acheived by keeping track of visited cells, and choosing the next cell at random from the unvisited neighbouring cells. Each visited cell is added to the fringe. If the current cell has all the neighbours that are already been visited, then the pointer is moved to the last cell visited cell which has unvisited neighbours, by recursively popping the last cell in the fringe.

Whenever we move to next cell, an edge is deleted that is connecting the 2 cells, and one more edge is deleted that is connecting the two vertices whose edge is crossed. I am using the vertices to draw the maze and using cells to track path of the cell. So, to move the rat from one cell to other, I check if the two cells have an edge. If not, the rat can move, otherwise only direction of the rat is changed, and it is not translated to the next cell.

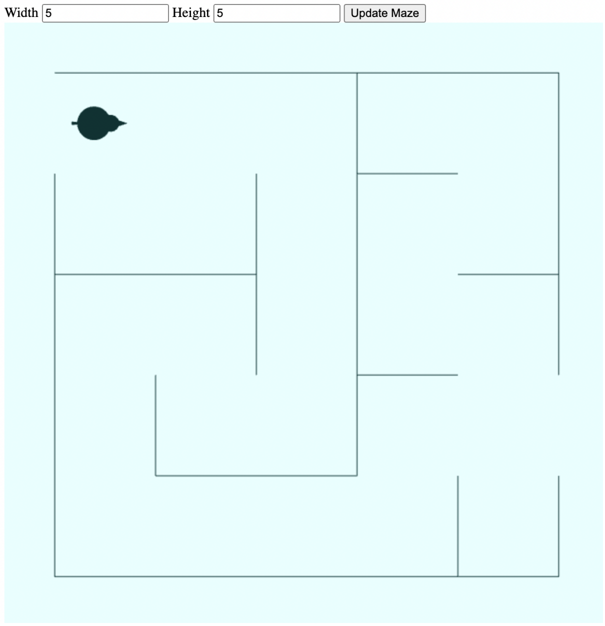
For rat, I have created 2 circles and two triangles, and used TRIANGLES to draw the rat. With each hit of arrow key, I rotate the points of these geometrical objects by relevant degrees.
Rat Geometry-



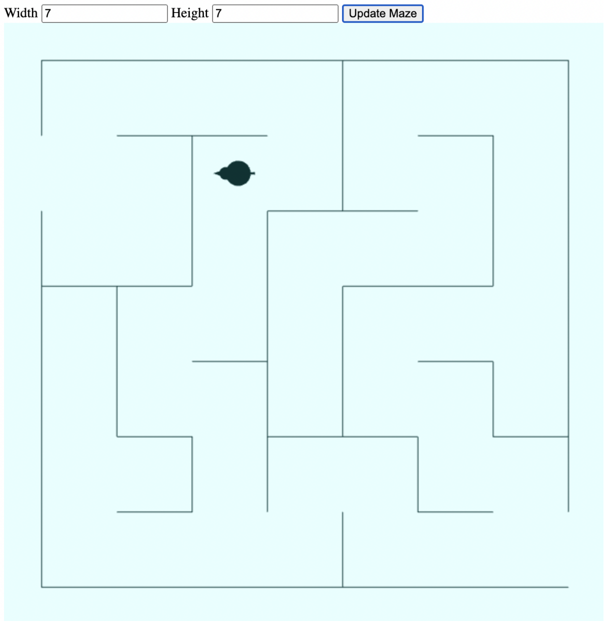
Results



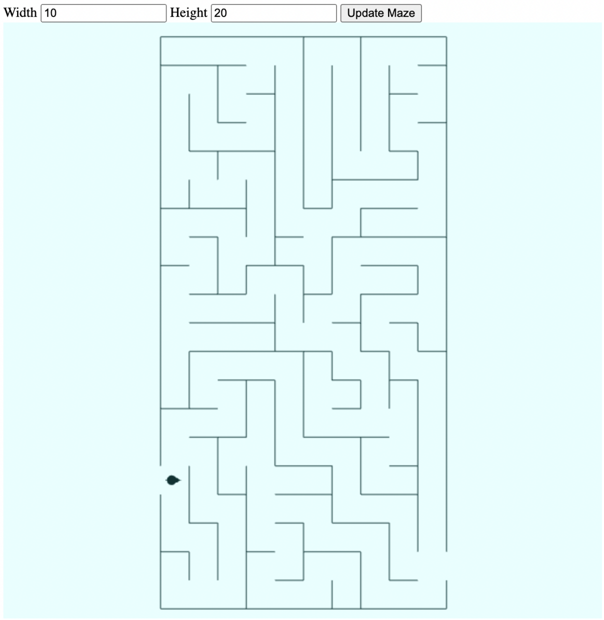
Width 20 Height 20

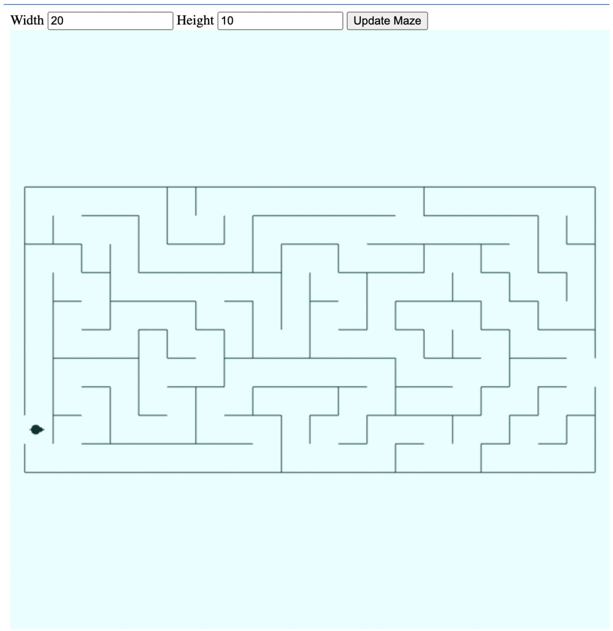


Width 5 Height 5



Width 7 Height 7


Width 10 Height 20


Width 20 Height 10