1. Maze Gen as Graph
    a. The vertices would be each “box” within the maze
    b. When the “boxes” are connected to other “boxes” and not blocked by walls
    c. I think depth-first would work best, randomly choosing a direction to go until it eventually ends up at the “ending cell”. Then it would work it’s way back up to unvisited cells and continue the same previous process until it reaches a cell that is surrounded by visited cells. Then it would backtrack and continue this process until the entire maze is generated.
2. Maze Solve as Graph
    a. Vertices would be each cell
    b. Edges would be when a cell can be reached from another cell without being blocked by walls
    c. Depth first again works well, going as deep as possible until it finds the “ending cell” and backtracking when it gets stuck.