CS 314: Final Project

Procedurally generated maze solving

By Yusdel Lorenzo

Overview

This project is comprised of a generator, solver and test python script. By using all three in tangent you can generate a N x N maze which uses breadth first search to knock down walls thus making the maze. Once the maze is generated a svg file is produced containing the maze. For testing purposes, I took the maze and converted it to a 809 x 1008 jpg file in which the starting point of the maze is (400, 984) and the ending point of the maze is (398, 25) in order to traverse the maze from the bottom to the top.

Running "**python3 Generator.py"** runs the generator file which asks for maze dimensions and outputs a maze.svg file.

Running “**python3 Solver.py <maze_image> output.png**” runs the A* algorithm on a particular image and generates a completed maze image with a red line showing the shortest path from the start to the goal.

Running “**python3 Test.py**” it runs the A star algorithm on the entire test-set folder which is filled with various mazes.

By calculating the time taken for each maze and taking the average we can find out how the performance of the algorithm A star is impacted by a procedurally generated maze environment. Future research projects could compare the results found here with other path finding algorithms.