

Final Project Documentation

Paul Plew

April 30, 2021

▼ Summary

For the final project I decided to make a program that generates mazes procedurally. The original idea was to have two maze generation algorithms but that was too ambitious and I was not proud of the second algorithm. The program generates a maze using the depth first algorithm. I mashed the playing, saving and generation of the maze into one screen. I spend a lot of time making this program efficient, so it runs extremely fast.

I decided to skip the maze solving AI for now, but I will revisit this project at some point when I am more capable and implement it. In the place of the AI I polished the code to make it easy to read and make the program visually appealing.

▼ How to Use

My final project is pretty simple to use, it opens on a start screen with two buttons. Both buttons will then lead to mazes that can be solved. The top button will animate the generation of a maze using the depth-first maze generation algorithm, and the bottom will generate a maze immediately. Once the mazes are created the user will have the option to create a new maze, save the maze, or play the maze. If the user selects the new button a new maze will be generated in the same way as the original. If they select the save button the browser will either save the maze immediately or prompt the user on how they want to proceed. After the maze is saved it can be printed out and solved by hand. The play button will start a timer and enter a new mode where the user can solve the maze using the arrow keys. There are two buttons, one will go back a page and the other will reset the timer and the user's location in the maze. When the maze is completed the screen will switch, the timer will pause, and the path taken will be shown. At this point the user can return to the home screen and start over.

▼ Proposed Plan vs Actual Plan

▽ Proposed Plan

- April 9: Finish Start page and depth-first generator
- April 16: Finish Kruskal's maze generator
- April 23: Make Printable Maze page & Playable Maze page
- April 25: Finish Maze solving AI
- April 30: Finalize, fix bugs, Add color selection and size selectiond

▽ Actual Plan

- April 9: Finish depth-first generator
- April 16: Bug fix and Start Screen
- April 23: Maze Saving
- April 25: Difficulty Selection
- April 30: Playable Maze mode and Bug fixes

▼ Best Part

The best part of my code is the maze generation mode. This mode reuses all of the code from the animation mode, but uses a while loop to make the cells until the stack is empty. This approach is elegant and only a few lines of code, but it didn't start that way. I went through a lot of trial and error to make the maze generate instantly, and when done the final result is amazingly simple.

▼ Peer Review Input

My peer review didn't have a lot of negative feedback the only thing they advised me on was to make my code more human readable. I spent some time moving methods around to make the flow of the program make more sense, and to remove unnecessary if statements and functions.