CSCI 200 - Fall 2023
Foundational Programming Concepts & Design

A6 - SFML: Maze Runner

→This assignment is due by Thursday, December 07, 2023, 11:59 PM.←

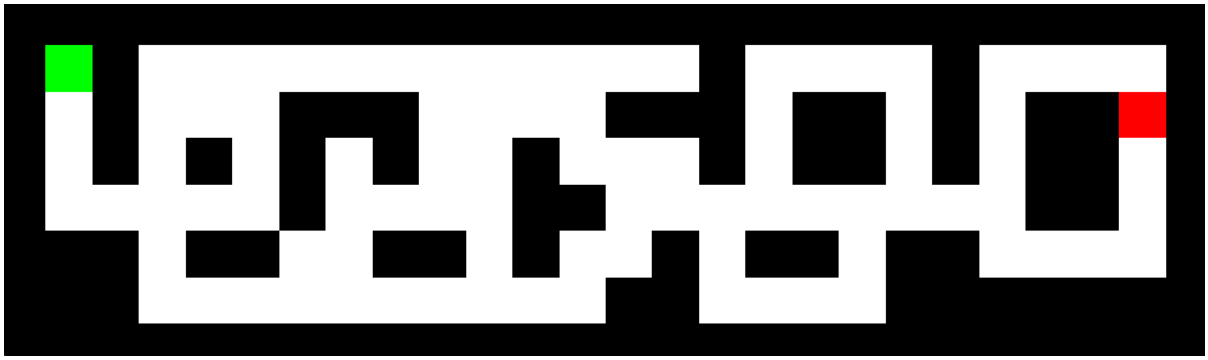
→ As with all assignments, this must be an individual effort and cannot be pair programmed. Any debugging assistance must follow the course collaboration policy and be cited in the comment header block for the assignment.←

→ Do not forget to complete the following labs with this set: L6A, L6B, L6C ←

→ Do not forget to complete zyBooks Assignment 6 for this set.←

Jump To: **Rubric Submission**

For this assignment you will build upon **Lab6C**. This piece will now animate a solution to the maze, such as Maze 7 below.



Solve the Maze (Maybe)

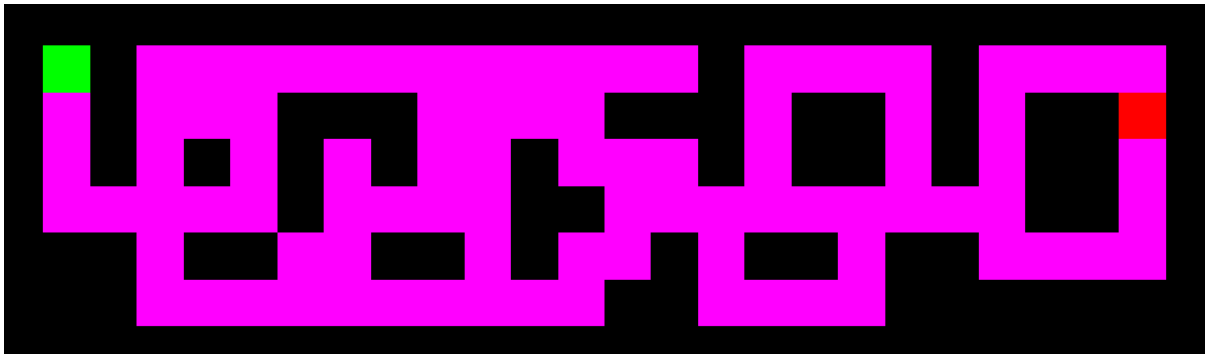
RMP-H

Before the maze has been drawn, ask the user via the terminal how they wish to solve the maze. Either by BFS or by DFS.

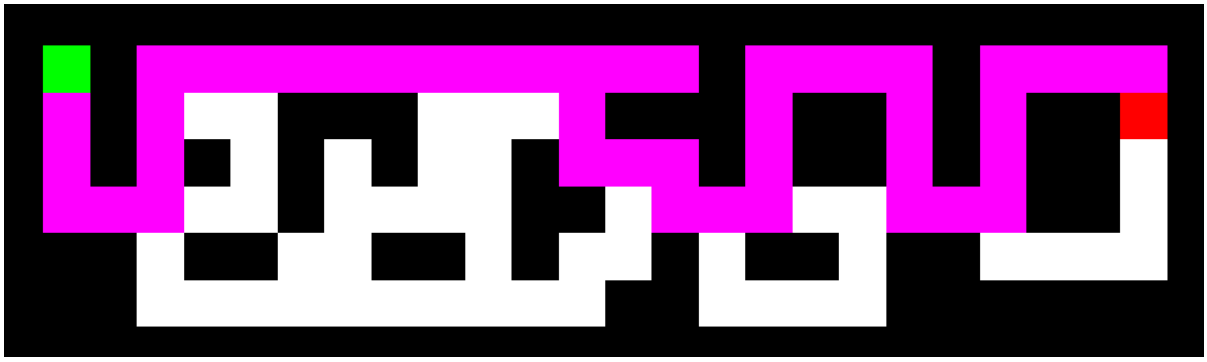
Based on the user's selection, search the maze using BFS with a Queue or DFS with a Stack as appropriate. Begin your search at the `S` space and continue until the `E` space is reached or the search is exhausted.

When the search is complete, draw the maze with visited cells colored Magenta.

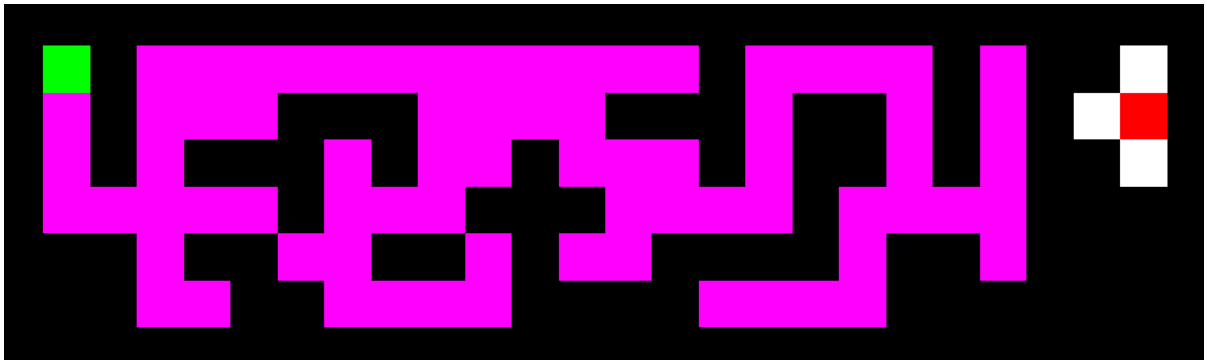
An example of a successful BFS search is shown below:



An example of a successful DFS search is shown below:



An example of an unsuccessful search is shown below:



Animating the Solve

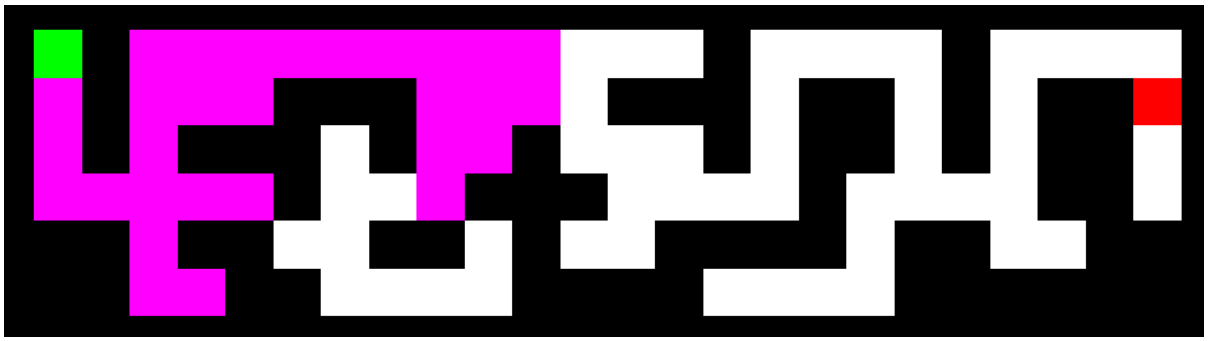
To animate each step of the search process, we'll use the draw loop as the loop for our search process. Each iteration, we will do a few steps:

- clear the current window
- Draw the current maze state (walls = Black, unvisited = white, visited = Magena, start = Green, end = Red)
- Get our next location to process (if it exists)
- Check if it's the end location
  - Print end reached
  - For the purposes of our animation - empty the list of nodes to process to halt the search
- If it's not the end location
  - Check each neighbor to potentially add to list to process
- display the current window
- check for events (window close, q or escape keypress)
- Tell SFML to sleep for a brief amount of time - without this, the animation will run as fast as the window can redraw. We'll explicitly put in a delay before drawing the next frame. This delay equates to our framerate. Choose a delay of 50 milliseconds, which equates to 20 frames per second. The following SFML call will sleep for `n` milliseconds:

  ```
  sf::sleep( sf::milliseconds(n) );
  ```

  You may tune this value up or down as you are testing to trace the process of your search algorithm.

An example of a partial search in progress is shown below:

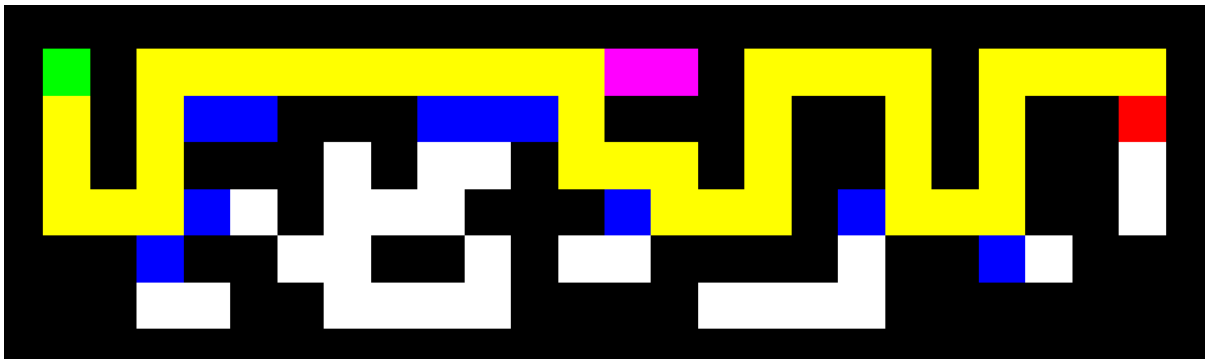


Extra Credit

There are two potential extra credit pieces:

1. Color potential cells to explore Blue. These are locations that have been added to the neighbors to process list but not yet visited to check.
2. Color the path from start to end in Yellow.

An example of both is shown below;



Testing

The graders will build your program with the Makefile you provide to match your code structure and run your program against five private test mazes.

Grading Rubric

Your submission will be graded according to the following rubric:

Points	Requirement Description
0.5	Submitted correctly by Thursday, December 07, 2023, 11:59 PM
0.5	Project builds without errors nor warnings.
2.0	**Best Practices** and **Style Guide** followed.
0.5	Program follows specified user I/O flow.
0.5	Public and private tests successfully passed.
2.0	Fully meets specifications.
6.00	**Total Points**

Extra Credit Points	Requirement Description

+1	Display neighboring squares to be checked in blue
+1	When a solution is found, color the path from start to end in yellow (this does not need to be the shortest path)

Submission

Always, **always**, **ALWAYS** update the header comments at the top of your main.cpp file. And if you ever get stuck, remember that there is LOTS of **help** available.

Zip together your `main.cpp, Makefile, *.hpp, *.h, *.cpp` files and name the zip file `A6.zip`. Upload this zip file to Canvas under A6.

→This assignment is due by Thursday, December 07, 2023, 11:59 PM.←

→ As with all assignments, this must be an individual effort and cannot be pair programmed. Any debugging assistance must follow the course collaboration policy and be cited in the comment header block for the assignment.←

→ Do not forget to complete the following labs with this set: L6A, L6B, L6C ←

→ Do not forget to complete zyBooks Assignment 6 for this set.←

Last Updated: 10/06/23 14:52

Any questions, comments, corrections, or request for use please contact jpaone {at} mines {dot} edu.

[Jump to Top] [Site Map]