COMP167 - Maze Builder

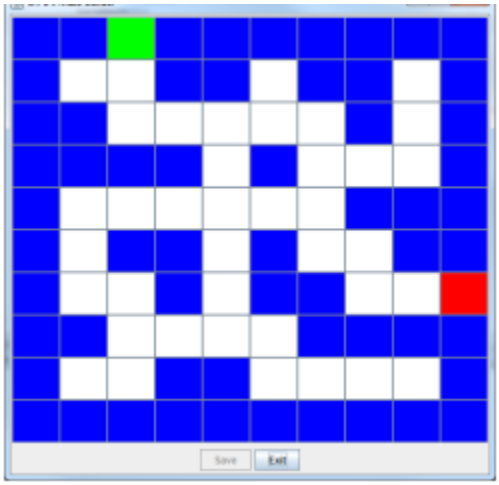
A maze consists of a series of walls and paths with an entrance and exit. We can represent a maze with zeros and ones where a zero indicates a position/tile in the maze containing a wall and a one represents part of the walkable pathway. The maze has a single entry point and a single exit point represented by the characters 'S' and 'E' respectively. You are going to write an application that will facilitate the creation of new mazes graphically. The user will first specify the maze dimensions (e.g. 25 x 25) and you will present an initial board containing a matrix of buttons matching the dimensions of the maze. The user can then click the buttons/tiles and change the color from the wall color to the path color. Once the maze is complete, the user can then save the maze to a file containing zeros, ones and the entry and exit characters. The following shows a 10x10 maze along with its text file representation.

Output File:

The output file of your maze builder should have the row and column dimensions on the first line represented as integers. The next rows and columns should then match the dimensions of the maze where zeros represent walls and ones represent paths. The entrance should be marked by an 'S' and the exit by an 'E'. The entry and exit should be the only non-wall tiles on the maze edges.

Input:

You should prompt the user for the number of rows and columns in the maze using JOptionPane.




```
10 10
00S0000000
0110010010
0011111010
0000101110
0111111000
0100101100
011010011E
0011110000
0110011110
0000000000
```

Commands:

Your application should provide Save and Exit commands. The Save command should write out the maze file according to the output file format specified in the Output File section above. The Exit command should also ask the user if a Save should be performed before exiting.

Tiles/Buttons:

You are free to use the four colors of your choice in the maze: walls, path, entry and exit. When a user clicks a tile/button, it should toggle from wall to path to entry to exit and then back to wall on successive clicks. You do not have to error check your maze (e.g. making sure that there is only one entry and exit). Again, all tiles on the maze edges should be walls except the entry and exit squares.

Stages:

Basic Maze - Shows the Stage with a Scene containing a BorderPane. The center zone of the BorderPane will contain a nxn GridPane. Each cell of the GridPane will contain a Button. The bottom zone of the BorderPane will conain an HBox. Add the “Save” and “Exit” buttons to the HBox and set the alignment so that the buttons are centered. Make sure your maze buttons are created with their backgrounds set to the wall background color (blue is the color I used for walls).

Basic Maze Interactive - The user can click the tiles and they will toggle between the four maze colors. Code will be provided to handle the events related to the mouse clicks.

Save Maze - Your Stage displays the basic maze plus a second panel containing the Save and Exit buttons. Clicking the Save button will save your maze to a file. The Save operation basically traverses your 2D array of buttons and outputs a 0, 1 , 'S' or 'E' depending on the color of the button/tile. The JButton method getBackground() will return a Color object representing the background color of the button. The constructor for the JFrame class should provide a parameter with the output file name.

Exit Maze - Clicking the Exit button should present the user the option to save (using the JOptionPane) then exit the application (System.Exit(0)).

Complete the Save Maze option for the full 20 points.