The Maze Programming Assignment

Task: Create a GUI based application that will give the option to create a maze randomly or read it from file. The GUI should incorporate colour for each cell as described/illustrated below. A mouse is placed in an open position. One of your tasks is to write a **recursive method** that helps the mouse to find its way (s) out and output the path using the character '+'. The find path method is invoked when the user clicks on the 'Find Path' button. The mouse can only move up, down, left or right. The GUI will be updated with path found or a message indicating that no path was found.

In the randomly created maze: Generate and output a maze made up of a specific number of rows and columns. The maze will have border spaces, 'B', open spaces, 'O', and an exit, 'X', placed randomly on the borders. The exit cannot be in any of the 4 corners.

Example 1: Below is an example of a 7x11 maze.

B	B	B	B	B	B	B	B	B	B	B
B	O	O	O	B	O	B	O	O	O	B
B	O	B	O	B	O	B	O	B	B	B
B	O	B	O	S	O	O	O	O	O	B
B	O	B	B	B	B	O	B	B	O	B
B	O	B	O	O	O	O	O	B	O	B
B	B	B	B	B	B	B	X	B	B	B

Rows = 7; Columns= 11; B = Barrier; O = Open path
X = is the exit S = mouse's starting position inside the maze
+ = the path to exit from the maze

Sample input file for when maze is read from file:

- The maze size and position of the mouse in the Maze are read from a file

```
5
6
B
O
S
X
BBBBBB
BOOOBB
BSBOBB
BOOOOB
BBBBXB
```

Legend:
R - Rows
C - Columns
B - Border
O - Open path
S - Starting position for the mouse
X - Exit position from the maze

```
BBBBBB
BOOOBB
BSBOBB
BOOOOB
BBBBXB
```

Below is the solution for Example 1 shown above.

B	B	B	B	B	B	B	B	B	B	B
B	O	O	O	B	O	B	O	O	O	B
B	O	B	O	B	O	B	O	B	B	B
B	O	B	O	S	+	+	O	O	O	B
B	O	B	B	B	B	+	B	B	O	B
B	O	B	O	O	O	+	+	B	O	B
B	B	B	B	B	B	B	X	B	B	B

Additional Considerations:

- **Before beginning your to program, you will write algorithm to plan out how you will generate the random maze, how your will load the maze from file, and the recursive method that is responsible for finding the path out of the maze.**
- Your program should prompt the user for the file to load – different files may be used for testing your program. So be sure to follow the file format provided.
- Simplify your GUI as much as possible.
- Be sure to adhere to coding standards discussed in class.
- Make good use of methods in your implementation.
- **There should be no sharing of strategy, design, or code with any other students. This assignment is designed to demonstrate YOUR knowledge, problem solving ability, and command of the java programming language.**

Rubric

Category	Level			
Knowledge				
• Programming Syntax and Code conventions followed – naming, structure, brackets, loops	1	2	3	4
• Maze is read from file, adhering to file specifications and stored in a 2D array	1	2	3	4
Thinking				
• Algorithm provides all steps in great detail (the "what" and the "how") • Demonstrates insightful understanding of the task	1	2	3	4
• Effective use of recursion to find path to exit maze – using recursion to traverse your maze (2D array) with no unnecessary iterations • Recursive method has well defined base case(s) and recursive steps	1	2	3	4
• Program is efficient avoiding unnecessary iterations, decisions, method calls, and memory allocation • Effective use of methods	1	2	3	4
Communication:				
• Commenting – program is properly formatted, indented, documented, meaningful names chosen • Formatting – appropriate use of white space, code is neat and easy to read • GUI layout is user friendly and visually appealing	1	2	3	4
Application:				
• Program prompts for the file to upload the maze from. Maze is read from file and stored in a 2D array.	1	2	3	4
• GUI represents maze loaded from file using appropriate colours • GUI includes a button to invoke the recursive method that will 'Find Path'	1	2	3	4
• Program correctly reports whether path is found or not found.	1	2	3	4