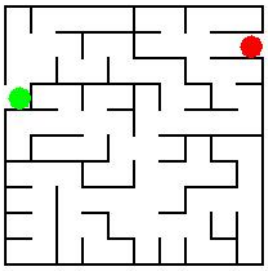
Assignment 1: 2D Maze Drawing and Customization

Out: 3rd September 2015 **Due**: 15th September 2015 at 11:55pm



Files

Maze files

Several files have been provided to you. The basic data file is the maze file (e.g. maze-10x10.txt shown above). The maze file contains data in the following order:

Number of rows in maze

Number of columns in maze

Start cell of maze and end cell of maze, each in terms of row and column in that order

An integer code for each cell, row-wise with each row going from left to right

The integer code uses only the 4 rightmost bits in the integer (thus the only valid values are from 0 to 15). These bits, from left to right, represent the left, previous row, right, and next row neighbors of a cell. A bit is 1 if there is a wall between this cell and its neighbor. For example, if a cell code is 13 (1101), it means that this cell has walls between itself and (1) left neighbor (2) neighbor on next row, same column (3) neighbor on previous row, same column. The '0' indicates that there is no wall to its right.

Several other maze files have been provided to you. **Note that the row and column numbers begin with 0.**

GenerateMaze.jar

This is a runnable jar file that can produce mazes. To generate a 40x40 maze in a file called "maze-40x40.txt", this program can be run from the command prompt as follows:

```
java –jar GenerateMaze.jar –dims 40 40
```

PathFinder.jar

This runnable jar file "solves" the maze. For each cell in the maze, it computes the first step of the shortest way to get to the destination cell. It writes this information about each cell in a file. To generate this information on the above produced maze-40x40.txt maze into a file "maze-40x40-paths.txt", this program can be run from the command prompt as follows:

```
java –jar PathFinder.jar maze-40x40.txt
```

The path file itself is simple to understand: it contains a cell (row, column) and then the next cell (row, column) that is on the shortest path from the first cell to the destination cell specified in the maze file.

MazeDrawToFile.jar

This is a runnable jar file that draws a maze and exports it as a jpeg image. To draw the above maze maze-10x10.txt into a file called "maze-10x10.jpg", this program can be run from the command prompt as follows:

```
java –jar MazeDrawToFile.jar maze-10x10.txt
```

Thus, to generate everything for a maze of size 40x40, these programs produce three files: maze-40x40.txt, maze-40x40-paths.txt and maze-40x40.jpg.

Note: the maze is drawn top-down, i.e. the first row of the maze is the topmost one. Therefore, "previous row" means the row above, and "next row" means the row below (read the description for the maze file above).

What to do:

Part 1

Write a C++ program that reads in this maze file, store it suitably, and then draw it as shown above (without the circles for start and end). The drawing should take up the entire window. The maze should retain its shape (aspect ratio) if the window is resized, and should be the biggest so that it tightly fits in the window.

You may choose any suitable size for the maze (i.e. your maze cell size could be 1, 10 or 10000), so long as it is visible as specified above.

You will be better off designing your program well, breaking it down into classes. You will extend this code in future assignments.

Part 2

Using the mouse, you should be able to draw a rectangle on the screen (press, drag, draw from upper left to lower right). The rectangle should be visible as the mouse is being dragged.

Upon releasing, the program should modify the maze in the following way:

1. Determine the rectangular region of cells in the maze that the rectangle overlaps with.

2. Add walls around these cells so that they are barricaded off.
3. Remove all walls from inside this rectangular region.
4. At random, remove exactly two walls so that there is exactly one way in and one way out of this rectangular region in the maze.
5. Save this maze in a file. If this file is given as input to the MazeDrawToFile.jar program above, it should print the correct maze.
6. Refresh the screen to show the modified maze.

What/How to submit: Submit the entire Visual solution project. Please clean the project and delete the database file in the project folder (it will be huge in size) before submitting! Make sure that your source file(s) are suitably commented.