

# CS335 Programming Problem 3 - “Maze”

## Graphics, Multimedia and User-Interface Design

### Due Thursday, November 1

## 1. Introduction

This project requires you to implement a Java stand-alone application that generates and solves mazes. You will provide visual display and control of the maze parameters via a graphical user interface.

## 2. Features

The program must run in two primary modes: maze generation and computer-automated maze solving.

### 1. Maze Generation

Your program must generate the maze using a randomized depth-first search algorithm (there is a pointer to an explanation below). The user must be allowed to visualize the maze construction step based on user-selected parameters (resolution in rows and columns, speed). The result – the randomly generated maze – should be displayed for the computer to solve. It is not acceptable to store the solution path as the maze is generated for playback during the automated solving stage. Your solver must have no advance knowledge of the maze paths.

### 2. Automated Depth-First Search Solver

Your program should solve the maze it generates using a depth-first search algorithm and a stack data structure. The solution should be visualized as the solver runs, in a manner similar to the way the example visualizes the solver. The speed of the visualization of the solver’s path through the maze must be controllable by the user. As with the example, show cells that are on the active path in one color, backtracked cells (visited but determined to be dead ends) in a second color, and unvisited cells in the default color. In the example, yellow is the active path, black is unvisited, and grey is backtracked.

### 3. The Graphical User Interface

On the GUI you must provide the following functionality for the user:

Start/stop	Start (or stop) the automated/user-guided solvers
Speed	Change speed of the visualization of maze generation and solver
Rows/columns	Set maze resolution (10 by 10 to 50 by 50; rows/cols independent)
Generate	Generate a new maze based on current parameters
Percent visited	Display a running percentage of number of cells visited

The specifics of the design of these elements are your responsibility. The arrangement and usability of these controls will be worth 25% of the grade you receive for your solution.

## 3. Algorithmic Details

### Maze generation

The tutorial at

<http://www.mazeworks.com/mazegen/mazetut/index.htm>

describes the randomized depth-first maze generation algorithm. Use this algorithm to generate mazes given a grid size. Key requirement: always place the start point in the upper left cell, and the end point (exit) in the lower right cell.

## Maze solving

Your automated solver should use depth-first search and the right (or left) wall rule. For example, follow the right wall until you hit a dead end (no exit, all adjacent cells visited). Mark cells as “visited” as you move. When a dead-end is reached, *backtrack* to the nearest place where there is another path to take that is available. Follow it until it dead-ends, etc. This algorithm is not particularly efficient, but is guaranteed to find a solution if one exists.

Note that because of the backtracking involved you will need to store path information in a data structure as the maze is traversed.

## 4. Example

An implemented example in Java found on the Internet is available at

<http://www.mazeworks.com/mazegen/index.htm>

This example is purely for your visualization and learning. You must implement a solution yourself. You may not use this code in your solution! Note that the code is structured as an applet, not a stand-alone application.

## 5. What to Turn In

Use the on-line “turn in” page at the class webpage to submit your Java code and all icons/files necessary to compile and run your code under the Java 2 SDK platform from Sun. In addition, provide an external documentation file “maze.txt” which is a text file that tells how to compile and run your program and gives a detailed summary of the program design (internal data structures, major algorithms, etc).

