Project 1

CS 320

75 points, due Feb. 16, 2005

A maze can be looked at as an array of $m \times n$ square cells, where m is the number of rows and n is the number of columns in the maze. Each cell has the property that one to three of its sides (or walls) is present. If two walls are present there is a path into the cell and a path leaving the cell. If three walls are present, the cell is a dead end. It is also possible to have a cell with a single wall. (Note: a maze can be represented by a graph in which each node represents a cell and each branch a connection between cells, i.e., the absence of a wall between adjacent cells.)

You are to use OpenGL to construct and display a maze where m and n are command line inputs to the program. The particular type of maze you are to construct is one in which every cell is connected to every other cell and there is exactly one simple path between any pair of cells.

One method to construct such a maze is to start with an array of cells, each of which has all four walls present. We then remove various walls to construct the desired maze. Consider the following algorithm. Initially all cells are said to be unvisited. We start by selecting (visiting) a random cell. We connect this cell to one of its neighbors by randomly visiting one of its unvisited neighbors recursively. We remove the wall between the two cells to connect them. If any of the other three neighbors has not been visited, we recursively visit them. If we reach a dead end (a cell all of whose neighbors has been visited), we return. When all cells are visited, the first visit call returns and we have a maze. (Note that the algorithm requires a data structure to keep track of which cells have been visited and which have not been visited. This will be discussed in class.) We can then remove two pieces of the outside wall to create an entrance and an exit to the maze. You are also to show a path in another color from between the entrance and exit. This is another simple, recursive algorithm.

The graphics part of this project is very simple: The overall size of the maze should be made slightly smaller than the size of the rendering window. Divide the dimensions of the maze by the number of rows and columns to determine the size of a cell. Begin drawing lines. You can do most of the work before using any graphics at all.

The following will be taken into consideration in evaluating your work:

1. The design and generality of your solution.
2. Completeness of testing. (Of course, you need to provide evidence of testing — you will lose points for not documenting this.)
3. The readability of your code.

Submitting Your Project

We will follow this procedure for all remaining projects this semester. Your solution is to be e-mailed to me at kelliher@goucher.edu (not bluebird). All project files should be sent as attachments in a single e-mail. You may collect all the files into a single ZIP archive. You should send all files necessary for me to build your program from source (generally, this is all .h and .c files), as well as any documentation and test files. You should send an ASCII file, named README.txt, describing the rest of the attached files. I will build your program from source and run it for myself. Your project is due at the beginning of class on the 16th. My standard late penalty (10%/three days) will apply.