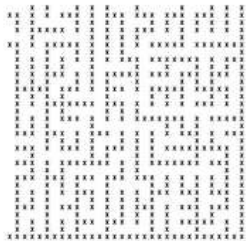


Overview

In this assignment you will implement code to randomly generate a maze. The maze your code produces is a 32 x 32 grid stored in a flattened bit array where each cell is a one bit value of 0 - blocked off wall or 1 - passage. When a cell is a passage then the four neighboring cells to its north, south, east and west will be walls. The algorithm used is a backtracking random walk that starts by opening the cell in the center of the grid and then at every iteration the algorithm picks a random neighbor of the current open cell and sets it to 1 as well as the wall cell between the two cells. The cells processed will be stored in a collection data structure provided for you for this purpose. Once the generation of the maze is complete, you will print the maze using ASCII symbols as follows, if the cell is a wall then you will display " X" (note the extra space before X) and if the cell is open then you will display two empty spaces " ". Here is a sample of a generated maze using the algorithm described below and you can verify that you can reach any open cell from any other open cell:



Starter Code:

The following zip file ([HW2-starter.zip](#)) includes the following:

- main.cpp - your code goes here
- Collection.cpp, Collection.h - Code provided for you to help you write the makeMaze function (see below).

Representation:

- The grid is a 32 x 32 bit array where each cell is one bit
- You will use `uint64_t` to store the cells so each `uint64_t` will store two rows of the grid. Therefore you will need 16 `uint64_t` integers for the grid stored in a 1-dimensional array.
- The array belongs to a struct that also owns two `int` variables storing the width (columns) and height (rows) of the grid.
- For each cell/bit the value 0 is blocked or wall while the value 1 is an open room.

Printing the maze:

- You will print the maze in ASCII characters based on the value of the cell/bit.
- If the bit is 0 then you print "X" (a space and X)
- If the bit is 1 then you print " " (two spaces)
- The extra space is to increase the width of the maze so it is closer to a square.