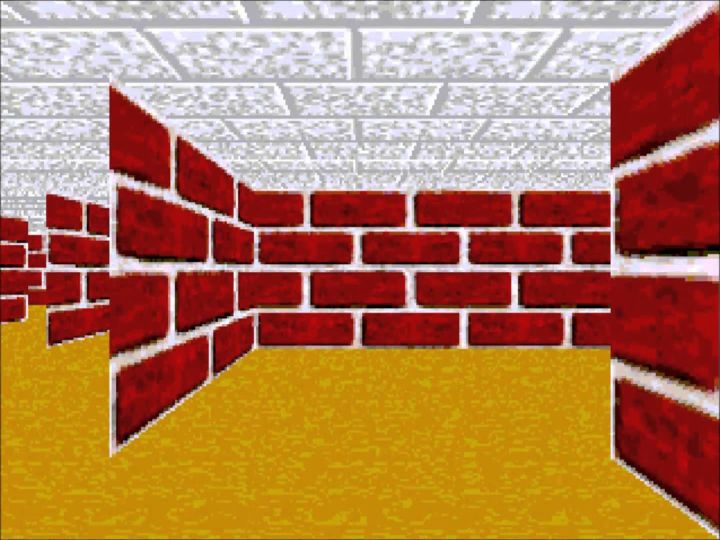
Maze Generator

Minotaurs apply within

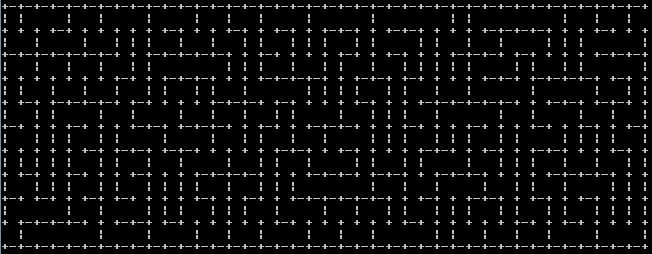


So despite the console output above, I had just started working with 3D graphics in OpenGL. I did several of the tutorials at <http://www.opengl-tutorial.org>. Most of it was pretty straight forward, most likely due to my understanding of 3D from working in the Visual Effects industry. The most difficult part of it all was trying to get the damned thing to compile, link, and run within Visual Studio. I had only used Visual Studio once before to write a Windows 8 calculator app, and had never adjusted environment variables and compiler directives before. It was a frustrating mess. But! Eventually I got my dependencies compiled and linked, and could move forward.

At one point in the year a friend of mine had nostalgically mentioned that old windows screensaver that involved the computer walking through a maze. You remember it, right?

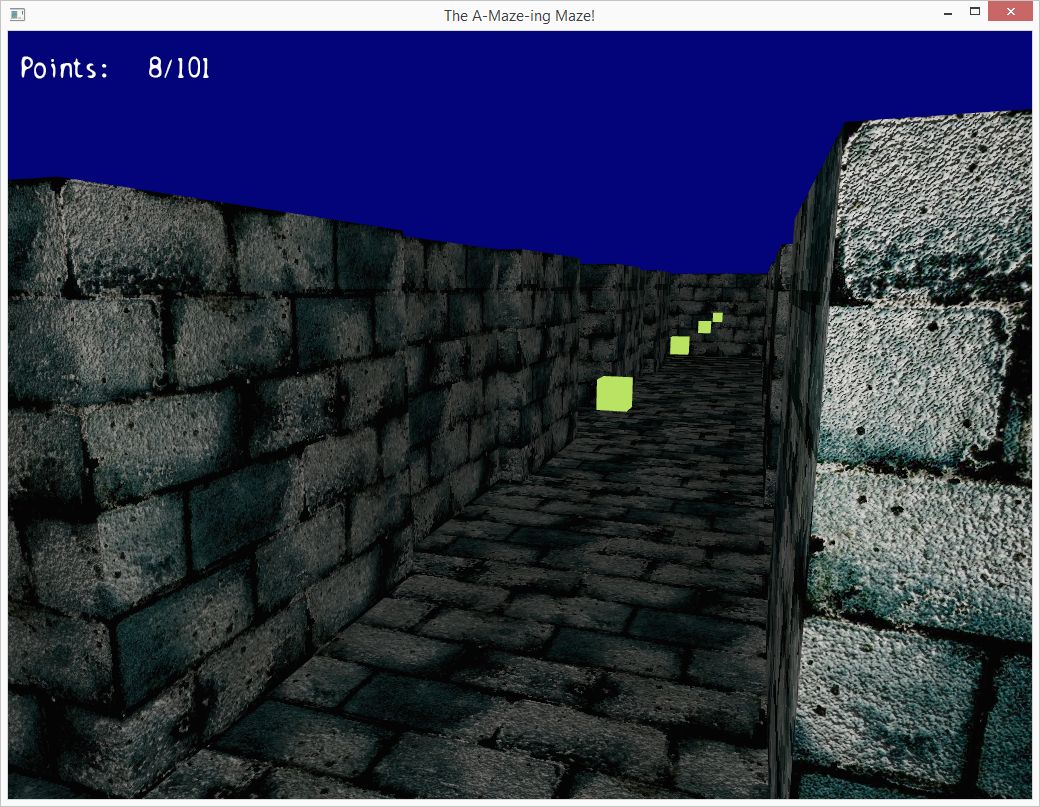


I realized that I could build an entire maze game with the cube I had been playing with. Rather than struggle with Visual Studio *more*, I went back to my usual text editor and command line compiler, and thought about generating mazes. I had read Mike Bostock’s blog article on maze generation (<http://bost.ocks.org/mike/algorithms/#maze-generation>) a while ago and put together a maze algorithm of my own.



My algorithm ended up being a simple depth-first random-walk. Exploring a maze in 3d is already confusing and difficult enough to navigate, without the trouble of any loops or complex structure.

With that done, I took the code into visual studio and used it to direct the placement of walls, flooring, and Pacman-esque trail markers.



I can make no guarantees that this will run on anyone’s particular machine. I still don’t know entirely what was going on when I was linking to libraries. The code can be read though.

Download program here

Download Visual Studio Project here