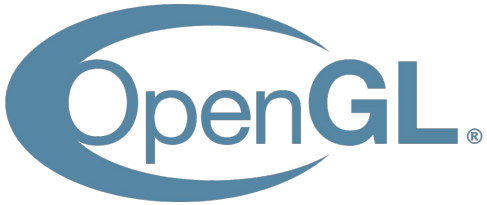
Demise

3D Game Engine and Game

Andrew Weller and Trey Nevitt

The Story

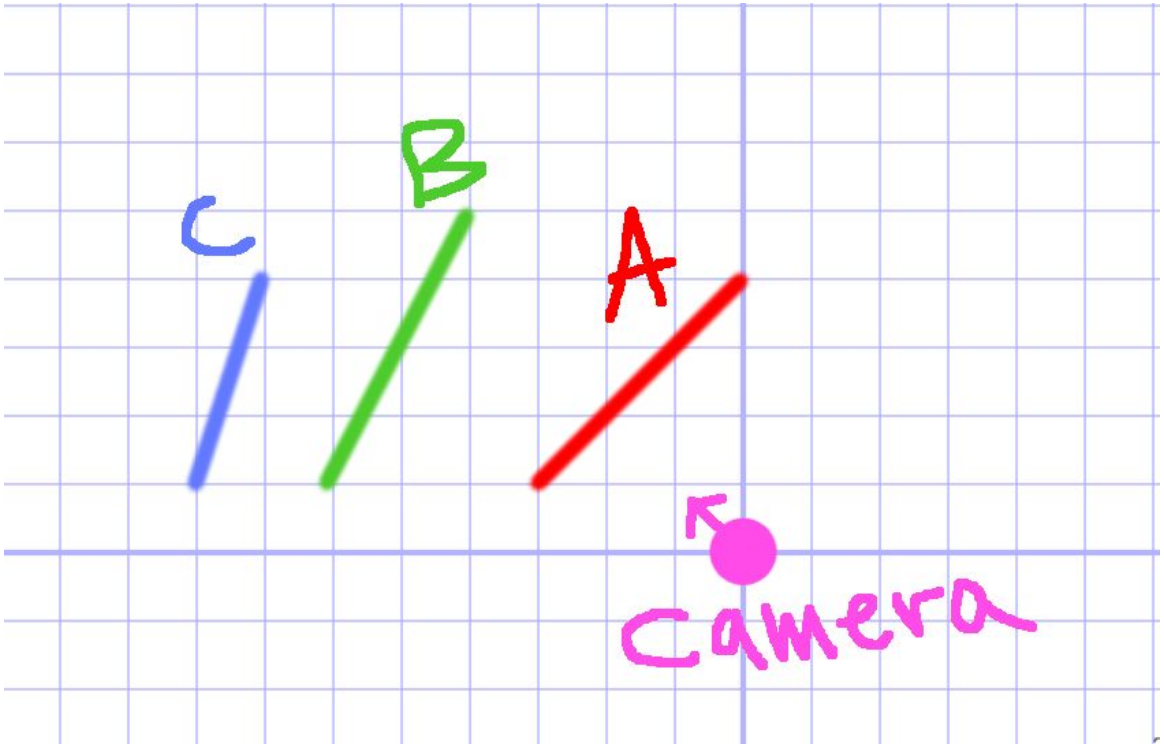
Since the era of NES games to today's Triple-A titles, many developers have created game engines that enable the player to explore a vast environment. Though open-source libraries like OpenGL and DirectX, combined with game engines like Unity and Unreal Engine, are designed to make 3D development easier for game developers, exactly how are these engines made in the first place?



The Problem

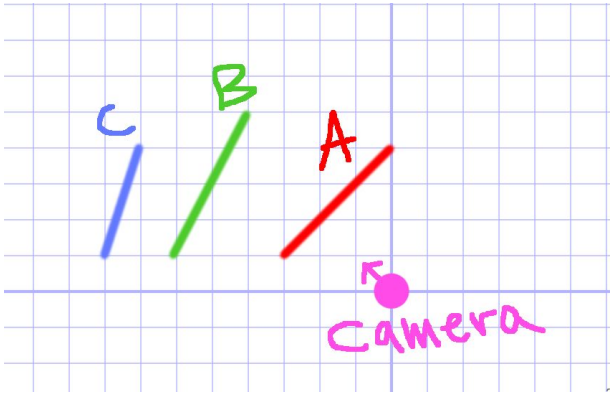
In order to convey a feeling of depth and perspective through a 2D medium—in our case, the computer screen—we need to render polygons that become our walls. But how exactly does the program know what angle the walls should be at? And how does it know which wall should be in front of another? On top of all that, the program should also be able to render efficiently.

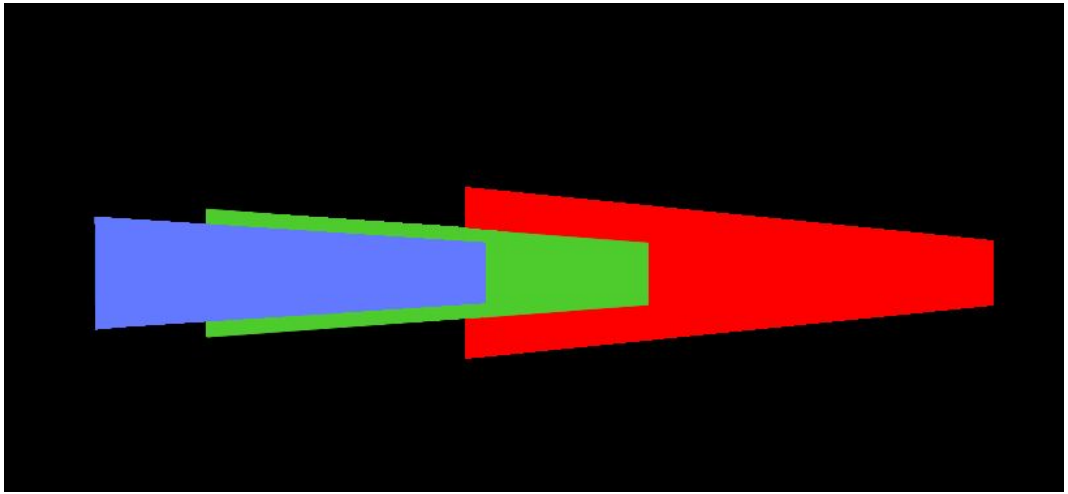
A Simple Map Layout



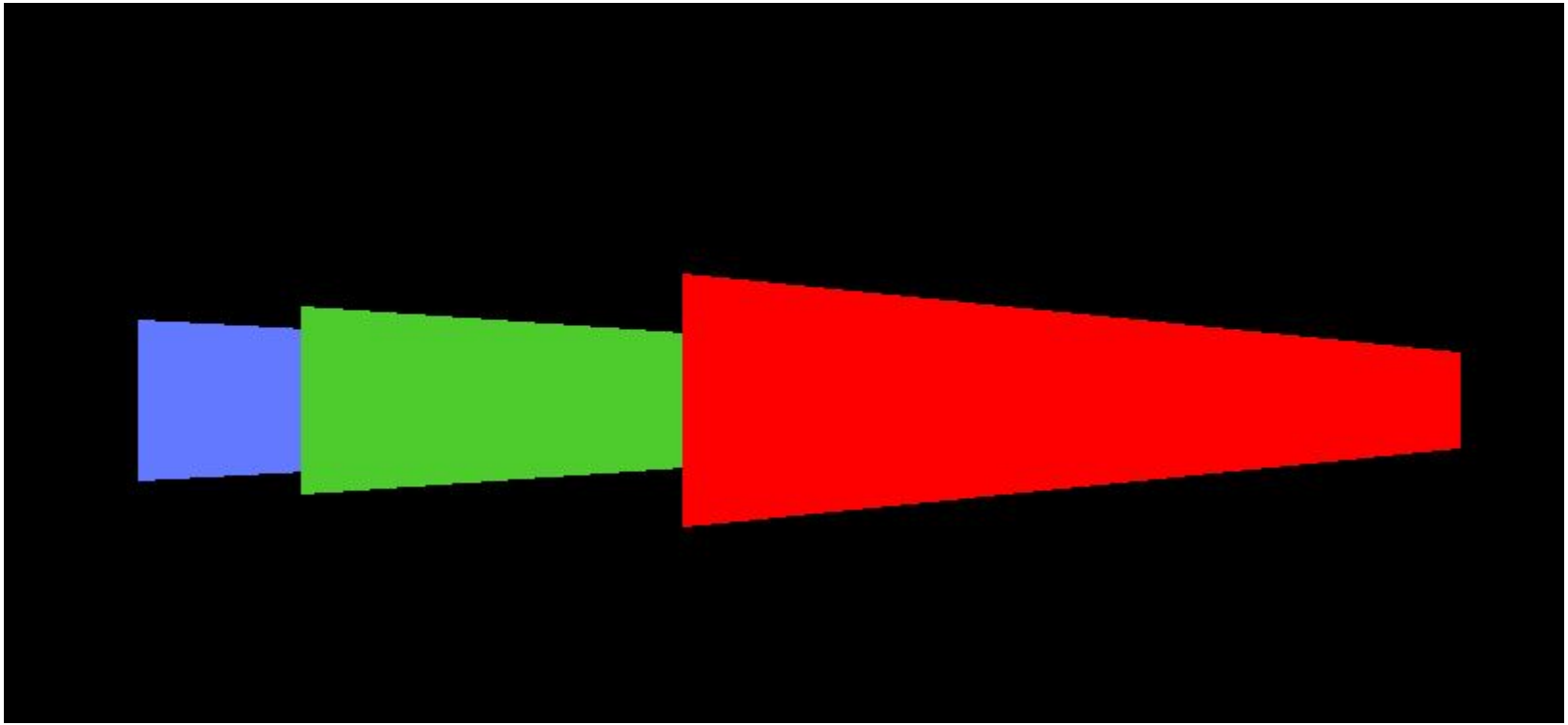
Incorrect Render Order

Currently, if we render the polygons in their alphabetical order (A → B → C), then the image below is what would occur, which is not what we want if we want a proper 3D environment.





Correct Render Order

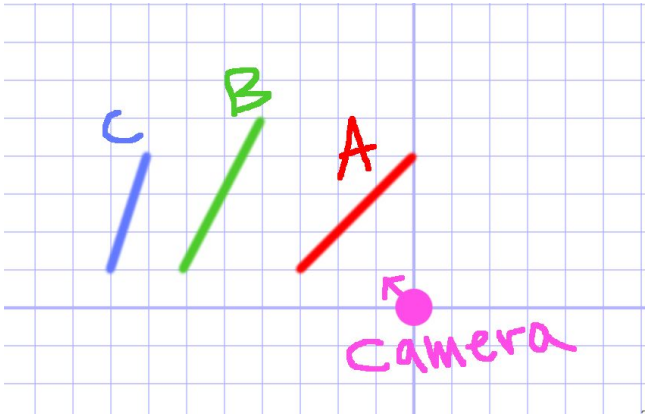


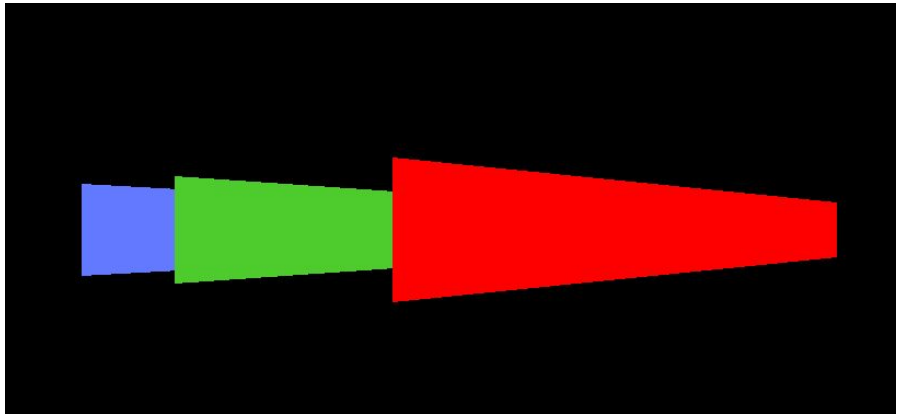
How to achieve correct order?

Projected Polygons are drawn on top of eachother

This means that polygons in the very back must be drawn first, followed by polygons in front of that.

So, the correct draw order is C → B → A, which results in this





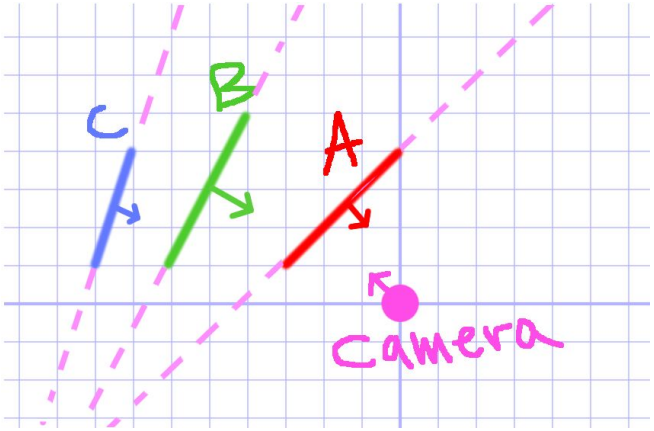
The Binary Space Partition

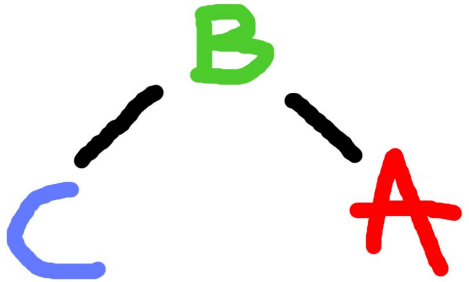
A lot like a BST.

Walls are organized in a tree structure, with nodes that contain a “wall” inside of the level.

All nodes to the right of the root are in front of it, all nodes to the left of the root are behind it.

*Front or back is arbitrary



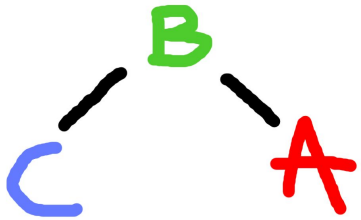


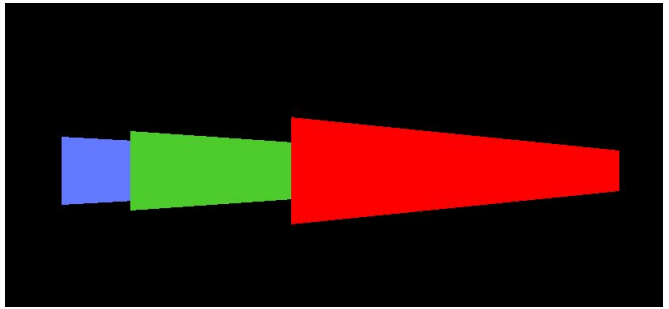
Traversal

Start at root

If the camera is in front of the root, render all the walls behind this node, then render this node, then render the walls in front of this node & vice-versa. This is done recursively on the whole tree.

This is a very simple, efficient, in-order traversal. It allows us to render the level geometry in the correct order in O(n) time. Incredible!





The Game

- Puzzle game?
- First person shooter?
- Lots of possibilities.

End