Matthew Gillatt
CS464
11/05/2015

Final Project Proposal

My final project is a solo project. I would like to implement the suggested idea “DOOM-like game using BSP Trees.” I am big fan of classic video games, and I have some hobbyist-level experience programming them. I even made a “hello world”-level map for the original DOOM! That was focused more on editing features provided by a source port though, instead of making map geometry or understanding the source code (which has been made free, but I haven't tried looking at it).

My preliminary research in BSP Trees seemed somewhat straightforward. Most references cite DOOM as an example. I believe I understand how a BSP Tree is constructed and traversed. However, most examples use “Painter's Algorithm”, which draws all polygons in order from farthest to closest. It's a great demonstration on how simple it is to calculate the order polygons are from the viewpoint, but I'm certain this project intends for a more efficient algorithm in the style of DOOM, in which polygons are drawn closest to farthest, ignoring pixels already drawn to the screen. This may be a late goal, as Painter's Algorithm can be used to test the BSP Tree creation and drawing, it's possible the efficient drawing algorithm may not be implemented in the final version, but it is a goal nonetheless.

Applying this to WebGL sounds like much of the drawing code must be handled manually, instead of simply drawing every polygon in an array. I'll need to construct a map to traverse, probably in a separate HTML file in case I want to try other maps. I'll need to build BSP Trees in WebGL, and traverse it in the drawing code. To prevent overdraw, it seems like the fragment shader should keep track of what pixels have been drawn this step, perhaps using a 2D array, leaving the fragment alone if it has been drawn. I'd also like to use mouse and keyboard controls to move around the map, and use the map geometry to do things like stopping before walls, and walking up stairs. Extra features that make it into a playable game, or make it look nicer are optional.

Matthew Gillatt
CS464
11/05/2015

A list of features, ordered by most important first:

- Map geometry from separate HTML files.
- Create a BSP Tree from the map geometry.
- Draw geometry using BSP Tree and Painter's Algorithm.
- Move through map with keyboard and mouse controls.
- Draw Heads-Up Display on the screen. Can display test variables.
- Draw geometry using BSP Tree from closest to farthest without overdraw.
- Draw 2D Things in the map, such as monsters and collectible items.
- 2D Things using different sprites when facing different angles, like DOOM monsters.
- Movable Things, such as destroyable monsters or projectiles created by the player.
- Sound effects and/or music.
- DOOM-style lighting, which depends on the lighting defined per room, and distance from the player.
- Original graphics, instead of ripping them from DOOM. (Yeah, I've made original pixel graphics!)