Jesse Young

4 March 2017

Final Project

CMSC 405-6381

Design -- Classes, Variables, and Methods

- blocks[] - array of types of blocks, containing a static draw index as well as the width, height and length of block.
- objects[] - contains regular shaped blocks, for easy binding.
- MatrixStack - provides ability to push/pop, rotate, scale, and translate matrices.
- MouseWheelHandler(e) - sets translation in z-direction based on mouse scroll.
- toggleScroll() - enable/disable z-translation based on mouse scroll.
- box(width, height, length) - returns a box with given dimensions, in same format as basic-object-models-IFS.js
- drawTexturedPrimitive(primitiveType, texCoords, vertices, drawMethod) - draw a primitive with the current texture.
- drawTextureRectangle(width, height, texNum, stretch) - draws a rectangle using texture number texNum. If stretch is true it will stretch the texture to fit the surface.
- drawBox(width, height, length, color) - draws a box with color, without lighting.
- drawSpecularObject(object, diffuseColor) - draws the given object with the given diffuse color, using an element vertex buffer.
- move(cols, rows) - translates the current matrix by peg size (a peg is the 'knob' on a block and surrounding space.
- elevateBy(height) - translates current matrix in y-direction.
- drawPegs(onBlock, color) - given an object, will draw a grid of pegs on top of that object in given color.
- drawBlock(blockNum, color) - uses blockNum as an index to object[] and draws that block (with pegs at the top) at the current location.
- drawE(color), drawLadder(color), drawFlower(color), drawDoor(color), drawWindow(color), drawBlockCircle(color), drawBlockHC(color), drawBlockArch(color), drawDog(), drawFence() - these draw all the special kinds of blocks by series of translations, rotations, and scales. They all return the currentMatrix back to normal once complete.
- toggleDoor(), toggleWindow(), toggleLadder(), toggleExplode() - alternates animations for movable objects in the scene.
- handleExplode() - gradually changes the height of space in between layers of blocks. (my favorite)
- draw() - sets up the model view matrix and then draws the scene.
- loadTexture(url, textureUnit, textureObj) - loads an image into a texture slot on GPU.
- installModel(modelData) - binds given object to an array buffer and sends vertex and vertex normals to GPU (uses STATIC_DRAW)
- initGL() - handle basic initialization of WebGL, including three shader programs.

- init() - initializes the canvas and adds all event listeners.
- doKey(evt) - handles keyboard responses
- frame() - enables animation by increasing a frame number and requesting a new frame.

User's Guide

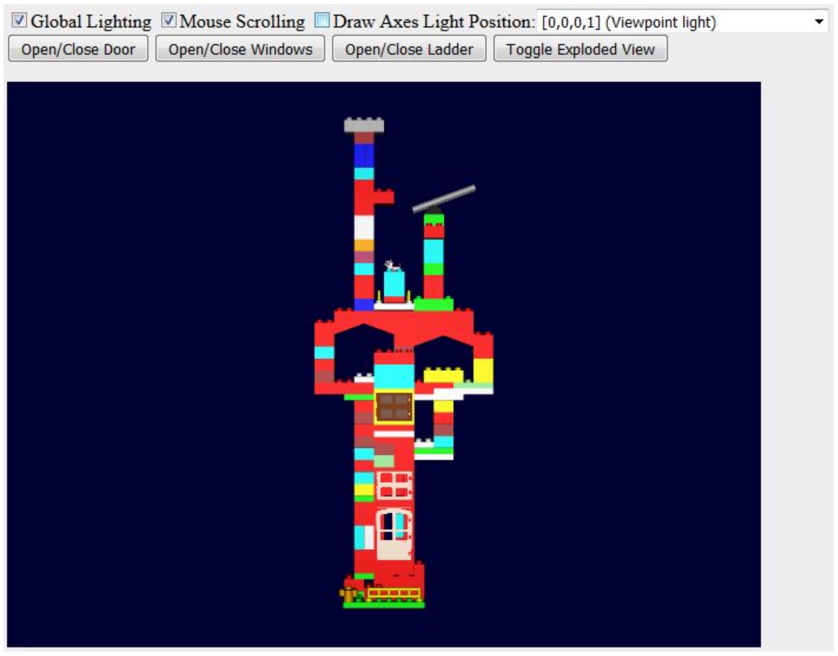
--How To Run The Project

- Open "public_html\index.html" with a web browser.

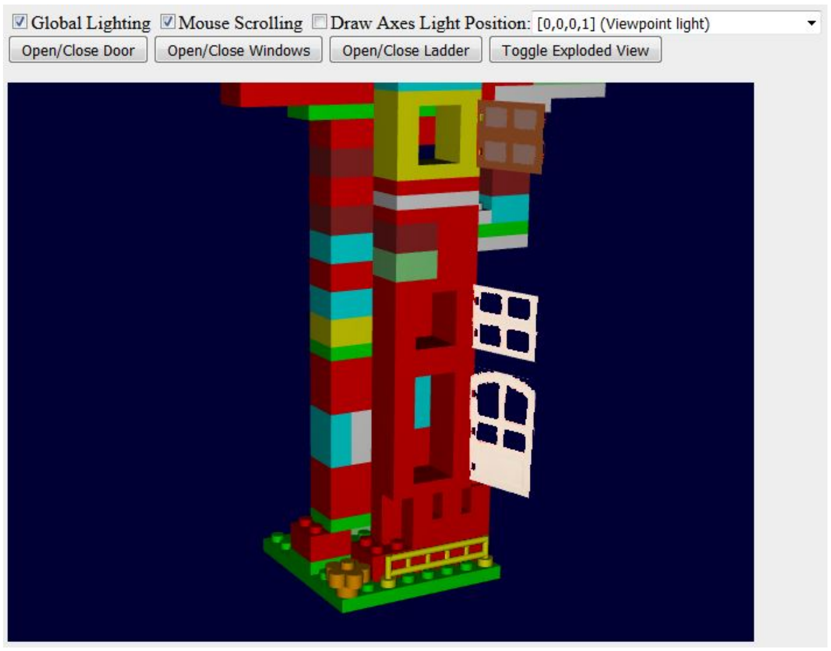
--Special Features

- Enable/disable lighting
- Enable/disable mouse scrolling
- Open/close door and windows
- Exploded view

--Screen Shots

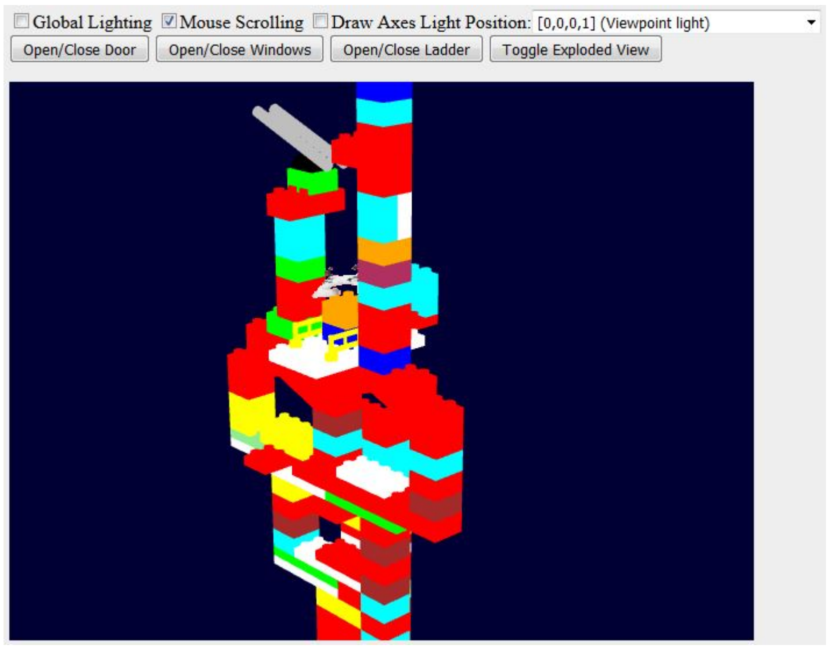

☑ Global Lighting ☑ Mouse Scrolling ☐ Draw Axes Light Position: [0,0,0,1] (Viewpoint light)

(default view)



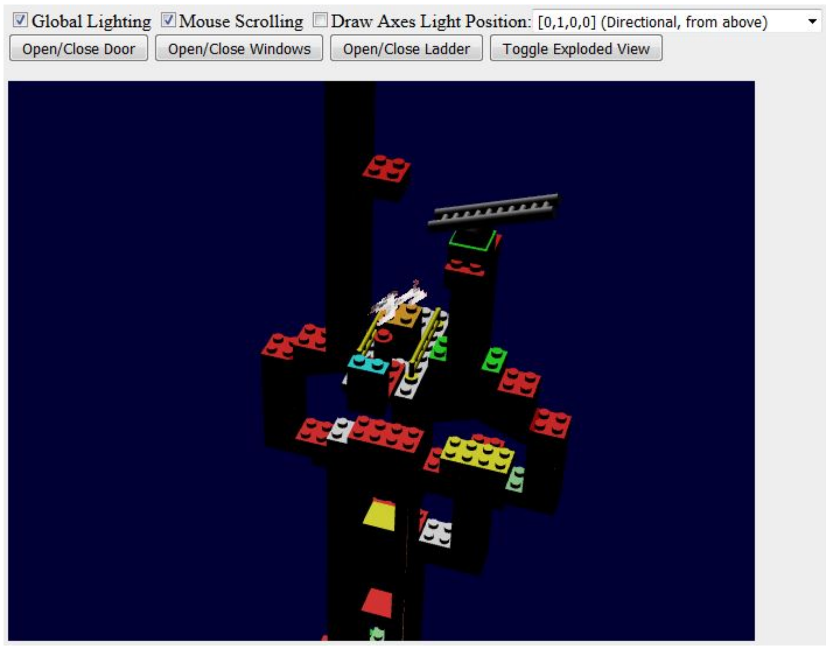
☑ Global Lighting ☑ Mouse Scrolling ☐ Draw Axes Light Position: [0,0,0,1] (Viewpoint light)
Open/Close Door
Open/Close Windows
Open/Close Ladder
Toggle Exploded View

(doors and windows open)

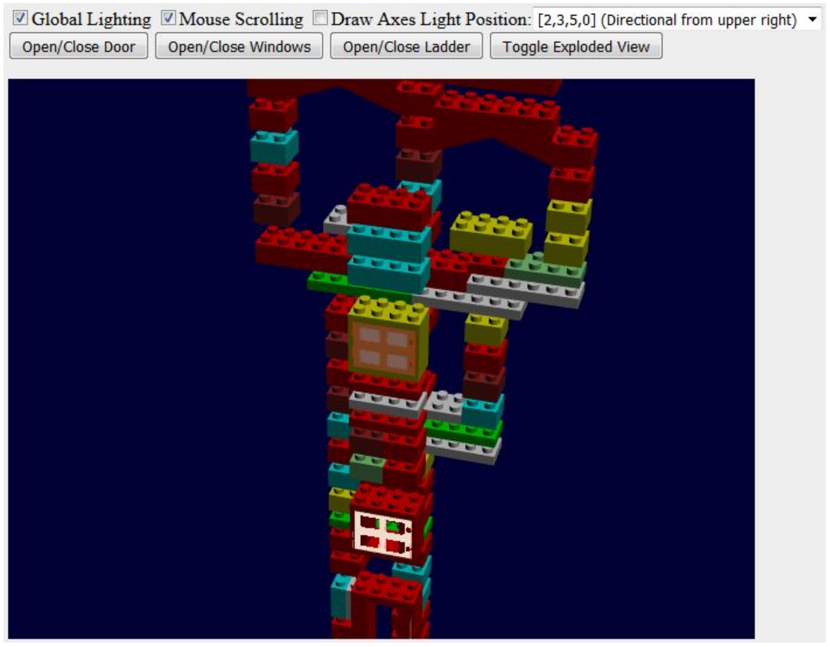


☐ Global Lighting ☑ Mouse Scrolling ☐ Draw Axes Light Position: [0,0,0,1] (Viewpoint light)
Open/Close Door
Open/Close Windows
Open/Close Ladder
Toggle Exploded View

(global lighting disabled)



(light from above)



(exploded view)

Test Plan

--Test Package

- The test package is located in the "Young_Jesse_Final_Project \test" folder.

--Expectations

- Draw a 3D tower of LEGOs, designed by my son. Should allow the user basic interaction and use different light sources as well as some extra functionality.

Lessons Learned

--Memories of Work

This was one of my favorite projects. First of all I have to mention that the inspiration for this project came from my son, Mason. He built a LEGO tower on the living room floor and I just had to try to create it myself. I included a picture of his creation below the references section.

I have to admit I was a little lost for a while on this one too. The implementation of the vertex and fragment shaders had me pulling out my hair. I couldn't figure out how to use more than one shader so I scoured through the example code and eventually found some good starting points.

I had a difficult time implementing the element array buffer as well, the example that I started from only had one shader program so it did not explain how to use multiple shader programs with element array buffers. Also, I didn't understand why there was a single element array buffer. It took a lot of tinkering but eventually I had a breakthrough and things made a lot more sense. It's funny how much easier it is to read code than the textbook!

References

University of Maryland University College. 2017. Module 7.1.5 Rotation by Mouse. [Online Course Material]. Retrieved from: http://polaris.umuc.edu/~jroberts/CMSC405/c7/s1.html

University of Maryland University College. 2017. Module 7.2.2 Specular Reflection and Phong Shading. [Online Course Material]. Retrieved from: http://polaris.umuc.edu/~jroberts/CMSC405/c7/s2.html

University of Maryland University College. 2017. Module 7.4.4 Dynamic Cubemap Textures. [Online Course Material]. Retrieved from: http://polaris.umuc.edu/~jroberts/CMSC405/c7/s4.html

The Original LEGO Tower, by Mason Young