**Justify development choices for your 3D scene. Think about why you chose your selected objects. Also consider how you were able to program for the required functionality.**

The choice for my scene was simply to choose something that I care about. It seemed natural to go with my wall of guitars and amplifiers. Of course, since it would take forever to model every last one of them I simplified it down to just my favourite guitar as well as my favourite amp- my schecter apocalypse and peavey 6505. The process was surprisingly straightforwards, and it reminded me a lot of when I first started game development. At the time, I didn’t know how to use blender or do any 3d modelling (and I still don’t, I just ended up finding a 3d artist), so I made a bunch of things with just primitive shapes in Unity scenes. Programming the required functionality was easy enough when I got to stick to the guide, though I suspect that there is a lot of carryover and improvement to be made with future endeavors in openGL. I think the first thing I’d do is make it so that I could run the scene and move an object around in the scene though- so that you don’t have to repeatedly use the shortcuts F5 -> alt+f4 -> alt+f4 over and over to adjust things within the scene.

**Explain how a user can navigate your 3D scene. Explain how you set up to control the virtual camera for your 3D scene using different input devices.**

The tools for navigation include left, right, forwards, backwards, upwards, downwards movement, 6 degrees of freedom, as well as full control over viewing angle. In addition to this, there is control over the speed of movement, and control over the view perspective- either orthographic of standard perspective. It’s set up like any other editor would be, wasd for movement, mouse for viewing angle, and an additional q/e for up and down, with o and p swapping between perspectives.

**Explain the custom functions in your program that you are using to make your code more modular and organized. Ask yourself, what does the function you developed do and how is it reusable?**

Almost all of the functions made in this program are reusable- they’re designed to be easily digestible and understandable. Each of them simplifies the process of creating and reusing objects, textures, lights, textures, and materials, by creating accessible, readable methods to input parameters for. In addition, we are provided with a shape library, so as to not have to define 100000 vertices. Combined, these make for incredibly reusable, modular code, that simplifies the arduous process of placing objects, lights, textures, etc, within a scene. This greatly diminishes the number of aforementioned F5 -> alt+f4 -> alt+f4 loops. These functions all come together under what pretty much boils down to a single method- the setupScene method, and are constructed as pillars to sort of support it. The other header and source files are supplementary to it, and almost all of the action takes place in the same spot. It’s very easy to follow and work within.