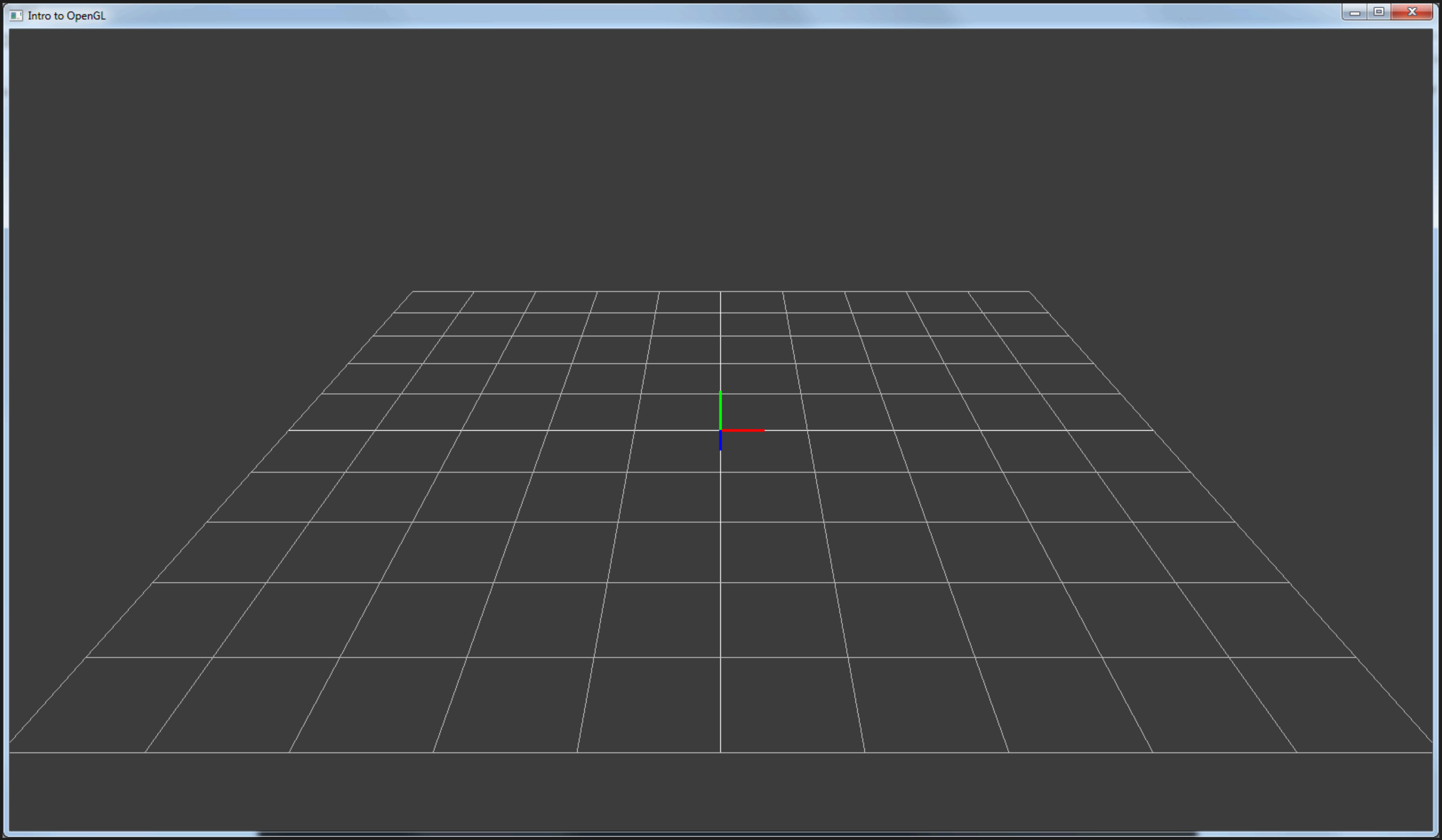
Intro to OpenGL

This part of the assessment focused on setting up the project and getting a window created and displayed to the user in OpenGL (Open Graphics Library). When using OpenGL, you normally use what is called a wrangler to make getting input and setting up everything related to the window easier. This project was no exception and used GLFW (OpenGL Framebuffer Wrangler).

The way the application is setup is through a 'BaseApplication' class. The base application has an update loop, and a function to create the context as a baseline for all applications. When you inherit from it you must define a startup, draw, and shutdown behavior since these functions are pure virtual functions from the parent class. Because of this it is impossible to create an instance of 'BaseApplication'. Instead, you can use a pointer which points to a 'BaseApplication' class type and have it point to a child class on the heap instead. This allows you to call all of the virtual functions of 'BaseApplication' but keep those calls generic to any child class you decide to create and point to.

This stage in the graphics assessment has taught me about pure virtual functions which are used to create a class which declares functions, but does not define them. This is useful for creating a base object which is used to provide an interface to function calls, but allows separate definitions of the function per child.



Once finished, the output looks something similar to this. A grid is drawn as well as a transform gizmo centered at the origin.

In the original project the ‘Gizmos’ class was created by AIE and was used to draw the grid and transform gizmo. In the screenshot and the compiled program my own gizmos class is used instead. The basic concept remains the same however, as this is a working example of how to create an OpenGL context using this type of setup. The program was checked for memory leaks while running the update loop and none were found.