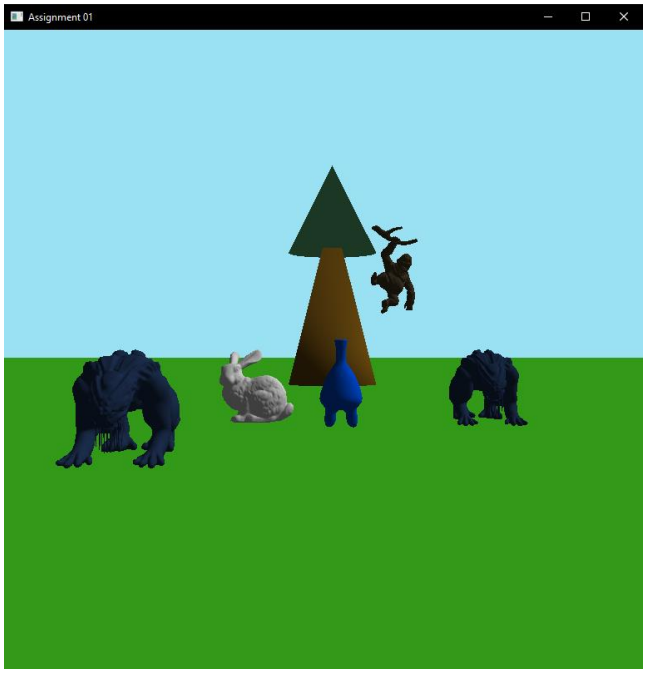
Assignment 01

Farees Siddiqui 100-780-513

February 20, 2023



This report will talk about my design decisions for the first assignment of this class. We are using OpenGL and GLFW for the graphics API as well as "tiny_obj_loader" to load .obj files which are used to render custom objects. The final product looks like the following. Throughout the rest of this report I will be talking about different decisions I made to streamline the development of this program. The first major change I employed actually has nothing to do with the program itself, but to do with my development environment. I personally do not really like using Microsoft Visual Studio which is why I decided to use VSCode (Visual Studio Code) for this assignment. This came with a few challenges, the main one being having to build the libraries from source and having to compile and link them manually.

How to Build the Program

I decided to use Visual Studio Code for my assignment so the build process is slightly different, the following shows exactly how to run the program.

1- Change directory into the "bin" folder (cd Assignment01/bin)
2- Run the command `make` this will run the necessary compilation and linking
3- Run the program using the command ./Assignment01.exe

NOTE: the make commands must be run from the bin folder

The following screenshot shows the entire workflow.

```
PS C:\Users\faree\Documents\School\ThirdYear\CSCI3090U\Assignments> cd .\Assignment01\bin\
PS C:\Users\faree\Documents\School\ThirdYear\CSCI3090U\Assignments\Assignment01\bin> make
g++ -I../include -c ../src/*.cpp ../src/*.cc
g++ *.o -o Assignment01 -L../lib -lglfw3dll -lopengl32 -lgdi32 -lglew32 -lglu32 -lm
PS C:\Users\faree\Documents\School\ThirdYear\CSCI3090U\Assignments\Assignment01\bin> .\Assignment01.exe
```

Design Decisions

The first major design decision I had was to use a mesh to render all of my objects and shapes. Each object that is loaded from an object file is converted into a mesh and added to a Model

vector. Everything in the Model vector is draw. For the custom shapes, I have a function that sets up the vertex and index buffers for the shape, and adds that data to a mesh struct, and returns it. This was all off the rendering for everything in the scene could be done simply by looping over the Model vector and drawing every mesh inside it. I also used transformation matrices to make sure that all of the objects were facing the correct way and that. Another major design change that I made was that I decided to use Booleans to store the state of a key (Pressed or Released) this let me hold down a key to navigate the scene rather than having to register a new key press every time. This made the movement a lot more intuitive and made the program easier to use in general. The final thing I did was that I used a master object for one of my alien objects so that it could be copied meaning different instances of this master object could be made.