Cover Letter

Software Development (Computer Games)

3rd year Games Programming 2

Phillip Ross S1905606

*I confirm that the code contained in this file (other than that provided or authorised) is all my own work and has not been submitted elsewhere in fulfilment of this or any other award*.

*Phillip Ross*

Contents

Contents

[Files 3](#_Toc92987937)

[Camera.h 3](#_Toc92987938)

[Mesh.h 3](#_Toc92987939)

[Mesh.cpp 4](#_Toc92987940)

[Obj\_loader.h/obj\_loader.cpp 4](#_Toc92987941)

[Screen.h 4](#_Toc92987942)

[Screen.cpp 4](#_Toc92987943)

[Shader.h 5](#_Toc92987944)

[Shader.cpp 5](#_Toc92987945)

[Sound.h 6](#_Toc92987946)

[Sound.cpp 6](#_Toc92987947)

[Texture.h 7](#_Toc92987948)

[Texture.cpp 7](#_Toc92987949)

[Transform.h 8](#_Toc92987950)

[Game.h 9](#_Toc92987951)

[Game.cpp 9](#_Toc92987952)

[Main.cpp 10](#_Toc92987953)

# Files

## Camera.h

This file creates a camera as shown below.

A screenshot of a computer

Description automatically generated with medium confidence

In here we also find the position of the camera as well as the view projection matrix.

Text

Description automatically generated

There are also some function and variables for movement that could be used if the user wanted to move the camera within the scene.

## Mesh.h

Text

Description automatically generatedThis file is used in order to create a sphere which becomes the mesh that we load models and textures onto. We also track the position and rotation of the mesh so that we can move and manipulate it in the scene.

## Mesh.cpp

Here we draw the vertices for the mesh as well as loading the model onto the mesh using buffers and vertex array objects. and updating our information on the mesh.

## Obj\_loader.h/obj\_loader.cpp

This file loads the .obj file of the model we want on our mesh. This was given to us in a previous lab by our lecturer for use.

## Screen.h

Text

Description automatically generatedThis file creates the screen/display for the user. In here we create functions which create the screen as well as edit it.

## Screen.cpp

Here we create our openGL context as well as setting up our display window. We set its size here and have functions to clear the screen or swap the buffers around.

## Shader.h

Text

Description automatically generatedIn shader.h we create functions to bind, update and create our shader.

## Shader.cpp

Text

Description automatically generatedIn here we can bind a shader to objects in our case we will bind the shader to our meshes. We also create our shader here and make sure there are no errors with it.

## Sound.h

This file creates function for us to add audio through music and sound effects, we also create two variables.

## Text Description automatically generatedSound.cpp

This is where we have all our audio related functions. These allows us to add tracks or sound

Text

Description automatically generatedeffects as well as play them.

## Texture.h

Text

Description automatically generatedIn this header file we have the constructor, destructor, and a bind function as well as a variable called texture handler.

## Texture.cpp

Text

Description automatically generatedIn this file we load in the texture then wrap and bind it on our mesh.

## Transform.h

This is the file we use to handle the position, rotation, and scale. It contains getters for the model and model view projection matrix.

Text

Description automatically generated

## Game.h

Text

Description automatically generatedThis file is where we culminate most of our program as it includes all the previous header files and uses a lot of their function. We create functions related to the screen, the models/meshes, audio and we create a function for collision detection. We have also created a function called drawGame() where we will begin to draw everything in our scene to the screen/display. We also create loop() which contains our game loop. There are a few more variables made as well, backgroundMusic is used to store Music that is used in the playSound() function. Whistle is similar but instead of storing music it just stores a sound effect whistle if we chose to use it. For my program we will only need the background music for now.

## Game.cpp

In the initialiseSystems() function we load in our models and music (if we were including sound effects that would also be done here). We also create our camera and load in our shader.

Text

Description automatically generated

Text

Description automatically generatedOur loop() function plays the background music from the position of mesh 1 as well as calling the collision detection between mesh 1 and mesh 2. I made the collision detection between mesh 1 and mesh 2 only so I could make sure it was working.

A picture containing text

Description automatically generatedThe collision() function handles the details of how the collision detection works by finding the distance between the centre of the two spheres.

Text

Description automatically generatedI also have a playSound() function which will play my chosen audio in this case my music but if I wanted to it can also play any sound effects

After there is just the drawGame() function which is where everything is put into the scene. So this is where the meshes are given their position, rotation and scale for the scene. They also have their shaders and textures given to them.

## Main.cpp

Text

Description automatically generatedThis is where the program begins and ends it leads into the game.h and .cpp files and runs the game.

Overall, this project creates an output in the form of a screen with models which are customised with collision, and have audio.