# COMP3015 GAME GRAPHICS PIPELINES - Initial Prototype

Contents

[COMP3015 GAME GRAPHICS PIPELINES - Initial Prototype 1](#_Toc66731392)

[Which version Visual Studio and Operating System you used to test/write the code locally? 1](#_Toc66731393)

[How does it work? 1](#_Toc66731394)

[How does your code fit together and how should a programmer navigate it? 1](#_Toc66731395)

[Anything else which will help us to understand how your prototype works. 1](#_Toc66731396)

[A Link to the unlisted YouTube Video 1](#_Toc66731397)

## Which version Visual Studio and Operating System you used to test/write the code locally?

Microsoft Visual C++ 2019

## How does it work?

The shader I implemented was the Blinn-Phong reflection shading model, the shader calculates the shininess the ambient reflectivity, diffuse reflectivity, and specular reflectivity of objects.

I have two ambient lights set up in the game.

I have made two Wavefront Object files but also use one of the models that was provided in one of the labs, the pig because it worked well in my scene so I kept it as a demonstration what heavy shininess and heavy ambient reflectivity, diffuse reflectivity, or specular reflectivity but nothing else does to an object by setting the objects to…

prog.setUniform("Material.Kd", 0.0f, 0.0f, 0.0f);

prog.setUniform("Material.Ks", 0.0f, 0.0f, 0.0f);

prog.setUniform("Material.Ka", 20.0f, 20.0f, 20.0f);

prog.setUniform("Material.Shininess", 180.0f);

prog.setUniform("Material.Kd", 0.0f, 0.0f, 0.0f);

prog.setUniform("Material.Ks", 20.0f, 20.0f, 20.0f);

prog.setUniform("Material.Ka", 0.0f, 0.0f, 0.0f);

prog.setUniform("Material.Shininess", 180.0f);

prog.setUniform("Material.Kd", 20.0f, 20.0f, 20.0f);

prog.setUniform("Material.Ks", 0.0f, 0.0f, 0.0f);

prog.setUniform("Material.Ka", 0.0f, 0.0f, 0.0f);

The other two objects made by me on blender are a truck and a box of chocolate these have their uniforms set to…

prog.setUniform("Material.Kd", 1.4f, 0.4f, 0.4f);

prog.setUniform("Material.Ks", 20.0f, 20.9f, 20.9f);

prog.setUniform("Material.Ka", 0.5f, 0.5f, 0.5f);

prog.setUniform("Material.Shininess", 1.0f);

for the box, and…

prog.setUniform("Material.Kd", 1.4f, 0.4f, 0.4f);

prog.setUniform("Material.Ks", 0.9f, 0.9f, 0.9f);

prog.setUniform("Material.Ka", 0.5f, 0.5f, 0.5f);

prog.setUniform("Material.Shininess", 10.0f);

for the truck.

The scene has a few options for lighting, this being location, ambient light intensity, and diffuse and specular light intensity. The calculations are done in the fragment shader but the specific numbers that will be put into the equation are random, this means that every time you load the project, the two light locations, ambient lights, and diffuse and specular light intensity will be random.

My program contains five models and one cube, the cube and two of the models have two textures put over them, one is a JPG and one is a PNG which allows for transparent backgrounds, something a JPG cannot do. By mixing the two image types you can allow the JPG to exist in the PNG’s transparency.

By using the rand() function I have managed to make all the lights in the scene random, whenever you load the project the lights will be different.

The information which will be used in the shader is declared in “scenebasic\_unifrom.cpp”.

The video linked at the bottom of this write up goes into more detail as to how the code works exactly.

## How does your code fit together and how should a programmer navigate it?

The most important files in the project are “basic\_uniform.frag” which is the fragment shader, “basic\_uniform.vert” which is the vertex shader, “scenebasic\_uniform.h” which is the header for the file where the majority of the implementation can be found, and “scenebasic\_uniform.cpp” which is the file where most of the implementation is, this is where the randomisation is done to calculate how mainly the fragment shader can calculate the lighting and materials. There are two textures that have been implemented, these are located in the texture file and are “ChocolatePackage” which is a png and “Chocolate” which is a jpg. There are two models I made, they can both be found in the media folder and are called “Bar” and “BarBox” which are Wavefront Object files. Those are the only files I have edited in the template.

## A Link to the unlisted YouTube Video

<https://youtu.be/i1LAvXPXSoo>

## References

AWOL, 2018. *Chocolate*. [image] Available at: <https://awol.junkee.com/wp-content/uploads/2018/02/12144792\_1069749433049779\_3972883000153605047\_n-1.jpg> [Accessed 16 March 2021].

Cadbury, n.d. *Cadbury Dairy Milk Chocolate Bar 200G*. [image] Available at: <https://digitalcontent.api.tesco.com/v2/media/ghs/d87fd7be-7caf-4cb5-b27c-8a33f8a589c5/snapshotimagehandler\_1842386856.jpeg?h=225&w=225> [Accessed 16 March 2021].