

# Intermediate Computer Graphics Midterm

## Group Members

Parag Gurung - 100780288

Ewan Chang - 100787343

Ryan Yee - 100785620

Prime number (11)

## Game Selection

We chose Super Mario Bros, where players take control of *Mario*. Bricks, Pipes, and Mystery blocks are a few of the obstacles that players will have to overcome in-order to get to the end of the level. The win condition is triggered when Mario successfully touches the flagpole, if mario falls, gets touched by an enemy, or runs out of time, the player loses.

## Explanation of Concepts

**Explain the limitations of not using the programmable stages of the graphics pipeline to enhance your game.**

The graphics pipeline allows your 3D game scene to be viewed in a 2D image. Each stage of the pipeline helps to enhance your game, and by not using the pipeline, your game will not look the best it can. The main limitation of not using the graphics pipeline is not having full control of the specific stage you are in as that can change the outcome of the final image on screen. This can hold back your game as you can't freely edit what the 3D scenes will be rendered to the 2D screen due to the fact you will not be going to each stage to personalize them to your liking. An example of this is trying to have a certain part of a shape in your game to be a certain gradient color with each color having different presences.

By not using the graphics pipeline, you aren't able to go in the specific stage to edit what the 3 colors will be used and the amount of presence of each one. Another limitation of not using the programmable stages of the graphics pipeline is not being able to relieve your main processor (CPU). This is because the programmable steps are moved into the graphics pipeline and GPU of

your computer. With having a lot of stress on the CPU, coding and rendering your game on your computer will be slowed down significantly.

### **Explain how the Phong lighting model allows you to create a glass feel for objects within the game.**

The Phong lighting model comprises three components: Ambient, Diffuse, and Specular. Ambient lighting allows an object to retain some of its color even when it is dark. The diffuse simulates the directional impact of a light source on an object, allowing for the side of the object that is facing the light source to become brighter; the specular lighting allows for a glossy look, creating a bright area on an object. Specular lighting is also based off of the direction at which the player looks at the object from, as well as the direction of the light. These three components of the Phong Lighting model allow for the simulation of a glass-like looking object.

### **Explain what approach allows you to create a horror feel using shaders.**

To create a horror feeling using shaders, the first step is creating a vertex and fragment shader. For the shader, we would approach this by having a cube map showing off a picture of a dark scary background like an old graveyard. Starting off in the vertex shader, here the vertices will be placed and be passed to the fragment shader. In the fragment shader the color correction will be applied and shown in the main code. In the main code, the jpegs will be plugged in by having 6 jpegs for having XYZ negative and XYZ positive to complete the cubemap with each jpg being a scary graveyard background.

## **Implementation Explanation**

### **Explain how to make scene elements which emit light upon interactions.**

One way to make it so scene elements emit light upon interactions is by attaching a light to the object. The light will have a trigger collider returning a boolean value that will either be true or false depending on whether the player's collider or another collider interacts with it. If the boolean value returns true, it can be set to turn on the light or do another effect like flashing or increasing the brightness, and if it returns false the light can be turned off. A separate class for the light will be created, where what happens when an object interacts with the trigger collider

will be. In the scene.cpp, that behavior will be attached to the object that has the interactable light attached to it.

**Explain how you implemented the shader for this Midterm and indicate why this choice was made.**

For the shader this midterm we used a cubemap of a mario background. This was implemented by creating a new template in photoshop for the free mario background to be imported in and saved as a XYZ pos or XYZ neg jpg file. After this was done, a folder was created in cubemaps called mario and was inputted in the texturecube line of code. We made this choice for the game because since we chose the mario assignment we wanted to enhance the theme by having a mario background. Whenever the player moved around we wanted them to make it seem that the player was in the mario world.

**Repo Link:** <https://github.com/MrKurg/IntermCGMidterm>

### **Resources**

<https://wallpapers.com/wallpapers/classic-super-mario-wallpaper-8k-ultra-hd-wallpaper-and-background-6w71s2i3g53zwku9.html> -(Cubemap Mario background)

<https://sketchfab.com/3d-models/mario-obj-c549d24b60f74d8f85c7a5cbd2f55d0f> <-(Mario Model)

<https://www.turbosquid.com/3d-models/free-suzanne-head-3d-model/678405> <-(Monkey Mesh)