CG Midterm: Written Portion

Team:

Shesh Gaur (100786399) - Programmer
Gareth Dawes (100776314) - Programmer
Aayush Shrestha (100780255) - Artist

Last Digits: 9 + 4 + 5 = 18

(even)

Game: Pac Man

You play as Pac Man. There is a ghost enemy that will be chasing you, if he hits you, you lose, and will respawn. Wall / Obstacles are placed in the scene. To win you must collect 5 pellets.

Explanation of Concepts:

Select the playable character and use it to explain the graphics pipeline stages associated with Vertex shader, Geometry Shader, and fragment shader.
Starting in the Vertex Shader, mesh data from the model of the character is loaded in. This includes, vertices, normals, etc. In the vertex shader, the parameters can be controlled to morph the character's mesh, or even apply animations. The geometry shader is an optional stage of the pipeline. It can be used to transform the base mesh, and it's changes persist to the next run of the pipeline. It can be used to apply a wide variety of effects to the player. Finally, we have the fragment shader. This stage determines the colour of each pixel on screen. For a player character, it's typical to use data like a UV-Mapped Texture to get the diffuse colour, and maybe a normal map to add extra detail or depth to the character's clothes.

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

The phong lighting model uses the following 3 parameters:
Ambient: This is the ambient base colour of the object. If this wasn't here, it would seem as though there was no naturally spreading light through-out the scene, which would feel unnatural.
Diffuse: This is the colour that appears "reflected" off the object. Using light calculations involving the normals on an object and the dot product, the degree to which each pixel is coloured is determined here.

Specular: This is the lighting component that reflects off the surface in a uniform and sharp way. With a high shininess, or specular value on an object, you can that metallic feel, with the sharp reflection off the object.

Explain what approach allows you to create a winter feel using Shaders.
Having a shader that gives a blue or a cooler colored tint to the game would create a winter feel. Having particle shaders for things like snow would also help convey the winter feel. Winter is associated with colder weather and a cooler environment colour wise as well.
Another approach for this could be using post-processing to lighten the forecast in the game environment.

Explanation of How to Implement:

A dynamic light that gives the effect of changing the scene (e.g., day passing, seasonal changes, etc.). This includes proper light behaviour when moving away or closer to objects.
In our prototype we implemented a light that fades to red when an enemy has detected you. We implemented this by lerping the colour parameter of the light from white to red, while updating our lighting UBO every frame. In the file “multiple_point_lights.glsl”, where the lighting contribution of every light in the scene is done, the colour parameter is used in calculating the diffuse.

Explain how you implemented the shader for this Midterm and indicate why this choice was made.
The shader used most often in our prototype is the “frag_blinn_phong_textured”. I’ve already discussed how the lighting calculations are done for this shader in the previous question, as the light contribution calculations are delegated to “multiple_point_lights.glsl”. After the lightAccumulation is calculated, this is then multiplied by a UV mapped texture taken from the uniform u_Material, providing our final result. For our PacMan prototype, we found that using this shader for elements like our walls and the ghosts fulfilled all of our needs when it came to diffuse and specular lighting on textures. Using a lighting model more advanced then this wouldn’t mesh well with the retro aesthetic of pac-man.