159.235 Assignment 3 - A 3D Graphics Scene

This assignment covers the concepts of coordinate transformations (in homogeneous coordinates), ray tracing, illumination, shading, and texture mapping. The necessary theory is discussed in the lectures together with programming tips.

Getting started

The Stream site provides some startup **`.java`** code together with some images you can use as texture maps. You can find this under the "Assignment 3" section.

- Bring up IntelliJ and create a new project - call it **`Assignment3`** (or whatever). Create a new package called **`nz.ac.massey.a3`**.
- Copy all the **`.java`** files from Stream in to the package folder in IntelliJ.
- Copy the images into the top level folder of your project
- Run **`Main.java`** from IntelliJ. You should see a JPanel/JFrame with a blue square at the centre. This is the view of the graphics scene comprising a single plane surface from the point-of-view of an imaginary camera placed on the z-axis and looking down. Try changing the camera and target positions and see what happens.
- Have a look at the code and read the comments and figure out what is going on. Note that the planar surface is rendered by the ray tracing method.

Going further

The next steps are to try and render other shapes, and implement shading, colouring, and texture mapping.

- Complete the implementation of the **`placeModel()`** method in the **`Placement`** class. This will set up the transformations to/from local to world space. Once this method is set up, try changing the position, orientation, and size of the unit plane surface.
- Now try and draw a sphere. You will need to complete the ray intersection computation in the **`Sphere`** class which is a subclass of **`SurfaceGeometry`**. You should see the sphere rendered as a filled circle on your display.
- Complete the shading to include the ambient, diffuse, and specular reflection terms according to the Phong reflection model. You need to take into account the position of ray intersection with the surface as well as the positions of the light source and view point. Implement this in the **`Material`** class.
- Complete the texture mapping procedure so that a 2d image can be pasted onto any of the surfaces (planar and spherical). Do this by making a **`TextureMap`** class in **`TextureMap.java`**.
- Try and create 6 planes and manipulate them so that they form the 6 faces of a cube. Place the cube in the scene by treating it as a compound figure.

Requirements

With these tasks complete, use your project to generate a 3D graphics scene comprising planar and spherical surfaces. It is possible to create impressive scenes using just these two types of surfaces.

Marking

Basic surfaces

You will receive the marks as indicated, if your graphics scene showcases the following:

- A plane surface with a texture pasted on it.

  [4 marks]

- A uniform colour sphere with a shiny surface (ie specular highlight)

  [4 marks]

- A sphere with a texture pasted on it (eg the beachball)

  [4 marks]

- The visible faces of a uniform colour cube correctly placed and shaded. The cube should be located at any arbitrary position in the scene—not just aligned with the principle axes in the world coordinates.

  [5 marks]

Graphics Scene

Here you can create a scene using several instances of planes, spheres, and cubes. Marks will be awarded for initiative. Be creative!

[4 marks]

Shadowing

Marks here will be earned if you can show shadows of objects in your scene. Here we are using just a single point source of light for illumination. You can place this at any suitable location in the scene. Use of ambient light can soften the shadows.

[4 marks]

Note

This assignment is to be completed using the ideas discussed in the lectures together with your own Java programming skills. You are not being asked to do your own research in advanced APIs and toolkits. **Do not use Java 3D, OpenGL, Vulkan, DirectX, Blender, etc.**

A graphical user interface is not required in this assignment. Your program should directly display the rendered scene to the screen upon running the program. An additional class to do this, `Display`, is provided in the startup code.

This assignment will count up to 25% of your final grade.

Due date: **2023 October 20 - 11:55 pm**.