CSCD 471

3D Advanced Computer Graphics

Revised Assignment 1

- Take the view position at (0.0f, 0.0f, 30.5f);
- Unitize the model by enclosing in a box (-1, -1, -1) to (1, 1, 1);
- Uniformly scale the model by 7.0f;
- Draw a disc of radius ‘10’ at the bottom of the bunny;
- You are implementing the basic assignment using Phong Shading;
- Perform Gouraud shading as well!
- So you need two different sets of shaders.
- In your application, switch between these two types of shading using a boolean variable;

Revised Assignment 1

- Draw a disc of radius ‘10’ at the bottom of the bunny
  - Declare a function in “objloader.cpp” that finds the dimension of the bounding box of the model;
  - You can find the minimum y position;
  - Draw a disc of radius ‘10’ with center at (0.0, min_y, 0.0)

Revised Assignment 1

- Switching between different shaders:
  - In main program in ‘Initialize()’ function:
  - Load two different set of shaders as follows:

```
program = LoadShaders("phong.vs", "phong.fs");
programOne = LoadShaders("gouraud.vs", "gouraud.vs");
```

Revised Assignment 1

- In ‘Display()’ function:

```
if (phong_shade){

glUseProgram(program);
// specify the uniform variables for this program
// draw bunny and disc
:
}

else{
glUseProgram(programOne);
// specify the uniform variables for this program
// draw bunny and disc
:

}
```

Revision

```
GLfloat cubeVertices[] = {
      // Positions      // Texture Coords
      -0.5f, -0.5f, -0.5f,  0.0f, 0.0f,
       0.5f, -0.5f, -0.5f,  1.0f, 0.0f,
       0.5f,  0.5f, -0.5f,  1.0f, 1.0f,
       0.5f,  0.5f, -0.5f,  1.0f, 1.0f,
      -0.5f,  0.5f, -0.5f,  0.0f, 1.0f,
      -0.5f, -0.5f, -0.5f,  0.0f, 0.0f,
      :
      :
      :
}
```

Revision

- void glVertexAttribPointer(GLuint index, GLint size, GLenum type, GLboolean normalized, GLsizei stride, const GLvoid * pointer);

```
GLuint cubeVAO, cubeVBO;
glGenVertexArrays(1, &cubeVAO);
glGenBuffers(1, &cubeVBO);
glBindVertexArray(cubeVAO);
glBindBuffer(GL_ARRAY_BUFFER, cubeVBO);
glBufferData(GL_ARRAY_BUFFER, sizeof(cubeVertices), &cubeVertices, GL_STATIC_DRAW);
glEnableVertexAttribArray(0);
glVertexAttribPointer(0, 3, GL_FLOAT, GL_FALSE, 5 * sizeof(GLfloat), (GLvoid*)0);
glEnableVertexAttribArray(1);
glVertexAttribPointer(1, 2, GL_FLOAT, GL_FALSE, 5 * sizeof(GLfloat), (GLvoid*)(3 * sizeof(GLfloat)));
glBindVertexArray(0);
```