

ETGG2801 – Graphics 1

Lab 2: Part 2 – Transformations, Forward Kinematics, Lambert Shading

Due on 10/13/2015

Overview

In this lab, you will be using the base code provided (**lab2part2base.zip**) to render and control a 3D model of a SCARA and a VIPER robot.

Tasks

1. Create a **Scene** class for representing a 3D scene.
 - a. The class should have the following methods:
 - i. **addObject(o)** – add an object, o, to the scene
 - ii. **removeObject(o)** – remove an object, o, from the scene
 - iii. **update(dtime)** – update all objects in the scene
 - iv. **render()** – render all objects in the scene
2. Add an instance of the Scene class to **MyDelegate**
 - a. Call scene's update and render methods from MyDelegate's update and render methods.
3. Add instances of the **Scara** and **Viper** robots to the scene class
 - a. Place the scara at position (0, -0.75, -3, 1), with an orientation of (45, -45, 0, 1)
 - b. Place the viper at position (0, 0.20, -3, 1), with an orientation of (45, -45, 0, 1).
4. Allow the robots' joints to be manipulated using the keyboard. You can assign keys to each joint, or come up with a way to switch between which joint on which robot is actively being controlled. Use joint limits to limit the range of values for each joint.
5. Add a point light to the scene and update the vertex and fragment shaders to support it
 - a. Position the light at (0.0, 0.0, 5.0, 1.0)
 - b. The shaders will implement Lambert shading
6. Since Lambert shading requires the surface normal, you will need to implement a method for generating the normal for each vertex of a model (see **generateNormals** in the Model class, which is called from the OBJReader's **readFile** method). DO NOT USE EXPORTED NORMALS FROM BLENDER. The normals will be stored in a vertex buffer object (VBO), just like the position and color data—you will need to create and initialize this buffer in the Model class' **loadToVRAM** method.