CSE 5542, Lab Assignment 2
Hierarchical Transformation, due on 02/21/2012 11:59PM

1 Goals

In this homework, we will learn how to tessellate 3D primitives and how to build a body model using hierarchical transformation.

2 Requirements

Please write a 3D OpenGL program that contains the following features.

2.1 3D Primitives (2 Points)

Since OpenGL\(^1\) does not have functions to draw curvilinear shapes, we will write functions to draw them by ourselves. We discussed how to draw a sphere during class, and its function can be found in the solar system example. Here please write functions to draw cube, cone, and sphere as polygonal meshes in wireframe:

**Box (1 Point).** A box is centered at the origin of its local space and its function uses six variables: size in the X direction, size in the Y direction, size in the Z direction, and a color vector.

**Truncated cone (1 Point).** A truncated cone uses the +Z axis as its axis and the XY plane as its bottom plane. Therefore, the origin of its local space should be the center of its bottom. Its function uses eight variables: bottom radius, top radius, height, number of slices, number of stacks, and a color vector. The number of slices determines how many vertices are used around the circle, and the number of stacks determines how many vertices are used along the height. This is a generic shape. When the top radius is zero, it is a cone; and when the bottom radius is equal to the top radius, it is a cylinder.

**Sphere.** A sphere is centered at the origin of its local space and its function uses six variables: radius, number of slices, number of stacks, and a color vector. You can simply use the example code.

When you write these three functions, please use the Vertex Buffer Objects (VBO). DO NOT use `glBegin` and `glEnd` any more.

2.2 3D Scene (2 Points)

Construct a static 3D scene using the functions defined in Section 2.1. The scene should contain a ground plane. Place some objects on the ground plane as decorations. For example, you can use a cylinder and a cone to create a pine tree, or you can use cylinders and boxes to create a vehicle. Use your imagination! No camera motion is required in this lab assignment. But you may want to place the camera (using `gluLookAt`) in a way that the whole scene can be conveniently viewed. You may use the solar system example as a reference.

2.3 Blobby man (6 points)

The hierarchical structure of a blobby man is shown in Figure 1a.

\(^1\)GLU and GLUT have such functions. But please DO NOT use them in this lab assignment.
The model (2 Points). Based on this structure, use 3D primitives and hierarchical transformation to create such a blobby man model. You can use ellipsoids to represent the body part as Figure 1a shows, by scaling a sphere nonuniformly. Alternatively, you can use cylinders or boxes. You can define each body part as a class and formulate the whole body using the tree data structure, similar to the solar system example. But if you are more comfortable with basic C programming, it is okay to simply write the whole drawing process procedurally in the Handle_Display function.

Body motion (3 Points). Create a set of body motion functions and allow the user to access to them through the keys. You can specify the amount of motion by yourself.

- When the f key is pressed, the whole body moves forward; when the b key is pressed, the whole body moves backward.
- When the t key is pressed, the whole body rotates counter clockwise; when the T key is pressed, the whole body rotates clockwise.
- When the l key is pressed, the body lifts its left arm and right leg, and lowers its right arm and left leg.
- When the r key is pressed, the body lifts its right arm and left leg, and lowers its left arm and right leg.

Animation (1 point). Using the Handle_Idle function and the body motion functions to create a walking animation. You may add more motions to make the walking process more realistic. For example, you may bend the elbows/knees, when the arms/legs are moving.

3 Bonus Credit (1 point)

Allow the user to toggle on/off the first-person view when the user press the v key. In the first-person view, the camera is placed at the head of the body and the viewing direction is the direction that the body is facing to. When the body moves, the view should change accordingly.

4 Submission Guideline

Please submit the source code (without the executable or object files) by email to our grader: Xiaoyin Ge (gex AT cse.ohio-state.edu). Use “CSE 5542 Lab 2” as the email subject.