Monash University
CSE3313 Computer Graphics

Tutorial 2 2007

13th August 2007

Objectives

- Understand the basics of transformations and the use of the matrix stack;
- Understand how to define multiple viewports and ensure correct aspect ratio in response to user events;
- Develop a strategy for the multiple viewports in assignment 1;
- Understand parameter editing and animation components of superformula editing.

Transformations and the Matrix Stack

On the CSE3313 web site you will find the *Glitch* OpenGL tutor program. Download the program and load some examples.

Exercise 1.1 (transforms) *Load the glitch tutorial file,* `simple`*.*

Insert calls to `glTranslatef` and `glRotatef`. Try editing the parameters and notice the changes visually.

Now try changing the order between `glTranslatef` and `glRotatef`. Note the difference in behaviour.

Exercise 1.2 (compound rotation) *Set up the glitch program so it can perform a compound rotation about the centroid of the object (as per lecture 10).*

Exercise 1.3 (viewport transformation) *Set up the glitch program so it can perform the compound viewport transformation (as per lecture 10).*

Exercise 1.4 (compound rotation) *Experiment with* `glMatrixMode`, `glLoadIdentity`, `glPushMatrix`, `glPopMatrix`. *Your goal should be to understand how these commands work internally in OpenGL.*

Glitch allows you to look at the contents of the current PROJECTION and MODELVIEW matrices. Note how their values change as you execute different commands.

Assignment 1 – Part 2

From the last tutorial. You should have developed an internal representation (data structure) for the superformula and be able to draw it in a window using OpenGL.

The next step is to be able to interactively edit the parameters and animate them as they change.

A demonstration will show how to edit the superformula interactively and what kind of results are expected for the first assignment.

Exercise 1.5 (Three viewports) *Create an OpenGL program that displays to three viewports, as per the specification in the first assignment. Allow the user to resize the window, while maintaining correct aspect ratio and display area in world co-ordinate space.*