Assignment 4
CSCI-4965: Three-Dimensional Computer Graphics

Due: Thursday, November 30, 2000, 10:00am

1 Overview

In this assignment, you are to write a 3D flight simulation program for a plane flying over an environment you create. For additional information and updates on the assignment, please see the assignment web page at `www.cs.rpi.edu/courses/fall00/graphics/assign4.html`.

2 The Flight Simulator Task

You are to write an OpenGL program to create a simple flight simulator. The displayed scene is the pilot's view as the plane moves. Your assignment consists of the following tasks:

1. Create an environment modeled using your favorite shapes (simple polygonal shapes are fine). For example, you can create mountains using polygons, and represent a lake by a blue polygon. You may texture map the objects for greater scene realism.

2. Create a plane to fly over the scene. The plane can be created from simple shapes such as polygons, cylinders, or cones. The displayed scene should be the pilot's view of the environment. You may assume the pilot is at the nose of the plane if you wish.

3. Provide the plane with motion capabilities. The initial position of the plane is specified by its (x, y, z) world coordinates. Its initial orientation is specified by its $(\theta_x, \theta_y, \theta_z)$ angles. These three angles specify sequential rotations of the plane about the X_w, Y_w, and Z_w world coordinate axes respectively. There is an orthogonal local coordinate frame attached to the plane such that the X_p axis points to the pilot's right, the Y_p axis points up from the pilot's seat, and the Z_p axis points to the back of the plane. Then *roll* corresponds to a rotation about the plane's Z_p axis, *yaw* corresponds to a rotation about the plane's Y_p axis, and *pitch* corresponds to a rotation about the plane's X_p axis. The user should be able to control the position and the orientation of the plane as specified by the keyboard functions below.

    - `f` key: The plane translates forward (in the direction of the negative Z_p axis) by a constant amount.
    - Right/left arrow keys: The plane rotates (rolls) 10 degrees clockwise/counterclockwise about its Z_p axis with each key press.
    - Down/up arrow keys: The plane rotates (pitches) 10 degrees clockwise/counterclockwise about its X_p axis with each key press.

- `y/Y` keys: The plane rotates (yaws) 10 degrees clockwise/counterclockwise about its Y_p axis with each key press.
- `i` key: The plane is returned to its initial configuration.
- `q` key: The program quits.

4. Create an animation of the scene as the plane flies over the environment. The flyby should include changes in both the position and orientation of the plane. A click of the right mouse button should start and stop the animation.
5. Your README file must describe the complete flight simulator functionality, including plane initial configuration, environment modeling information, and especially any extensions.
6. Here are some possible extensions to your assignment:

   - Add other environment features such as an airport runway or a rotating windmill.
   - Enable the user to control the roll velocity and pitch velocity of the plane based on the mouse's (appropriately scaled) horizontal and vertical position respectively. Enable/disable this mode by a click of the left mouse button.
   - Model two or more planes flying in formation, such that the user can select the current plane (and viewpoint) by cycling through the planes in sequence by pressing the `c` key. Input keyboard/mouse events modify the state of the current plane.

3 Handin

The code must be submitted no later than 10:00 am on November 30, 2000. **You are responsible for ensuring that your code can compile and run on the Sun Ultra10s in the OOT Lab (Amos Eaton 117) or in the Sparc Lab (Amos Eaton 217).** Hand in your code using the submit script.

You must hand in your source code (source and header files) along with a Makefile to compile it. Also include a README file with the following information (in addition to information requested above): your name, instructions on how to compile the code and run it, known bugs or limitations, any extra credit enhancements, and any other relevant information.

4 Grading

Your assignment will be graded as follows (100 points total):

1. Create scene of environment. 30 points
2. Translational and rotational motion control for plane. 40 points
3. Animation of plane flyby. 20 points
4. Code structure, clarity, and documentation. 10 points

Extra credit: You can earn up to 10 additional points for special features and creative enhancements to the assignment requirements.

Lateness policy: Late submissions will incur a penalty of 20% for the first day (24-hour period) after the submission deadline, and an additional 10% per day for subsequent days.