

**Computer Science 411/611**  
**Virtual Reality Systems**  
**Spring 2009**  
**Homework 2 (Midterm) – Free Fly**

**Due: Thursday, 2/19/2009**

### Overview

In this project, you will develop a program that allows the user to navigate a simple environment using the mouse. You can use this project as a base program to build upon for the next assignment, in which you will use the VR helmet as the input device.

### Description

Given the code available on the class website, develop a mouse-controlled system for dynamically viewing 3-D scenes from arbitrary position and orientation. Use two windows, as follows:

- view window
  - output only
  - window size should be 640x480
  - should display the current scene from the current view at every instant
  
- A/E/R/Color control window
  - input only
  - window size should be 512x512
  - cursor position within this window specifies (at every instant) desired azimuth and elevation for the view
  - azimuth is specified along the x direction and ranges from  $-\pi$  to  $\pi$
  - elevation is specified along the y direction and ranges from  $-\pi/2$  to  $\pi/2$
  
  - (optional) display a 2-D coordinate system with sufficient labels/grid marks to clearly specify current A/E values
  - (optional) create a third window 256x256 to show roll angle (i.e., a compass) with grid marks

Button functions (with cursor in A/E/R/Color control window):

- left button down: continuous zoom forward along current view direction, i.e.,

```
viewdir = (viewpt - eyept);  
eyept += increment * viewdir;  
viewpt = eyept + viewdir;  
(or viewpt += increment * viewdir;)
```

- right button down: continuous zoom backward along current view direction, i.e.,

```
viewdir = (viewpt - eyept);
eyept -= increment * viewdir;
viewpt = eyept + viewdir;
(or viewpt -= increment * viewdir;)
```

- middle button down: continuous roll, i.e.,

```
roll += increment;
```

Key functions (with cursor in A/E control window):

- 's' – return to start view
- 'd' – toggle roll direction
- 'r'/'R' – increase/decrease diffuse red component
- 'g'/'G' – increase/decrease diffuse green component
- 'b'/'B' – increase/decrease diffuse blue component
- 'q'/'Q' – quit

Other functions can be added to enhance your viewer.

For the viewing object, change the solid cube shown in class to a solid pyramid, or any another interesting object. You may want to write code to import obj files, which can be generated by a variety of modeling applications, such as Maya.

Students in 611 must also add rectangles for surrounding walls, ceiling, and floor, and shade them in an interesting way.

## Submission Requirements

You should submit the following:

- a hardcopy of your program (with your name, date, and compile line in a comment section at the beginning of your code)
- a sheet describing any special features of your project
- an electronic copy of your program (tar'ed and gzip'ed) sent to me at [tadavis@cs.clemson.edu](mailto:tadavis@cs.clemson.edu)