

COMP3201 – Computer Graphics

Assignment 2: “3D Beginnings”

Assignment Due: electronic submission by 6pm on Wednesday 12 April 2006

Task:

Construct a 3D environment for the prairie dog.

Objectives:

- To understand how to create a 3D scene in Computer Graphics.
- In particular, to understand the basics of Computer Graphics, including
 - Display Lists
 - Matrix stacks and their manipulation
 - Projection matrices
 - Cartesian navigation with additional pitch and yaw control
 - Aspect ratios of windows, viewport and viewing volume
 - OpenGL error checking

Submission

Assignments are to be submitted electronically via the web at <http://submit.itee.uq.edu.au/> . Code must **not** be compressed (e.g. zipped) for submission.

Code Style

Code must be written in ISO C90 and conform to the coding style available from http://www.itee.uq.edu.au/~comp3201/c_coding_style.html . Code must compile under Solaris with the Makefile available for download from <http://www.itee.uq.edu.au/~comp3201/Makefile> .

Remarking of Assignments

Any request for an assignment to be re-marked must be made in writing (not e-mail) to the lecturer (and not to the tutors). The request must contain details of why a re-mark should be granted (e.g. you believe there has been an error in marking a particular section of the code). Such requests must be made within 2 weeks of the assignment-feedback being sent to the students.

Demo Executable

A demo version of this assignment can be downloaded from the web site (choose your platform) and run, to allow verification of the functionality of this program. Depth testing and back face culling are provided in the demo to make the demo easier to use but are *not* part of the assignment. Only students enrolled in comp7204 need to implement the exploded view and the annotation.

Prairie Dog

All information required to create the prairie dog and their burrow can be found in the demo. Pressing the ‘n’ key allows you to cycle from position information to colour information to off. The demo deliberately leaves some information out. In such cases you are to make a reasonable assumption as to what the values are.

Program Functionality

- Background colour is a sky blue. R = 0.2, G = 0.3, B=0.5, A = 1.0.
- Escape key exits the program.
- ' ' [space] toggles the shading mode between flat and smooth shading.
- 'w' is a wire/fill toggle for the polygon mode.
- Resizing events are handled and the aspect ratio of the window, viewport and viewing volume are kept identical.
- The projection matrix is set to be a perspective projection with the near clipping plane 20 units from the camera and the far clipping plane set to 2000 units. The field of view in the y direction is set to 50 degrees.
- The land lies in the x-z plane and is a brown colour, representing a mixture of grass and dirt. R=0.48, G=0.28, B=0.20, A=1.0.
- The world is 1000 x 1000 units.
- Prairie dogs are located at random in the scene, however they are restricted to be standing upright on their burrows and not floating in the air. The positions and orientation of these objects are randomly chosen at the object's creation and constant thereafter.
- The "camera" position is updated on every redraw event. Holding a key down (generating keyboard events which should in turn create redraw events) will allow the user to fly through the scene.
- The 'a' and 'z' keys increase and decrease the elevation respectively. The elevation is along the y axis of the modelling coordinates.
- The left and right arrow keys increase/decrease the roll of the "camera".
- The up and down arrow keys increase/decrease the speed at which the "camera" moves through the scene.
- 's' and 'x' increase/decrease the pitch of the "camera"
- 'd' and 'c' increase/decrease the yaw of the "camera"
- In the demo, the yaw of the "camera" is updated using the following calculation:
$$\text{yaw} += \text{roll} * 0.1 / \text{speed};$$
- '0' [zero] sets the speed of the "camera" to zero.
- OpenGL errors must be checked for once per frame.
- F1 is an information toggle. The information will be displayed as a "Heads Up Display". The display will show the current navigation speed and the key bindings. The following code snippet demonstrates one possible way to display a C string in OpenGL.

```
void displayStr(const char *str, GLfloat x, GLfloat y)
{
    /* choose what font we wish to use */
    void* fontName = GLUT_BITMAP_HELVETICA_12;

    unsigned int index = 0;

    /* where do we want the text to go */
    glRasterPos2f(x, y);

    /* Display the string on the screen */
    while( str[index] != 0 && index < 1000 )
    {
        glutBitmapCharacter( fontName, str[index] );
        ++index;
    }
}
```

Additional COMP7204 Functionality

In addition to the above functionality, COMP7204 students must also implement

- the exploded view of the prairie dogs and mound (toggled via the 'e' key) and
- the annotation of the prairie dog and mound (toggled via the 'n' key).