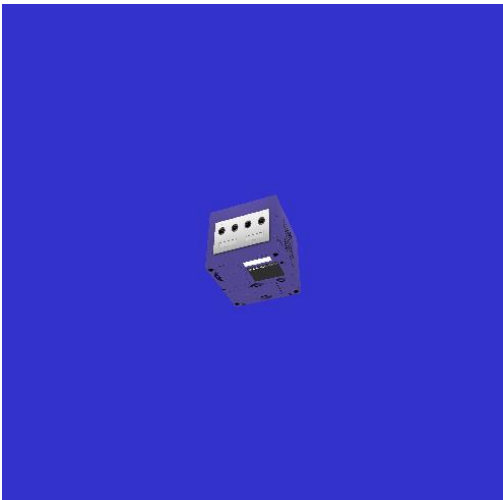


Erik Anchondo
3-5-19
CSE 520
Homework 3

1. Wrote a program that displays a textured cube where each face on the cube has a different texture applied to it. The cube is able to rotate and scale using user key input. The 3d object is a GameCube video game console.



Here is the code to the program:

```
#include <GL/gl.h>  
#include <GL/glu.h>  
#include <GL/glut.h>  
#include <stdlib.h>  
#include <stdio.h>  
#include <string.h>  
#include "imageio.h"
```

```

int texImageWidth;
int texImageHeight;
int window;
static GLuint texName[6];           //texture names
int angleX= 0, angleY = 0, angleZ = 0; //rotation angles
float scale = 1;

char maps[][20] =
{
    "gc_front.png",
    "gc_back.png",
    "gc_right.png",
    "gc_left.png",
    "gc_top.png",
    "gc_bottom.png"
};
GLubyte *makeTexImage( char *loadfile )
{
    int i, j, c, width, height;
    GLubyte *texImage;
    texImage = loadImageRGBA( (char *) loadfile, &width, &height);
    texImageWidth = width;
    texImageHeight = height;
    return texImage;
}

void init(void)
{
    glClearColor (0.2, 0.2, 0.8, 0.0);
    glShadeModel(GL_FLAT);
    glEnable(GL_DEPTH_TEST);
    glPixelStorei(GL_UNPACK_ALIGNMENT, 1);
    glGenTextures(6, texName);
    for ( int i = 0; i < 6; ++i ) {
        GLubyte *texImage = makeTexImage( maps[i] );
        if ( !texImage ) {
            printf("\nError reading %s \n", maps[i] );
            continue;
        }
        glBindTexture(GL_TEXTURE_2D, texName[i]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER,
GL_NEAREST);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER,
GL_NEAREST);
    }
}

```

```

    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGBA, texImageWidth,
                texImageHeight, 0, GL_RGBA, GL_UNSIGNED_BYTE, texImage);
    delete texImage;
}
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glEnable(GL_TEXTURE_2D);
    glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_DECAL);
    float x0 = -1.0, y0 = -1, x1 = 1, y1 = 1, z0 = 1;
    float face[6][4][3] =
    {
        {{x0, y0, z0}, {x1, y0, z0}, {x1, y1, z0}, {x0, y1, z0}},
        {{x0, y1, -z0}, {x1, y1, -z0}, {x1, y0, -z0}, {x0, y0, -z0}},
        {{x1, y0, z0}, {x1, y0, -z0}, {x1, y1, -z0}, {x1, y1, z0}},
        {{x0, y0, z0}, {x0, y1, z0}, {x0, y1, -z0}, {x0, y0, -z0}},
        {{x0, y1, z0}, {x1, y1, z0}, {x1, y1, -z0}, {x0, y1, -z0}},
        {{x0, y0, z0}, {x0, y0, -z0}, {x1, y0, -z0}, {x1, y0, z0}}
    };
    glEnable( GL_CULL_FACE );
    glCullFace ( GL_BACK );
    glPushMatrix();
    glRotatef( anglex, 1.0, 0.0, 0.0);
    glRotatef( angley, 0.0, 1.0, 0.0);
    glRotatef( anglez, 0.0, 0.0, 1.0);
    glScalef(scale, scale, scale);
    for ( int i = 0; i < 6; ++i ) {
        glBindTexture(GL_TEXTURE_2D, texName[i]);
        glBegin(GL_QUADS);
            glTexCoord2f(0.0, 0.0); glVertex3fv ( face[i][0] );
            glTexCoord2f(1.0, 0.0); glVertex3fv ( face[i][1] );
            glTexCoord2f(1.0, 1.0); glVertex3fv ( face[i][2] );
            glTexCoord2f(0.0, 1.0); glVertex3fv ( face[i][3] );
        glEnd();
    }
    glPopMatrix();
    glFlush();
    glDisable(GL_TEXTURE_2D);
}

void keyboard(unsigned char key, int x, int y)
{
    switch(key) {
        case 'x':

```

```

    anglex = ( anglex + 3 ) % 360;
    break;
case 'X':
    anglex = ( anglex - 3 ) % 360;
    break;
case 'y':
    angley = ( angley + 3 ) % 360;
    break;
case 'Y':
    angley = ( angley - 3 ) % 360;
    break;
case 'z':
    anglez = ( anglez + 3 ) % 360;
    break;
case 'Z':
    anglez = ( anglez - 3 ) % 360;
    break;
case 's':
    scale += 0.020;
    break;
case 'S':
    scale -= 0.020;
    break;
case 27:
    glutDestroyWindow(window);
    exit(0);
}
glutPostRedisplay();
}

void reshape(int w, int h)
{
    glViewport(0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluPerspective(60.0, (GLfloat) w/(GLfloat) h, 1.0, 30.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
    gluLookAt ( 0, 0, 5, 0, 0, 0, 0, 1, 0 );
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(500, 500);

```

```

glutInitWindowPosition(100, 100);
window = glutCreateWindow(argv[0]);
init();
glutDisplayFunc(display);
glutReshapeFunc(reshape);
glutKeyboardFunc(keyboard);
glutMainLoop();
return 0;
}

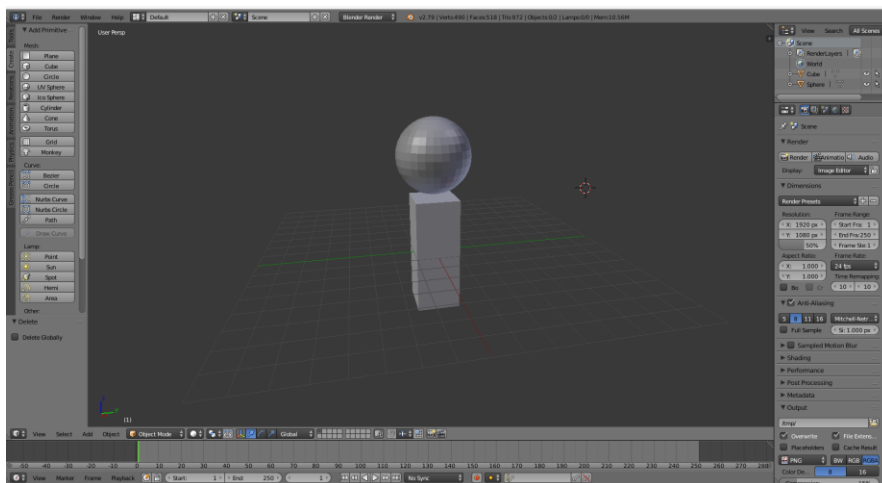
```

2. Made a 3d model of a person using blender.

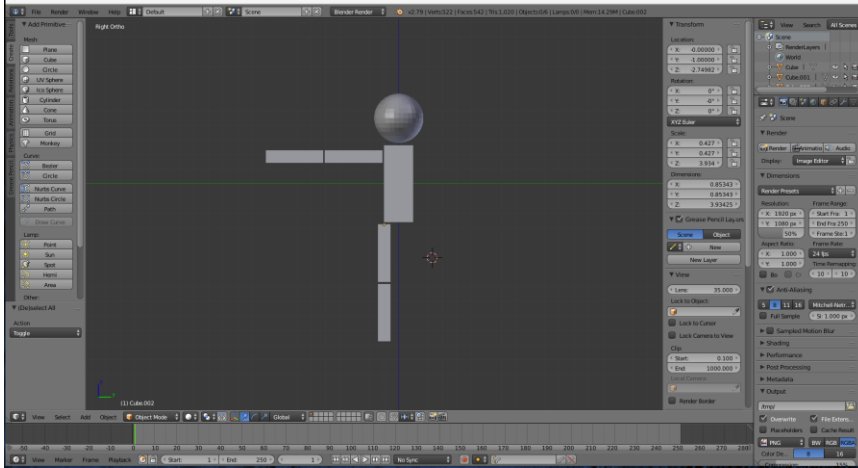


Steps:

- 1) Right click and select any unwanted objects. Then press x key to delete them.
- 2) Go to the create tab on the left and select any object you wish to start off with.
- 3) Place down objects to be the head and body. I used a sphere and cube. You can scale the by pressing the s key, translate using the g key, and rotate using the r key. To translate on a specific axis first press any of the keys used before and then press x, y, or z key to transform the object on the x, y, and z axis.



4) Place down more objects to make the arm and leg on one side as so.



5) Next select the leg and arm, then go to object tab at the bottom. Select mirror and then select mirror on y axis.

