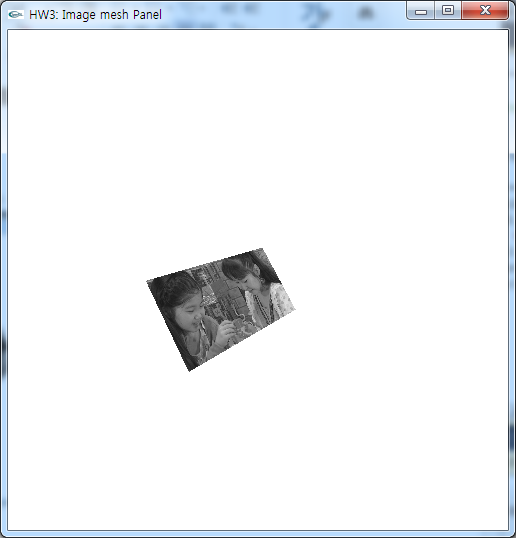
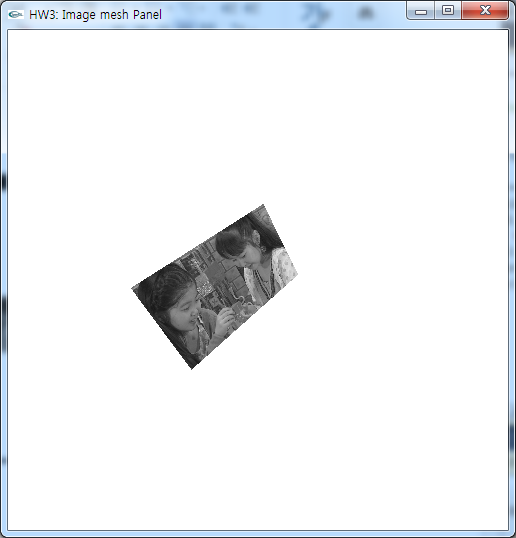
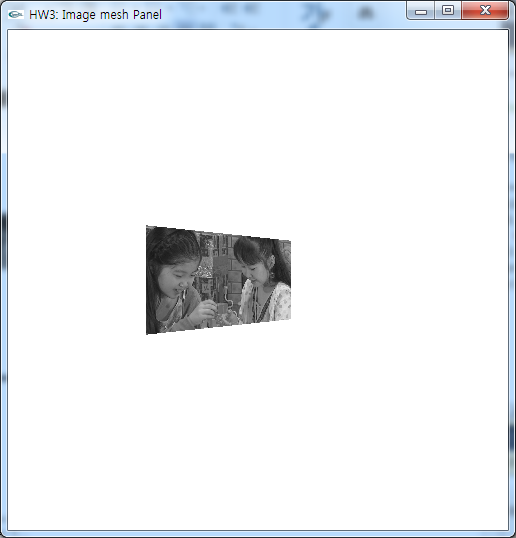
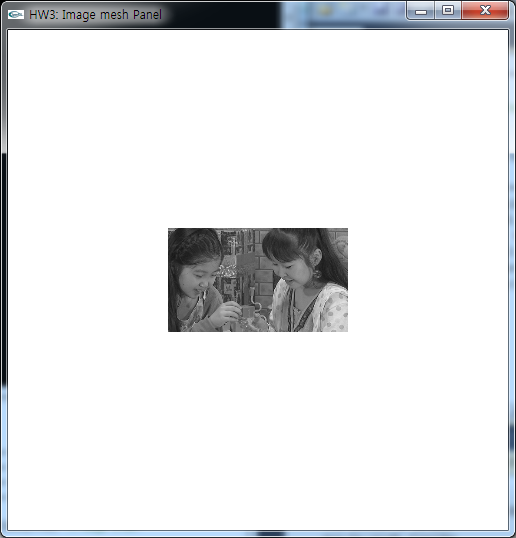
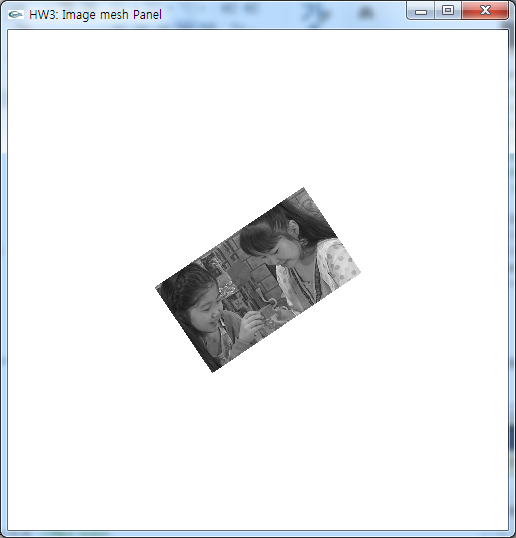
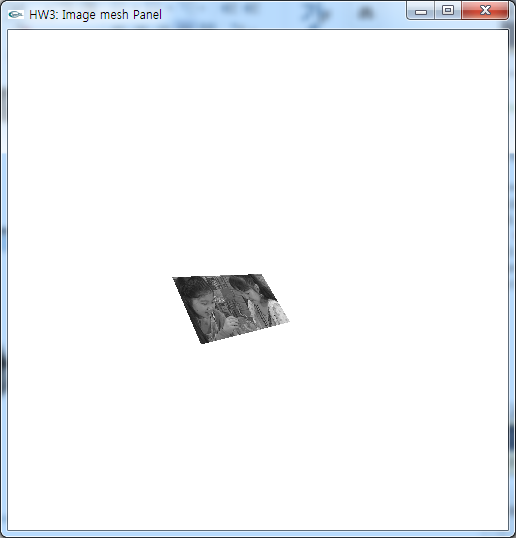
**HW#3 : Image mesh panel**

**201133216 정유석**

ScreenShot (Sequence = First -> x -> r -> y -> z -> o)





CODE

#include <stdio.h>

#include <math.h>

#include <gl/glut.h>

static GLfloat theta = 0.0; //Make angle

static GLint axis = 1;

static GLfloat viewer[3] = { 0.0, 0.0, 500.0 }; //Initial camera position

static GLint rot = 0; //rotation about z-axis by 20 angle

static unsigned char buff[240][416]; //image size

void init()

{

glMatrixMode(GL\_PROJECTION); //Set the Projection matrix

glLoadIdentity();

glClearColor( 1.0, 1.0, 1.0, 1.0 ); //Clear color

gluPerspective(85.0, 1.0, -500, 500.0); //Use perspective projection

glEnable ( GL\_DEPTH\_TEST ); //Depth test on

glMatrixMode(GL\_MODELVIEW); //Set the ModelView matrix

glLoadIdentity();

// read data

FILE \*fpt=fopen("testo.y", "rb");

fread(buff, 416\*240, sizeof(char), fpt);

fclose(fpt);

}

void spin\_cube()

{

glutPostRedisplay();

}

void keyboard\_handler(unsigned char key, int x, int y)

{

/\*

Keyboard ‘®¢çr’®? – toggle rotation about z-axis by 20 angle

Keyboard ‘®¢çx’®? – move camera position by -5.0 in x-axis

Keyboard ‘®¢çX’®? – move camera position by +5.0 in x-axis

Keyboard ‘®¢çy’®? – move camera position by -5.0 in y-axis

Keyboard ‘®¢çY’®? – move camera position by +5.0 in y-axis

Keyboard ‘®¢çz’®? – move camera position by -5.0 in z-axis

Keyboard ‘®¢çZ’®? – move camera position by +5.0 in z-axis

Keyboard ‘®¢ço’®? – reset camera position

\*/

if (key == 'x') viewer[0] -= 5.0; if (key == 'X') viewer[0] += 5.0;

if (key == 'y') viewer[1] -= 5.0; if (key == 'Y') viewer[1] += 5.0;

if (key == 'z') viewer[1] -= 5.0; if (key == 'Z') viewer[2] += 5.0;

if (key == 'o') { viewer[0] = 0.0; viewer[1] = 0.0; viewer[2] = 500.0; }

if (key == 'r'){ //rotation about z-axis by 20 angle

theta += 20.0;

if( theta > 360.0 ) theta -= 360.0;

}

}

void display()

{

glClear( GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT ); //Clear buffer

glLoadIdentity();

gluLookAt( viewer[0], viewer[1], viewer[2], 0.0, 0.0, 0.0, 0.0, 1.0, 0.0 );//Set camera

glRotatef( theta, 0.0, 0.0, 1.0 );

glScalef( 1.0, -1.0, 1.0 ); //Y-axis should be reflected (𝑦=-𝑦)

glTranslatef( -208, -120, -128 ); //Make panel.

GLint xp, yp, z;

for ( int y=1; y<240-1; y++ )

{

for ( int x=1; x<416-1; x++ )

{

glBegin( GL\_POLYGON ); //make image

xp=x+0; yp=y+0; z=(GLfloat)buff[yp][xp];

glColor3f( (GLfloat)z/255, (GLfloat)z/255, (GLfloat)z/255 );

glVertex3i( xp, yp, 0);

xp=x+1; yp=y+0; z=(GLfloat)buff[yp][xp];

glColor3f( (GLfloat)z/255, (GLfloat)z/255, (GLfloat)z/255 );

glVertex3i( xp, yp, 0);

xp=x+1; yp=y+1; z=(GLfloat)buff[yp][xp];

glColor3f( (GLfloat)z/255, (GLfloat)z/255, (GLfloat)z/255 );

glVertex3i( xp, yp, 0);

xp=x+0; yp=y+1; z=(GLfloat)buff[yp][xp];

glColor3f( (GLfloat)z/255, (GLfloat)z/255, (GLfloat)z/255 );

glVertex3i( xp, yp, 0);

glEnd();

}

}

glFlush();

glutSwapBuffers(); //change buffer in background

}

int main(int argc, char\* argv[])

{

glutInit( &argc, (char\*\*)argv );

glutInitDisplayMode( GLUT\_DOUBLE | GLUT\_RGB | GLUT\_DEPTH ); //Clare display mode

glutInitWindowSize( 500, 500 );

glutCreateWindow("HW3: Image mesh Panel");

glutIdleFunc(spin\_cube);

glutKeyboardFunc(keyboard\_handler); //Receipt keyboard event

glutDisplayFunc(display); //Call display function

init();

glutMainLoop();

return 0;

}