#include<GL\glut.h>

#include<iostream>

using namespace std;

float WinWid = 400.0, WinHei = 400.0, Angle = 0.0;

void Draw()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glPushMatrix();

glRotatef(Angle, 0.0, 0.0, 1.0);

glBegin(GL\_LINES);

for (float i = -WinWid/2; i<= WinWid/2; i += 20.0)

{

glVertex2f(i, -WinHei / 2);

glVertex2f(i, WinHei / 2);

}

for (float i = -WinHei / 2; i <= WinHei / 2; i += 20.0)

{

glVertex2f(-WinWid / 2, i);

glVertex2f(WinWid / 2, i);

}

glEnd();

glPopMatrix();

glutSwapBuffers(); // for GLUT\_DOUBLE

}

void Initialize()

{

glClearColor(0.0, 0.0, 0.0, 1.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-WinWid/2, WinWid/2, -WinHei/2, WinHei/2, -200.0, 200);

glMatrixMode(GL\_MODELVIEW);

}

void Timer(int value)

{

/\* switch(value)

{

case 0: glColor3f(1.0, 1.0, 1.0);

break;

case 1: glColor3f(1.0, 0.0, 0.0);

}\*/

//glRotatef(1, 0.0, 0.0, 1.0);

glutPostRedisplay();

glutTimerFunc(10, Timer, 0);

}

void Keyboard(unsigned char key, int x, int y)

{

switch (key)

{

case 'a': Angle++;

break;

case 'd': Angle--;

break;

}

cout << Angle;

}

void SKeyboard(int key, int x, int y)

{

switch (key)

{

case GLUT\_KEY\_LEFT: Angle++;

break;

case GLUT\_KEY\_RIGHT: Angle--;

break;

}

}

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

{

//Initialization

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(WinWid, WinHei);

glutInitWindowPosition(100, 200);

glutCreateWindow("Fuck me in my ass");

//Registration

glutDisplayFunc(Draw);//Drawing

glutTimerFunc(500, Timer, 0); // animation func registration

glutKeyboardFunc(Keyboard);

glutSpecialFunc(SKeyboard);

Initialize();

glutMainLoop();

return 0;

}