#include <windows.h>

#include <GL/glut.h>

GLfloat i = 0.0f;

void initGL()

{

glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque

}

void Idle()

{

glutPostRedisplay();//// marks the current window as needing to be redisplayed

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();//Reset the current matrix

glPushMatrix(); //glPushMatrix copies the top matrix and pushes it onto the stack, while glPopMatrix pops the top matrix off the stack

glRotatef(i,0.1,0.0,0.0);//i=how many degree you want to rotate around an axis

// glTranslatef(0.f,0.0f,0.f);

glBegin(GL\_QUADS);

glColor3f(1.0f, 0.0f, 0.0f);

glVertex2f(0.1f, 0.1f);

glVertex2f( 0.6f, 0.1f);

glVertex2f( 0.6f, 0.5f);

glVertex2f( 0.1f, 0.5f);

glEnd();

glPopMatrix();

i+=0.1f;

glFlush();

}

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

{

glutInit(&argc, argv); // Initialize GLUT

glutInitWindowSize(320, 320);

glutCreateWindow("Model Transform");

glutDisplayFunc(display);//

initGL();

glutIdleFunc(Idle);//glutIdleFunc sets the global idle callback to be func so a GLUT program can perform background processing tasks or continuous animation when window system events are not being received.

glutMainLoop();

return 0;}