Windmill

#include<windows.h>

#include <GL/glut.h>

#include <cmath>

float rotation\_angle = 0.0f;

float rotation\_speed = 1.0f;

void draw\_blades()

{

glColor3f(0.5, 0.5, 0.5);

glBegin(GL\_TRIANGLES);

glVertex2f(0, 0);

glVertex2f(0.3, 0.2);

glVertex2f(0.3, -0.2);

glEnd();

}

void draw\_windmill() {

glPushMatrix();

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

glBegin(GL\_QUADS);

glVertex2f(-0.1, -1);

glVertex2f(0.1, -1);

glVertex2f(0.1, -0.3);

glVertex2f(-0.1, -0.3);

glEnd();

glTranslatef(0.0f, -0.3f, 0.0f);

glRotatef(rotation\_angle, 0.0f, 0.0f, 1.0f);

glColor3f(0.8f, 0.8f, 0.8f);

for (int i = 0; i < 3; i++) {

draw\_blades();

glRotatef(120.0f, 0.0f, 0.0f, 1.0f);

}

glPopMatrix();

}

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glLoadIdentity();

draw\_windmill();

glFlush();

glutSwapBuffers();

}

void update(int value) {

rotation\_angle += rotation\_speed;

if (rotation\_angle > 360.0f) {

rotation\_angle -= 360.0f;

}

glutPostRedisplay();

glutTimerFunc(10, update, 0);

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutCreateWindow("OpenGl");

glClearColor(0.0f, 0.0f, 0.0f, 0.0f);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);

glutDisplayFunc(display);

glutTimerFunc(10, update, 0);

glutMainLoop();

return 0;

}