Implement animation principles for any object

\*/

#include <iostream>

#include <math.h>

#include <time.h>

#include <GL/glut.h>

using namespace std;

int y = 0; // y-coordinate instead of x

int flag = 0;

void init() {

glClearColor(1.0, 1.0, 1.0, 0.0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0, 640, 0, 480);

}

void object1() {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1, 1, 0); // Yellow color

glBegin(GL\_TRIANGLES); // Drawing a triangle

glVertex2i(320, y + 60); // Top vertex

glVertex2i(280, y); // Bottom left vertex

glVertex2i(360, y); // Bottom right vertex

glEnd();

glutSwapBuffers();

}

void timer(int) {

glutPostRedisplay();

glutTimerFunc(1000 / 60, timer, 0);

if (flag == 0) {

y = y + 3; // Moving downwards

}

if (flag == 1) {

y = y - 3; // Moving upwards

}

if (y >= 440) {

flag = 1; // Change direction when reaching bottom

}

if (y <= 0) {

flag = 0; // Change direction when reaching top

}

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_DOUBLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(200, 200);

glutCreateWindow("Animation");

init();

glutDisplayFunc(object1);

glutTimerFunc(1000, timer, 0);

glutMainLoop();

return 0;

}

//output:



