**Moving Object:**

#include<cstdio>

#include <GL/gl.h>

#include <GL/glut.h>

GLfloat position = 0.0f;

GLfloat speed = 0.1f;

void update(int value)

{

if(position <= -1.0)

position = 1.0f;

position -= speed;

glutPostRedisplay();

glutTimerFunc(100, update, 0);

}

//https://sl.bing.net/buRp9srADhA

void display() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glPushMatrix();

glTranslatef(position,0.0f, 0.0f);

glBegin(GL\_QUADS);

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

glVertex2f(-0.2f, -0.2f);

glVertex2f( 0.2f, -0.2f);

glVertex2f( 0.2f, 0.2f);

glVertex2f(-0.2f, 0.2f);

glEnd();

glBegin(GL\_TRIANGLES);

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

glVertex2f(0.2f,-0.2f);

glVertex2f(0.5f, 0.0f);

glVertex2f(0.2f, 0.2f);

glEnd();

glPopMatrix();

glFlush();

}

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

glutInit(&argc, argv);

glutInitWindowSize(320, 320);

//glutInitWindowPosition(50, 50);

glutCreateWindow("Translation Animation");

glutDisplayFunc(display);

glutTimerFunc(100, update, 0);

glutMainLoop();

return 0;

}

//glutTimerFunc: This function sets a timer callback.

//It schedules a function to be called after a specified amount of time has passed.

//(100, update, 0):

//The first argument (100) represents the time interval in milliseconds. In this case, it’s set to 100 milliseconds (or 0.1 seconds).

//The second argument (update) is the callback function that will be executed when the timer expires. In your example, the function named update will be called.

//The third argument (0) is an integer value that you can use to pass additional data to the callback function (if needed).

**Changing Display**

#include<cstdio>

#include <GL/gl.h>

#include <GL/glut.h>

GLfloat position = 0.0f;

GLfloat speed = 0.1f;

GLfloat position1 = 0.0f;

GLfloat speed1 = 0.1f;

void update(int value) {

if(position <-1.0)

position = 1.0f;

position -= speed;

glutPostRedisplay();

glutTimerFunc(100, update, 0);

}

void update1(int value1) {

if(position1 <-1.0)

position1 = 1.0f;

position1 -= speed1;

glutPostRedisplay();

glutTimerFunc(100, update1, 0);

}

void display1\_view()

{

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glPushMatrix();

glTranslatef(0.0f,position1, 0.0f);

glBegin(GL\_QUADS);

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

glVertex2f(-0.2f, -0.2f);

glVertex2f( 0.2f, -0.2f);

glVertex2f( 0.2f, 0.2f);

glVertex2f(-0.2f, 0.2f);

glEnd();

glBegin(GL\_TRIANGLES);

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

glVertex2f(0.2f,-0.2f);

glVertex2f(0.5f, 0.0f);

glVertex2f(0.2f, 0.2f);

glEnd();

glPopMatrix();

glFlush();

}

void display1(int b)

{

glutDisplayFunc(display1\_view);

}

void display() {

glClearColor(0.0f, 0.0f, 0.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glPushMatrix();

glTranslatef(position,0.0f, 0.0f);

glBegin(GL\_QUADS);

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

glVertex2f(-0.2f, -0.2f);

glVertex2f( 0.2f, -0.2f);

glVertex2f( 0.2f, 0.2f);

glVertex2f(-0.2f, 0.2f);

glEnd();

glBegin(GL\_TRIANGLES);

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

glVertex2f(0.2f,-0.2f);

glVertex2f(0.5f, 0.0f);

glVertex2f(0.2f, 0.2f);

glEnd();

glPopMatrix();

glutTimerFunc(5000,display1,0);

glFlush();

}

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

glutInit(&argc, argv);

glutInitWindowSize(320, 320);

//glutInitWindowPosition(50, 50);

glutCreateWindow("Translation Animation");

glutDisplayFunc(display);

glutTimerFunc(100, update, 0);

glutTimerFunc(5000, update1, 0);

glutMainLoop();

return 0;

}