**Animation**

#include <GL/glut.h>

// Global variables

float squareX = 0.0; // Initial x-coordinate of the square

// Function to draw the square

void drawSquare() {

glBegin(GL\_QUADS);

glVertex2f(squareX, 0.0); // Bottom-left vertex

glVertex2f(squareX + 0.2, 0.0); // Bottom-right vertex

glVertex2f(squareX + 0.2, 0.2); // Top-right vertex

glVertex2f(squareX, 0.2); // Top-left vertex

glEnd();

}

// Function to display the scene

void display() {

glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer

glColor3f(1.0, 1.0, 1.0); // Set color to white

drawSquare(); // Draw the square

glFlush(); // Flush OpenGL buffers to the window

}

// Function to update the animation

void update(int value) {

// Move the square to the right

squareX += 0.01;

// Wrap around to the left if the square goes off the screen

if (squareX > 1.1) {

squareX = -0.3;

}

// Redraw the scene

glutPostRedisplay();

// Call update function again after 16 milliseconds

glutTimerFunc(16, update, 0);

}

// Function to initialize OpenGL

void init() {

glClearColor(0.0, 0.0, 0.0, 1.0); // Set the background color to black

glMatrixMode(GL\_PROJECTION); // Set the matrix mode to projection

glLoadIdentity(); // Load the identity matrix

gluOrtho2D(-1.0, 1.0, -1.0, 1.0); // Set the clipping area

}

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

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

glutInitWindowSize(400, 400); // Set the window size

glutCreateWindow("Basic Animation"); // Create a window with the given title

glutDisplayFunc(display); // Set the display callback function

glutTimerFunc(0, update, 0); // Set the update function and the delay

init(); // Initialize OpenGL parameters

glutMainLoop(); // Enter the GLUT event loop

return 0;

}