**Name: Prathamesh Gadekar**

**Roll No: 15**

**Class: SE(IT)**

// C program to demonstrate drawing a circle using OpenGL

#include<GL/glut.h>

#include<math.h>

#define pi 3.142857

// function to initialize

void myInit (void)

{

glClearColor(0.0, 0.0, 0.0, 1.0); // making background color black

glColor3f(0.0, 1.0, 0.0); // making picture color green (in RGB mode)

glPointSize(1.0); // breadth of picture boundary is 1 pixel

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(-780, 780, -420, 420); // setting window dimension in X- and Y- direction

}

void display (void)

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_POINTS);

float x, y, i;

// iterate y up to 2\*pi, i.e., 360 degree

// with small increment in angle as

// glVertex2i just draws a point on specified co-ordinate

for ( i = 0; i < (2 \* pi); i += 0.001)

{

// let 200 is radius of circle and as,

// circle is defined as x=r\*cos(i) and y=r\*sin(i)

x = 200 \* cos(i);

y = 200 \* sin(i);

glVertex2i(x, y);

}

glEnd();

glFlush();

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

// giving window size in X- and Y- direction

glutInitWindowSize(1366, 768);

glutInitWindowPosition(0, 0);

// Giving name to window

glutCreateWindow("Circle Drawing");

myInit();

glutDisplayFunc(display);

glutMainLoop();

}

// C program to demonstrate drawing a triangle using OpenGL

#include <GL/glut.h>

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glBegin(GL\_TRIANGLES);

glVertex2f(-0.5f, -0.5f);

glVertex2f( 0.0f, 0.5f);

glVertex2f( 0.5f, -0.5f);

glEnd();

glFlush();

}

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

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL Test");

glutDisplayFunc(display);

glutMainLoop();

return 0;

}

// C program to demonstrate drawing a square using OpenGL

#include <GL/glut.h>

void display(void) {

glClear( GL\_COLOR\_BUFFER\_BIT);

glColor3f(0.0, 1.0, 0.0);

glBegin(GL\_POLYGON);

glVertex3f(2.0, 4.0, 0.0);

glVertex3f(8.0, 4.0, 0.0);

glVertex3f(8.0, 6.0, 0.0);

glVertex3f(2.0, 6.0, 0.0);

glEnd();

glFlush();

}

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

glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(100,100);

glutInitWindowSize(300,300);

glutCreateWindow ("square");

glClearColor(1.0, 1.0, 1.0, 0.0); // black background

glMatrixMode(GL\_PROJECTION); // setup viewing projection

glLoadIdentity(); // start with identity matrix

glOrtho(0.0, 10.0, 0.0, 10.0, -1.0, 1.0); // setup a 10x10x2 viewing world

glutDisplayFunc(display);

glutMainLoop();

}