SSN COLLEGE OF ENGINEERING, KALAVAKKAM

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

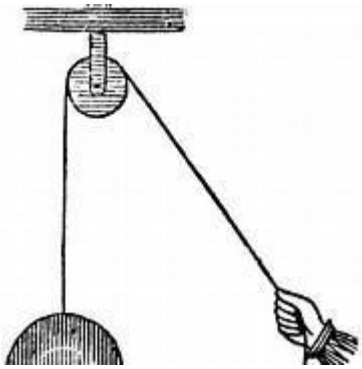
UCS1712 – GRAPHICS AND MULTIMEDIA LAB

Lab Test – Tuesday Batch **Date: 19-10-23** **K3,CO1,CO2**

1. Using Bresenham's Line drawing and Circle drawing algorithms draw Captain America's shield.

2. Draw ceiling fan and apply 2-D transformations to simulate the rotating motion.

3. Draw a scene depicting the pulley action of lifting a weight and simulate the motion.



Register Number : 205001098
Name : Shalini N
Question number : 1

Code :

```
//
//  main.cpp
//  test1
//
//  Created by exam1 on 19/10/23.
//
#include <GLUT/glut.h>
#include <iostream>
#include <cmath>
using namespace std;
int x,y;
void myInit(){
   glClearColor(1.0,1.0,1.0,0.0);
   glColor3f(1.0f,0.0f,0.0f);
   glPointSize(10);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0.0,1000.0,0.0,1000.0);
}
void plotPixel(int x,int y) {
   glBegin(GL_POINTS);
   glVertex2d(x,y);
   glEnd();
//    glFlush();
}
void draw_line(int xstart, int ystart,int xend, int yend)
{


   int dx,dy,incx,incy,inc1,inc2,e;
   dx=abs(xend-xstart);
   dy=abs(yend-ystart);

   incx=1;
   if(xend<xstart) incx=-1;
   incy=1;
   if(yend<ystart) incy=-1;

   int x,y;
   x=xstart;
   y=ystart;

   if(dx>dy)
```

```
    {
        plotPixel(x, y);
        e=2*dy-dx;
        inc1=2*(dy-dx);
        inc2=2*dy;
        for(int i=0;i<dx;i++) {
            if(e>=0) {
                e+=inc1;
                y+=incy;
            } else {
                e+=inc2;
            }
            x+=incx;
            plotPixel(x, y);
        }
    } else {
        plotPixel(x, y);
        e=2*dx-dy;
        inc1=2*(dx-dy);
        inc2=2*dx;
        for(int i=0;i<dy;i++) {
            if(e>=0) {
                e+=inc1;
                x+=incx;
            } else {
                e+=inc2;
            }
            y+=incy;
            plotPixel(x, y);
        }

    }
}
void comb(int xp,int yp,int xc,int yc) {
    plotPixel(xp+xc, yp+yc);
    plotPixel(yp+xc, xp+yc);
    plotPixel(-xp+xc, -yp+yc);
    plotPixel(-yp+xc, -xp+yc);
    plotPixel(-xp+xc, yp+yc);
    plotPixel(-yp+xc, xp+yc);
    plotPixel(xp+xc, -yp+yc);
    plotPixel(yp+xc, -xp+yc);
}
void draw_circle(int radius,int xc,int yc) {
    int p0=1-radius;
    int xs=0;
    int ys=radius;
    comb(xs,ys,xc,yc);

    while(xs<ys) {
        xs+=1;
        int pnew=p0+2*xs+1;
        if(p0 >= 0) {
            ys-=1;
            pnew-=2*ys;
        }
```

```
        comb(xs,ys,xc,yc);
        p0=pnew;
    }

}
void myDisplay()
{
    glClear(GL_COLOR_BUFFER_BIT);


    glPointSize(15);

    glColor3f(0.0f,0.0f,1.0f);
    draw_circle(150, 500, 500);
    draw_circle(165, 500, 500);
    draw_circle(180, 500, 500);
    draw_circle(100, 500, 500);
    draw_circle(115, 500, 500);
    draw_circle(135, 500, 500);

    draw_circle(15, 500, 500);
    draw_circle(35, 500, 500);
    draw_circle(60, 500, 500);
    draw_circle(80, 500, 500);
    draw_circle(115, 500, 500);
    draw_circle(135, 500, 500);




    glColor3f(1.0f,0.0f,0.0f);
    draw_circle(200, 500, 500);
    glColor3f(1.0f,0.0f,0.0f);
    draw_circle(215, 500, 500);
    glColor3f(1.0f,0.0f,0.0f);
    draw_circle(230, 500, 500);

    glColor3f(0.5f,0.5f,0.5f);
    draw_circle(250, 500, 500);
    glColor3f(0.5f,0.5f,0.5f);
    draw_circle(265, 500, 500);
    glColor3f(0.5f,0.5f,0.5f);
    draw_circle(280, 500, 500);

    glColor3f(1.0f,0.0f,0.0f);
    draw_circle(300, 500, 500);
    glColor3f(1.0f,0.0f,0.0f);
    draw_circle(315, 500, 500);
    glColor3f(1.0f,0.0f,0.0f);
    draw_circle(330, 500, 500);

    glPointSize(15);
    draw_line(450,580,500,700);
    draw_line(550,580,500,700);
    draw_line(320,580,450,580);
    draw_line(550,580,680,580);
```

```
    draw_line(550,500,680,580);
    draw_line(320,580,450,500);
    draw_line(450,500,370,370);
    draw_line(550,500,630,370);
    draw_line(370,370,500,450);
    draw_line(500,450,630,370);

//    glColor3f(0.5f,0.5f,0.5f);
//    draw_circle(380, 500, 500);


    glFlush();

}
int main(int argc,char * argv[]) {
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(1000, 1000);
    glutCreateWindow("test-lab");

    glutDisplayFunc(myDisplay);
    myInit();
    glutMainLoop();
    return 1;
}
```

output :