Name : Shrikant Gavhale

Div : A

Roll no : 26

Circle Drawing Algorithm

**Code:**

#include<windows.h>

#include<stdio.h>

#include<math.h>

#include<GL/glut.h>

// Center of the circle = (320, 240)

int xc = 320, yc = 240;

// Plot eight points using circle's symmetrical property

void plot\_point(int x, int y)

{

glBegin(GL\_POINTS);

glVertex2i(xc+x, yc+y);

glVertex2i(xc+x, yc-y);

glVertex2i(xc+y, yc+x);

glVertex2i(xc+y, yc-x);

glVertex2i(xc-x, yc-y);

glVertex2i(xc-y, yc-x);

glVertex2i(xc-x, yc+y);

glVertex2i(xc-y, yc+x);

glEnd();

}

// Function to draw a circle using bresenham's

// circle drawing algorithm

void bresenham\_circle(int r)

{

int x=0,y=r;

float pk=(5.0/4.0)-r;

/\* Plot the points \*/

/\* Plot the first point \*/

plot\_point(x,y);

// int k;

/\* Find all vertices till x=y \*/

while(x < y)

{

x = x + 1;

if(pk < 0)

pk = pk + 2\*x+1;

else

{

y = y - 1;

pk = pk + 2\*(x - y) + 1;

}

plot\_point(x,y);

}

glFlush();

}

// Function to draw two concentric circles

void concentric\_circles(void)

{

//Clears buffers to preset values

glClear(GL\_COLOR\_BUFFER\_BIT);

int radius = 100;

bresenham\_circle(radius);

}

void Init()

{

glClearColor(0.0,1.0,1.0,0.0); //clear values for RGBA

glColor3f(1.0f,0.0f,0.0f); //set current color - RGB

glPointSize(3.0); //diameters of rasterized points

glMatrixMode(GL\_PROJECTION); // specifies current matrix

glLoadIdentity();

gluOrtho2D(0.0,800.0,0.0,600.0);

}

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

{

/\* Initialise GLUT library \*/

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); //Set the initial display mode

glutInitWindowPosition(0,0);

glutInitWindowSize(800,600);

glutCreateWindow("Bresenham Circle Drawing");

/\* Initialize drawing colors \*/

Init();

/\* Call the displaying function \*/

glutDisplayFunc(concentric\_circles);

/\* Keep displaying untill the program is closed \*/

glutMainLoop();

}

Output:

