Mid-Point Ellipse Algorithm

Theory:

An ellipse is defined as the set of points such that the sum of the distances from two fixed point / positions (foci) is same for all points.

The general equation of ellipse is :

$$\frac{(x - x_c)^2}{r_x^2} + \frac{(y - y_c)^2}{r_y^2} = 1$$

In polar form,

$$x = x_c + r_x \cos\theta$$
$$y = y_c + r_y \sin\theta .$$

Algorithm:

Step 1 : Start

Step 2 : Declare variables $x_c, y_c, r_x, r_y, x_0, y_0, P_0, P_k, P_{k+1}$

Step 3 : Read values of x_c, y_c, r_x, r_y .

Step 4 : Obtain the first point on an ellipse centered on origin (x_0, y_0) by initializing x_0 and y_0 as

$$x_0 = 0$$
$$y_0 = r$$

Step 5 : Calculate the initial value of the decision parameter in region 1 as :

$$P_{10} = r_y^2 - r_x^2 r_y + \frac{1}{4} r_x 2$$

Step 6 : For each x_k position in region 1, starting at $k = 0$, perform the following test.

If $P_{1k} < 0$,

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k$

$P_{1k+1} = P_{1k} + 2r_y^2 x_{k+1} + r_y^2$

else

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k - 1$

$P_{1k+1} = P_{1k} + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_y^2$

and continue until $2r_y^2 x \geq 2r_y^2 y$.

Step 7: Calculate the initial decision parameter in region 2 using the last point (x_0, y_0) calculated in region 1 as

$$P_{20} = r_y^2\left(x_0 + \frac{1}{2}\right)^2 + r_x^2(y_0 - 1)^2 - r_x^2 r_y^2$$

Step 8: At each y_k position in region 2, starting at $k=0$, perform the following test.

If $P_{2k} > 0$,

$x_{k+1} = x_k$

$y_{k+1} = y_k - 1$

$P_{2k+1} = P_{2k} - 2r_x^2 y_{k+1} + r_x^2$

else

$x_{k+1} = x_k + 1$

$y_{k+1} = y_k - 1$

$P_{2k+1} = P_{2k} + 2r_y^2 x_{k+1} - 2r_x^2 y_{k+1} + r_x^2$

Step 9: Determine the symmetry points in the other three quadrants.

Step 10: Move each calculated pixel positions (x, y) onto the elliptical path centered on (x_c, y_c) and plot the co-ordinate values.

$x = x + x_c$.

$y = y + y_c$

Step 11: Repeat the steps for region 2 until $y < 0$.

Step 12: Stop.

Discussion:

In this lab, we used mid-point ellipse drawing algorithm to draw an ellipse. Unlike in circle, there are two parameters in ellipse. Each quadrant of the ellipse is divided into two regions each. If $2r_y^2x > 2r_x^2y$, then the region is known as region 2. We performed the operation as shown in algorithm above.

Conclusion:

Hence, in this lab we drew an ellipse using mid-point ellipse drawing algorithm.

SOURCE CODE:

```
#include <graphics.h>
#include <stdlib.h>
#include <stdio.h>
#include <conio.h>

void ellipse1(float,float,float,float);
int main(void)
{
   /* request auto detection */
   int gdriver = DETECT, gmode, errorcode;
   int i;

   /* initialize graphics and local variables */
   initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");

   /* read result of initialization */
   errorcode = graphresult();
   if (errorcode != grOk)  /* an error occurred */
   {
      printf("Graphics error: %s\n", grapherrormsg(errorcode));
      printf("Press any key to halt:");
      getch();
      exit(1); /* terminate with an error code */
   }
```

```
/* draw the circle */


ellipse1(300,200,250,150);

ellipse1(300,200,240,140);

ellipse1(300,200,230,130);

ellipse1(300,200,220,120);

ellipse1(300,200,210,110);

 ellipse1(300,200,200,100);

 ellipse1(300,200,190,90);

ellipse1(300,200,180,80);

for(i=0;i<25;i++)

{

        circle(300,200,i);

        setcolor(RED);

}

        for(i=0;i<5;i++)

        {

                circle(280,120,i);

                setcolor(RED);

        }

        for(i=0;i<5;i++)

        {

                circle(260,112,i);

                setcolor(GREEN);

        }

        for(i=0;i<5;i++)
```

```
{
         circle(230,106,i);
         setcolor(BLUE);
}
for(i=0;i<5;i++)
{
        circle(220,96,i);
        setcolor(RED);
}
for(i=0;i<5;i++)
{
        circle(210,90,i);
        setcolor(YELLOW);
}
for(i=0;i<5;i++)
{
        circle(136,110,i);
        setcolor(GREEN);
}
for(i=0;i<5;i++)
{
        circle(105,120,i);
        setcolor(RED);
}
        for(i=0;i<5;i++)
{
        circle(400,337,i);
```

```
                        setcolor(BLUE);
            }
  /* clean up */
  getch();
  closegraph();
  return 0;
}

void ellipse1(float xc,float yc,float rx, float ry)
{
 float p1=(ry*ry-rx*rx*ry+rx*rx/4);
 float p2;
 float x=0,y=ry;

 while(2*ry*ry*x<=2*rx*rx*y)
 {
  if(p1<0)
  {
   x=x+1;
   y=y;
   p1=p1+2*ry*ry*x+ry*ry;
  }
  else
  {
   x++;
   y--;
   p1=p1+2*ry*ry*x+ry*ry-2*rx*rx*y;
 }
 putpixel(xc+x,yc+y,WHITE);
 putpixel(xc+x,yc-y,WHITE);
 putpixel(xc-x,yc+y,WHITE);
 putpixel(xc-x,yc-y,WHITE);
}
p2=(ry*ry*(x+1/2)*(x+1/2)+rx*rx*(y-1)*(y-1)-rx*rx*ry*ry);

while(y!= 0)
{
 if(p2>0)
 {
  x=x;
  y=y-1;
  p2=p2-2*rx*rx*y+rx*rx;
 }
 else
 {
  x++;
  y--;
  p2=p2-2*rx*rx*y+rx*rx+2*ry*ry*x;
 }
 putpixel(xc+x,yc+y,WHITE);
 putpixel(xc+x,yc-y,WHITE);
 putpixel(xc-x,yc+y,WHITE);
 putpixel(xc-x,yc-y,WHITE);
```

```
}
//*/
}
```

OUTPUT: