Lab Exercise Date 23-01-2019

1. Write a C program to draw ellipse with major axis as 100, minor axis as 50 and centre as (200, 200) using Mid-point's ellipse drawing algorithm.

2. Write a C program to draw a polygon with coordinates: (100, 30), (50, 70), (70, 70), (30,150), (150,150) and fill it using Scan-Line Polygon Filling Algorithm

Instructions

Steps of Midpoint Ellipse Algorithm

Midpoint ellipse algorithm plots(finds) points of an ellipse on the first quadrant by dividing the quadrant into two regions. Each point(x, y) is then projected into other three quadrants (-x, y), (x, -y), (-x, -y) i.e. it uses 4-way symmetry.

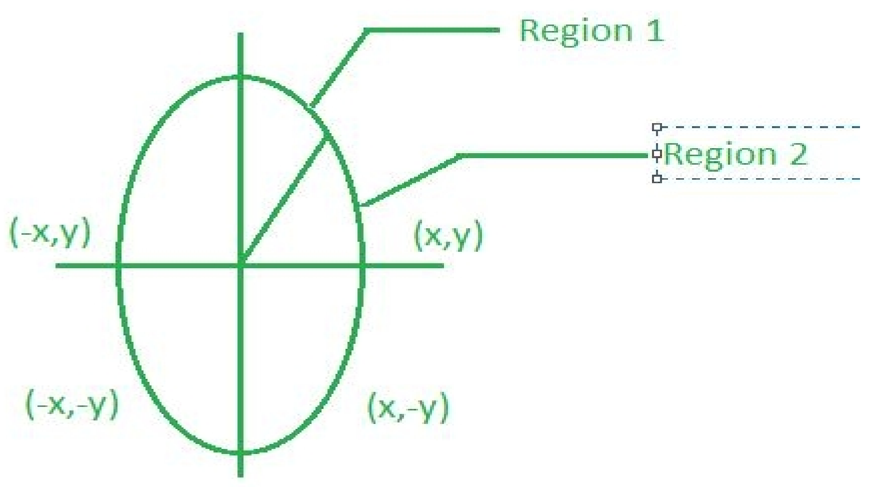
Function of ellipse:

$f_{ellipse}(x, y)=r_y^2x^2+r_x^2y^2-r_x^2r_y^2$

$f_{ellipse}(x, y)<0$ *then (x, y) is inside the ellipse.*

$f_{ellipse}(x, y)>0$ *then (x, y) is outside the ellipse.*

$f_{ellipse}(x, y)=0$ *then (x, y) is on the ellipse*



1. Input r_x, r_y and ellipse center (x_c, y_c)
2. Assume ellipse to be centered at origin and obtain the first point on it as: $(x_0, y_0) = (0, r_y)$
3. Obtain the initial decision parameter for region 1 as: $p1_0 = r_y^2 + 1/4r_x^2 - r_x^2r_y$
4. For every x_k position in region 1, starting at k=0, perform the following tests :

   If $p1_k<0$ then the next point along the ellipse is (x_{k+1}, y_k) and $p1_{k+1} = p1_k + 2r_y^2x_{k+1} + r_y^2$

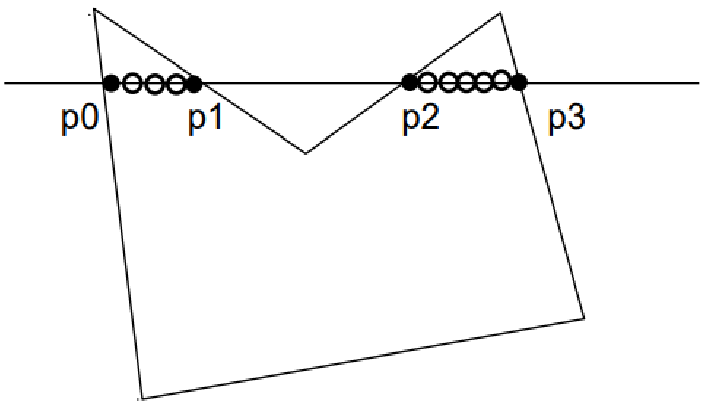
   Else, the next point is $(x_k + 1, y_k - 1)$ and $p1_{k+1} = p1_k + 2r_y^2x_{k+1} – 2r_x^2y_{k+1} + r_y^2$

5. Obtain the initial value in region 2 using the last point (x_0, y_0) of region 1 as:
   $p2_0=r_y^2(x_0+1/2)^2+r_x^2 (y_0-1)^2-r_x^2r_y^2$
6. At each y_k in region 2 starting at k =0 perform the following task.
   If $p2_k<0$ the next point is (x_k, y_{k+1}) and $p2_{k+1}=p2_k-2r_x^2y_{k+1}+r_x^2$
   Else, the next point is (x_{k+1}, y_{k-1}) and $p2_{k+1}=p2_k+2r_y^2x_{k+1} -2r_x^2y_{k+1}+r_x^2$
7. Now obtain the symmetric points in the three quadrants and plot the coordinate value as:
   $x=x+ x_c$, $y=y+ y_c$
8. Repeat the steps for region 1 until $2r_y^2x >= 2r_x^2y$

Steps of Scan- Line Polygon Filling Algorithm:

This algorithm works by intersecting scanline with polygon edges and fills the polygon between pairs of intersections. The following steps depict how this algorithm works.

1. Draw a polygon with given coordinates
2. Intersect scanline with polygon edges
3. Fill between pairs of intersections



Basic algorithm:

For y = ymin to ymax

1) intersect scanline y with each edge

2) sort intersections by increasing x [p0,p1,p2,p3]

3) fill pairwise (p0 −> p1, p2−> p3, ...)

Special handling to improve the performance:

a) Make sure to only fill the interior pixels

b) Intersection has an integer X coordinate

c) Intersection is an edge end point