

## United Technical College, Bharatpur, Chitwan

## Lab3

Midpoint Circle Generation Algorithm

Date Assigned: Dec 05, 2022

Date Due: Dec 12, 2022

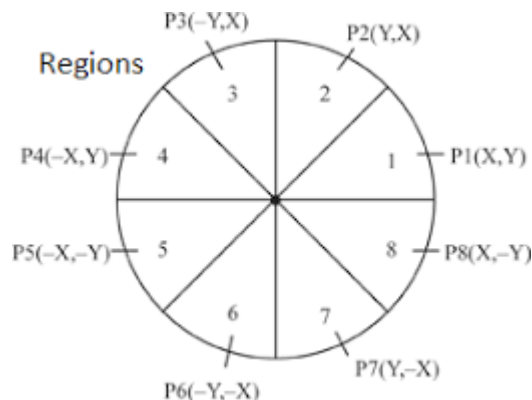
Using Midpoint circle generation algorithm which is a variant of Bresenham's line algorithm, write a C-program to generate pixel activation list for drawing a circle with a given center of circle  $P(x,y)$  and a radius  $r$ ?

**Aim:** To implement midpoint circle generation algorithm or Bresenham's circle algorithm for drawing a circle of given center  $(x,y)$  and radius  $r$ .

**Description:**

Circles have the property of being highly symmetrical, which is handy when it comes to drawing them on a display screen.

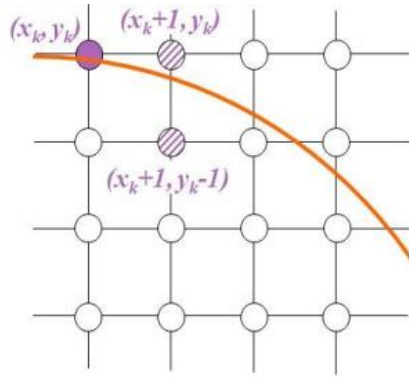
- We know that there are 360 degrees in a circle. First we see that a circle is symmetrical about the x axis, so only the first 180 degrees need to be calculated.
- Next we see that it's also symmetrical about the y axis, so now we only need to calculate the first 90 degrees.
- Finally we see that the circle is also symmetrical about the 45 degree diagonal axis, so we only need to calculate the first 45 degrees.
- We only need to calculate the values on the border of the circle in the first octant. The other values may be determined by symmetry.



**Figure: 8 way symmetry of the circle**

Bresenham's circle algorithm calculates the locations of the pixels in the first 45 degrees. It assumes that the circle is centered on the origin. So for every pixel  $(x,y)$  it calculates, we draw a pixel in each of the eight octants of the circle. This is done till when the value of the y coordinate equals the x coordinate. The pixel positions for determining symmetry are given in the below algorithm.

## United Technical College, Bharatpur, Chitwan



- Assume that we have just plotted point  $(x_k, y_k)$
- The next point is a choice between  $(x_k+1, y_k)$  and  $(x_k+1, y_k-1)$
- We would like to choose the point that is nearest to the actual circle
- So we use decision parameter here to decide.

8

### Algorithm:

1. Input radius  $r$  and circle center  $(x_c, y_c)$ , then set the coordinates for the first point on the circumference of a circle centered on the origin as:

$$(x_c, y_c) = (0, r)$$

2. Calculate the initial value of the decision parameter as:

$$P_0 = \frac{5}{4} - 1$$

3. Starting with  $k = 0$  at each position  $x_k$ , perform the following test. If  $p_k < 0$ , the next point along the circle centered on  $(0,0)$  is  $(x_k+1, y_k)$  and:

$$P_{k+1} = P_k + 2 * x_{k+1} + 1$$

Otherwise the next point along the circle  $(x_k+1, y_k-1)$  and:

$$P_{k+1} = P_k + 2 * x_{k+1} + 1 - 2 * y_{k+1}$$

4. Determine symmetry points in the other seven octants
5. Move each calculated pixel position  $(x, y)$  onto the circular path centered  $(x_c, y_c)$  to plot the coordinate values:
6. Repeat steps 3 to 5 until  $x \geq y$ .

**Midpoint Circle Drawing Algorithm C Program:** Students are expected to write c program in lab.

