**Experiment No.-3**

**Student Name: PUJA KUMARI UID: 20BCA1448**

**Branch: BCA Section/Group: 20BCA5-B**

**Semester: 5th Date of Performance: 1.9.22**

**Subject Name: COMPUTER GRAHICS LAB Subject Code: 20CAP-316**

1. **Aim/Overview of the practical:** Scan Convert a circle with center (300, 400) and radius 120 using Bresenham’s circle algorithm.
2. **Task to be done:** Using the algorithm draw the circle.
3. **Concept used:**

## **Bresenham's Circle Algorithm:**

**Step1:**Start Algorithm

**Step2:** Declare p, q, x, y, r, d variables  
        p, q are coordinates of the center of the circle  
        r is the radius of the circle

**Step3:** Enter the value of r

**Step4:** Calculate d = 3 - 2r

**Step5:** Initialize       x=0  
          &nbsy= r

**Step6:** Check if the whole circle is scan converted  
            If x > = y  
            Stop

**Step7:** Plot eight points by using concepts of eight-way symmetry. The center is at (p, q). Current active pixel is (x, y).  
                putpixel (x+p, y+q)  
                putpixel (y+p, x+q)  
                putpixel (-y+p, x+q)  
                putpixel (-x+p, y+q)  
                putpixel (-x+p, -y+q)  
                putpixel (-y+p, -x+q)  
                putpixel (y+p, -x+q)  
                putpixel (x+p, -y-q)

**Step8:** Find location of next pixels to be scanned  
            If d < 0  
            then d = d + 4x + 6  
            increment x = x + 1  
            If d ≥ 0  
            then d = d + 4 (x - y) + 10  
            increment x = x + 1  
            decrement y = y - 1

**Step9:** Go to step 6

**Step10:** Stop Algorithm

1. **Steps/Commands involved to perform practical:**

#include <graphics.h>

#include <stdlib.h>

#include <stdio.h>

#include <conio.h>

#include <math.h>

void CirclePlot(int xc,int yc,int x,int y)

{

putpixel(x+xc,y+yc,YELLOW);

putpixel(x+xc,-y+yc,YELLOW);

putpixel(-x+xc,-y+yc,YELLOW);

putpixel(-x+xc,y+yc,YELLOW);

putpixel(y+xc,x+yc,YELLOW);

putpixel(y+xc,-x+yc,YELLOW);

putpixel(-y+xc,-x+yc,YELLOW);

putpixel(-y+xc,x+yc,YELLOW);

}

void BresenhamCircle(int xc,int yc,int r)

{

int x=0,y=r,d=3-(2\*r);

CirclePlot(xc,yc,x,y);

while(x<=y)

{

if(d<=0)

{

d=d+(4\*x)+6;

}

else

{

d=d+(4\*x)-(4\*y)+10;

y=y-1;

}

x=x+1;

CirclePlot(xc,yc,x,y);

}

}

int main(void)

{

int xc,yc,r,gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "");

errorcode = graphresult();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to halt:");

getch();

exit(1);

}

printf("\t\*\*\*\*\*\*\*\*\*BRESENGHAM'S CIRCLE DRAWING ALGORITHM\*\*\*\*\*\*\*\*\*\t\n\n");

printf("Enter the values of xc and yc :");

scanf("%d%d",&xc,&yc);

printf("Enter the value of radius :");

scanf("%d",&r);

BresenhamCircle(xc,yc,r);

printf("\n\n\t\t\t\t\t\tSubmitted By : PUJA KUMARI (20BCA1448)\t\t\t\t\t\t\n\n");

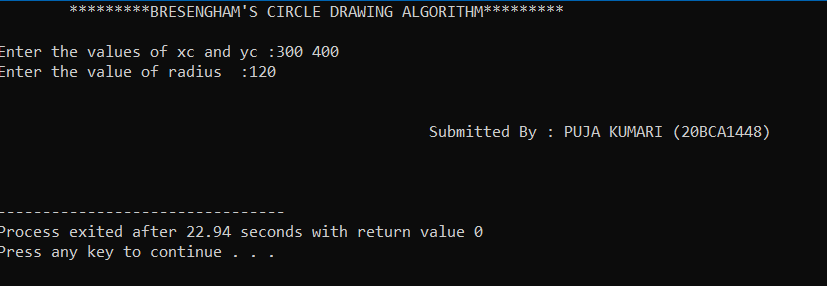
getch();

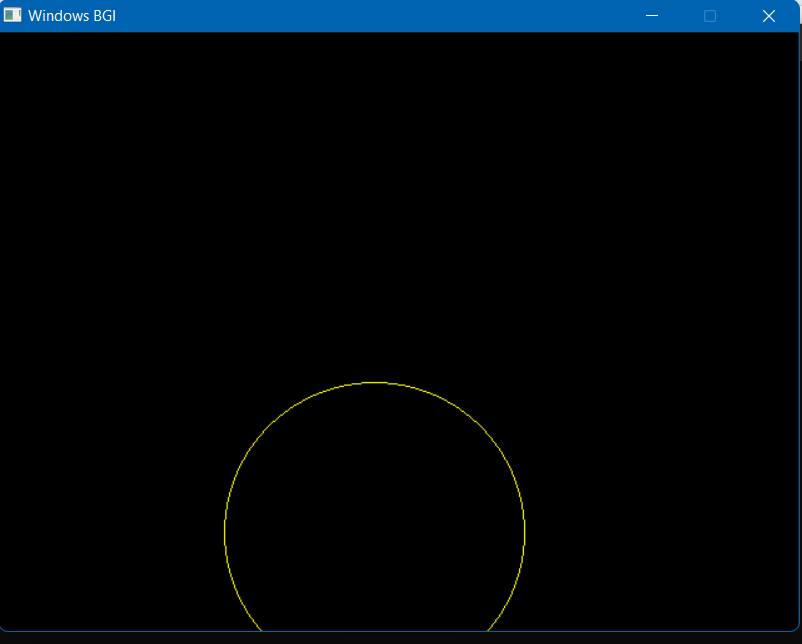
closegraph();

return 0;

}

1. **Result/Output/Writing Summary:**

****

****

**Learning outcomes (What I have learnt):**

1. I have learnt about how to draw circle using Bresenham’s circle drawing algothrim.

1. I have learnt about the advantages of using this algorithm.

3. Bresenham’s Circle Drawing Algorithm is a circle drawing algorithm that selects the nearest pixel position to complete the arc. The unique part of this algorithm is that is uses only integer arithmetic which makes it, significantly, faster than other algorithms using floating point arithmetic in classical processors.

Evaluation Grid:

|  |  |  |  |
| --- | --- | --- | --- |
| Sr. No. | Parameters | Marks Obtained | Maximum Marks |
| 1. | Worksheet |  | 10 |
| 2. | Demonstration/Performance /Pre Lab Quiz |  | 5 |
| 3. | Post Lab Quiz |  | 5 |