



Semester	S.E. Semester IV – CMPN
Division & Batch	Div B Batch 3
Subject	Computer Graphics
Laboratory Teacher:	Divya Nimbalkar
Laboratory	M310A

Student Name	Atharva Sheramkar
Roll Number	24102B0069
Grade and Subject Teacher's Signature	

Experiment Number	3	
Experiment Title	Midpoint Circle Drawing algorithm	
Problem Statement	Clearly depict every octant drawn leading to full circle plotting	
Resources / Apparatus Required	Hardware: Desktop	Software: Dev C++

Code	<pre> #include <graphics.h> #include <stdio.h> #include <conio.h> void midcircle(int xc, int yc, int r) { int dx, dy, x = 0, y = r; int p=(5/4)-r; int i=0; putpixel(xc,yc,WHITE); while(x <= y) { //plot points //1st delay(10); putpixel(xc+x,yc-y,BLUE); //2nd delay(10); putpixel(xc+y,yc-x,MAGENTA); //3rd delay(10); putpixel(xc+y,yc+x,CYAN); //4th delay(10); putpixel(xc+x,yc+y,RED); //5th delay(10); putpixel(xc-x,yc+y,GREEN); //6th delay(10); putpixel(xc-y,yc+x,BROWN); //7th delay(10); putpixel(xc-y,yc-x,LIGHTRED); //8th delay(10); putpixel(xc-x,yc-y,YELLOW); } } </pre>
------	---

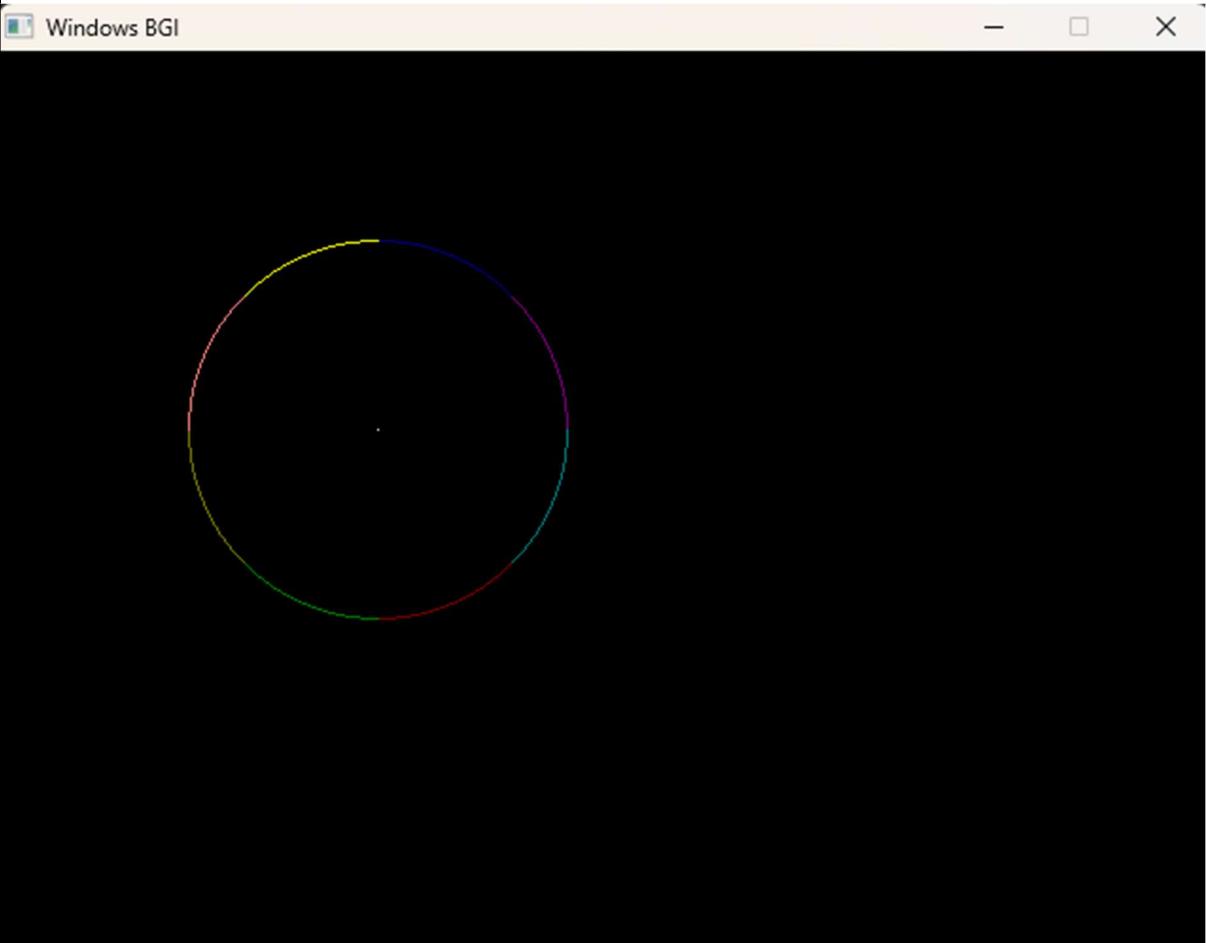
```
if(p<0)
{
    p=p + 2*x + 3;
}
else
{
    p=p + 2*x - 2*y + 5;
    //decrement y
    y--;
}
//increment x
x++;
}

}

int main() {
    int xc, yc,r;
    int gd = DETECT, gm;

    printf("Enter the coords of first point (xc yc) and radius : ");
    scanf("%d %d %d", &xc, &yc, &r);

    initgraph(&gd, &gm, (char*)"");
    midcircle(xc, yc, r);
    getch();
    closegraph();
    return 0;
}
```



Conclusion	In this experiment, the Midpoint Circle Generation Algorithm was utilized to render a circular arc on the screen by calculating and plotting pixel coordinates incrementally. By leveraging the eight-way symmetry of a circle, the algorithm efficiently calculates points for one-forty-fifth degree octant and reflects them across all other sectors, significantly reducing computational overhead.
------------	--

