**Lab-2**

*Course title: Computer Graphics Lab*

*Course code: CSE-304*

*3rd Year 1st Semester Examination 2022*

**Date of Submission**: 04 June 2023

****

**Submitted to-**

**Dr. Mohammad Shorif Uddin**

*Professor*

**Dr. Morium Akter**

*Associate Professor*

*Department of Computer Science and Engineering*

*Jahangirnagar University*

*Savar, Dhaka-1342*

|  |  |  |  |
| --- | --- | --- | --- |
| **Sl** | Class Roll | Exam Roll | Name |
| 01 | 374 | 202186 | Mahfuz Anam |

**1.Midpoint Circle Algorithm**

**Source code**

#include<bits/stdc++.h>

#include<graphics.h>

using namespace std;

void midpoint\_circle(int cx, int cy, int r)

{

int x,y,decision;

x=0,y=r,decision=1-r;

while (x <= y)

{

putpixel(cx + x, cy + y, WHITE);

putpixel(cx - x, cy + y, WHITE);

putpixel(cx + x, cy - y, WHITE);

putpixel(cx - x, cy - y, WHITE);

putpixel(cx + y, cy + x, WHITE);

putpixel(cx - y, cy + x, WHITE);

putpixel(cx + y, cy - x, WHITE);

putpixel(cx - y, cy - x, WHITE);

if (decision < 0) decision += (2\*x+3);

else

{

decision += (2 \* (x - y) + 5);

y--;

}

x++;

}

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int cx,cy,r;

cout<<"Midpoint Circle Algorithm"<<endl;

cout << "Enter center coordinates (x, y) and radius of the circle: ";

cin >> cx >> cy >>r;

midpoint\_circle(cx, cy, r);

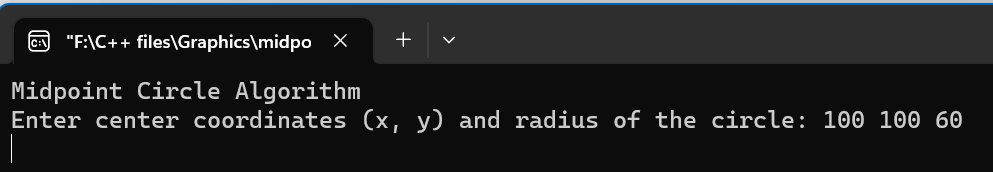
getch();

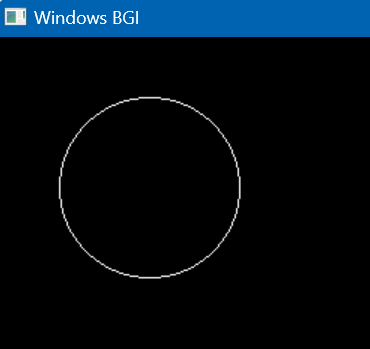
closegraph();

return 0;

}

**Output:**

****

****

**2.Scan conversion of an Ellipse**

**Source code**

#include <graphics.h>

#include <bits/stdc++.h>

using namespace std;

void EllipseScanConversion(int centerX, int centerY, int a, int b)

{

int x = 0;

int y = b;

int aSq = a \* a;

int bSq = b \* b;

int twoASq = 2 \* aSq;

int twoBSq = 2 \* bSq;

int decision = bSq - aSq \* b + aSq / 4;

while (twoBSq \* x < twoASq \* y)

{

putpixel(centerX + x, centerY + y, WHITE);

putpixel(centerX - x, centerY + y, WHITE);

putpixel(centerX + x, centerY - y, WHITE);

putpixel(centerX - x, centerY - y, WHITE);

if (decision <= 0)

{

x++;

decision += twoBSq \* x + bSq;

}

else

{

x++;

y--;

decision += twoBSq \* x - twoASq \* y + bSq;

}

}

decision = bSq \* (x + 0.5) \* (x + 0.5) + aSq \* (y - 1) \* (y - 1) - aSq \* bSq;

while (y >= 0)

{

putpixel(centerX + x, centerY + y, WHITE);

putpixel(centerX - x, centerY + y, WHITE);

putpixel(centerX + x, centerY - y, WHITE);

putpixel(centerX - x, centerY - y, WHITE);

if (decision > 0)

{

y--;

decision += aSq - twoASq \* y;

}

else

{

y--;

x++;

decision += twoBSq \* x - twoASq \* y + aSq;

}

}

}

int main()

{

int gd = DETECT, gm;

initgraph(&gd, &gm, "");

int centerX, centerY, a, b;

cout << "Enter center coordinates (x, y): ";

cin >> centerX >> centerY;

cout << "Enter semi-major axis (a): ";

cin >> a;

cout << "Enter semi-minor axis (b): ";

cin >> b;

EllipseScanConversion(centerX, centerY, a, b);

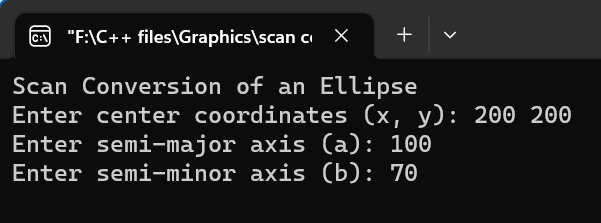
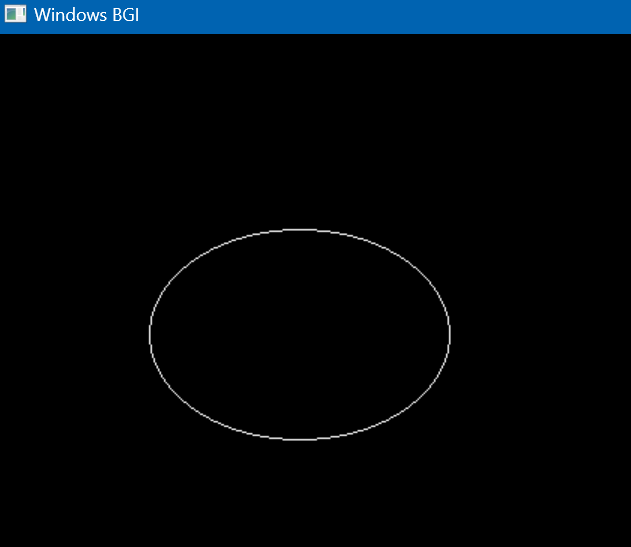
getch();

closegraph();

return 0;

}

**Output:**

** **