**ST. Xavier's College**

**Maitighar, Kathmandu**



**LAB ASSIGNMENT #5**

**Midpoint Circle Algorithm in C++ Builder**

# 

# SUBMITTED BY

Victor Kumar Sapkota

O13BSCCSIT047

Sem IV

**SUBMITTED TO**

Er. Anil K. Sah

Department of Computer Science

**STATEMENT: Implement midpoint circle algorithm in C++ builder**

**ALGORITHM:**

**Step 1:** Input radius r and circle centre (xc, yc), and obtain the first point on circle centered at origin as

(x0, y0) = (0, r).

**Step 2:** Calculate initial decision parameter

Po= 1 - r

**Step 3:** At each xk position, starting at k=0, perform the tests:

If Pk<0 next point along the center at (0, 0) is (xk+1, yk)

Pk+1= Pk+2 xk+1+1

Otherwise, the next point along the circle is (xk+1, yk-1)

Pk+1= Pk+2 xk+1+1-2 yk+1

Where 2 xk+1=2 xk+2 and 2 yk+1=2 yk+2

**Step 4:** Determine symmetry point on the other seven octants.

**Step 5:** Move each calculated positions (x,y) in to circle path centered at (xc, yc) as

x=x+xc and y=y+yc

**Step 6:** Repeat steps 3 through 5 until x≥y.

**Step 7:** Terminate.

**SOURCE CODE:**

//---------------------------------------------------------------------------

#include <vcl\vcl.h>

#pragma hdrstop

#include "circle.h"

//---------------------------------------------------------------------------

#pragma resource "\*.dfm"

TForm1 \*Form1;

//---------------------------------------------------------------------------

\_\_fastcall TForm1::TForm1(TComponent\* Owner)

: TForm(Owner)

{

}

//---------------------------------------------------------------------------

void \_\_fastcall TForm1::Button1Click(TObject \*Sender)

{

int xc,yc,r;

xc=StrToInt(Edit2->Text);

yc=StrToInt(Edit3->Text);

r=StrToInt(Edit1->Text);

int p,x,y;

x=0;

y=r;

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,0,255);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,255,255);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(255,255,255);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(255,255,0);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(255,0,0);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(127,127,127);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(127,255,0);

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,255,127); p=1-r;

while(x<y)

{

if(p<0)

{

x=x+1;

p=p+2\*x+1;

}

else

{

x=x+1;

y=y-1;

p=p+2\*(x-y)+1;

}

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(0,0,255);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,255,255);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(255,255,255);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(255,255,0);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(255,0,0);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(127,127,127);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(127,255,0);

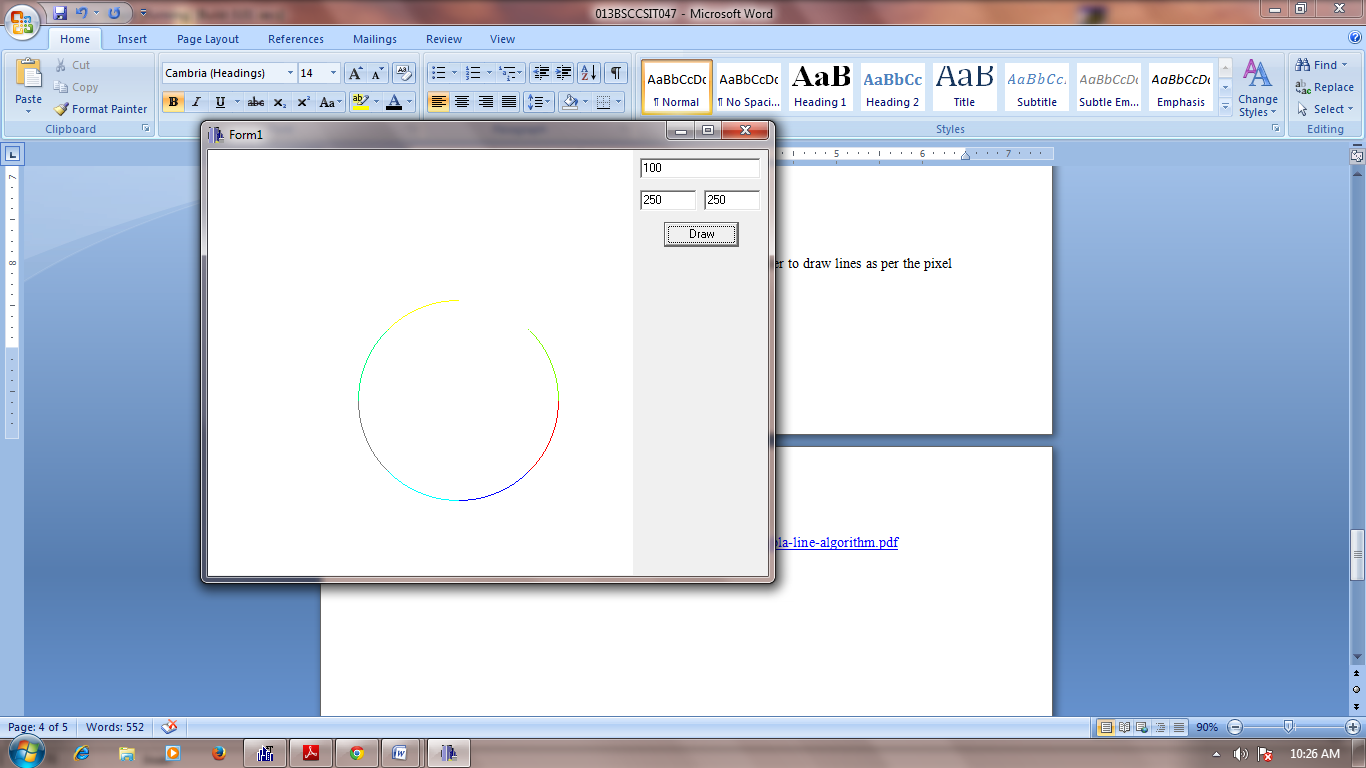
Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,255,127);

}

}

//---------------------------------------------------------------------------

**OUTPUT SCREENSHOT**

****

**CONCLUSION:**

The midpoint circle algorithm was implemented in C++ Builder to draw a circle as per the centre coordinates and radius input by the user. Different colors were used for arcs drawn using the symmetric property of circle.

**REFERENCE:**

<http://www.eazynotes.com/notes/computer-graphics/algorithms/midpoint-circle-algorithm.pdf>