ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Computer Graphics Lab Assignment #5**

**Submitted by:**

Arun Sanjel

013BSCCSIT010

**Submitted to:**

|  |  |
| --- | --- |
| Er. Anil Sah  Lecturer, St. Xavier’s College |  |

**STATEMENT**

“Draw a circle using midpoint circle Algorithm where radius and center are provided by user.”

**ALGORITHM**:

1. Input the radius and circle centre(xc,yc) and obtain the first point on circle centers at :

(X0, y0)= (0, r).

1. Calculate initial decision parameter

P0=1-r

1. At each x, position starting at k=0 perform the tests

If p0<0 next point along the circle centre at (0, 0) is (xk+1,yk)

Pk+1=pk+2xk+1+1)

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

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

1. Determine symmetry point on the other seven octants
2. Move each calculated pixels positions (x,y) in to circle path centered at (xc,yc) as

x=x+xc

y=y+yc

1. Repeat 3 through 5 until x>=y

**SOURCE CODE:**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "midCir.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

void drawCircle(int x,int y, int xC,int yC,TCanvas \*Canvas){

Canvas->Pixels[xC+x][yC+y] = RGB(255,0,0);

//Symmetric Properties

Canvas->Pixels[xC+y][yC+x] = RGB(0,255,0);

Canvas->Pixels[xC+y][yC-x] = RGB(0,0,255);

Canvas->Pixels[xC+x][yC-y] = RGB(155,155,155);

Canvas->Pixels[xC-x][yC-y] = RGB(155,155,0);

Canvas->Pixels[xC-y][yC-x] = RGB(0,155,155);

Canvas->Pixels[xC-y][yC+x] = RGB(0,0,0);

Canvas->Pixels[xC-x][yC+y] = RGB(255,0,255);

}

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

{

outPanel->Canvas->FillRect(ClientRect);

int x,y,xC,yC;

int P,r;

xC = StrToInt(inCx->Text);

yC = StrToInt(inCy->Text);

r = StrToInt(inRadius->Text);

P = 1 -r;

x=0;

y=r;

while(x<=y){

x++;

if(P<0){

P+=2\*x+1;

}

else{

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

y--;

}

drawCircle(x,y,xC,yC,outPanel->Canvas);

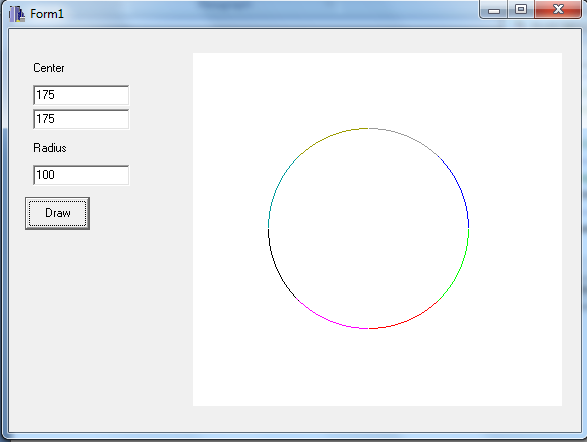
}

}

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

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

**OUTPUT**



**CONCLUSION:**

Thus, circle was drawn using midpoint algorithm where user provided center and radius of the circle using C++ builder.