ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**Computer Graphics Lab Assignment #5**

**Submitted by:**

Bishal Pandey

013BSCCSIT016

4th sem

**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 "Unit1.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

void \_\_fastcall TForm1::circleClick(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(255,0255,0);

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(0,255,0);

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

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

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

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

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(255,0,255);

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

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

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

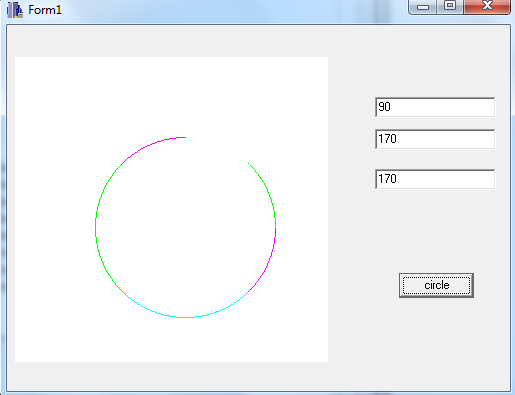
Image1->Canvas->Pixels[xc-y][yc-x]=RGB(0,255,0);

}

}

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

**OUTPUT SCREENSHOT**



**CONCLUSION:**

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