**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**COMPUTER GRAPHICS**

**LAB ASSIGNMENT#5**

**Submitted by:**

Binod Paneru

013BSCCSIT014

**Submitted to:**

|  |  |
| --- | --- |
| **Er. Anil K. Sah** |  |

Lecturer

Department of Computer Science

Date of submission: 21st August, 2015

**Statement: Implement midpoint circle algorithm.**

**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=

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 3 through 5 until x≥y.

Step 7: terminate.

**SOURCE CODE**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

int xc,yc,r;

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

void \_\_fastcall TForm1::Edit3Change(TObject \*Sender)

{

r=StrToInt(Edit3->Text);

}

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

void \_\_fastcall TForm1::Go\_For\_CircleClick(TObject \*Sender)

{

int p,x,y;

x=0;

y=r;

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(10,100,255);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(0,199,100);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(255,59,47);

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

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(100,100,0);

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

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

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(100,255,0); 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(255,0,255);

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

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(255,100,255);

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

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(255,0,100);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(127,275,127);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(127,255,100);

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(152,255,127);

}

}

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

void \_\_fastcall TForm1::Edit1Change(TObject \*Sender)

{

xc=StrToInt(Edit1->Text);

}

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

void \_\_fastcall TForm1::Edit2Change(TObject \*Sender)

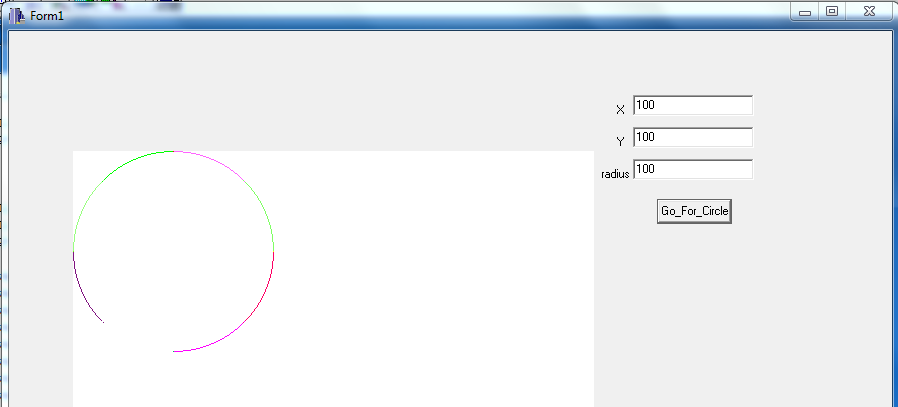
{

yc=StrToInt(Edit2->Text);

}

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

**Output:**

****

**Conclusion:**

Hence, midpoint circle algorithm was implemented by using the c++ builder.