**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU



Computer Graphics

Assignment #5

Submitted By:

Abhishek Tamrakar

013BSCCSIT003

2nd year/ 4th semester

Submitted to:

|  |  |
| --- | --- |
| Er. Anil K. Sah  Lecturer  Department of Computer Science |  |

**STATEMENT**

**“TO 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=5/4-1

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: End.

**SOURCE CODE:**

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

#include <vcl\vcl.h>

#pragma hdrstop

int xc,yc,R;

#include "cir.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

xc = StrToInt(Edit1->Text);

}

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

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

{

yc = StrToInt(Edit2->Text);

}

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

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

{

R=StrToInt(Edit3->Text);

}

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

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

{

int p0,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(127,255,255);

Image1->Canvas->Pixels[xc-x][yc-y] = RGB(225,0,255);

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

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

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

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

p0=1-R;

while(x<y)

{

if(p0<0)

{

x=x+1;

p0=p0+2\*x+1;

}

else

{

x=x+1;

y=y-1;

p0=p0+2\*x+1-2\*y;

}

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(127,255,255);

Image1->Canvas->Pixels[xc-x][yc-y] = RGB(225,0,255);

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

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

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

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

}

}

**OUTPUT SCREENS:**

****

**CONCLUSION:**

Hence, the program to implement the circle using Mid-point algorithm in C++ builder was implemented.