ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



COMPUTER GRAPHICS

Lab Assignment #5

**Submitted by:**

Urendra Khakurel  
013BSCCSIT045

**Submitted to:**

Er. Anil K Sah

Lecturer  
Department of Computer Science

**Date of Submission: Aug 14 , 2015**

**STATEMENT: Implement DDA line drawing algorithm in C++ builder**

**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)

{

}

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

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

{

int r,xc,yc,x,y,pk;

r=StrToInt(Edit1->Text);

xc=StrToInt(Edit2->Text);

yc=StrToInt(Edit3->Text);

x=0;

y=r;

pk=1-r;

Image1->Canvas->Pixels[x][y]=RGB(100,250,250);

while(x<y)

{

if(pk<0)

{

pk=pk+2\*(x+1)+1;

x++;

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(100,250,250);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(10,250,150);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(100,250,200);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(200,0,0);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(0,200,0);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(0,0,200);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(57,56,250);

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(59,200,150);

}

else

{

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

x++;

y--;

Image1->Canvas->Pixels[xc+x][yc+y]=RGB(100,250,250);

Image1->Canvas->Pixels[xc-x][yc+y]=RGB(10,250,150);

Image1->Canvas->Pixels[xc+x][yc-y]=RGB(100,250,200);

Image1->Canvas->Pixels[xc-x][yc-y]=RGB(200,0,0);

Image1->Canvas->Pixels[xc+y][yc+x]=RGB(0,200,0);

Image1->Canvas->Pixels[xc-y][yc+x]=RGB(0,0,200);

Image1->Canvas->Pixels[xc+y][yc-x]=RGB(57,56,250);

Image1->Canvas->Pixels[xc-y][yc-x]=RGB(60,200,150);

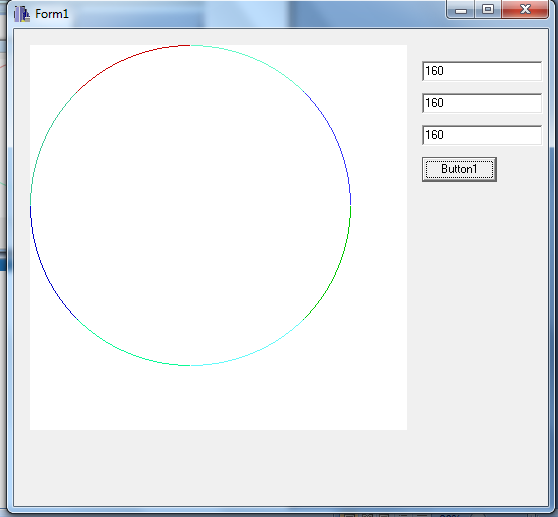
}

}

}

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

**OUTPUT**

****