**ST.XAVIER’S COLLEGE**

**(Affiliated to Tribhuwan Universitry)**

Maitighar, Kathmandu



Computer Graphics

Assignment #3

Submitted By:

Dikita Tuladhar

013BSCCSIT018

2nd year/ 4th semester

Submitted to:

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

Date of Submission: 21st August, 2015

**DRAW A CIRCLE.**

**ALGORITHM:**

Step 1: Set X = 0 and Y = R

Step 2: Set P = 1 – R

Step 3: Repeat While (X < Y)

Step 4: Call Draw Circle (Xc, Yc, X, Y)

Step 5: Set X = X + 1

Step 6: If (P < 0) Then

Step 7: P = P + 2X + 6

Step 8: Else

Step 9: Set Y = Y – 1

Step 10: P = P + 2(X – Y) + 1 [End of If]

Step 11: Call Draw Circle (Xc, Yc, X, Y) [End of While]

Step 12: Exit

**SOUTRCE CODE:**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "cir.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int x0,y0,r;

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

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

: TForm(Owner)

{

}

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

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

{

x0=Edit1->Text.ToInt();

}

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

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

{

y0=Edit2->Text.ToInt();

}

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

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

{

r=StrToInt(Edit3->Text);

}

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

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

{

Image1->Canvas->MoveTo(x0,y0);

int p0,x,y;

x=0;

y=r;

Image1->Canvas->Pixels[x+x0][y+x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-x0][y+y0] = RGB(0,0,225);

Image1->Canvas->Pixels[x+x0][y-y0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-x0][y-y0] = RGB(0,0,225);

Image1->Canvas->Pixels[x+y0][y+x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x+y0][y-x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-y0][y+x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-y0][y-x0] = RGB(0,0,225);

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[x+x0][y+x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-x0][y+y0] = RGB(0,0,225);

Image1->Canvas->Pixels[x+x0][y-y0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-x0][y-y0] = RGB(0,0,225);

Image1->Canvas->Pixels[x+y0][y+x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x+y0][y-x0] = RGB(0,0,225);

Image1->Canvas->Pixels[x-y0][y+x0] = RGB(0,0,225);

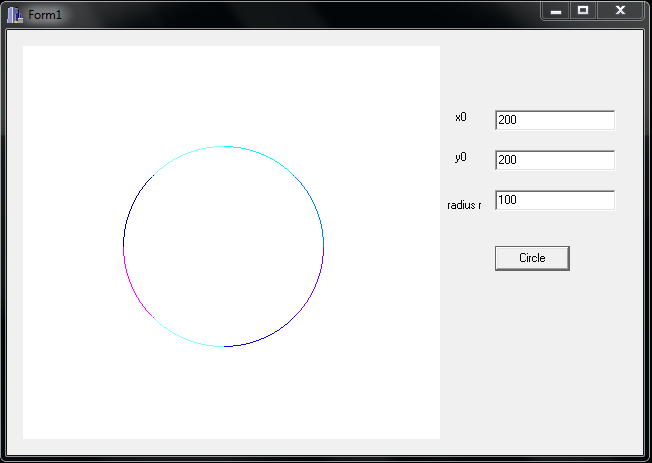
Image1->Canvas->Pixels[x-y0][y-x0] = RGB(0,0,225);

}

}

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

**OUPUT SCREEN:**

****

**CONCLUSION:**

Hence, the mid-point algorithm to make circle was implemented by using C++ builder.