**ST.XAVIER’S COLLEGE**

MAITIGHAR, KATHMANDU

****

**Database Management System**

Assignment #5

Submitted By:

Aashish Raj Shrestha

013BSCCSIT002

2nd year/ 4th semester

Submitted to:

|  |  |
| --- | --- |
| Er. Anil Shah  Lecturer  Department of Computer Science |  |

1. **Statement:**

Draw a Circle using Mid-Point Circle Algorithm

1. **Algorithm:**

**Mid-Point Circle ( Xc, Yc, R):**

Description: Here Xc and Yc denote the x – coordinate and y – coordinate of the center of the

circle. R is the radius.

1. Set X = 0 and Y = R

2. Set P = 1 – R

3. Repeat While (X < Y)

4. Call Draw Circle(Xc, Yc, X, Y)

5. Set X = X + 1

6. If (P < 0) Then

7. P = P + 2X + 6

8. Else

9. Set Y = Y – 1

10. P = P + 2(X – Y) + 1

[End of If]

11. Call Draw Circle(Xc, Yc, X, Y)

[End of While]

12. Exit

**Draw Circle (Xc, Yc, X, Y):**

1. Call PutPixel(Xc + X, Yc, + Y)

2. Call PutPixel(Xc - X, Yc, + Y)

3. Call PutPixel(Xc + X, Yc, - Y)

4. Call PutPixel(Xc - X, Yc, - Y)

5. Call PutPixel(Xc + Y, Yc, + X)

6. Call PutPixel(Xc - Y, Yc, + X)

7. Call PutPixel(Xc + Y, Yc, - X)

8. Call PutPixel(Xc - Y, Yc, - X)

9. Exit

1. **Source Code:**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "MIDCIRCLE.h"

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

#pragma resource "\*.dfm"

int x0,y0,radius;

TForm1 \*Form1;

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

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

: TForm(Owner)

{

}

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

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

{

x0=StrToInt(Edit1->Text);

}

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

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

{

y0=StrToInt(Edit2->Text);

}

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

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

{

radius=StrToInt(Edit3->Text);

}

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

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

{

int p0,x,y;

x=0;

y=radius;

Image1->Canvas->Pixels[x0+x][y0+y] = RGB(0,0,255);

Image1->Canvas->Pixels[x0-x][y0+y] = RGB(0,255,255);

Image1->Canvas->Pixels[x0+x][y0-y] = RGB(127,255,255);

Image1->Canvas->Pixels[x0-x][y0-y] = RGB(225,0,255);

Image1->Canvas->Pixels[x0+y][y0+x] = RGB(127,0,255);

Image1->Canvas->Pixels[x0+y][y0-x] = RGB(0,127,255);

Image1->Canvas->Pixels[x0-y][y0+x] = RGB(127,255,255);

Image1->Canvas->Pixels[x0-y][y0-x] = RGB(0,0,127);

p0=1-radius;

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[x0+x][y0+y] = RGB(0,0,255);

Image1->Canvas->Pixels[x0-x][y0+y] = RGB(0,255,255);

Image1->Canvas->Pixels[x0+x][y0-y] = RGB(127,255,255);

Image1->Canvas->Pixels[x0-x][y0-y] = RGB(225,0,255);

Image1->Canvas->Pixels[x0+y][y0+x] = RGB(127,0,255);

Image1->Canvas->Pixels[x0+y][y0-x] = RGB(0,127,255);

Image1->Canvas->Pixels[x0-y][y0+x] = RGB(127,255,255);

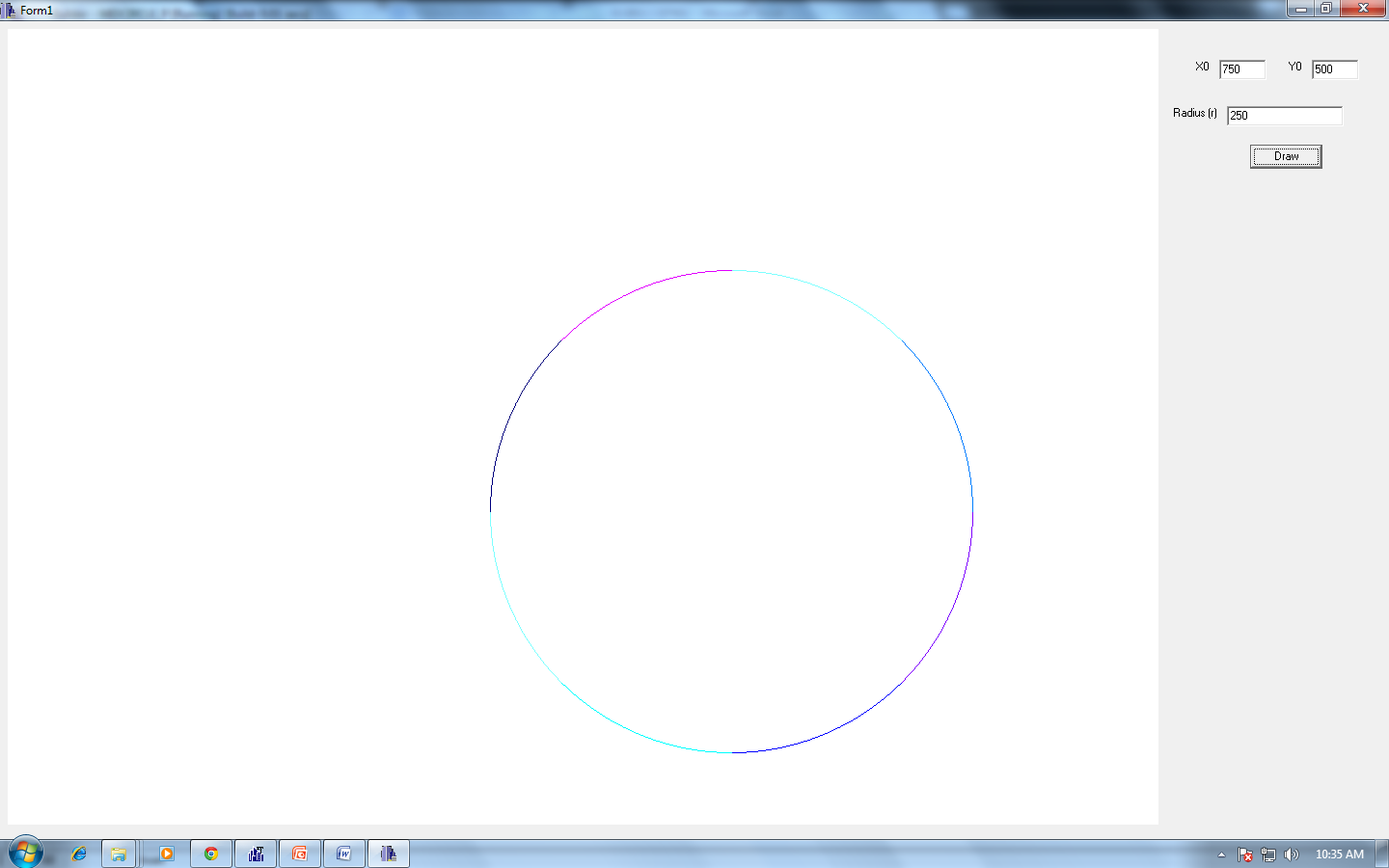
Image1->Canvas->Pixels[x0-y][y0-x] = RGB(0,0,127);

}

}

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

1. **Output:**

**

*Figure I: Implementation of Mid-point Circle Algorithm*

1. **Conclusion:**

Hence, mid-point circle algorithm was implemented to draw a circle using C++ Builder.