ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**Computer Graphics**

**Lab Assignment # 5**

**Mid-Point Circle Algorithm in C++ Builder**

**SUBMITTED BY:**

**Apil Neupane**

**013BSCCSIT008**

**SUBMITTED TO:**

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

Submission Date: 21thAugust 2015

**STATEMENT**

**Write a program to draw a circle using mid-point circle Algorithm in C++ Builder.**

**SOURCE CODE:**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "Circle.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

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

void draw(int x, int y, int xc, int yc);

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

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

: TForm(Owner)

{

}

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

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

{

r = StrToInt(Edit1->Text);

}

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

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

{

xc = StrToInt(Edit2->Text);

}

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

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

{

yc = StrToInt(Edit3->Text);

}

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

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

{

x = 0;

y = r;

p = 1-r;

Image1->Canvas->Pixels[xc+x][yc+y] = RGB(123,213,246);

Image1->Canvas->Pixels[xc-x][yc+y] = RGB(120,120,130);

Image1->Canvas->Pixels[xc+x][yc-y] = RGB(137,140,139);

Image1->Canvas->Pixels[xc-x][yc-y] = RGB(103,105,109);

Image1->Canvas->Pixels[xc+y][yc+x] = RGB(255,130,165);

Image1->Canvas->Pixels[xc-y][yc+x] = RGB(190,170,175);

Image1->Canvas->Pixels[xc+y][yc-x] = RGB(0,130,197);

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

while(x<y)

{

if(p<0)

{

x++;

p = p+2\*x+1;

}

else

{

x++;

y--;

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

}

Image1->Canvas->Pixels[xc+x][yc+y] = RGB(123,213,246);

Image1->Canvas->Pixels[xc-x][yc+y] = RGB(120,120,130);

Image1->Canvas->Pixels[xc+x][yc-y] = RGB(137,140,139);

Image1->Canvas->Pixels[xc-x][yc-y] = RGB(103,105,109);

Image1->Canvas->Pixels[xc+y][yc+x] = RGB(255,130,165);

Image1->Canvas->Pixels[xc-y][yc+x] = RGB(190,170,175);

Image1->Canvas->Pixels[xc+y][yc-x] = RGB(0,130,197);

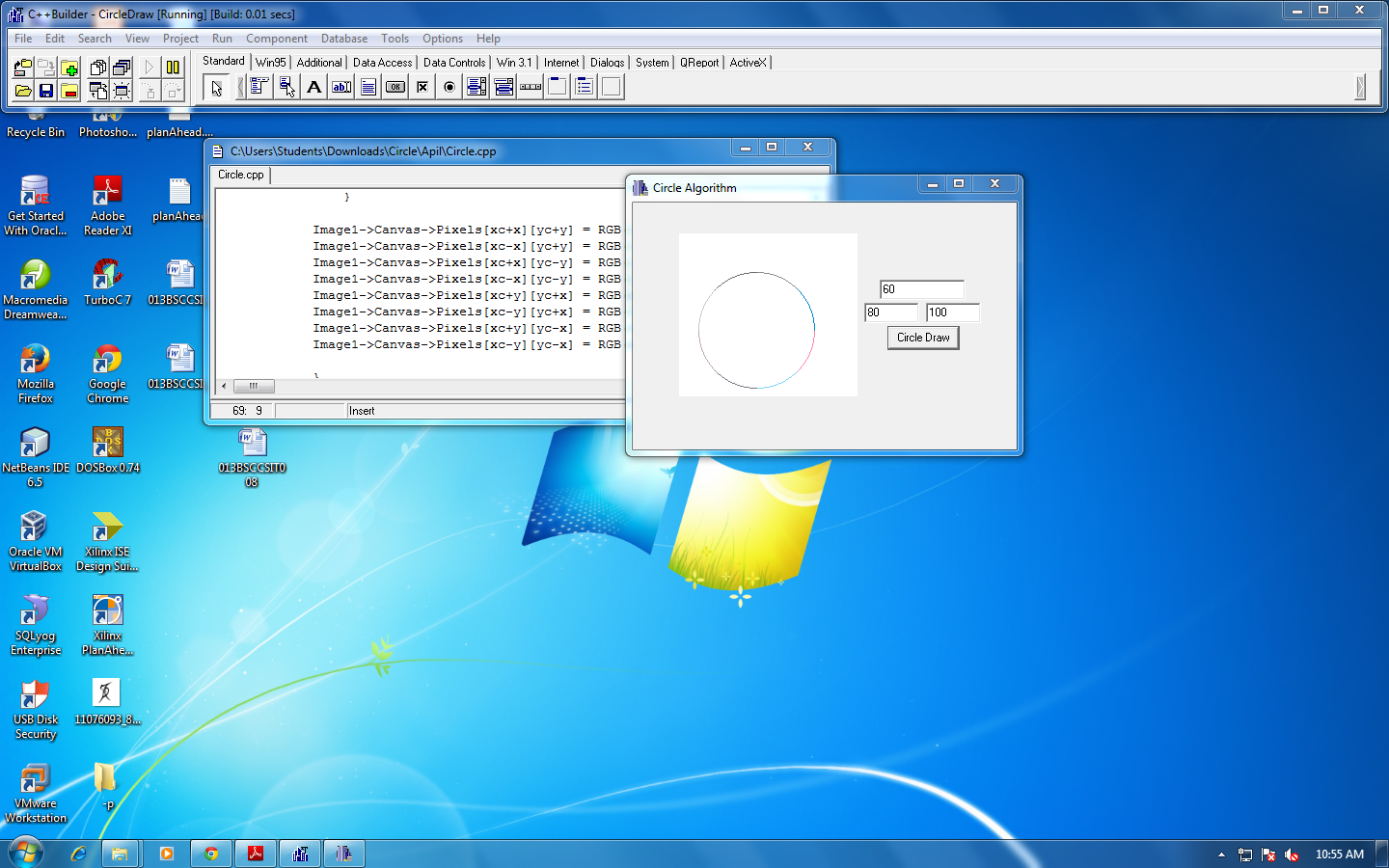
Image1->Canvas->Pixels[xc-y][yc-x] = RGB(200,200,200);

}

}

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

**OUTPUT:**



**CONCLUSION:**

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