ST. XAVIER’S COLLEGE

**(Affiliated to Tribhuvan University)**

Maitighar, Kathmandu



**CG Lab Assignment #4**

**Bresenham Line Algorithm in C++ Builder**

**Submitted by:**

Biraj Poudyal  
013BSCCSIT015

**Submitted to:**

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

**Date of Submission: August 18, 2015**

**Algorithm:**

1. Input the two line endpoints and the left endpoint at (x0,y0)
2. Load (x0,y0) into frame buffer, i.e. plot the first point.
3. Calculate constants 2∆x, 2∆y and obtain first decision parameter p0 = 2∆y – ∆x
4. At each xk along the line, starting at k = 0, perform the following test,

If pk < 0, next point is (xk+1,yk) and pk+1 = pk + 2∆y

Otherwise, next point to plot is (xk+1,yk+1) and pk+1 = pk + 2∆y – 2∆x

1. Repeat step 4 ∆x times.

**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 x1,y1,x2,y2;

x1=StrToInt(Edit1->Text);

y1=StrToInt(Edit2->Text);

x2=StrToInt(Edit3->Text);

y2=StrToInt(Edit4->Text);

int x, y, dx, dy, pk, k, xEnd;

dx=abs(x2-x1);

dy=abs(y2-y1);

if(x1>x2)

{

x = x2;

y = y2;

xEnd = x1;

}

else

{

x = x1;

y = y1;

xEnd = x2;

}

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

pk=2\*dy-dx;

while (x<=xEnd)

{

if(pk<0)

{

x=x+1;

y=y;

pk=pk+2\*dy;

}

else

{

x=x+1;

y=y+1;

pk= pk+2\*dy-2\*dx;

}

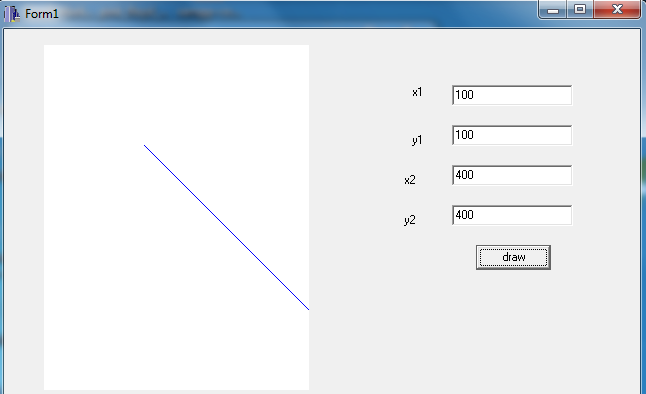
Image1->Canvas->Pixels[x][y]=RGB(0,0,255);

}

}

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

**Output screen:**



**Conclusion:**

The Bresenham Line Algorithm was implemented in C++ Builder to draw lines as per the pixel coordinates input by the user.