ST. XAVIER’S COLLEGE

**Maitighar, Kathmandu**

**(Affiliated to Tribhuvan University)**



**Computer Graphics**

**Lab Assignment #8**

**REFELCT AN IMAGE ON X-AXIS AND Y-AXIS IN C++ BUILDER**

**Submitted By**

Alok Shrestha

013BSCIT005

B.Sc. CSIT Year II/IV Semester

**Submitted To**

Er. Anil Shah

Lecturer

Department of Computer Science

St. Xavier’s College

Maitighar, Kathmandu

**Submitted On**

September 2015

**STATEMENT**

**TO IMPLEMENT TRANSFORMATION (REFLECT)**

**ALGORITHM**

**For Reflection**

1. Get the width and height of the source image
2. Get parameter for reflection axis (1 for x-axis, 2 for y-axis)
3. For each point i in width

For each point j in height

If parameter==1

The translated point (x’, y’) is given by

x' = i

y’ = - j

If parameter==2

The translated point (x’, y’) is given by

x' = - i

y’ = j

Plot the points (x’, y’) with the same color as source in destination

1. Stop

(General concept for display screens: origin is in top left of the screen and y-axis is positive downwards and hence there is no negative coordinates)

**SOURCE CODE**

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

#include <vcl\vcl.h>

#pragma hdrstop

#include "Unit1.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int x, y;

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

/\*By aalooksth@inbox.com\*/

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

: TForm(Owner)

{

}

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

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

{

x = Image1->Width;

y = Image1->Height;

for(int i=0; i<=x; i++)

{

for(int j=0; j<=y; j++)

{

int a=i+x; //Imagining own axis with O(x,y) & axes as we do in paper

int b=j;

Image2->Canvas->Pixels[a][b]=Image1->Canvas->Pixels[i][j];

}

}

Image2->Canvas->MoveTo(x,0);

Image2->Canvas->LineTo(x,2\*y);

Image2->Canvas->MoveTo(0,y);

Image2->Canvas->LineTo(2\*x,y);

}

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

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

{

int parm=StrToInt(Edit1->Text);

int a, b;

for(int i=0; i<=x; i++)

{

for(int j=0; j<=y; j++)

{

switch(parm)

{

case 1:

a=i+x;

b=-j+2\*y; //whenever y coordinate is to be operated on 2\*y needs to be added

break;

case 2:

a=-i+x;

b=j;

break;

}

Image2->Canvas->Pixels[a][b]=Image1->Canvas->Pixels[i][j];

}

}

//for demarcating axes

Image2->Canvas->MoveTo(x,0);

Image2->Canvas->LineTo(x,2\*y);

Image2->Canvas->MoveTo(0,y);

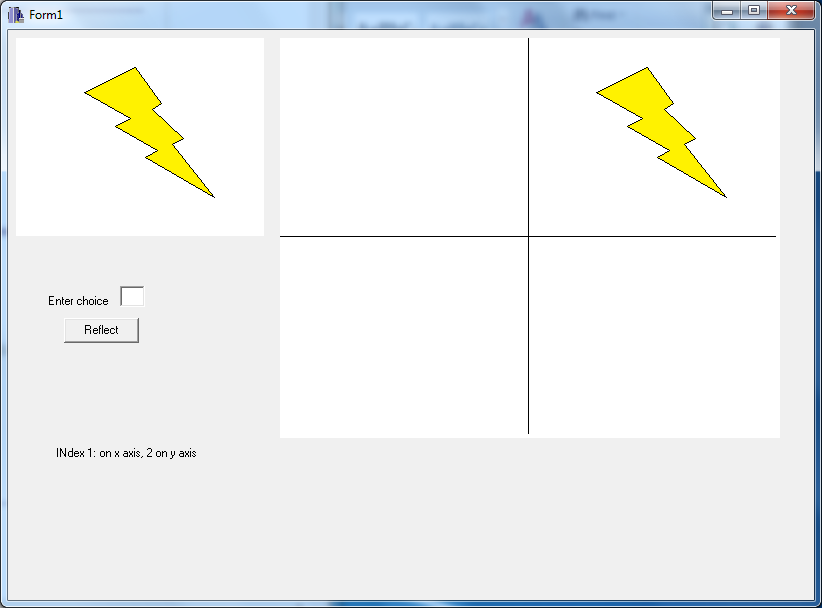
Image2->Canvas->LineTo(2\*x,y);

}

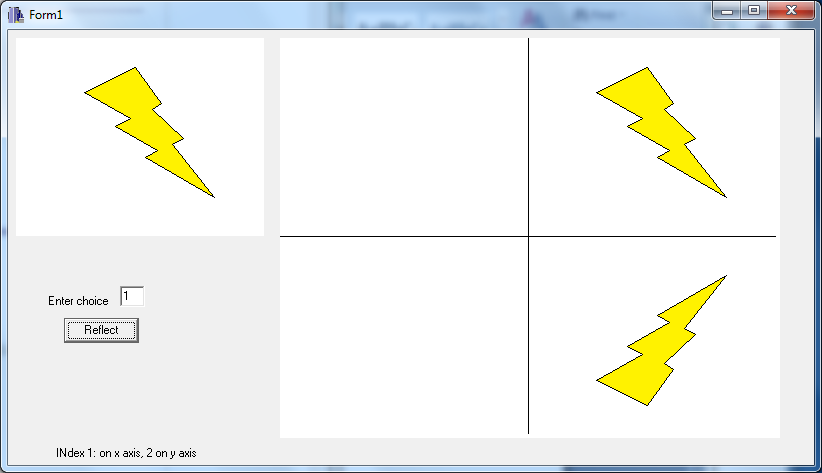
//---------------------------------------------------------------------------

**OUTPUT/s**

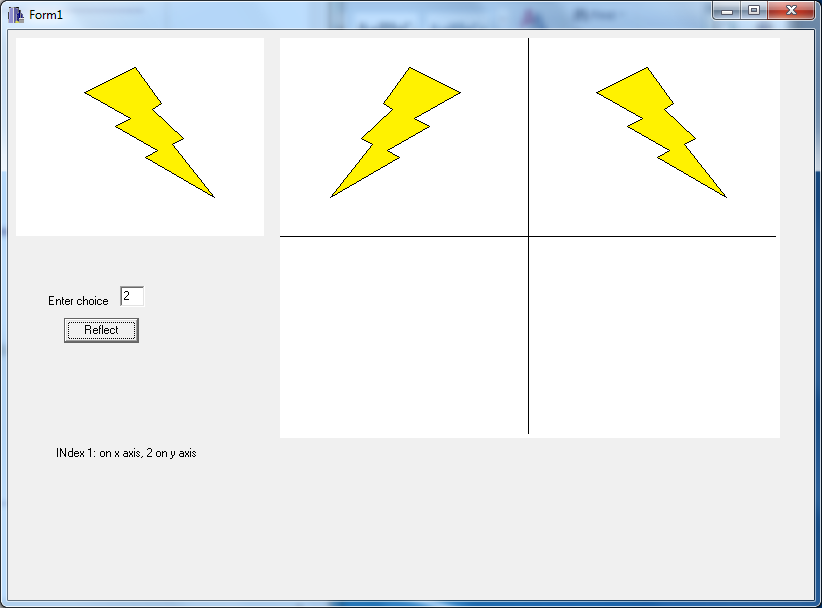
**Basic Interface**

****

**First Run: Reflection on x-axis (parm=1)**

****

**Second Run: Scaling by Sx=2, Sy=3**



**CONCLUSION**

Hence, the reflection of the image was performed in x-axis and y-axis were performed in C++builder.