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**

Arjun Neupane

013BSCIT009

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 "Unit8.h"

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

#pragma resource "\*.dfm"

TForm1 \*Form1;

int a,b,i,j,k,x,y;

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

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

: TForm(Owner)

{

}

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

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

{

x = Image1->Height;

y = Image1->Width;

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

{

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

{

a = -i;

b = -j;

a=a+x;

b=b+y;

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

}

}

}

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

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

{

x = Image1->Height;

y = Image1->Width;

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

{

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

{

a = i;

b = -j;

b=b+y;

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

}

}

}

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

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

{

x = Image1->Height;

y = Image1->Width;

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

{

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

{

a = -i;

b = j;

a=a+x;

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

}

}

}

**Output:**

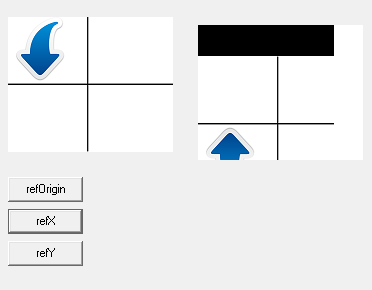


Fig: reflection about X-axis

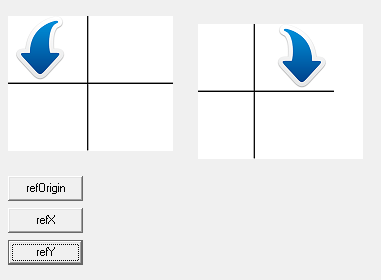


Fig: reflection about y-axis

**Conclusion:**

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