'========================================================================================

'Project: Lab 1

'Title: Game1 Project

'File Name: Game1.vb

'Date completed: February 25th, 2013.

'Authors: Ryu Muthui

'Class: CS161 Winter 2013

'Description: This is a game project created to demonstrate various introductory

' skills in game development. The user can click on each of control

' buttons to display the specified graphical actions.

' \*Random Dots asks the user how many random colored dots(pixels) will

' to be placed on the panel.

' \*Picture displays an image and mirrors it just to the right.

' \*Geometric Shapes displays 3 sets of shapes on screen.

' \*Connect displays 3 sets of gradient line connecting the 3 shapes.

' \*Gradient fills the graphics panel with 2D color gradient settings.

' The user can select Reset to clear the screen and Exit ends the

' application.

'========================================================================================

Option Explicit On

Option Strict On

Imports System.Threading

Imports System.Drawing.Point

Imports System.Drawing.Drawing2D

Public Class frmGame1

Dim clrTemp As Color

Dim graCanvas As Graphics

Dim bmpCanvas As Bitmap

Dim graPicture As Graphics

Dim bmpSource As Bitmap

Dim bmpCopy As Bitmap

Dim cshtWidth As Short

Dim cshtHeight As Short

Dim strImageLocation As String = "..\Images\BlankImage.png"

Private Sub frmGame1\_Load(sender As Object, e As System.EventArgs) Handles Me.Load

'--------------------------------------------------------------------------------

'Description: Initailizes various graphic and bitmap sources and sets info texts.

' Called by: btnReset\_Click

'--------------------------------------------------------------------------------

graPicture = pnlGraphics.CreateGraphics

graCanvas = pnlGraphics.CreateGraphics

bmpSource = New Bitmap(strImageLocation)

cshtWidth = CShort(bmpSource.Width)

cshtHeight = CShort(bmpSource.Height)

bmpCopy = New Bitmap(cshtWidth, cshtHeight, graPicture)

bmpCanvas = New Bitmap(pnlGraphics.Width, pnlGraphics.Height, graPicture)

lblInformation.Text = "Welcome to Game1! Click the control buttons to display" \_

& " various graphics in the center panel!"

End Sub

Private Sub sUpdateScreen()

'--------------------------------------------------------------------------------

'Description: A subrutine to update and redraw on the panel when called.

' Called By: btnRandomDots\_Click, btnGeometricShapes\_Click, btnPictures\_Click

' btnGradient\_Click, pnlPictures\_Paint, btnConnect\_Click

'--------------------------------------------------------------------------------

graPicture.DrawImageUnscaled(bmpSource, 50, 50)

graPicture.DrawImageUnscaled(bmpCopy, cshtWidth + 51, 50)

graCanvas.DrawImageUnscaled(bmpCanvas, 0, 0)

End Sub

Private Sub btnRandomDots\_Click(sender As System.Object, e As System.EventArgs) \_

Handles btnRandomDots.Click

'--------------------------------------------------------------------------------

'Description: Asks the user how many random colored dots(pixels) to place inside

' the panel. Enclosed in Try/Catch block so that only whole positive

' numbers can be accepted.

' Calls: sUpdateScreen

'--------------------------------------------------------------------------------

Dim lngNumOfDots As Long

Dim shtWidth As Short = CShort(pnlGraphics.Width)

Dim shtHeight As Short = CShort(pnlGraphics.Height)

Dim intX As Integer

Dim intY As Integer

Dim i As Long

Dim bytRed As Byte

Dim bytGreen As Byte

Dim bytBlue As Byte

Dim lngStart As Long

Dim lngStop As Long

Randomize()

Try

lngNumOfDots = CLng(InputBox("How many dots would you like to draw on " \_

& "screen?", , CStr(CLng(10000))))

If lngNumOfDots <= 0 Then

MessageBox.Show(("Invalid Input: Enter positive whole number values."), \_

"Invalid input", MessageBoxButtons.OK)

Else

lngStart = Environment.TickCount

For i = 0 To lngNumOfDots

intX = CShort(Int(Rnd() \* shtWidth))

intY = CShort(Int(Rnd() \* shtHeight))

bytRed = CByte(Int(Rnd() \* 256))

bytGreen = CByte(Int(Rnd() \* 256))

bytBlue = CByte(Int(Rnd() \* 256))

clrTemp = Color.FromArgb(bytRed, bytBlue, bytGreen)

bmpCanvas.SetPixel(intX, intY, clrTemp)

Application.DoEvents()

Next i

graCanvas.DrawImageUnscaled(bmpCanvas, 0, 0)

lngStop = Environment.TickCount

lblInformation.Text = ((lngStop - lngStart) / 1000) \_

.ToString & " Seconds to draw on panel." & Chr(13) & CInt(lngNumOfDots \_

/ (lngStop - lngStart)).ToString & " Dots per second." & Chr(13) & \_

"CPU: Intel(R) Core 2CPU 6600 @ 2.4GHz" & Chr(13) & "Graphics Card" \_

& ": NVIDIA GeForce GTX550Ti"

End If

Catch InvalidCastException As InvalidCastException

MessageBox.Show(("Invalid Input: Enter positive whole number values."), \_

"Invalid input", MessageBoxButtons.OK)

End Try

Call sUpdateScreen()

End Sub

Private Sub btnGeometricShapes\_Click(sender As System.Object, e As System.EventArgs \_

) Handles btnGeometricShapes.Click

'--------------------------------------------------------------------------------

'Description: Draws 3 sets of shapes on screen using GDI+ draw commands.

' Calls: sUpdateScreen, btnReset\_Click

'--------------------------------------------------------------------------------

Call btnReset\_Click(sender, e)

btnConnect.Enabled = True

lblInformation.Text = "Basketball, Baseball, and a Triangle drawn using GDI+ " \_

& "draw commands."

'basketball

Dim pntBasketBall1(2) As Point

pntBasketBall1(0).X = 75

pntBasketBall1(0).Y = 170

pntBasketBall1(1).X = 100

pntBasketBall1(1).Y = 220

pntBasketBall1(2).X = 75

pntBasketBall1(2).Y = 280

Dim pntBasketBall2(2) As Point

pntBasketBall2(0).X = 170

pntBasketBall2(0).Y = 170

pntBasketBall2(1).X = 150

pntBasketBall2(1).Y = 220

pntBasketBall2(2).X = 170

pntBasketBall2(2).Y = 280

graCanvas.FillEllipse(Brushes.OrangeRed, 50, 150, 150, 150)

graCanvas.DrawEllipse(Pens.Black, 50, 150, 150, 150)

graCanvas.DrawLine(Pens.Black, 50, 225, 200, 225)

Thread.Sleep(1000)

graCanvas.DrawLine(Pens.Black, 125, 150, 125, 300)

Thread.Sleep(1000)

graCanvas.DrawCurve(Pens.Black, pntBasketBall1)

Thread.Sleep(1000)

graCanvas.DrawCurve(Pens.Black, pntBasketBall2)

Thread.Sleep(1000)

'Triangle

Dim ptnTriangle(2) As Point

ptnTriangle(0).X = 300

ptnTriangle(0).Y = 150

ptnTriangle(1).X = 410

ptnTriangle(1).Y = 240

ptnTriangle(2).X = 310

ptnTriangle(2).Y = 260

graCanvas.DrawLine(Pens.YellowGreen, 300, 50, 300, 160)

Thread.Sleep(1000)

graCanvas.DrawLine(Pens.YellowGreen, 300, 160, 400, 120)

Thread.Sleep(1000)

graCanvas.DrawLine(Pens.YellowGreen, 400, 120, 300, 50)

Thread.Sleep(1000)

'BaseBall

Dim pntBaseBall1(2) As Point

pntBaseBall1(0).X = 545

pntBaseBall1(0).Y = 175

pntBaseBall1(1).X = 565

pntBaseBall1(1).Y = 205

pntBaseBall1(2).X = 550

pntBaseBall1(2).Y = 250

Dim pntBaseBall2(2) As Point

pntBaseBall2(0).X = 615

pntBaseBall2(0).Y = 175

pntBaseBall2(1).X = 595

pntBaseBall2(1).Y = 205

pntBaseBall2(2).X = 610

pntBaseBall2(2).Y = 250

graCanvas.FillEllipse(Brushes.White, 530, 160, 100, 100)

graCanvas.DrawEllipse(Pens.Black, 530, 160, 100, 100)

Thread.Sleep(1000)

graCanvas.DrawCurve(Pens.Red, pntBaseBall1)

Thread.Sleep(1000)

graCanvas.DrawCurve(Pens.Red, pntBaseBall2)

Thread.Sleep(1000)

Call sUpdateScreen()

End Sub

Private Sub btnPictures\_Click(sender As System.Object, e As System.EventArgs) \_

Handles btnPictures.Click

'--------------------------------------------------------------------------------

'Description: Draws an image and mirrors that image next to it using

' transitonal effects.

' Calls: sUpdateScreen(), btnReset\_Click

'--------------------------------------------------------------------------------

Call btnReset\_Click(sender, e)

Dim clrTemp As Color

Dim i As Integer

Dim j As Integer

Dim shtX As Short

Dim shtY As Short

strImageLocation = "..\Images\SunSetBeach.jpg"

graPicture = pnlGraphics.CreateGraphics

bmpSource = New Bitmap(strImageLocation)

cshtWidth = CShort(bmpSource.Width)

cshtHeight = CShort(bmpSource.Height)

bmpCopy = New Bitmap(cshtWidth, cshtHeight, graPicture)

lblInformation.Text = "Picture Image that is mirrored in with transitonal" \_

& " effects."

For i = 0 To cshtWidth - 1

For j = 0 To cshtHeight - 1

clrTemp = bmpSource.GetPixel(i, j)

Next j

Call sUpdateScreen()

Next i

'mirror effect

For i = 0 To cshtWidth \* 5

For j = 0 To cshtHeight - 1

shtX = CShort(Int(Rnd() \* cshtWidth))

shtY = CShort(Int(Rnd() \* cshtHeight))

clrTemp = bmpSource.GetPixel(shtX, shtY)

bmpCopy.SetPixel(cshtWidth - 1 - shtX, shtY, clrTemp)

Application.DoEvents()

Next j

Call sUpdateScreen()

Next i

'transitioining effect

For i = 0 To cshtWidth - 1

For j = 0 To cshtHeight - 1

clrTemp = bmpSource.GetPixel(i, j)

bmpCopy.SetPixel(cshtWidth - 1 - i, j, clrTemp)

Next j

Call sUpdateScreen()

Next i

pnlGraphics.BackgroundImage = My.Resources.BackGround

End Sub

Private Sub btnGradient\_Click(sender As System.Object, e As System.EventArgs) \_

Handles btnGradient.Click

'--------------------------------------------------------------------------------

'Description: Fills the graphics panel with 2D color gradient settings.

' Calls: sUpdateScreen(), btnReset\_Click

'--------------------------------------------------------------------------------

Dim intWidth As Integer = pnlGraphics.Width - 1

Dim intHeight As Integer = pnlGraphics.Height - 1

Dim intX As Integer = 0

Dim intY As Integer = 0

Dim bytRed As Byte = 0

Dim bytGreen As Byte = 0

Dim clrPoint As Color

Call btnReset\_Click(sender, e)

lblInformation.Text = "Panel Screen filled with a 2D color gradient." & Chr(13) \_

& "Pure Red @ Top Left to Pure Black @ Bottom Left." & Chr(13) & "Pure " \_

& "Black to Pure Green from left to right of Panel."

graCanvas = pnlGraphics.CreateGraphics

bmpCanvas = New Bitmap(intWidth + 1, intHeight + 1, graCanvas)

For intX = 0 To intWidth

For intY = 0 To intHeight

bytRed = CByte(255 - Int(intY / intHeight \* 255))

bytGreen = CByte(Int(intX / intWidth \* 255))

clrPoint = Color.FromArgb(bytRed, bytGreen, 0)

bmpCanvas.SetPixel(intX, intY, clrPoint)

Application.DoEvents()

Next intY

graCanvas.DrawImageUnscaled(bmpCanvas, 0, 0)

Next intX

Call sUpdateScreen()

End Sub

Private Sub btnConnect\_Click(sender As System.Object, e As System.EventArgs) \_

Handles btnConnect.Click

'--------------------------------------------------------------------------------

'Description: Draws gradient lines matching with the color of the Geometric

' Shapes, connecting the 3 shapes.

' Calls: sUpdateScreen()

'--------------------------------------------------------------------------------

Dim i As Integer

lblInformation.Text = "Gradient Line starting from the Basketball to the " \_

& "Trianle. From the Triangle to the Baseball and back to the Basketball." \_

& Chr(13) & "Gradient line matches with the shape colors."

For i = 0 To 5

Dim lineGradientBrush As New LinearGradientBrush(New Point(125, 225), \_

New Point(330, 110), Color.FromArgb(255, 125, 0), Color.FromArgb \_

(0, 255, 64))

Dim pen As New Pen(lineGradientBrush)

graCanvas.DrawLine(pen, 125 + i, 225 + i, 330 + i, 110 + i)

Next i

Thread.Sleep(1000)

For i = 0 To 5

Dim lineGradientBrush As New LinearGradientBrush(New Point(330, 110), \_

New Point(580, 210), Color.FromArgb(0, 255, 64), Color.FromArgb \_

(255, 255, 255))

Dim pen As New Pen(lineGradientBrush)

graCanvas.DrawLine(pen, 330 + i, 110 + i, 580 + i, 210 + i)

Next i

Thread.Sleep(1000)

For i = 0 To 5

Dim lineGradientBrush As New LinearGradientBrush(New Point(580, 210), \_

New Point(125, 225), Color.FromArgb(255, 255, 255), Color.FromArgb \_

(255, 125, 0))

Dim pen As New Pen(lineGradientBrush)

graCanvas.DrawLine(pen, 580 + i, 210 + i, 125 + i, 225 + i)

Next i

Call sUpdateScreen()

End Sub

Private Sub btnReset\_Click(sender As System.Object, e As System.EventArgs) \_

Handles btnReset.Click

'--------------------------------------------------------------------------------

'Description: Resets the screen to it's default state.

' Calls: frmGame1\_Load

' Called By: btnGeometricShapes\_Click, btnGradient\_Click,

' btnPictures\_Click,

'--------------------------------------------------------------------------------

bmpSource.Dispose()

bmpCanvas.Dispose()

bmpSource.Dispose()

graCanvas.Clear(pnlGraphics.BackColor)

Call frmGame1\_Load(sender, e)

strImageLocation = "..\Images\BlankImage.png"

bmpSource = New Bitmap(strImageLocation)

btnConnect.Enabled = False

pnlGraphics.BackColor = Color.Transparent

End Sub

Private Sub pnlPictures\_Paint(ByVal sender As Object, \_

ByVal e As System.Windows.Forms.PaintEventArgs) Handles pnlGraphics.Paint

'--------------------------------------------------------------------------------

'Description: This helps keep the images drawn on screen stay on the screen.

' Calls: sUpdateScreen

'--------------------------------------------------------------------------------

Call sUpdateScreen()

End Sub

Private Sub btnExit\_Click(sender As System.Object, e As System.EventArgs) \_

Handles btnExit.Click

'--------------------------------------------------------------------------------

'Description: Presents the user with a MessageBox and exit the application.

'--------------------------------------------------------------------------------

MessageBox.Show(("Thanks for playing Game 1."), "Good Bye!")

Me.Close()

End Sub

End Class