

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Imaging;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

using ComponentFactory.Krypton.Toolkit;

namespace PhotoEditor

{

public partial class Form1 : KryptonForm

{

public Form1()

{

InitializeComponent();

}

int Iwidth=0,Iheight=0;

private void Form1\_DragDrop(object sender, DragEventArgs e)

{

}

private void Form1\_Load(object sender, EventArgs e)

{

ParamPro.tool = null;

After.AllowDrop = true;

//artBoard.Height = ParamPro.abHeight;

//artBoard.Width = ParamPro.abWidth;

//Theam.ImageLocation = Image.FromFile("PhotoEditor\\Resources\\moon\_606807.png");

}

//byte[] img;

Bitmap b2 = null,org = null;

public string path, filename;

private void Filter1\_Click(object sender, EventArgs e)

{

b2.RotateFlip(RotateFlipType.Rotate180FlipY);

After.Image = b2;

}

private void trackBar1\_Scroll(object sender, EventArgs e)

{

}

private void Save\_Click(object sender, EventArgs e)

{

}

private void kryptonButton2\_Click(object sender, EventArgs e)

{

b2.RotateFlip(RotateFlipType.Rotate180FlipX);

After.Image = b2;

}

private void kryptonButton3\_Click(object sender, EventArgs e)

{//Black and White Filter

int i, j;

for (i = 0; i < b2.Width; i++)

{

for (j = 0; j < b2.Height; j++)

{

int a = b2.GetPixel(i, j).A;

int r = b2.GetPixel(i, j).R;

int g = b2.GetPixel(i, j).G;

int b = b2.GetPixel(i, j).B;

int v = (r+g+b)/3;

//int ta, tr, tg, tb;

b2.SetPixel(i, j, Color.FromArgb(a, v, v, v));

}

}

After.Image = b2;

}

Bitmap u1, u2, u3, u4, u5;

public void undo()

{

}

private void kryptonButton4\_Click(object sender, EventArgs e)

{

int i, j;

for (i = 0; i < b2.Width; i++)

{

for (j = 0; j < b2.Height; j++)

{

int a = b2.GetPixel(i, j).A;

int r = b2.GetPixel(i, j).R;

int g = b2.GetPixel(i, j).G;

int b = b2.GetPixel(i, j).B;

int tr, tg, tb;

tr = 255 - r;

tg = 255 - g;

tb = 255 - b;

b2.SetPixel(i, j, Color.FromArgb(a, tr, tg, tb));

}

}

After.Image = b2;

}

private void saveToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFile.Filter = "Image Files(Image Files(\*,jepg;\*.jpg;\*.png;\*.bmp;\*.gif;\*)|\*.jpeg;\*.jpg;\*.png;\*.bmp;\*.gif;)";

string file = SaveFile.FileName;

ImageFormat format = ImageFormat.Png;

if (SaveFile.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

string ext = System.IO.Path.GetExtension(SaveFile.FileName);

switch (ext)

{

case ".jpg":

format = ImageFormat.Jpeg;

break;

case ".bmp":

format = ImageFormat.Bmp;

break;

}

After.Image.Save(SaveFile.FileName, format);

}

}

bool draw;

int drag=0;

Point px, py;

private void After\_MouseDown(object sender, MouseEventArgs e)

{

py = e.Location;

draw = true;

}

private void After\_MouseMove(object sender, MouseEventArgs e)

{

Control c = sender as Control;

Pen p = new Pen(ForeColor.BackColor, 5);

Pen er = new Pen(Color.Transparent, 10);

try

{

if (draw)

{

px = e.Location;

int nx = px.X - py.X;

int ny = px.Y - py.Y;

Graphics g = Graphics.FromImage(b2);

if (ParamPro.tool == "pen")

{

g.DrawLine(p, px, py);

py = px;

After.Refresh();

}

else if (ParamPro.tool == "erasor")

{

g.Clear(Color.Transparent);

After.Image = b2;

}

else if (ParamPro.tool == "move")

{

c.Top = e.Y + c.Top - py.Y;

c.Left = e.X + c.Left - py.X;

}

Refresh();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void After\_MouseUp(object sender, MouseEventArgs e)

{

px = e.Location;

draw = false;

if (rect != null && ParamPro.tool == "crop")

{

Bitmap croped = new Bitmap(rect.Width, rect.Height);

Graphics cr = Graphics.FromImage(croped);

cr.DrawImage(b2,0,0,rect,GraphicsUnit.Pixel);

After.Image = croped;

}

}

private void After\_Paint(object sender, PaintEventArgs e)

{

if (ParamPro.tool == "crop")

{

Graphics g = Graphics.FromImage(b2);

e.Graphics.DrawRectangle(Pens.Red,getRect());

}

}

private Rectangle getRect()

{

rect = new Rectangle();

rect.X = Math.Min(py.X,px.X);

rect.Y = Math.Min(py.Y, px.Y);

rect.Width = Math.Abs(py.X-px.X);

rect.Height = Math.Abs(py.X-px.X);

return rect;

}

private void Red\_Leave(object sender, EventArgs e)

{

}

private void kryptonTrackBar1\_ValueChanged\_1(object sender, EventArgs e)

{

Red.Text = kryptonTrackBar1.Value.ToString();

int i, j;

for (i = 0; i < b2.Width; i++)

{

for (j = 0; j < b2.Height; j++)

{

int a = b2.GetPixel(i, j).A;

int r = b2.GetPixel(i, j).R;

int g = b2.GetPixel(i, j).G;

int b = b2.GetPixel(i, j).B;

int val = kryptonTrackBar1.Value;

b2.SetPixel(i, j, Color.FromArgb(a, val, g, b));

}

}

After.Image = b2;

}

private void kryptonTrackBar2\_ValueChanged(object sender, EventArgs e)

{

Green.Text = kryptonTrackBar2.Value.ToString();

int i, j;

for (i = 0; i < b2.Width; i++)

{

for (j = 0; j < b2.Height; j++)

{

int a = b2.GetPixel(i, j).A;

int r = b2.GetPixel(i, j).R;

int g = b2.GetPixel(i, j).G;

int b = b2.GetPixel(i, j).B;

int val = kryptonTrackBar2.Value;

b2.SetPixel(i, j, Color.FromArgb(a, r, val, b));

}

}

After.Image = b2;

}

private void Green\_Leave(object sender, EventArgs e)

{

}

private void kryptonTrackBar3\_ValueChanged(object sender, EventArgs e)

{

Blue.Text = kryptonTrackBar3.Value.ToString();

int i, j;

for (i = 0; i < b2.Width; i++)

{

for (j = 0; j < b2.Height; j++)

{

int a = b2.GetPixel(i, j).A;

int r = b2.GetPixel(i, j).R;

int g = b2.GetPixel(i, j).G;

int b = b2.GetPixel(i, j).B;

int val = kryptonTrackBar2.Value;

b2.SetPixel(i, j, Color.FromArgb(a, r, g, val));

}

}

After.Image = b2;

}

private void Blue\_TextChanged(object sender, EventArgs e)

{

kryptonTrackBar3.Value = Convert.ToInt32(Blue.Text);

}

private void Blue\_Leave(object sender, EventArgs e)

{

}

int tmode = 1,textTool = 0,txt=0,cx=0,cy=0;

private void Theam\_Click(object sender, EventArgs e)

{

if (tmode == 1)

{

tmode = 0;

this.BackColor = Color.FromArgb(50, 50, 50);

tableLayoutPanel17.BackColor = Color.FromArgb(0,0,0);

Theam.Image = global::PhotoEditor.Properties.Resources.moon;

}

else

{

this.BackColor = Color.LightGray;

Theam.Image = global::PhotoEditor.Properties.Resources.sun;

tmode = 1;

}

}

public void zoom(Bitmap bm,Size size)

{

if (size.Height != 1 || size.Width != 1)

{

Bitmap zoom = new Bitmap(bm, Convert.ToInt32(Iwidth \* size.Width), Convert.ToInt32(Iheight \* size.Height));

Graphics gpu = Graphics.FromImage(zoom);

b2 = zoom;

}

else

{

Bitmap zoom = new Bitmap(bm, Convert.ToInt32(Iwidth), Convert.ToInt32(Iheight));

Graphics gpu = Graphics.FromImage(zoom);

b2 = zoom;

}

After.Image = b2;

}

public void addText(string str,int x,int y)

{

string Text = str;

PointF firstLocation = new Point(x, y);

Brush bru = new SolidBrush(ForeColor.BackColor);

using (Graphics graphics = Graphics.FromImage(b2))

{

using (Font arialFont = new Font("Arial", 10))

{

graphics.DrawString(Text, arialFont,bru, firstLocation);

}

}

After.Image = b2;

}

private void pictureBox2\_Click(object sender, EventArgs e)

{

if(textTool == 0)

{

textTool =1;

pictureBox2.Image = Image.FromFile(@"E:\22ICT28\Semester 1\Software Programming\PhotoEditor\PhotoEditor\Resources\text\_3721819.png");

}

else if(textTool == 1)

{

textTool = 0;

textBox.Text = null;

textBox.Visible = false;

pictureBox2.Image = Image.FromFile(@"E:\22ICT28\Semester 1\Software Programming\PhotoEditor\PhotoEditor\Resources\text\_3721901.png");

}

}

private void After\_Click(object sender, EventArgs e)

{

if (textTool == 1)

{

textInput inputtxt = new textInput();

inputtxt.Visible = true;

inputtxt.Size = new Size(25,40);

if (txt == 0)

{

cx = Cursor.Position.X - 125;

cy = Cursor.Position.Y - 100;

inputtxt.Location = new Point(cx, cy);

inputtxt.Visible = true;

this.Controls.Add(inputtxt);

txt = 1;

}

else if (txt == 1)

{

addText(inputtxt.Text, cx, cy);

inputtxt.Visible = false;

textBox.Text = null;

txt = 0;

this.Controls.Remove(inputtxt);

inputtxt.Dispose();

}

}

}

private void ForeColor\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

int cdR = colorDialog1.Color.R;

int cdG = colorDialog1.Color.G;

int cdB = colorDialog1.Color.B;

ForeColor.BackColor = Color.FromArgb(cdR,cdG,cdB);

}

}

private void backColor\_Click(object sender, EventArgs e)

{

if (colorDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

int cdR = colorDialog1.Color.R;

int cdG = colorDialog1.Color.G;

int cdB = colorDialog1.Color.B;

backColor.BackColor = Color.FromArgb(cdR, cdG, cdB);

}

}

private void textBox1\_Click(object sender, EventArgs e)

{

if (fontDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

Font font = fontDialog1.Font;

}

}

private void kryptonDockingManager1\_PageCloseRequest(object sender, ComponentFactory.Krypton.Docking.CloseRequestEventArgs e)

{

}

private void trackBar1\_Scroll\_1(object sender, EventArgs e)

{

zoom(b2, new Size(trackBar1.Value,trackBar1.Value));

}

private void resetToolStripMenuItem\_Click(object sender, EventArgs e)

{

b2 = org;

After.Image = org;

}

private void After\_DragEnter(object sender, DragEventArgs e)

{

e.Effect = DragDropEffects.Copy;

}

private void After\_DragDrop(object sender, DragEventArgs e)

{

try

{

var data = e.Data.GetData(DataFormats.FileDrop);

if (data != null)

{

var file = data as string[];

if (file.Length > 0)

{

Image img = System.Drawing.Image.FromFile(file[0]);

Bitmap b1 = new Bitmap(img);

b2 = new Bitmap(b1, b1.Width / 3, b1.Height / 3);

org = b2;

Iwidth = b2.Width;

Iheight = b2.Height;

After.Image = b2;

label1.Text = file[0];

data = null;

}

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void kryptonButton5\_Click(object sender, EventArgs e)

{//reset button

b2 = new Bitmap(org,org.Width,org.Height);

After.Image = b2;

}

public void ClearImage()

{

Graphics g = Graphics.FromImage(b2);

g.Clear(Color.Wheat);

}

private void pictureBox5\_Click(object sender, EventArgs e)

{

ParamPro.toolS(7);

}

int Erasor = 0;

private void pictureBox10\_Click(object sender, EventArgs e)

{

ParamPro.toolS(8);

if (ParamPro.tool == "erasor")

{

}

}

int tx = 300, ty = 350,i = 1;

//PictureBox[] layers = new PictureBox[20];

//LayersPanel[] lpanel = new LayersPanel[20];

public void pictureBox13\_Click(object sender, EventArgs e)

{//For Create a new Layer

var name = "Layer " + (i).ToString();

addLayer(name,"layer");

}

public void addLayer(string n,string type)

{

PictureBox layer = new PictureBox();

layer.Location = new Point(artBoard.Location.X - 18, artBoard.Location.Y - 18);

layer.Size = new Size(artBoard.Width, artBoard.Height);

layer.BackColor = backColor.BackColor;

//layer.BackColor = Color.Transparent;

layer.Dock = DockStyle.Fill;

layer.Anchor = AnchorStyles.None;

layer.Click += new System.EventHandler(ClickHandler);

layer.MouseDown += new MouseEventHandler(Layer\_DownEvent);

layer.MouseUp += new MouseEventHandler(Layer\_UpEvent);

layer.MouseMove += new MouseEventHandler(LayerM);

layer.Paint += new PaintEventHandler(LPaint);

layer.BorderStyle = BorderStyle.FixedSingle;

if (type == "img")

{

layer.Image = b2;

layer.BackColor = Color.Transparent;

}

layer.Name = n;

tx += 100; ty += 100;

artBoard.Controls.Add(layer);

layer.BringToFront();

ParamPro.LName = n;

LayersPanel lpanel = new LayersPanel();

lpanel.Name = n;

lpanel.Click += new EventHandler(ClickLayar);

lpanel.pictureBox1.Name = n;

lpanel.pictureBox1.Click += new EventHandler(HideVis);

flowLayoutPanel1.Controls.Add(lpanel);

i += 1;

}

PictureBox curlay = null; LayersPanel curpan,prepan;

private void HideVis(object sender , EventArgs e)

{

var pan = sender as Control;

var lay = ((PictureBox)artBoard.Controls.Find(pan.Name, true)[0]);

if (lay.Visible == true)

{

lay.Visible = false;

}

else

{

lay.Visible = true;

}

}

private void ClickLayar(object sender,System.EventArgs e)

{

curpan = sender as LayersPanel;

var name = curpan.Name;

if (curlay != null)

{

prepan.BackColor = Color.White;

prepan.ForeColor = Color.Black;

}

curlay = ((PictureBox)artBoard.Controls.Find(name, true)[0]);

nb = new Bitmap(curlay.Width, curlay.Height);

nb.MakeTransparent();

lg = Graphics.FromImage(nb);

curpan.BackColor = Color.Blue;

curpan.ForeColor = Color.White;

prepan = curpan;

}

private void LPaint(object sender, PaintEventArgs e)

{

if (ParamPro.tool == "crop")

{

//Graphics g = curlay.CreateGraphics();

e.Graphics.DrawRectangle(Pens.Red, getRect());

}

}

private void LayerM(object sender, MouseEventArgs e)

{

PictureBox c = curlay;

Pen p = new Pen(ForeColor.BackColor, 5);

Pen er = new Pen(Color.Transparent, 10);

px = e.Location;

try

{

if (draw && c != null)

{

int nx = px.X - py.X;

int ny = px.Y - py.Y;

if (ParamPro.tool == "pen")

{

lg.DrawLine(p, px, py);

py = px;

c.Image = nb;

}

else if (ParamPro.tool == "erasor")

{

lg.Clear(Color.Transparent);

c.Refresh();

}

else if (ParamPro.tool == "move")

{

c.Top = e.Y + c.Top - py.Y;

c.Left = e.X + c.Left - py.X;

}

Refresh();

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

Bitmap nb; Graphics lg;

private void Layer\_DownEvent(object sender, MouseEventArgs e)

{

draw = true;

Active = sender as Control;

Cursor = Cursors.Hand;

py = e.Location;

}

private void Layer\_UpEvent(object sender, MouseEventArgs e)

{

draw = false;

Active = null;

Cursor = Cursors.Default;

}

Control Active;

private void ClickHandler(object sender,System.EventArgs e)

{

//MessageBox.Show("Its Work");

}

private void pictureBox14\_Click(object sender, EventArgs e)

{//For Hide Layers

artBoard.Controls.Remove(curlay);

curlay.Dispose();

LayersPanel.Controls.Remove(curpan);

curpan.Dispose();

}

private void LayersList\_SelectedIndexChanged(object sender, EventArgs e)

{

}

private void propartiesToolStripMenuItem\_Click(object sender, EventArgs e)

{

pd = new ProDialog();

pd.Show();

}

ProDialog pd;

private void panel7\_DragEnter(object sender, DragEventArgs e)

{

e.Effect = DragDropEffects.Copy;

}

private void panel7\_DragDrop(object sender, DragEventArgs e)

{

ProDialog pr = new ProDialog();

pr.Dock = DockStyle.Fill;

pr.TopLevel = false;

panel7.Controls.Add(pr);

pr.Show();

}

private void openToolStripMenuItem1\_Click(object sender, EventArgs e)

{

try

{

OpenFileDialog prompt = new OpenFileDialog

{

InitialDirectory = @"D:\",

Title = "Browse Text Files",

CheckFileExists = true,

CheckPathExists = true,

DefaultExt = "te",

Filter = "Image Files(Image Files(\*,jepg;\*.jpg;\*.png;\*.bmp;\*.gif;\*)|\*.jpeg;\*.jpg;\*.png;\*.bmp;\*.gif;)",

FilterIndex = 2,

RestoreDirectory = true,

ReadOnlyChecked = false,

ShowReadOnly = false

};

if (prompt.ShowDialog() == DialogResult.OK)

{

path = prompt.FileName;

filename = Path.GetFileName(path);

label1.Text = filename;

string imglog = path;

ParamPro.path = imglog;

ParamPro.fname = filename;

Image img = Image.FromFile(imglog);

ImageFormat format = img.RawFormat;

Bitmap b1 = new Bitmap(imglog);

b2 = new Bitmap(b1, b1.Width / 3, b1.Height / 3);

org = new Bitmap(b1, b1.Width / 3, b1.Height / 3);

Iwidth = b2.Width;

Iheight = b2.Height;

After.Image = b2;

closeToolStripMenuItem.Enabled = true;

}

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

private void closeToolStripMenuItem\_Click(object sender, EventArgs e)

{

string message = "Do you want to close this window?";

string title = "Close Window";

MessageBoxButtons buttons = MessageBoxButtons.YesNo;

DialogResult result = MessageBox.Show(message, title, buttons);

if (result == DialogResult.Yes)

{

After.Image = null;

b2 = null;

closeToolStripMenuItem.Enabled = false;

}

}

private void saveAsToolStripMenuItem\_Click(object sender, EventArgs e)

{

SaveFile.Filter = "Bitmap Image (.bmp)|\*.bmp|JPEG Image (.jpeg)|\*.jpeg|Png Image (.png)|\*.png";

string file = SaveFile.FileName;

ImageFormat format = ImageFormat.Png;

if (SaveFile.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

string ext = System.IO.Path.GetExtension(SaveFile.FileName);

switch (ext)

{

case ".jpg":

format = ImageFormat.Jpeg;

break;

case ".bmp":

format = ImageFormat.Bmp;

break;

}

After.Image.Save(SaveFile.FileName, format);

}

}

private void saveToolStripMenuItem1\_Click(object sender, EventArgs e)

{

int width = artBoard.Size.Width;

int height = artBoard.Size.Height;

using (Bitmap bmp = new Bitmap(width, height))

{

artBoard.DrawToBitmap(bmp, new Rectangle(0, 0, width, height));

bmp.Save(@"C:\Users\Thamilselven\Desktop\testBitmap.jpeg", ImageFormat.Jpeg);

saveAsToolStripMenuItem.Enabled = true;

}

}

private void pictureBox9\_Click(object sender, EventArgs e)

{

ParamPro.toolS(5);

}

private void After\_MouseHover(object sender, EventArgs e)

{

}

private void artBoard\_MouseHover(object sender, EventArgs e)

{

if (ParamPro.tool == "pen")

{

this.Cursor = Cursors.PanNW;

}

else if (Erasor == 1)

{

this.Cursor = Cursors.PanNW;

}

else if (drag == 1)

{

this.Cursor = Cursors.Cross;

}

}

private void artBoard\_MouseLeave(object sender, EventArgs e)

{

this.Cursor = Cursors.Arrow;

}

private void Bright\_TextChanged(object sender, EventArgs e)

{

if (Bright.Text != string.Empty)

{

kryptonTrackBar4.Value = Convert.ToInt32(Bright.Text);

}

}

private void kryptonTrackBar4\_ValueChanged(object sender, EventArgs e)

{

Blue.Text = kryptonTrackBar3.Value.ToString();

int i, j;

for (i = 0; i < b2.Width; i++)

{

for (j = 0; j < b2.Height; j++)

{

int a = b2.GetPixel(i, j).A;

int r = b2.GetPixel(i, j).R;

int g = b2.GetPixel(i, j).G;

int b = b2.GetPixel(i, j).B;

//int val = 255-kryptonTrackBar2.Value;

int nr = (r + trackBar1.Value > 255) ? 255 : (r + trackBar1.Value < 0) ? 0 : r + trackBar1.Value;

int ng = (g + trackBar1.Value > 255) ? 255 : (g + trackBar1.Value < 0) ? 0 : g + trackBar1.Value;

int nb = (b + trackBar1.Value > 255) ? 255 : (b + trackBar1.Value < 0) ? 0 : b + trackBar1.Value;

b2.SetPixel(i, j, Color.FromArgb(a, (byte)r, (byte)g,(byte)b));

}

}

After.Image = b2;

}

private void Green\_TextChanged(object sender, EventArgs e)

{

kryptonTrackBar2.Value = Convert.ToInt32(Green.Text);

}

private void Red\_TextChanged(object sender, EventArgs e)

{

kryptonTrackBar1.Value = (Convert.ToInt32(Red.Text) > 255) ? 255 : Convert.ToInt32(Red.Text);

}

Rectangle rect;

private void pictureBox12\_Click(object sender, EventArgs e)

{

ParamPro.toolS(9);

}

int vis = 1;

private void pictureBox16\_Click(object sender, EventArgs e)

{//For the show hide layer icon change.

if (vis ==1)

{

pictureBox16.Image = global::PhotoEditor.Properties.Resources.hide\_2767146;

vis = 0;

}

else

{

pictureBox16.Image = global::PhotoEditor.Properties.Resources.visible\_5340135;

vis = 1;

}

}

private void artBoard\_MouseMove(object sender, MouseEventArgs e)

{

int cx = e.Location.X;

int cy = e.Location.Y;

label11.Text = "X: " + cx.ToString()+" Y: "+cy.ToString();

}

private void pictureBox11\_Click(object sender, EventArgs e)

{

ParamPro.toolS(10);

}

private void kryptonTrackBar4\_Scroll(object sender, EventArgs e)

{

Bright.Text = kryptonTrackBar4.Value.ToString();

After.Image = AdjustBrightness(b2, kryptonTrackBar4.Value);

}

public static Bitmap AdjustBrightness(Bitmap Image, int Value)

{

Bitmap TempBitmap = Image;

Bitmap NewBitmap = new Bitmap(TempBitmap.Width, TempBitmap.Height);

Graphics NewGraphics = Graphics.FromImage(NewBitmap);

float FinalValue = (float)Value / 255.0f;

float[][] FloatColorMatrix ={

new float[] {1, 0, 0, 0, 0},

new float[] {0, 1, 0, 0, 0},

new float[] {0, 0, 1, 0, 0},

new float[] {0, 0, 0, 1, 0},

new float[] {FinalValue, FinalValue, FinalValue, 1, 1}

};

ColorMatrix NewColorMatrix = new ColorMatrix(FloatColorMatrix);

ImageAttributes Attributes = new ImageAttributes();

Attributes.SetColorMatrix(NewColorMatrix);

NewGraphics.DrawImage(TempBitmap, new Rectangle(0, 0, TempBitmap.Width, TempBitmap.Height), 0, 0, TempBitmap.Width, TempBitmap.Height, GraphicsUnit.Pixel, Attributes);

Attributes.Dispose();

NewGraphics.Dispose();

return NewBitmap;

}

private void duplicateToolStripMenuItem\_Click(object sender, EventArgs e)

{

var name = "Layer "+i.ToString();

addLayer(name, "img");

}

private void sendBackToolStripMenuItem\_Click(object sender, EventArgs e)

{

curlay.SendToBack();

}

private void Form1\_KeyDown(object sender, KeyEventArgs e)

{

if (e.Control && e.KeyCode == Keys.L)

{

var name = "Layer " + (i).ToString();

addLayer(name, "layer");

}

if (e.Control && e.KeyCode == Keys.N)

{

MessageBox.Show("Done");

}

}

private void bringFrontToolStripMenuItem\_Click(object sender, EventArgs e)

{

curlay.BringToFront();

}

}

}