

**Name:**

**Session:**

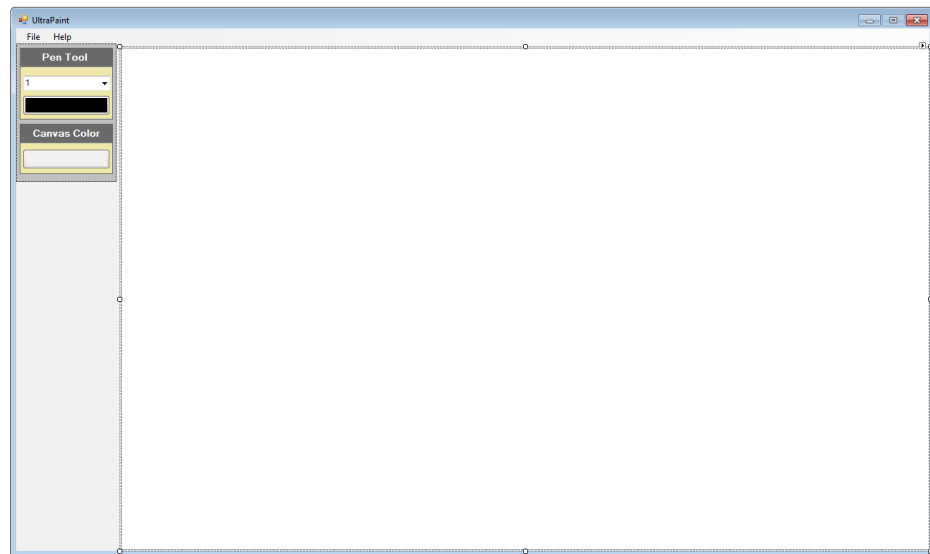
## **Programming II**

**Lab Exercise 4.7.2024**

### **Making a Drawing Program**

In this lab exercise you will build a simple drawing program. I will give you the basic code for the various functionalities of the program then I encourage you to improve the program. For this program, I used a form that had dimensions of 1290 x 766.

#### **The Interface**



On this form you will need the following Controls.

btnPenColor – allows selection of Pen Color as well as shows the color

btnCanvasColor – allows selection of Canvas Color as well as shows the color

pnlPenTool – purely cosmetic

pnlCanvasColor – purely cosmetic

cboPenSize – Allows selection of Pen width

pbDraw – used to hold graphics object

#### **Menu Structure**

File

About

New

About UltraPaint

Save

Open

Exit

After you have completed your interface, it is time to write the code.

1. Start by declaring some variables global to the form. Note the nullable integer variables (see handout on nullable variables):

```
//Declare global variables
bool startPaint = false;
Graphics g;
Bitmap bmp;

//nullable int variables capable storing null value
int? initX = null;
int? initY = null;
```

2. Add the following code to the constructor:

```
//Create graphics object at same location as pbDraw
g = pbDraw.CreateGraphics();

//Create Bitmap object same size as as pbDraw
bmp = new Bitmap(pbDraw.Width, pbDraw.Height);

//Associate bitmap object with picturebox
pbDraw.Image = bmp;
```

3. Add the following code to the btnPenColor\_Click event handler

```
//Open Color Dialog and Set BackColor of btnPenColor if user click on OK
ColorDialog c = new ColorDialog();
if (c.ShowDialog() == DialogResult.OK)
{
    btnPenColor.BackColor = c.Color;
}
```

4. Add the following code to the btnCanvasColor\_Click event handler

```
ColorDialog c = new ColorDialog();
if (c.ShowDialog() == DialogResult.OK)
{
    //Color all pixels in the bitmap with background color
    for (int x = 0; x < bmp.Width; x++)
        for (int y = 0; y < bmp.Height; y++)
            bmp.SetPixel(x, y, c.Color);
    pbDraw.BackColor = c.Color;
    btnCanvasColor.BackColor = c.Color;
}
```

5. Add the following code to the pbDraw\_MouseDown event handler

```
startPaint = true;
```

6. Add the following code to the pbDraw\_MouseMove event handler:

```
if (startPaint)
{
    //Setting the Pen BackColor and line Width
    Pen p = new Pen(btnPenColor.BackColor, float.Parse(cboPenSize.Text));

    //Associate graphics object to bitmap
    g = Graphics.FromImage bmp);

    //Drawing the line.
    g.DrawLine(p, new Point(initX ?? e.X, initY ?? e.Y), new Point(e.X, e.Y));

    //Copy bitmap to picturebox
    pbDraw.Image = bmp;

    //set beginning point for next line
    initX = e.X;
    initY = e.Y;
}
```

7. Add the following code to the pbDraw\_MouseUp event handler:

```
startPaint = false;
initX = null;
initY = null;
```

8. Add the following code to the newToolStripMenu\_Click event handler:

```
//Clearing the graphics from the picturebox(pbDraw)
g.Clear(pbDraw.BackColor);

//Create graphics object from bitmap
g = Graphics.FromImage bmp);

//Setting the BackColor of pbDraw and btnCanvasColor to White on Clicking
//New under File Menu
pbDraw.BackColor = Color.White;
btnCanvasColor.BackColor = Color.White;

//Color all pixels in the bitmap with background color
```

```

for (int x = 0; x < bmp.Width; x++)
    for (int y = 0; y < bmp.Height; y++)
        bmp.SetPixel(x, y, Color.White);

//Associate picturebox image with bitmap
pbDraw.Image = bmp;

```

9. Add the following code to the saveToolStripMenu\_Click event handler:

```

SaveFileDialog save = new SaveFileDialog();
save.Filter = "BMPfiles|*.bmp";
if (save.ShowDialog() == DialogResult.OK)
{
    pbDraw.Image.Save(save.FileName);
}

```

10. Add the following code to the openToolStripMenu\_Click event handler:

```

OpenFileDialog open = new OpenFileDialog();
open.Filter = "BMPfiles|*.bmp";
if (open.ShowDialog() == DialogResult.OK)
{
    Bitmap newImage = new Bitmap(open.FileName);
    bmp = newImage;
    pbDraw.Image = bmp;
}

```

11. Add the following code to the exitToolStripMenu\_Click event handler:

```

if (MessageBox.Show("Do you want to Exit?", "Exit",
    MessageBoxButtons.YesNo,
    MessageBoxIcon.Information) == DialogResult.Yes)
{
    Application.Exit();
}

```

12. Add the following code to the aboutUltraPaintToolStripMenu\_Click event handler:

```

About a = new About();
a.ShowDialog();

```

**When you have completed your project submit a screen shot of a painting you make.**