NANYANG POLYTECHNIC

Course: EGDF20
Module: EGE202 Application Programming

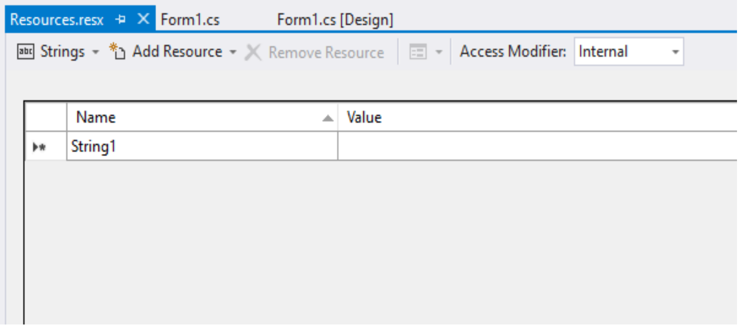
Practical 7b: Paint Application: Handling Advanced GUI Interactions

Objectives: At the end of this lab, the student should be able to describe some of the core elements and operations involved in graphics handling and advanced GUI interactions.

<u>Exercise 1 – Adding Editing Tools to Paint Program</u>

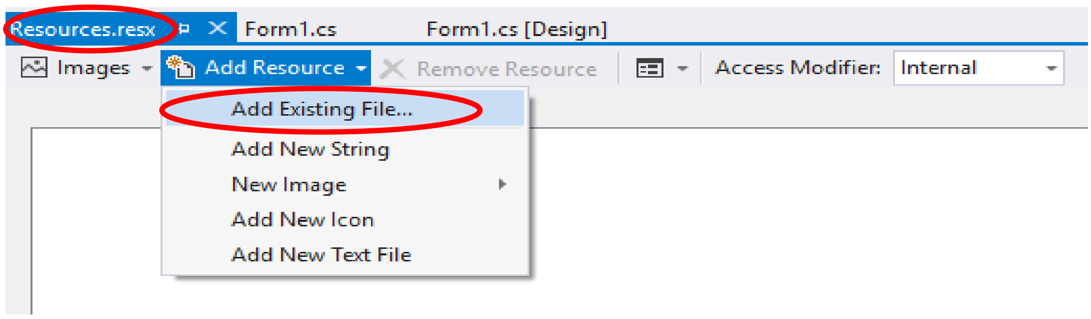
Part 1: Implementing Editing Tools

1. Using the same solution/project from Practical 7a open, from the ***Solution Explorer*** window, expand the ***Properties*** node and double click on ***Resources.resx*** to open up the ***Resources.resx*** tab.



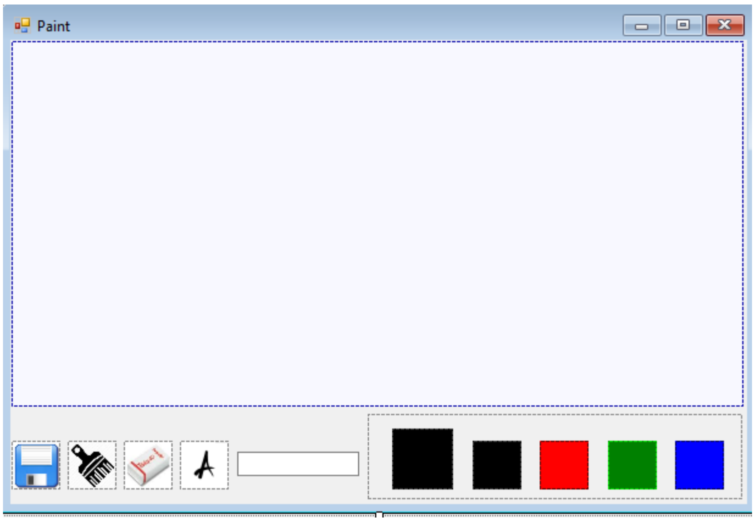
Name	Value
String1	

2. Locate “save.jpg”, “clear.jpg”, “erasor.jpg” and “text.jpg” files on your PC hard drive. From the ***Resources.resx*** tab choose ***Images*** and ***Add Resource->Add Existing File …*** to add these files as image resources that you can use in the application.



3. From the ***Toolbar***, drag in 1 *TextBox* and 4 *PictureBox* control into the *MainForm* window. Modify the properties based on the table below.

{Name} From	{Name} To	Property	Value
txtBox1	txtBoxText	{Text}	
pcitureBox1	picBoxSave	{Image}	Click … to add save.jpg
		{SizeMode}	StretchImage
pcitureBox2	picBoxClear	{Image}	Click … to add clear.jpg
		{SizeMode}	StretchImage
pcitureBox3	picBoxErase	{Image}	Click … to add erasor.jpg
		{SizeMode}	StretchImage
pcitureBox4	picBoxText	{Image}	Click … to add text.jpg
		{SizeMode}	StretchImage



4. On the ***Form Designer*** **double click** the **picBoxClear control** to create the *picBoxClear_Click()* event handler. Add the following codes:

```
private void picBoxClear_Click(object sender, EventArgs e)
{
    g = Graphics.FromImage(bm);
    Rectangle rect = picBoxMain.ClientRectangle;
    g.FillRectangle(new SolidBrush(Color.GhostWhite), rect);
    g.Dispose();
    picBoxMain.Invalidate();
}
```

5. Build and test the application.

6. On the ***Form Designer* double click** the **picBoxErase control** to create the *picBoxErase_Click()* event handler. Add the following codes:

```
private void picBoxErase_Click(object sender, EventArgs e)
{
    brush = new SolidBrush(picBoxMain.BackColor);
    picBoxBrushColor.Image = Properties.Resources.erasor;
    flagErase = true;
}
```

7. Add the following code at the starting of the **MainForm** class.

```
public partial class MainForm : Form
{
    Bitmap bm;

    Graphics g;
    Pen pen = new Pen(Color.Black, 5);
    SolidBrush brush = new SolidBrush(Color.Black);
    Point startP = new Point(0, 0);
    Point endP = new Point(0, 0);
    bool flagDraw = false;
    bool flagErase = false;
    bool flagText = false;
    string strText;
```

8. Modify the mouse move event handler.

```
private void picBoxMain_MouseMove(object sender, MouseEventArgs e)
{
    if (flagDraw == true)
    {
        endP = e.Location;
        g = Graphics.FromImage(bm);
        if (flagErase == false)
            g.DrawLine(pen, startP, endP);
        else
            g.FillRectangle(brush, endP.X, endP.Y, 20, 20);
        g.Dispose();
        picBoxMain.Invalidate();
    }
    startP = endP;
}
```

9. Build and test the paint application.

No	Actions	Observation
1	Click on the erase PictureBox	Did you spot any error in the GUI? How to fix it?
2	Rerun the application • Draw something on the application • Use the eraser to erase some part of the drawing • Choose Red color and continue drawing	Did you spot any logic error? How to fix it?

10. On the ***Form Designer*** **double click** the **picBoxText control** to create the *picBoxText_Click()* event handler. Add the following codes:

```
private void picBoxText_Click(object sender, EventArgs e)
{
    picBoxBrushColor.Image = Properties.Resources.text;
    flagDraw = false;
    flagText = true;
}
```

11. **Modify** the mouse down event handler.

```
private void picBoxMain_MouseDown(object sender, MouseEventArgs e)
{
    startP = e.Location;
    if (flagText == false)
    {
        if (e.Button == MouseButtons.Left)
            flagDraw = true;
    }
    else
    {
        strText = txtBoxText.Text;
        g = Graphics.FromImage(bm);
        Font font = new Font("Arial", 12);
        brush = new SolidBrush(Color.Black);
        g.DrawString(strText, font, brush, startP.X, startP.Y);
        g.Dispose();
        picBoxMain.Invalidate();
    }
}
```

12. Build and test the paint application.

No	Actions	Observation
1	Click on the Text PictureBox Input the word “Hello” Click anywhere on the Paint program.	
2	Next, choose Red color and continue drawing	Did you spot any logic error? How to fix it?
3	Next, try erasing	Did you spot any logic error? How to fix it?

Exercise 2 – Programming with Common Dialog Classes (SaveFileDialog)

Part 1: Saving the Paint Artwork to File

1. On the ***Form Designer*** **double click** the **picBoxSave control** to create the *picBoxSave_Click()* event handler. Add the following codes:

```
private void picBoxSave_Click(object sender, EventArgs e)
{
    using (SaveFileDialog sfdlg = new SaveFileDialog())
    {
        sfdlg.Title = "Save Dialog";
        sfdlg.Filter = "Image Files(*.BMP)|*.BMP|All files (*.*)|*.*";
        if (sfdlg.ShowDialog(this) == DialogResult.OK)
        {
            using (Bitmap bmp = new Bitmap(picBoxMain.Width, picBoxMain.Height))
            {
                Rectangle rect = new Rectangle(0, 0, bmp.Width, bmp.Height);
                picBoxMain.DrawToBitmap(bmp, rect);
                bmp.Save(sfdlg.FileName, ImageFormat.Bmp);
                MessageBox.Show("File Saved Successfully");
            }
        }
    }
 }
```

2. Run and test the application.
3. Save the Paint artwork and verify that the image file is created.