Graphics Device Interface

Also known as the GDI

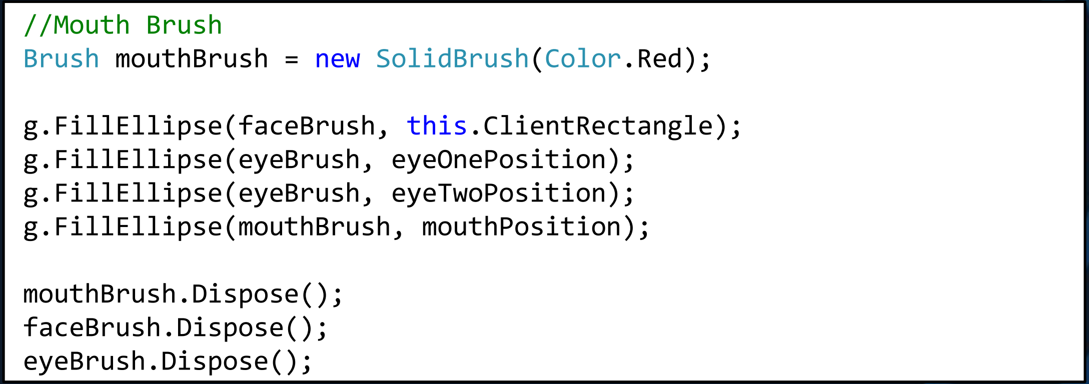
Canberra Institute of Technology

aie

SPECIALIST EDUCATORS IN
GAMES, ANIMATION & FILM VFX

Contents

- What is GDI?
- Using GDI
- GDI and Windows Forms.
- A few examples
- Summary
- References



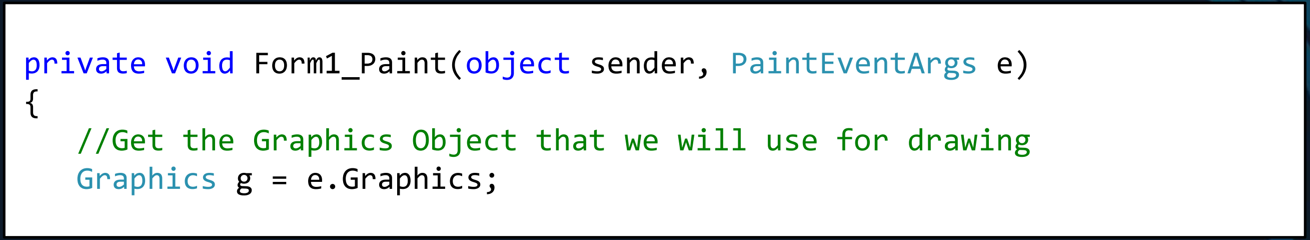
```
//Mouth Brush
Brush mouthBrush = new SolidBrush(Color.Red);

g.FillEllipse(faceBrush, this.ClientRectangle);
g.FillEllipse(eyeBrush, eyeOnePosition);
g.FillEllipse(eyeBrush, eyeTwoPosition);
g.FillEllipse(mouthBrush, mouthPosition);

mouthBrush.Dispose();
faceBrush.Dispose();
eyeBrush.Dispose();
```

What is GDI?

- The GDI (Graphic Device Interface) is a set of common controls built to enable graphics on both screens and printers.

- Supported through the Common Language Runtime.

Using GDI

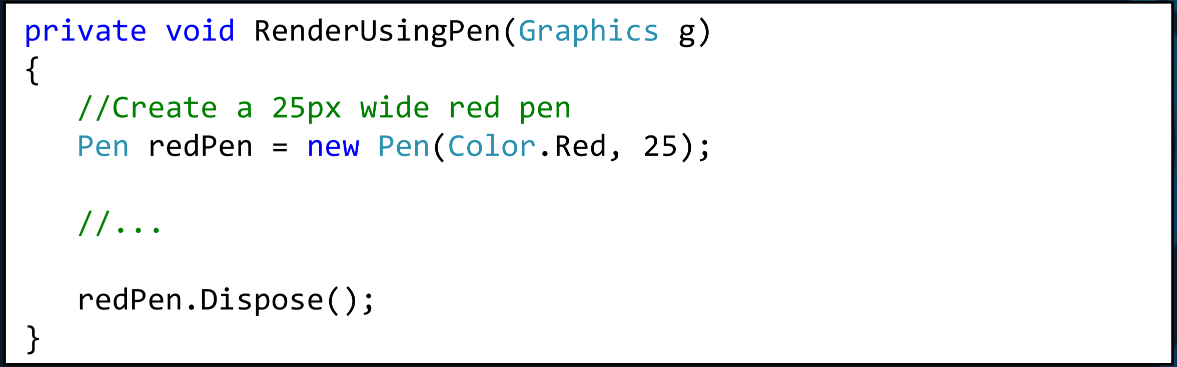


```
private void Form1_Paint(object sender, PaintEventArgs e)
{
   //Get the Graphics Object that we will use for drawing
   Graphics g = e.Graphics;
```

- All GDI draw functions are called on a Graphics object owned by the form/control that is to be modified.

- Windows Forms makes accessing the Graphics object for the form very easy

Using GDI



```
private void RenderUsingPen(Graphics g)
{
    //Create a 25px wide red pen
    Pen redPen = new Pen(Color.Red, 25);

    //...

    redPen.Dispose();
}
```

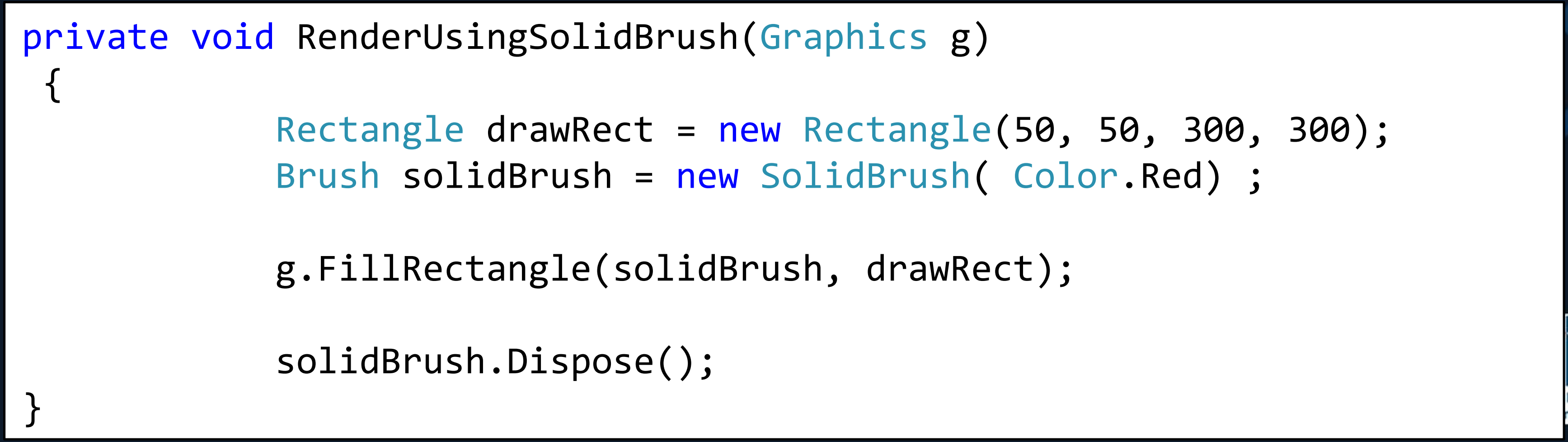
- Selecting the colour of the rendered shape is done using Pens and Brushes.

- Different types of Pens and Brushes produce different colours and effects when they are used.

Brushes

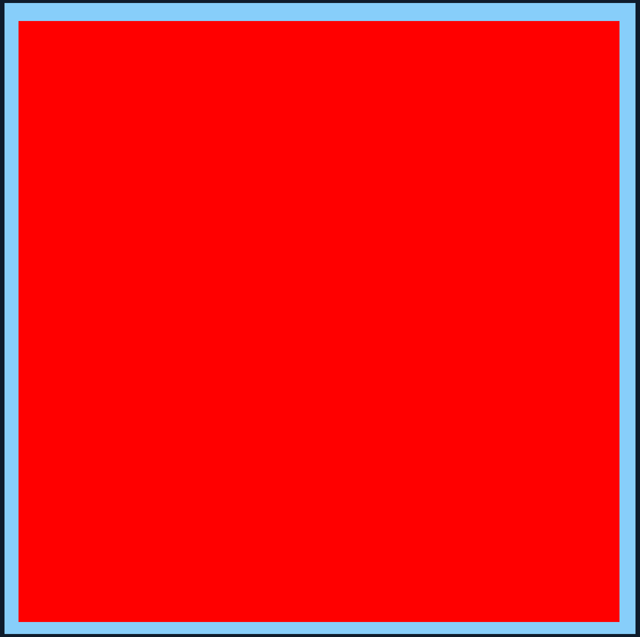
- Brushes are used for all “Fill<type>” functions.
- There are different types of Brushes that produce different effects.

Brush Examples - SolidBrush



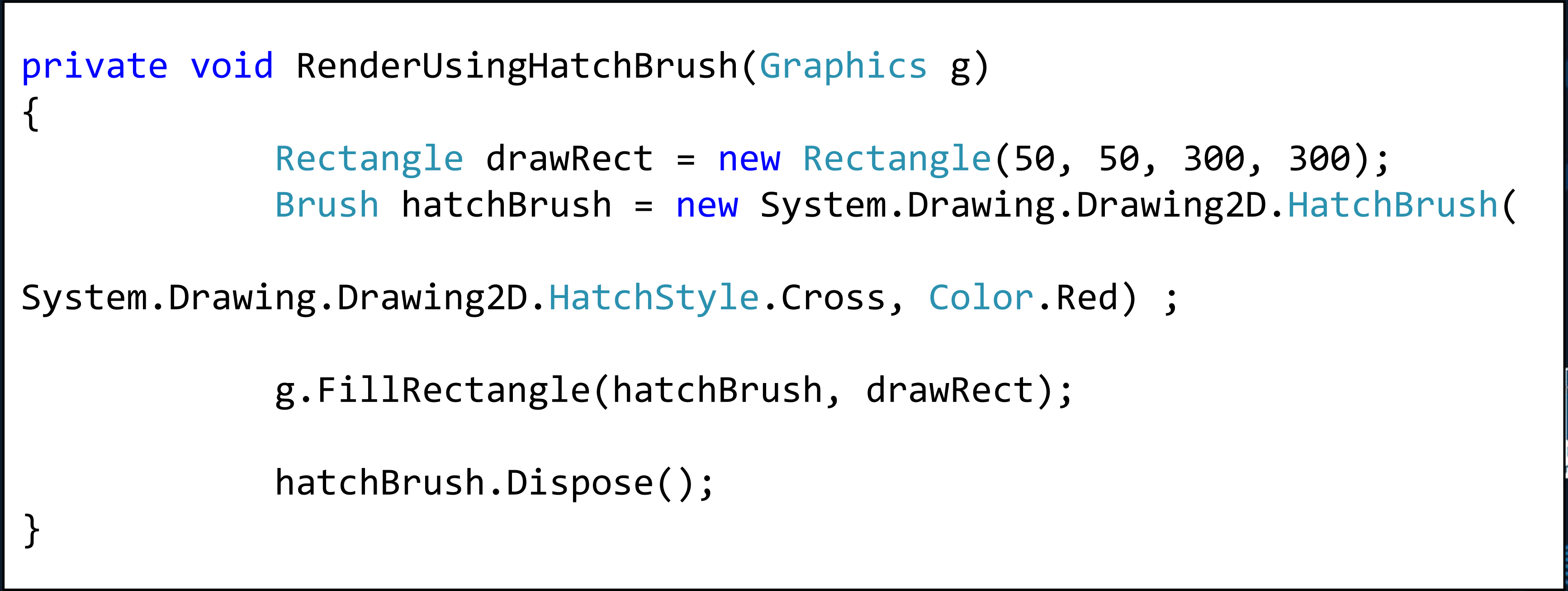
```
private void RenderUsingSolidBrush(Graphics g)
 {
            Rectangle drawRect = new Rectangle(50, 50, 300, 300);
            Brush solidBrush = new SolidBrush( Color.Red) ;

            g.FillRectangle(solidBrush, drawRect);

            solidBrush.Dispose();
}
```

Brush Examples - SolidBrush



HatchBrush

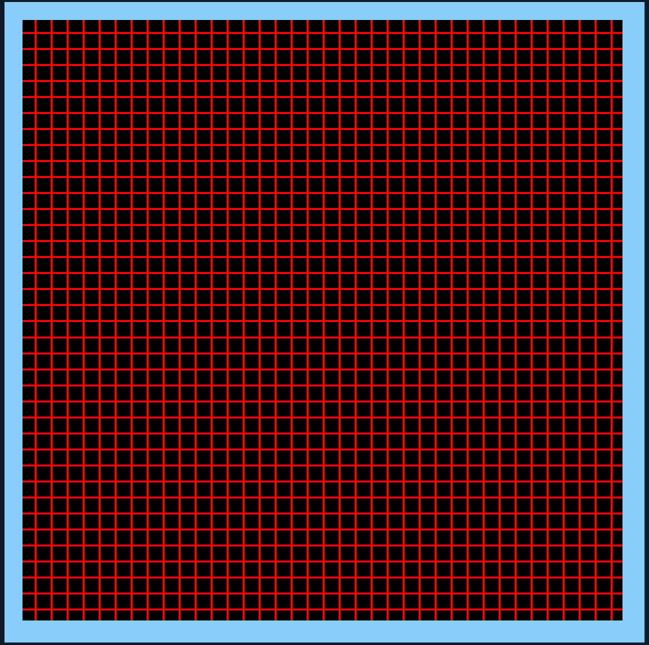
Brush Examples - HatchBrush



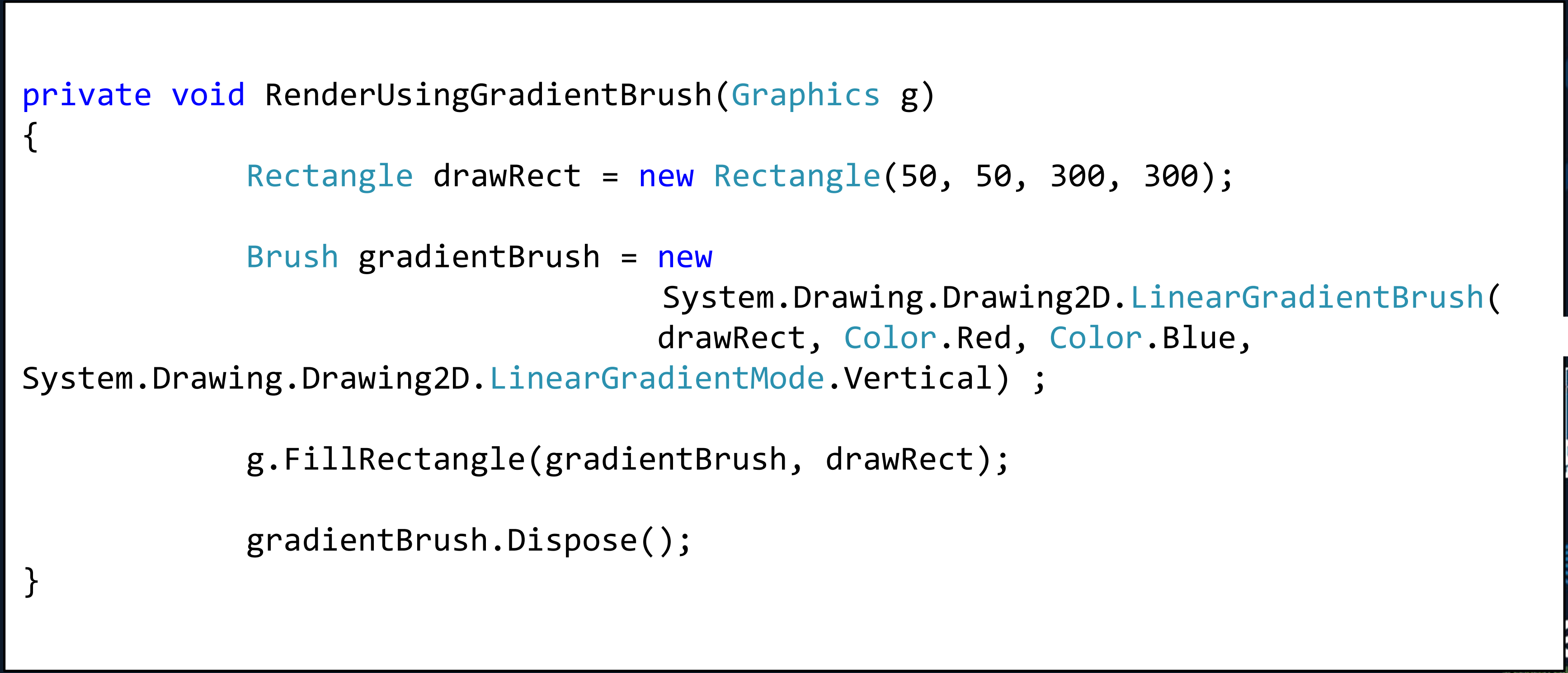
```
private void RenderUsingHatchBrush(Graphics g)
{
            Rectangle drawRect = new Rectangle(50, 50, 300, 300);
            Brush hatchBrush = new System.Drawing.Drawing2D.HatchBrush(

System.Drawing.Drawing2D.HatchStyle.Cross, Color.Red) ;

            g.FillRectangle(hatchBrush, drawRect);

            hatchBrush.Dispose();
}
```

HatchBrush

Brush Examples - HatchBrush

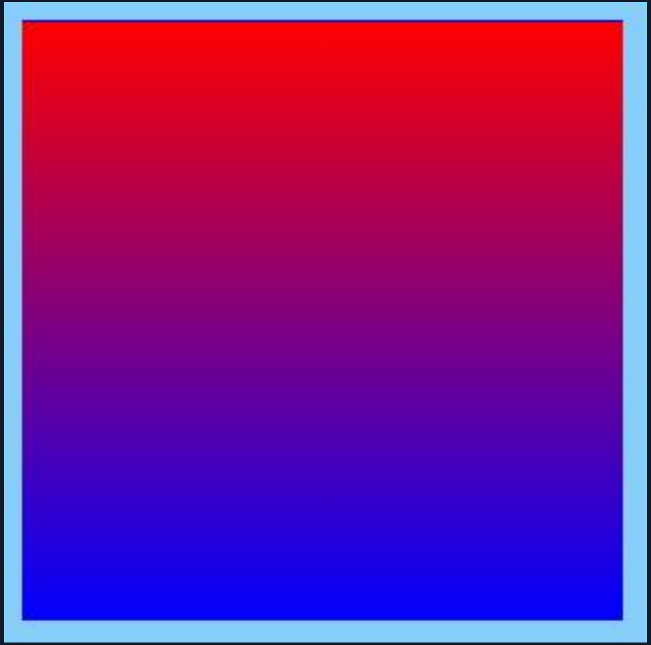


Brush Examples – LinearGradientBrush



```
private void RenderUsingGradientBrush(Graphics g)
{
            Rectangle drawRect = new Rectangle(50, 50, 300, 300);

            Brush gradientBrush = new
                                  System.Drawing.Drawing2D.LinearGradientBrush(
                                  drawRect, Color.Red, Color.Blue,
System.Drawing.Drawing2D.LinearGradientMode.Vertical) ;

            g.FillRectangle(gradientBrush, drawRect);

            gradientBrush.Dispose();
}
```

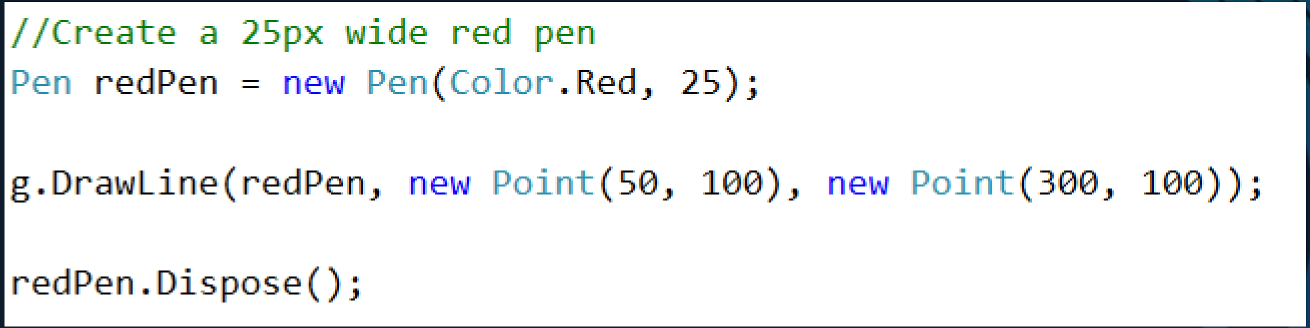
Brush Examples – LinearGradientBrush



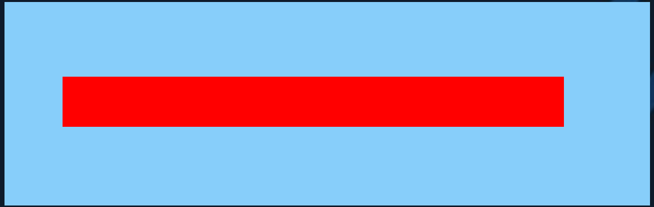
Other Brush Types

- **TextureBrush –** Render a pattern based on a loaded image.
- **PathGradientBrush –** Much more complex version of the Gradient Brush

Pens



```
//Create a 25px wide red pen
Pen redPen = new Pen(Color.Red, 25);

g.DrawLine(redPen, new Point(50, 100), new Point(300, 100));

redPen.Dispose();
```



- Pens are used by all Draw<type> functions

- They can also be used drawing outlined shapes.

Shapes

- FillRectangle
- FillEclipse
- FillPie
- FillPath
- FillPolygon

- DrawLine
- DrawRectangle
- DrawEcilpse
- DrawPie
- DrawPath

And more...

GDI and Windows Forms

- Override the Paint Event:

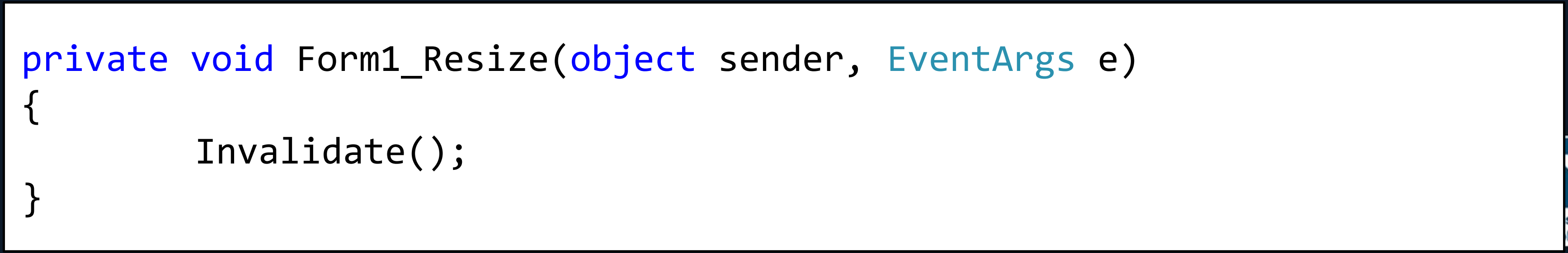
```
private void Form1_Paint(object sender, PaintEventArgs e)
{
            //Get the Graphics Object that we will use for drawing
            Graphics g = e.Graphics;
```

Window bounds can be found using the ClientRectangle:

```
 //This holds the area of the window
Rectangle clientArea = this.ClientRectangle;
```

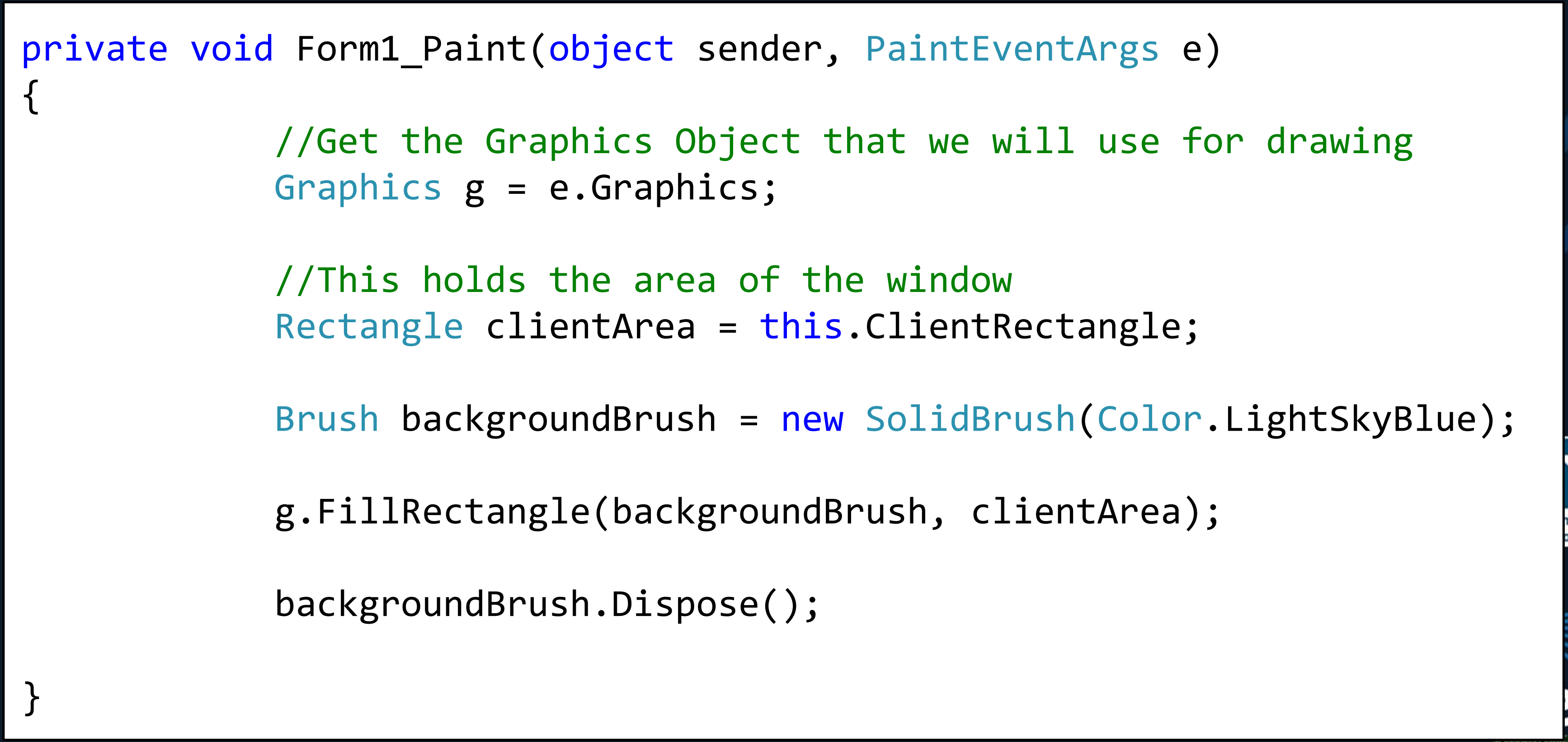
GDI and Windows Forms

- Call the Invalidate() function to force a window redraw.



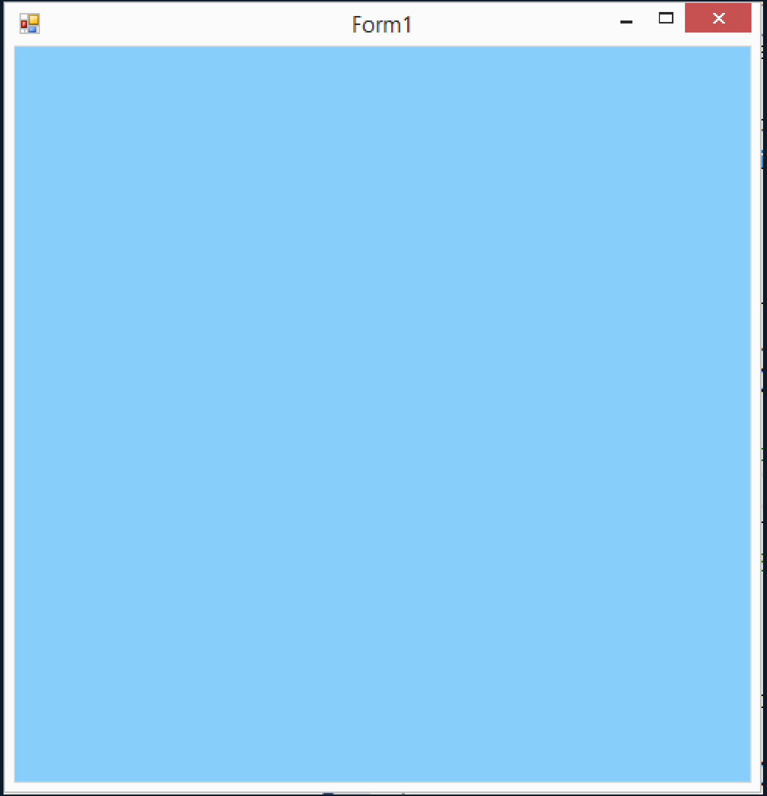
```
private void Form1_Resize(object sender, EventArgs e)
{
        Invalidate();
}
```

A few examples – Colour the background

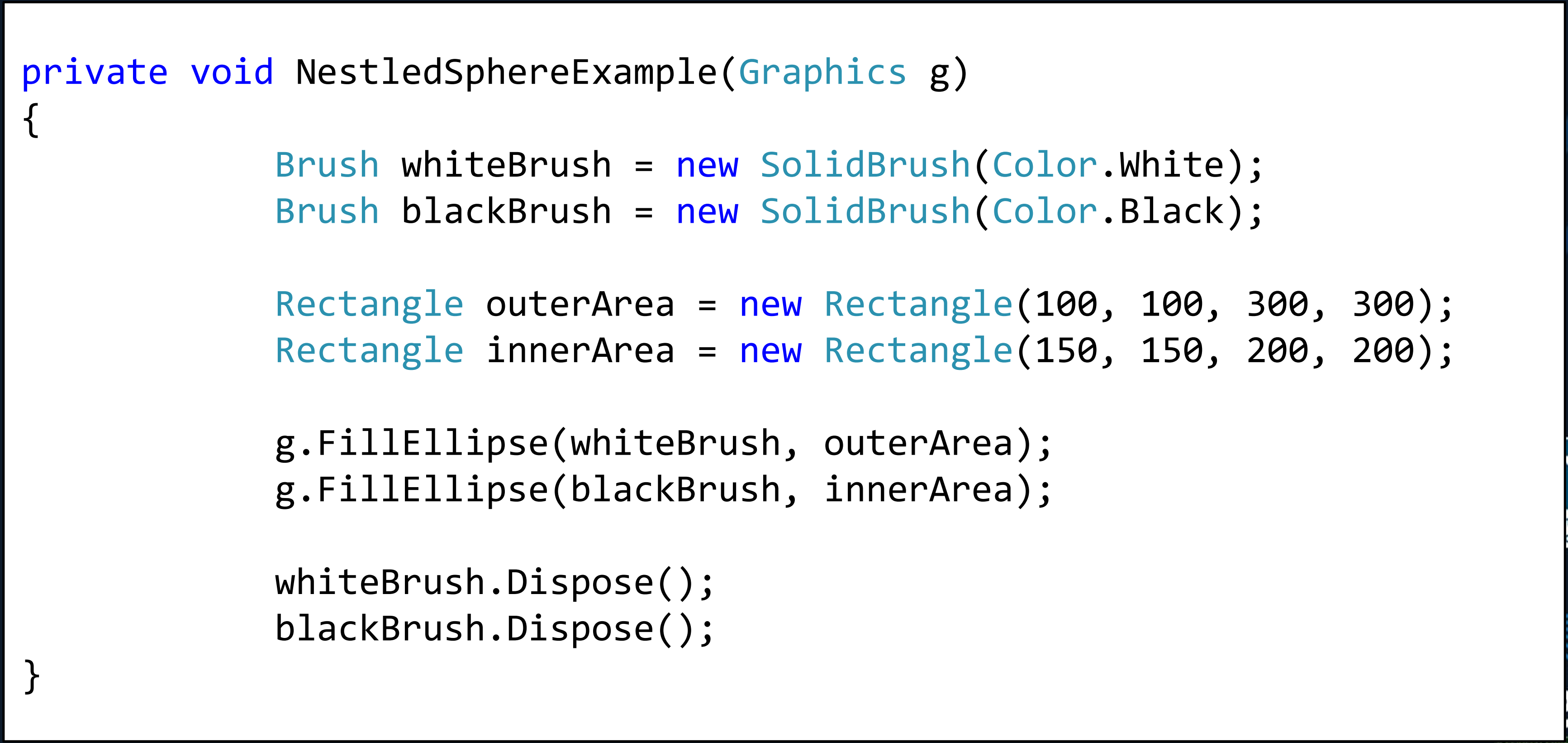


```
private void Form1_Paint(object sender, PaintEventArgs e)
{
            //Get the Graphics Object that we will use for drawing
            Graphics g = e.Graphics;

            //This holds the area of the window
            Rectangle clientArea = this.ClientRectangle;

            Brush backgroundBrush = new SolidBrush(Color.LightSkyBlue);

            g.FillRectangle(backgroundBrush, clientArea);

            backgroundBrush.Dispose();

}
```

A few examples – Colour the background

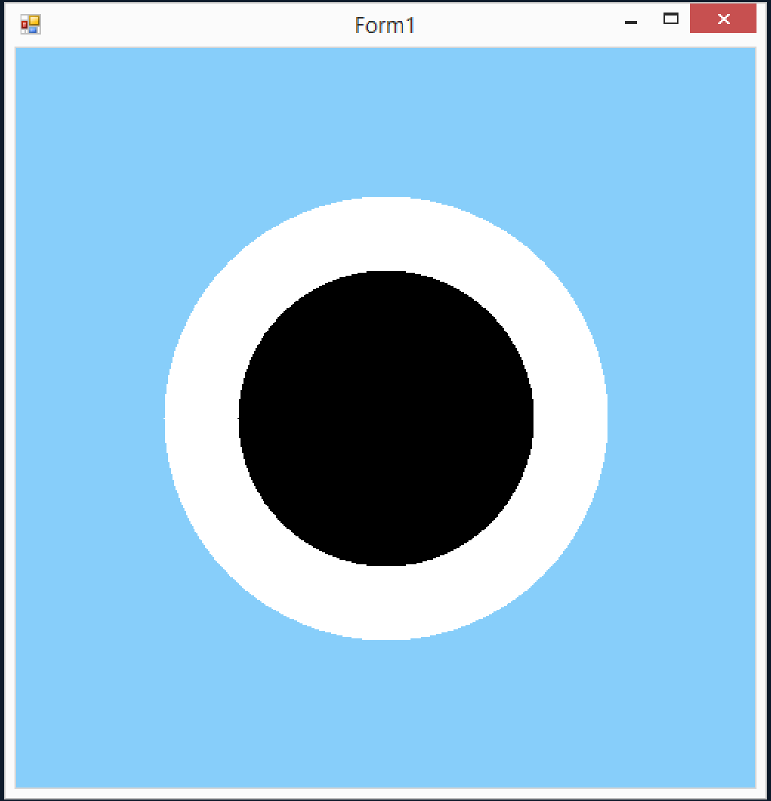


A few examples – Nestled Spheres



```
private void NestledSphereExample(Graphics g)
{
            Brush whiteBrush = new SolidBrush(Color.White);
            Brush blackBrush = new SolidBrush(Color.Black);

            Rectangle outerArea = new Rectangle(100, 100, 300, 300);
            Rectangle innerArea = new Rectangle(150, 150, 200, 200);

            g.FillEllipse(whiteBrush, outerArea);
            g.FillEllipse(blackBrush, innerArea);

            whiteBrush.Dispose();
            blackBrush.Dispose();
}
```

A few examples – Nestled Spheres



Summary

- The Graphics Device Interface is a series of classes that allow us to draw to the screen (or printer)
- Part of the Common Runtime Libraries.
- Pens and Brushes can be used to Fill and Draw shapes of different kinds.

References

- Microsoft, 2015, *Graphics and Drawing in Windows Forms*, MSDN
  - https://msdn.microsoft.com/en/us/library/a36fascx%28v=vs.110%29.aspx