WINDOWS PROGRAMMING (With C#.NET)

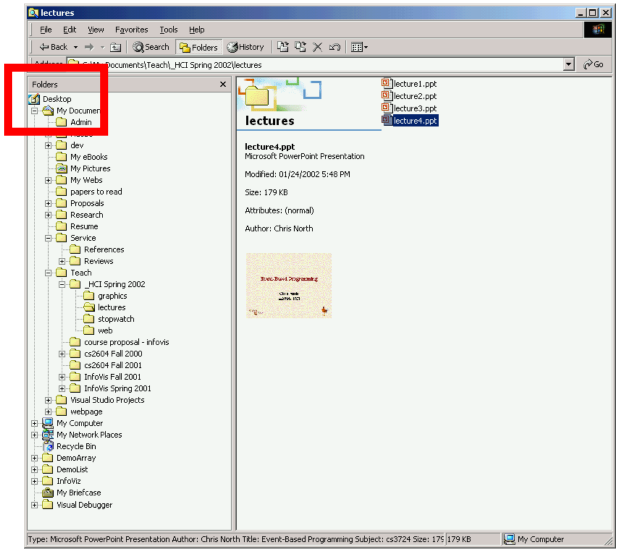
Chapter 3:
Graphics and Timers

TS. Lê Văn Vinh
Khoa Công nghệ Thông tin
Trường ĐHSPKT TP. HCM

Contents

- **Graphics**
- **Timers**

Pixels



lecture4.ppt
Microsoft PowerPoint Presentation

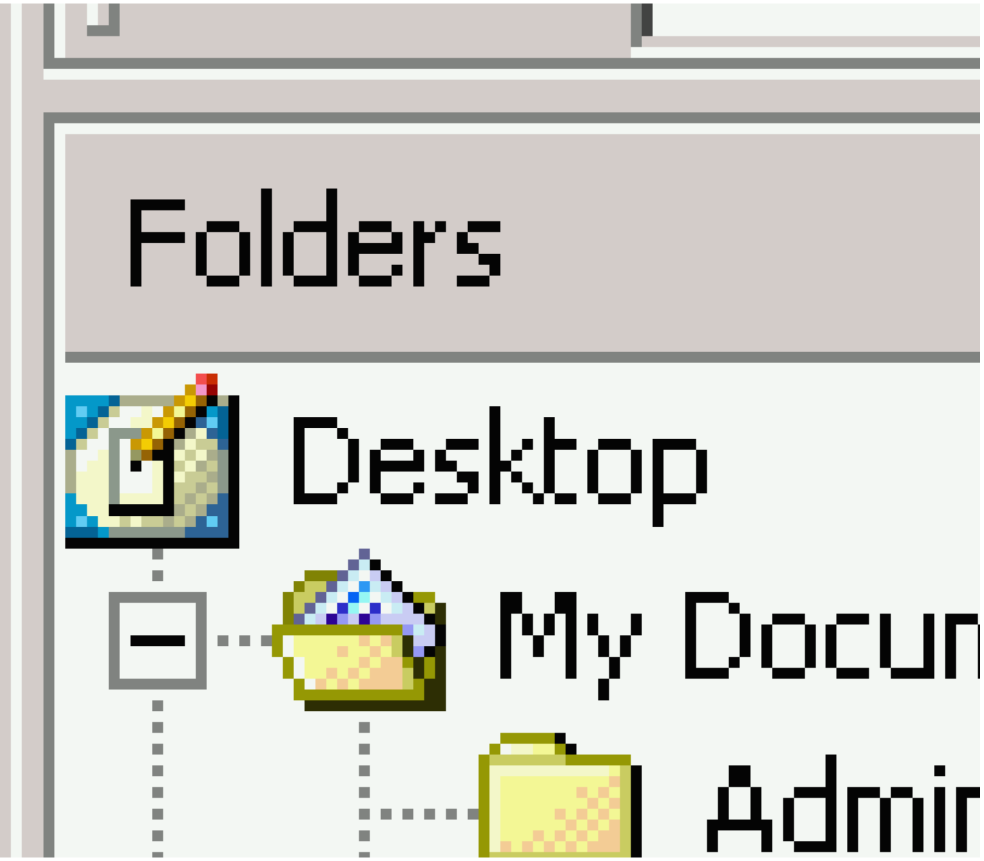
Modified: 01/24/2002 5:48 PM

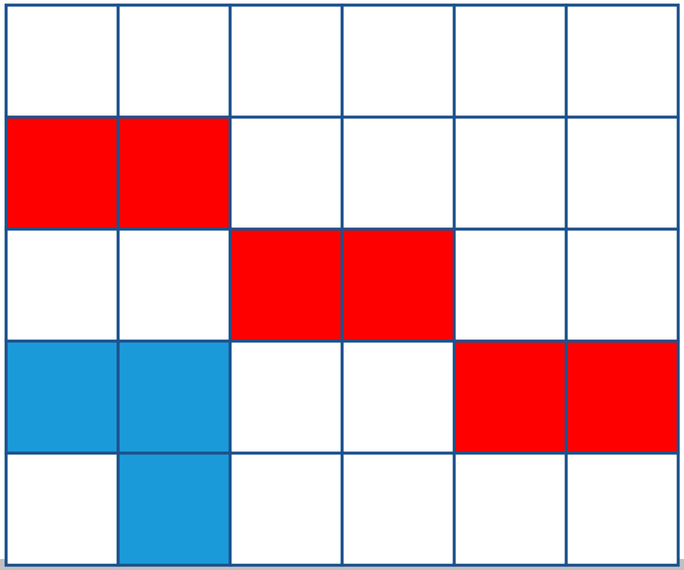
Size: 179 KB

Attributes: (normal)

Author: Chris North

Type: Microsoft PowerPoint Presentation Author: Chris North Title: Event-Based Programming Subject: cs3724 Size: 179 | 179 KB | My Computer

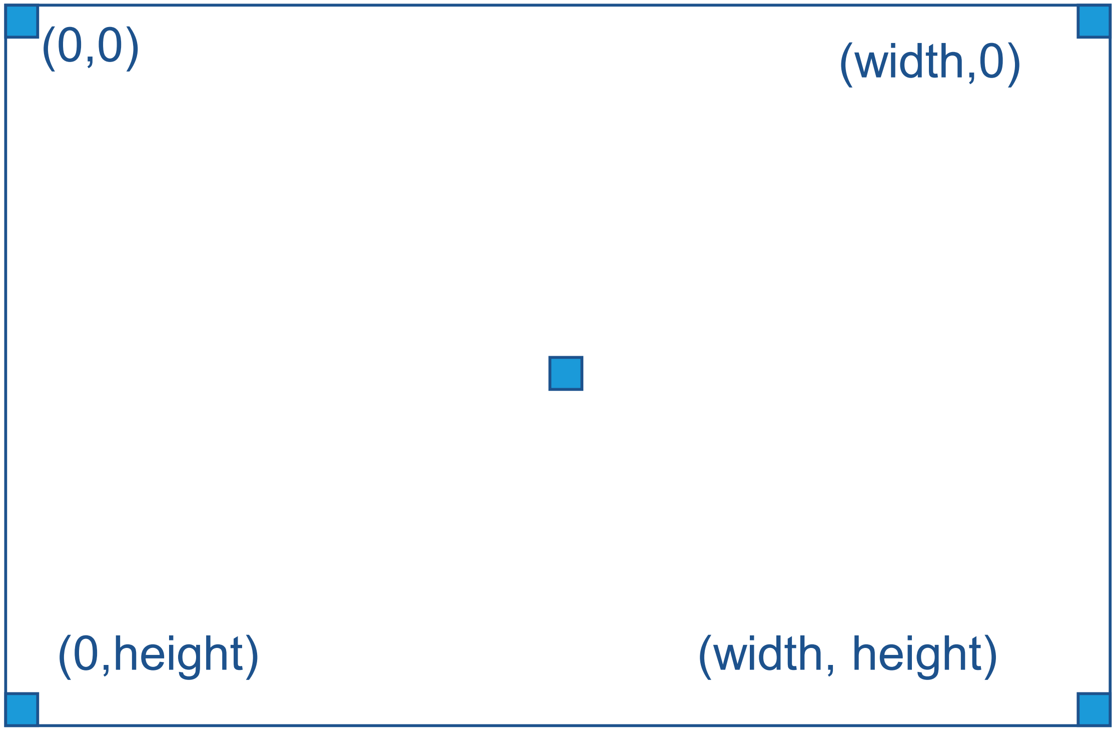




⬜	⬜	⬜	⬜	⬜	⬜
🟥	🟥	⬜	⬜	⬜	⬜
⬜	⬜	🟥	🟥	⬜	⬜
🟦	🟦	⬜	⬜	🟥	🟥
⬜	🟦	⬜	⬜	⬜	⬜

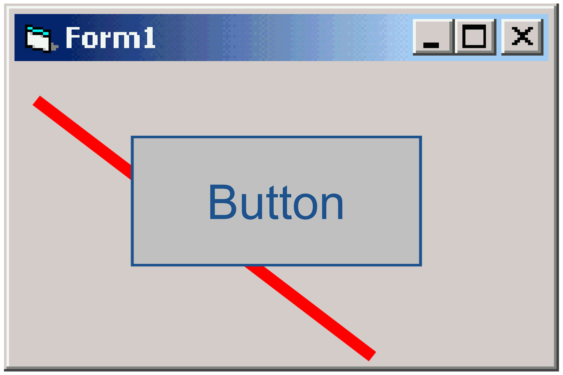
Coordinate System

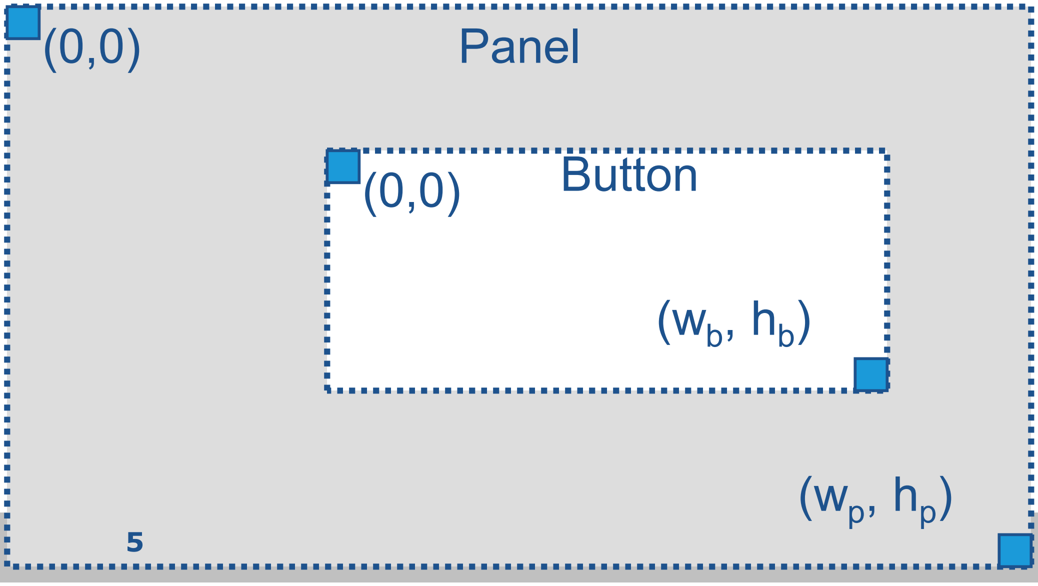
- Upside-down Cartesian



Component Hierarchy

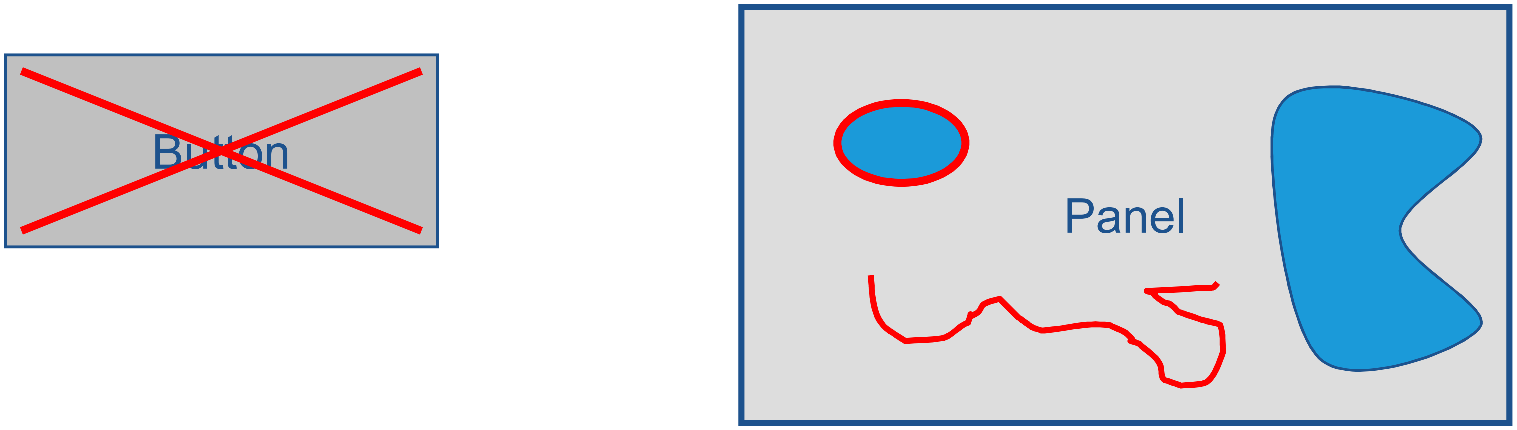
- Each component has its own subwindow
  - Subwindow = rectangular area within parent component
  - Has own coordinate system
- Clipping:
  - Can’t paint outside its subwindow
  - Can’t paint over child components





Painting Components

- Can paint any component
- Panel is good for custom graphics area



Painting in C#

1. The GDI+ graphics library:

```
using System.Drawing;
```

2. Get the “graphics context” of a component:

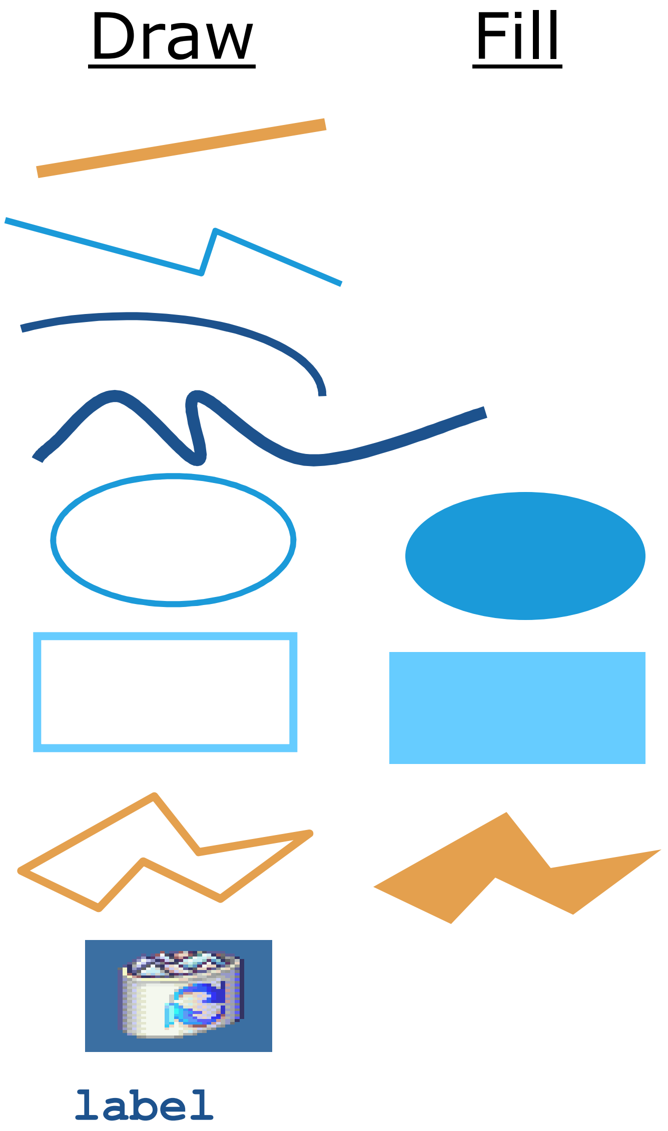
```
Graphics g = myPanel.CreateGraphics();
```

3. Paint in it:

```
g.DrawLine(pen, x1,y1, x2,y2);
```

Graphics Primitives

- Line (pt1,pt2)
- Lines (pt list)
- Arc (rect)
- Curves, Bezier (pt list)
- Ellipse (rect)
- Rectangle (rect)
- Polygon (pt list)
- Image (img, x,y)
- String (string, x,y)

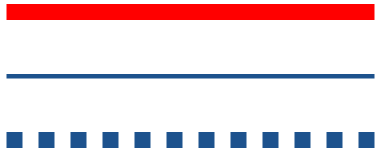


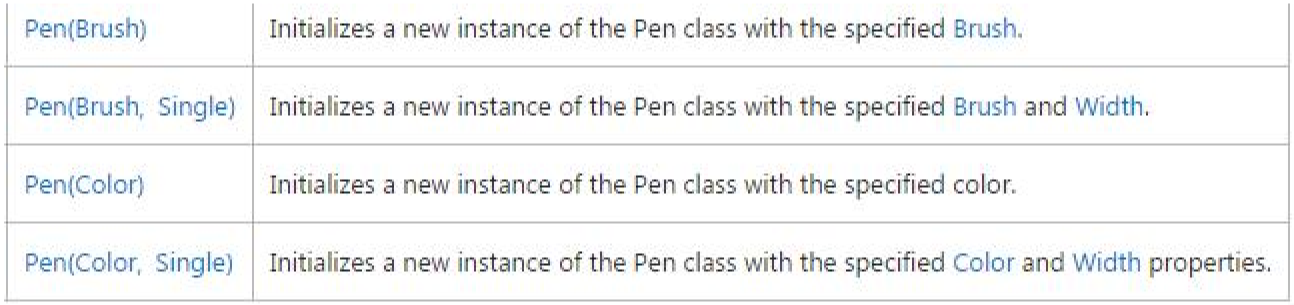
Draw Fill

label

Graphics Attributes

- Pen (for lines)
  - Properties
    - Color
    - Width
    - DashStyle
    - …
  - Constructors





Pen(Brush)	Initializes a new instance of the Pen class with the specified Brush.
Pen(Brush, Single)	Initializes a new instance of the Pen class with the specified Brush and Width.
Pen(Color)	Initializes a new instance of the Pen class with the specified color.
Pen(Color, Single)	Initializes a new instance of the Pen class with the specified Color and Width properties.

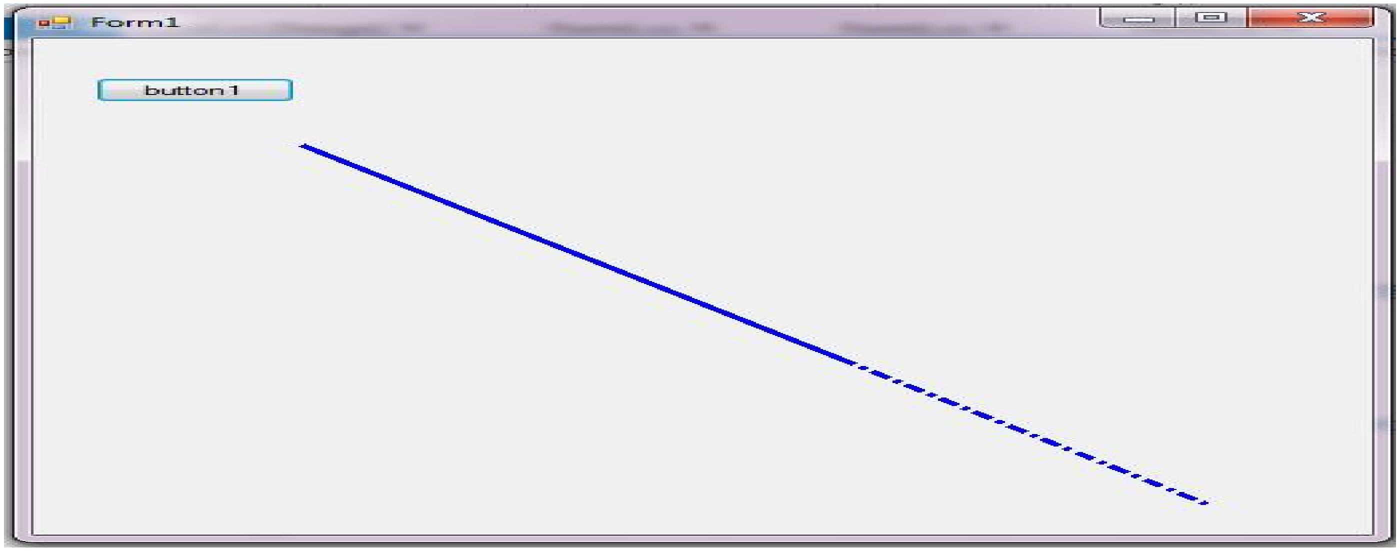
  - Methods: https://msdn.microsoft.com/en-us/library/system.drawing.pen(v=vs.110).aspx

Graphics Attributes

- Pen (for lines)
  - Example

```
Graphics gp = this.CreateGraphics();
Pen myPen = new Pen(Color.Blue, 3);
gp.DrawLine(myPen, 100, 100, 300, 300);

myPen.DashStyle = System.Drawing.Drawing2D.DashStyle.DashDot;
gp.DrawLine(myPen, 300, 300, 500, 500);
```



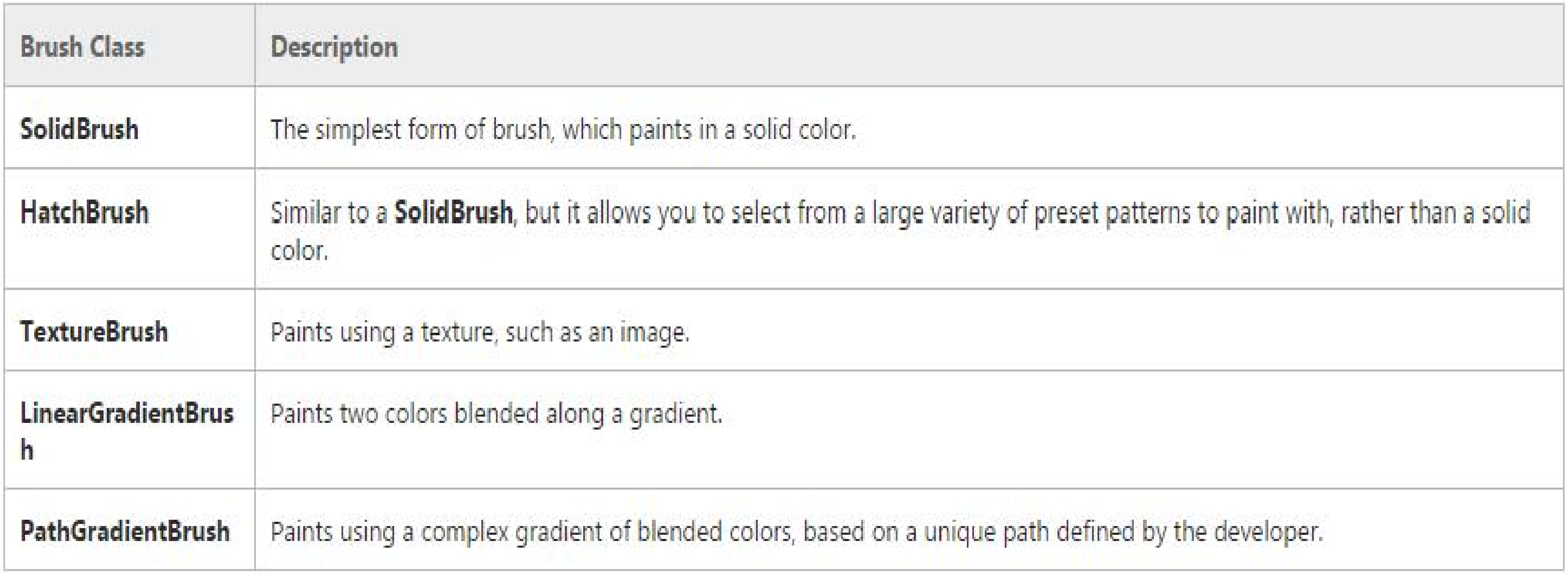
Graphics Attributes

- Pen (for lines)
  - DrawArc()
  - DrawBezier()
  - DrawClosedCurve()
  - DrawEllipse()
  - DrawIcon()
  - DrawImage()
  - DrawLine()
  - DrawLines()
  - DrawRectangle()
  - DrawPolygon()
  - DrawString()
  - .....

Graphic Atributes

Brush

- https://msdn.microsoft.com/en-us/library/aa983677(v=vs.71).aspx



Brush Class	Description
SolidBrush	The simplest form of brush, which paints in a solid color.
HatchBrush	Similar to a **SolidBrush**, but it allows you to select from a large variety of preset patterns to paint with, rather than a solid color.
TextureBrush	Paints using a texture, such as an image.
LinearGradientBrush	Paints two colors blended along a gradient.
PathGradientBrush	Paints using a complex gradient of blended colors, based on a unique path defined by the developer.

Graphic Atributes

Brush

- **Examples:**
- **SolidBrush**

- SolidBrush myBrush = new SolidBrush(myColor);
- gp.FillEllipse(myBrush, 70, 70, 90, 90);

- **HatchBrush**
- System.Drawing.Drawing2D.HatchBrush myBrush = new System.Drawing.Drawing2D.HatchBrush(System.Drawing.
  Drawing2D.HatchStyle.Plaid, Color.Red, Color.Blue);
- gp.FillEllipse(myBrush, 70, 70, 90, 90);
- https://msdn.microsoft.com/en-us/library/aa983677(v=vs.71).aspx

Graphics Attributes

- Brush (for filling)
  - FillEllipse()
  - FillPie()
  - FillPolygon()
  - FillRectangle()
  - FillRegion()

Color

- Combinations of Red, Green, Blue
- Each in [0, 255]

- C#:
  - Color myColor = Color.FromArgb(255, 150, 0)
- Color myColor = Color.Black;
-

Timer

```
public partial class Form1 : Form
    {
        System.Windows.Forms.Timer t = new System.Windows.Forms.Timer();
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            t.Interval = 1500; // specify interval time as you want
            t.Tick += new EventHandler(timer_Tick);
            t.Start();
        }
        private void timer_Tick(object sender, EventArgs e)
        {
            //Call method
            MessageBox.Show("ksd");
        }
```