

Công nghệ .NET

Bài 12 – Graphics 2D, Brush

Nguyễn Thành Trung – Khoa CNTT

Email: trung.nguyenthanh1@phenikaa-uni.edu.vn

WPF

- Graphics 2D
- Brush

Graphic-2D

- Shape:
 - Là một loại UIElement cho phép bạn vẽ hình dạng ra màn hình. Bởi vì chúng là các phần tử UI, các đối tượng Shape có thể được sử dụng bên trong các phần tử Panel và hầu hết các điều khiển.
 - Properties
 - Stroke: Mô tả cách vẽ đường viền của hình dạng.
 - StrokeThickness: Mô tả độ dày của đường viền của hình dạng.
 - Fill: Mô tả cách sơn bên trong của hình dạng.
 - Các thuộc tính dữ liệu để chỉ định tọa độ và đỉnh, được đo bằng pixel và không phụ thuộc vào thiết bị.

Graphic-2D

- Các đối tượng đồ hoạ cơ bản
 - Line (đoạn thẳng)
 - Ellipse (hình elip)
 - Polygon (đa giác)
 - Polyline (chuỗi đoạn thẳng)
 - Rectangle (chữ nhật)
 - Path (hình phức hợp)
- ⇒ kế thừa từ đối tượng cơ sở Shape

Graphic-2D

- Các đối tượng đồ hoạ cơ bản: một số thuộc tính chung
 - Stroke: Mô tả màu sắc đường viền của một hình hoặc màu của một đoạn thẳng.
 - StrokeThickness: Độ dày của đường viền.
 - Fill: Cách tô phần bên trong của một hình.
 - Data: Mô tả các tọa độ, các đỉnh của một hình, đơn vị đo là pixel.

Graphic-2D

- Line:

- Cho phép vẽ một đường thẳng giữa hai điểm.
- Ví dụ: Cho một panel Canvas, vẽ các line cơ bản

```
<Line
  X1="10" Y1="10"
  X2="50" Y2="50"
  StrokeThickness="4"
  Canvas.Left="112" Canvas.Top="102">
  <Line.Stroke>
    <RadialGradientBrush GradientOrigin="0.5,0.5" Center="0.5,0.5" RadiusX="0.5" RadiusY="0.5">
      <RadialGradientBrush.GradientStops>
        <GradientStop Color="Red" Offset="0" />
        <GradientStop Color="Blue" Offset="0.25" />
      </RadialGradientBrush.GradientStops>
    </RadialGradientBrush>
  </Line.Stroke>
</Line>
```

Toạ độ

Độ dày

Các thuộc tính màu sắc

```
<Line
  X1="10" Y1="60"
  X2="150" Y2="60"
  Stroke="Black"
  StrokeThickness="4" Canvas.Left="67" Canvas.Top="102"/>
```

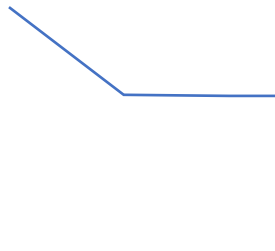
Khoảng cách so với panel

Graphic-2D

- Polyline:

- Chuỗi nhiều đoạn thẳng liên tiếp nối với nhau.
- Một Polyline gồm N đoạn thẳng thì được định nghĩa bởi N+1 điểm.
- Ví dụ

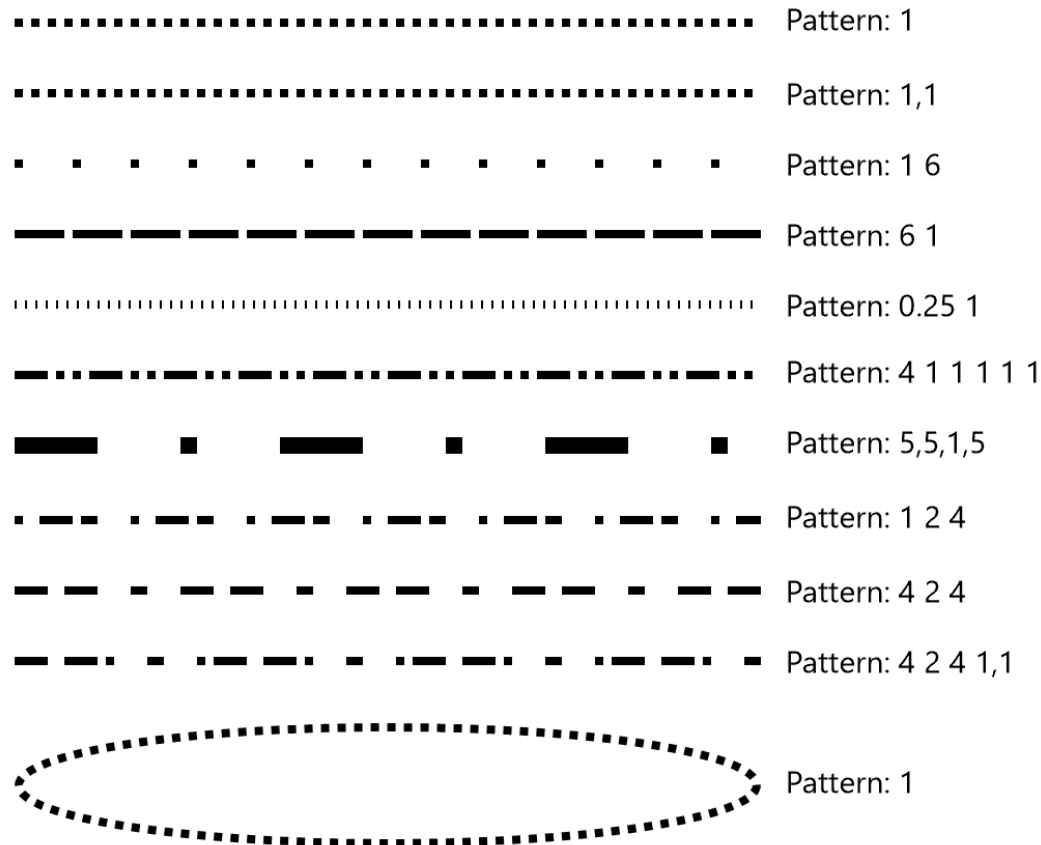
```
<Polyline  
  Points="10,110 110,110 110,10" Stroke="■"Blue"  
  StrokeThickness="4"  
  StrokeDashArray="4 1 2 1"  
  Canvas.Left="92" Canvas.Top="178" />
```



Một polyline gồm hai đoạn nét
đứt nối nhau với tọa độ ba đỉnh
(10,110) (110,110) (110,10)

Graphic-2D

- Shape.StrokeDashArray Property

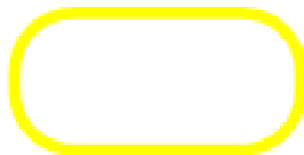


Graphic-2D

- Rectangle (hình chữ nhật)

- được xác định bởi tọa độ của góc trên trái và độ rộng, độ cao
- các thuộc tính đi kèm cho đường viền (màu sắc, độ dày, kiểu dáng) và tô phần bên trong của hình.

- Ví dụ



```
<Rectangle
  Width="100"
  Height="50"
  Fill=■"Blue"
  Canvas.Left="10"
  Canvas.Top="25" />
<Rectangle
  Width="100"
  Height="50"
  RadiusX="20"
  RadiusY="20"
  Stroke=■"Yellow"
  StrokeThickness="4"
  Canvas.Left="10"
  Canvas.Top="115" />
```

Graphic-2D

- Rectangle (hình chữ nhật)
 - Các thuộc tính RadiusX, RadiusY là bán kính của hình ellipse để tạo ra các góc tròn của hình chữ nhật.
 - Ví dụ:
 - Vẽ hình chữ nhật có đường viền nét đứt màu đỏ độ dày 6 pixel, được tô màu xanh lá
 - Vẽ hình chữ nhật các góc vát tròn có đường viền nét đứt màu hồng độ dày 3 pixel, không được tô màu

Graphic-2D

- Ellipse, Circle

- Xác định bởi tọa độ của góc trên trái và độ rộng, độ cao của hình chữ nhật ngoại tiếp của Ellipse.
- Đặc biệt: hình tròn là hình Ellipse với chiều rộng và chiều cao bằng nhau.
- Ví dụ

```
<Ellipse  
  Width="100"  
  Height="50"  
  Fill=■ "Blue"  
  Canvas.Left="145"  
  Canvas.Top="25" />
```

Độ rộng, độ cao

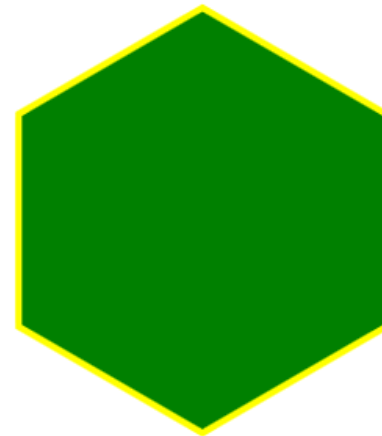
Tọa độ góc trên trái

Graphic-2D

- Ellipse, Circle
 - Ví dụ: Vẽ hình tròn rỗng có viền xanh nét đứt.

Graphic-2D

- Polygon: Một Polygon N đỉnh được định nghĩa bởi một tập hợp N cặp tọa độ tương ứng với mỗi đỉnh của nó.



```
<Polygon
  Name="hexagon"
  Stroke=■"Yellow"
  StrokeThickness="4.5"
  Fill=■"Green"
  Points="176,30 302.44,103 302.44,249 176,322 49.5603,249 49.5603,103"
  Canvas.Left="340" Canvas.Top="-10" Height="331" Width="315" />
```

Graphic-2D

- Bezier (đường cong)
 - Sử dụng đối tượng Path để tạo nên những hình phức tạp, gồm nhiều phần nối với nhau.
 - Ví dụ: sử dụng đối tượng Path tạo nên một hình gồm một đường cong Bezier, cuối đường cong là một đoạn thẳng nối ngược trở lại điểm đầu.

<Path

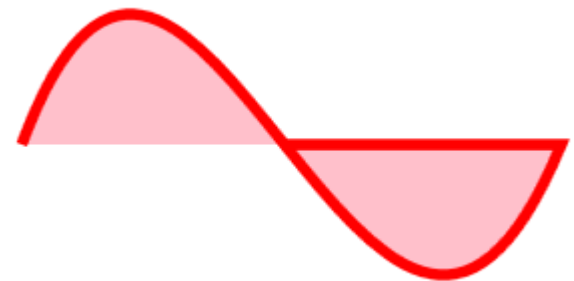
Data="M 10,100 C 100,0 200,200 300,100 H 150"

Stroke=■"Red"

StrokeThickness="6"

Fill=■"Pink"

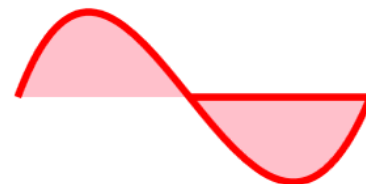
Canvas.Top="56.005" Canvas.Left="319" Height="145.995" Stretch="Fill"/>



Graphic-2D

- Bezier (đường cong)

```
<Path
  Data="M 10,100 C 100,0 200,200 300,100 H 150"
  Stroke=■"Red"
  StrokeThickness="6"
  Fill=■"Pink"
  Canvas.Top="56.005" Canvas.Left="319" Height="145.995" Stretch="Fill"/>
```



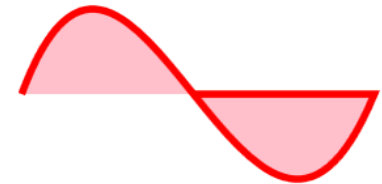
- Các thuộc tính của Data:

- M: đường cong bắt đầu từ điểm có tọa độ (10,100) tính theo hệ tọa độ của Canvas chứa Path này.
- C: dùng để khai báo hai điểm điều khiển (Control Point) của đường cong. Sau hai Control Point là điểm kết thúc của đường cong, trong ví dụ trên, tọa độ điểm kết thúc là (300,100).
- Z: xác định đường kẻ ngang nối từ điểm kết thúc tới điểm khởi đầu của đường cong. Nếu muốn đặt đường kẻ này tới một điểm nào đó trên đường nằm ngang thì ta thay Z bằng H.

Graphic-2D

- Bezier (đường cong)

```
<Path
  Data="M 10,100 C 100,0 200,200 300,100 H 150"
  Stroke=■"Red"
  StrokeThickness="6"
  Fill=■"Pink"
  Canvas.Top="56.005" Canvas.Left="319" Height="145.995" Stretch="Fill"/>
```



- Lưu ý: Các thông số của thuộc tính Data có phân biệt chữ hoa, chữ thường.
 - Chữ hoa (M, C, Z/H) thì tọa độ điểm được tính theo vị trí tuyệt đối
 - Chữ thường (m, c, z/h) thì tọa độ được tính theo vị trí tương đối.

Brush

- Brush (chổi tô) có thể sử dụng để tô nền của một nút bấm (Button), tô các nét chữ (Text) hay tô màu bên trong cho một đối tượng hình học như hình chữ nhật, đa giác,...
- Các đối tượng áp dụng

Đối tượng	Thuộc tính tô vẽ
Border	BorderBrush, Background
Control	Background, Foreground
Panel	Background
Pen	Brush
Shape	Fill, Stroke
TextBlock	Background

Brush

- Các cách tô:
 - Tô màu đồng nhất (Solid Color)
 - Tô đổ màu theo tuyến tính (Linear Gradient Color)
 - Tô đổ màu dọc theo bán kính hình tròn (Radial Gradient Color)
 - Tô bằng phương pháp sử dụng ảnh bitmap.

Brush

- Các cách tô:
 - Tô màu đồng nhất (Solid Color)

```
<Rectangle Width="75" Height="75">  
  <Rectangle.Fill>  
    <SolidColorBrush Color=■ "Red" />  
  </Rectangle.Fill>  
</Rectangle>
```

```
<Rectangle Width="75" Height="75">  
  <Rectangle.Fill>  
    <SolidColorBrush Color=■ "#FF0000" />  
  </Rectangle.Fill>  
</Rectangle>
```

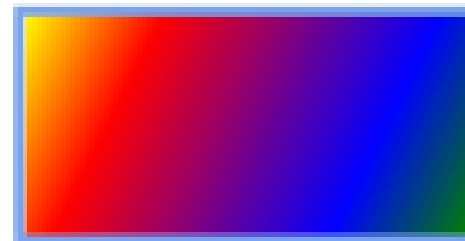
```
<Rectangle Width="75" Height="75" Fill=■ "#FF0000"/>
```

3 cách

Brush

- Các cách tô:
 - Tô đổ màu theo tuyến tính (Linear Gradient Color)

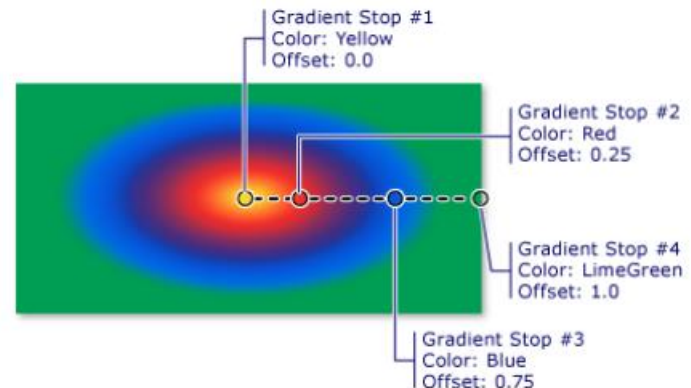
```
<Rectangle  
  Width="100"  
  Height="50"  
  Canvas.Left="10"      •  
  Canvas.Top="25" >  
  <Rectangle.Fill>  
    <LinearGradientBrush StartPoint="0,0" EndPoint="1,1">  
      <GradientStop Color=■ "Yellow" Offset="0.0" />  
      <GradientStop Color=■ "Red" Offset="0.25" />  
      <GradientStop Color=■ "Blue" Offset="0.75" />  
      <GradientStop Color=■ "Green" Offset="1.0" />  
    </LinearGradientBrush>  
  </Rectangle.Fill>  
</Rectangle>
```



Brush

- Các cách tô:
 - Tô đổ màu dọc theo bán kính hình tròn (Radial Gradient Color)

```
<Rectangle Width="200" Height="100">  
  <Rectangle.Fill>  
    <RadialGradientBrush GradientOrigin="0.5,0.5" Center="0.5,0.5" RadiusX="0.5" RadiusY="0.5">  
      <GradientStop Color="Yellow" Offset="0" />  
      <GradientStop Color="Red" Offset="0.25" />  
      <GradientStop Color="Blue" Offset="0.75" />  
      <GradientStop Color="LimeGreen" Offset="1" />  
    </RadialGradientBrush>  
  </Rectangle.Fill>  
</Rectangle>
```

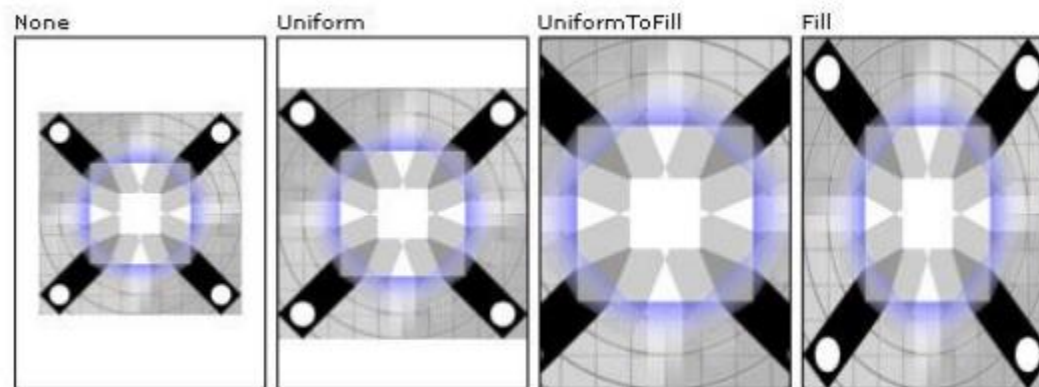


Brush

- Các cách tô:
 - Tô bằng phương pháp sử dụng ảnh bitmap
 - Sử dụng đối tượng ImageBrush để tô một vùng bằng ảnh có sẵn (Bitmap, JPG,...)
 - Thuộc tính ImageSource: chỉ định đường dẫn đến tệp hình ảnh.
 - Thuộc tính Stretch: chỉ định các co giãn hình khi tô vùng, giá trị mặc định của Stretch là Fill, gồm các giá trị:
 - None: không tự động co giãn hình.
 - Uniform: co giãn hình trùng khít với một chiều của vùng tô nhưng giữ nguyên tỷ lệ của ảnh gốc.
 - UniformToFill: co giãn hình phủ kín vùng tô nhưng giữ nguyên tỷ lệ của ảnh gốc.
 - Fill: co giãn hình phủ kín vùng tô, không giữ tỷ lệ ảnh.

Brush

- Các cách tô:
 - Tô bằng phương pháp sử dụng ảnh bitmap
 - Thuộc tính Stretch: chỉ định các co dẫn hình khi tô vùng, giá trị mặc định của Stretch là Fill, gồm các giá trị:

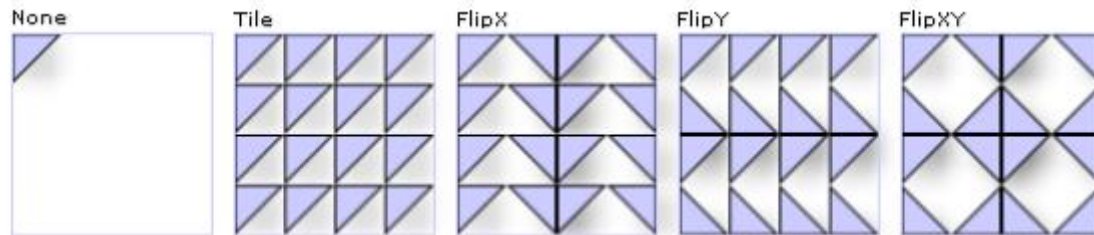


Brush

- Các cách tô:
 - Tô bằng phương pháp sử dụng ảnh bitmap
 - Thuộc tính TileMode chỉ định kiểu xếp lớp, gồm các giá trị sau:
 - None: Không xếp lớp.
 - Tile: Xếp lớp để phủ kín vùng cần tô.
 - FlipX: Xếp lớp để phủ kín vùng cần tô, hình được lật theo chiều ngang.
 - FlipY: Xếp lớp để phủ kín vùng cần tô, hình được lật theo chiều dọc.
 - FlipXY: Xếp lớp để phủ kín vùng cần tô, hình được lật theo cả hai chiều.

Brush

- Các cách tô:
 - Tô bằng phương pháp sử dụng ảnh bitmap
 - Thuộc tính TileMode chỉ định kiểu xếp lợp, gồm các giá trị sau:



Brush

- Các cách tô:
 - Tô bằng phương pháp sử dụng ảnh bitmap
 - Ví dụ

```
<Rectangle Canvas.Left="190" Canvas.Top="115" Height="150" Width="150" Stroke="■"Green" >  
    <Rectangle.Fill>  
        <ImageBrush ImageSource="Images\639479.jpg" TileMode="FlipXY"  
                    Viewport="0,0,25,25" ViewportUnits="Absolute" />  
    </Rectangle.Fill>  
</Rectangle>
```

