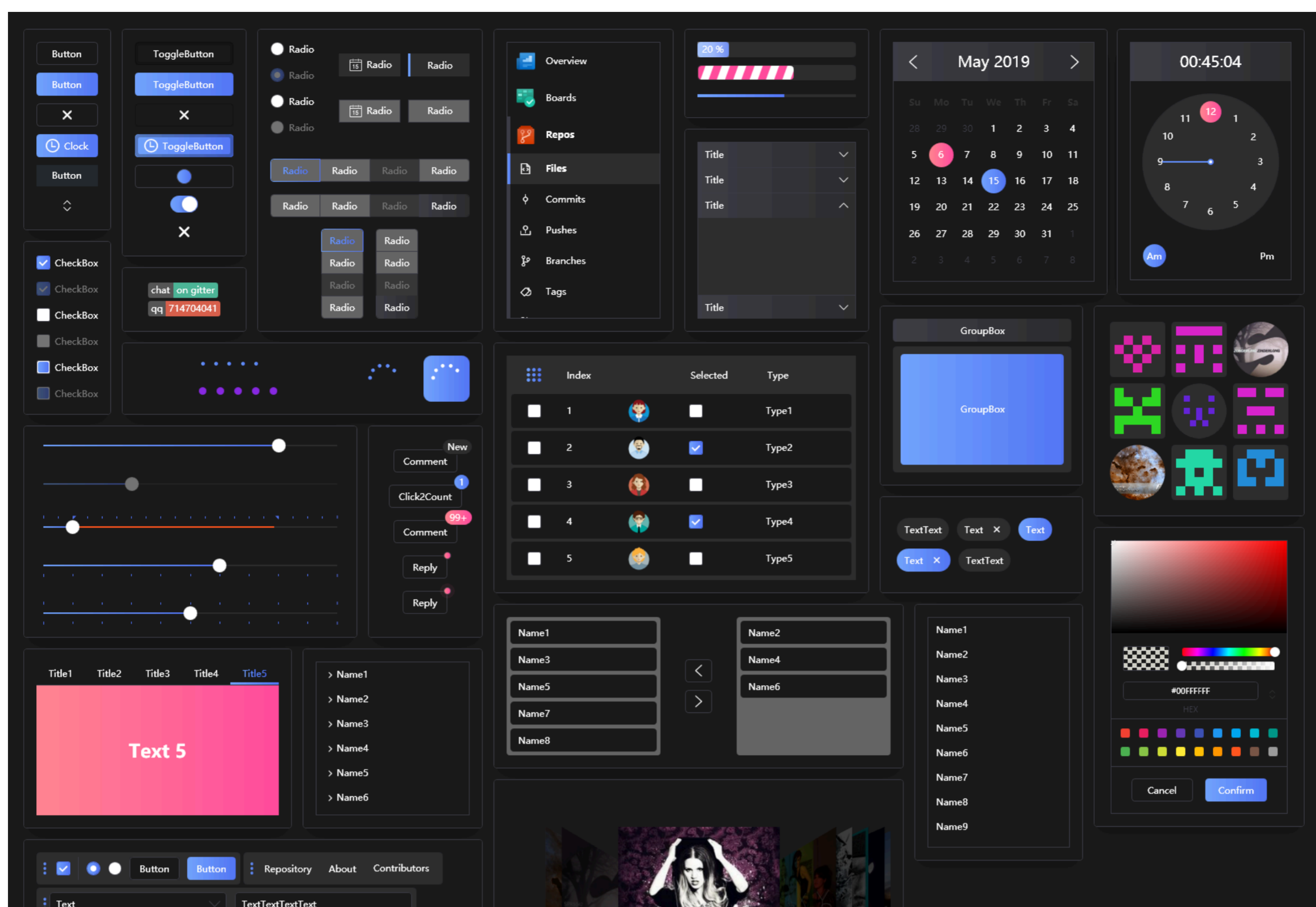


WPF绘图指南：用XAML轻松实现圆、线、矩形、文字、图片等创意元素

作者：微信公众号：【架构师老卢】

12-6 11:11

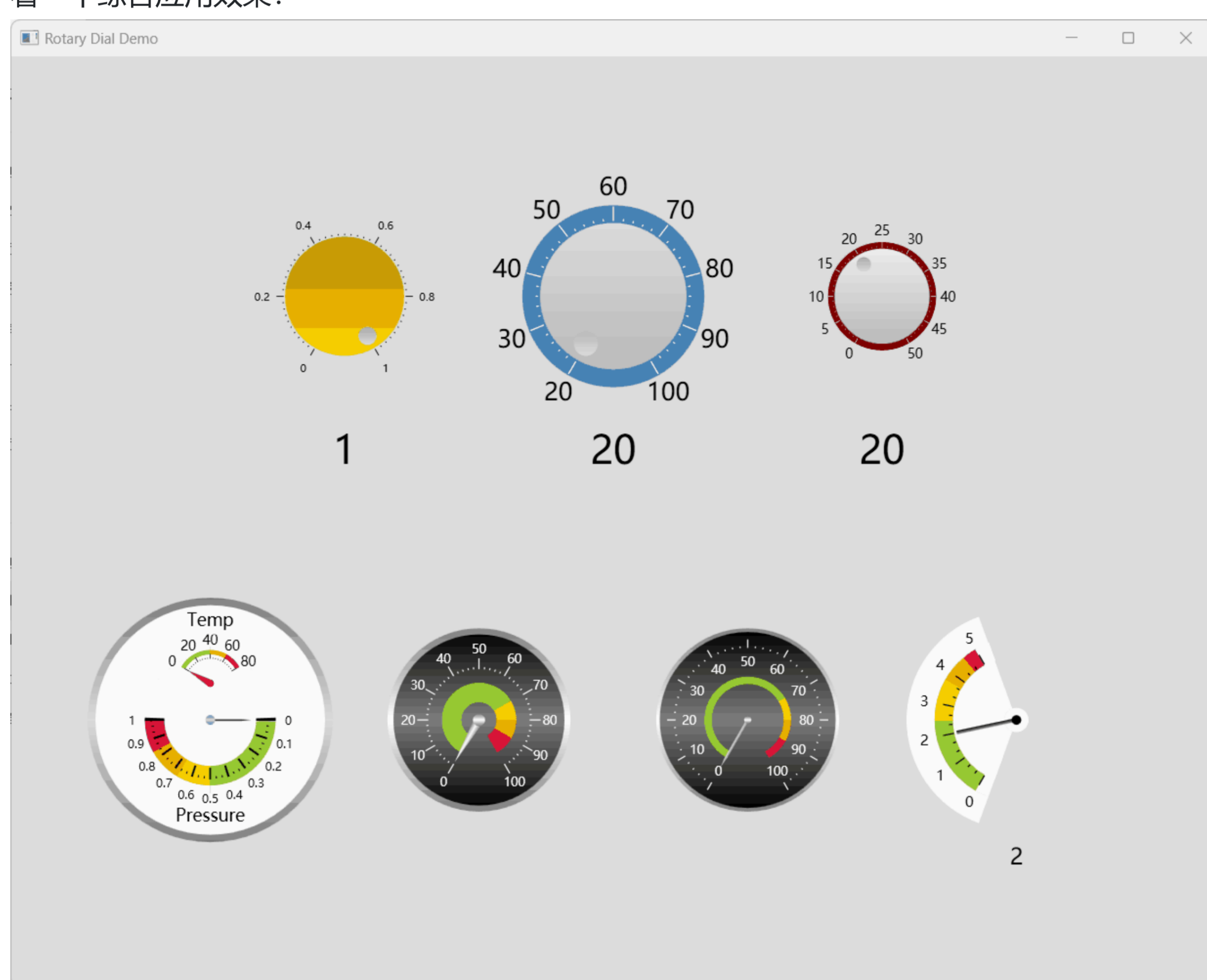
~ 158



概述：在WPF中，通过使用不同的元素如Ellipse、Line、Rectangle等，可以轻松绘制各种图形，包括圆、线条、椭圆、矩形、多边形等。同时，通过TextBlock展示文字，Image展示图片，以及Path创建路径和曲线，使得图形的绘制变得灵活多样。通过简单的XAML代码，开发者可以快速构建各种图形和界面元素

在WPF中，你可以使用Ellipse、Line、Rectangle、Polygon等元素来绘制基本的形状，同时可以使用TextBlock来显示文字，Image来展示图片，以及Path元素来创建复杂的路径和曲线。以下是详细的步骤和示例源代码：

看一个综合应用效果：



源代码文未提供

下面来看下基础绘制方法：

1. 画圆

```
1 <Canvas>
2   <Ellipse Width="50" Height="50" Fill="Blue"/>
3 </Canvas>
```

2. 画线条

```
1 <Canvas>
2   <Line X1="0" Y1="0" X2="100" Y2="100" Stroke="Black" StrokeThickness="2"/>
3 </Canvas>
```

3. 画椭圆

```
1 <Canvas>
2   <Ellipse Width="100" Height="50" Fill="Green"/>
3 </Canvas>
```

4. 画矩形

```
1 <Canvas>
2   <Rectangle Width="100" Height="50" Fill="Red"/>
3 </Canvas>
```

5. 画多边形

```
1 <Canvas>
2   <Polygon Points="0,0 50,0 50,50" Fill="Yellow"/>
3 </Canvas>
```

6. 画文字

```
1 <Canvas>
2   <TextBlock Text="Hello WPF!" FontSize="20" Foreground="Purple"/>
3 </Canvas>
```

7. 画图片

```
1 <Canvas>
2   <Image Source="your_image_path.jpg" Width="100" Height="100"/>
3 </Canvas>
```

8. 画路径

```
1 <Canvas>
2   <Path Stroke="Orange" StrokeThickness="2">
3     <Path.Data>
4       <PathGeometry>
5         <PathFigure StartPoint="0,0">
6           <LineSegment Point="50,50"/>
7           <LineSegment Point="100,0"/>
8         </PathFigure>
9       </PathGeometry>
10    </Path.Data>
11  </Path>
12 </Canvas>
```

9. 画曲线

```
1 <Canvas>
2   <Path Stroke="Brown" StrokeThickness="2">
3     <Path.Data>
4       <PathGeometry>
5         <PathFigure StartPoint="0,0">
6           <QuadraticBezierSegment Point1="50,100" Point2="100,0"/>
7         </PathFigure>
8       </PathGeometry>
9     </Path.Data>
10  </Path>
11 </Canvas>
```

10. 画贝塞尔曲线

```
1 <Canvas>
2   <Path Stroke="Gray" StrokeThickness="2">
3     <Path.Data>
4       <PathGeometry>
5         <PathFigure StartPoint="0,0">
6           <BezierSegment Point1="25,50" Point2="75,50" Point3="100,0"/>
7         </PathFigure>
8       </PathGeometry>
9     </Path.Data>
10  </Path>
11 </Canvas>
```

以上示例中，`Canvas`是一个用于布局子元素的容器，你可以根据需要选择不同的容器。在实际项目中，你可能需要使用更复杂的布局，例如`Grid`或`StackPanel`。

请注意，这只是基本示例，你可以根据需求修改属性以及组合不同的元素来创建更复杂的图形。实际场景里如果想实现动画效果一般需要辅助代码来实现。

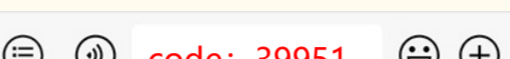
源代码获取：公众号回复消息【[code: 39951](#)】

相关代码下载地址



重要提示!：取消关注公众号后将无法再启用回复功能，不支持解封!

第一步: 微信扫码关注公众号“架构师老卢”

第二步: 在公众号聊天框发送 **code: 39951**，如： 获取下载地址

第三步: 恭喜你，快去下载你想要的资源吧