五專資工三甲

視窗程式設計

5BGGGGGH 維約凱 ZHUANG JUN KAI

題目

請依照上課範例,寫一個可以在視窗上畫圖的程式,功能如下:

- 有選單(Menu),工具列(ToolBar)和狀態列(StatusBar)
- 可以選擇要畫的圖形(Line, Rectangle, Ellipse)
- 可以選擇圖形的邊框顏色、填滿顏色以及邊框粗細。
- 可以清除畫布。

請將你的程式碼與程式執行畫面,輸出並彙整成一個 pdf 檔案來繳交, pdf 檔名請以學號+作業編號方式命名,列如第一個作業檔名為 A123456789ex1. pdf。請注意作業繳交要求、日期與格式。

```
1 using System;
 2 using System.Linq;
 3 using System.Windows;
 4 using System.Windows.Controls;
 5 using System.Windows.Input;
 6 using System.Windows.Media;
 7 using System.Windows.Shapes;
 9 namespace painterDraw
10 {
       /// <summary>
11
       /// Interaction logic for MainWindow.xaml
12
13
       /// </summary>
       public partial class MainWindow : Window
14
15
            string shapeType = "Line"; // 要有初始值
16
            int strokeThickness = 1;
17
18
           Color strokeColor = Colors.Red;
           Color fillColor = Colors.Yellow;
19
20
            Point start, dest;
21
            public MainWindow()
22
23
                InitializeComponent();
24
                strokeColorPicker.SelectedColor = strokeColor;
25
                fillColorPicker.SelectedColor = fillColor;
26
            }
27
28
            private void ShapeButton_Click(object sender, RoutedEventArgs
29
             e)
30
31
                var targetRadioButton = sender as RadioButton;
                shapeType = targetRadioButton.Tag.ToString();
32
33
34
                //MessageBox.Show(shapeType);
            }
35
36
            private void strokeThicknessSlider_ValueChanged(object sender,
37
              RoutedPropertyChangedEventArgs<double> e)
38
                strokeThickness = Convert.ToInt32
39
                  (strokeThicknessSlider.Value);
40
            }
41
            private void myCanvas_MouseMove(object sender, MouseEventArgs
42
             e)
            {
43
44
                dest = e.GetPosition(myCanvas);
                DisplayStatus();
45
                if(e.LeftButton == MouseButtonState.Pressed)
46
47
48
                    switch (shapeType)
49
```

```
...專三上\zhuang-c-sharp\painterDraw\MainWindow.xaml.cs
```

```
-
```

```
case "Line":
50
51
                            var line = myCanvas.Children.OfType<Line>
                      ().LastOrDefault();
                            line.X2 = dest.X;
52
53
                            line.Y2 = dest.Y;
54
                            break;
55
                        case "Rectangle":
56
                            Point origin2 = new Point
57
58
                                 X = Math.Min(start.X, dest.X),
59
                                 Y = Math.Min(start.Y, dest.Y)
60
                            };
61
                            var rect = myCanvas.Children.OfType<Rectangle>
62
                      ().LastOrDefault();
                            rect.SetValue(Canvas.LeftProperty, origin2.X);
63
                            rect.SetValue (Canvas.TopProperty, origin2.Y);
64
65
                            rect.Width = Math.Abs(dest.X - start.X);
                            rect.Height = Math.Abs(dest.Y - start.Y);
66
67
                            break:
                        case "Ellipse":
68
                            Point origin3 = new Point
69
70
                                 X = Math.Min(start.X, dest.X),
71
                                 Y = Math.Min(start.Y, dest.Y)
72
73
                            };
74
75
                            var ellipse = myCanvas.Children.OfType<Ellipse> >
                      ().LastOrDefault();
76
                            ellipse.SetValue
                      (Canvas.LeftProperty,origin3.X);
                            ellipse.SetValue(Canvas.TopProperty,
77
                      origin3.Y);
78
                            ellipse.Width = Math.Abs(dest.X - start.X);
79
                            ellipse.Height = Math.Abs(dest.Y - start.Y);
80
                            break;
                    }
81
82
                }
            }
83
84
            private void myCanvas_MouseLeftButtonDown(object sender,
85
              MouseButtonEventArgs e)
86
            {
87
                start = e.GetPosition(myCanvas);
                myCanvas.Cursor = Cursors.Cross;
88
89
                switch (shapeType)
90
                {
91
                    case "Line":
92
                        DrawLine(Colors.Gray, 1);
93
                        break;
94
                    case "Rectangle":
95
                        var rect = new Rectangle
96
                        {
```

```
...專三上\zhuang-c-sharp\painterDraw\MainWindow.xaml.cs
97
                             Stroke = Brushes.Gray,
98
                             StrokeThickness = 1,
99
                             Fill = Brushes.LightGray
100
                         };
101
                         myCanvas.Children.Add(rect);
102
                         rect.SetValue(Canvas.LeftProperty, start.X);
103
                         rect.SetValue(Canvas.TopProperty, start.Y);
104
                         break;
                     case "Ellipse":
105
106
                         var ellipse = new Ellipse
107
                         {
                             Stroke = Brushes.Gray,
108
109
                             StrokeThickness = 1,
                             Fill = Brushes.LightGray
110
111
                         };
                         myCanvas.Children.Add(ellipse);
112
                         ellipse.SetValue(Canvas.LeftProperty, start.X);
113
114
                         ellipse.SetValue (Canvas.TopProperty, start.Y);
115
                         break;
116
                 DisplayStatus();
117
             }
118
119
120
121
             private void DisplayStatus()
122
                 int LineCount = myCanvas.Children.OfType<Line>().Count();
123
124
                 int rectCount = myCanvas.Children.OfType<Rectangle>().Count >
                   ();
                 int ElliCount = myCanvas.Children.OfType<Ellipse>().Count
125
                   ():
                 coordinateLabel.Content = $"座標點:({Math.Round(start.X)},
126
                                                                               P
                   {Math.Round(start.Y)}) : ({Math.Round(dest.X)} ,
                   {Math.Round(dest.Y)})";
127
                 shapeLabel.Content = $"Line : {LineCount} Rectangle :
                   {rectCount} Ellipse : {ElliCount}";
             }
128
129
130
             private void strokeColorPicker_SelectedColorChanged(object
               sender, RoutedPropertyChangedEventArgs<Color?> e)
             {
131
132
                 strokeColor = (Color)strokeColorPicker.SelectedColor;
133
             }
134
135
             private void fillColorPicker_SelectedColorChanged(object
               sender, RoutedPropertyChangedEventArgs<Color?> e)
             {
136
137
                 fillColor = (Color)fillColorPicker.SelectedColor;
             }
138
139
             private void myCanvas_MouseLeftButtonUp(object sender,
140
               MouseButtonEventArgs e)
141
```

```
142
                 switch (shapeType)
143
                 {
144
                     case "Line":
145
                         var line = myCanvas.Children.OfType<Line>
                       ().LastOrDefault();
146
                         line.Stroke = new SolidColorBrush(strokeColor);
147
                         line.StrokeThickness = strokeThickness;
148
                         break;
149
                     case "Rectangle":
150
                         var rect = myCanvas.Children.OfType<Rectangle>
                                                                               P
                       ().LastOrDefault();
                         rect.Stroke = new SolidColorBrush(strokeColor);
151
152
                         rect.Fill = new SolidColorBrush(fillColor);
153
                         rect.StrokeThickness = strokeThickness;
154
                         break;
                     case "Ellipse":
155
156
                         var elli = myCanvas.Children.OfType<Ellipse>
                       ().LastOrDefault();
157
                         elli.Stroke = new SolidColorBrush(strokeColor);
158
                         elli.StrokeThickness = strokeThickness;
159
                         elli.Fill = new SolidColorBrush(fillColor);
160
                         break;
                 }
161
162
                 myCanvas.Cursor = Cursors.Arrow;
             }
163
164
             private void MenuItem_Click(object sender, RoutedEventArgs e)
165
166
167
                 myCanvas.Children.Clear();
             }
168
169
             private void DrawLine(Color color, int thickness)
170
171
                 Brush stroke = new SolidColorBrush(color);
172
173
                 Line line = new Line
174
                 {
175
                     Stroke = stroke,
176
                     X1 = start.X
177
                     Y1 = start.Y,
178
                     X2 = dest.X
                     Y2 = dest.Y
179
180
                 };
181
                 myCanvas.Children.Add(line);
182
             }
        }
183
184 }
185
```

...專三上\zhuang-c-sharp\painterDraw\MainWindow.xaml.cs

```
<Window x:Class="painterDraw.MainWindow"</pre>
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
        xmlns:mc="http://schemas.openxmlformats.org/markup-
          compatibility/2006"
        xmlns:xctk="http://schemas.xceed.com/wpf/xaml/toolkit"
        mc:Ignorable="d"
        Title="2023 wpf painter" Height="551" Width="800">
    <Grid>
        <DockPanel LastChildFill="False">
            <Menu Height="25" DockPanel.Dock="Top">
                <MenuItem Header="檔案">
                    <MenuItem Header="新增畫布" />
                </MenuItem>
                <MenuItem Header="工具">
                    <MenuItem Header="清除畫布" Click="MenuItem_Click"/>
                </MenuItem>
            </Menu>
            <ToolBarTray DockPanel.Dock="Top" Height="30">
                <ToolBar Width="489" Height="30"
                  HorizontalAlignment="Center">
                    <Label Content="筆刷色彩" />
                    <xctk:ColorPicker x:Name="strokeColorPicker" Width="100">
                    DisplayColorAndName="True"
                   SelectedColorChanged="strokeColorPicker_SelectedColorChan>
                   qed"/>
                    <Slider x:Name="strokeThicknessSlider" Width="100"</pre>
                                                                              P
                   VerticalAlignment="Center" Value="1"
                                                                              P
                   IsSnapToTickEnabled="True"
                   ValueChanged="strokeThicknessSlider_ValueChanged" />
                    <Label Content="{Binding Value,</pre>
                   ElementName=strokeThicknessSlider}" Width="30"
                   HorizontalContentAlignment="Center"/>
                    <Label Content="筆刷色彩" />
                    <xctk:ColorPicker x:Name="fillColorPicker" Width="100"</pre>
                   DisplayColorAndName="True"
                   SelectedColorChanged="fillColorPicker_SelectedColorChange">
                   d"/>
                </ToolBar>
                <ToolBar Width="220">
                    <RadioButton Content="直線" Tag="Line" IsChecked="True"
                   Click="ShapeButton_Click" Width="60"/>
                    <RadioButton Content="矩形" Tag="Rectangle"
                   Click="ShapeButton_Click" Width="60"/>
                    <RadioButton Content="橢圓" Tag="Ellipse"
                   Click="ShapeButton_Click" Width="60"/>
                </ToolBar>
            </ToolBarTray>
            <Canvas x:Name="myCanvas" DockPanel.Dock="Top" Height="400"</pre>
              Background= "#FFFFBEC"
              MouseLeftButtonDown="myCanvas_MouseLeftButtonDown"
              MouseMove="myCanvas_MouseMove"
```

輸出結果:

