```
1 using System;
 2 using System.Ling;
 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 _2023_WpfApp3
10 {
       /// <summary>
11
       /// Interaction logic for MainWindow.xaml
12
13
       /// </summary>
14
       public partial class MainWindow : Window
15
       {
16
           String shapeType = "Line";
           Color strokeColor = Colors.Red;
17
           Color fillColor = Colors.Yellow;
18
           int strokeThickness = 1;
19
20
21
           Point start, dest;
22
23
           public MainWindow()
24
25
               InitializeComponent();
               strokeColorPicker.SelectedColor = strokeColor;
26
27
               fillColorPicker.SelectedColor = fillColor;
           }
28
29
30
           private void ShapeButton_Click(object sender,
             RoutedEventArgs e)
31
               var targetRadioButton = sender as RadioButton;
32
               shapeType = targetRadioButton.Tag.ToString();
33
               //MessageBox.Show(shapeType);
34
35
           }
36
           private void strokeThicknessSlider_ValueChanged
37
             (object sender,
             RoutedPropertyChangedEventArgs<double> e)
38
           {
               strokeThickness = Convert.ToInt32
39
                  (strokeThicknessSlider.Value);
40
           }
41
42
           private void myCanvas_MouseMove(object sender,
             System.Windows.Input.MouseEventArgs e)
```

```
...ngyunn\C_SHAP_APP\WpfApp1\MainWindow.xaml.cs
                                                                     2
43
            {
44
                dest = e.GetPosition(myCanvas);
45
                DisplayStatus();
46
47
                if (e.LeftButton == MouseButtonState.Pressed)
48
                     Point origin = new Point
49
50
51
                         X = Math.Min(start.X, dest.X),
52
                         Y = Math.Min(start.Y, dest.Y)
53
                     double width = Math.Abs(dest.X - start.X);
54
55
                     double height = Math.Abs(dest.Y - start.Y);
56
57
                     switch (shapeType)
58
                     {
                         case "Line":
59
60
                             var line =
                    myCanvas.Children.OfType<Line>().LastOrDefault→
                    ();
61
                             line.X2 = dest.X;
62
                             line.Y2 = dest.Y;
63
                             break;
                         case "Rectangle":
64
65
                             var rect =
                    myCanvas.Children.OfType<Rectangle>
                    ().LastOrDefault();
                             rect.Width = width;
66
                             rect.Height = height;
67
68
                             rect.SetValue(Canvas.LeftProperty,
                    origin.X);
69
                             rect.SetValue(Canvas.TopProperty,
                    origin.Y);
70
                             break;
71
                         case "Ellipse":
72
                             var ellipse =
                    myCanvas.Children.OfType<Ellipse>
                    ().LastOrDefault();
                             ellipse.Width = width;
73
                             ellipse.Height = height;
74
75
                             ellipse.SetValue(Canvas.LeftProperty, >
                     origin.X);
76
                             ellipse.SetValue(Canvas.TopProperty, >
                    origin.Y);
77
                             break;
```

}

}

78

79

```
...ngyunn\C_SHAP_APP\WpfApp1\MainWindow.xaml.cs
                                                                      3
 80
             }
 81
             private void myCanvas_MouseLeftButtonDown(object
 82
               sender, System.Windows.Input.MouseButtonEventArgs
               e)
             {
 83
                 start = e.GetPosition(myCanvas);
 84
 85
                 myCanvas.Cursor = Cursors.Cross;
 86
 87
                 switch (shapeType)
 88
                     case "Line":
 89
 90
                         var line = new Line
 91
 92
                              Stroke = Brushes.Gray,
 93
                              StrokeThickness = 1,
                             X1 = start.X
 94
 95
                              Y1 = start.Y,
                             X2 = dest.X
 96
                             Y2 = dest.Y
 97
 98
                         };
 99
                         myCanvas.Children.Add(line);
100
                         break;
                     case "Rectangle":
101
102
                         var rect = new Rectangle
                         {
103
                              Stroke = Brushes.Gray,
104
                              StrokeThickness = 1,
105
```

```
Fill = Brushes.LightGray,
106
107
                         myCanvas.Children.Add(rect);
108
                         rect.SetValue(Canvas.LeftProperty,
109
                         rect.SetValue(Canvas.TopProperty,
110
                     start.Y);
111
                         break;
                     case "Ellipse":
112
113
                         var ellipse = new Ellipse
114
115
                             Stroke = Brushes.Gray,
                             StrokeThickness = 1,
116
                             Fill = Brushes.LightGray,
117
                         };
118
                         myCanvas.Children.Add(ellipse);
119
                         ellipse.SetValue(Canvas.LeftProperty,
120
                    start.X);
121
                         ellipse.SetValue(Canvas.TopProperty,
```

```
...ngyunn\C_SHAP_APP\WpfApp1\MainWindow.xaml.cs
```

```
L
```

```
start.Y);
122
                         break;
123
                 DisplayStatus();
124
            }
125
126
            private void DisplayStatus()
127
128
129
                 int lineCount = myCanvas.Children.OfType<Line>
                   ().Count();
                 int rectCount =
130
                   myCanvas.Children.OfType<Rectangle>().Count();
131
                 int ellipseCount =
                                                                     P
                   myCanvas.Children.OfType<Ellipse>().Count();
132
                 coordinateLabel.Content = $"座標點: ({Math.Round
                   (start.X)}, {Math.Round(start.Y)}) :
                   ({Math.Round(dest.X)}, {Math.Round(dest.Y)})";
                 shapeLabel.Content = $"Line: {lineCount},
133
                                                                     P
                   Rectangle: {rectCount}, Ellipse:
                                                                     P
                   {ellipseCount}";
            }
134
135
136
             private void strokeColorPicker_SelectedColorChanged
                                                                     P
              (object sender,
                                                                     P
              RoutedPropertyChangedEventArgs<Color?> e)
137
             {
                 strokeColor = (Color)
138
                                                                     7
                   strokeColorPicker.SelectedColor;
            }
139
140
             private void fillColorPicker_SelectedColorChanged
141
              (object sender,
              RoutedPropertyChangedEventArgs<Color?> e)
142
             {
143
                 fillColor = (Color)fillColorPicker.SelectedColor;
             }
144
145
             private void clearMenuItem_Click(object sender,
146
              RoutedEventArgs e)
147
             {
                 myCanvas.Children.Clear();
148
                 DisplayStatus();
149
            }
150
151
             private void myCanvas_MouseLeftButtonUp(object
152
              sender, MouseButtonEventArgs e)
153
             {
```

```
...ngyunn\C_SHAP_APP\WpfApp1\MainWindow.xaml.cs
                                                                     5
154
                 switch (shapeType)
155
                 {
156
                     case "Line":
157
                         var line = myCanvas.Children.OfType<Line> >
                    ().LastOrDefault();
                         line.Stroke = new SolidColorBrush
158
                    (strokeColor);
159
                         line.StrokeThickness = strokeThickness;
160
                         break;
                     case "Rectangle":
161
                         var rect =
162
                    myCanvas.Children.OfType<Rectangle>
                    ().LastOrDefault();
163
                         rect.Stroke = new SolidColorBrush
                    (strokeColor);
                         rect.Fill = new SolidColorBrush
164
                    (fillColor);
                         rect.StrokeThickness = strokeThickness;
165
166
                         break;
                     case "Ellipse":
167
                         var ellipse =
168
                    myCanvas.Children.OfType<Ellipse>
                    ().LastOrDefault();
169
                         ellipse.Stroke = new SolidColorBrush
                    (strokeColor);
170
                         ellipse.Fill = new SolidColorBrush
                    (fillColor);
171
                         ellipse.StrokeThickness =
                    strokeThickness;
172
                         break;
                 }
173
174
                 myCanvas.Cursor = Cursors.Arrow;
175
            }
176
        }
177 }
```