```
 1  using Microsoft.Win32;
 2  using System;
 3  using System.IO;
 4  using System.Linq;
 5  using System.Windows;
 6  using System.Windows.Controls;
 7  using System.Windows.Input;
 8  using System.Windows.Media;
 9  using System.Windows.Media.Imaging;
10  using System.Windows.Shapes;
11
12  namespace painter
13  {
14      /// <summary>
15      /// Interaction logic for MainWindow.xaml
16      /// </summary>
17      public partial class MainWindow : Window
18      {
19          string shapeType = "Line"; // 要有初始值
20          string actionType = "Draw";
21          int strokeThickness = 1;
22          Color strokeColor = Colors.Red;
23          Color fillColor = Colors.Yellow;
24
25          Point start, dest;
26          public MainWindow()
27          {
28              InitializeComponent();
29              strokeColorPicker.SelectedColor = strokeColor;
30              fillColorPicker.SelectedColor = fillColor;
31          }
32
33          private void ShapeButton_Click(object sender, RoutedEventArgs e)
34          {
35              var targetRadioButton = sender as RadioButton;
36              shapeType = targetRadioButton.Tag.ToString();
37              actionType = "Draw";
38              //MessageBox.Show(shapeType);
39          }
40
41          private void strokeThicknessSlider_ValueChanged(object sender,
              RoutedPropertyChangedEventArgs<double> e)
42          {
43              strokeThickness = Convert.ToInt32
                (strokeThicknessSlider.Value);
44          }
45
46          private void myCanvas_MouseMove(object sender, MouseEventArgs e)
47          {
```

```
48              dest = e.GetPosition(myCanvas);
49              DisplayStatus();
50
51              switch (actionType)
52              {
53                  case "Draw":
54                      if (e.LeftButton == MouseButtonState.Pressed)
55                      {
56                          switch (shapeType)
57                          {
58                              case "Line":
59                                  var line = myCanvas.Children.OfType<Line>
                    ().LastOrDefault();
60                                  line.X2 = dest.X;
61                                  line.Y2 = dest.Y;
62                                  break;
63                              case "Rectangle":
64                                  Point origin2 = new Point
65                                  {
66                                      X = Math.Min(start.X, dest.X),
67                                      Y = Math.Min(start.Y, dest.Y)
68                                  };
69
70                                  var rect =
                    myCanvas.Children.OfType<Rectangle>().LastOrDefault();
71                                  rect.SetValue(Canvas.LeftProperty,
                    origin2.X);
72                                  rect.SetValue(Canvas.TopProperty,
                    origin2.Y);
73                                  rect.Width = Math.Abs(dest.X - start.X);
74                                  rect.Height = Math.Abs(dest.Y - start.Y);
75                                  break;
76                              case "Ellipse":
77                                  Point origin3 = new Point
78                                  {
79                                      X = Math.Min(start.X, dest.X),
80                                      Y = Math.Min(start.Y, dest.Y)
81                                  };
82
83                                  var ellipse =
                    myCanvas.Children.OfType<Ellipse>().LastOrDefault();
84                                  ellipse.SetValue(Canvas.LeftProperty,
                    origin3.X);
85                                  ellipse.SetValue(Canvas.TopProperty,
                    origin3.Y);
86                                  ellipse.Width = Math.Abs(dest.X -
                    start.X);
87                                  ellipse.Height = Math.Abs(dest.Y -
                    start.Y);
```

```
88                                  break;
89                              case "Polyline":
90                                  var polyline =
                        myCanvas.Children.OfType<Polyline>().LastOrDefault();
91                                  polyline.Points.Add(dest);
92                                  break;
93                          }
94                      }
95                      break;
96                  case "Erase":
97                      var shape = e.OriginalSource as Shape;
98                      myCanvas.Children.Remove(shape);
99                      if (myCanvas.Children.Count == 0) myCanvas.Cursor =
                        Cursors.Arrow;
100                     break;
101             }
102
103
104         }
105
106         private void myCanvas_MouseLeftButtonDown(object sender,
              MouseButtonEventArgs e)
107         {
108             start = e.GetPosition(myCanvas);
109             myCanvas.Cursor = Cursors.Cross;
110
111             if(actionType == "Draw")
112             {
113                 switch (shapeType)
114                 {
115                     case "Line":
116                         DrawLine(Colors.Gray, 1);
117                         break;
118                     case "Rectangle":
119                         var rect = new Rectangle
120                         {
121                             Stroke = Brushes.Gray,
122                             StrokeThickness = 1,
123                             Fill = Brushes.LightGray
124                         };
125                         myCanvas.Children.Add(rect);
126                         rect.SetValue(Canvas.LeftProperty, start.X);
127                         rect.SetValue(Canvas.TopProperty, start.Y);
128                         break;
129                     case "Ellipse":
130                         var ellipse = new Ellipse
131                         {
132                             Stroke = Brushes.Gray,
133                             StrokeThickness = 1,
```

```
134                             Fill = Brushes.LightGray
135                         };
136                         myCanvas.Children.Add(ellipse);
137                         ellipse.SetValue(Canvas.LeftProperty, start.X);
138                         ellipse.SetValue(Canvas.TopProperty, start.Y);
139                         break;
140                     case "Polyline":
141                         var polyline = new Polyline
142                         {
143                             Stroke = Brushes.Gray,
144                             StrokeThickness = 1,
145                             Fill = Brushes.LightGray
146                         };
147                         myCanvas.Children.Add(polyline);
148                         break;
149                 }
150             }
151             DisplayStatus();
152         }
153 
154         private void DisplayStatus()
155         {
156             int LineCount = myCanvas.Children.OfType<Line>().Count();
157             int rectCount = myCanvas.Children.OfType<Rectangle>().Count();
158             int ElliCount = myCanvas.Children.OfType<Ellipse>().Count();
159             int polylineCount = myCanvas.Children.OfType<Polyline>().Count
                ();
160             coordinateLabel.Content = $"座標點:({Math.Round(start.X)} ,
                {Math.Round(start.Y)}) : ({Math.Round(dest.X)} , {Math.Round
                (dest.Y)})";
161             shapeLabel.Content = $"Line : {LineCount} Rectangle :
                {rectCount} Ellipse : {ElliCount}, Polyline:
                {polylineCount}";
162         }
163 
164         private void strokeColorPicker_SelectedColorChanged(object sender,
               RoutedPropertyChangedEventArgs<Color?> e)
165         {
166             strokeColor = (Color)strokeColorPicker.SelectedColor;
167         }
168 
169         private void fillColorPicker_SelectedColorChanged(object sender,
              RoutedPropertyChangedEventArgs<Color?> e)
170         {
171             fillColor = (Color)fillColorPicker.SelectedColor;
172         }
173 
174         private void myCanvas_MouseLeftButtonUp(object sender,
              MouseButtonEventArgs e)
```

```
175         {
176             if (actionType == "Draw")
177             {
178                 switch (shapeType)
179                 {
180                     case "Line":
181                         var line = myCanvas.Children.OfType<Line>
                        ().LastOrDefault();
182                         line.Stroke = new SolidColorBrush(strokeColor);
183                         line.StrokeThickness = strokeThickness;
184                         break;
185                     case "Rectangle":
186                         var rect = myCanvas.Children.OfType<Rectangle>
                        ().LastOrDefault();
187                         rect.Stroke = new SolidColorBrush(strokeColor);
188                         rect.Fill = new SolidColorBrush(fillColor);
189                         rect.StrokeThickness = strokeThickness;
190                         break;
191                     case "Ellipse":
192                         var elli = myCanvas.Children.OfType<Ellipse>
                        ().LastOrDefault();
193                         elli.Stroke = new SolidColorBrush(strokeColor);
194                         elli.StrokeThickness = strokeThickness;
195                         elli.Fill = new SolidColorBrush(fillColor);
196                         break;
197                     case "Polyline":
198                         var poly = myCanvas.Children.OfType<Polyline>
                        ().LastOrDefault();
199                         poly.Stroke = new SolidColorBrush(strokeColor);
200                         poly.Fill = new SolidColorBrush(fillColor);
201                         poly.StrokeThickness = strokeThickness;
202                         break;
203                 }
204                 myCanvas.Cursor = Cursors.Arrow;
205             }
206         }
207 
208         private void clear_canva(object sender, RoutedEventArgs e)
209         {
210             myCanvas.Children.Clear();
211         }
212 
213         private void eraseButton_Click(object sender, RoutedEventArgs e)
214         {
215             actionType = "Erase";
216             myCanvas.Cursor = Cursors.Hand;
217             DisplayStatus();
218         }
219 
```

```
220         private void saveCanvas(object sender, RoutedEventArgs e)
221         {
222             SaveFileDialog saveFileDialog = new SaveFileDialog();
223             saveFileDialog.Title = "儲存畫布";
224             saveFileDialog.Filter = "Png檔案|*.png|所有檔案|*.*";
225 
226             if (saveFileDialog.ShowDialog() == true)
227             {
228                 // Create a RenderTargetBitmap to capture the canvas
                      content
229                 RenderTargetBitmap renderBitmap = new RenderTargetBitmap(
230                     (int)myCanvas.ActualWidth,
231                     (int)myCanvas.ActualHeight,
232                     64d, 64d, PixelFormats.Default);
233 
234                 // Render the canvas to the RenderTargetBitmap
235                 renderBitmap.Render(myCanvas);
236 
237                 // Create a BitmapEncoder (e.g., PNGEncoder) to save the
                      image
238                 PngBitmapEncoder encoder = new PngBitmapEncoder();
239                 encoder.Frames.Add(BitmapFrame.Create(renderBitmap));
240 
241                 // Create a file stream using the user-selected file name
242                 string fileName = saveFileDialog.FileName;
243                 using (FileStream fs = new FileStream(fileName,
                      FileMode.Create))
244                 {
245                     encoder.Save(fs);
246                 }
247 
248                 //MessageBox.Show($"Canvas content saved as {fileName}");
249             }
250         }
251 
252         private void DrawLine(Color color, int thickness)
253         {
254             Brush stroke = new SolidColorBrush(color);
255             Line line = new Line
256             {
257                 Stroke = stroke,
258                 X1 = start.X,
259                 Y1 = start.Y,
260                 X2 = dest.X,
261                 Y2 = dest.Y
262             };
263             myCanvas.Children.Add(line);
264         }
265     }
```

```
266  }
267
```