```
using System;
using System.Linq;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Shapes;

namespace _2023_WpfApp3
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        String shapeType = "Line";
        Color strokeColor = Colors.Red;
        Color fillColor = Colors.Yellow;
        int strokeThickness = 1;

        Point start, dest;

        public MainWindow()
        {
            InitializeComponent();
            strokeColorPicker.SelectedColor = strokeColor;
            fillColorPicker.SelectedColor = fillColor;
        }

        private void ShapeButton_Click(object sender,
          RoutedEventArgs e)
        {
            var targetRadioButton = sender as RadioButton;
            shapeType = targetRadioButton.Tag.ToString();
            //MessageBox.Show(shapeType);
        }

        private void strokeThicknessSlider_ValueChanged
          (object sender,
          RoutedPropertyChangedEventArgs<double> e)
        {
            strokeThickness = Convert.ToInt32
              (strokeThicknessSlider.Value);
        }

        private void myCanvas_MouseMove(object sender,
          System.Windows.Input.MouseEventArgs e)
        {
            dest = e.GetPosition(myCanvas);
            DisplayStatus();

            if (e.LeftButton == MouseButtonState.Pressed)
            {
                Point origin = new Point
                {
                    X = Math.Min(start.X, dest.X),
                    Y = Math.Min(start.Y, dest.Y)
                };
                double width = Math.Abs(dest.X - start.X);
                double height = Math.Abs(dest.Y - start.Y);

                switch (shapeType)
                {
                    case "Line":
                        var line = myCanvas.Children.OfType<Line>().LastOrDefault();
                        line.X2 = dest.X;
                        line.Y2 = dest.Y;
                        break;
                    case "Rectangle":
                        var rect = myCanvas.Children.OfType<Rectangle>().LastOrDefault();
                        rect.Width = width;
                        rect.Height = height;
                        rect.SetValue(Canvas.LeftProperty, origin.X);
                        rect.SetValue(Canvas.TopProperty, origin.Y);
                        break;
                    case "Ellipse":
                        var ellipse = myCanvas.Children.OfType<Ellipse>().LastOrDefault();
                        ellipse.Width = width;
                        ellipse.Height = height;
                        ellipse.SetValue(Canvas.LeftProperty, origin.X);
                        ellipse.SetValue(Canvas.TopProperty, origin.Y);
                        break;
                }
            }
        }

        private void myCanvas_MouseLeftButtonDown(object sender, System.Windows.Input.MouseButtonEventArgs e)
        {
            start = e.GetPosition(myCanvas);
            myCanvas.Cursor = Cursors.Cross;

            switch (shapeType)
            {
                case "Line":
                    var line = new Line
                    {
                        Stroke = Brushes.Gray,
                        StrokeThickness = 1,
                        X1 = start.X,
                        Y1 = start.Y,
                        X2 = dest.X,
                        Y2 = dest.Y
                    };
                    myCanvas.Children.Add(line);
                    break;
                case "Rectangle":
                    var rect = new Rectangle
                    {
                        Stroke = Brushes.Gray,
                        StrokeThickness = 1,
                        Fill = Brushes.LightGray,
                    };
                    myCanvas.Children.Add(rect);
                    rect.SetValue(Canvas.LeftProperty, start.X);
                    rect.SetValue(Canvas.TopProperty, start.Y);
                    break;
                case "Ellipse":
                    var ellipse = new Ellipse
                    {
                        Stroke = Brushes.Gray,
                        StrokeThickness = 1,
                        Fill = Brushes.LightGray,
                    };
                    myCanvas.Children.Add(ellipse);
                    ellipse.SetValue(Canvas.LeftProperty, start.X);
                    ellipse.SetValue(Canvas.TopProperty,
                start.Y);
                    break;
            }
            DisplayStatus();
        }

        private void DisplayStatus()
        {
            int lineCount = myCanvas.Children.OfType<Line>().Count();
            int rectCount = myCanvas.Children.OfType<Rectangle>().Count();
            int ellipseCount = myCanvas.Children.OfType<Ellipse>().Count();
            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}, Rectangle: {rectCount}, Ellipse: {ellipseCount}";
        }

        private void strokeColorPicker_SelectedColorChanged(object sender, RoutedPropertyChangedEventArgs<Color?> e)
        {
            strokeColor = (Color)strokeColorPicker.SelectedColor;
        }

        private void fillColorPicker_SelectedColorChanged(object sender, RoutedPropertyChangedEventArgs<Color?> e)
        {
            fillColor = (Color)fillColorPicker.SelectedColor;
        }

        private void clearMenuItem_Click(object sender, RoutedEventArgs e)
        {
            myCanvas.Children.Clear();
            DisplayStatus();
        }

        private void myCanvas_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
        {
            switch (shapeType)
            {
                case "Line":
                    var line = myCanvas.Children.OfType<Line>().LastOrDefault();
                    line.Stroke = new SolidColorBrush(strokeColor);
                    line.StrokeThickness = strokeThickness;
                    break;
                case "Rectangle":
                    var rect = myCanvas.Children.OfType<Rectangle>().LastOrDefault();
                    rect.Stroke = new SolidColorBrush(strokeColor);
                    rect.Fill = new SolidColorBrush(fillColor);
                    rect.StrokeThickness = strokeThickness;
                    break;
                case "Ellipse":
                    var ellipse = myCanvas.Children.OfType<Ellipse>().LastOrDefault();
                    ellipse.Stroke = new SolidColorBrush(strokeColor);
                    ellipse.Fill = new SolidColorBrush(fillColor);
                    ellipse.StrokeThickness = strokeThickness;
                    break;
            }
            myCanvas.Cursor = Cursors.Arrow;
        }
    }
}
```