五專資工三甲

視窗程式設計

5B0G0024

莊鈞凱

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。

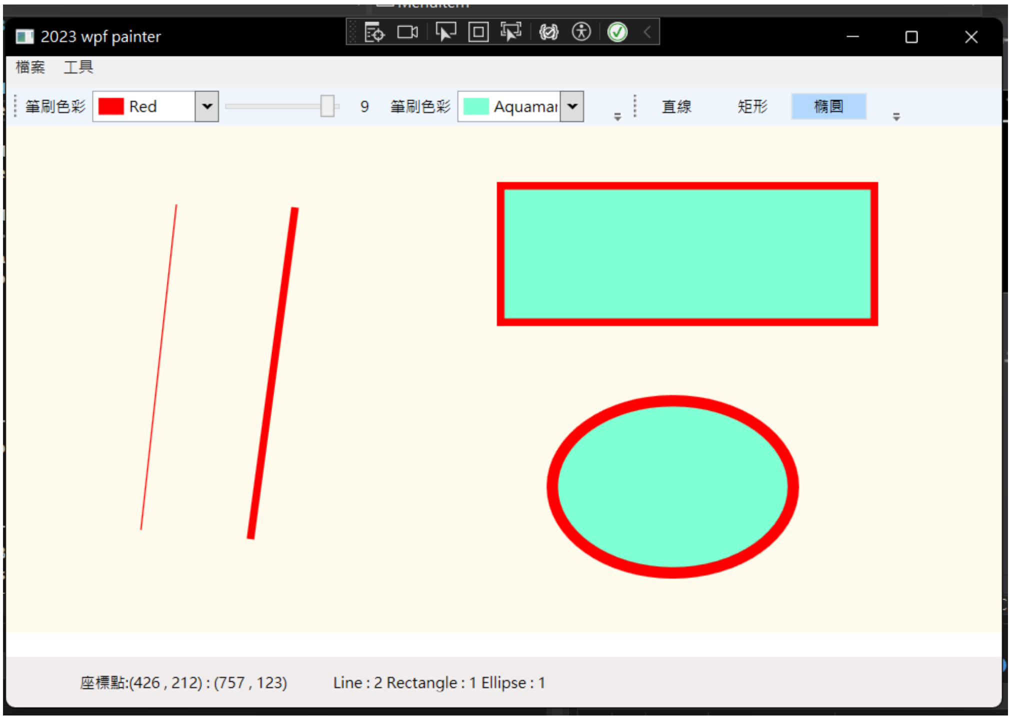
請將你的程式碼與程式執行畫面，輸出並彙整成一個pdf檔案來繳交，pdf檔名請以學號+作業編號方式命名，列如第一個作業檔名為A123456789ex1.pdf。請注意作業繳交要求、日期與格式。

```
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 painterDraw
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        string shapeType = "Line"; // 要有初始值
        int strokeThickness = 1;
        Color strokeColor = Colors.Red;
        Color fillColor = Colors.Yellow;

        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, MouseEventArgs e)
        {
            dest = e.GetPosition(myCanvas);
            DisplayStatus();
            if(e.LeftButton == MouseButtonState.Pressed)
            {
                switch (shapeType)
                {
                    case "Line":
                        var line = myCanvas.Children.OfType<Line>().LastOrDefault();
                        line.X2 = dest.X;
                        line.Y2 = dest.Y;
                        break;
                    case "Rectangle":
                        Point origin2 = new Point
                        {
                            X = Math.Min(start.X, dest.X),
                            Y = Math.Min(start.Y, dest.Y)
                        };

                        var rect = myCanvas.Children.OfType<Rectangle>().LastOrDefault();
                        rect.SetValue(Canvas.LeftProperty, origin2.X);
                        rect.SetValue (Canvas.TopProperty, origin2.Y);
                        rect.Width = Math.Abs(dest.X - start.X);
                        rect.Height = Math.Abs(dest.Y - start.Y);
                        break;
                    case "Ellipse":
                        Point origin3 = new Point
                        {
                            X = Math.Min(start.X, dest.X),
                            Y = Math.Min(start.Y, dest.Y)
                        };

                        var ellipse = myCanvas.Children.OfType<Ellipse>().LastOrDefault();
                        ellipse.SetValue(Canvas.LeftProperty,origin3.X);
                        ellipse.SetValue(Canvas.TopProperty, origin3.Y);
                        ellipse.Width = Math.Abs(dest.X - start.X);
                        ellipse.Height = Math.Abs(dest.Y - start.Y);
                        break;
                }
            }
        }

        private void myCanvas_MouseLeftButtonDown(object sender, MouseButtonEventArgs e)
        {
            start = e.GetPosition(myCanvas);
            myCanvas.Cursor = Cursors.Cross;
            switch (shapeType)
            {
                case "Line":
                    DrawLine(Colors.Gray, 1);
                    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 ElliCount = 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 : {ElliCount}";
        }

        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 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 elli = myCanvas.Children.OfType<Ellipse>().LastOrDefault();
                    elli.Stroke = new SolidColorBrush(strokeColor);
                    elli.StrokeThickness = strokeThickness;
                    elli.Fill = new SolidColorBrush(fillColor);
                    break;
            }
            myCanvas.Cursor = Cursors.Arrow;
        }

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

        private void DrawLine(Color color, int thickness)
        {
            Brush stroke = new SolidColorBrush(color);
            Line line = new Line
            {
                Stroke = stroke,
                X1 = start.X,
                Y1 = start.Y,
                X2 = dest.X,
                Y2 = dest.Y
            };
            myCanvas.Children.Add(line);
        }
    }
}
```

```
<Window x:Class="painterDraw.MainWindow"
    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
         ged"/>
          <Slider x:Name="strokeThicknessSlider" Width="100"
         VerticalAlignment="Center" Value="1"
         IsSnapToTickEnabled="True"
         ValueChanged="strokeThicknessSlider_ValueChanged" />
          <Label Content="{Binding Value,
         ElementName=strokeThicknessSlider}" Width="30"
         HorizontalContentAlignment="Center"/>
          <Label Content="筆刷色彩" />
          <xctk:ColorPicker x:Name="fillColorPicker" Width="100"
         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"
       Background=□"#FFFFFBEC"
       MouseLeftButtonDown="myCanvas_MouseLeftButtonDown"
       MouseMove="myCanvas_MouseMove"
```

```xml
              MouseLeftButtonUp="myCanvas_MouseLeftButtonUp"/>
            <StatusBar DockPanel.Dock="Bottom" Height="40">
                <StackPanel Orientation="Horizontal" Margin="50,0,0,0">
                    <Label x:Name="coordinateLabel" Content="座標"
                   Width="200" Height="25"></Label>
                    <Label x:Name="shapeLabel"  Width="250" Height="25"/>
                </StackPanel>
            </StatusBar>
        </DockPanel>

    </Grid>
</Window>
```

輸出結果：



筆刷色彩 Red 9 筆刷色彩 Aquamar

座標點:(426 , 212) : (757 , 123)   Line : 2 Rectangle : 1 Ellipse : 1