Tutorial – Drawing and Mouse events

(Windows Store App)

Chih-Yu Kang
Ver. 1.0, 7th November, 2012
Bo Jun Chen
Ver. 3.0 24th November, 2016
Jyun Wei Chen
Ver. 4.0 21th November, 2022

Contents

About This Document

在本次的練習中，將會練習到如何在 Windows Store App 中使用 Canvas 繪出矩形和圓形，以及如何在 Canvas 中註冊點擊及移動的事件。

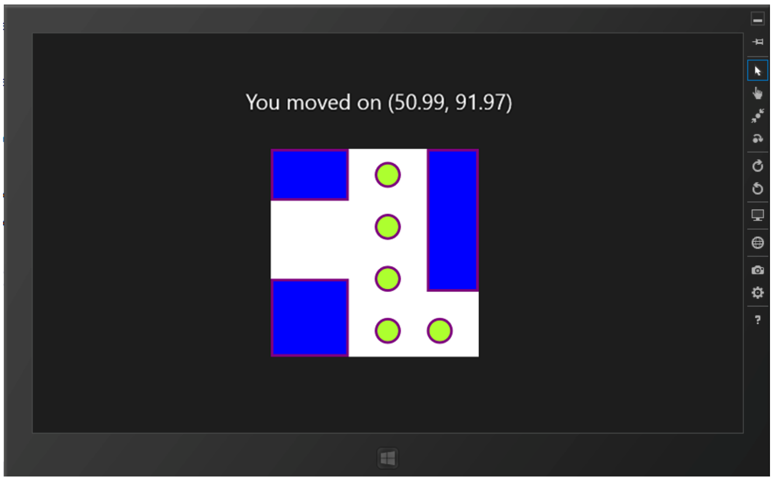

You moved on (50.99, 91.97)

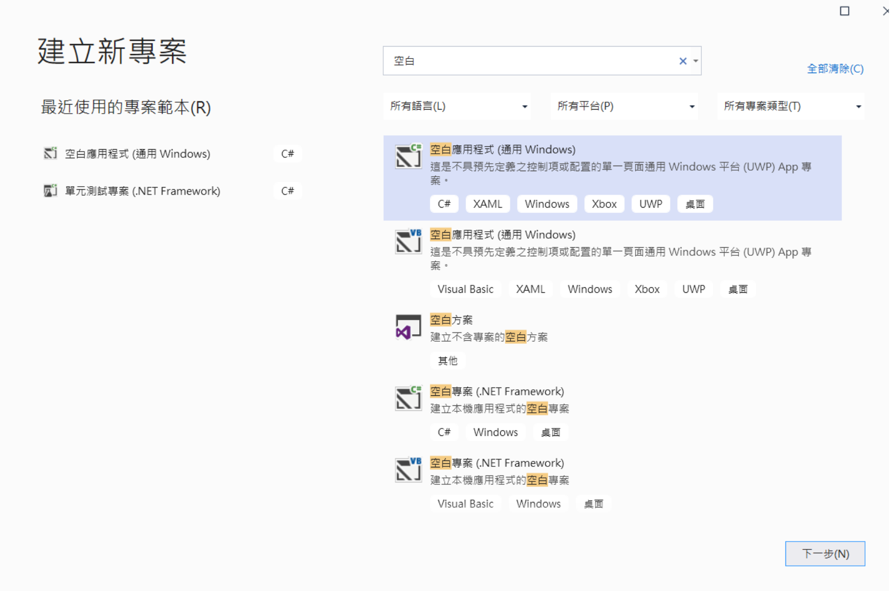
Figure 1 程式執行結果

Tutorial – Drawing and Mouse Events

Step 1 Setup Environment

打開 Visual Studio 2019 Community 後，建立新專案 > 選擇空白應用程式(通

用 Windows)，如 Figure 2，然後輸入專案名稱，如 Figure 3，在 [新增通用 Windows 平台專案] 對話方塊中，接受預設的目標版本和最低版本設定，如 Figure 4。即可成功建立一個空白的 Windows Store App 專案。



建立新專案

最近使用的專案範本(R)

- 空白應用程式 (通用 Windows) C#
- 單元測試專案 (.NET Framework) C#

空白

所有語言(L) 所有平台(P) 所有專案類型(T)

空白應用程式 (通用 Windows)
這是不具預先定義之控制項或配置的單一頁面通用 Windows 平台 (UWP) App 專案。
C# XAML Windows Xbox UWP 桌面

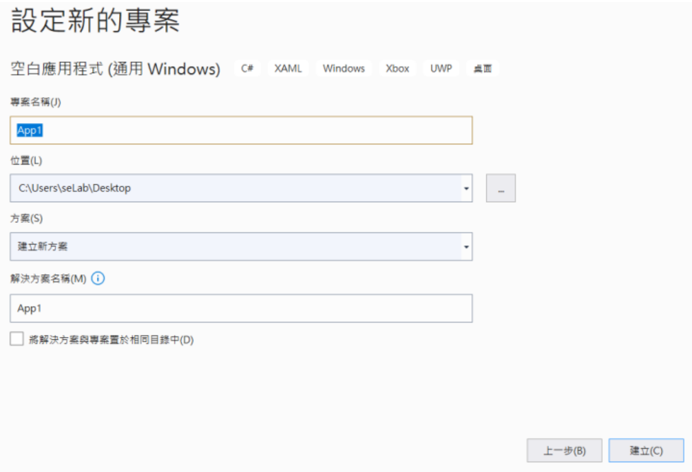
空白應用程式 (通用 Windows)
這是不具預先定義之控制項或配置的單一頁面通用 Windows 平台 (UWP) App 專案。
Visual Basic XAML Windows Xbox UWP 桌面

空白方案
建立不含專案的空白方案
其他

空白專案 (.NET Framework)
建立本機應用程式的空白專案
C# Windows 桌面

空白專案 (.NET Framework)
建立本機應用程式的空白專案
Visual Basic Windows 桌面

Figure 2 開啟新專案



設定新的專案

空白應用程式 (通用 Windows) C# XAML Windows Xbox UWP 桌面

專案名稱(J)
App1

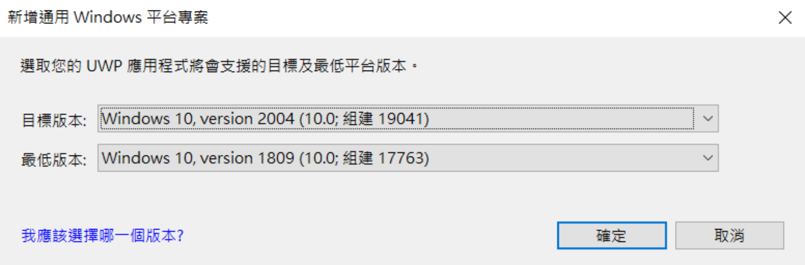
位置(L)
C:\Users\seLab\Desktop

方案(S)
建立新方案

解決方案名稱(M)
App1

□ 將解決方案與專案置於相同目錄中(D)

Figure 3 輸入專案名稱



新增通用 Windows 平台專案

選取您的 UWP 應用程式將會支援的目標及最低平台版本。

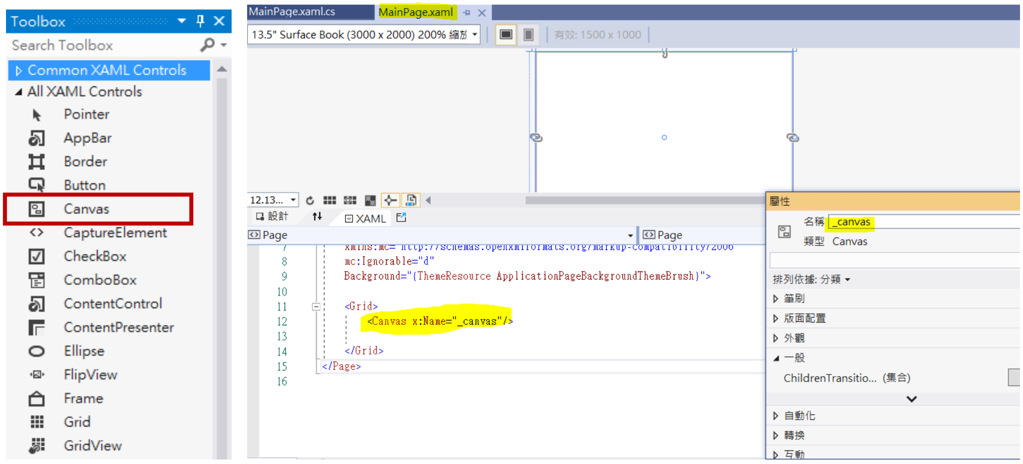
目標版本: Windows 10, version 2004 (10.0; 組建 19041)

最低版本: Windows 10, version 1809 (10.0; 組建 17763)

Figure 4 選擇預設的 UWP 版本

Step 2　Create a Canvas

從方案總管中，點選 MainPage.xaml，從 Toolbox 中點選 Canvas 並拖拉至 MainPage.xaml 中，並將 canvas 命名為 _canvas (Figure 5)。



```
xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
mc:Ignorable="d"
Background="{ThemeResource ApplicationPageBackgroundThemeBrush}">

    <Grid>
        <Canvas x:Name="_canvas"/>

    </Grid>
</Page>
```

名稱 _canvas
類型 Canvas

Figure 5 Toolbox 拖拉 canvas 及命名

然後開啟 MainPage.xaml.cs 檔案，並且加入以下程式碼

```
public sealed partial class MainPage : Page
{
        private const int CANVAS_SIZE = 400;
        private const int BALL_SIZE = 50;
        private const double STROKE_THICKNESS = 5;
        private SolidColorBrush _whiteColor =
                                    new SolidColorBrush (Colors.White);
        private SolidColorBrush _blueColor =
                                    new SolidColorBrush (Colors.Blue);
        private SolidColorBrush _blackColor =
                                    new SolidColorBrush (Colors.Black);
        private SolidColorBrush _yellowColor =
                                    new SolidColorBrush(Colors.Yellow);
        private SolidColorBrush _greenYellowColor =
                             new SolidColorBrush(Colors.GreenYellow);
        private SolidColorBrush _purpleColor =
                                    new SolidColorBrush(Colors.Purple);
        public MainPage()
        {
            this.InitializeComponent();
            _canvas.Width = CANVAS_SIZE;
            _canvas.Height = CANVAS_SIZE;
            _canvas.Background = _whiteColor;
        }
}
```

之後按下執行鍵 (Figure 6)，可以看到執行結果如 Figure 7 所示。

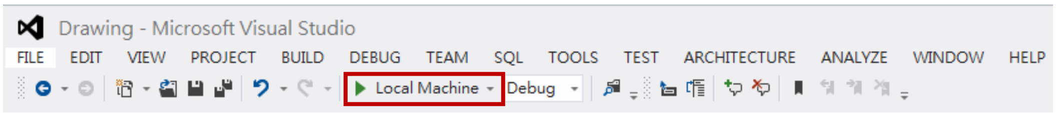


Figure 6 按下執行

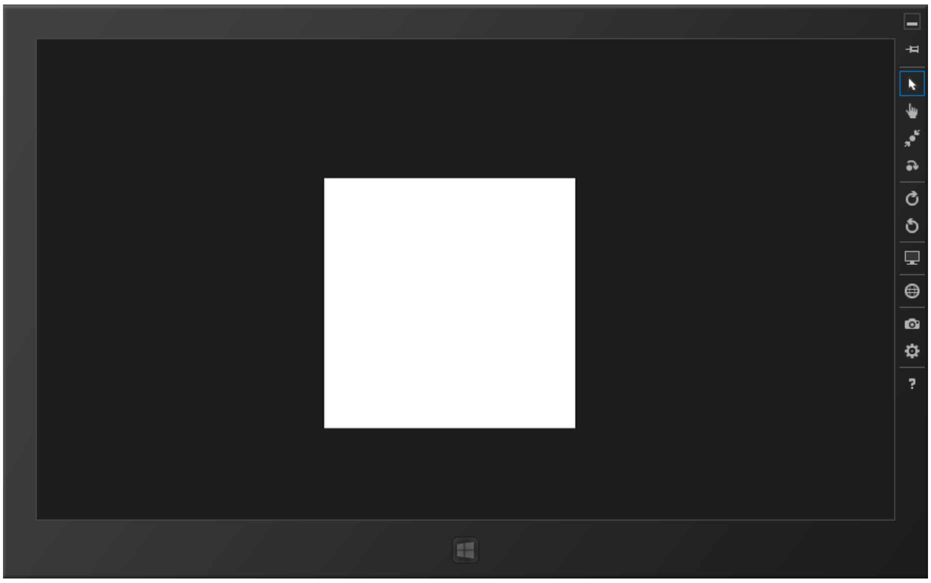


Figure 7 執行結果

Step 3 Draw some Rectangle and Ellipse

接著要在 Canvas 中加入一些圖形，先在 MainPage.xaml.cs 中加入 InitializeShape 的函式

```
private Shape InitializeShape(Shape shape, int left, int top, int right,
int bottom, SolidColorBrush fillColorBrush)
{
     shape.Margin = new Thickness(left, top, right, bottom);
     shape.Width = right;
     shape.Height = bottom;
     shape.Fill = fillColorBrush;
     return shape;
}
```

接著在 MainPage()中加入：

```
// Three blue blocks
Rectangle rectangle = new Rectangle();
InitializeShape(rectangle, 0, 0, 150, 100, _blueColor);
_canvas.Children.Add(rectangle);

rectangle = new Rectangle();
InitializeShape(rectangle, 0, 250, 150, 150, _blueColor);
_canvas.Children.Add(rectangle);

rectangle = new Rectangle();
InitializeShape(rectangle, 300, 0, 100, 275, _blueColor);
_canvas.Children.Add(rectangle);

// Five balls
Ellipse ellipse = new Ellipse();
InitializeShape(ellipse, 200, 25, BALL_SIZE, BALL_SIZE,
_greenYellowColor);
_canvas.Children.Add(ellipse);

ellipse = new Ellipse();
InitializeShape(ellipse, 200, 125, BALL_SIZE, BALL_SIZE,
_greenYellowColor);
_canvas.Children.Add(ellipse);

ellipse = new Ellipse();
InitializeShape(ellipse, 200, 225, BALL_SIZE, BALL_SIZE,
_greenYellowColor);
_canvas.Children.Add(ellipse);
```

```
ellipse = new Ellipse();
InitializeShape(ellipse,    200,    325,    BALL_SIZE,    BALL_SIZE,
_greenYellowColor);
_canvas.Children.Add(ellipse);

ellipse = new Ellipse();
InitializeShape(ellipse,    300,    325,    BALL_SIZE,    BALL_SIZE,
_greenYellowColor); _canvas.Children.Add(ellipse);
```

按下執行鍵，可以看到結果如 Figure 8

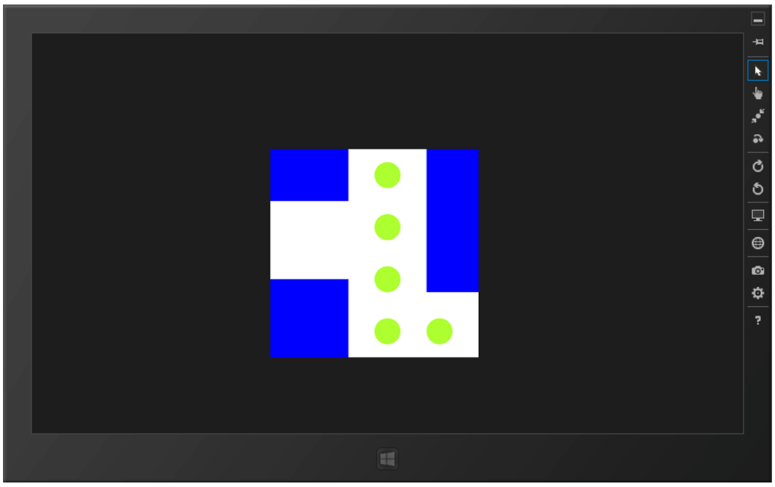


Figure 8 繪圖執行結果

Step 4 The shape StrokeThickness

做完剛剛的練習後，已經可以看到有三個矩形和五個圓形，接著我們要為他們加上外框，修改 InitializeShape 的副函式為以下，執行結果如 Figure 9：

```
private Shape InitializeShape(Shape shape, int left, int top, int right,
                int bottom, SolidColorBrush fillColorBrush)
{
     shape.Margin = new Thickness(left, top, right, bottom);
     shape.Width = right;
     shape.Height = bottom;
     shape.Fill = fillColorBrush;
     shape.Stroke = _purpleColor;
     shape.StrokeThickness = STROKE_THICKNESS;
     return shape;
}
```

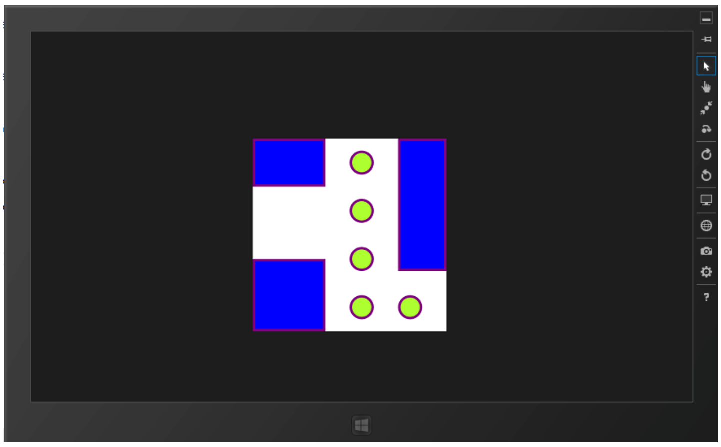


Figure 9 加上邊框執行結果

Step 5 Create a textBlock

在 MainPage.xaml 中，從工具箱加入一個 textBlock 放在 canvas 上方，將其命名為 _textBlock (命名方式同 Step 2)，其 textBlock 與 canvas 的相對位置如 Figure 1。

Step 6 Add Mouse events

開啟 MainPage.xaml.cs 並在 MainPage() 中補充以下程式碼設定 textBlock 預設文字，以及註冊滑鼠點擊和移動的事件。

```
_textBlock.Text = "Where is the mouse?";
_canvas.PointerPressed += PressOnCanvas;
_canvas.PointerMoved += MoveOnCanvas;
```

然後我們在 MainPage.xaml.cs 中增加滑鼠點擊和滑鼠移動事件的方法。

```
        //滑鼠點擊
        private void PressOnCanvas(object sender, PointerRoutedEventArgs e)
        {
            double pressX = Math.Round(e.GetCurrentPoint(_canvas).Position.X, 2);
            double pressY = Math.Round(e.GetCurrentPoint(_canvas).Position.Y, 2);
            _textBlock.Text = "You pressed on (" + pressX + ", " + pressY + ")";
        }
```

```
        //滑鼠移動
        private void MoveOnCanvas(object sender, PointerRoutedEventArgs e)
        {
            double moveX = Math.Round(e.GetCurrentPoint(_canvas).Position.X, 2);
            double moveY = Math.Round(e.GetCurrentPoint(_canvas).Position.Y, 2);
            _textBlock.Text = "You moved on (" + moveX + ", " + moveY + ")";
        }
```

按下執行後，可以看到結果如 Figure 1

- The End -