五專資工三甲

視窗程式設計

老師：陳定宏

陳聖允

學號：5B0G0011

程式說明

請設計一個視窗程式，讓使用者輸入三個數值後，判斷此三個數值是否可以構成三角形。你的程式必須滿足以下要求。

三個 TextBox 來輸入三個邊長，如果某個輸入邊長不是數值(使用 Double.TryParse 方法)，或是數值小於 0 則顯示訊息方塊要求使用者重新輸入。

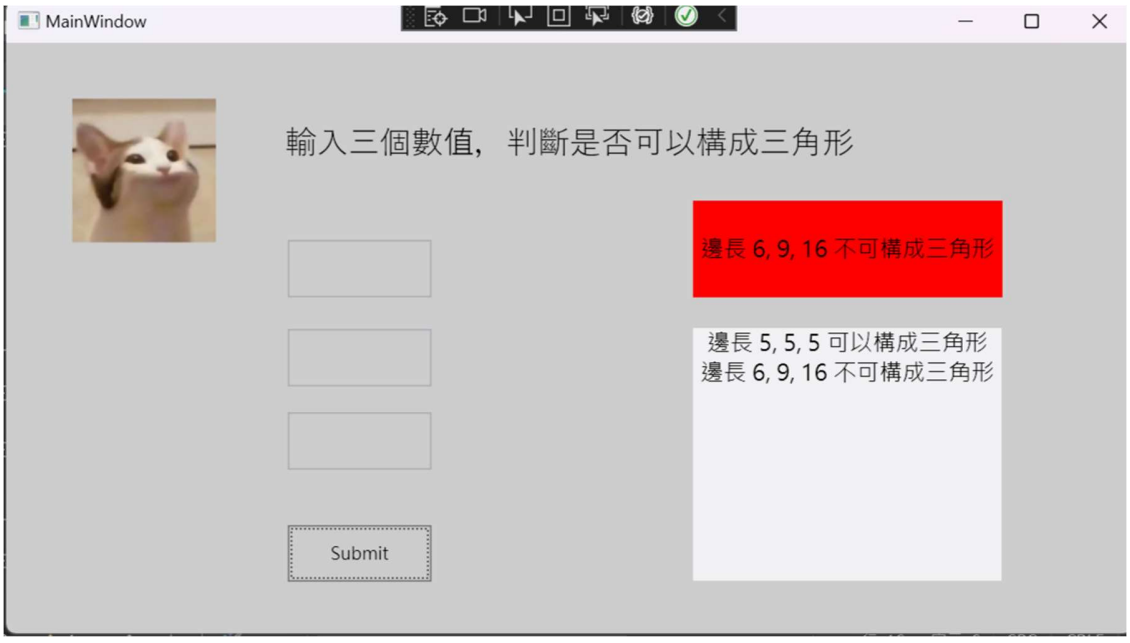
一個 Label 元件用來呈現此三個數值是否可以構成三角形。舉例如果三邊長 5,6,7 可構成三角形，則將此 Label 背景顏色設為綠色，並顯示"邊長 5, 6, 7 可構成三角形；否則將 Label 背景顏色設為紅色，並顯示"邊長 3,4,8 不可構成三角形"。

設計一個 Triangle 類別來儲存三角形資訊。屬性為三個 double 變數用來儲存三邊長資訊，以及一個 bool 變數用來儲存此三邊長是否可構成三角形。以 List<Triangle>型態來產生 triangles 物件，用以儲存所有判斷的三角形，並將所有的測試過程顯示於一個 TextBlock 元件上。

執行結果:

下面連結是動態執行結果

https://media.giphy.com/media/aArrDDyGO2A3U8FQDs/giphy.gif



```
<Window x:Class="WpfApp1.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:local="clr-namespace:WpfApp1"
        mc:Ignorable="d"
        Title="MainWindow" Height="450" Width="800">
    <Grid Background="#FFCDCDCD">
        <TextBox HorizontalAlignment="Left" Margin="196,137,0,0"
          TextWrapping="Wrap" Text="" Name="TextBox01"
          VerticalAlignment="Top" Width="100" Height="40"
          TextAlignment="Center" FontSize="18"
          PreviewKeyDown="TextBox_PreviewKeyDown">
            <TextBox.Background>
                <SolidColorBrush Color="White" Opacity="0"/>
            </TextBox.Background>
        </TextBox>
        <TextBox HorizontalAlignment="Left" Margin="196,199,0,0"
          TextWrapping="Wrap" Text="" Name="TextBox02"
          VerticalAlignment="Top" Width="100" Height="40"
          TextAlignment="Center" FontSize="18"
          PreviewKeyDown="TextBox_PreviewKeyDown">
            <TextBox.Background>
                <SolidColorBrush Color="White" Opacity="0"/>
            </TextBox.Background>
        </TextBox>
        <TextBox HorizontalAlignment="Left" Margin="196,257,0,0"
          TextWrapping="Wrap" Text="" Name="TextBox03"
          VerticalAlignment="Top" Width="100" Height="40"
          TextAlignment="Center" FontSize="18"
          PreviewKeyDown="TextBox_PreviewKeyDown">
            <TextBox.Background>
                <SolidColorBrush Color="White" Opacity="0"/>
            </TextBox.Background>
        </TextBox>
        <Button Content="Submit" HorizontalAlignment="Left"
          Margin="196,336,0,0" VerticalAlignment="Top" Height="39"
          Width="100" Click="Button_Click">
            <Button.Background>
                <SolidColorBrush Color="#FFDDDDDD" Opacity="0"/>
            </Button.Background>
        </Button>
        <Label Content="輸入三個數值，判斷是否可以構成三角形"
          HorizontalAlignment="Center" Margin="0,39,0,0"
          VerticalAlignment="Top" Height="58" Width="480"
          HorizontalContentAlignment="Center"
          VerticalContentAlignment="Center" FontSize="22" FontFamily="Yu
           Gothic UI Semilight"/>
        <TextBlock HorizontalAlignment="Left" Height="176"
          Margin="478,199,0,0"  Name="Cout" VerticalAlignment="Top"
```

```
            Width="215" FontSize="16" TextAlignment="Center"
            TextWrapping="Wrap" Padding="0,0,0,10"
            UseLayoutRounding="True" ScrollViewer.CanContentScroll="True">
              <TextBlock.Background>
                  <SolidColorBrush Color="#FFFBFBFD" Opacity="0.8"/>
              </TextBlock.Background>
          </TextBlock>

          <Label Content="" HorizontalAlignment="Left"
            Margin="478,110,0,0" VerticalAlignment="Top" Height="67"
            Width="215" Name="ltest" HorizontalContentAlignment="Center"
            VerticalContentAlignment="Center" FontSize="16"
            Foreground="Black" >
              <Label.Background>
                  <SolidColorBrush Color="White" Opacity="0.8"/>
              </Label.Background>
          </Label>
          <Image x:Name="cat" HorizontalAlignment="Left" Height="100"
            Margin="46,39,0,0" VerticalAlignment="Top" Width="100"
            Source="/cat1.jpeg"/>

      </Grid>
  </Window>

```

```
 1  using System;
 2  using System.Collections.Generic;
 3  using System.Windows;
 4  using System.Windows.Controls;
 5  using System.Windows.Input;
 6  using System.Windows.Media;
 7  using System.Windows.Media.Imaging;
 8  using System.Windows.Threading;
 9
10  namespace WpfApp1
11  {
12      /// <summary>
13      /// Interaction logic for MainWindow.xaml
14      /// </summary>
15      public partial class MainWindow : Window
16      {
17          List<Triangle> triangles = new List<Triangle>();
18          public MainWindow()
19          {
20              InitializeComponent();
21
22              timer.Interval = TimeSpan.FromSeconds(0.099);
23              timer.Tick += Timer_Tick;
24
25              originalBitmap = new BitmapImage();
26              originalBitmap.BeginInit();
27              originalBitmap.UriSource = new Uri(originalImagePath,
                  UriKind.Relative);
28              originalBitmap.EndInit();
29              cat.Source = originalBitmap;
30          }
31
32
33
34          private void Button_Click(object sender, RoutedEventArgs e)
35          {
36              Double numberA,numberB,numberC;
37
38              bool A1 = Double.TryParse(TextBox01.Text, out numberA);
39              bool B2 = Double.TryParse(TextBox02.Text, out numberB);
40              bool C3 = Double.TryParse(TextBox03.Text, out numberC);
41
42              if ( !A1 || !B2 || !C3 || numberA <=0 || numberB <=0 ||
                    numberC<=0)
43              {
44                  MessageBox.Show("請輸入正確數值不可小於0或是空白", "輸入錯
                      誤");
45                  return;
46              }
47              Triangle triangle = new Triangle(numberA, numberB,
                    numberC);
48              if (triangle.IsValid)
49              {
```

```
50                  ltest.Content = $"邊長 {numberA}, {numberB}, {numberC}
                      可以構成三角形";
51                  ltest.Background = new SolidColorBrush((Color)
                      ColorConverter.ConvertFromString("#5FEB64"));
52
53              }
54              else
55              {
56                  ltest.Content = $"邊長 {numberA}, {numberB}, {numberC}
                      不可構成三角形";
57                  ltest.Background = Brushes.Red;
58              }
59              Cout.Text += $"{ltest.Content}\n";
60              TextboxReset();
61              string newImgagePath = "cat2.jpeg";
62              BitmapImage bitmap = new BitmapImage();
63              bitmap.BeginInit();
64              bitmap.UriSource = new Uri(newImgagePath,
                  UriKind.Relative);
65              bitmap.EndInit();
66              cat.Source = bitmap;
67              timer.Start();
68
69
70          }
71          private void TextBox_PreviewKeyDown(object sender, KeyEventArgs
                e)
72          {
73              if (e.Key == Key.Enter)
74              {
75                  // 防止換行
76                  e.Handled = true;
77
78                  // 尋找下一個 TextBox
79                  TraversalRequest request = new TraversalRequest
                      (FocusNavigationDirection.Next);
80                  UIElement elementWithFocus = Keyboard.FocusedElement as
                       UIElement;
81                  if (elementWithFocus != null)
82                  {
83                      elementWithFocus.MoveFocus(request);
84                  }
85              }
86          }
87          private DispatcherTimer timer = new DispatcherTimer();
88
89          private string originalImagePath = "cat1.jpeg";
90
91          private BitmapImage originalBitmap;
92
93          private void Timer_Tick(object? sender, EventArgs e)
94          {
95
```

```
 96              // 停止計時器
 97              timer.Stop();
 98
 99              // 切換回原始圖片
100              cat.Source = originalBitmap;
101
102          }
103          private void TextboxReset()
104          {
105              TextBox01.Text = "";
106              TextBox02.Text = "";
107              TextBox03.Text = "";
108
109          }
110      }
111  }
112
```

```
1  namespace WpfApp1
2  {
3      internal class Triangle
4      {
5          public double SideA { get; set; }
6          public double SideB { get; set; }
7          public double SideC { get; set; }
8          public bool IsValid { get; set; }
9          public string? Message { get; set; }
10
11         public Triangle(double a, double b, double c)
12         {
13             SideA = a;
14             SideB = b;
15             SideC = c;
16             IsValid = IsTriangleValid(a, b, c);
17         }
18
19         private bool IsTriangleValid(double a, double b, double c)
20         {
21             return (a + b > c) && (b + c > a) && (c + a > b);
22         }
23     }
24 }
25
```