HomeWork2: Tracing the Code

發現的 bug:

在 select mode 情況下, select 的物件有包含 ConnectionLine 的話, 按
 Edit → Delete 就會 crash

因為是 delete 動作出現問題,所以點開 DeleteAction.cs 看用到哪些function (如圖 1),然後看到 DestoryAllCombinations(),點前往實作,然後設定 break point (如圖 2),注意到 Combination.Count 是迴圈的條件,所以之後會把這個變數釘選起來。RemoveCombination(),前往實作,設定break point

```
Program.cs 4
                 UMLEditorForm.cs ∓
                                         MenuBar.cs 4
                                                           EditContextMenu.cs 7
                                                                                    DeleteAction.cs
C# UMLEditor
                                               ▼ UMLEditor.Actions.DeleteAction
                                                                                                  → 💮
                    If the <see cref="Shape"/> to be deleted contains any <see cref="CompositionUbject"/
                /// then the <see cref="Shape"/>s hold by the same <see cref="CompositionObject"/> will
                /// be deleted, too.
                /// </remarks>
                /// <seealso cref="UMLAction.Trigger()"/>
                12 個參考
                public override void Trigger()
                    foreach (Shape shape in Canvas.SelectedShapes)
                       shape.DestroyAllCombinations();
                        Canvas.RemoveShape(shape);
                    base.Trigger();
```

圖 1

同樣點進去 Destory()、RemoveCombination(), 前往實作, 設定 break point (如圖 3、4)

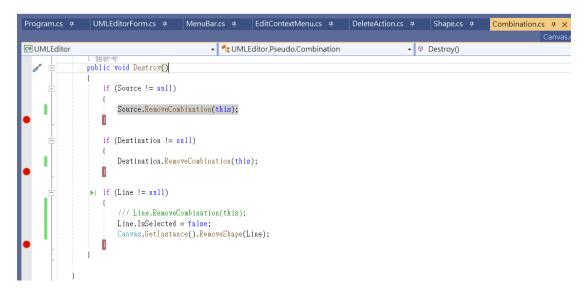


圖 3

```
Program.cs #
                 UMLEditorForm.cs #
                                         MenuBar.cs ∓
                                                          EditContextMenu.cs #
                                                                                  DeleteAction.cs ₽
                                                                                                       Shape.cs 7
C# UMLEditor

▼ UMLEditor.Shapes.Shape

                                                                                                ▼ RemoveCombinat
                /// Please be aware that the <see cref="Combination"/> itself will
                /// not remove this <see cref="Shape"/> on <see cref="Canvas"/> automatically.
               /// </remarks>
               2 個參考
               public virtual void RemoveCombination(Combination combination)
                   Debug.Assert(combination != null);
                   Combinations.Remove(combination);
```

圖 4

開始 test run:先畫兩個 usercase,再隨便畫一種 ConnectionLine,然後用 select mode 選取,再按 Edit → Delete,可以看到目前有一個 Combination,就 是那條 ConnectionLine 的(如圖 5)

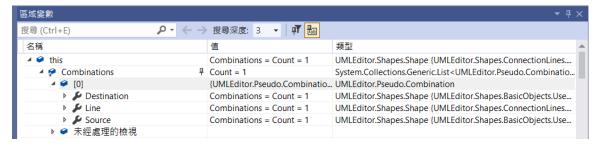


圖 5

繼續執行,可以看到 Source 和 Destination 都有成功 Remove combination (如圖 6、7),但 Line 沒有 Remove combination (如圖 8)

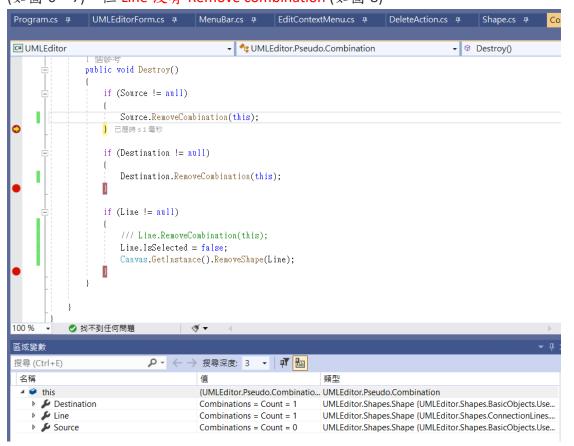


圖 6

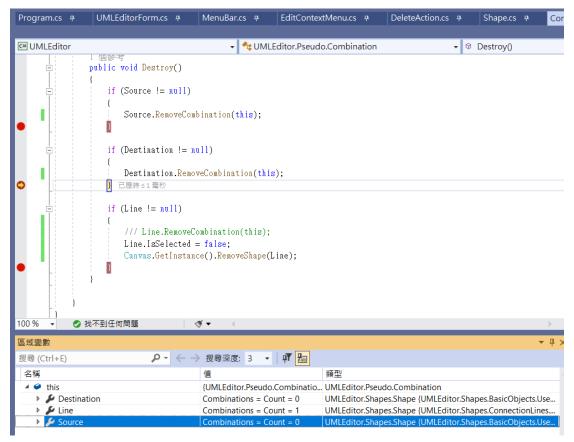
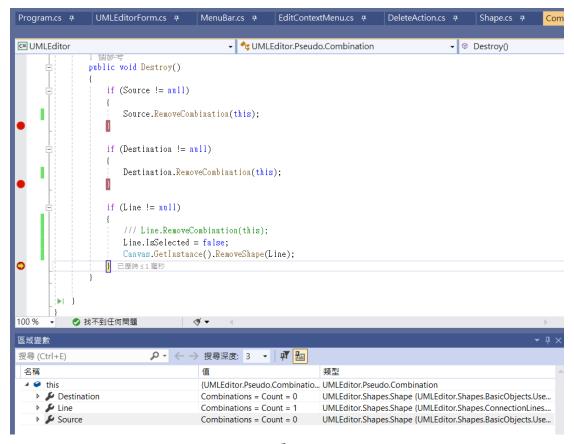
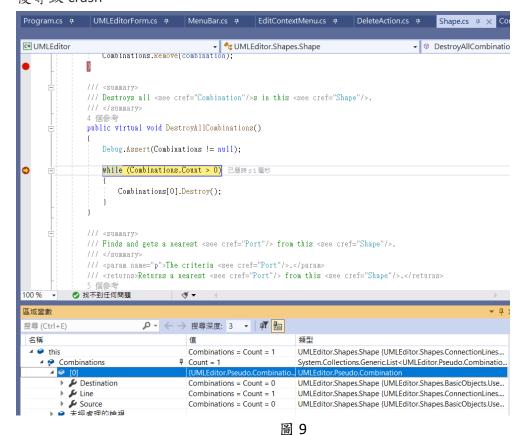


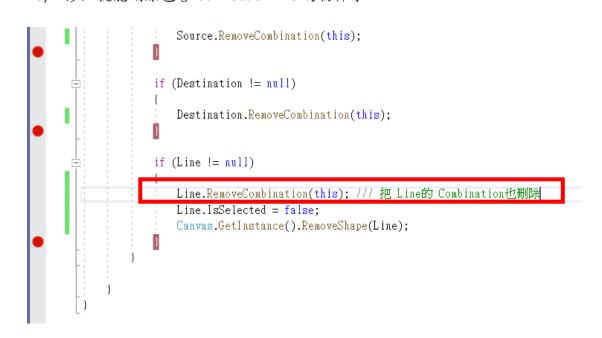
圖 7



因為 Line 沒有 Remove combination,所以才會進入無限迴圈(如圖 9),最後導致 crash



最後修改程式碼,在 Destory()裡 Line 的部分加上 RemoveCombination (如圖10),修正就能删除包含 ConnectionLine 的物件了



2. Group 之後的 composite 物件在 select 之後,想要用滑鼠 drag 移動位置, 選取框框會正常移動,但是裡面的 Shape 物件沒有跟著移動(如圖 11、12)

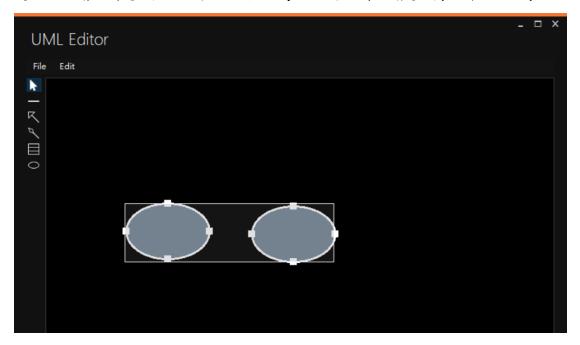


圖 11

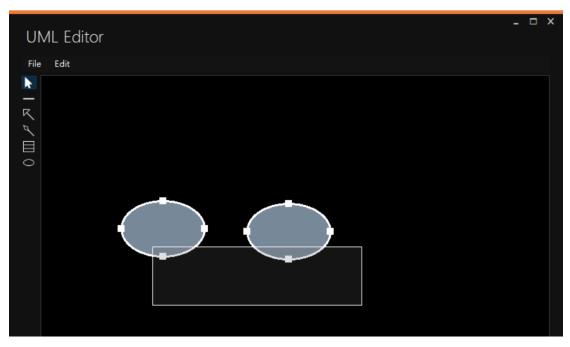


圖 12

因為是在 Select mode 情況下出問題,所以進入 SelectMode.cs,並設置 break point (如圖 13)。再移動 composite 物件後,可以發現進入到 else 條件(如圖 14)

```
| Max(F) 編輯(E) 檢視(V) Git(G) 專案(P) 建置(B) 傾錯(D) 測試(S) 分析(N) 工具(T) 延伸模組(X) 視窗(W) 説明(H) 授尋(Ctrl+Q)
② ▼ ② | 18] ▼ 😩 🕍 🗳 「フ ▼ 🤻 ▼ Debug ▼ Any CPU ▼ ▶ 組織(C) ▼ 🔌 🛝 🔯 📦 🐧 🔻 🐧 🧸 🤚 唯 📱 🥞 📗
處理序: [2036] UML Editor.exe
                                ▼ 図 週期事件 ▼ 執行緒:
Shape.cs # IMouseEvents.cs # SelectMode.cs # X
                / <seeatiso crei= mode.onmouseprag />
個象字
                                                                                   → 🕅 OnMouseDrag(object sender, MouseDragEventArg → 💠
             public override void OnMouseDrag(object sender, MouseDragEventArgs e)
                 if (_hasSelectArea)
                    //No shape selected means formed a selected area
                    _currentMousePoint.X = e.X;
_currentMousePoint.Y = e.Y;
                    UpdateSelectedArea(Shape):
                 {
                    foreach (Shape shape in Canvas.SelectedShapes)
                       shape.Move(e.OffsetX, e.OffsetY);
                       /// shape只是外面那層(Count=1),裡面包含的兩個shapes才是真的圖形,所以再用一個 loop去修改 /// for ( int i=0; i<shape.Count; i++)
                             shape.GetShape(i).Move(e.OffsetX, e.OffsetY);
                 base.OnMouseDrag(sender, e);

    ▼ 找不到任何問題

→ 行: 102 字元: 14 SPC CRLF
```

圖 13

```
★ 檔案(F) 編輯(E) 檢視(V) Git(G) 專案(P) 建置(B) 傾錯(D) 測試(S) 分析(N) 工具(T) 延伸模組(X) 視窗(W) 説明(H) 授尋(Ctrl+Q)
◎ - ○ | 👸 - 👝 🖺 🛂 | り - 🦿 - | Debug - | Any CPU - | ▶ 組織(C) - 🔌 🐚 | 🚳 📮 | Ⅱ 🔹 💍 | 分 ᅷ 📑 🔭 | 🧏 👆 帽 | 🖫 🧏 |
處理序: [2036] UML Editor.exe
                                ▼ 図 週期事件 ▼ 執行緒: [5728] 主執行緒
                                                                               ▼ ▼ ¼ 堆疊框架: UMLEditor.Modes.SelectMode.OnMous ▼
            IMouseEvents.cs ∓ SelectMode.cs ∓ X

    ▼ UMLEditor.Modes.SelectMode
    <seeaiso crei= mode.onmouseprag />

                                                                                     → ♥ OnMouseDrag(object sender, MouseDragEventArg → ‡

□ UMLEditor

             public override void OnMouseDrag(object sender, MouseDragEventArgs e)
                    //No shape selected means formed a selected area
                    _currentMousePoint.X = e.X;
_currentMousePoint.Y = e.Y;
                    UpdateSelectedArea(Shape):
                    foreach (Shape shape in Canvas.SelectedShapes)
                        shape.Move(e.Offset%, e.Offset%);
                        /// shape只是外面那層(Count=1),裡面包含的兩個shapes才是真的圖形,所以再用一個 loop去修改
                        /// for ( int i=0; i<shape.Count; i++)
                              shape.GetShape(i).Move(e.OffsetX, e.OffsetY);
                       7// 1
                 base.OnMouseDrag(sender, e);
          ② 找不到任何問題      ▼
                                                                                                    ▶ 行: 102 字元: 13 SPC CRLF
```

圖 14

再設置 Trace point (如圖 15),看看 shape 到底是什麼、有幾個?可以看出每次移動, shape 都只有一個,而且還包含 Shapes (如圖 16)



圖 15

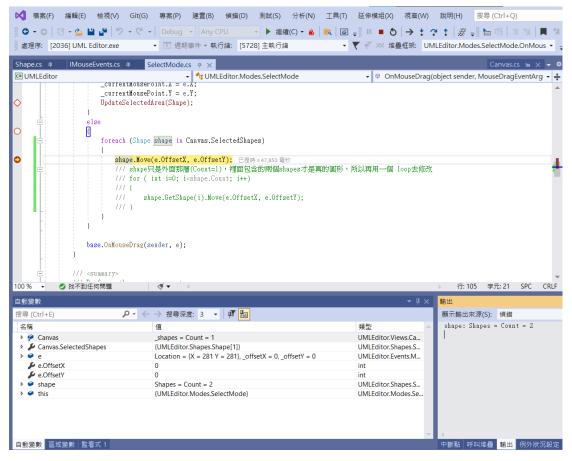


圖 16

可以看到 shape 裡的 Shapes[0]、[1] 才是 composite 物件裡面的圖案 (usecase 或 class),也就是說圖案沒有執行 Move function (如圖 17)

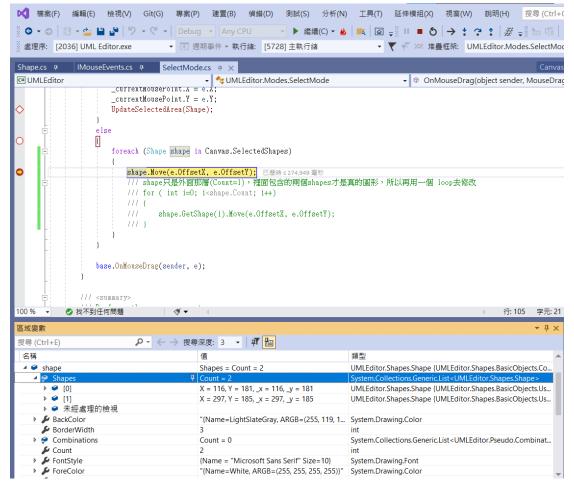


圖 17

因此修正需要加上一層迴圖,把 shape 裡的 Shapes 都 call Move funciton。本來想直接改成 shape.Shapes[i].Move(e.OffsetX, e.OffsetY),但因為是保護層級所以不能直接存取,如圖 18

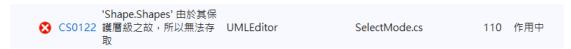


圖 18

在輸入過程,系統提示有 Shape 物件有 GetShape 的 function 可以用(如圖19、20),因此加上一層 for loop,把 shape 裡的 Shapes 都 call Move function (如圖21),之後就能正常移動 composite 物件了

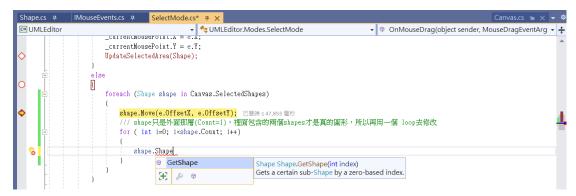


圖 19

圖 20

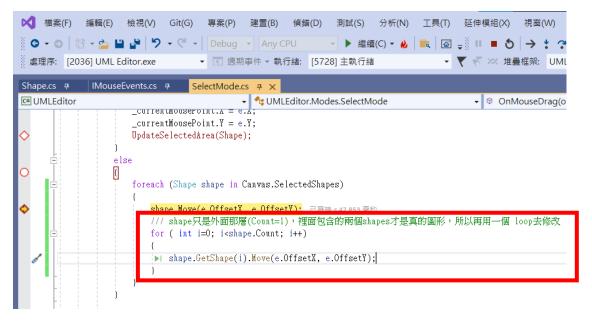


圖 21

3. 當存在多個 Shape 時,位於 Canvas 下方的 Shape 有時會沒辦法 select 起來 (如圖 22)

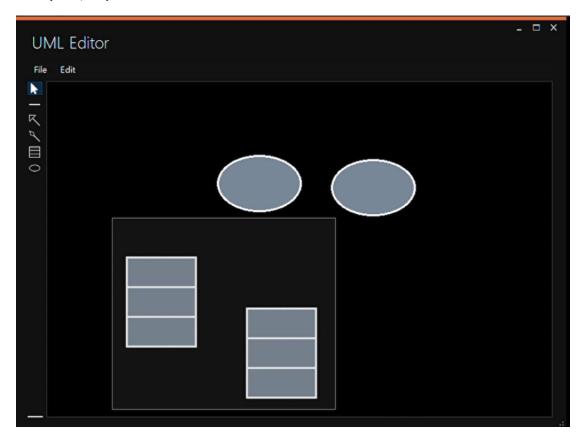


圖 22

因為一樣是 Select mode 下出問題,所以進入 SelectMode.cs 設置 break point。因為看起來 MouseDown 的部分很正常(選取範圍都有顯示出來),感覺是 MouseUp 的問題,所以在 OnMouseUp function 設定 break point (如圖 23)

```
編輯(E) 檢視(V) Git(G) 專案(P) 建置(B) 慎錯(D) 測試(S) 分析(N) 工具(T) 延伸模組(X)
- ▶ 繼續(C) - 心 | 🚉 | 🐼 🚚 💵 👅 💍
                            ▼ 図 週期事件 ▼ 執行緒:
                                                                               ▼ ▼ 72. 堆壘
處理序: [11904] UML Editor.exe
Shape.cs 7
             SelectMode.cs ∓ ×
C# UMLEditor

→ MUMLEditor.Modes.SelectMode

▼ Ø OnMo
              /// <param name="sender">lhe <seealso cret="object"/> which triggers the event.</param>
              /// <param name="e">The <seealso cref="EventArgs"/> containing information of mouse.</param>
             /// <seealso cref="Mode.OnMouseUp"/>
              9 個參考
             public override void OnMouseUp(object sender, MouseEventArgs e)
                 if (_hasSelectArea)
                    Canvas.RemoveShape(Shape);
                    DeselectAll();
                    MultipleSelect(Shape);
                    _hasSelectArea = false;
                    Shape = null;
                    Shape pressedShape = GetPressedShape(new Point(e.X, e.Y));
                    if (pressedShape = null) DeselectAll();
                 base.OnMouseUp(sender, e);
              /// <summary>
              /// Deselect all <see cref="Shape"/>s on the <see cref="Canvas"/>.
                              ≪ ▼
```

圖 23

可以看到進入_hasSelectArea = True 的部分(如圖 24),可以推測 MultipleSelect 可能有問題,所以點前往實作 (如圖 25)

```
編輯(E) 檢視(V) Git(G) 專案(P) 建置(B) 偵錯(D) 測試(S) 分析(N) 工具(T) 延伸模組(X)
▼ ▶ 繼續(C) ▼ W ■ G ■ ■ 5
                            ▼ 週期事件 ▼ 執行緒: [15364] 主執行緒
。 處理序: [11904] UML Editor.exe
Shape.cs #
            SelectMode.cs ₹ X
C# UMLEditor

▼ UMLEditor.Modes.SelectMode

▼ Ø OnMo
              /// <param name="sender">The <seealso cref="object"/> which triggers the event.</param>
             /// <param name="e">The <seealso cref="EventArgs"/> containing information of mouse.</param>
             /// <seealso cref="Mode.OnMouseUp"/>
             9 個參老
             public override void OnMouseUp(object sender, MouseEventArgs e)
                 if (_hasSelectArea)
                { 已歷時≤22,647毫秒
                    Canvas.RemoveShape(Shape);
                    DeselectAll();
                    MultipleSelect(Shape);
                    _hasSelectArea = false;
                    Shape = null;
                else
                    Shape pressedShape = GetPressedShape(new Point(e.X, e.Y));
                    if (pressedShape = null) DeselectAll();
                 base.OnMouseUp(sender, e);
```

圖 24

```
SelectMode.cs ∓ ×
Shape.cs #
C# UMLEditor
                                                                                                    ▼ <sup>©</sup> MultipleSelect(Sh

→ MUMLEditor.Modes.SelectMode

                    return null;
                /// <summary>
                /// Selects all <see cref="Shape"/>s which are within the area covered by
                /// the <see cref="SelectedArea"/>.
                /// </summarv>
                /// <param name="selectedArea">The selection area.</param>
                1 個參考
                private void MultipleSelect(Shape selectedArea)
                    Point p1 = new Point(selectedArea.X, selectedArea.Y);
                    Point p2 = new Point(selectedArea.X + selectedArea.Width, selectedArea.X + selectedArea.Height);
                    foreach (Shape shape in Canvas.Shapes)
                        if (shape.IsWithin(p1, p2))
                            Canvas.RemoveShape(shape);
                            Canvas.AddShape(shape);
                            shape.IsSelected = true;
```

圖 25

因為 break point 會一直停下來,會打斷 select 的動作,所以加入 Trace point (如圖 26),看一下 shape 的 X、Y 和 selectArea 的 p1 和 p2,還有判斷 IsWithin 的 bool 值(如圖 27、28、29)

```
★ / 檀案(F) 編輯(E) 檢視(V) Git(G) 專案(P) 建置(B) 傾錯(D) 測試(S) 分析(N) 工具(T) 延伸模組(X) 視窩(W) 説明(H) 授
▼ ▶ 繼續(C) ▼ & | a | a | a | a | b | b | b | a | a | a | b | b | b | b | c | b | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c | c 
盧理序: [11904] UML Editor.exe ▼ 图 週期事件 ▼ 執行緒:
                                                                                                                                                                                        ママン 堆疊框架:
Shape.cs 7 SelectMode.cs 7 X
C# UMLEditor
                                                                                                                                                                                                    ▼ MultipleSelect(Shape selectedAr

▼ UMLEditor.Modes.SelectMode

                                /// Selects all <see cref="Shape"/>s which are within the area covered by
                                /// the <see cref="SelectedArea"/>.
                                /// </summarv>
                                /// <param name="selectedArea">The selection area.</param>
                                1 個參考
                                private void MultipleSelect(Shape selectedArea)
                                       Point p1 = new Point(selectedArea.X, selectedArea.Y);
                                       Point p2 = new Point(selectedArea.X + selectedArea.Width, selectedArea.X + selectedArea.Height);
                                        foreach (Shape shape in Canvas.Shapes)
                                                if (shape.IsWithin(p1, p2))
                                                        Canvas.RemoveShape(shape);
                                                        Canvas.AddShape(shape):
                                                        shape.IsSelected = true:
                                /// <summary>
                                /// Select the given <see cref="Shape"/> and deselects all other <see cref="Shape"/>s.
              ▼ 找不到任何問題
                                                                     ∅ ▼
                                                                                                                                                                                                                                              行: 186
```

圖 26

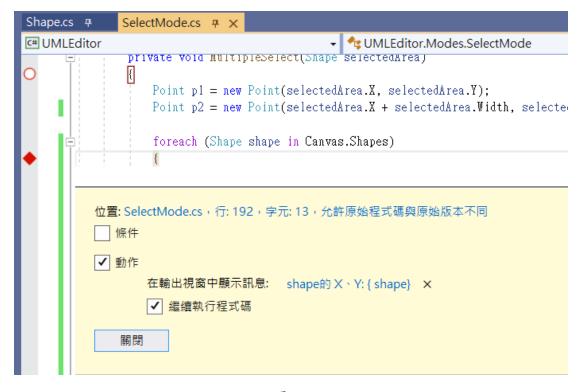


圖 27



圖 28

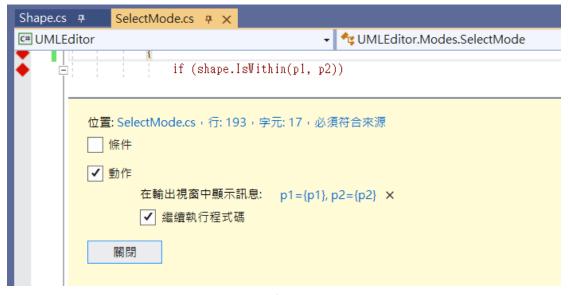


圖 29

先畫兩個圖案,一個在 Canvas 上面,一個在下面,然後看看框上面和下面圖案的差別。可以發現下面的圖案的 p2 的 Y 很奇怪(如圖 30),因為從上面圖案的輸出推測,p1 是左上的 point,p2 則是右下的 point,所以正常情況下,p1.Y 應該小於 p2.Y (如圖 31)

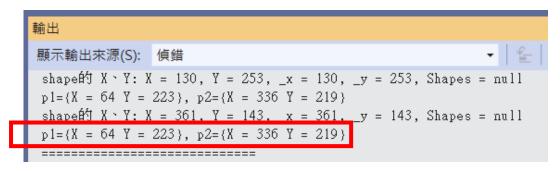


圖 30

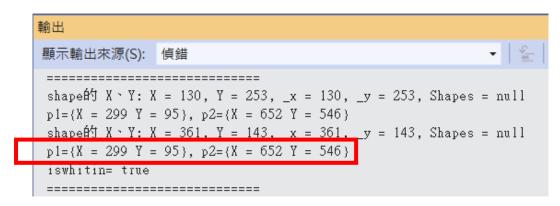


圖 31

p2 的 y 的部分,selectedArea.X + selectedArea.Height 要改成 selectedArea.Y + selectedArea.Height (如圖 32),修正之後就能正常選取了

圖 32