# HomeWork2: Tracing the Code

**發現的 bug：**

1. **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


圖 1

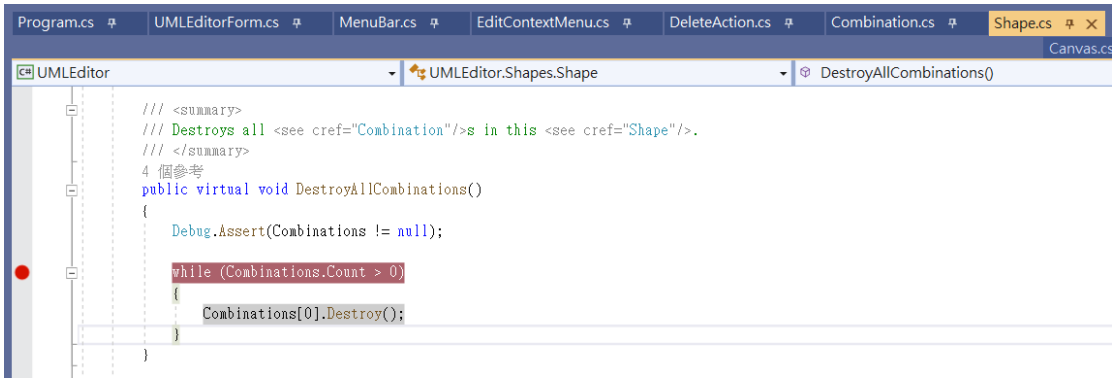


圖 2

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

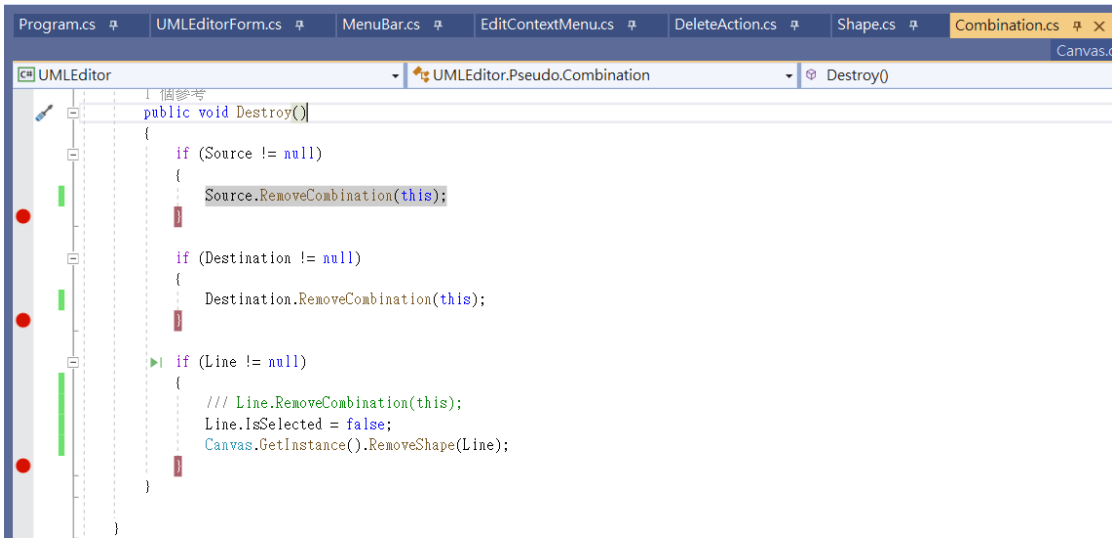


圖 3

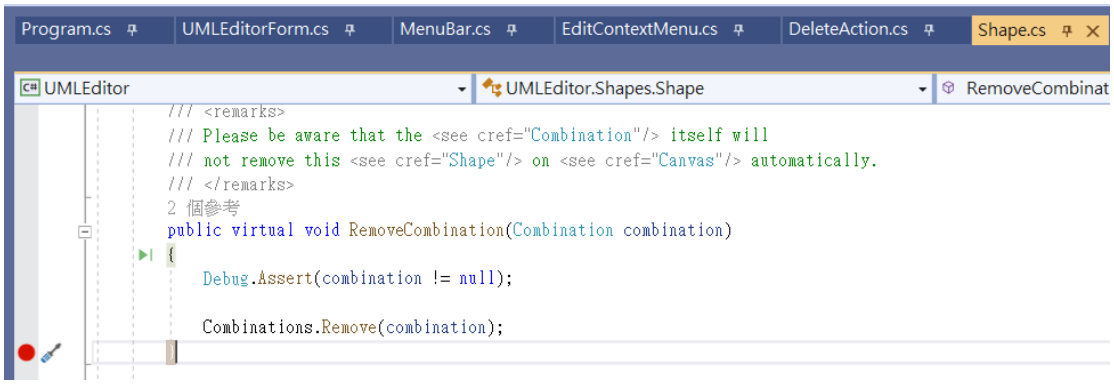


圖 4

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)

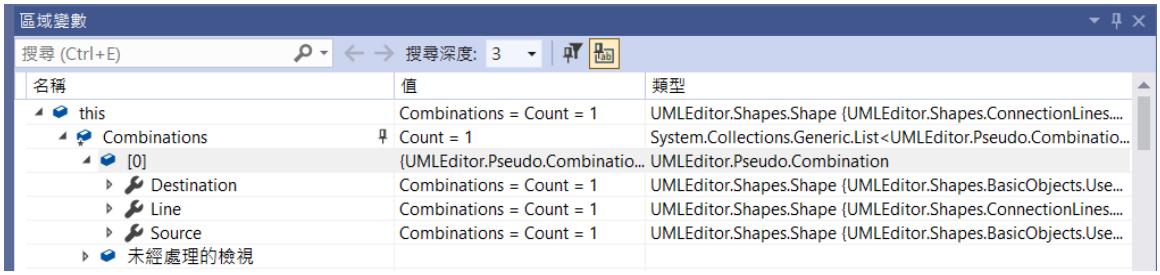


圖 5

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)

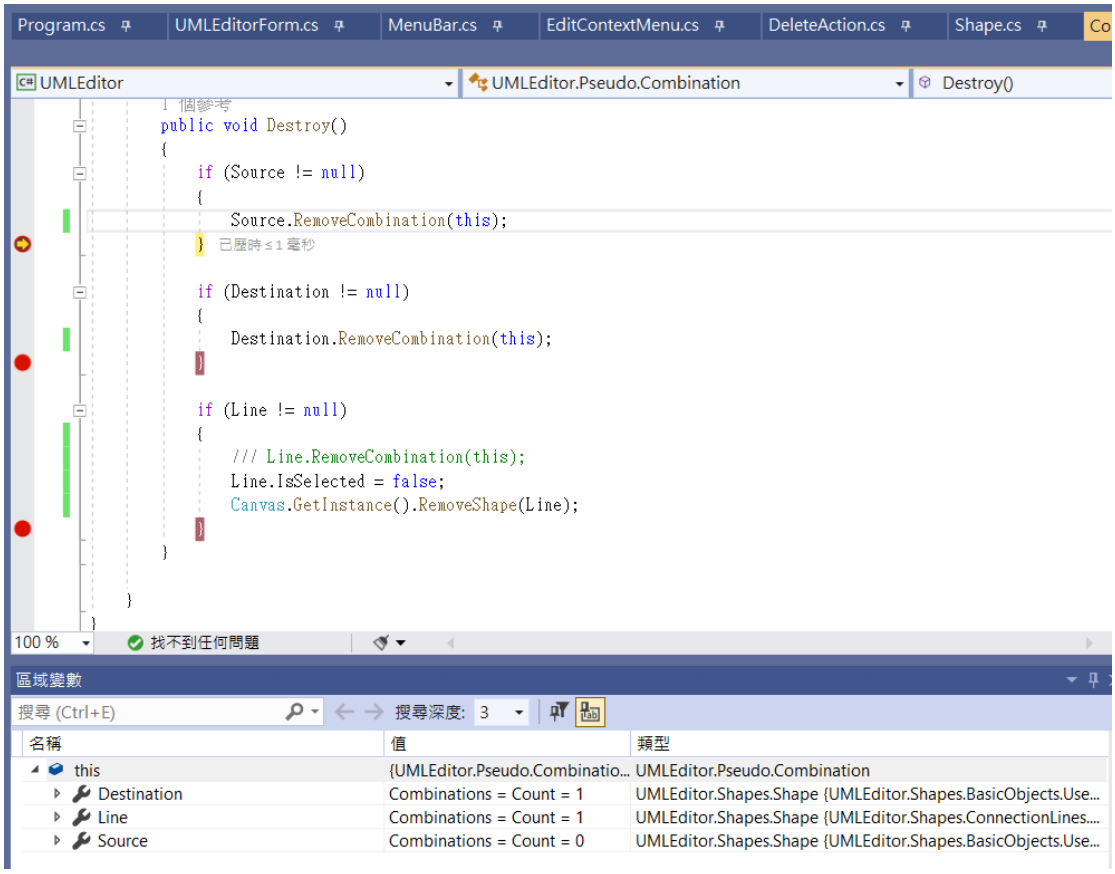


圖 6

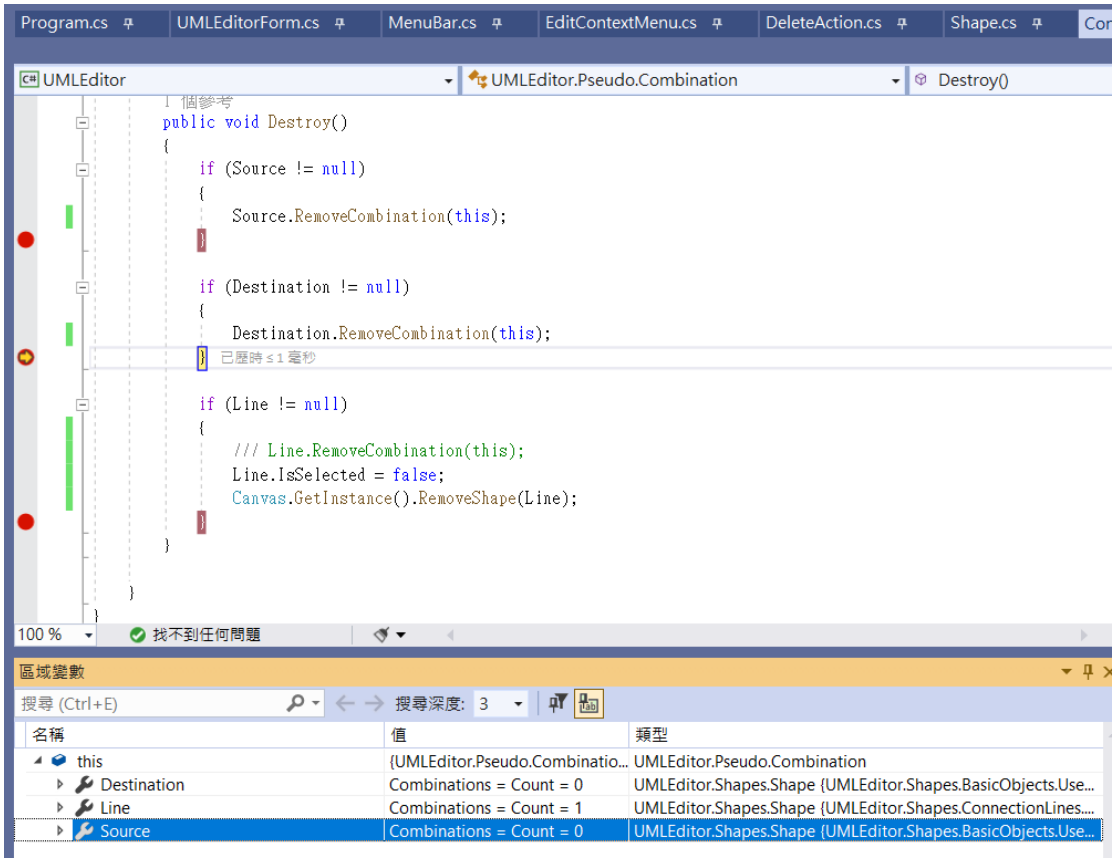
Program.cs
UMLEditorForm.cs
MenuBar.cs
EditContextMenu.cs
DeleteAction.cs
Shape.cs
UMLEditor
UMLEditor.Pseudo.Combination
Destroy()
1 個參考
public void Destroy()
{
if (Source != null)
{
Source.RemoveCombination(this);
}
if (Destination != null)
{
Destination.RemoveCombination(this);
} 已歷時 ≤ 1 毫秒
if (Line != null)
{
/// Line.RemoveCombination(this);
Line.IsSelected = false;
Canvas.GetInstance().RemoveShape(Line);
}
}
}
100 %
找不到任何問題
區域變數
搜尋 (Ctrl+E)
搜尋深度: 3
名稱
值
類型
this
{UMLEditor.Pseudo.Combinatio...
UMLEditor.Pseudo.Combination
Destination
Combinations = Count = 0
UMLEditor.Shapes.Shape {UMLEditor.Shapes.BasicObjects.Use...
Line
Combinations = Count = 1
UMLEditor.Shapes.Shape {UMLEditor.Shapes.ConnectionLines....
Source
Combinations = Count = 0
UMLEditor.Shapes.Shape {UMLEditor.Shapes.BasicObjects.Use...

圖 7

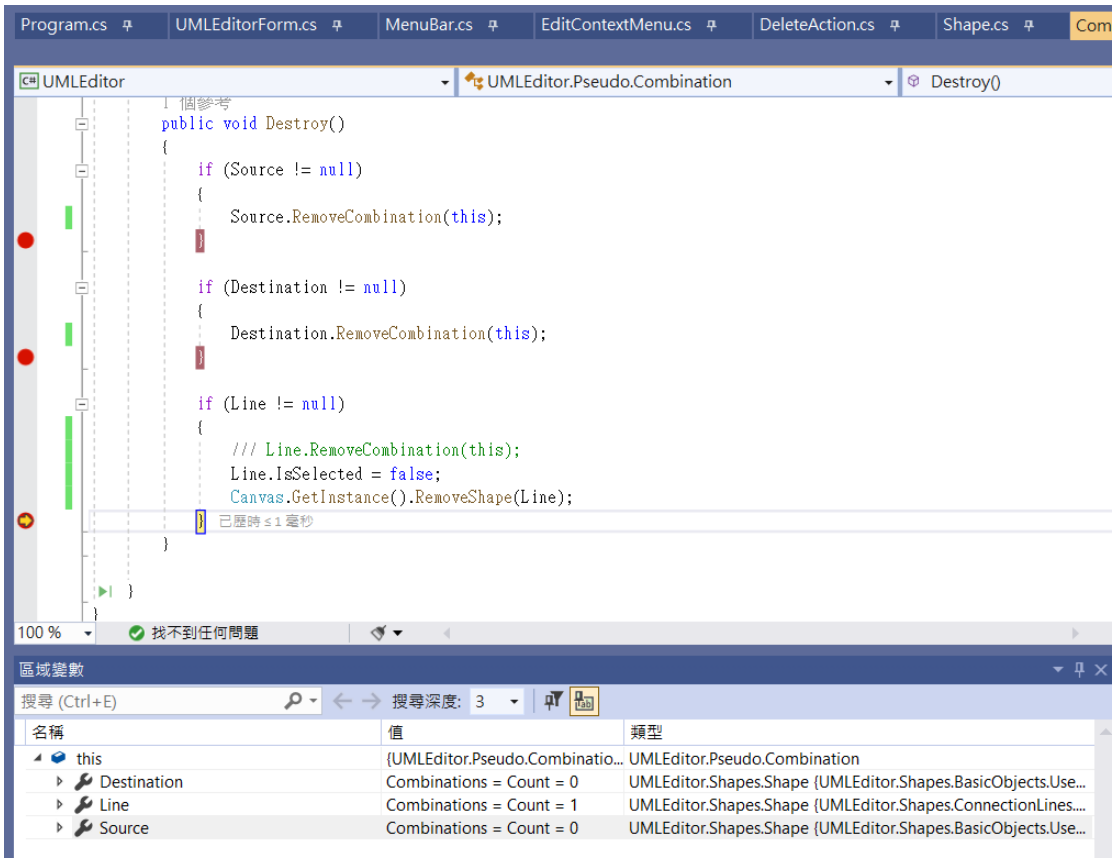
Program.cs
UMLEditorForm.cs
MenuBar.cs
EditContextMenu.cs
DeleteAction.cs
Shape.cs
UMLEditor
UMLEditor.Pseudo.Combination
Destroy()
1 個參考
public void Destroy()
{
if (Source != null)
{
Source.RemoveCombination(this);
}
if (Destination != null)
{
Destination.RemoveCombination(this);
}
if (Line != null)
{
/// Line.RemoveCombination(this);
Line.IsSelected = false;
Canvas.GetInstance().RemoveShape(Line);
} 已歷時 ≤ 1 毫秒
}
}
100 %
找不到任何問題
區域變數
搜尋 (Ctrl+E)
搜尋深度: 3
名稱
值
類型
this
{UMLEditor.Pseudo.Combinatio...
UMLEditor.Pseudo.Combination
Destination
Combinations = Count = 0
UMLEditor.Shapes.Shape {UMLEditor.Shapes.BasicObjects.Use...
Line
Combinations = Count = 1
UMLEditor.Shapes.Shape {UMLEditor.Shapes.ConnectionLines....
Source
Combinations = Count = 0
UMLEditor.Shapes.Shape {UMLEditor.Shapes.BasicObjects.Use...

圖 8

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

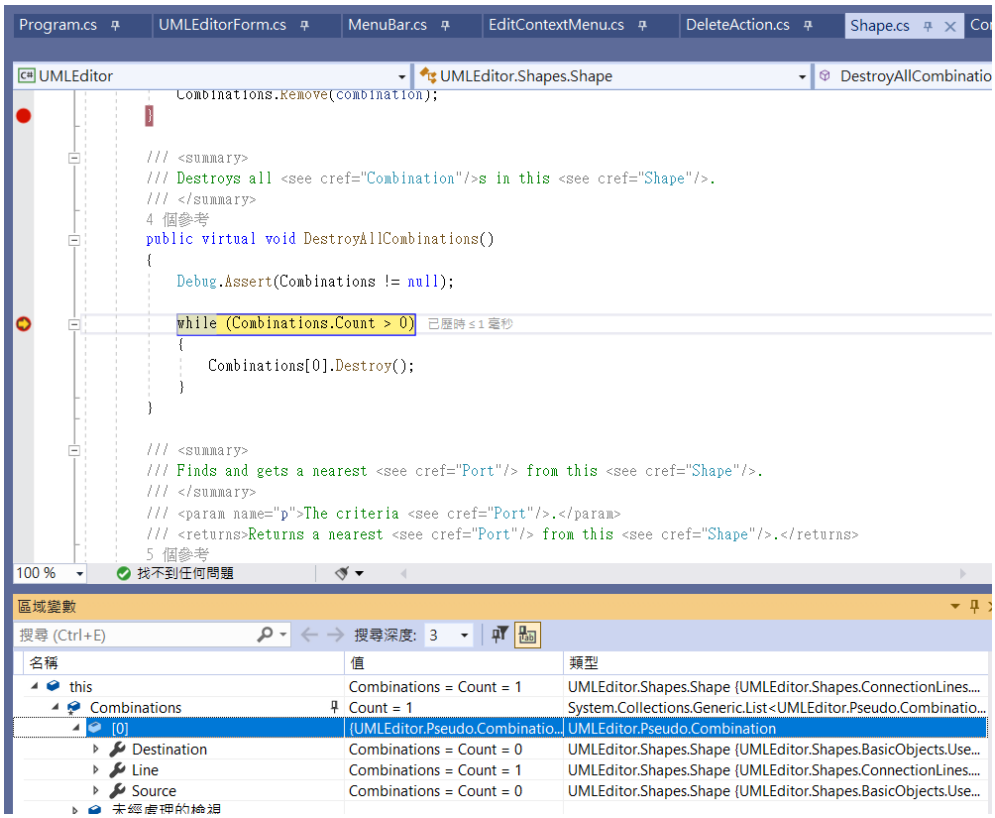


圖 9

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

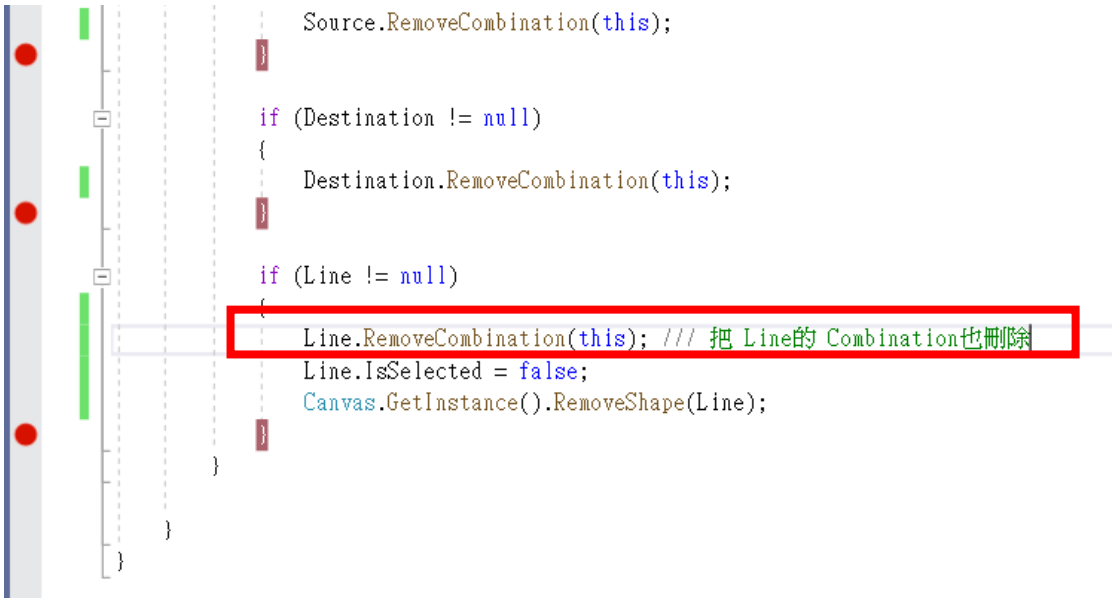


圖 10

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)**

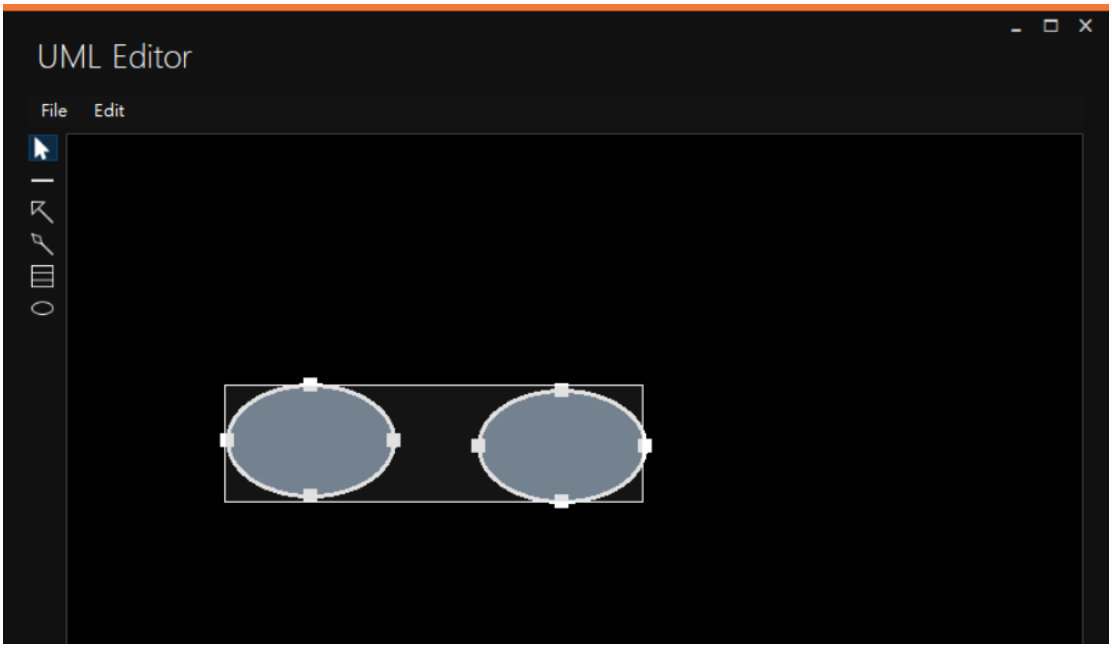


圖 11

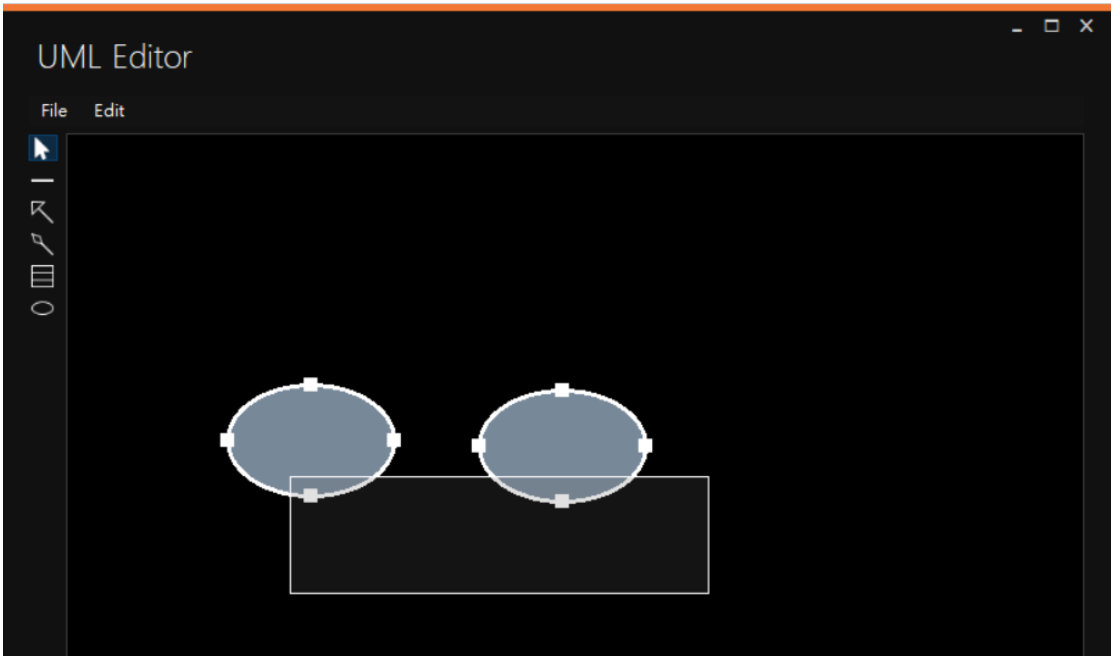


圖 12

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)

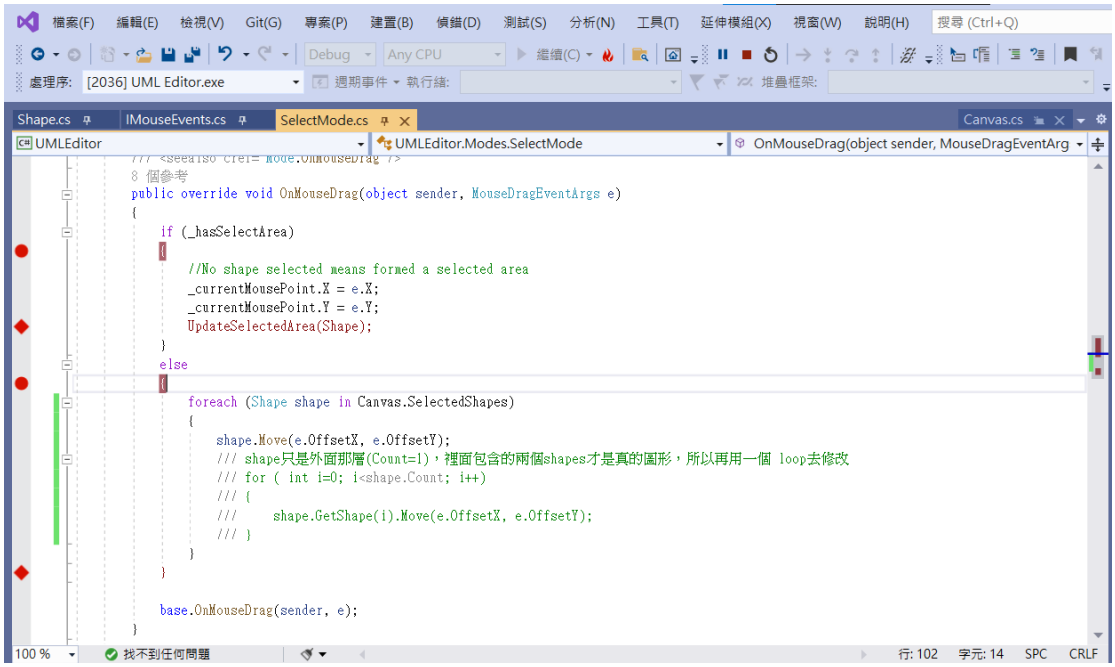


圖 13

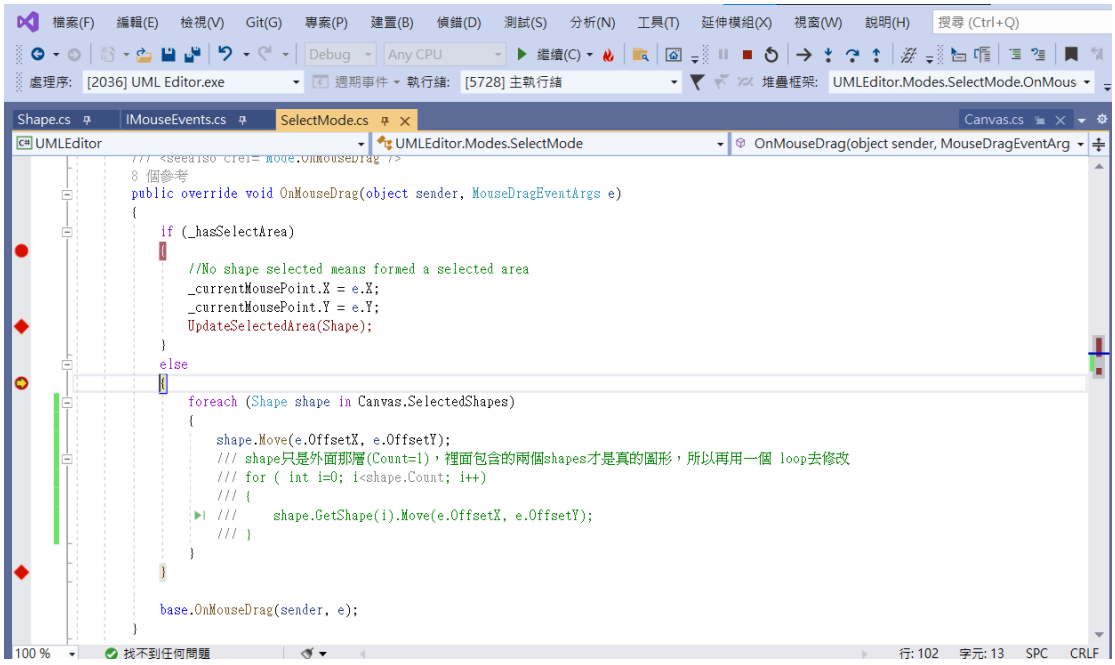


圖 14

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)

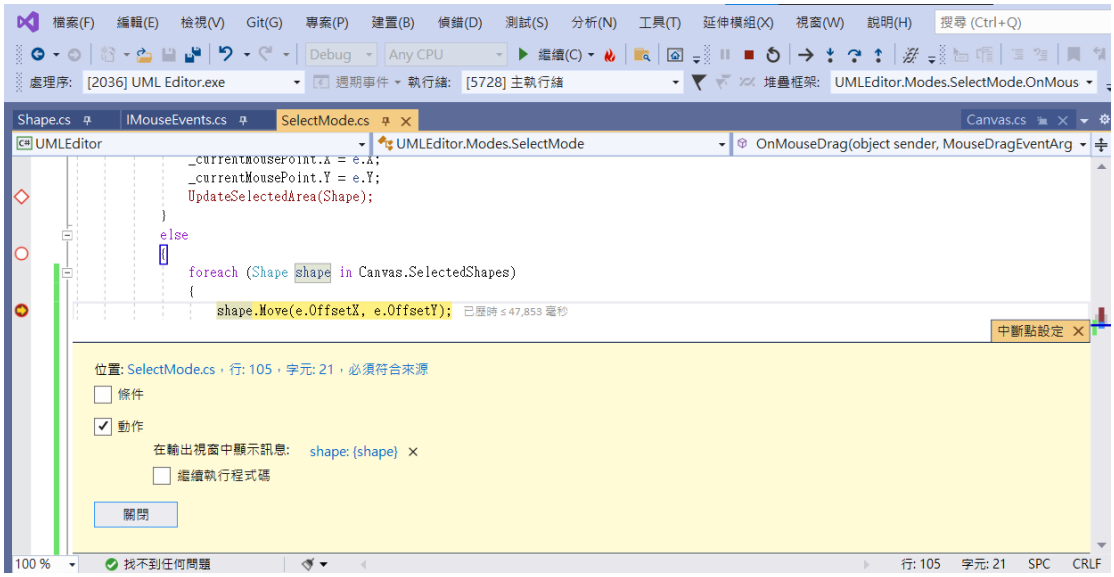


圖 15

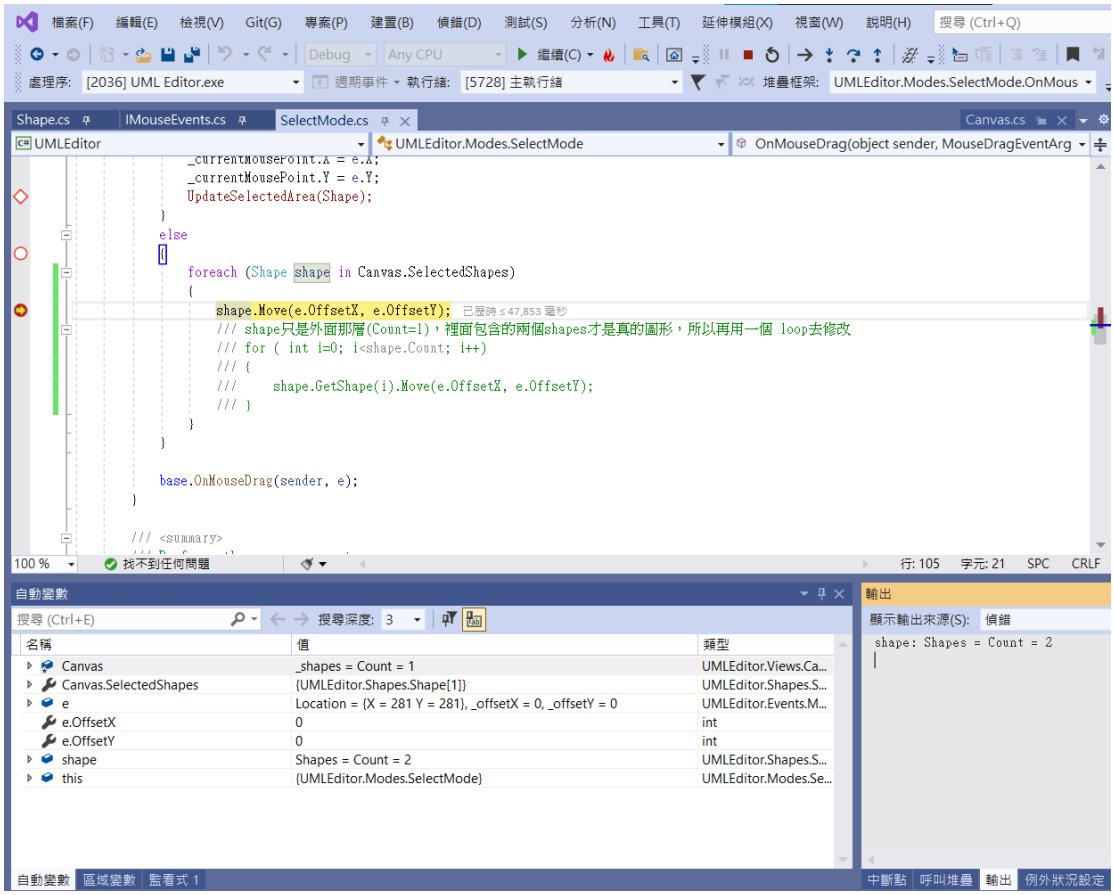


圖 16

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)

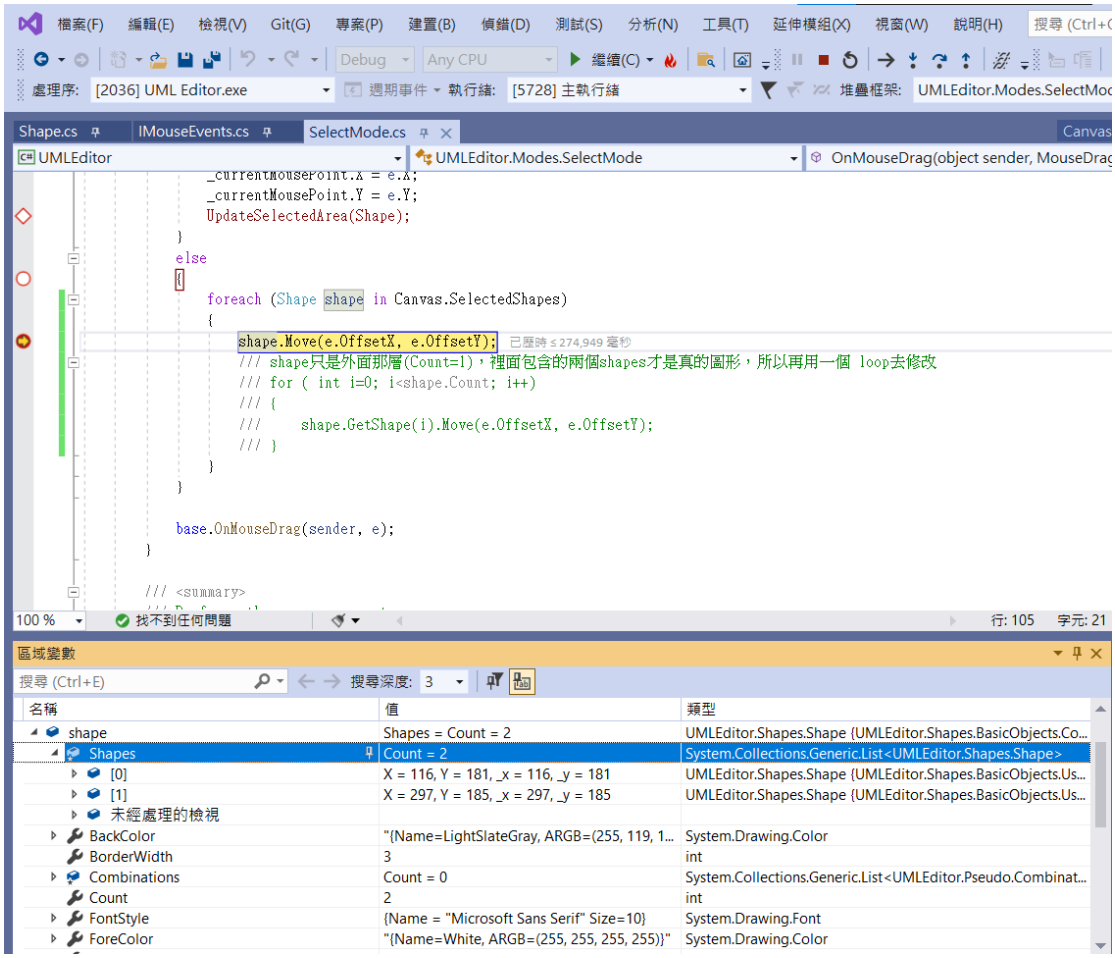


圖 17

因此修正需要加上一層迴圈，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

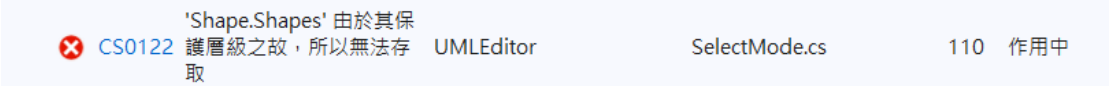


圖 18

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

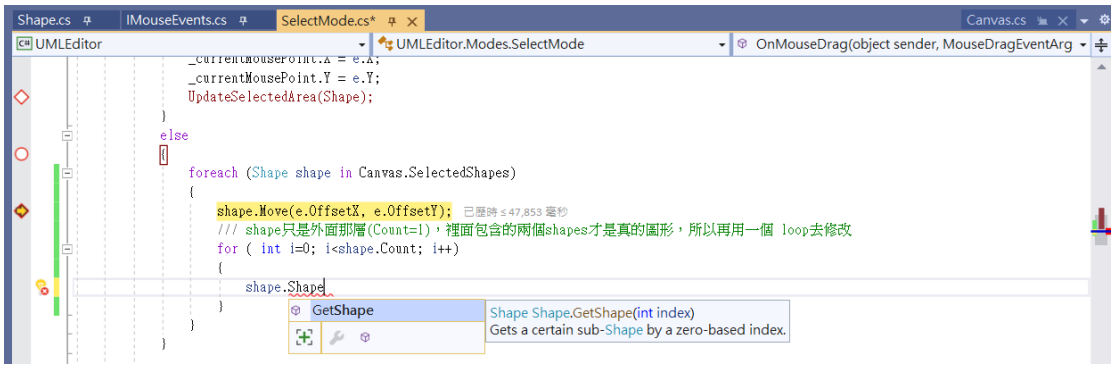


圖 19

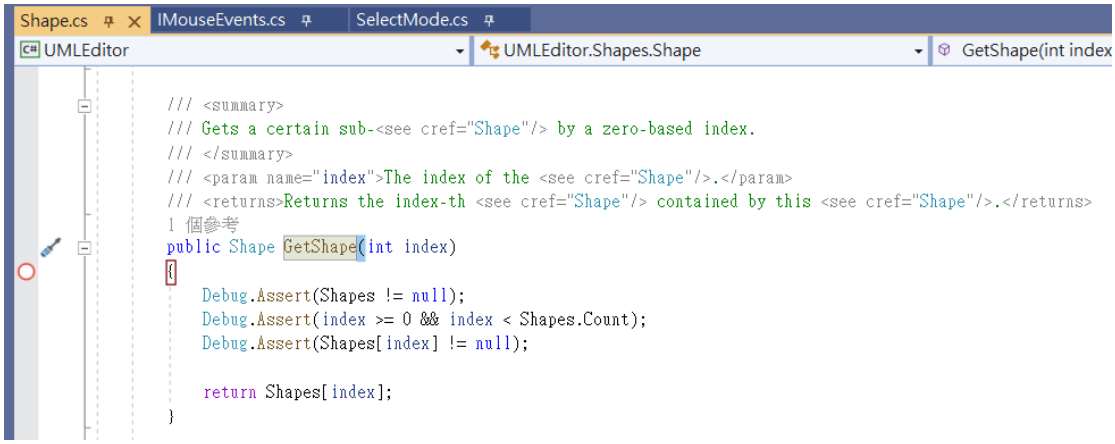


圖 20

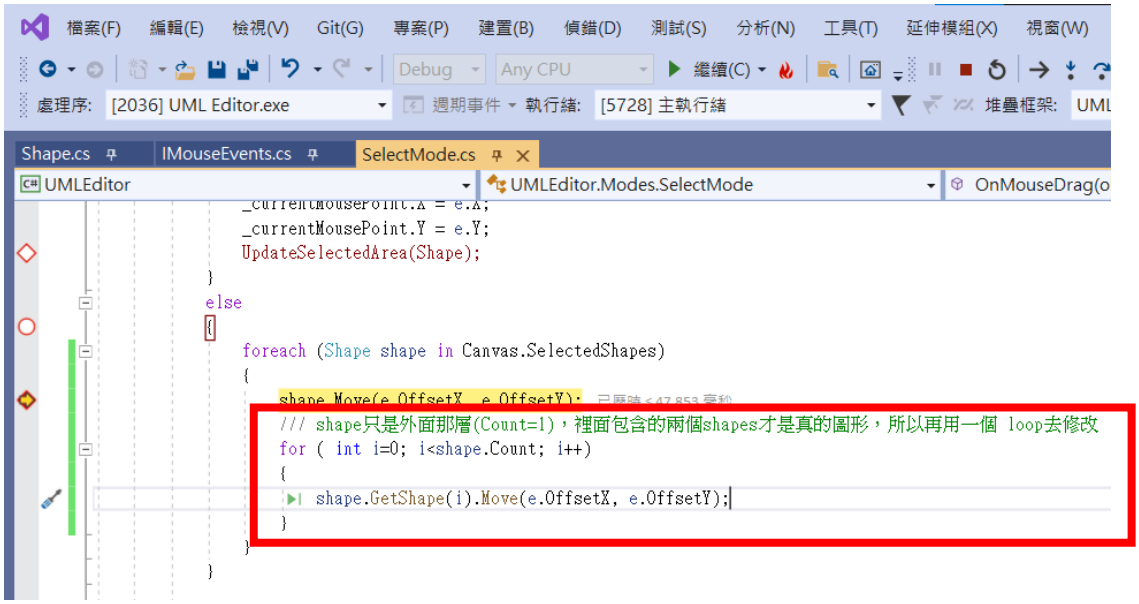


圖 21

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)**

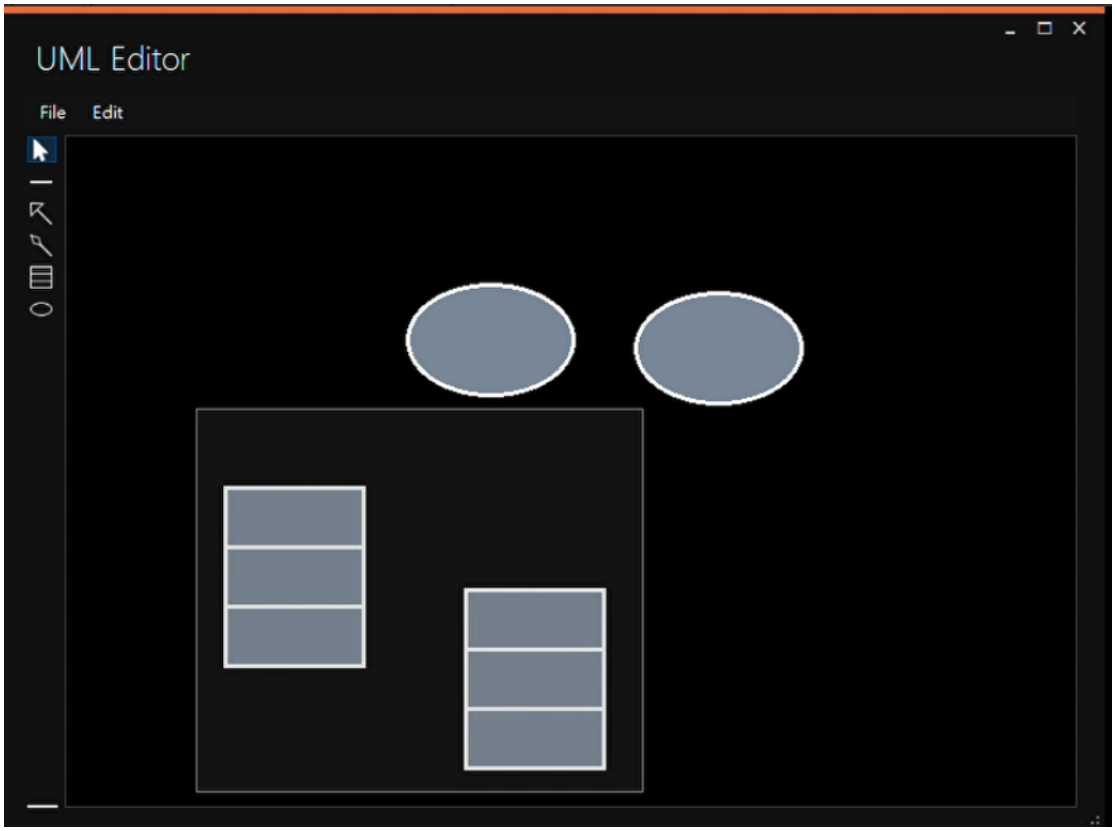


圖 22

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)

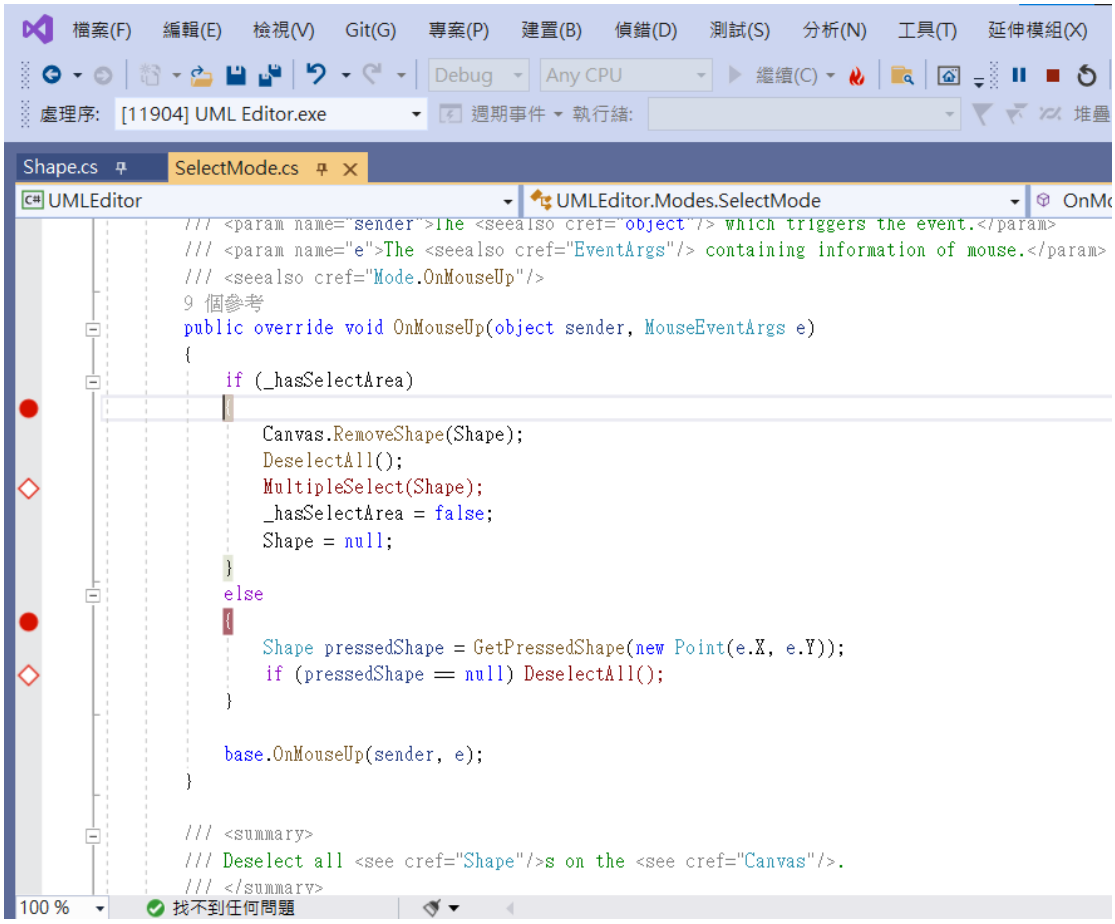


圖 23

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)

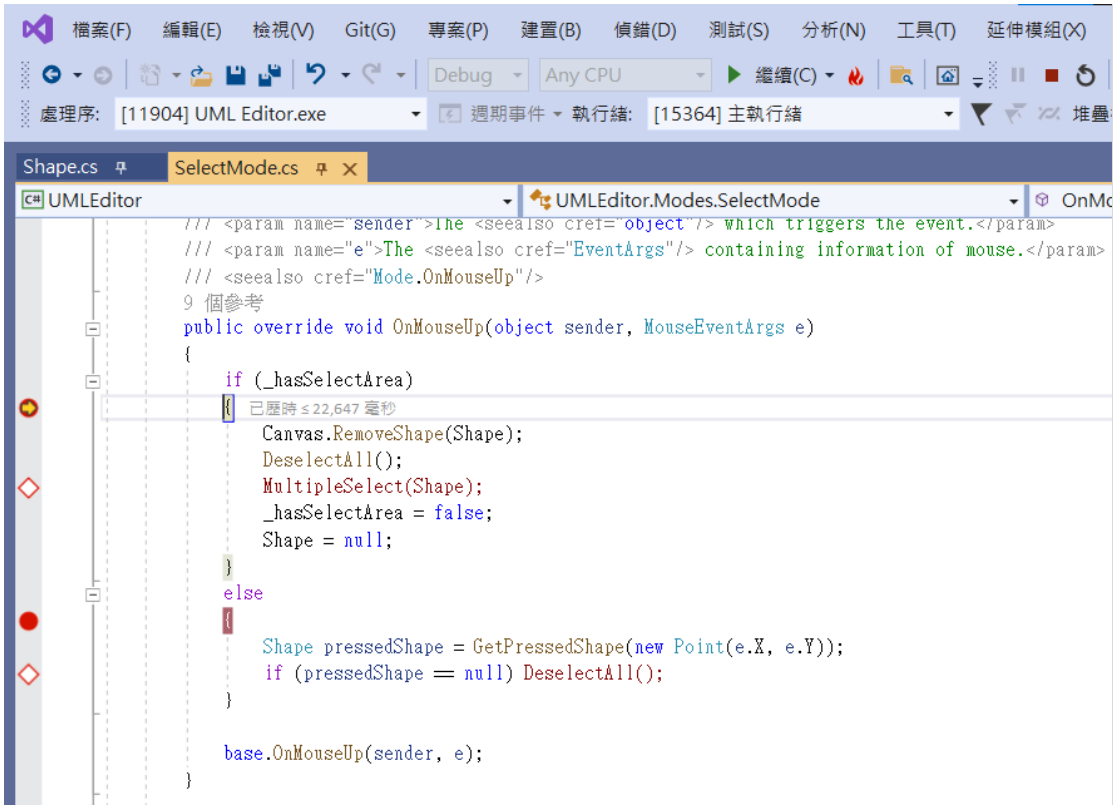


圖 24

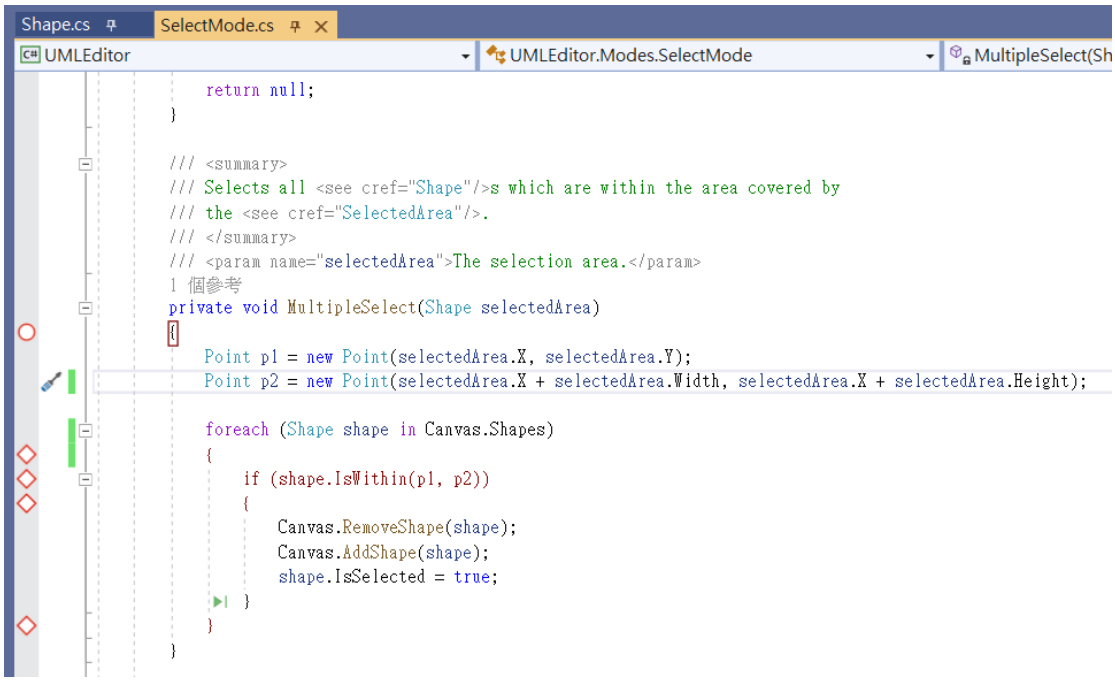


圖 25

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)

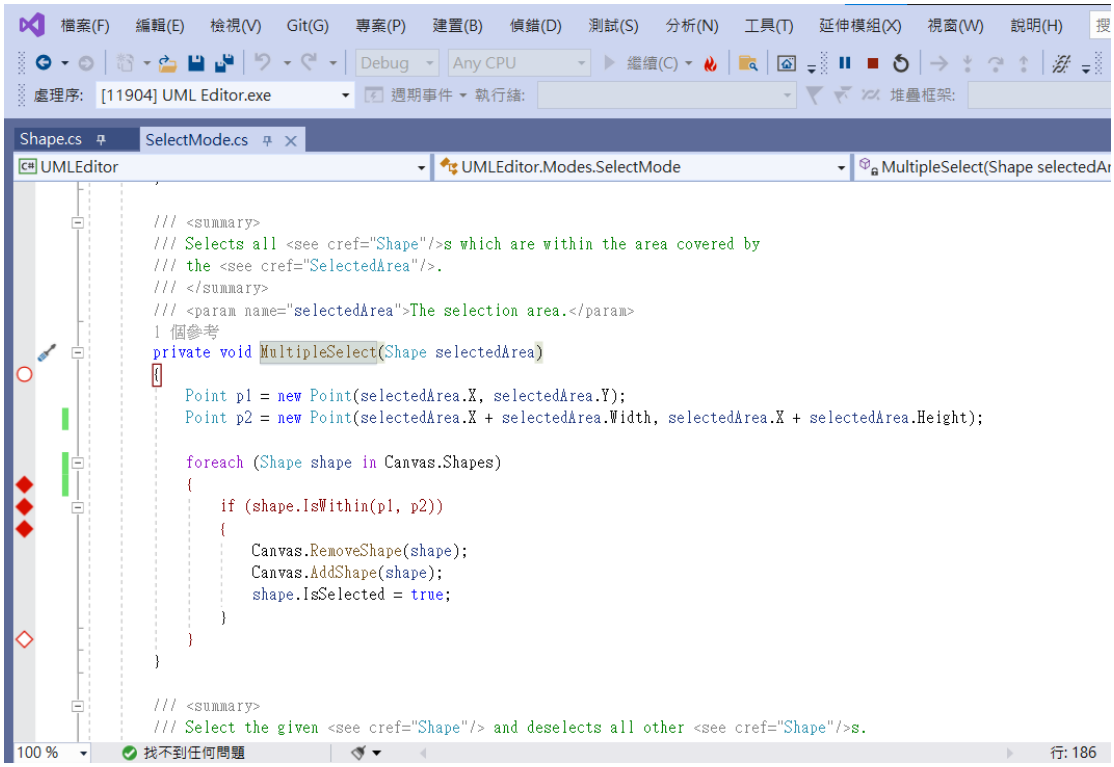


圖 26

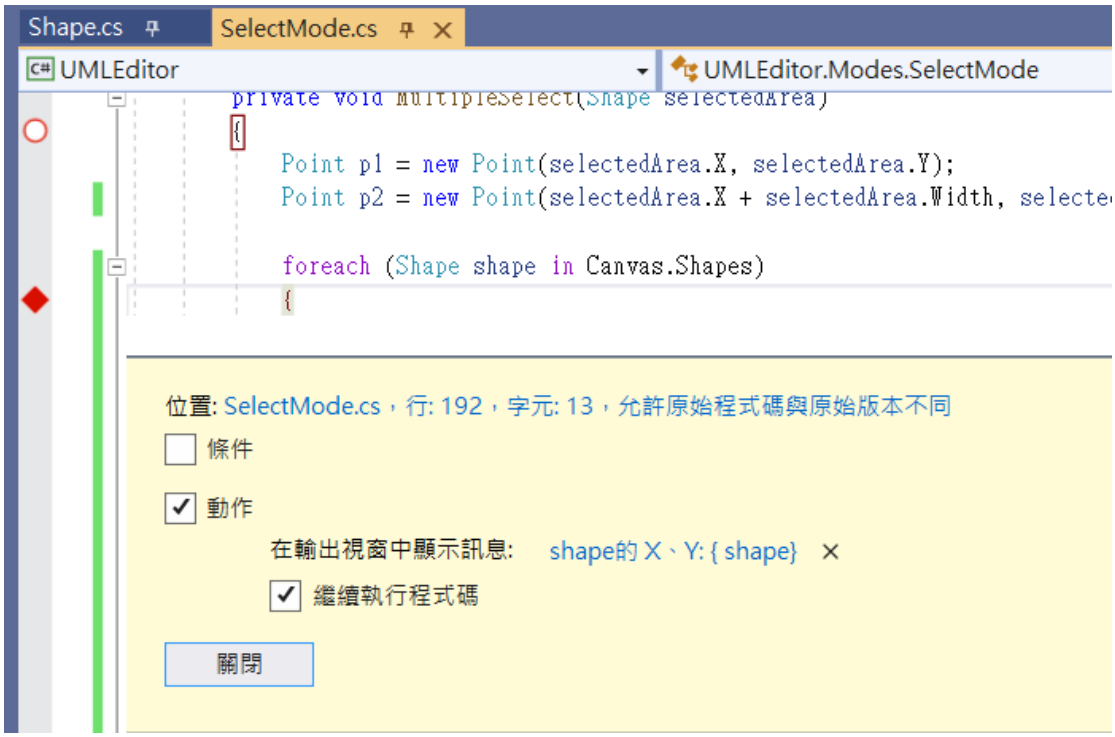


圖 27

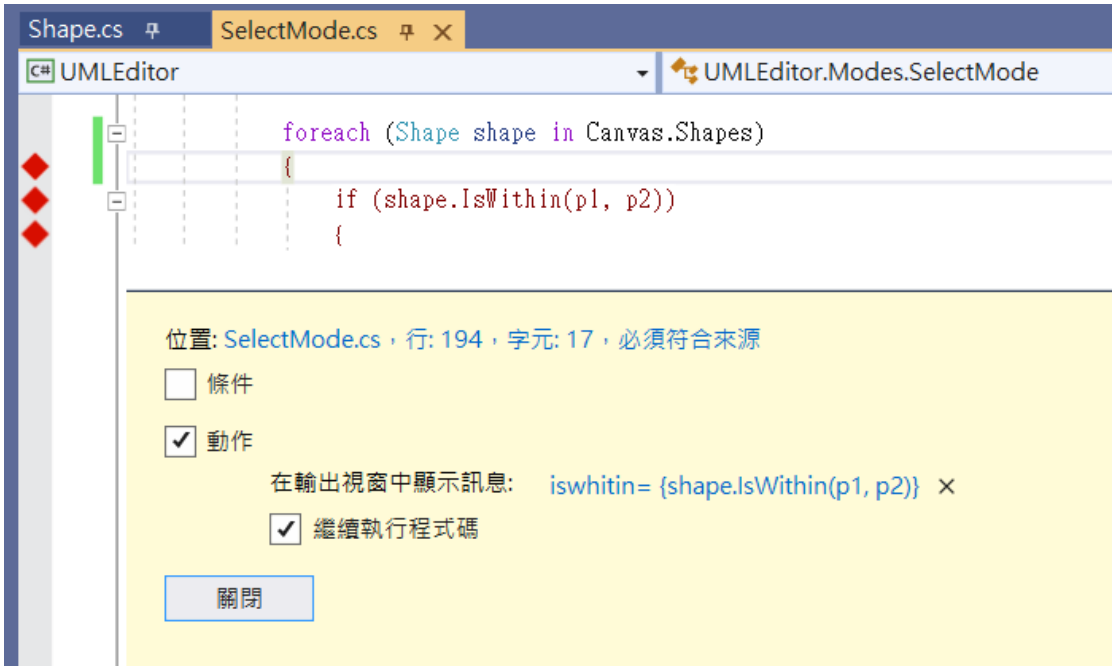
Shape.cs
SelectMode.cs
UMLEditor
UMLEditor.Modes.SelectMode
foreach (Shape shape in Canvas.Shapes)
{
if (shape.IsWithin(p1, p2))
{
位置: SelectMode.cs，行: 194，字元: 17，必須符合來源
條件
動作
在輸出視窗中顯示訊息:
iswhitin= {shape.IsWithin(p1, p2)}
繼續執行程式碼
關閉

圖 28

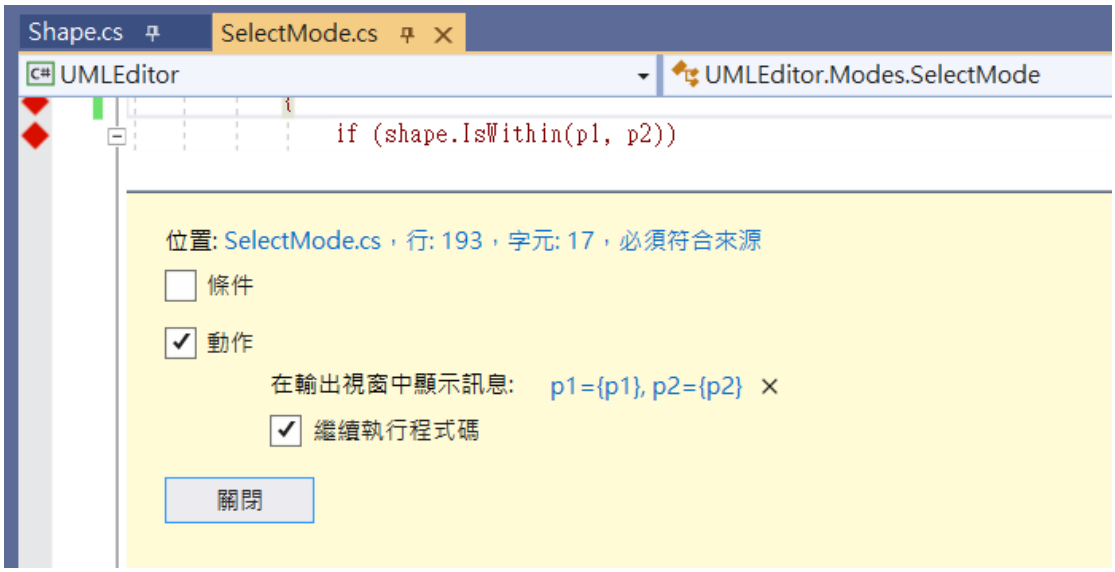
Shape.cs
SelectMode.cs
UMLEditor
UMLEditor.Modes.SelectMode
{
if (shape.IsWithin(p1, p2))
位置: SelectMode.cs，行: 193，字元: 17，必須符合來源
條件
動作
在輸出視窗中顯示訊息:
p1={p1}, p2={p2}
繼續執行程式碼
關閉

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)

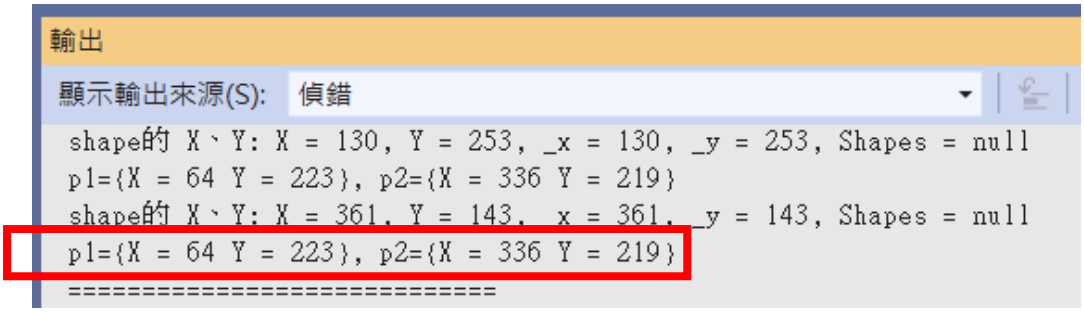


圖 30

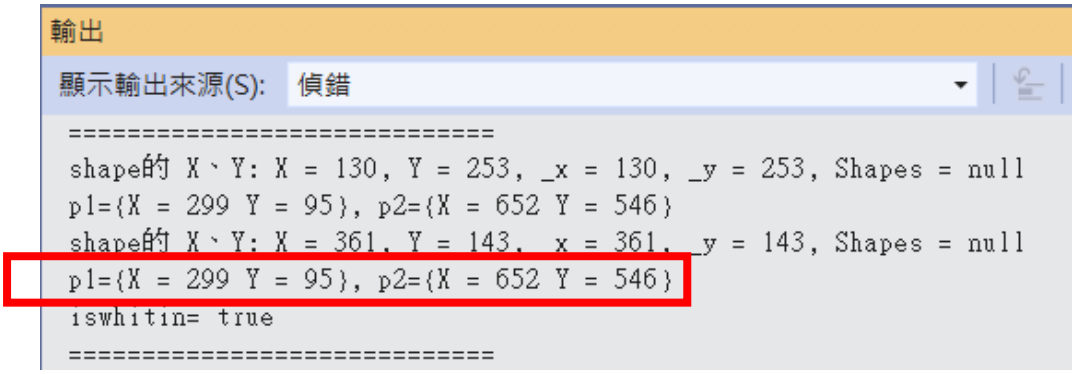


圖 31

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

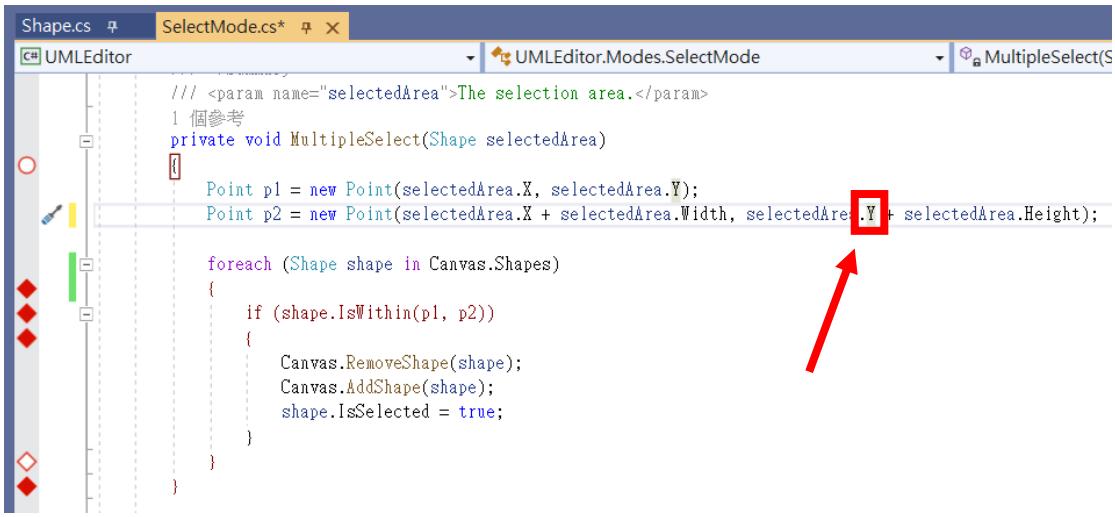


圖 32