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

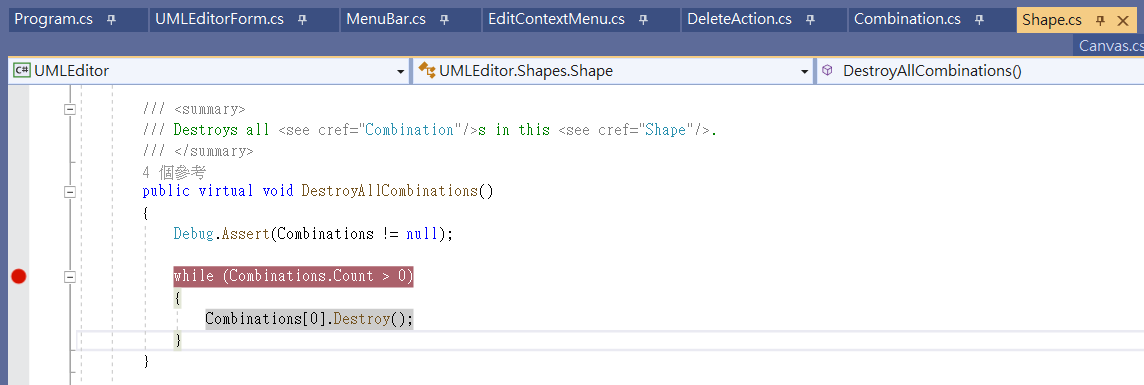


圖2

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

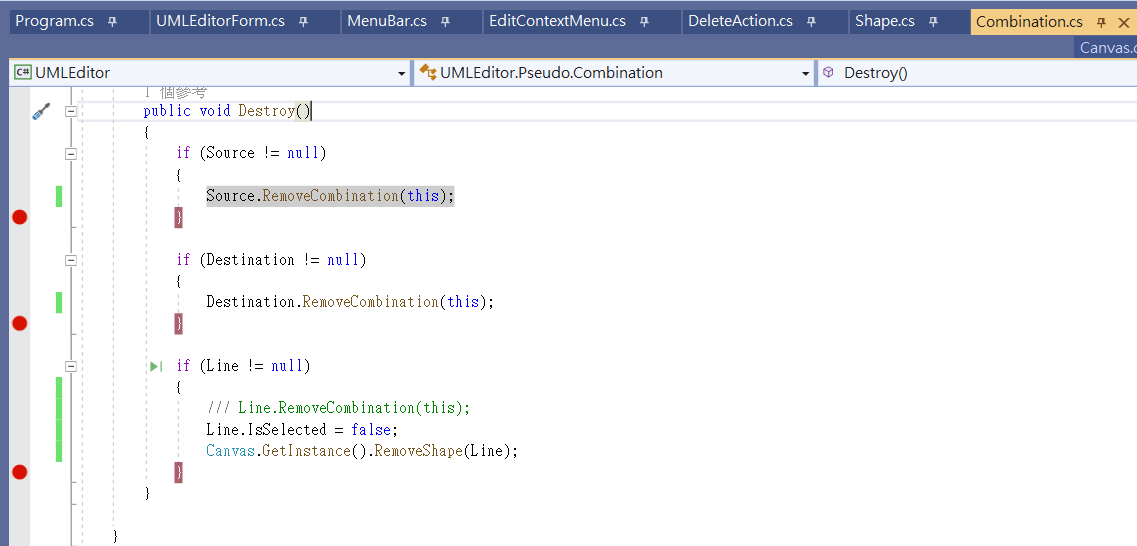


圖3

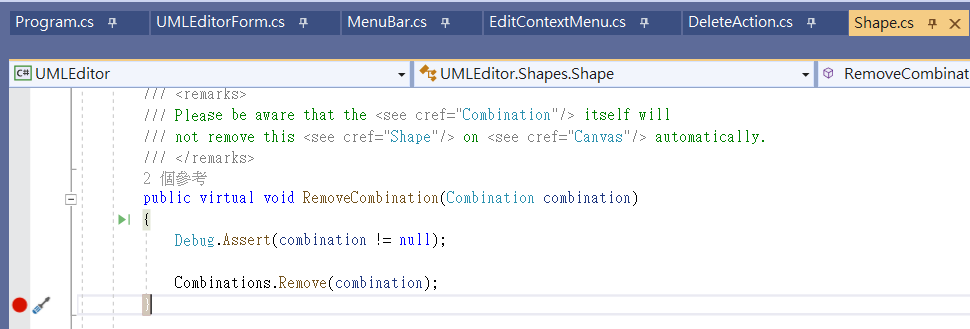


圖4

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)

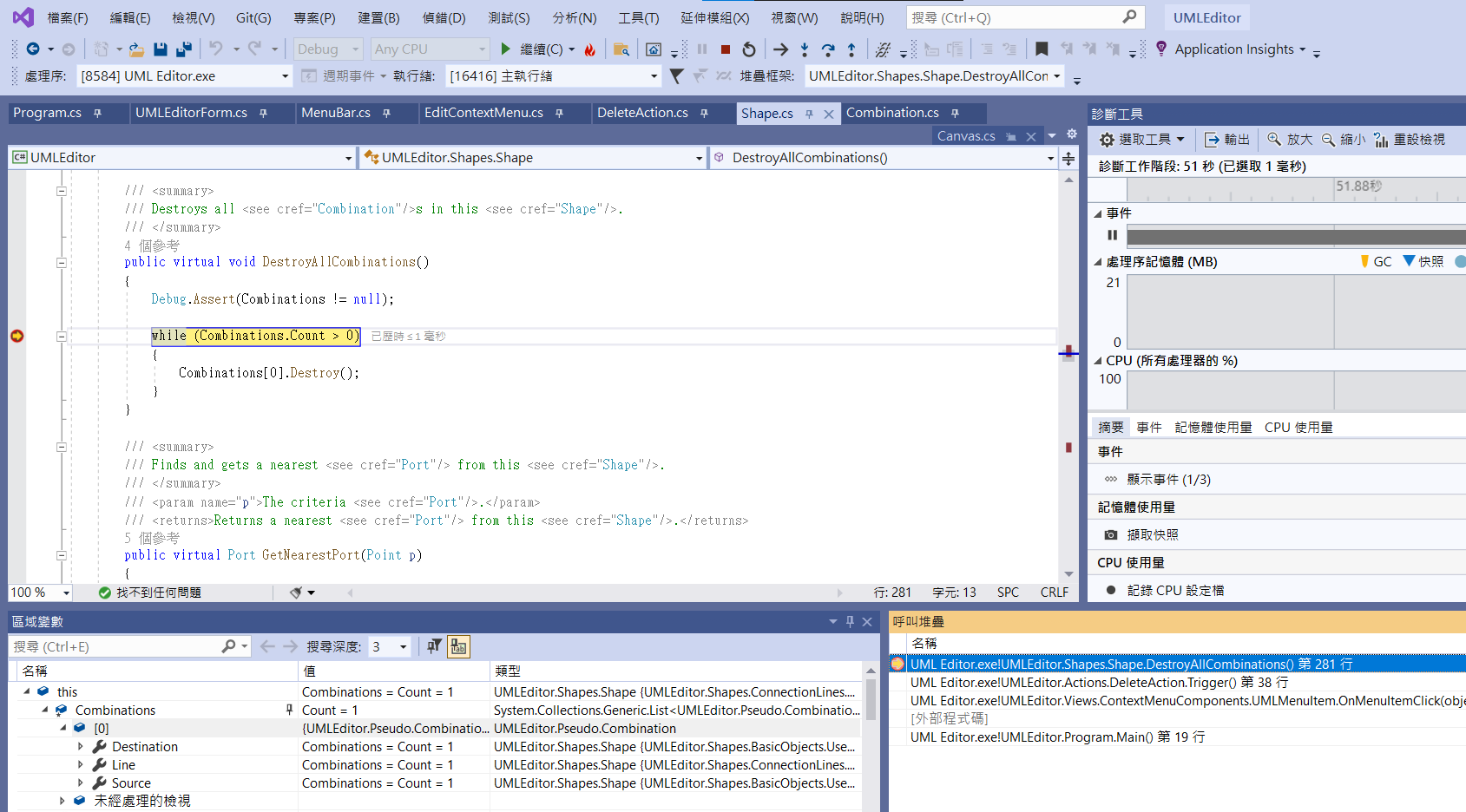


圖5

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)

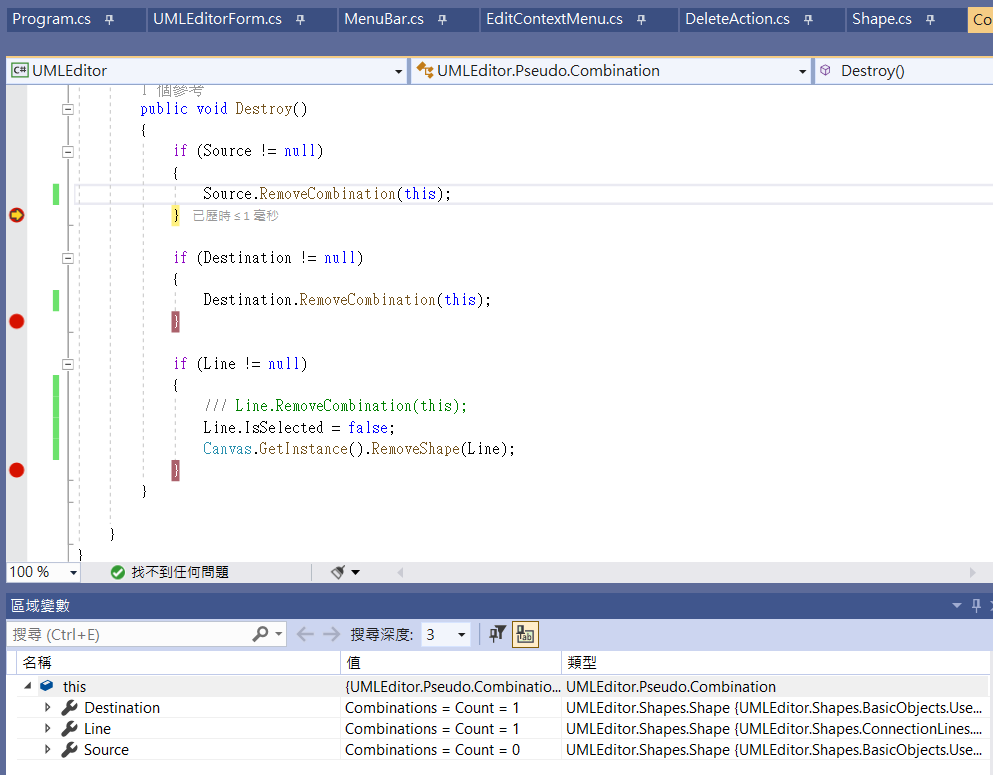


圖6

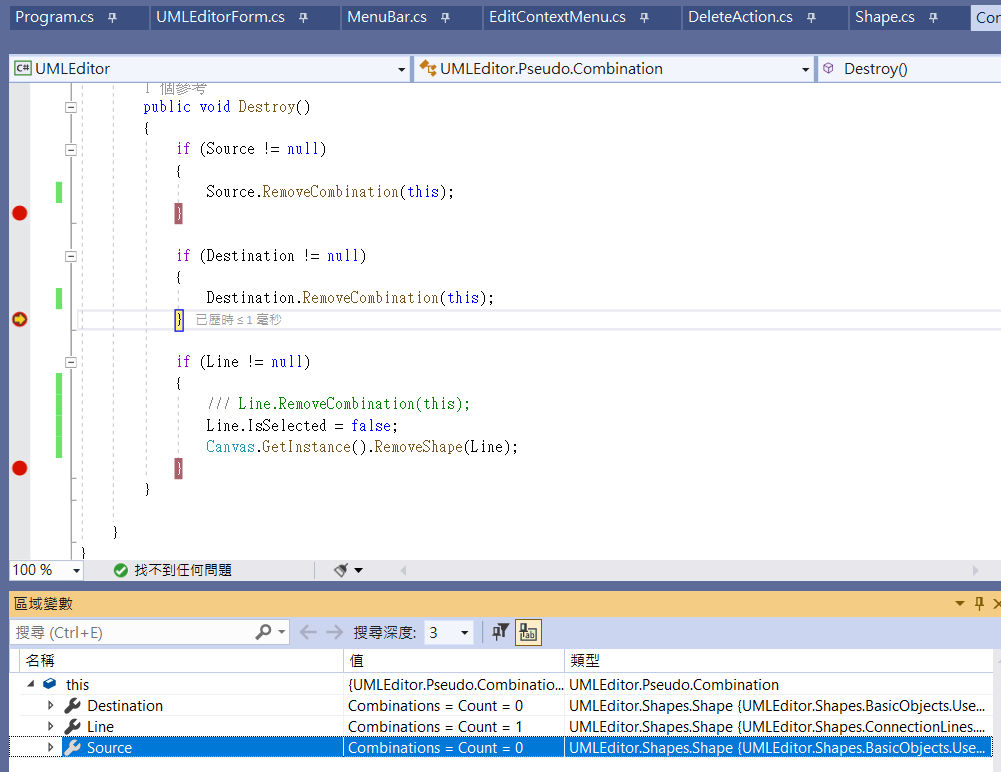


圖7

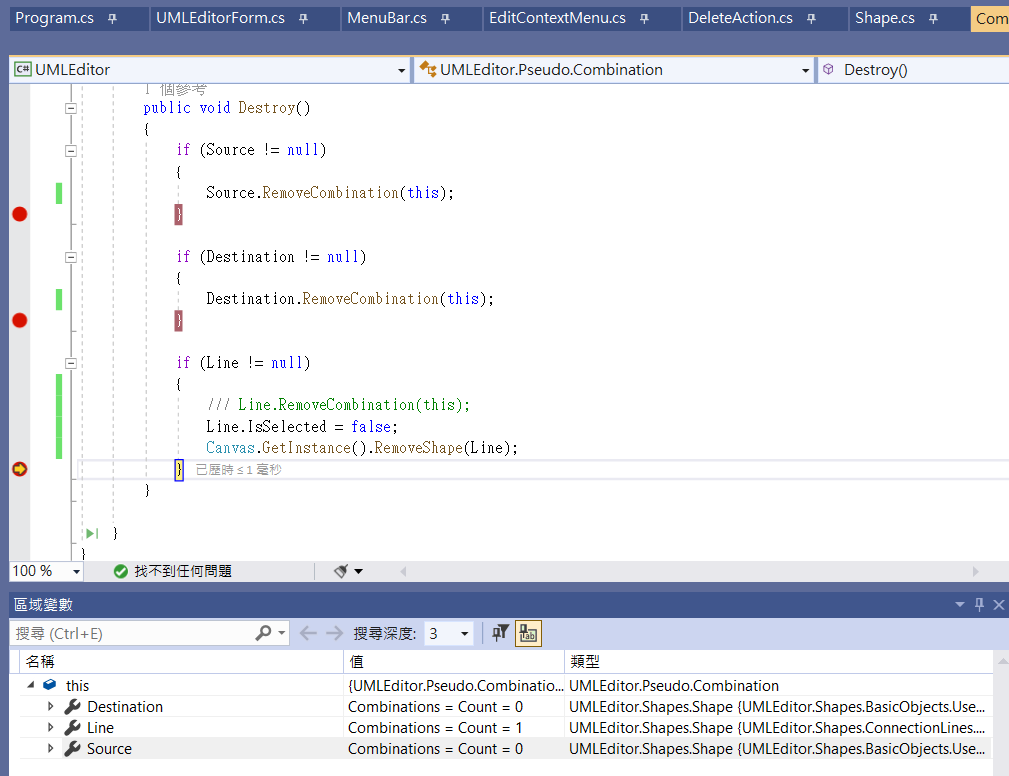


圖8

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

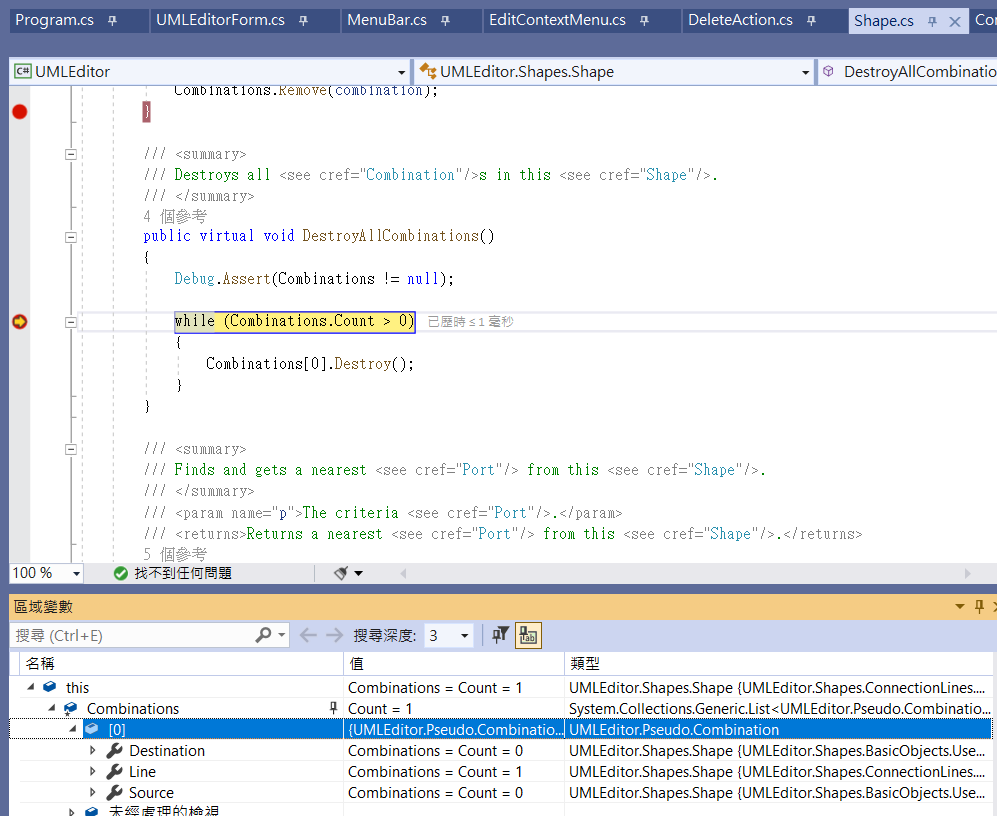


圖9

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

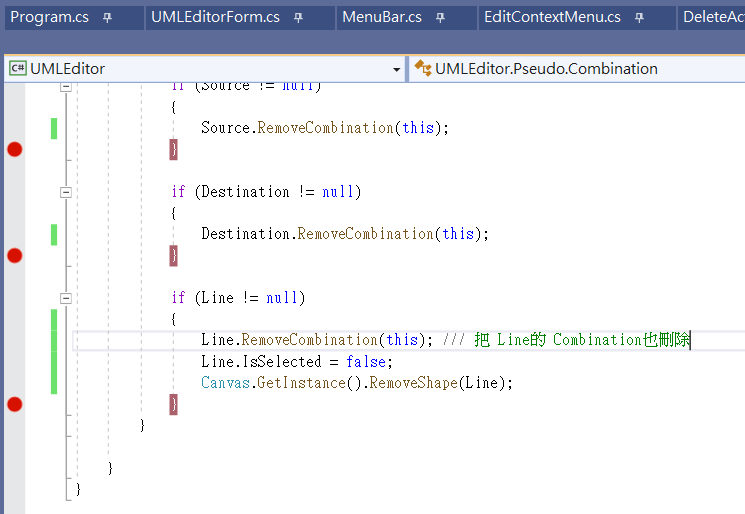


圖10

1. **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)**

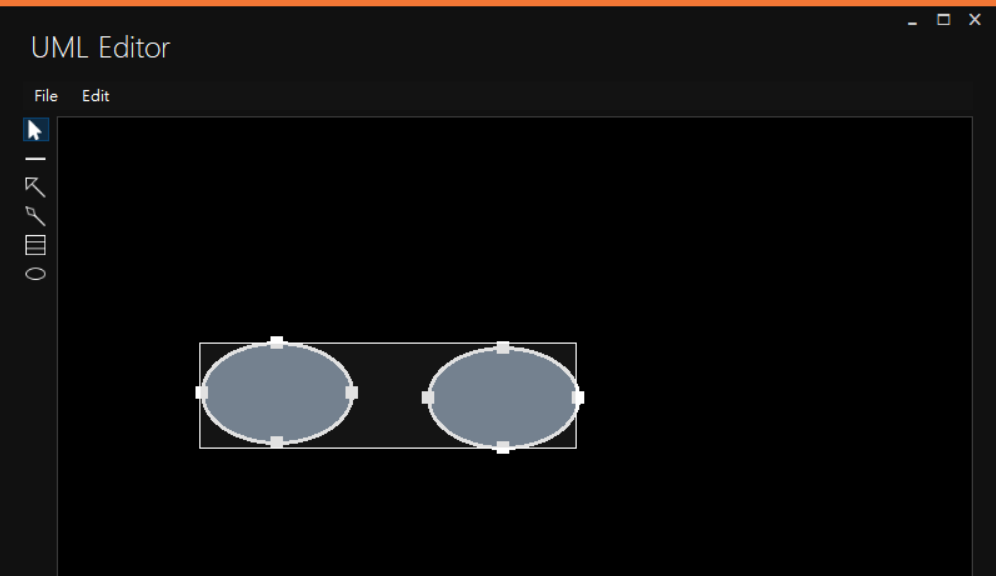


圖11

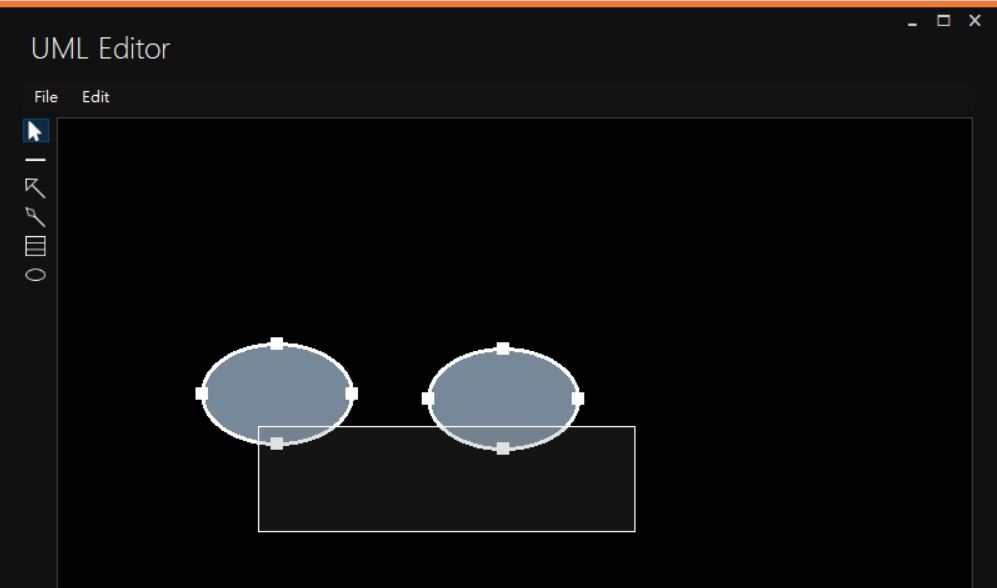


圖12

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)

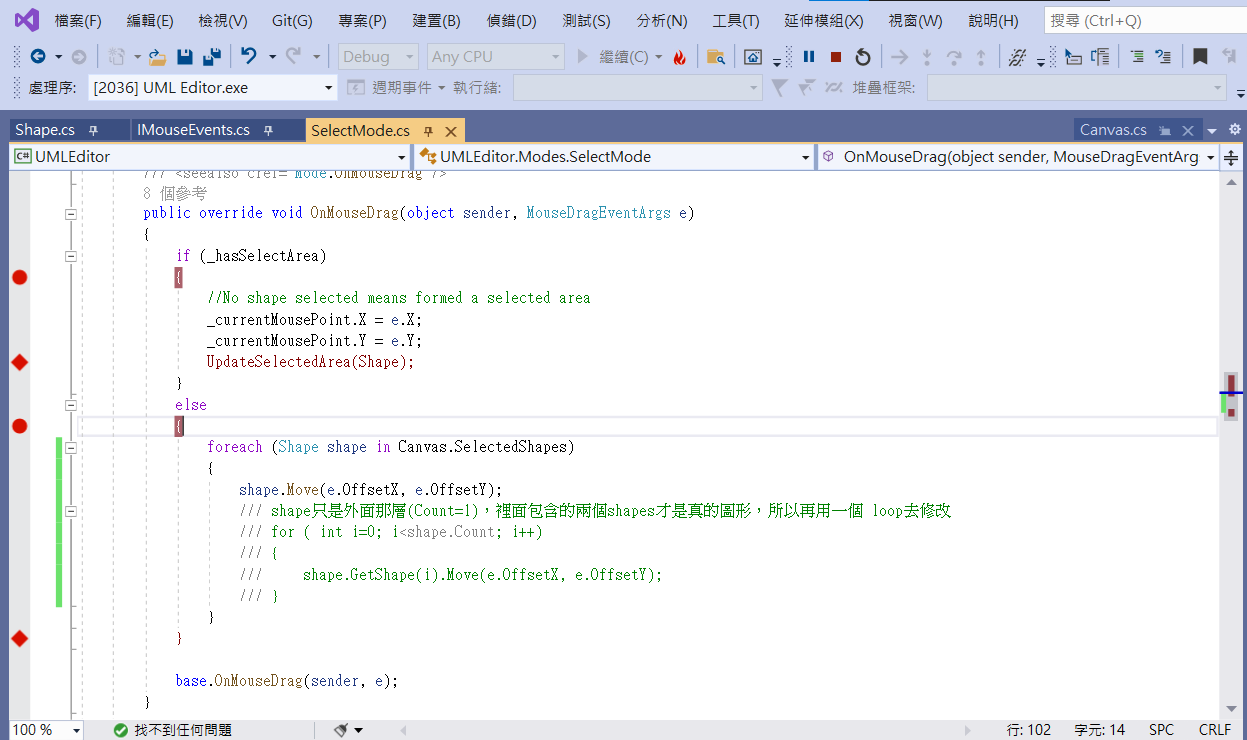


圖13

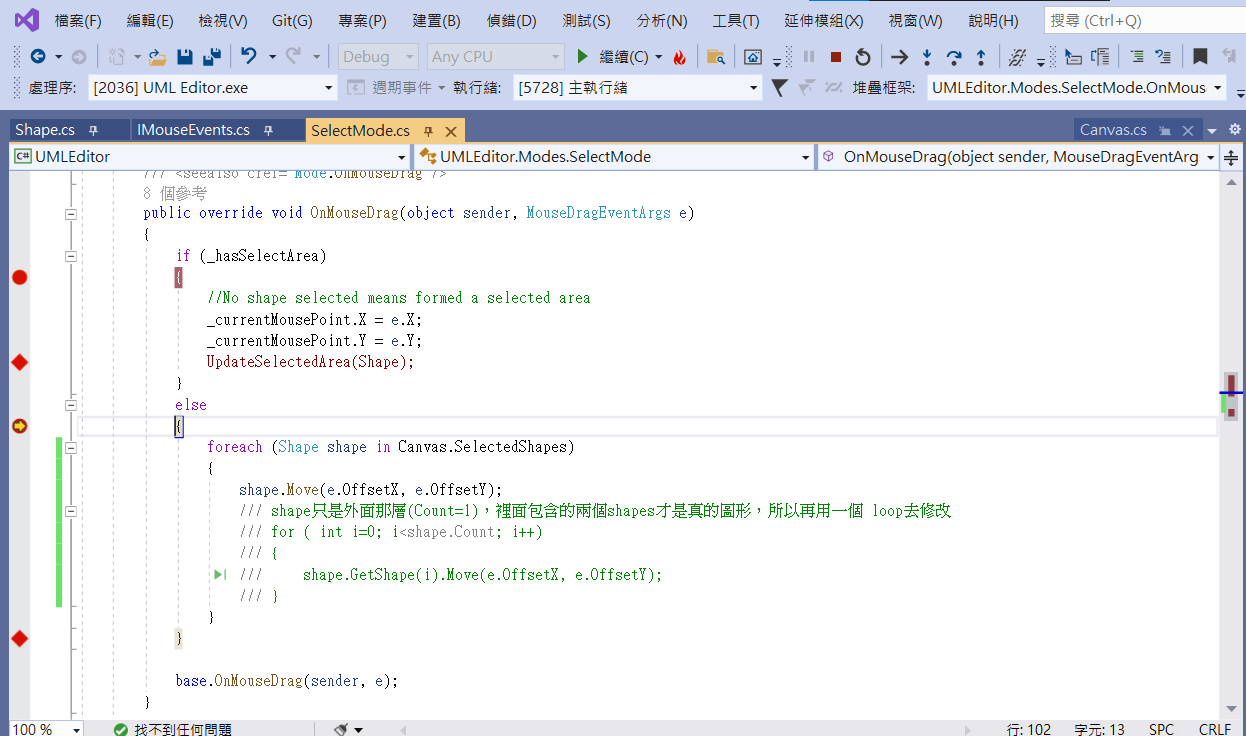


圖14

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)

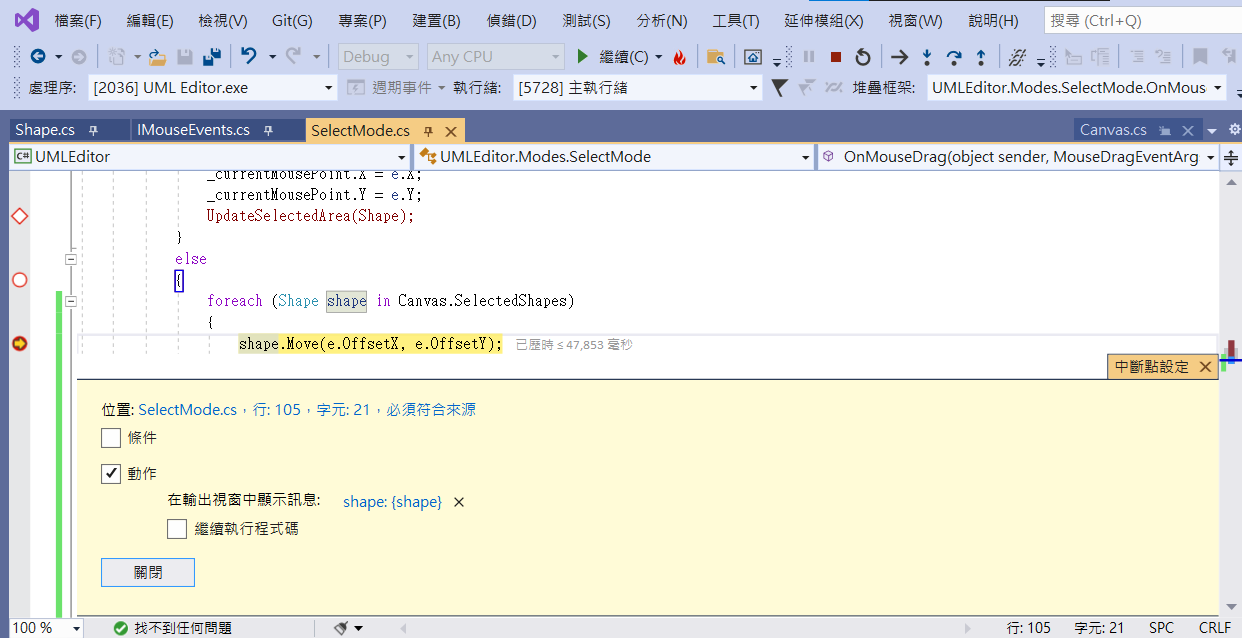


圖15

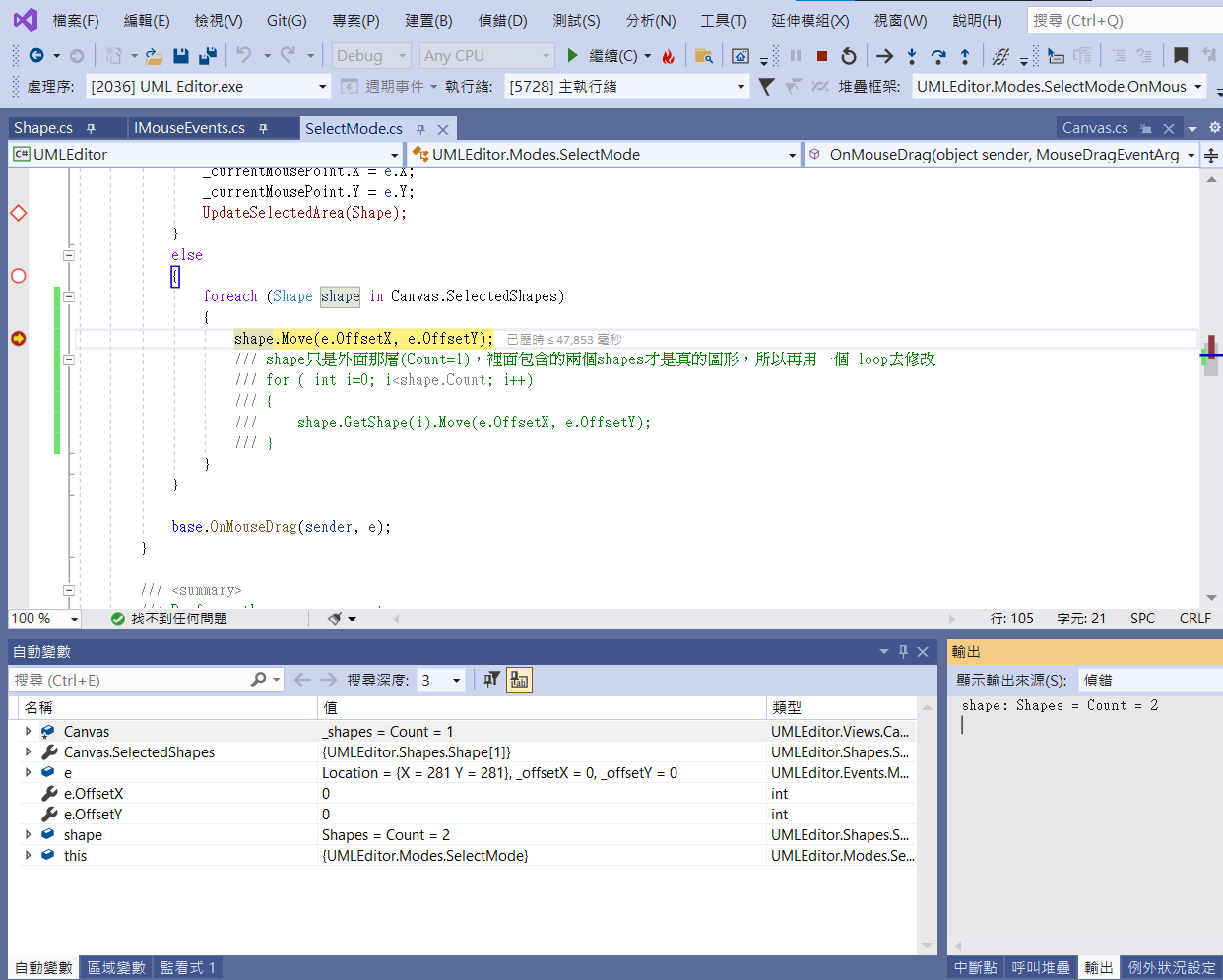


圖16

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)

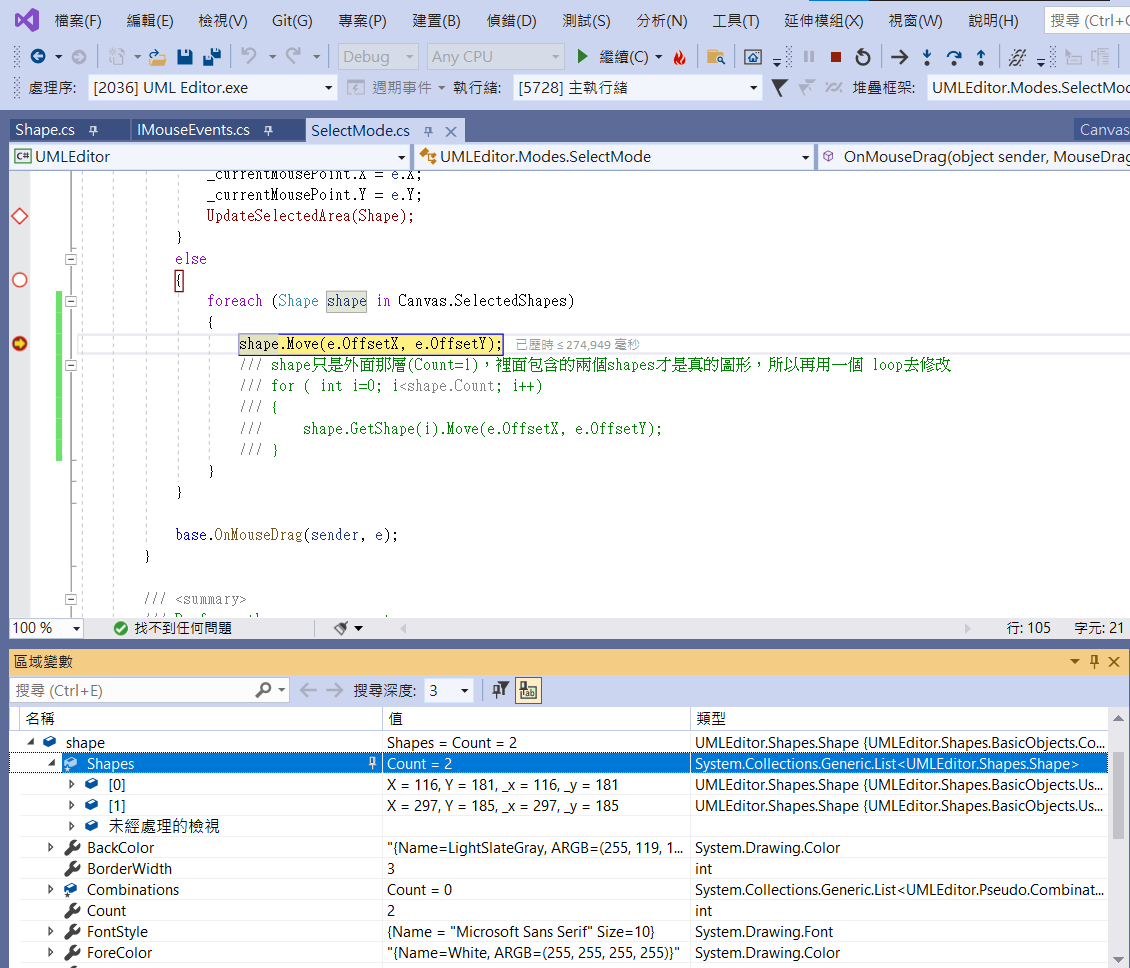


圖17

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

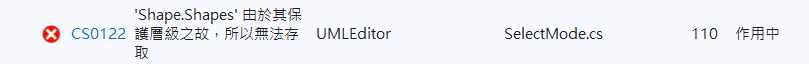


圖18

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

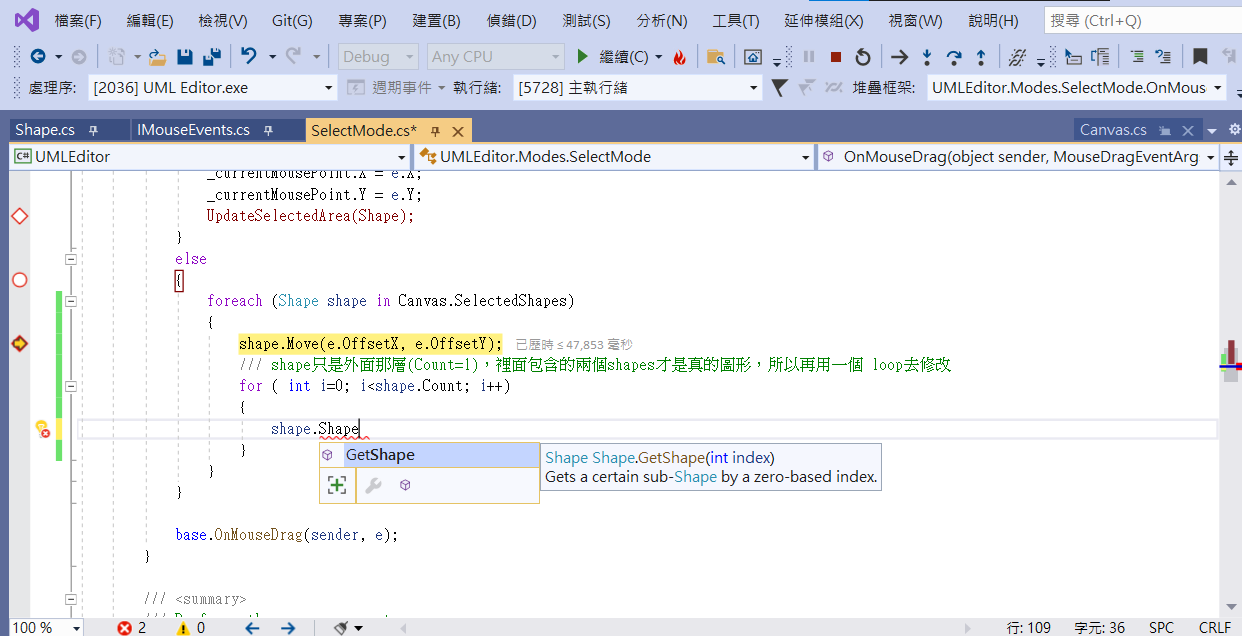


圖19

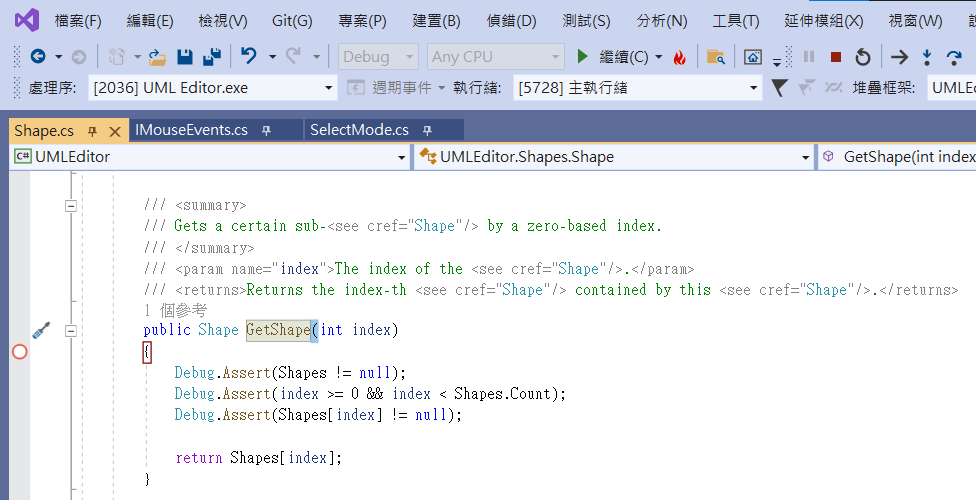


圖20

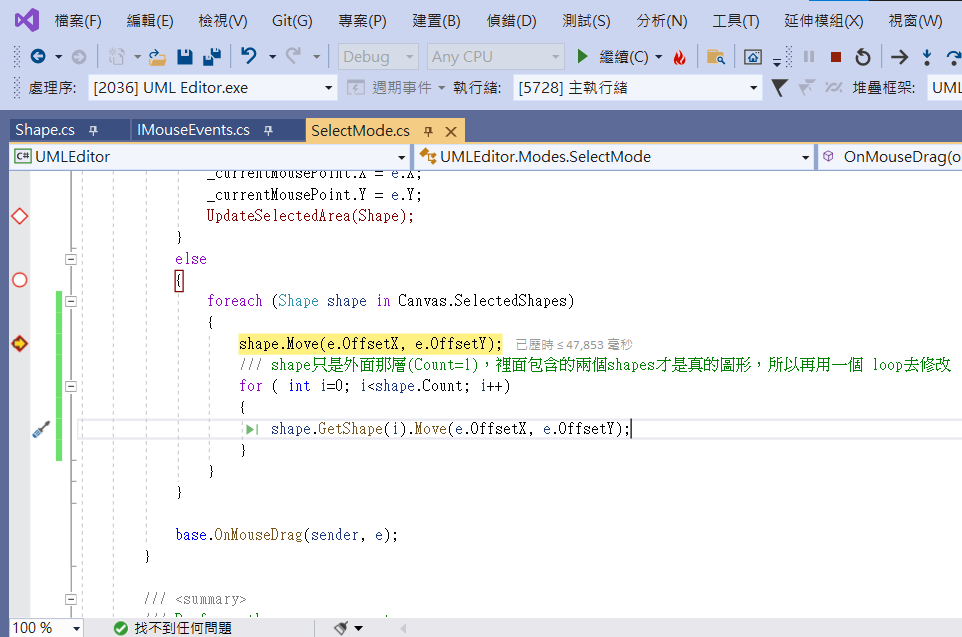


圖21

1. **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)**

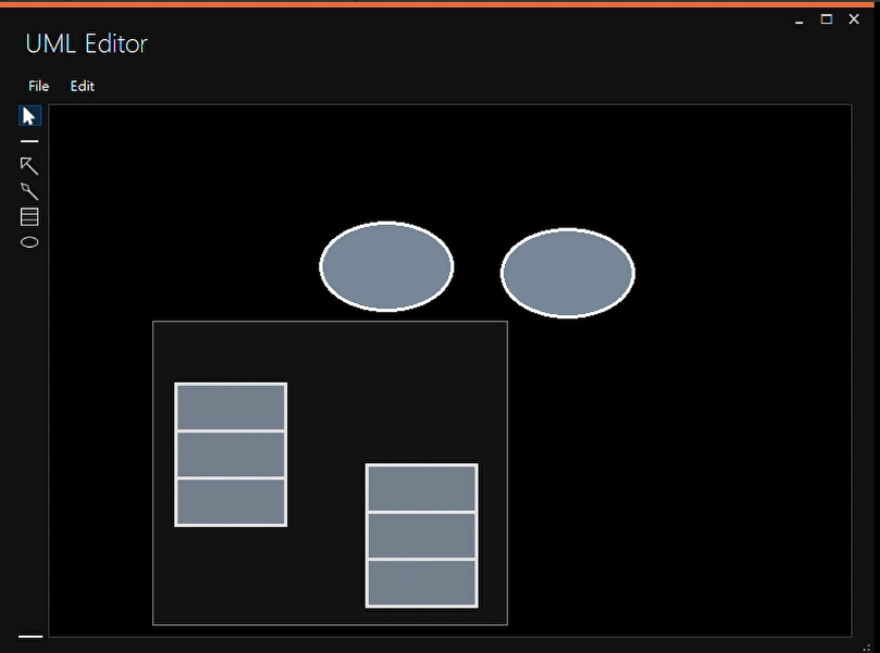


圖22

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)

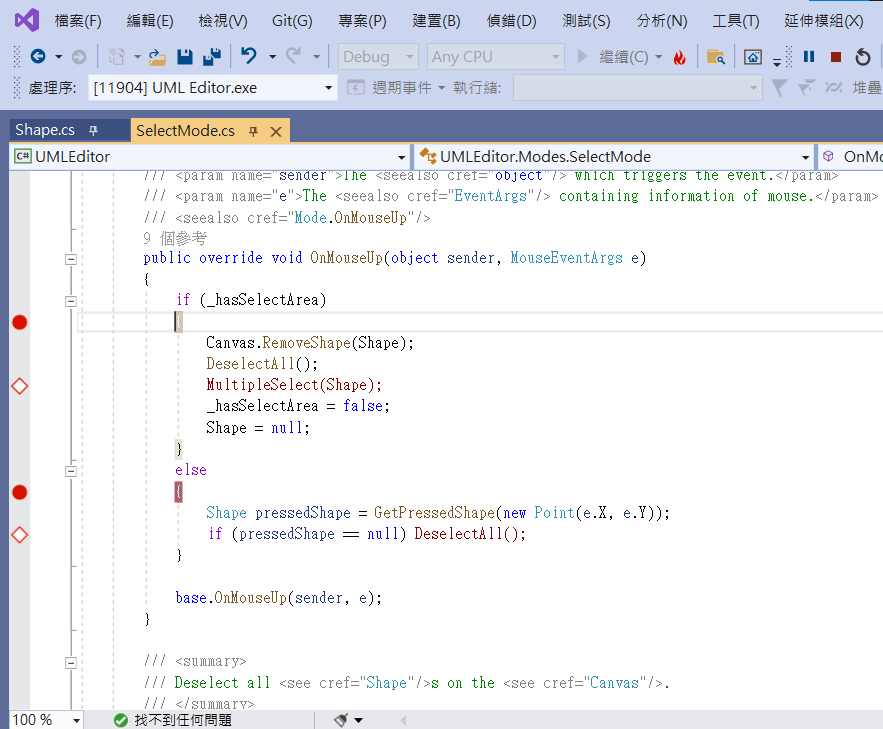


圖23

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)

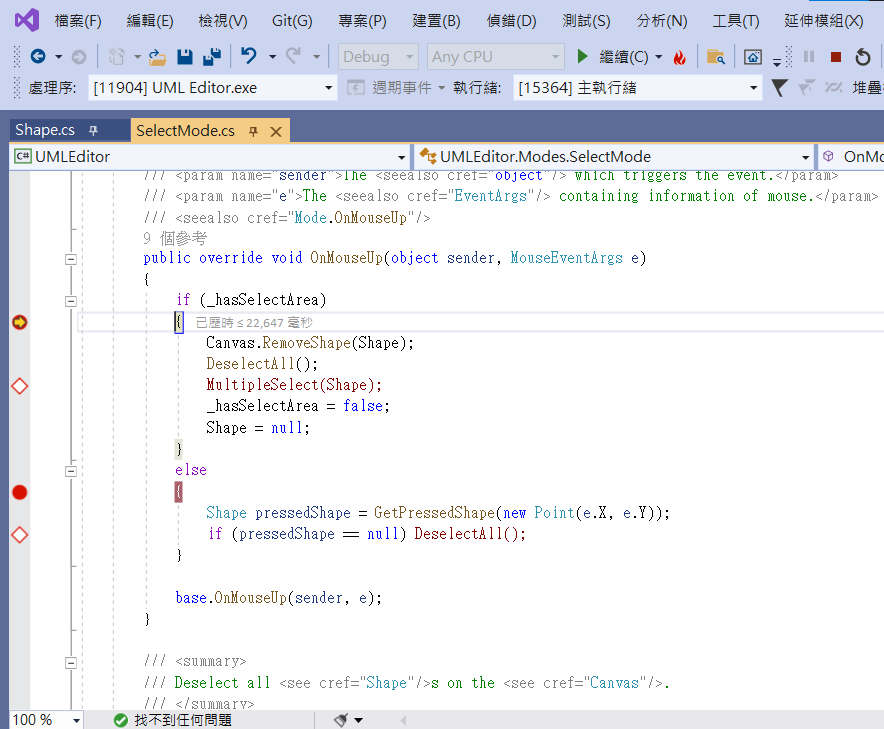


圖24

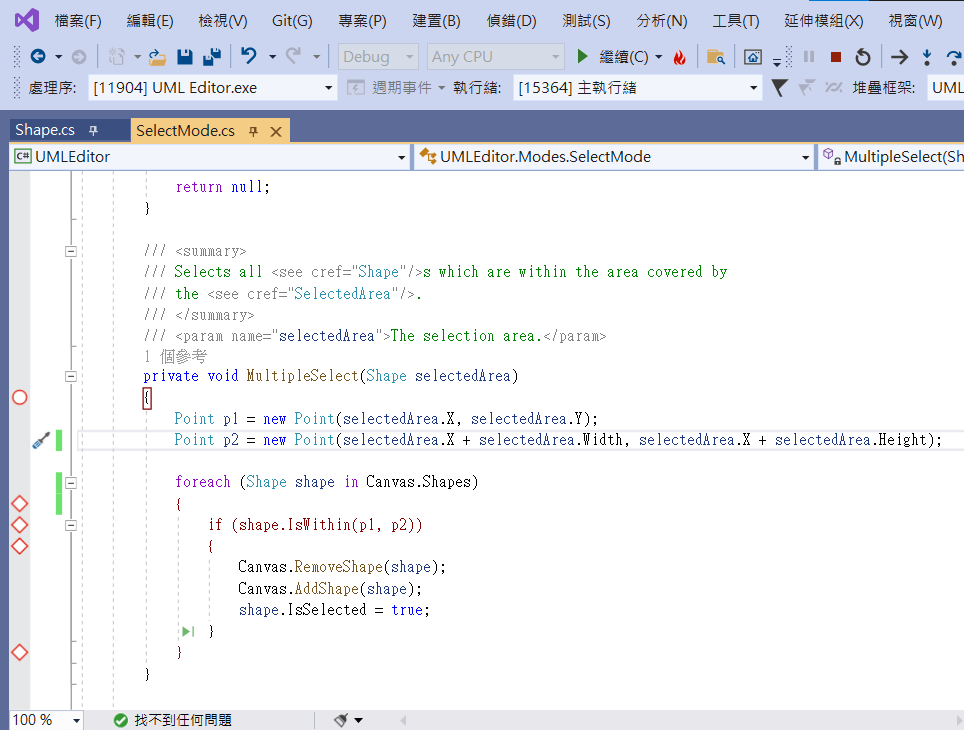


圖25

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)

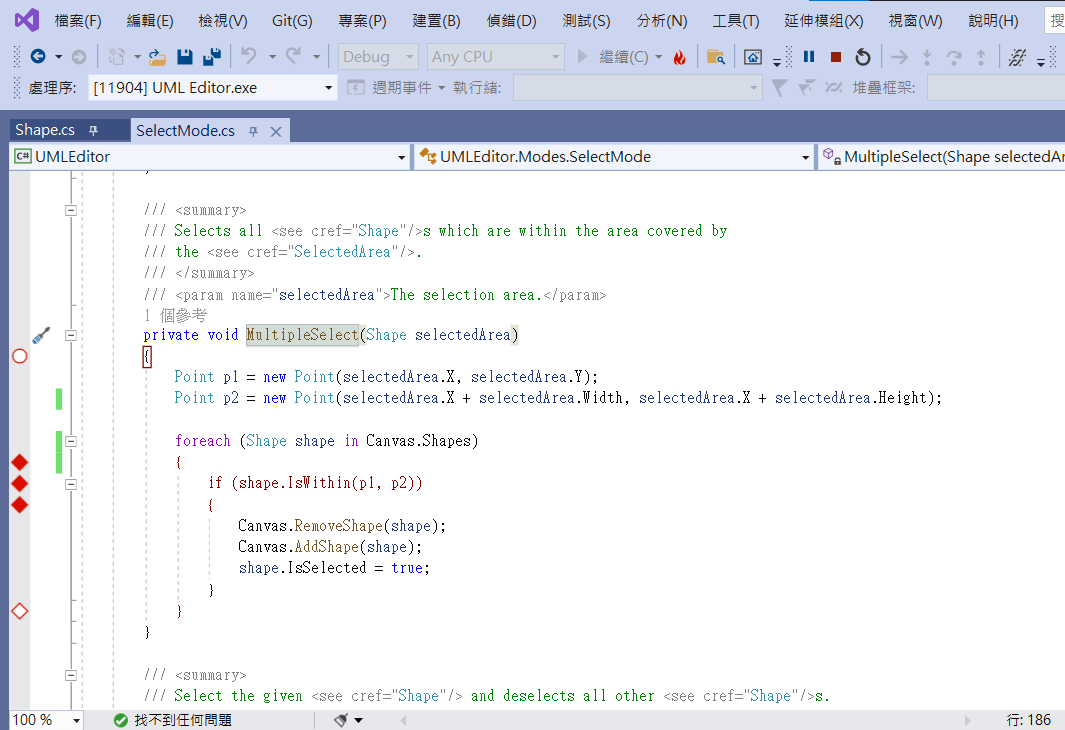


圖26

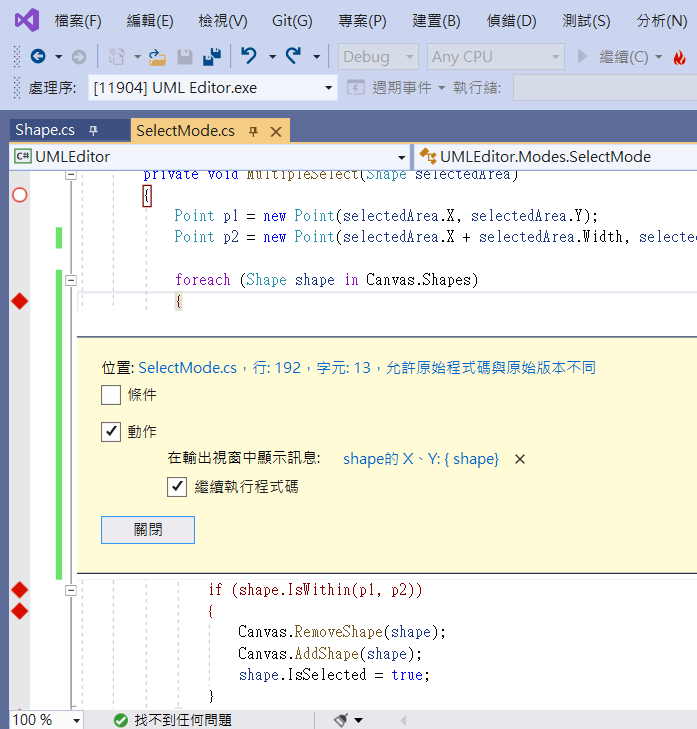


圖27

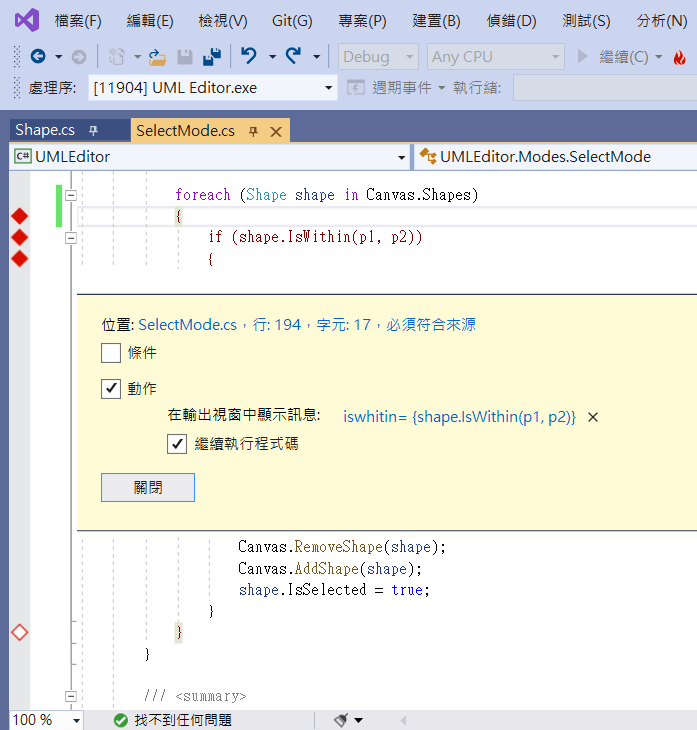


圖28

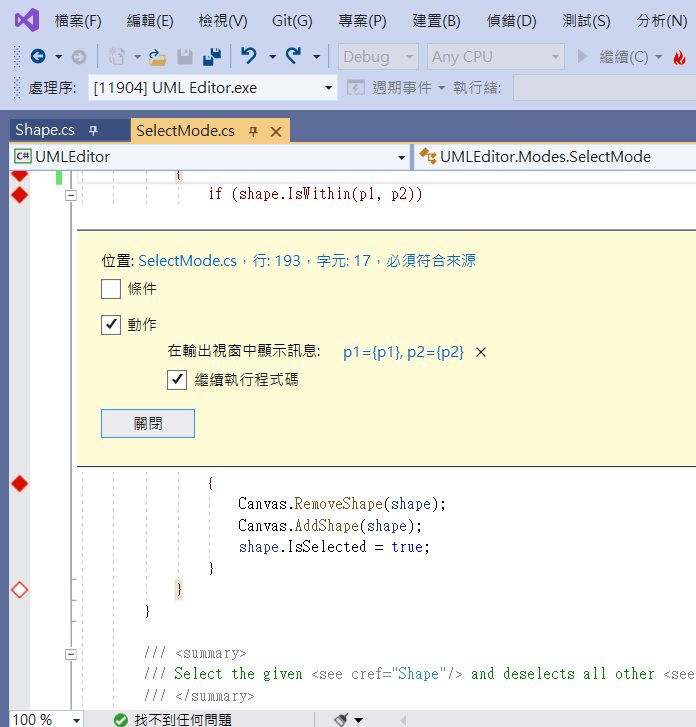


圖29

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)

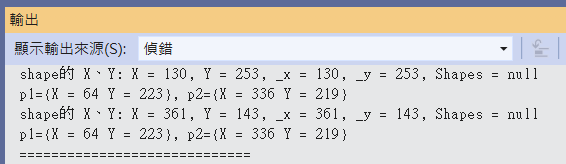


圖30

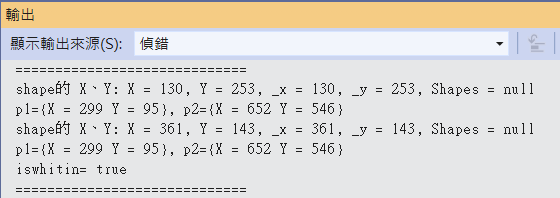


圖31

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

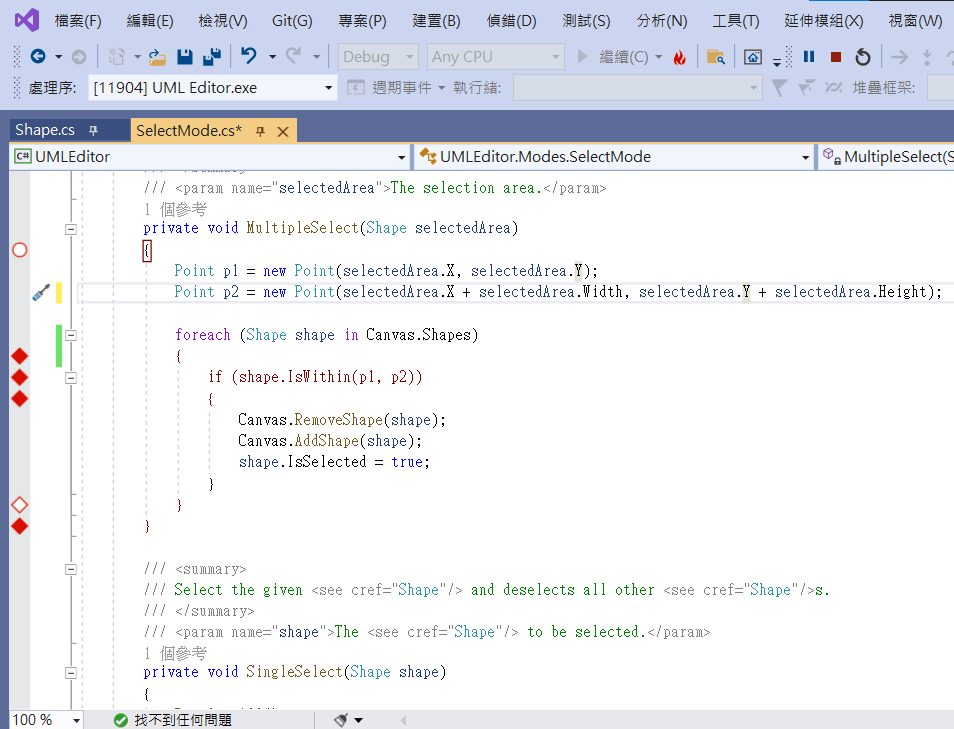


圖32