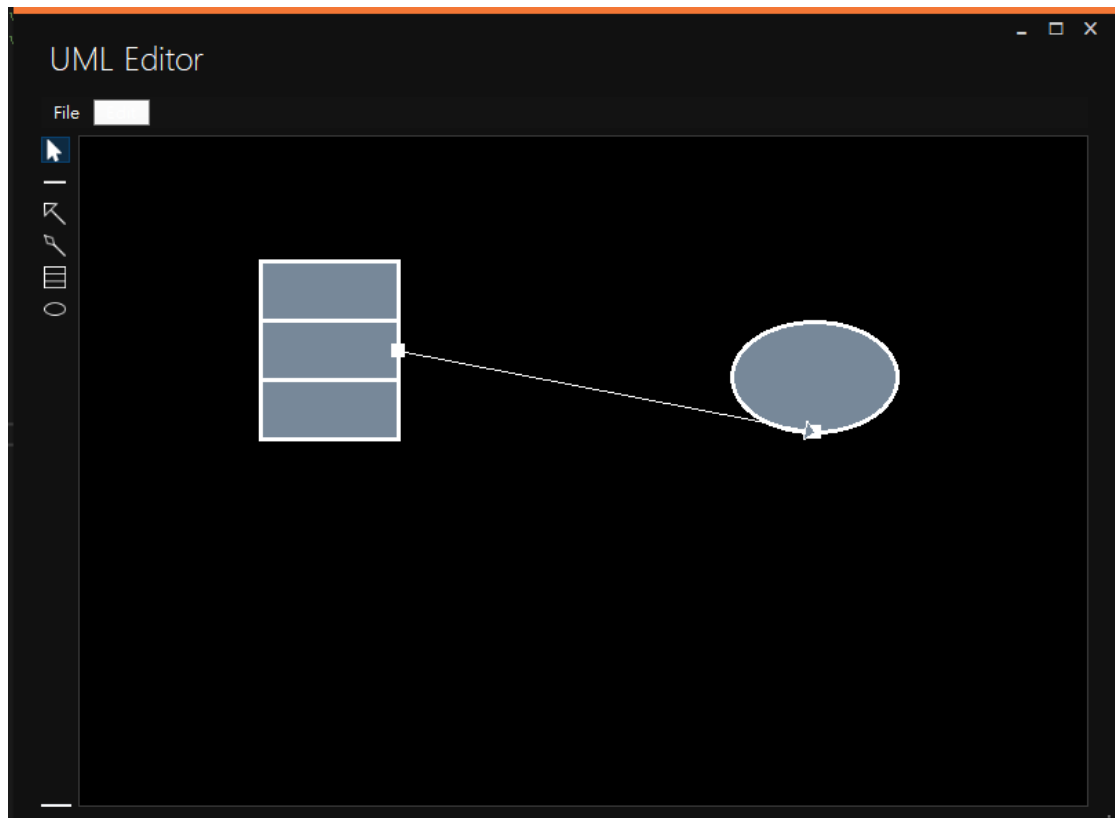


HW2 111522118

軟體工程實務 作業二報告

錯誤一

一開始先按照 use case format 實作，測試案例的過程中，做到了 **UseCase B.1 Creating an UML connection line**，其中要使用 Line 工具，使用之後想刪除 Line 工具時發現一刪除就會卡住，呈現以下狀態



利用全部中斷的功能，檢查程式卡住的點，經過幾次執行後發現應該是 combination 裡面的物件並未被刪除(combination.count 不減少永遠大於 0)，所以 DestroyAllCombinations()會一直不斷輪迴

```
public virtual void DestroyAllCombinations()
{
    Debug.Assert(Combinations != null);

    while (Combinations.Count > 0)
    {
        Combinations[0].Destroy();
    }
}
```

之後順著程式繼續檢查，發現 Destroy 裡面的 Line 的判定跟 Source 與 Destination 有些不同，少 RemoveCombination()這個部分，如下圖

```
1 個參考
public void Destroy()
{
    if (Source != null)
    {
        Source.RemoveCombination(this);
    }

    if (Destination != null)
    {
        Destination.RemoveCombination(this);
    }

    if (Line != null)
    {
        Line.IsSelected = false;
        Canvas.GetInstance().RemoveShape(Line);
    }
}
```

我將 Line.RemoveCombination(this);補上後即可正常刪除。

```
if (Line != null)
{
    Line.RemoveCombination(this);
    Line.IsSelected = false;
    Canvas.GetInstance().RemoveShape(Line);
}
```

錯誤二

在測試 **UseCase C.2 Select/Unselect a group of objects** 功能時，留意到“完全”有被額外 mark 出來，好像是解題暗示的感覺，我就多拖曳了幾次 select 發現偶爾會產生沒有完全框住但還是被選中的狀況，於是我將中斷點設置在 OnMouseMove 函式的並進行 select 的動作，一開始到斷點後逐步執行都很正常直到程式跑到 selectmode.cs 裡面的 MultipleSelect()，發現 P2 在計算高度的時候也是用 selectedArea.X 下去算，當即改成 selectedArea.Y 後執行就正常了

```

/// </summary>
/// <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.Y + selectedArea.Height);

    foreach (Shape shape in Canvas.Shapes)
    {
        if (shape.IsWithin(p1, p2))
        {
            Canvas.RemoveShape(shape);
            Canvas.AddShape(shape);
            shape.IsSelected = true;
        }
    }
}

```

錯誤三

測試 **UseCase D.1 Group objects** 的功能時，發現移動 group 時只有框會移動而已，內部的被 Group 的節點並不會跟著移動，把 break point 設在 onmusedrag 的部份，發現 code 跑到 move 的時候，move 的動作只跑到 CompositionObject，code 沒有跑到 CompositionObject 內部的 shapes 的 shape 們，因此額外加了一個 for 迴圈，如果傳入的 shape 的 count 不等於 0 (表示該 shape 內部還有 shapes)，也要 move shapes 裡面的 shape

```

public virtual void Move(int offsetX, int offsetY)
{
    X += offsetX;
    Y += offsetY;
    for (int i = 0; i < Ports.Length; i++)
    {
        if (Ports[i].Movable)
        {
            Ports[i].Move(offsetX, offsetY);
        }
    }
    for (int j = 0; j < Count; j++) {
        Shapes[j].Move(offsetX, offsetY);
    }
}

```