

Bug1 線條無法刪除

```
/// </summary>
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);
    }
}
```

改成

```
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;
        Line.RemoveCombination(this); //Bug1
        Canvas.GetInstance().RemoveShape(Line);
    }
}
```

原因是 line 沒有做到 RemoveCombination，因此會在一個迴圈中停重複。

Bug2 圖形無法和組合的框框一起移動

```
<summary>
Groups all given <see cref="Shape"/>s and gets the grouped
<see cref="CompositionObject"/>.
</summary>
<param name="shapes"><see cref="Shape"/>s to be grouped.</param>
<returns>Returns the grouped <see cref="CompositionObject"/></returns>
1 個參考
private CompositionObject GroupShapes(Shape[] shapes)

    CompositionObject compositionObject = new CompositionObject();

    foreach (Shape shape in shapes)
    {
        Canvas.RemoveShape(shape);
        compositionObject.Add(shape);
    }

    compositionObject.UpdateSize();
    return compositionObject;
```

改成

```
1 個參考
private CompositionObject GroupShapes(Shape[] shapes)
{
    CompositionObject compositionObject = new CompositionObject();
    foreach (Shape shape in shapes)
    {
        //if (shape is BasicObject)
        //{
        //    Canvas.RemoveShape(shape);
        //    compositionObject.Add(shape);
        //}
        compositionObject.Add(shape); //Bug2
    }
    compositionObject.UpdateSize(); //刪除比較好看
    return compositionObject;
}
```

原因是 shape 被刪除後變成 null，這樣加入的 shape 就是 null。

Bug3：ungroup 後的移動問題

```
/// </summary>
/// <param name="shapes">The <see cref="Shape"/>s to be u
1 個參考
private void UngroupShapes(Shape[] shapes)
{
    foreach (Shape shape in shapes)
    {
        if (shape.Count > 1) // Then it is a composittion
        {
            Canvas.RemoveShape(shape);

            while (shape.Count > 0)
            {
                Shape tempShape = shape.RemoveFirst();
                tempShape.IsSelected = false;
                Canvas.AddShape(tempShape);
            }
        }
    }
}
```

改成

```
/// <param name="shapes">The <see cref="Shape"/>s to be u
1 個參考
private void UngroupShapes(Shape[] shapes)
{
    foreach (Shape shape in shapes)
    {
        if (shape.Count > 1) // Then it is a composittion
        {
            Canvas.RemoveShape(shape);

            while (shape.Count == 0) //Bug3
            {
                Shape tempShape = shape.RemoveFirst();
                tempShape.IsSelected = false;
                Canvas.AddShape(tempShape);
            }
        }
    }
}
```

原因是 `shape.count == 0` 無法進到這個 `while`，沒有刪除之前的移動會有問題，進到這個 `while` 才會重繪該 `shape`。

Bug4

```
/// <summary>
/// Updates the location and size of <see cref="SelectedArea"/>.
/// </summary>
/// <param name="selectedArea">The <see cref="SelectedArea"/> to be updated.</param>
1 個參考
private void UpdateSelectedArea(Shape selectedArea)
{
    int left = _mousePressedPoint.X;
    int top = _mousePressedPoint.Y;
    int width = _currentMousePoint.X - _mousePressedPoint.X;
    int height = _currentMousePoint.Y - _mousePressedPoint.Y;

    selectedArea.SetLocation(left, top);
    selectedArea.SetSize(width, height);
}
}
```

```
/// <summary>
/// <param name="width">The value of width.</param>
/// <param name="height">The value of height.</param>
11 個參考
public virtual void SetSize(int width, int height)
{
    Width = width;
    Height = height;
}
}

/// <summary>
```

```
1 個參考
private void UpdateSelectedArea(Shape selectedArea)
{
    //int left = _mousePressedPoint.X; //Bug4
    //int top = _mousePressedPoint.Y;

    int width = _currentMousePoint.X - _mousePressedPoint.X;
    int height = _currentMousePoint.Y - _mousePressedPoint.Y;

    int left = width > 0 ? _mousePressedPoint.X : _mousePressedPoint.X - Math.Abs(width);
    int top = height > 0 ? _mousePressedPoint.Y : _mousePressedPoint.Y - Math.Abs(height);

    selectedArea.SetLocation(left, top);
    selectedArea.SetSize(width, height);
}
}
```

```
/// <param name="height">The value of height.</param>
11 個參考
public virtual void SetSize(int width, int height)
{
    Width = Math.Abs(width); //Bug4
    Height = Math.Abs(height);
}

/// <summary>
/// Sets the name of this <see cref="Shape"/>.
```

原因是 width 或 height 小於 0 時，無法 setsize，重新設定左上角的基準點(left top)，依據不同的情況調整即可。