

SnapScrollerPlugin

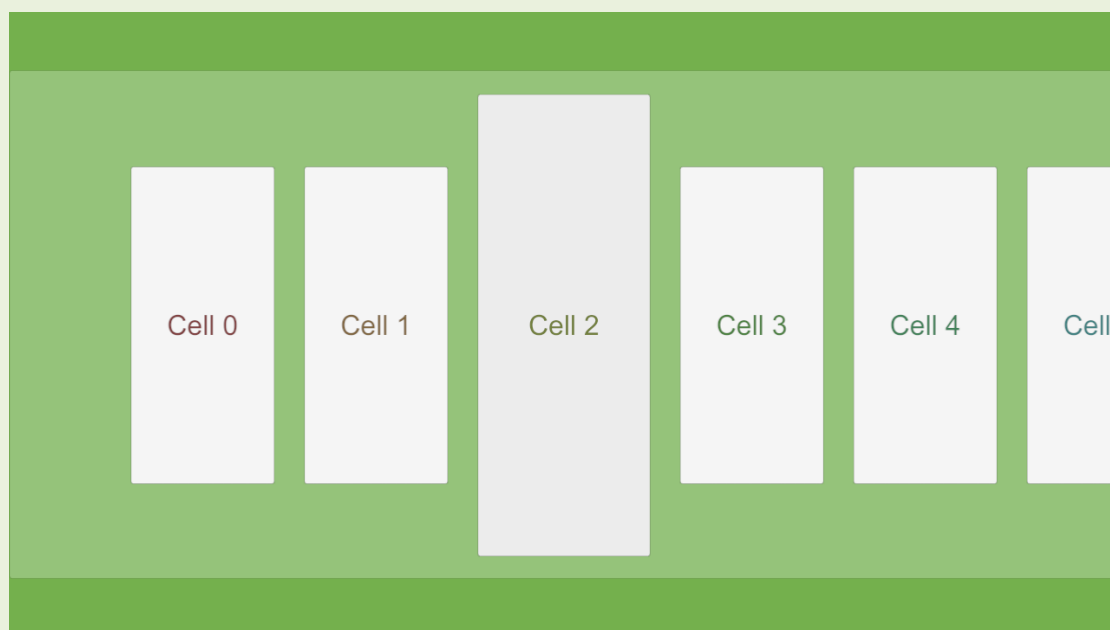
使用者手冊

魚丸®蕾娜 RenaWevin

SnapScrollerPlugin 是一款能夠讓你在 Unity 中
為 ScrollRect 實現自動定位的插件，並且被定位的物件會產生縮放的效果。

本插件有以下特點

- ◆ 針對卷軸正中心項目的自動定位
- ◆ 項目的自動縮放
- ◆ 使用物件池
- ◆ 支援無限循環滾動 (可選)



目錄

PDF 檔可直接點擊以下標題文字導覽至段落。

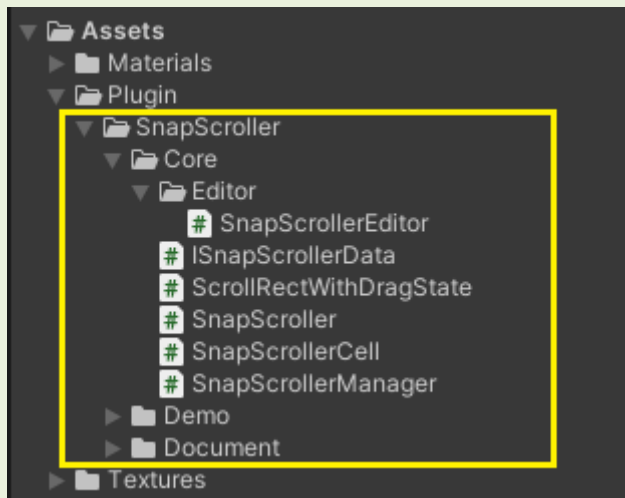
開始使用	3
安裝 SnapScrollerPlugin.....	3
極輕量使用.....	4
客製化使用.....	10
物件可調整參數解釋	16
SnapScroller 物件	16
SnapScrollerCell 物件	17

開始使用

安裝 SnapScrollerPlugin

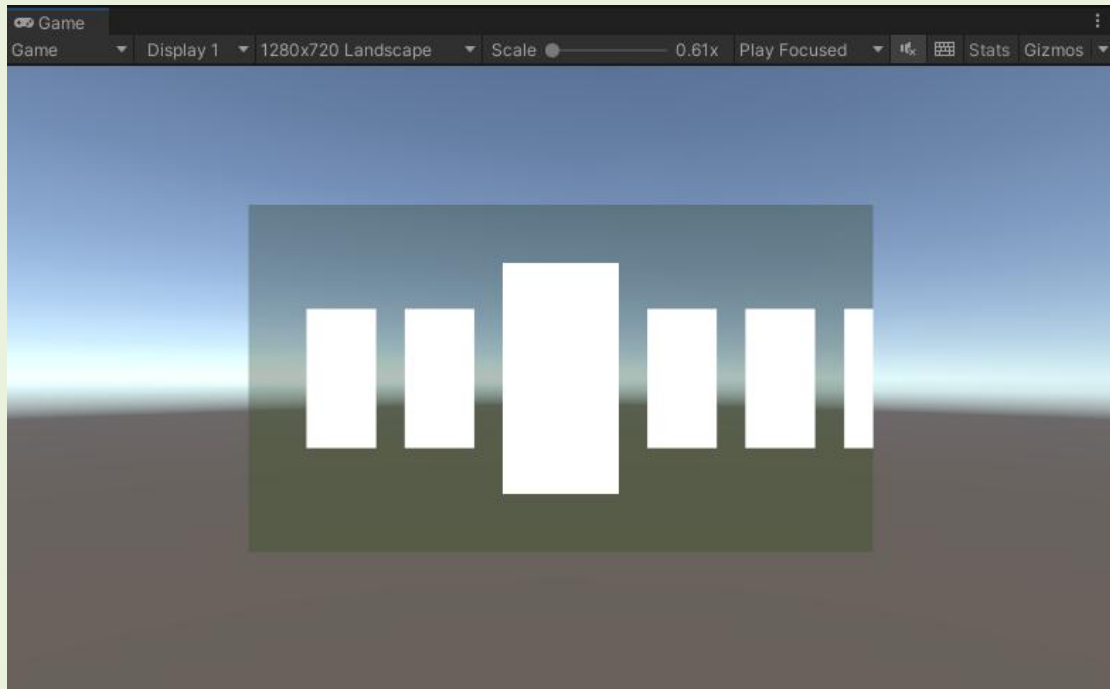
你可以從 Unity Asset Store 安裝此插件，安裝後應會有如下圖的檔案，請確保 SnapScroller 底下的 Core 資料夾內有如下的所有檔案，

Demo 與 Document 可在專案用不到時刪除。



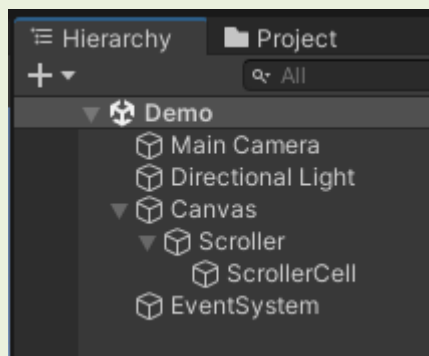
極輕量使用

以下將展示此插件最輕量的用法，將會製作出如下圖片中，有 10 個空白物件的 Scroller。



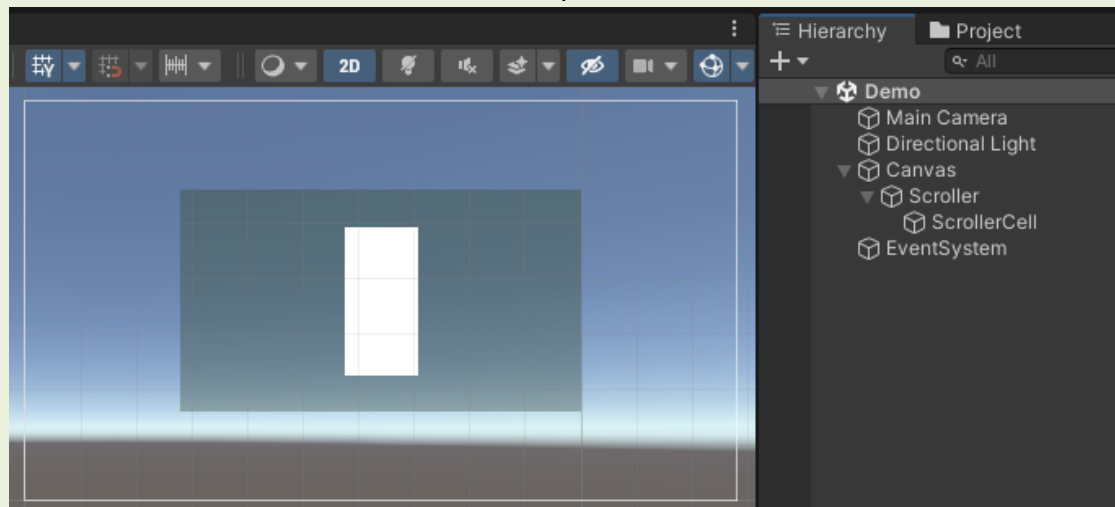
1. 創建場景物件

本步驟創建了一個新場景，新增了 UI 必要的 Canvas 與 EventSystem，接著在 Canvas 物件下新增 Scroller、ScrollerCell 遊戲空物件，最後的狀態如下圖。



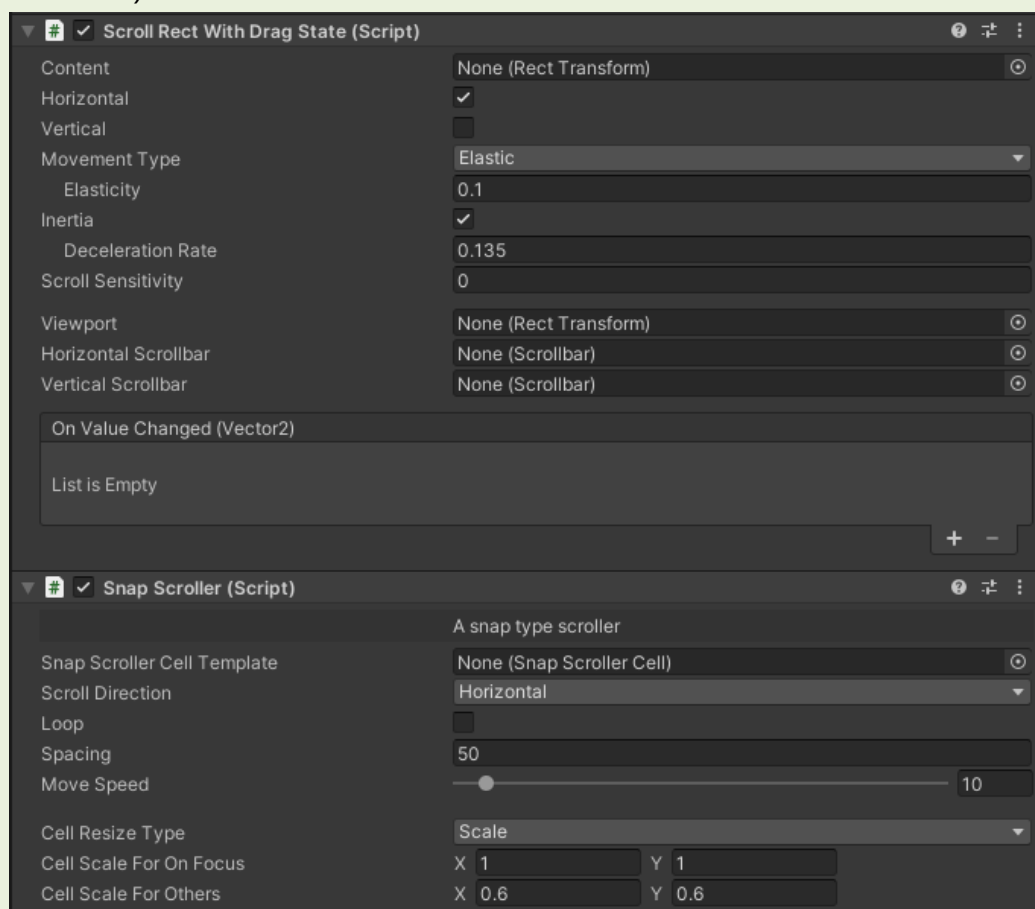
2. 為 UI 新增 component

本步驟將會為剛剛新增的物件新增 component 到遊戲物件中。



將 Scroller 物件(可滑動區域)與 ScrollerCell 物件(單個物件)縮放至你希望的大小 (Demo 中畫布為 1920×1080，Scroller 為 1080×600，Cell 為 200×400)。

在 Scroller 物件中新增 SnapScroller 腳本(同時 ScrollRectWithDragState 將被自動新增)，此時可調整部分參數。



依照滑動方向：

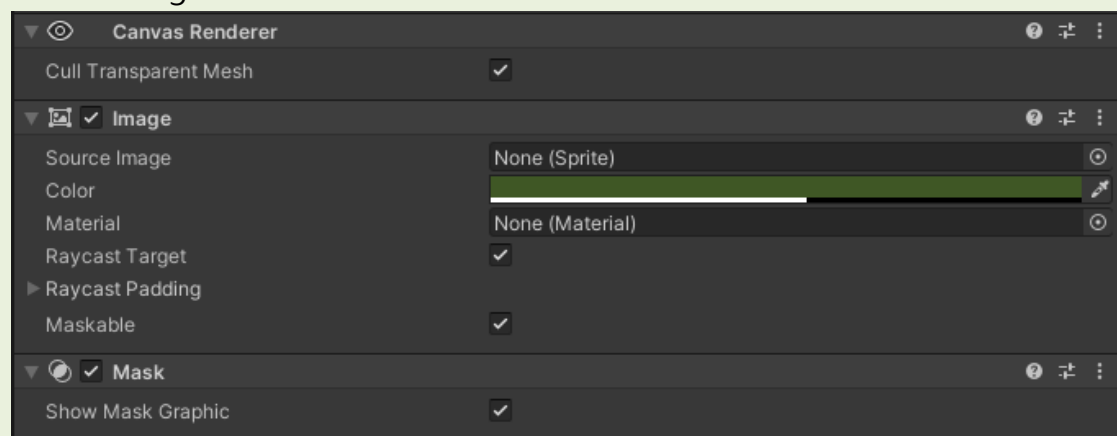
水平滑動請為 SnapScroller 的 ScrollerDirection(捲動方向)設置為 Horizontal，並將 ScrollRectWithDragState 的 Horizontal 打勾、Vertical 取消勾選；

垂直滑動則為 SnapScroller 的 ScrollerDirection 設置為 Vertical，

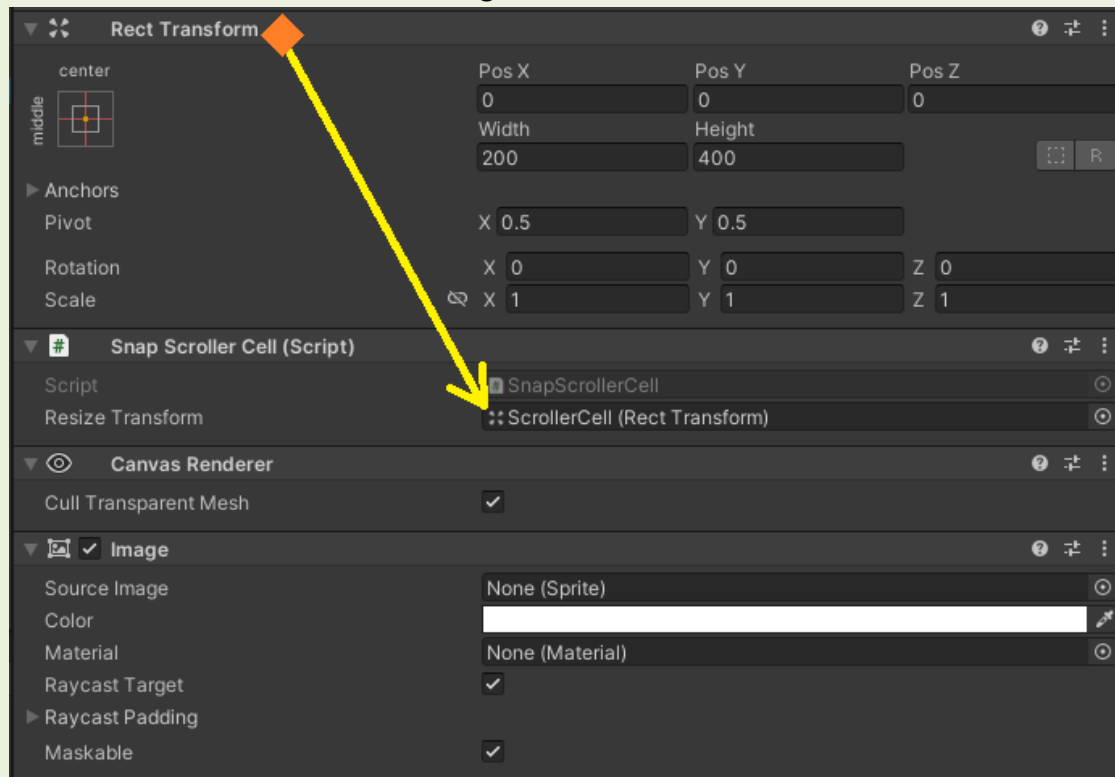
並將 ScrollRectWithDragState 的 Vertical 打勾、Horizontal 取消勾選。

接著為 Scroller 物件新增 Unity 內建的 Image 元件與 Mask 元件，用以接收觸控與隱藏範圍外的 Cell 物件，同時請確保 Image 的 Raycast Target 必須勾選，才能確實接收觸控滑動事件。

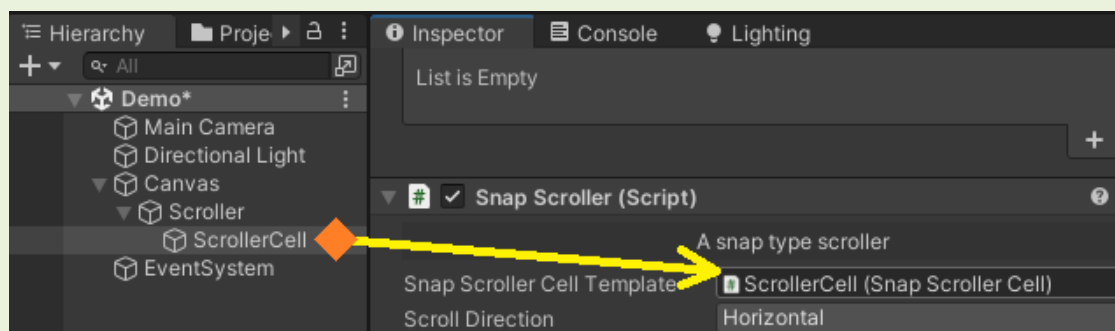
這裡的 Image 同時也可以給予 Scroller 底色與圖片，範例如下。



在 ScrollerCell 物件中新增 SnapScrollerCell 腳本與 Unity 內建的 Image 元件，將這個物件自己的 Rect Transform 元件拖曳給 SnapScrollerCell 腳本，使其在縮放時可以調整自己的大小；Image 元件則是給予 Cell 白色的外觀。



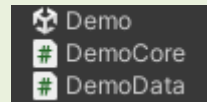
最後回到 Scroller 物件，將稍早的 ScrollerCell 拖曳給 SnapScroller 腳本的 Snap Scroller Cell Template 欄位，讓 Scroller 使用這個 Cell 做為 Prefab 來呈現內容。



3. 新增必要腳本

此步驟將會新增以下腳本：DemoData 與 DemoCore。

以下分別是兩個腳本的內容：



DemoData (定義為 class 或 struct 皆可) - 用以給予空資料容器

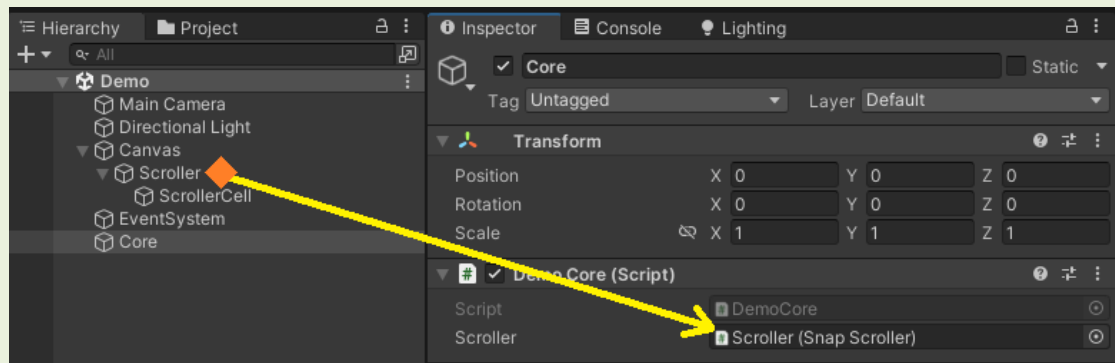
```
1 using RW.UI.SnapScrollerPlugin;
2
3 public struct DemoData : ISnapScrollerData {
4
5 }
```


DemoCore - 用以初始化 Scroller 與決定 cell 物件數量

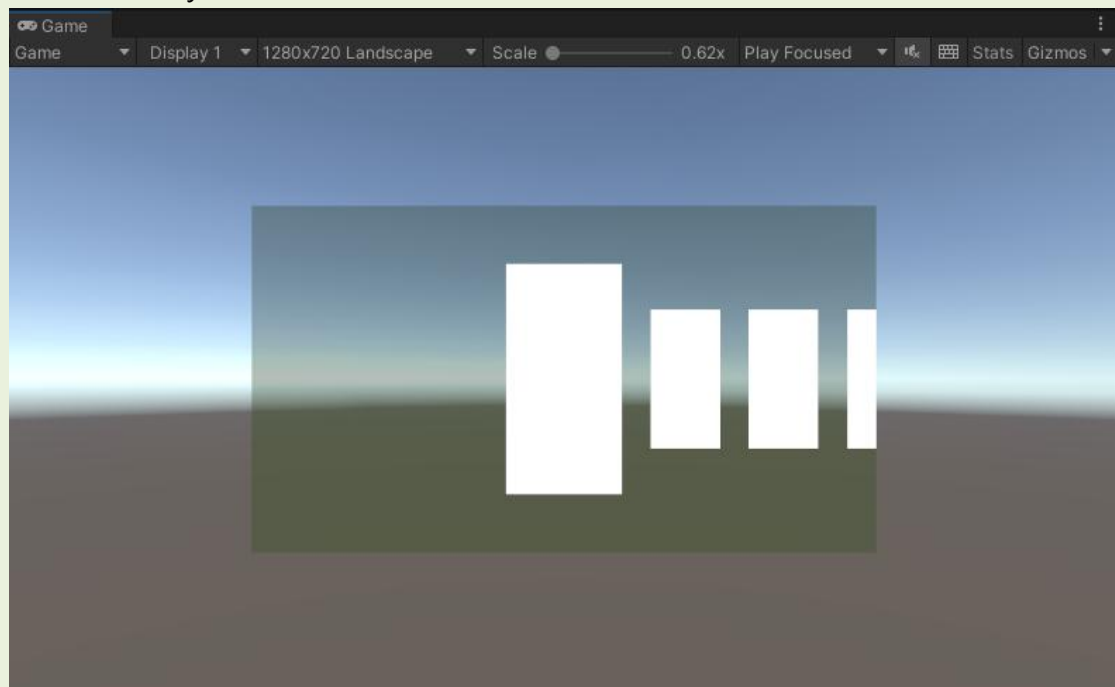
```
1 using RW.UI.SnapScrollerPlugin;
2 using UnityEngine;
3
4 public class DemoCore : MonoBehaviour {
5
6     private SnapScrollerManager manager;
7
8     [SerializeField]
9     private SnapScroller scroller;
10
11     void Start() {
12         manager = new SnapScrollerManager();
13         scroller.SetManager(manager);
14         manager.ClearData();
15         for (int i = 0; i < 10; i++) {
16             manager.AddData(new DemoData());
17         }
18         scroller.RefreshData();
19     }
20 }
```


4. 實裝腳本

在整個場景最外層新增一個名為 Core 的空遊戲物件，將 DemoCore 腳本掛上去，並且將 Scroller 物件拖曳進 DemoCore 腳本的 Scroller 欄位中。



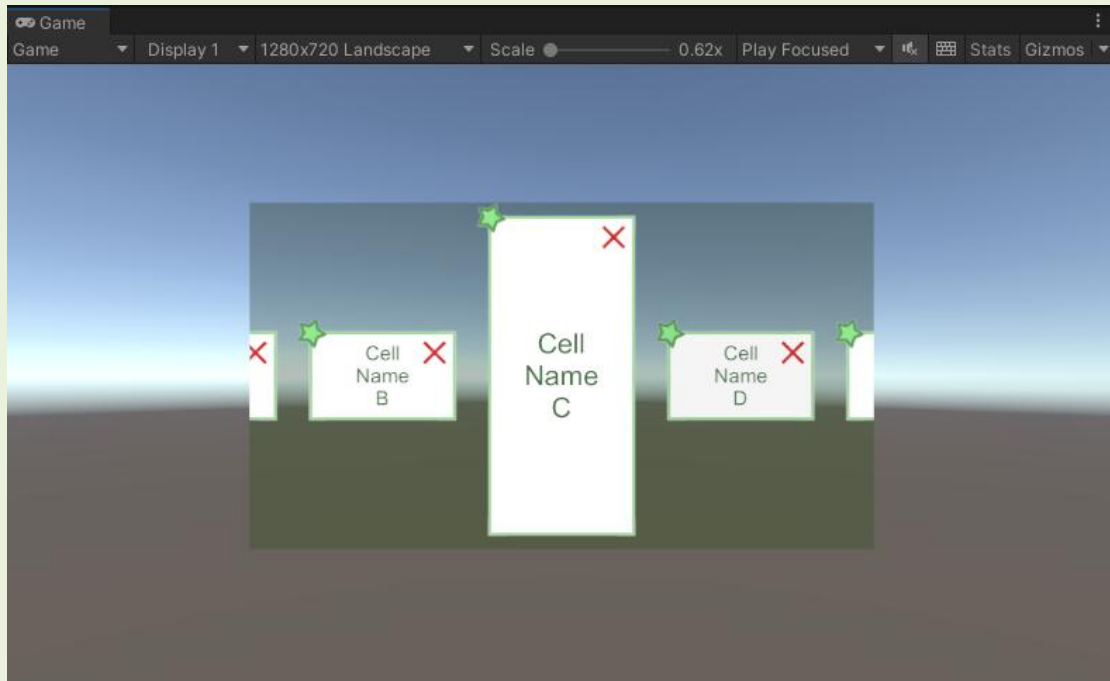
最後按下 Play 鍵  開始遊戲，到此全部完成。



可以左右滑動，總共會有 10 個白色 Cell 方框。

客製化使用

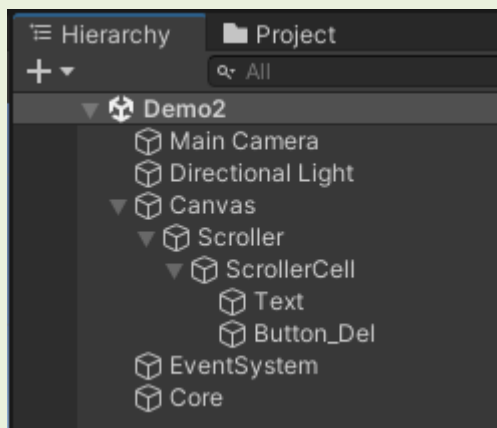
以下將展示此插件客製化的用法，將會製作出如下圖片中，有 10 個物件、每個物件有 A~J 的名稱、可以點擊物件使其被選取、並且可以刪除該項目的 Scroller。



1. 創建場景物件、UI 基本功能

本步驟如極輕量使用的做法。創建新場景、新增 UI 必要的 Canvas 與 EventSystem，接著新增以下物件：

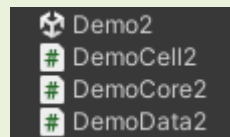
- ◆ Scroller：Scroller 本體
- ◆ ScrollerCell：Cell 物件本體
- ◆ Core：控制資料初始狀態的腳本放置區
- ◆ Text：Cell 中顯示的文字
- ◆ Button_Del：用以刪除項目的按鈕



2. 預先準備腳本

此步驟將會新增以下 3 個腳本：DemoData2、DemoCore2、DemoCell2。

以下分別是三個腳本的內容：



DemoData2 - 用以給予資料容器

```
1 using RW.UI.SnapScrollerPlugin;
2
3 public class DemoData2 : ISnapScrollerData {
4     public string text;
5 }
```

DemoCore2 - 用以初始化 Scroller 與決定 cell 資料內容與數量

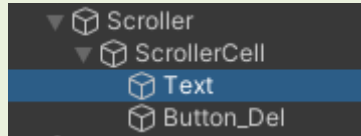
```
1 using RW.UI.SnapScrollerPlugin;
2 using UnityEngine;
3
4 public class DemoCore2 : MonoBehaviour {
5
6     private string[] names = {
7         "A", "B", "C", "D", "E", "F", "G", "H", "I", "J"
8     };
9
10    private SnapScrollerManager manager;
11
12    [SerializeField]
13    private SnapScroller scroller;
14
15    void Start() {
16        manager = new SnapScrollerManager();
17        scroller.SetManager(manager);
18        manager.ClearData();
19        for (int i = 0; i < 10; i++) {
20            manager.AddData(new DemoData2() {
21                text = $"Cell\nName\n{names[i]}"
22            });
23        }
24        scroller.RefreshData();
25    }
26 }
```

DemoCell2 - 用以自訂要在每一格 Cell 物件上顯示的 UI

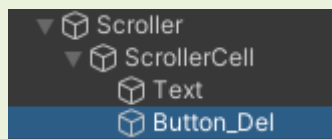
```
1 using RW.UI.SnapScrollerPlugin;
2 using UnityEngine;
3 using UnityEngine.UI;
4
5 public class DemoCell2 : SnapScrollerCell {
6
7     [SerializeField]
8     private Text text;
9     [SerializeField]
10    private Button btn_Select;
11    [SerializeField]
12    private Button btn_Del;
13
14    void Start() {
15        btn_Select.onClick.AddListener(OnClick_Btn_Select);
16        btn_Del.onClick.AddListener(OnClick_Btn_Del);
17    }
18
19    public override void OnSetData() {
20        DemoData2 data = GetData as DemoData2;
21        if (data == null) { return; }
22        text.text = data.text;
23    }
24
25    private void OnClick_Btn_Select() {
26        scroller.ScrollToIndex(cellIndex);
27    }
28
29    private void OnClick_Btn_Del() {
30        manager.RemoveDataAt(cellIndex);
31        scroller.RefreshData();
32    }
33
34 }
```

3. 為 UI 新增 component、實裝腳本

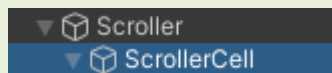
本步驟將會為剛剛的物件新增 UI-Component 到遊戲物件中，將會跳過所有樣式的調整步驟(顏色、滑動方向、文字內容等)。



在 Cell 底下的 Text 物件中，新增 Unity 內建的 Text 元件，並且調整此物件在 Cell 中的顯示方式(大小、文字顏色、字型大小等)。

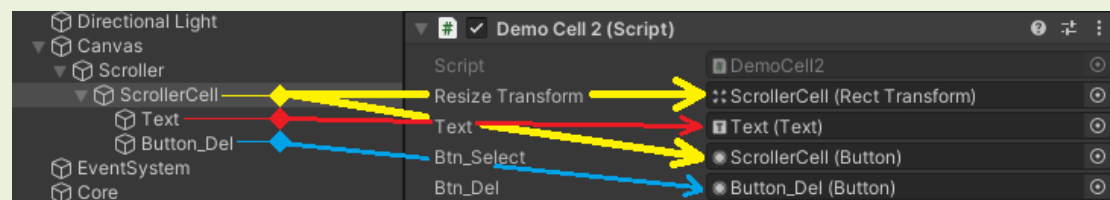


接著處理 Cell 底下的 Button_Del 物件，新增 Unity 內建的 Button 元件與相對應要顯示的外觀(如 Image 或 Text 元件)與顯示位置，Button 按下的事件稍早已在腳本內決定了。



接著設定 ScrollerCell 物件本體，先為此新增 DemoCell2 腳本，並且同時新增 Unity 內建的 Image 元件與 Button 元件，用以顯示外框與當作選擇按鈕。

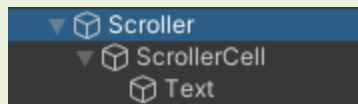
並且將以下物件拖曳到 ScrollerCell 中相對應的欄位：



ScrollerCell 物件 → ResizeTransform、Btn_Select 欄位

Text 物件 → Text 欄位

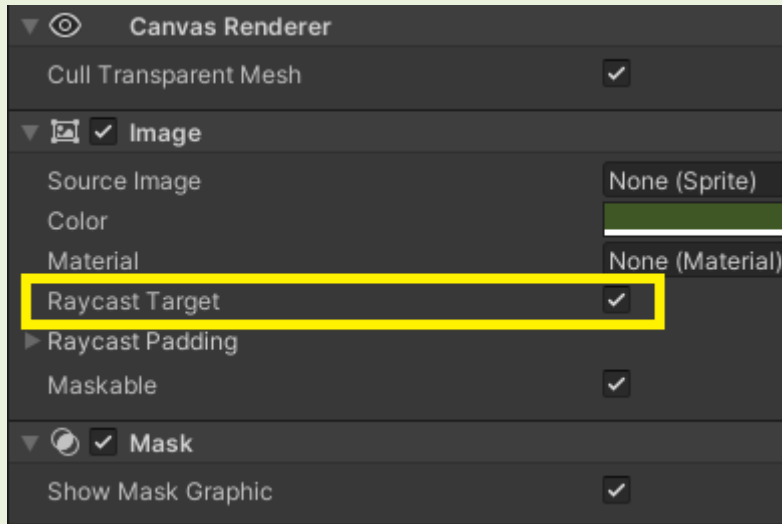
Button_Del 物件 → Btn_Del 欄位



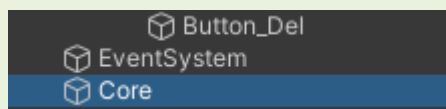
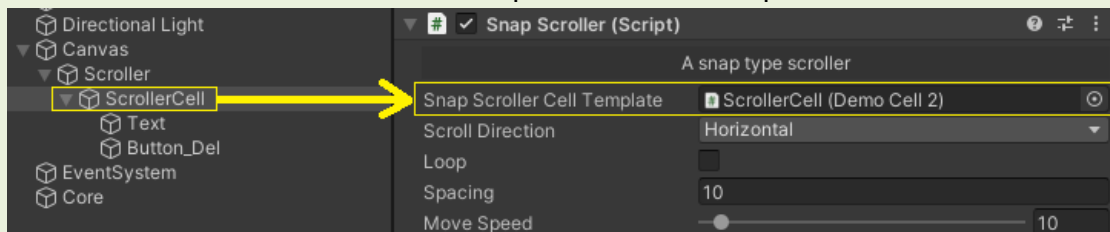
在 Scroller 物件中，新增 SnapScroller 腳本(同時 ScrollRectWithDragState 將被自動新增)，此為 Scroller 的本體。

並且為此物件新增 Unity 內建的 Image 元件與 Mask 元件，
用以接收觸控與隱藏範圍外的 Cell 物件，

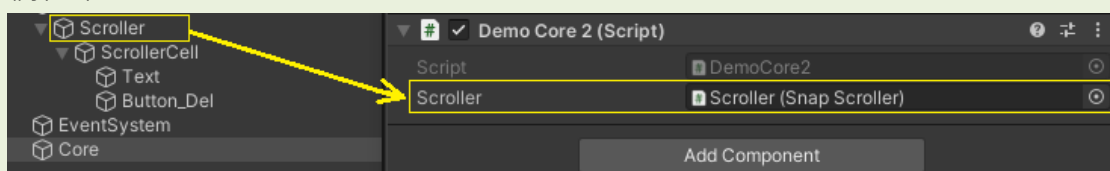
同時請確保 Image 的 Raycast Target 必須勾選，才能確實接收觸控滑動事件。




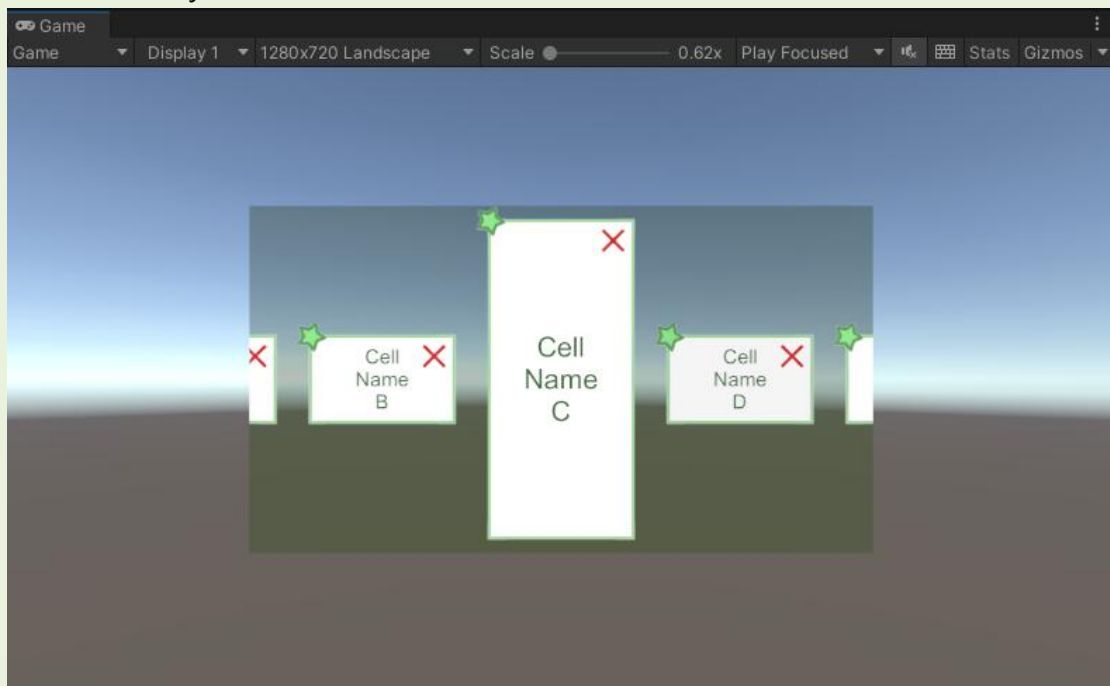
接著將 ScrollerCell 物件拖曳到 SnapScrollerCellTemplate 欄位上。



最後在 Core 物件中新增 DemoCore2 腳本，並且將 Scroller 物件拖曳至 Scroller 欄位中。



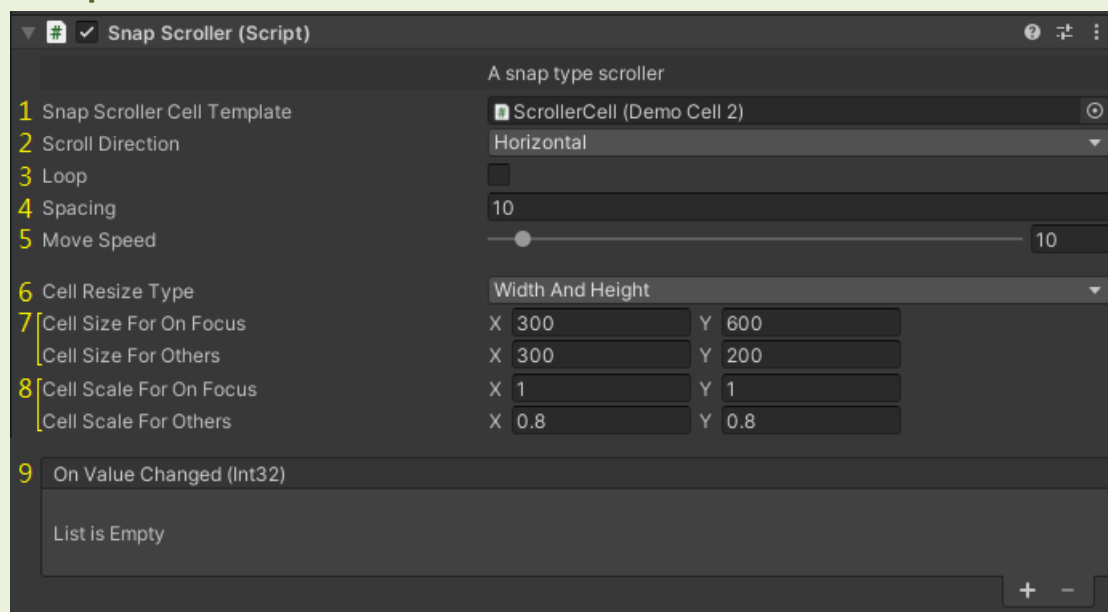
最後按下 Play 鍵  開始遊戲，到此全部完成。



Demo 中每個 Cell 會有 A~J 的名稱，點擊 Cell 物件可以快速滑動到該 Cell，並且點擊右上角的紅色 X 可以刪除該資料。

物件可調整參數解釋

SnapScroller 物件



1. Snap Scroller Cell Template

必要，用來指定 Cell 的模板，Cell 物件的外觀與腳本都是從這裡指定。

2. Scroll Direction

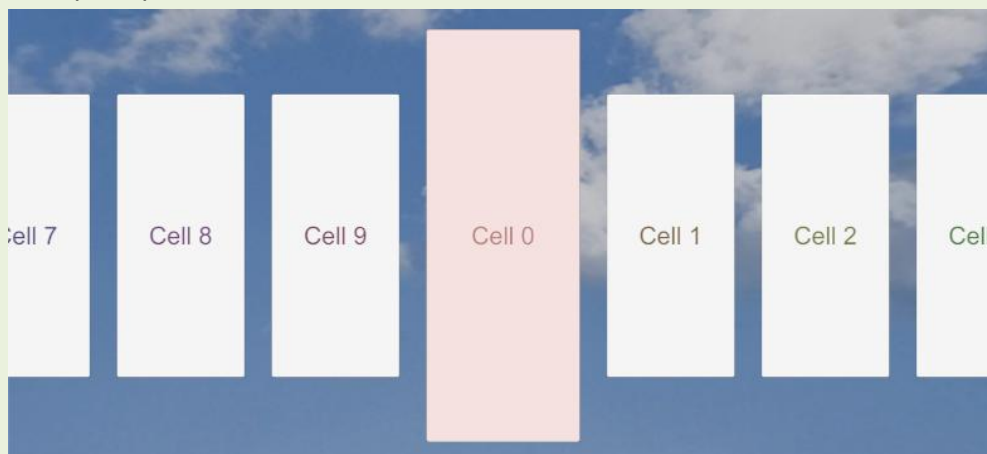
滑動的方向，需要 Scroll Rect 的配合。

Horizontal: 水平的、左右的滑動方向。

Vertical: 垂直的、上下的滑動方向。

3. Loop

打勾(啟用)後會使 Scroller 的項目有頭尾循環，最後一項與第一項會相連。



4. Spacing

每個 Cell 物件之間的時間寬度，單位為像素。

5. Move Speed

Scroller 的移動速度，使用於使用者放開滑鼠/手指時、或是呼叫 `scroller.ScrollTo` 系列方法時，Scroller 將物件移動到定位的速度。

6. Cell Resize Type

縮放 Cell 的方式，使用於移動 Scroller 的位置時，Cell 被 Focus 與否造成的縮放方法。

None: 完全不縮放。

Scale: 會使用 Cell 物件的 `localScale` 縮放。

Width And Height: 會直接更改 Cell 物件的寬與高。

7. Cell Size For On Focus / Others

縮放方法選用為 Width And Height 時，才會出現這個參數。

會決定 Cell 物件被 Focus(在正中間)或是未被 Focus 時的寬與高，

X 參數代表寬度，Y 參數代表高度。

8. Cell Scale For On Focus / Others

縮放方法選用為 Scale 時，才會出現這個參數。

會決定 Cell 物件被 Focus(在正中間)或是未被 Focus 時的 `LocalScale`。

9. On Value Changed (Int32)

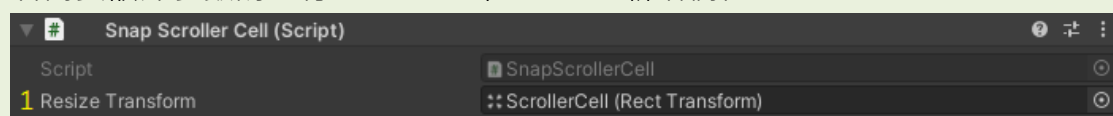
會在被 Focus 的 Cell 編號改變時呼叫一次的事件。

方法的 `int` 參數是改變後現在正被 Focus 的 Cell 編號。

SnapScrollerCell 物件

此處僅列出物件預設的參數，

若需要額外參數請自行 `override` 本 class 並新增欄位。



1. Resize Transform

用來指定在 Cell 物件被 Scroller 縮放時，會被操作的 Rect Transform。