

作業二

題目：(承一) 簡易 UNO 遊戲

物理一 B09202018 林家晞

一、動機及目的

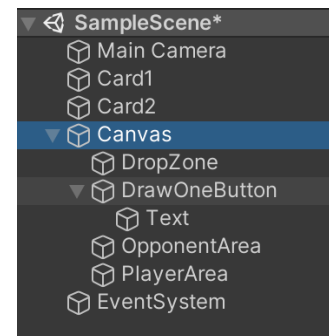
由兩次作業以及期末專題循序完成經典的 UNO 卡牌遊戲，並由過程中學習如何建立控制台應用程式以及視窗應用程式。



本次作業利用 Unity 結合 Visual Studio 進程式設計，初步建立了 UNO 遊戲畫面以及抽牌功能。

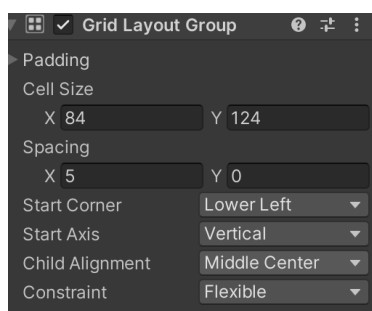
二、Unity 編輯解說

Unity 為一跨平台遊戲引擎，使用 C#與 C++語言進行開發。建立專案後，首先在 Canvas 底下建立三個屬性為 Image 的 gameObject：OpponentArea、PlayerArea、Dropzone，分別代表對戰方（電腦）的牌堆、玩家的牌堆以及桌面。其次建立抽牌按鈕 DrawOneButton，並建立新的程式碼「DrawOne.cs」連結至此遊戲物件。最後則是建立卡牌的 gameObject，連結至程式碼「playACard」。

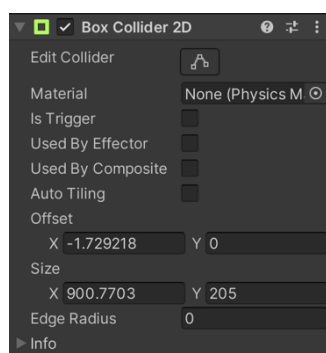


此外，每個物件可以加入 component 以達到程式設計所需的效果。下列為本次作業中運用到的 component。

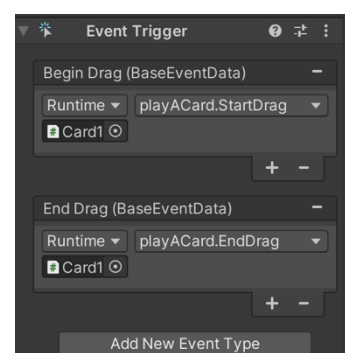
1. Grid Layout Group



2. Box Collider 2D



3. Event Trigger



三、程式碼解說

1. DrawOne.cs

1.1 建立卡牌

程式碼仿造先前的手法，利用巢狀清單儲存卡牌的顏色、數字/功能，並新增了對應 Source Image 的路徑。卡牌共有 108 張，並有 51 張 png 圖檔（參見下圖），其中圖案的設計參照 UNO Minimalista Card Game 的設計。



1.2 隨機抽牌功能

```
public void playerTurn()
{
    int rdn = UnityEngine.Random.Range(0, 108);
    //實例化物件
    GameObject playerCard = Instantiate(Card1, new Vector3(0, 0, 0), Quaternion.identity);
    playerCard.transform.SetParent(PlayerArea.transform, false);
    //更改物件Source Image
    string cardName = Convert.ToString(deck()[rdn][2]);
    playerCard.GetComponent<Image>().sprite = Resources.Load<Sprite>(cardName);
}

public void opponentTurn()
{
    //實例化物件，將對手的手牌翻轉180度
    GameObject opponentCard = Instantiate(Card2, new Vector3(0, 0, 0), Quaternion.Euler(0, 0, 180));
    opponentCard.transform.SetParent(OpponentArea.transform, false);
}
```

在設計過程中遇到了這樣的問題：*Random is ambiguous reference between 'Unity.Random' and 'System.Random'*，經查證後，其原因出在 UnityEngine 內部已有 Random 的套件，但功能與 System 底下的 Random 不盡相同，因此採用了前者。

playerTurn()函式由編號 0 至 107 的卡牌中隨機取出，從先前建立的 deck()之中取出其編號之卡牌的 Source Image 名稱，建立 Image 於 PlayerArea 母集下，並設定圖像為所取出的字串。opponentTurn()則有相同功能，但設定圖像為覆蓋的牌面。

1.3 Start(): 在第一幀前即被呼叫，藉此設定起始的五張手牌。

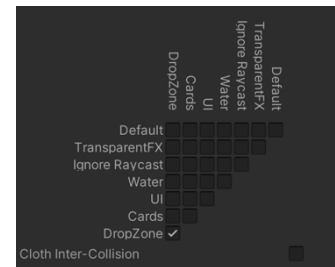
```
void Start()
{
    GameObject topcard = Instantiate(Card2, new Vector3(0, 0, 0), Quaternion.identity);
    topcard.transform.SetParent(DropZone.transform, false);
    for(int i=0; i<5; i++)
    {
        playerTurn();
        opponentTurn();
    }
}
```

2. PlayACard.cs

2.1 拖移功能

```
private void OnCollisionEnter2D(Collision2D collision)
{
    isOverDropZone = true;
}

private void OnCollisionExit2D(Collision2D collision)
{
    isOverDropZone = false;
}
```



在 Unity 引擎中的 Physics 設定卡牌僅會與 DropZone 碰撞，以 Collision2D 檢查卡牌覆蓋在 DropZone 之上。

```
//儲存拖移前的位置
public void StartDrag()
{
    startPosition = transform.position;
    isDragging = true;
}

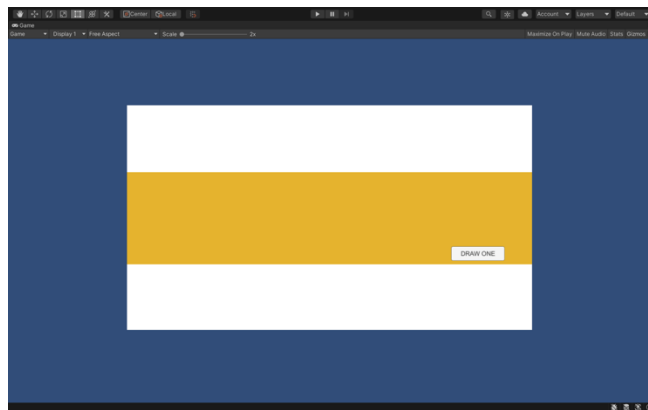
public void EndDrag()
{
    isDragging = false;
    if (isOverDropZone)
    {
        transform.SetParent(dropZone.transform, false);
    }
    else
    {
        transform.position = startPosition;
    }
}
```

StartDrag()與 EndDrag()分別對應 Event Trigger 中的 Begin Drag 與 End Drag 事件，其中 StartDrag()被啟動時紀錄拖曳前的位置，EndDrag()檢查停止拖曳時是否在 DropZone 上，若是則設卡牌 parent 為 DropZone，若非則將卡牌移回原本的位置。

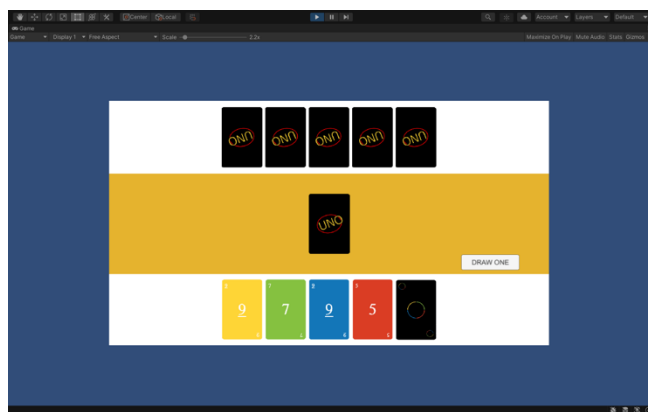
2.2 Update(): 遊戲過程中會不斷呼叫，藉此改變拖曳中卡牌的位置

```
void Update()
{
    if (isDragging)
    {
        transform.position = new Vector2(Input.mousePosition.x, Input.mousePosition.y);
    }
}
```

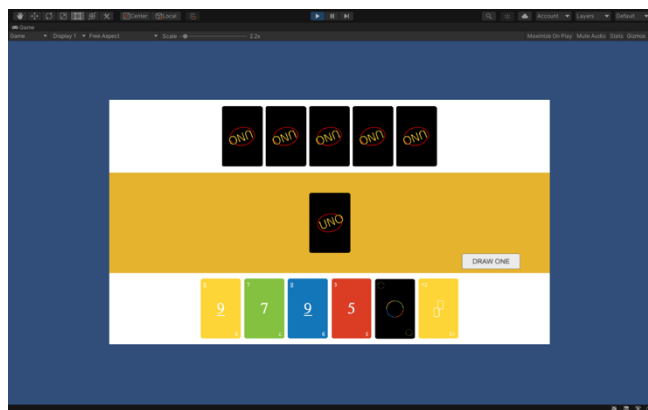
四、執行成果



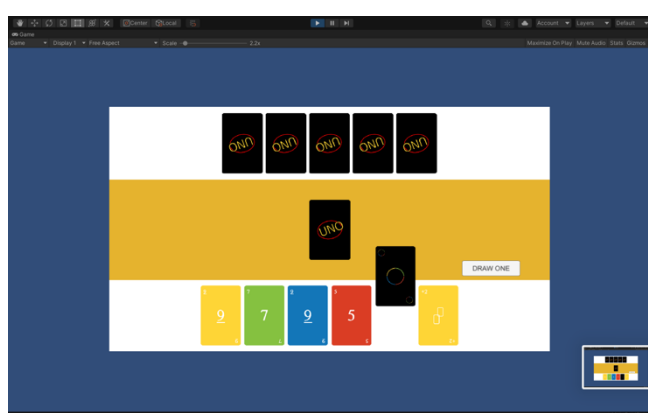
遊戲起始畫面



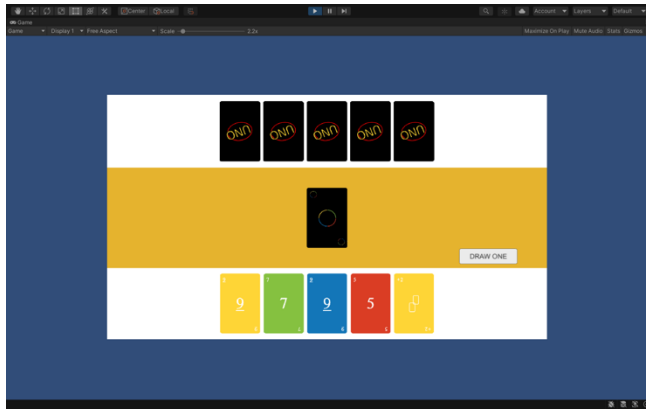
玩家與對手抽五張手牌



按下 DRAW ONE
隨機抽一張卡牌



拖曳手牌中任意卡牌放置於
黃色 DropZone 內



所出手牌覆蓋於牌堆之上

五、參考資料

1. 卡牌設計來源：UNO Minimalista Card Game Featuring Designer Graphics by Warleson Oliviera, 108 Cards, Kid, Family & Adult Game Night, Unique Gift Design Lovers Ages 7 Years & Older
<https://www.amazon.com/Mattel-Games-GYH69-UNO-Minimalista/dp/B089DR3N7S>
2. 程式碼參考：How to Create a 2D Card Game in Unity – Part 1 (Setup and Basic Game Architecture)
<https://www.youtube.com/watch?v=0-dUB52eEMk>
3. 程式碼參考：Resources.Load 文件
<https://docs.unity3d.com/ScriptReference/Resources.Load.html>
4. 程式碼參考：How do I change the source image of a Unity image component?
<https://stackoverflow.com/questions/64595434/how-do-i-change-the-source-image-of-a-unity-image-component>