

# Unity ACT Combat System

專題編號：108-CSIE-S002

執行期限：107 年第 1 學期至 108 年第 1 學期

指導教授：陳偉凱

專題參與人員：105590043 徐唯哲

## 一、摘要

本專題的主題是開發 3D 動作遊戲中的戰鬥系統並製作數道關卡作為演示，其特色是玩家在攻擊的過程中能以特殊組合鍵的指令切換角色的能力，藉此創造更多的連段方式來突破關卡，使得同樣的關卡能有不同的遊玩體驗來提升可玩性，讓玩家在創造各種攻擊組合與過關方式的過程中獲得成就感。

關鍵詞：Unity、Action Game、Combat System

## 二、緣由與目的

隨著硬體性能與遊戲引擎的進步，現今的遊戲開發很容易做到複數類型組合而成的規模，例如一款遊戲裡能同時出現角色扮演、射擊、動作等三種不同要素。而動作類型是一個經常出現的類別，其重點在於玩家與關卡中各種單位的互動，也就是戰鬥系統的部分。

因為對於這類型遊戲的熱愛且對遊戲機制的設計也有興趣，所以希望能以自己過去的遊戲經歷設計並且實作出一套獨特的戰鬥系統。

## 三、遊戲設計

### (一) 玩家角色

因為是玩家操作的對象，所以會有特殊的動作或技能來形成遊戲的特點。再來數值設定則會影響操作的風格，此系統設計上希望玩家能盡量使用高連段的進攻，所以設計一個能量條，從待機到攻擊

會有一次大消耗，連段中有持續小消耗，讓高連段的操作有較高的效益，使遊玩體驗能更偏向設計者所希望的。

### (二) 敵方行為

這個項目會影響玩家的操作策略與遊玩難度。在玩家已經進入攻擊射程內時並不會馬上執行攻擊，而是會由對峙與攻擊兩種狀態輪替，對峙時會在玩家的周圍環繞隨機時間後執行攻擊，讓玩家不能簡單地察覺敵方的行為來提升難度，希望玩家能找到正確的時機進攻做出高連段。

### (三) 戰鬥邏輯

流程上分為三個部分：

#### 1. 攻擊判定

在 Unity 動畫事件中安插判定程式，用協程在每幀以 Unity 的物理方法檢測碰撞，如果發生碰撞就要求系統處理戰鬥事件，並記錄此物件避免同碰撞框造成二次傷害。

#### 2. 傷害與角色狀態處理

戰鬥事件會有攻擊方、防守方兩個角色物件，並會依據防守方的角色狀態做不同的處理，例如一般狀態就直接呼叫防守方的設定傷害方法，防禦則會有傷害減免，反擊會呼叫攻擊方的被反擊方法。最後回傳處理結果

#### 3. 角色動作與音效呼叫

當事件處理呼叫角色物件方法或者協程取得回傳結果後在繼續執行 Animator 或 AudioSource 的操作讓畫面有正確的視覺、聽覺回饋。

## 四、工具與技術方法

### (一) 工具

1. Unity：遊戲引擎
2. Blender：3D 模型與動畫工具

### (二) 技術方法

#### 1. Animation Root Motion

角色位移通常會直接使用動畫自帶的位移量，讓遊戲畫面表現與 3D 工具中的呈現一樣。但是 Unity 預設取出的數值會出現偏差。所以要在 3D 工具中新增一個 Root Bone 作為明確的依據，並且在動畫設定中的 Motion 選定 Root Bone 當作 Root Motion Node，使得移動符合預期。

#### 2. 動作狀態旗標

本系統的角色動作有許多特殊需求，且須要準確地在特定時間點運作，例如動作銜接時機、轉向、位移縮放等。所以撰寫了旗標物件與 AnimatorInfo 可以掛載的旗標設定面板，讓動作在切換時能有各自的客製化功能。完成上述的需求。

#### 3. 行為樹設計敵方單位 AI

行為樹是目前常用的遊戲 AI 設計方法，樹狀結構能更清晰的設計各種決策與行為，使用 Behavior Designer 能視覺化編輯行為樹，並自行擴充特殊需求的行為節點，再搭配對應的 Animator，不須更動程式碼就能製作新的敵人加速開發。

## 五、成果展示

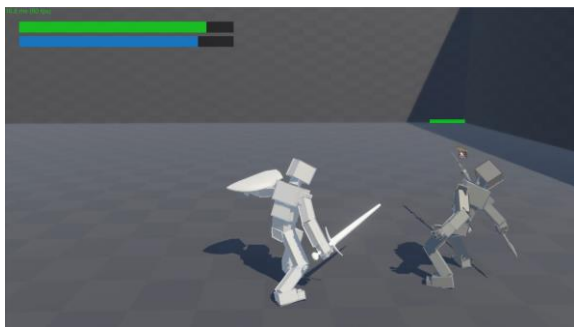


圖 1. 反擊動作成功情境

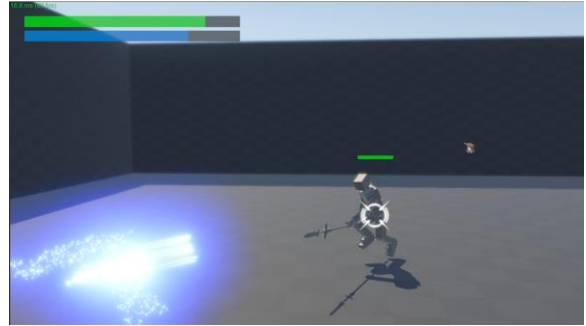


圖 2. 目標鎖定與技能施展



圖 3. Weapon Trail VFX

## 六、結論

由於是單人開發所以需要獨自解決遊戲開發各面向的需求，包含素材使用流程與美術相關效果這些非專長的部分，這是本專案主要的困難。

主系統為了方便子系統之間的呼叫以單例與中介者的概念設計，其他畫面功能的部份：使用 Blender 調整模型座標、動畫 Root Bone；使用 DoozyUI 完成 UI 事件、音效與介面切換，並配合物件池管理敵人血量條；特效的部分則使用粒子系統與 Shader Graph 製作專案所需的特效，最後整合進主系統中把整個專案完成。

## 參考文獻

[1] Unity | Documentation

<https://docs.unity3d.com/Manual/index.html>

[2] 蔡昇達(2016)，《設計模式與遊戲開發的完美結合》，新北：博碩