113程式語言期末專題構想

Unity A&B角色對戰

擲骰子定義命中A={1,2,3,4,5,6};B={1,2,3,4,5,6}

以A作為攻擊判定版型

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 旁白：對戰開始 | | | | | |
| 骰子狀況 | 判定 | 顯示 | 傷害倍率 | A產生情境 | B產生情境 |
| A>B | 命中 | Hit | 1 |  |  |
| A=6 and A>B | 暴擊 | high | 1.5 |  |  |
| A=B | 傷害減半 | low | 0.5 |  |  |
| A<B | 迴避 | Miss | 0 |  |  |
| HP歸零，角色倒下 | | | | | |
| 旁白：對戰結束 | | | | | |

A、B玩家間對戰

角色會因應狀態產生對話，觀眾對話呈現

AI間對戰

語言聊天模型

情境為A,B對戰透過投擲骰子以條件定義判斷對戰的傷害

1. B的初始HPA=50，HPB=50。

A,B擲骰子定義命中A={1,2,3,4,5,6};B={1,2,3,4,5,6} 。

Start

case01 ifA>B，

A攻擊成功，B攻擊失敗，

A普通攻擊命中B，A對B造成傷害Hit=A\*1，HPB-A\*1，HPA不變；

case02 ifA=6且A>B，

A攻擊成功且暴擊，B攻擊失敗，

A暴擊攻擊命中B，A對B造成傷害Hit=A\*1.5，HPB-A\*1.5，HPA不變；

case03 ifA=B，

A攻擊成功，B攻擊成功，

A弱化攻擊命中B，A對B造成傷害Hit=A\*0.5，HPB-A\*0.5，

B弱化攻擊命中A，B對A造成傷害Hit=B\*0.5，HPA-B\*0.5；

case04 ifA<B，

A攻擊失敗，B攻擊成功，

B普通攻擊命中A，B對A造成傷害Hit=B\*1，HPA-B\*1，HPB不變；

case05 ifA<B且B=6，

B攻擊成功且暴擊，A攻擊失敗，

B暴擊攻擊命中A，B對A造成傷害Hit=B\*1.5，HPA-B\*1.5，HPB不變；

每回合進行判定， when HPA=0 or HPB=0 結束。

藍牙骰子 實踐

**程式說明**

1. **初始化生命值**：HPA 和 HPB 各為 50。
2. **骰子擲骰**：利用 Random.Next(1, 7) 模擬骰子點數。
3. **條件判定**：依照 diceA 和 diceB 的結果執行相應的攻擊邏輯。
4. **輸出結果**：顯示每回合的攻擊傷害與剩餘生命值。
5. **結束條件**：當任一方生命值 <= 0 時結束遊戲並顯示勝利者。

將此程式貼至 Visual Studio Code，使用 Unity 的 C# 編譯器即可執行測試。

using System;

public class DiceBattle

{

static void Main(string[] args)

{

// 初始化角色生命值//

double HPA = 50, HPB = 50;

Random random = new Random();

Console.WriteLine("Dice Battle Start!");

while (HPA > 0 && HPB > 0)

{

// 擲骰子//

int diceA = random.Next(1, 7); // A 擲骰 (1-6)

int diceB = random.Next(1, 7); // B 擲骰 (1-6)

Console.WriteLine($"\nA rolled: {diceA}, B rolled: {diceB}");

// 判定攻擊結果//

if (diceA > diceB)

{

// Case 01: A 普通攻擊//

if (diceA == 6)

{

// Case 02: A 暴擊攻擊//

double damage = diceA \* 1.5;

HPB -= damage;

Console.WriteLine($"A lands a critical hit! Damage: {damage:F1}. HPB: {HPB:F1}");

}

else

{

double damage = diceA \* 1;

HPB -= damage;

Console.WriteLine($"A lands a normal hit! Damage: {damage:F1}. HPB: {HPB:F1}");

}

}

else if (diceA == diceB)

{

// Case 03: 雙方弱化攻擊//

double damageA = diceA \* 0.5;

double damageB = diceB \* 0.5;

HPA -= damageB;

HPB -= damageA;

Console.WriteLine($"Both hit! A deals {damageA:F1} to B. B deals {damageB:F1} to A.");

Console.WriteLine($"HPA: {HPA:F1}, HPB: {HPB:F1}");

}

else

{

// Case 04: B 普通攻擊//

if (diceB == 6)

{

// Case 05: B 暴擊攻擊//

double damage = diceB \* 1.5;

HPA -= damage;

Console.WriteLine($"B lands a critical hit! Damage: {damage:F1}. HPA: {HPA:F1}");

}

else

{

double damage = diceB \* 1;

HPA -= damage;

Console.WriteLine($"B lands a normal hit! Damage: {damage:F1}. HPA: {HPA:F1}");

}

}

// 判斷遊戲是否結束//

if (HPA <= 0)

{

Console.WriteLine("\nB wins! HPA reached 0.");

break;

}

if (HPB <= 0)

{

Console.WriteLine("\nA wins! HPB reached 0.");

break;

}

}

Console.WriteLine("Game Over!");

}

}

**撰寫角色邏輯腳本**

**將 Dice Battle 的邏輯改造成 Unity 腳本，並將它掛載到角色身上。**

**腳本一：PlayerStats.cs**

**這個腳本負責角色的基礎屬性和受傷邏輯。**

using UnityEngine;

public class PlayerStats : MonoBehaviour

{

// 角色初始 HP//

public float health = 50;

// 更新生命值，若 HP <= 0，角色死亡//

public void TakeDamage(float damage)

{

health -= damage;

if (health <= 0)

{

health = 0;

Debug.Log($"{gameObject.name} is defeated!");

}

}

}

** 將此腳本分別掛載到 PlayerA 和 PlayerB。**

** 可以在 Unity 編輯器中設定初始生命值（health）。**

**腳本二：DiceBattle.cs**

**這個腳本負責骰子對戰的邏輯和角色互動。**

using UnityEngine;

public class DiceBattle : MonoBehaviour

{

public GameObject playerA;

public GameObject playerB;

private PlayerStats statsA;

private PlayerStats statsB;

void Start()

{

// 取得角色的 PlayerStats 腳本//

statsA = playerA.GetComponent<PlayerStats>();

statsB = playerB.GetComponent<PlayerStats>();

// 開始戰鬥回合//

StartBattle();

}

void StartBattle()

{

while (statsA.health > 0 && statsB.health > 0)

{

// A 擲骰//

int diceA = Random.Range(1, 7);

// B 擲骰//

int diceB = Random.Range(1, 7);

Debug.Log($"A rolled: {diceA}, B rolled: {diceB}");

if (diceA > diceB)

{

if (diceA == 6)

{

// A 暴擊攻擊//

float damage = diceA \* 1.5f;

statsB.TakeDamage(damage);

Debug.Log($"A lands a critical hit! Damage: {damage}. B HP: {statsB.health}");

}

else

{

// A 普通攻擊//

float damage = diceA;

statsB.TakeDamage(damage);

Debug.Log($"A lands a normal hit! Damage: {damage}. B HP: {statsB.health}");

}

}

else if (diceA == diceB)

{

// 雙方弱化攻擊//

float damageA = diceA \* 0.5f;

float damageB = diceB \* 0.5f;

statsA.TakeDamage(damageB);

statsB.TakeDamage(damageA);

Debug.Log($"Both hit! A deals {damageA} to B. B deals {damageB} to A.");

Debug.Log($"A HP: {statsA.health}, B HP: {statsB.health}");

}

else

{

if (diceB == 6)

{

// B 暴擊攻擊//

float damage = diceB \* 1.5f;

statsA.TakeDamage(damage);

Debug.Log($"B lands a critical hit! Damage: {damage}. A HP: {statsA.health}");

}

else

{

// B 普通攻擊//

float damage = diceB;

statsA.TakeDamage(damage);

Debug.Log($"B lands a normal hit! Damage: {damage}. A HP: {statsA.health}");

}

}

// 檢查遊戲是否結束//

if (statsA.health <= 0)

{

Debug.Log("Player B wins!");

break;

}

if (statsB.health <= 0)

{

Debug.Log("Player A wins!");

break;

}

}

}

}

** 將此腳本掛載到一個空物件（例如 GameManager）。**

** 在 Unity 編輯器中，將 PlayerA 和 PlayerB 分別拖曳到腳本的 playerA 和 playerB 欄位。**

**3. 為角色添加生命條**

**選項 1：簡單生命條**

1. 新建一個 UI Slider，作為角色的生命條，放置在場景中。
2. 撰寫以下腳本同步更新生命條：

using UnityEngine;

using UnityEngine.UI;

public class HealthBar : MonoBehaviour

{

public Slider healthSlider;

public PlayerStats playerStats;

void Update()

{

// 假設初始 HP = 50//

healthSlider.value = playerStats.health / 50f;

}

}

1. 將此腳本掛載到生命條，並將 healthSlider 和對應的角色拖入。

相關規劃

1. 角色選定(半獸人) done
2. 動畫屬性呈現(攻擊) done
3. 控制器 done
4. 骰子規則撰寫 done
5. 判定框架 done
6. Code放置colab偵測 done
7. 對話框文字輸入 透過字元偵測為攻擊指令
8. 攻擊前後觀眾聊天 透過血量高低、支持者、攻擊判定等識別生成
9. 藍頻問題 不可控….
10. 構想討論 待回復….

--------------------------------------------------------------------------------

藍頻場景消失修正排除 重製拉場景跟角色(11/28)

一張含有 文字, 螢幕擷取畫面, 多媒體軟體, 軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 多媒體軟體, 軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 軟體, 多媒體軟體 的圖片

自動產生的描述一張含有 文字, 螢幕擷取畫面, 字型 的圖片

自動產生的描述

建議修正 待回復

一張含有 文字, 螢幕擷取畫面, 軟體, 作業系統 的圖片

自動產生的描述