



<第一人稱迷宮探索遊戲>

指導老師：呂威甫老師

組員：徐奕維、何彥呈

在這個世代，消費型電子遊樂軟體基本上充斥著我們的休閒項目，很多人的娛樂活動是玩遊戲，而作為許多初學者最初使用的遊戲引擎便屬Unity莫屬。它好上手，不只支援的平台多，且能獲得的資訊也非常充足。這些優點是我們考慮後所選擇Unity的原因，比起UE5來說對我們更加友善。因此我們以Unity來做為主要的開發軟體，來開發出一款屬於我們設計的遊戲。

作為單機版解謎探險遊戲，第一人稱給人的畫面帶入感是最重要的，許多玩家看重的不是遊玩性高不高，而是遊戲氣氛。我們使用的第一人稱模組是在Asset Store上找尋的簡易版(圖九)，然後再修改碰撞格參數，如圖(圖十)，音效則是使用內建的腳步聲。手電筒燈光是用Light內新增一個Point Light，再將光源縮小並更改顏色與亮度



第一人稱視角位移的程式碼

```
using UnityEngine;
using System.Collections;

public class FirstPersonLook : MonoBehaviour
{
    [SerializeField]
    Transform character;
    public float sensitivity = 2;
    public float smoothing = 0.5f;

    Vector3 rotation;
    void Start()
    {
        // Get the character from the FirstPersonController in scene.
        character = GameObject.FindObjectOfType<FirstPersonController>.transform;
    }

    void Update()
    {
        // Turn the main camera to the gaze vector.
        Vector3 lookDir = Quaternion.LookRotation(character.forward, character.up);

        // Rotate camera to look.
        rotation = new Vector3(rotation.eulerAngles.x, lookDir.eulerAngles.y, rotation.eulerAngles.z);
        Quaternion targetRotation = Quaternion.Euler(rotation);
        rotation = Vector3.Lerp(rotation, targetRotation.eulerAngles, smoothing);
        character.localEulerAngles = rotation;
    }
}
```

第一人稱按鍵移動的程式碼

```
using UnityEngine;
using System.Collections;

public class FirstPersonMove : MonoBehaviour
{
    public float speed = 5;

    [SerializeField]
    public bool isGrounded = true;

    public float jumping = 0;
    public float jumping2 = 0;
    public float jumping3 = 0;
    public float jumping4 = 0;

    Rigidbody rigidBody;
    // Ground on Physics to enable correct speed. Fill in the last value with scene.
    public LayerMask groundLayers = new LayerMask(-1);

    void Start()
    {
        // Get the rigidBody on this.
        rigidBody = GetComponent<Rigidbody>();

        // Rotate camera to look.
        rotation = new Vector3(rotation.eulerAngles.x, rotation.eulerAngles.y, rotation.eulerAngles.z);
        Quaternion targetRotation = Quaternion.Euler(rotation);
        rotation = Vector3.Lerp(rotation, targetRotation.eulerAngles, smoothing);
        character.localEulerAngles = rotation;
    }

    void Update()
    {
        // Rotate camera to look.
        rotation = new Vector3(rotation.eulerAngles.x, rotation.eulerAngles.y, rotation.eulerAngles.z);
        Quaternion targetRotation = Quaternion.Euler(rotation);
        rotation = Vector3.Lerp(rotation, targetRotation.eulerAngles, smoothing);
        character.localEulerAngles = rotation;
    }

    // Rotate camera to look.
    rotation = new Vector3(rotation.eulerAngles.x, rotation.eulerAngles.y, rotation.eulerAngles.z);
    Quaternion targetRotation = Quaternion.Euler(rotation);
    rotation = Vector3.Lerp(rotation, targetRotation.eulerAngles, smoothing);
    character.localEulerAngles = rotation;
}
```

