

# 2D橫向卷軸動作遊戲

組員:105021020 姚俊榮

105021103 陳亮宇

105021048 李昂霖

指導老師:何承遠 教授

# 目錄

- 一. 研究動機
- 二. 使用軟體介紹
- 三. 作品內容
- 四. 總結
- 五. 未來展望

## ► 研究動機

在這個科技日益精進的時代，人們的生活與電子遊戲已有著密不可分的关系。而組員們有著一份對遊戲的熱愛，使得我們下定決心要創造一個屬於自己的遊戲，一個大家喜歡且符合組員們心目中各種想法的作品，這是製作這個遊戲的原動力，也是驅使我們願意持續去精進技術的目標。

作為喜愛遊戲的大學生，我們也想自己製作一些遊戲來與大家一起分享，因此希望能開發出一款遊戲置應用程式以供玩樂。然而面對不同的群體，我們也希望製作出一個不只能玩、能動腦甚至能帶給人們更多的快樂的遊戲。



## ► 使用軟體介紹

### Unity

Unity 是一款由 Unity Technologies 研發的跨平台 2D / 3D 遊戲引擎，可用於開發 Windows、MacOS 及 Linux 平台的單機遊戲，PlayStation、XBox、Wii、3DS 和任天堂 Switch 等遊戲主機平台的電動遊戲，或是 iOS、Android 等行動裝置的遊戲。Unity 所支援的遊戲平台還延伸到了基於 WebGL 技術的 HTML5 網頁平台，以及 tvOS、Oculus Rift、ARKit 等新一代多媒體平台。除可以用於研發電子遊戲之外，Unity 還是被廣泛用於建築視覺化、即時三維動畫等類型互動內容的綜合型創作工具。

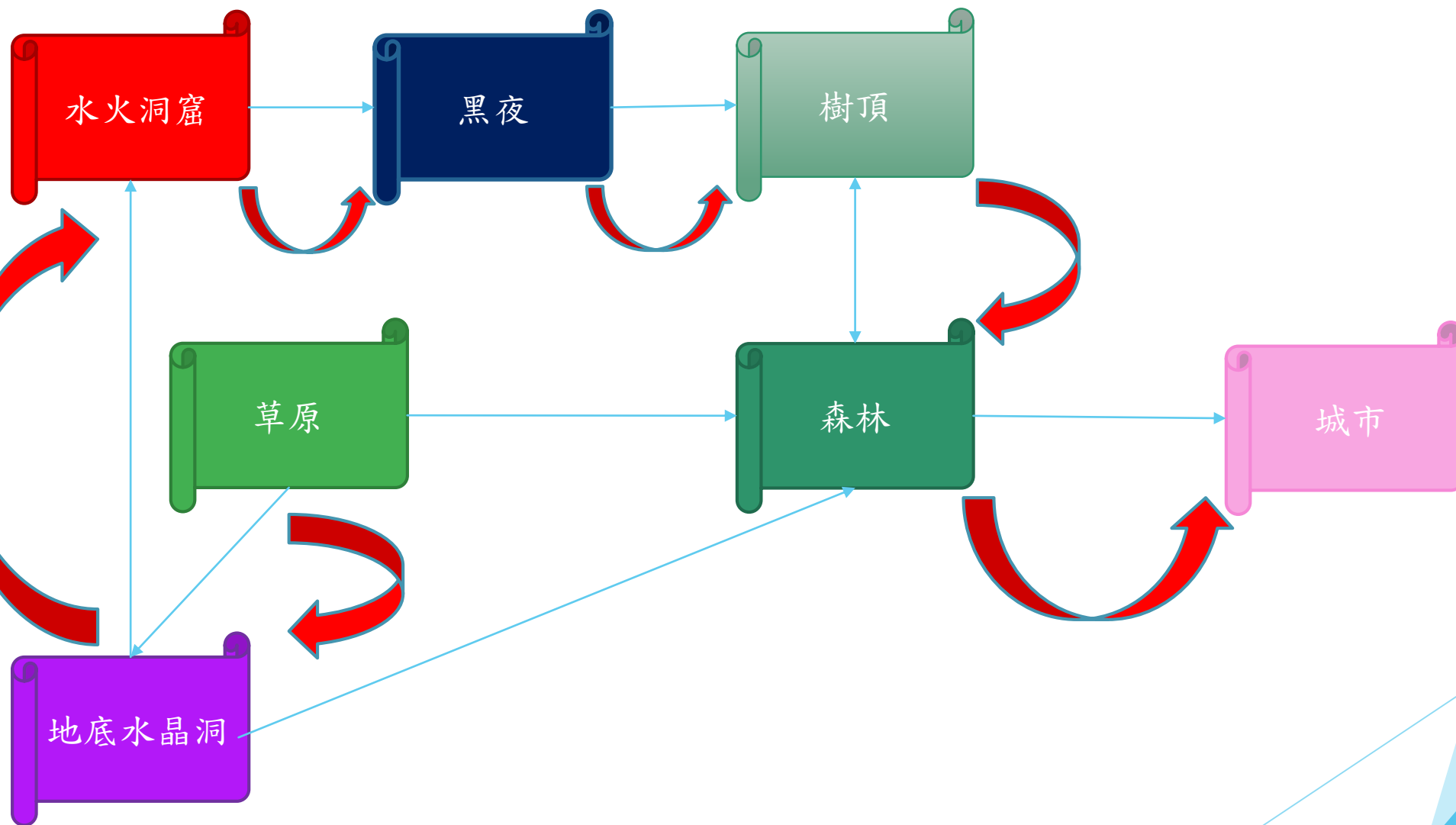


## ▶ 使用軟體介紹

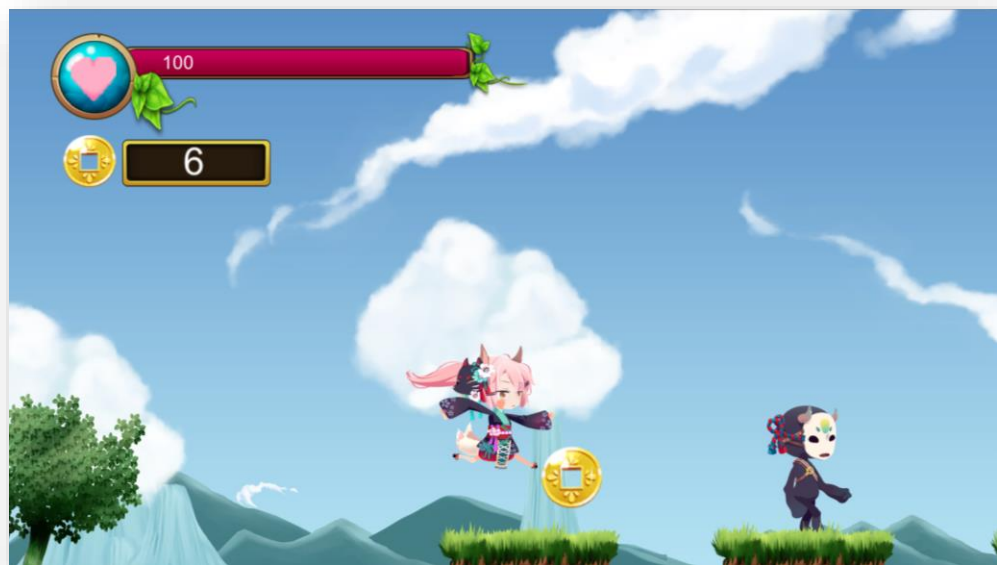
### Visual Studio

Visual Studio是微軟公司的開發工具套件系列產品。VS是一個基本完整的開發工具集，它包括了整個軟體生命週期中所需要的大部分工具，如UML工具、程式碼管控工具、整合開發環境（IDE）等等。所寫的目的碼適用於微軟支援的所有平台

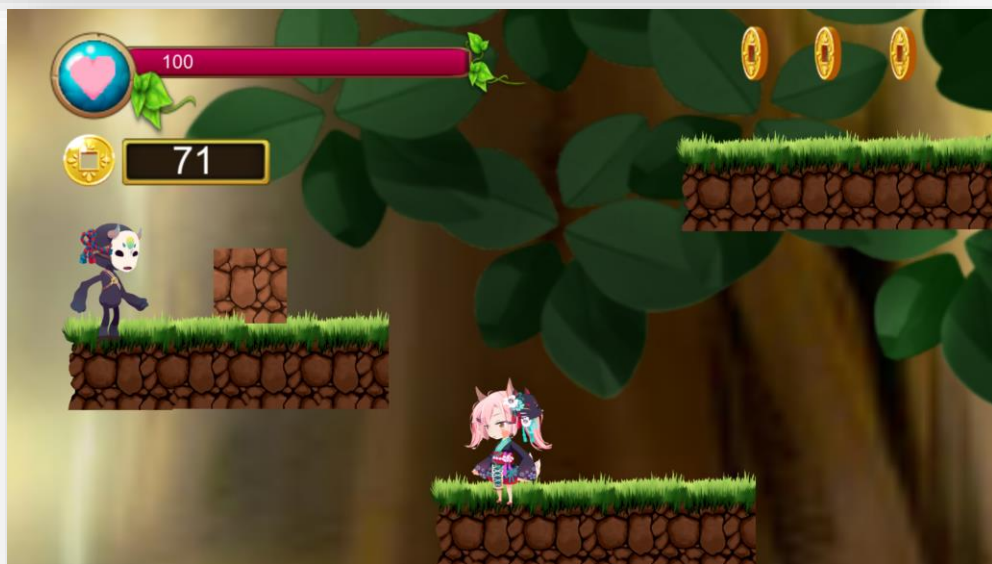
## ▶ 遊戲內容(地圖)



## ▶ 遊戲內容(地圖)



草原



森林

## ▶ 遊戲內容(地圖)



城市



地底水晶洞



## ▶ 遊戲內容(地圖)

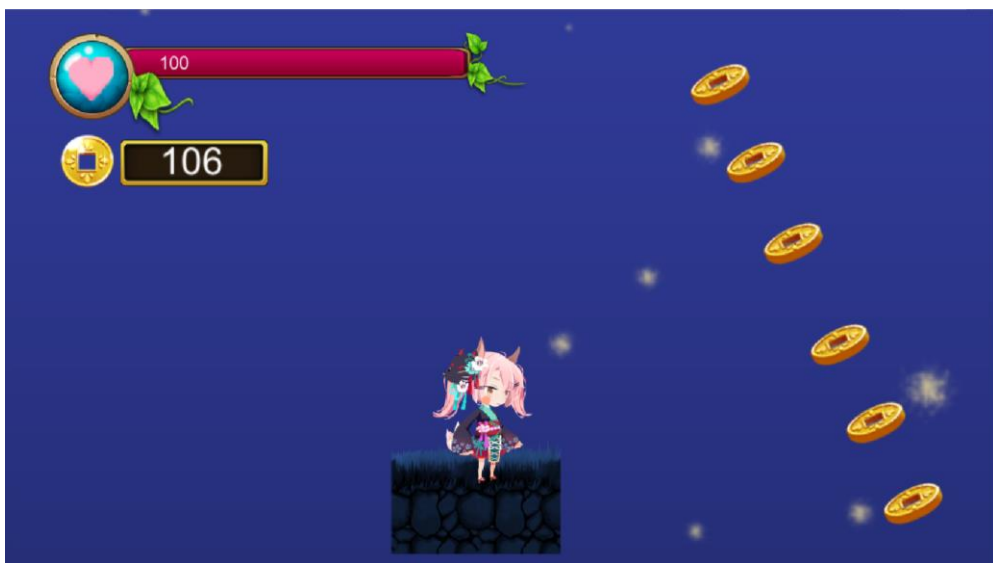


樹頂



水火洞窟

## ▶ 遊戲內容(地圖)



黑夜

# ▶ 遊戲內容(程式碼)

## EnemyAI

```
1 using UnityEngine;
2 using System.Collections;
3 using System.Collections.Generic;
4
5 public class EnemyAI : MonoBehaviour
6 {
7     public float speed; //設定speed
8     [Range(0, 0.5f)] //檢查線長度
9     public float distance;
10    private bool movingRight = true; //設定movingRight為true
11
12    //public Transform aiDetection;
13    public Transform groundDetection; //地面檢查線
14
15    void Update()
16    {
17        Vector2 start = groundDetection.position; //檢查線開始位置
18        Vector2 end = new Vector2(start.x, start.y - distance); //檢查線結束位置
19        Debug.DrawLine(start, end, Color.blue); //顯示檢查線
20        transform.Translate(Vector2.right * speed * Time.deltaTime); //向右移動
21
22        RaycastHit2D groundInfo = Physics2D.Raycast(groundDetection.position, Vector2.down, distance);
23        if (groundInfo.collider == false && movingRight == true) //檢查是否有地面並且movingRight為true
24        {
25            transform.eulerAngles = new Vector3(0, 180, 0); //敵人轉向
26            movingRight = false; //設定movingRight為false
27        }
28        else if (groundInfo.collider == false && movingRight == false) //檢查是否有地面並且movingRight為false
29        {
30            transform.eulerAngles = new Vector3(0, 0, 0); //敵人轉向
31            movingRight = true; //設定movingRight為true
32        }
33    }
34
35    void OnCollisionEnter2D(Collision2D collision)
36    {
37        if (collision.gameObject.tag == "enemy") //當碰撞到物件的tag為enemy
38        {
39            if (movingRight == true) //檢查movingRight是否為true
40            {
41                transform.eulerAngles = new Vector3(0, 180, 0); //敵人轉向
42                movingRight = false; //設定movingRight為false
43            }
44            else if (movingRight == false) //檢查movingRight是否為false
45            {
46                transform.eulerAngles = new Vector3(0, 0, 0); //敵人轉向
47                movingRight = true; //設定movingRight為true
48            }
49        }
50    }
51 }
```

## ▶ 遊戲內容(程式碼)

door

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.SceneManagement;
5
6  public class door : MonoBehaviour
7  {
8
9      [Header("連接到某場景")]
10
11     public string goToTheScene;
12     void OnTriggerEnter2D(Collider2D other) //判斷是否觸發
13     {
14         if (other.CompareTag("player")) //觸發物件的tag為player
15         {
16             SceneManager.LoadScene(goToTheScene); //讀取指定場景
17         }
18     }
19 }
20
```

## ▶ 遊戲內容(程式碼)

### GameDataManager

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public static class GameDataManager
6 {
7     public static float hitpoint = 100f; //設定全域變數hitpoint
8     public static int Score; //設定全域變數Score
9 }
10
```

### FloorTriggerZone

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class FloorTriggerZone : MonoBehaviour
6 {
7     public bool isDamaging; //設定bool isDamaging
8     public float damage = 100; //設定damage
9
10    private void OnTriggerEnter2D(Collider2D other) //判斷是否觸發
11    {
12        if (other.tag == "player") //觸發物件的tag為player
13            other.SendMessage((isDamaging?"TakeDamage" : "HealDamage", Time.deltaTime * damage); //判定是否為isDamaging 計算受到傷害或回復生命
14
15        if (other.gameObject.tag == "player") //觸發物件的tag為player
16            other.gameObject.GetComponent<play>().alive = false; //取得觸發事件的物件中的程式判斷alive是否為false
17    }
18 }
19
```

## ► 總結

我們的專題是以2D的方式來製作，在製作過程中有時會發現一些製作上的設計與程式並不相容，而使遊戲會發生一些失誤，而將遊戲從零開始製作到遊戲可以運行後，給其他的人遊玩時而有人覺得好玩時，我們也逐漸的從中得到成就感。

## ▶ 未來展望

- 一. 角色動作可以更加流暢
- 二. 有不一樣的角色造型供選擇
- 三. 能有多元、不同種類敵人
- 四. 讓角色、敵人能夠攻擊，也能進行防禦