



亞洲大學畢業專題報告

Unity遊戲製作 - 2.5D迷宮生存遊戲

指導教授:何承遠教授

組員: 任韋成
林昱成
吳濬宏
林郁憲

目錄

CONTENTS

01

研究動機

02

構想

03

進度安排

04

功能介紹

05

未來展望

06

參考資料



研究動機

Research Motivation



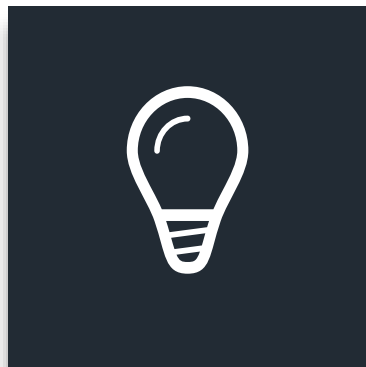
研究動機

Research motivation



大家決定好要一起製作我們的畢業專題之後，便開始討論要製作的內容，教授給我們幾個建議，像是公車路線的研究、網路伺服器方面的研究，以及遊戲製作，經過我們討論後，最後選擇做遊戲的原因是我們大家都喜歡玩電腦和手機遊戲，想要了解與我們生活息息相關的娛樂(遊戲)是如何製作出來的，於是我們便完成了我們的第一步，設定我們要研究的目標。





構想

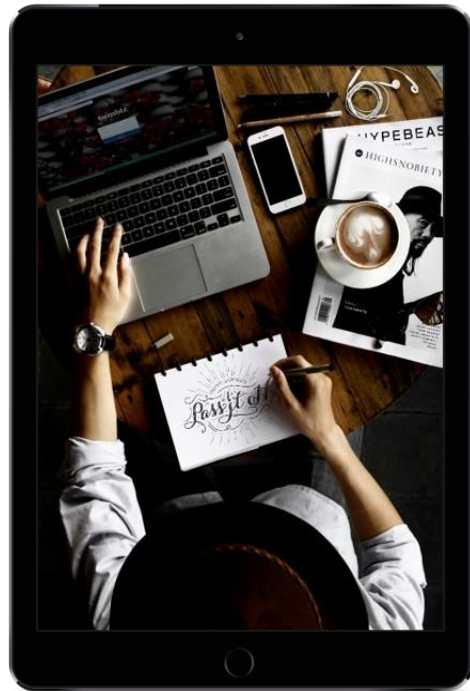
Idea

構想

Idea



一開始我們在討論要設計甚麼類型的遊戲，大家都有不同的想法像是卡牌遊戲、2D的平面闖關遊戲、以及2.5D的生存遊戲，最後我們選擇了2.5D的生存遊戲，因為我們認為2.5D遊戲比較有挑戰性，其他兩種類型主要是功能構想，我們希望程式構置的部分多一些並開始慢慢構想我們的遊戲最終目標。





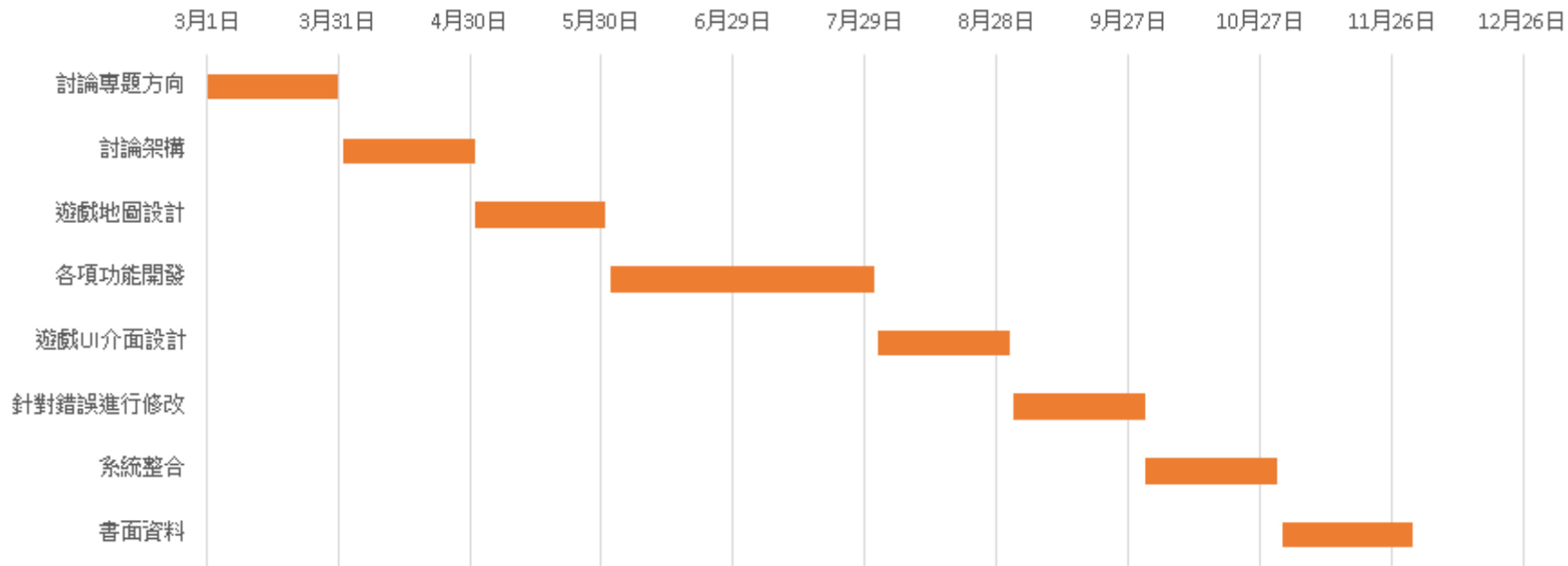
進度安排

Schedule



進度安排

Schedule





功能介紹

Function Introduction

☑ 使用工具 ☑ 遊戲玩法 ☑ 遊戲介紹

使用工具

Tools used



我們會選擇Unity的原因很簡單，因為我們組員們都是第一次接觸遊戲製作，初階遊戲製作的比較適合的選擇就是Unity，而且他的有很多內建的功能對於我們初學者來說都很友善，像是他有免費的模組可以讓我們選擇，對於不會建模組的我們來說有很大的幫助，還有很多說不完的優點。





遊戲玩法介紹

Playing Instruction

W

向前移動

A

向左移動

S

向後移動

D

向右移動

Space

發射子彈

J

角色加速



遊戲介紹

Game Introduction



人物正面



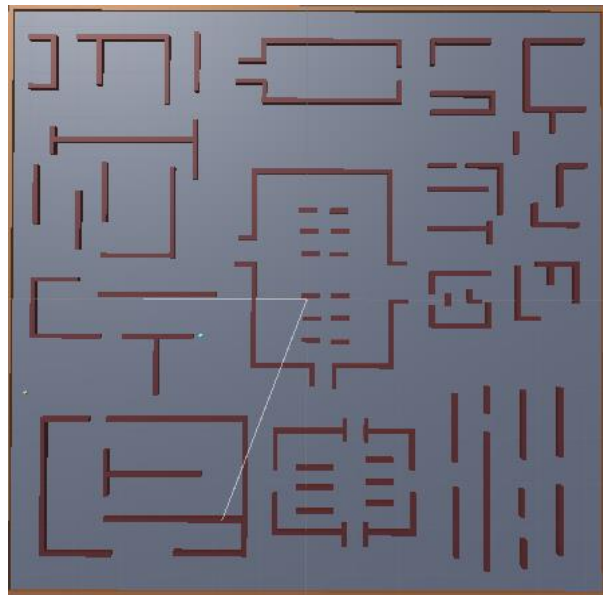
人物背面



人物側面



殭屍



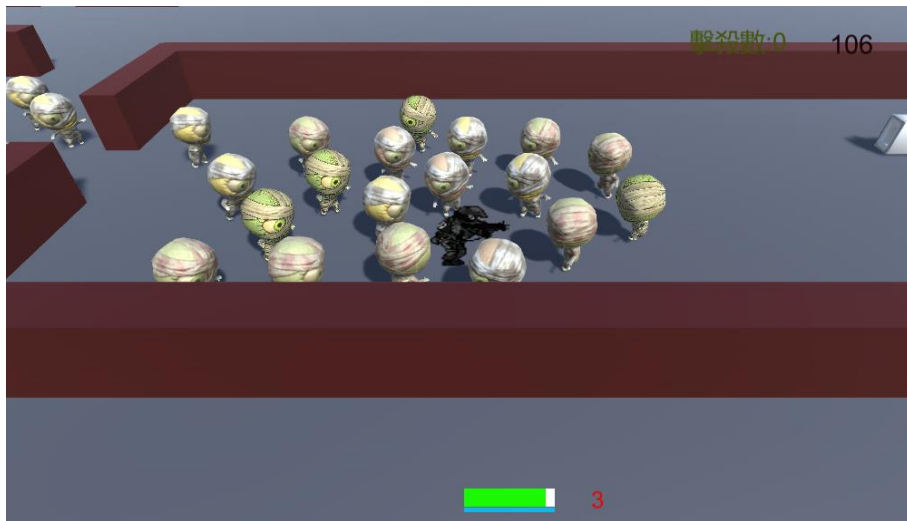
地圖全貌





遊戲介紹

Game Introduction



遊戲開始時，怪物會隨機生成，並且持續追蹤玩家，直到怪物被消滅或玩家死亡。



遊戲介紹

Game Introduction

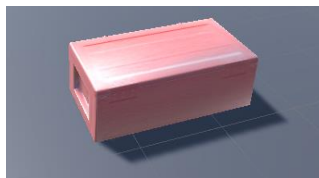


玩家可以透過發射子彈消滅怪物。



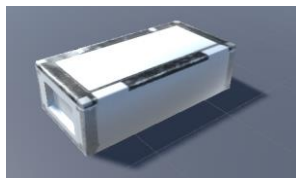
遊戲介紹

Game Introduction



醫療箱

撿到醫療箱
玩家會回復
10點血量



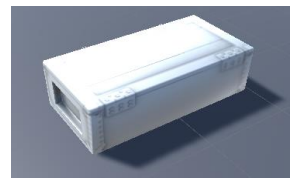
時間箱

撿到時間箱
遊戲剩餘時
間會減少5
秒鐘



能量箱

撿到能量箱
玩家會回復
10點能量



攻擊箱

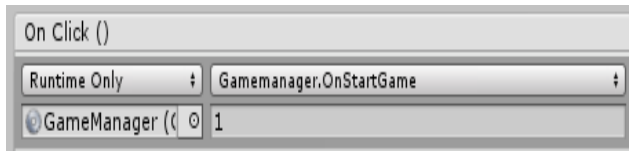
撿到攻擊箱
玩家會增加
10點攻擊力





遊戲介紹

Game Introduction



```
1
2  using System.Collections;
3  using System.Collections.Generic;
4  using UnityEngine;
5
6  public class GameManager : MonoBehaviour
7  {
8
9
10     public void OnStartGame(int SceneNumber)
11     {
12         Application.LoadLevel(SceneNumber);
13     }
14 }
15
```

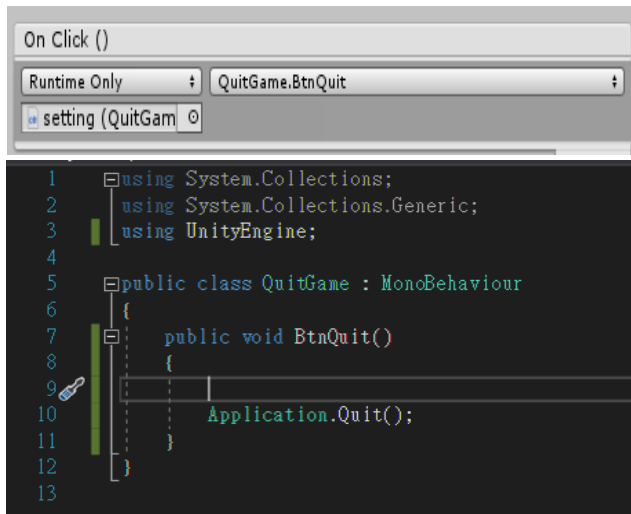
進入遊戲後的主畫面，可以在這裡看到遊戲的相關公告並控制主畫面的背景音量。





遊戲介紹

Game Introduction



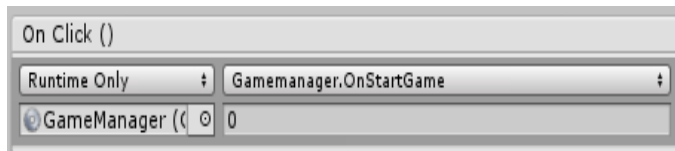
離開遊戲





遊戲介紹

Game Introduction



```
1  
2    using System.Collections;  
3    using System.Collections.Generic;  
4    using UnityEngine;  
5  
6    public class GameManager : MonoBehaviour  
7    {  
8  
9       public void OnStartGame(int SceneNumber)  
10       {  
11           Application.LoadLevel(SceneNumber);  
12       }  
13    }  
14  
15
```

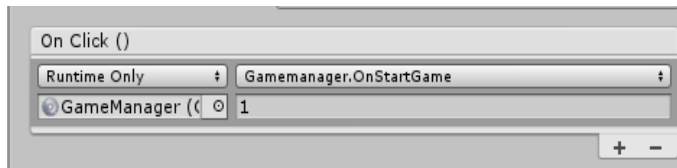
當玩家死亡後會跳出此畫面，選擇主畫面就會跳回到主畫面，按下重新開始後，可以快速再重新進行一局遊戲，按下離開遊戲的按鈕會直接關閉遊戲。





遊戲介紹

Game Introduction



```
1  
2    using System.Collections;  
3    using System.Collections.Generic;  
4    using UnityEngine;  
5  
6    public class GameManager : MonoBehaviour  
7    {  
8  
9       public void OnStartGame(int SceneNumber)  
10       {  
11           Application.LoadLevel(SceneNumber);  
12       }  
13    }  
14  
15
```

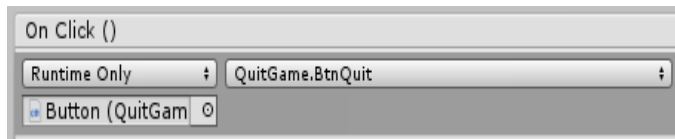
重新開始





遊戲介紹

Game Introduction



```
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class QuitGame : MonoBehaviour
6  {
7      public void BtnQuit()
8      {
9          Application.Quit();
10     }
11 }
12
13
```

離開遊戲





遊戲介紹

Game Introduction



```
//將綠色血條同步到當前血量長度
HealthBarP3.sizeDelta = new Vector2(currentHealth, HealthBarP3.sizeDelta.y);
Speedtime.sizeDelta = new Vector2(currentSpeed, Speedtime.sizeDelta.y);

if (HurtP3.sizeDelta.x > HealthBarP3.sizeDelta.x)
{
    //讓傷害量(紅色血條)逐漸追上當前血量

    HurtP3.sizeDelta += new Vector2(-1, 0) * Time.deltaTime * 15;
    if (currentHealth >= 100)
    {
        HurtP3.sizeDelta = new Vector2(currentHealth, HurtP3.sizeDelta.y);
    }
}
```

```
if (other.gameObject.tag == "Monster")
{
    if (start_Timer3)
    {
        StartCoroutine("timer3");//使用timer3的函式讓扣血時間有所間隔
        start_Timer3 = false;//當扣血後會讓start_Timer3=false 約1秒後會轉為True

        currentHealth = currentHealth - 10;
        currentHealth = Mathf.Clamp(currentHealth, 0, 100);
        Debug.Log(timer_i);
    }
    if (currentHealth <= 0)
    {
        HealthBarP3.sizeDelta = new Vector2(currentHealth, HealthBarP3.sizeDelta.y);
        Speedtime.sizeDelta = new Vector2(currentSpeed, Speedtime.sizeDelta.y);
        HurtP3.sizeDelta = new Vector2(currentHealth, HurtP3.sizeDelta.y);
        Destroy(gameObject);
    }
}
```

角色血條程式碼設定





遊戲介紹

Game Introduction



```
//子彈的發射點  
gun1 = GameObject.Find("gun1");  
gun2 = GameObject.Find("gun2");  
gun3 = GameObject.Find("gun3");  
gun4 = GameObject.Find("gun4");
```

角色子彈發射方位

```
void bullw() {  
    GameObject Bullet = Instantiate(GameObject.Find("bulletw")); //複製場景上一個叫做bulletw的物件  
    Bullet.transform.position = gun1.transform.position; //移動剛剛複製的物件至玩家前方的位置上  
    Bullet.AddComponent<BulletForword>(); //將剛剛複製的物件掛載名叫BulletForword的Script  
    Destroy(Bullet, 5); //5秒後將剛剛複製的ball移除  
}  
void bulls(){  
    GameObject Bullet = Instantiate(GameObject.Find("bullets")); //複製場景上一個叫做bullets的物件  
    Bullet.transform.position = gun3.transform.position; //移動剛剛複製的物件至玩家後方的位置上  
    Bullet.AddComponent<BulletBack>(); //將剛剛複製的物件掛載名叫BulletBack的Script  
    Destroy(Bullet, 5); //5秒後將剛剛複製的ball移除  
}  
void bulla() {  
    GameObject Bullet = Instantiate(GameObject.Find("bulleta")); //複製場景上一個叫做bulleta的物件  
    Bullet.transform.position = gun4.transform.position; //移動剛剛複製的物件至玩家左方的位置上  
    Bullet.AddComponent<BulletLeft>(); //將剛剛複製的物件掛載名叫BulletLeft的Script  
    Destroy(Bullet, 5); //5秒後將剛剛複製的ball移除  
}  
void bulld(){  
    GameObject Bullet = Instantiate(GameObject.Find("bulletd")); //複製場景上一個叫做bulletd的物件  
    Bullet.transform.position = gun2.transform.position; //移動剛剛複製的物件至玩家右方的位置上  
    Bullet.AddComponent<BulletRight>(); //將剛剛複製的物件掛載名叫BulletRight的Script  
    Destroy(Bullet, 5); //5秒後將剛剛複製的ball移除  
}
```

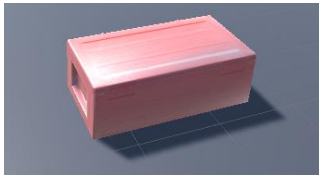
子彈發射





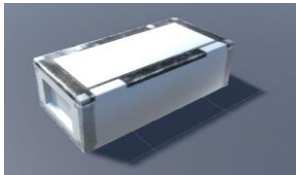
遊戲介紹

Game Introduction



醫療箱

```
if (other.gameObject.tag == "LifeBox")
{
    if (currentHealth < 100)
    {
        currentHealth += 10;
    }
    Destroy(other.gameObject);
}
if (other.gameObject.tag == "TimeBox")
{
    if (retime > 2)
    {
        retime -= 2;
        Destroy(other.gameObject);
    }
}
```

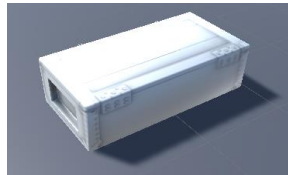


時間箱



能量箱

```
if (other.gameObject.tag == "SpeedBox")
{
    if (currentSpeed < 100)
    {
        currentSpeed += 10;
    }
    Destroy(other.gameObject);
}
if (other.gameObject.tag == "AttackBox")
{
    CancelInvoke("Atktime");//時間記錄停止
    Attacktime = 10;
    InvokeRepeating("Atktime", 1, 1); //一秒後，每秒重複呼叫timer函式。
    MonsterAttack.Atk = 20;
    Destroy(other.gameObject);
}
```



攻擊箱



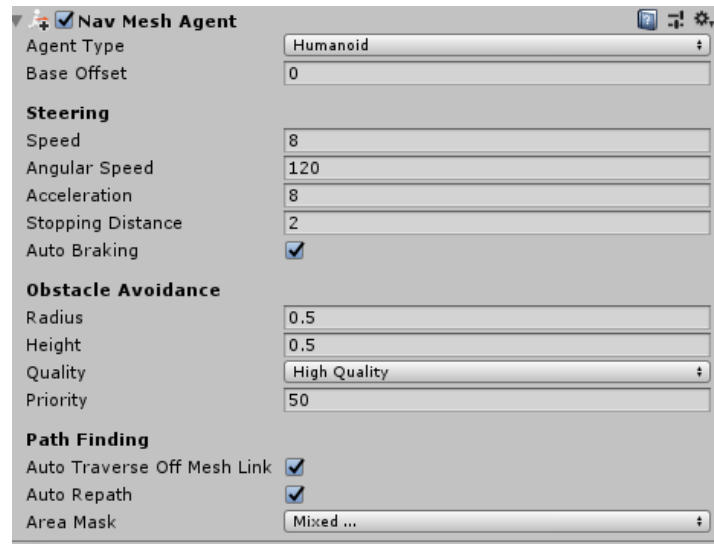


遊戲介紹

Game Introduction



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.AI;
5
6 public class monster1 : MonoBehaviour
7 {
8     public NavMeshAgent agent;
9     public GameObject target_obj;
10
11     // Start is called before the first frame update
12     void Start()
13     {
14         agent = GetComponent<NavMeshAgent>();
15     }
16
17
18
19     // Update is called once per frame
20     void Update()
21     {
22         agent.SetDestination(target_obj.transform.position);
23     }
24
25
26 }
```



怪物追蹤程式碼與設定





未來展望

Future Prospect

☑ 未來研究方向



未來研究方向

Research Objective

玩家模組更多樣

未來能夠增加更多角色的模組供玩家使用。

遊戲玩法更多元

設計更多的地圖與道具功能讓遊戲玩法更豐富。

增加多人模式

新增連線模式，讓玩家們能夠同時在一場遊戲裡進行合作或是對戰。



參考資料

<https://unity.com/>

山姆遊戲 <http://sammaru.blogspot.com/>

CG數位學習網 <http://www.cg.com.tw/Unity/>

米飯教學室 <https://www.youtube.com/user/smallricetw>

巴哈姆特Unity3D遊戲引擎

<https://forum.gamer.com.tw/A.php?bsn=60602>



感謝各位教授的批評指正

THANK YOU FOR WATCHING

報告人:任韋成，林昱成，吳濬宏，林郁憲

