

# 臺北市立大安高級工業職業學校專題製作報告

群 別： 資電群

參賽作品名稱： PIXEL ADVENTURE

關 鍵 詞： 2D 遊戲、covid-19

指導老師： 沈彥良

班 級： 綜高三愛

組 別： G110C01

組 長： 07 楊榮鈞

組 員： 03 邱于哲、05 溫家茂、11 駱懷安

中 華 民 國 一 一 〇 年 一 月 七 日

## 目錄

壹、	前言 .....	1
一、	製作動機 .....	1
二、	製作目的 .....	1
貳、	設備與軟體相關知識.....	2
一、	設備與軟體 .....	2
(一)	Unity .....	2
(二)	Visual Studio .....	3
(三)	Aseprite .....	3
參、	製作過程 .....	4
一、	場景設計 .....	4
二、	故事動畫 .....	6
三、	角色控制 .....	7
四、	怪物控制 .....	9
五、	物件繪製 .....	10
六、	背景音樂 .....	11
七、	整體程式架構 .....	11
八、	程式設計流程 .....	11
肆、	成果展示與討論 .....	12
一、	成果展示 .....	12
二、	問題與討論 .....	14
伍、	結論.....	15
陸、	參考資料.....	16
柒、	部分專題素材 .....	16

# 壹、前言

## 一、製作動機

在剛開始知道要專題製作時，組員們都在思考要往哪個方向製作，因為知道大方向是朝軟體遊戲方面，所以大家就開始想哪一類型的遊戲是我們能力所及又有趣，經過討論組員們決定製作類似遊戲天堂的2D小遊戲。

## 二、製作目的

由於2020這一年新冠肺炎疫情大爆發，衝擊全球世界經濟，造成人類恐慌，疫情嚴峻時期防疫觀念極其重要，因此此遊戲主軸就圍繞著這次疫情做出有教育意義的遊戲，又可以跟上時事。

## 貳、設備與軟體相關知識

### 一、設備與軟體

專題中所使用的設備與主要軟體詳如表 1。

表 1 設備與軟體

設備	軟體	語法
筆電	Unity	C#
	Visual Studio	
	Aseprite	

#### (一) Unity

Unity如圖1所示，是一個開發2D或3D的遊戲引擎。

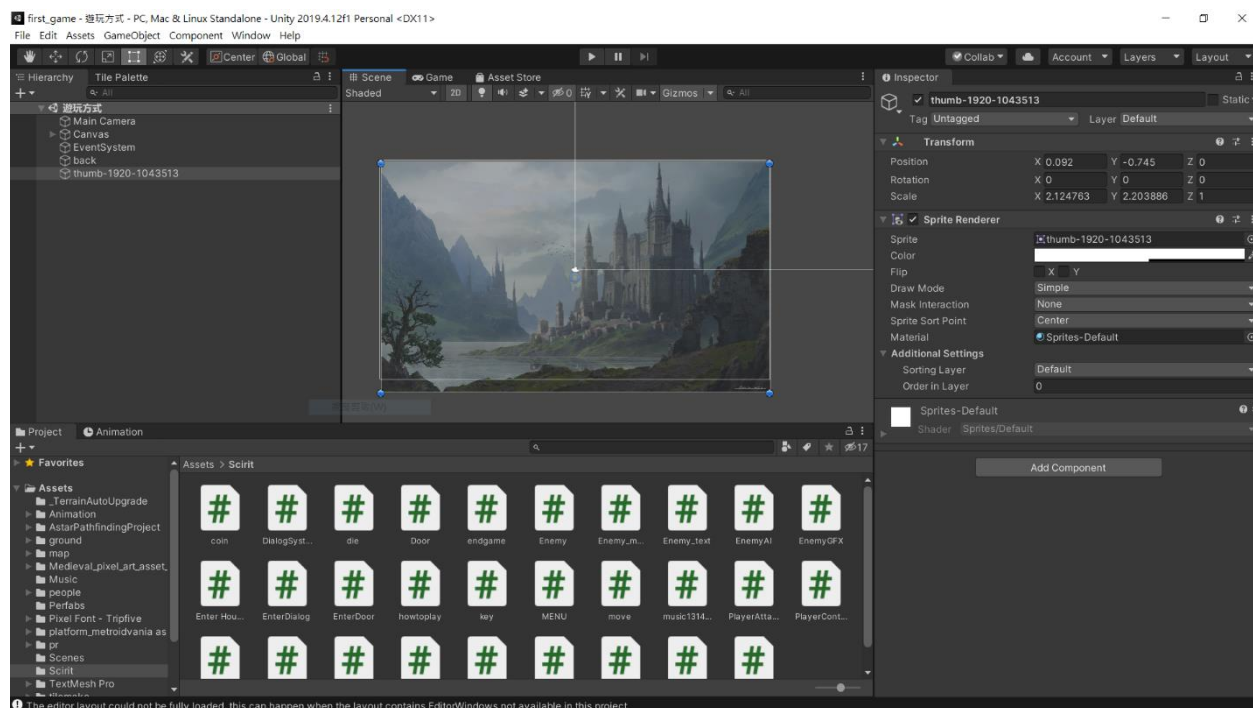


圖 1 Unity

## (二) Visual Studio

如圖2所示，Visual Studio是一個含有C#、C++、BASIC等的開發工具。我們這次的專題中，使用的是C#。

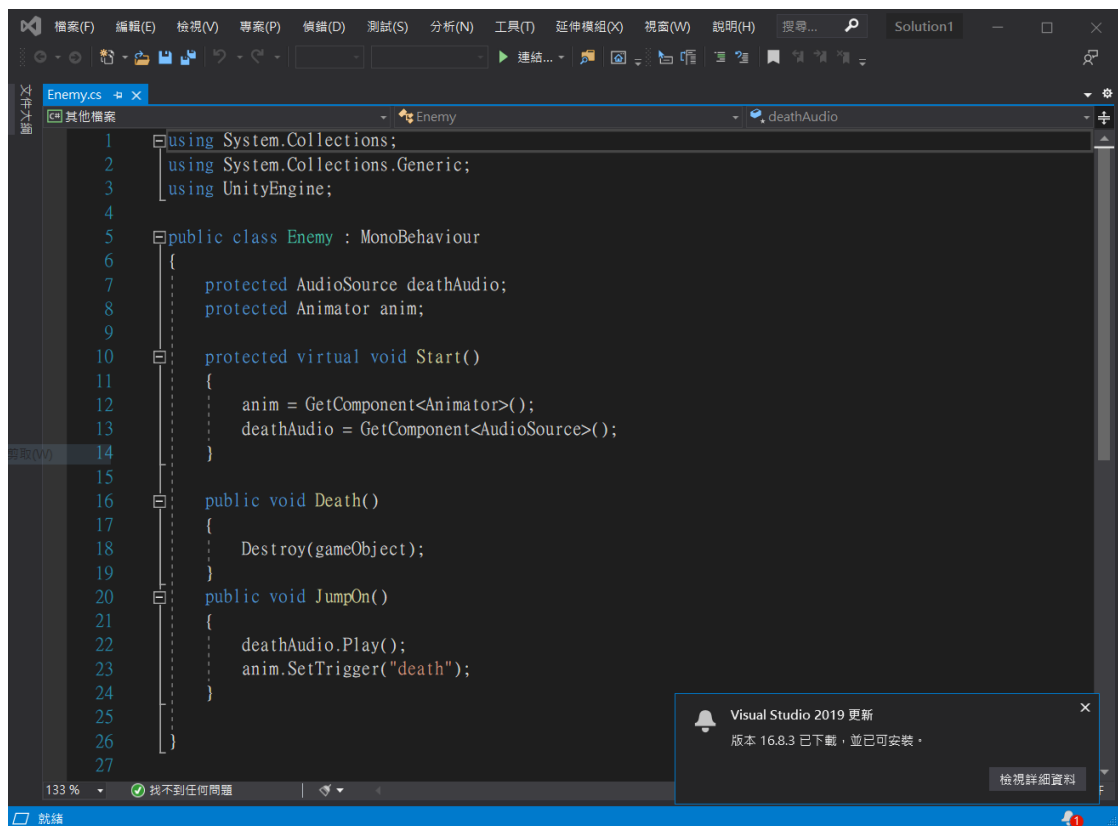


圖 2 Visual Studio

## (三) Aseprite

是一個可以繪製2D圖件的軟體，如圖3。

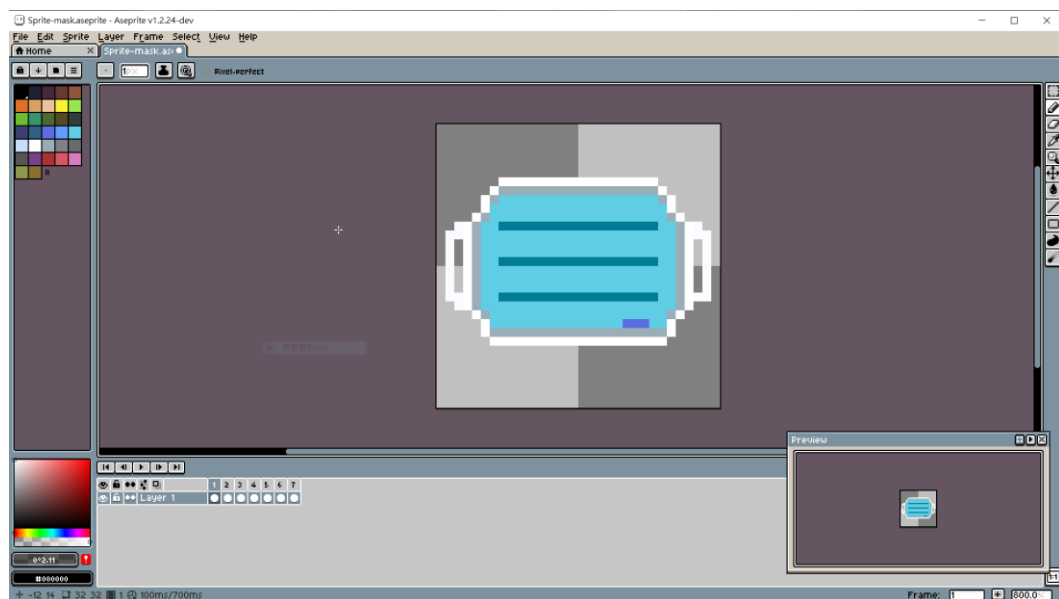


圖 3 Aseprite

## 參、製作過程

### 一、場景設計

我們場景製作的方式是利用Unity裡面的tilemap去製作的，其製作的方式類似於拼拼圖，如圖4。

我們製作的場景主要有四個，第一個是開始畫面，如圖5。第二個是遊戲的遊玩方式，如圖6。第三個是遊戲中的場景一，如圖7，此場景塑造出主角一人面對病毒大軍的危險路程。第四個是遊戲場景二，如圖8，此場景塑造出主角衝過難關，拿到了口罩和疫苗，並且悄悄地逃出魔王城。

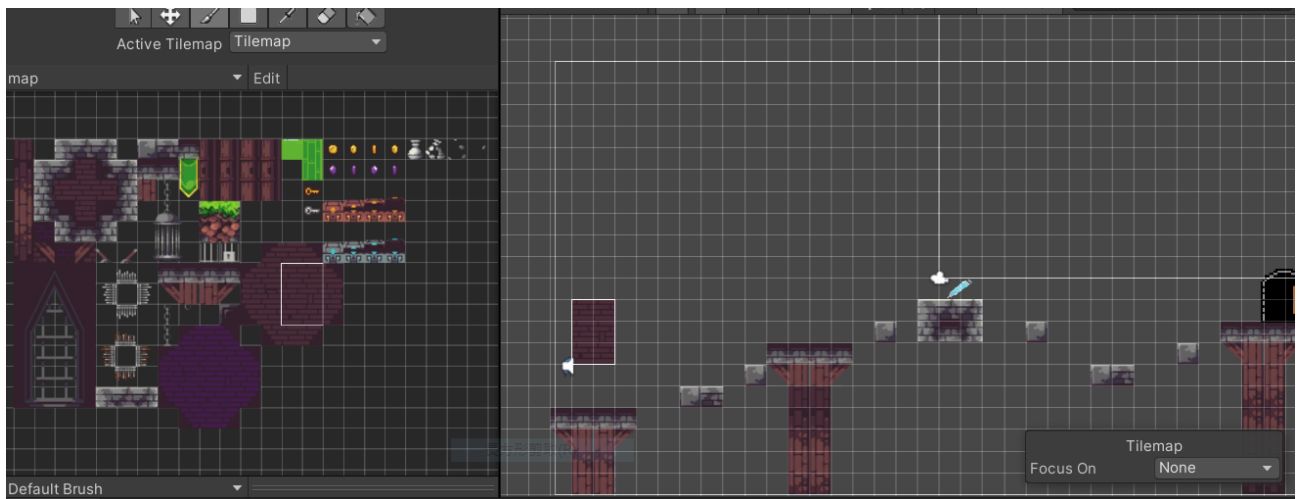


圖 4 tilemap 製作地圖



圖 5 開始畫面

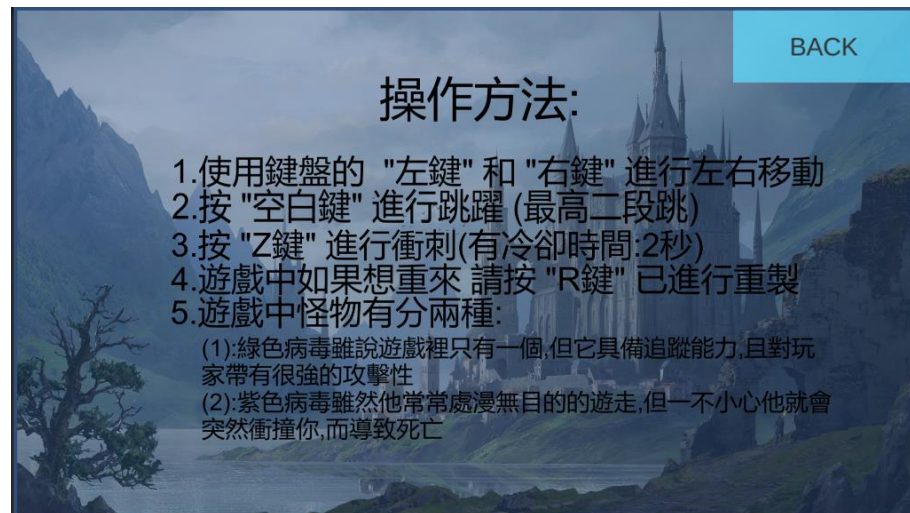


圖 6 遊玩方式



圖 7 遊戲中的場景一

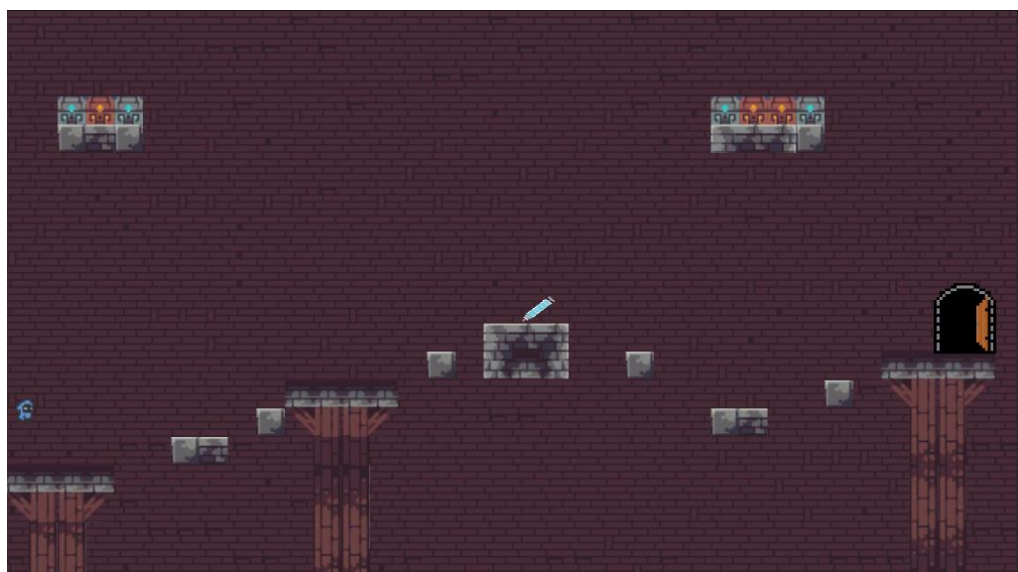


圖 8 遊戲中的場景

## 二、 故事動畫



圖 9 劇情用R鍵進行

我們專題的劇情是由玩家案下R鍵來進行劇情，分別有前後兩段劇情，第一段為故事開頭，如圖10。第二段為故事結尾，如圖11。

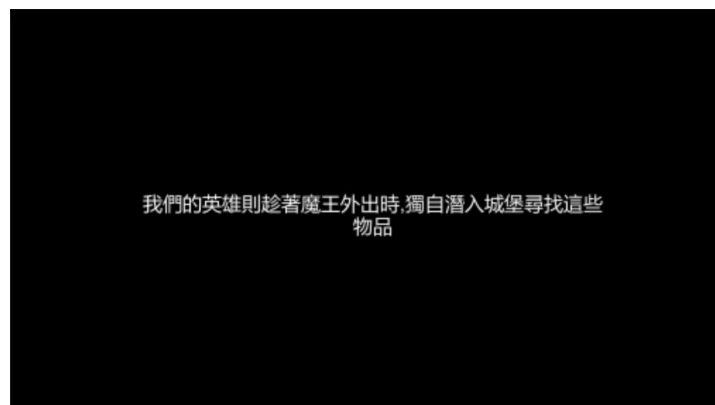


圖 10 開頭

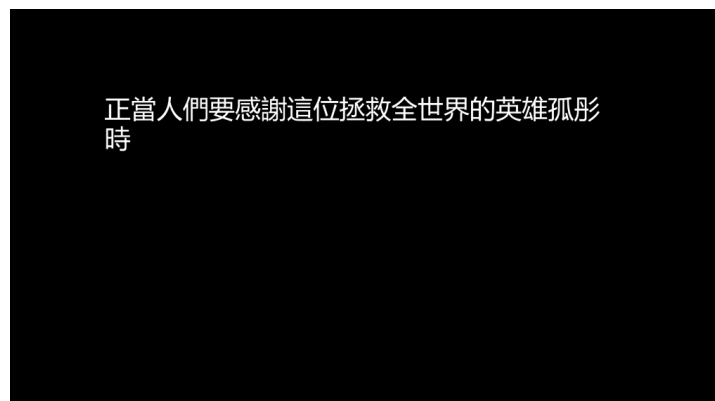


圖 11 結尾



### 三、 角色控制

角色的移動、死亡及碰撞掉落等的程式如圖12，下圖部分為操控角色的主要程式和碰到怪物，然後死亡的動畫等。

```
1  using System.Collections;
2  using System.Collections.Generic;
3  using System.Diagnostics;
4  using System.Globalization;
5  using System.Security.Cryptography;
6  using UnityEngine;
7  using UnityEngine.SceneManagement;
8  using UnityEngine.UI;
9
10
11
12
13  public class move : MonoBehaviour
14  {
15
16      public GameObject AAAA;
17      public AudioSource coinAudio;
18      public AudioSource jumpAudio;
19      public AudioSource runAudio;
20      private Animator anim;
21      private BoxCollider2D coll;
22      private Rigidbody2D rb;
23      public float jumpForce, speed;
24      public Transform groundCheck;
25      public LayerMask ground;
26      public float dieTime;
27
28
29      [Header("CD的UI组件")]
30      public Image cdImage;
31      [Header("Dash参数")]
32      public float dashTime; //dash時間
33      private float dashTimeLeft;
34      private float LastDash = -10f;
35      public float dashCoolDown;
36      public float dashSpeed;
37
38      public bool isGround, isJump, isDashing;
39
40      bool jumpPressed;
41      int jumpCount;
42      private float horizontalMove;
43
44
45
46      public int Coin = 0;
47
48      public Text coinNum;
49      private bool isHurt; //默認是false
50
51      public Transform keyFollowPoint;
52      public key followingKey;
53
54      void Start()
55      {
56
57          rb = GetComponent<Rigidbody2D>();
58          anim = GetComponent<Animator>();
59          coll = GetComponent<BoxCollider2D>();
60
61      }
62
63      void Update()
64      {
65          if (Input.GetKeyDown(KeyCode.R))
66          {
67              Invoke("Restart", 0f);
68          }
69          if (Input.GetButtonDown("Jump") && jumpCount > 0)
70          {
71              jumpPressed = true;
72          }
73      }
```

```

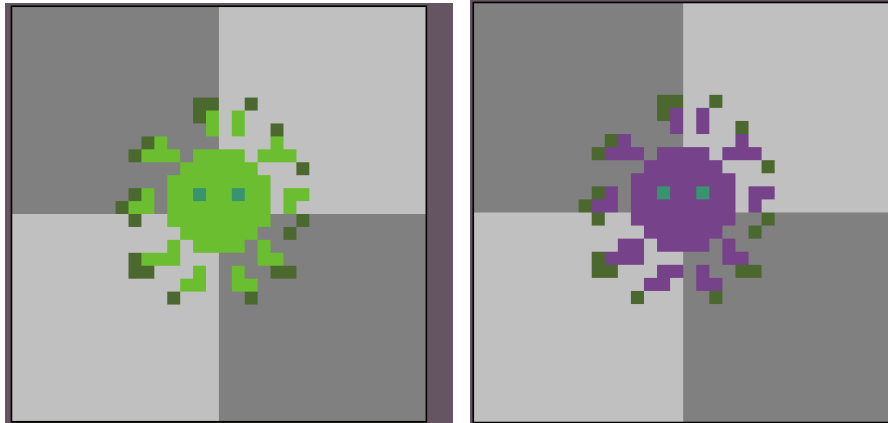
74         if(Input.GetKeyDown(KeyCode.Z))
75         {
76             if(Time.time >= (LastDash + dashCoolDown))
77             {
78                 //可以執行dash
79                 ReadyToDash();
80             }
81         }
82         cdImage.fillAmount -= 1.0f / dashCoolDown * Time.deltaTime;
83         coinNum.text = Coin.ToString();
84     }
85
86     private void FixedUpdate()...
87
88     void GroundMovement()...
89
90     void Jump()...
91
92     void SwitchAnim()...
93
94     //碰撞觸發氣
95     private void OnTriggerEnter2D(Collider2D collision) ...
96
97     //消滅敵人
98     private void OnCollisionEnter2D(Collision2D collision) ...
99
100     void world()...
101
102     void Restart()...
103
104     void ReadyToDash()...
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241     void Dash()...
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271     public void CoinCount()...
272
273
274
275     /* void Attack() ...
276
277
278
279
280
281
282
283
284

```

圖 12 角色主要的程式(部分收起)

#### 四、怪物控制

我們有兩種怪物分別是綠色及紫色兩種病毒，我們讓綠色的病毒在固定區域隨機走動，紫色的病毒則是跟隨著主角移動。



```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class Enemy_move : MonoBehaviour
6 {
7     public float speed;
8     public float startWaitTime;
9     private float waitTime;
10    public Transform movePos;
11    public Transform leftDownPos;
12    public Transform rightUpPos;
13
14    void Start()
15    {
16        waitTime = startWaitTime;
17        movePos.position = GetRandomPos();
18    }
19
20    void Update()
21    {
22        transform.position = Vector2.MoveTowards(transform.position, movePos.position, speed * Time.deltaTime);
23
24        if (Vector2.Distance(transform.position, movePos.position) < 0.1f)
25        {
26            if (waitTime <= 0)
27            {
28                movePos.position = GetRandomPos();
29                waitTime = startWaitTime;
30            }
31            else
32            {
33                waitTime -= Time.deltaTime;
34            }
35        }
36    }
37
38    Vector2 GetRandomPos()
39    {
40        Vector2 rndPos = new Vector2(Random.Range(leftDownPos.position.x, rightUpPos.position.x), Random.Range(leftDownPos.position.y, rightUpPos.position.y));
41        return rndPos;
42    }
43
44 }
45
46
47
```

圖 3 綠色病毒程式碼

```

472
473 /// <summary>Draws gizmos for the Seeker</summary>
474 public void OnDrawGizmos () {
475     if (lastCompletedNodePath == null || !drawGizmos) {
476         return;
477     }
478
479     if (detailedGizmos) {
480         Gizmos.color = new Color(0.7F, 0.5F, 0.1F, 0.5F);
481
482         if (lastCompletedNodePath != null) {
483             for (int i = 0; i < lastCompletedNodePath.Count-1; i++) {
484                 Gizmos.DrawLine((Vector3)lastCompletedNodePath[i].position, (Vector3)lastCompletedNodePath[i+1].position);
485             }
486         }
487     }
488
489     Gizmos.color = new Color(0, 1F, 0, 1F);
490
491     if (lastCompletedVectorPath != null) {
492         for (int i = 0; i < lastCompletedVectorPath.Count-1; i++) {
493             Gizmos.DrawLine(lastCompletedVectorPath[i], lastCompletedVectorPath[i+1]);
494         }
495     }
496 }
497
498 protected override int OnUpgradeSerializedData (int version, bool unityThread) {
499     if (graphMaskCompatibility != -1) {
500         Debug.Log("Loaded " + graphMaskCompatibility + " " + graphMask.value);
501         graphMask = graphMaskCompatibility;
502         graphMaskCompatibility = -1;
503     }
504     return base.OnUpgradeSerializedData(version, unityThread);
505 }
506
507
508

```

圖 14 紫色病毒程式碼(擷取一部份)

## 五、 物件繪製

我們利用Aseprite繪製了口罩(包含動畫)、疫苗、病毒、門(開、關)、人物頭像、設定鈕、開始遊戲的選項等。

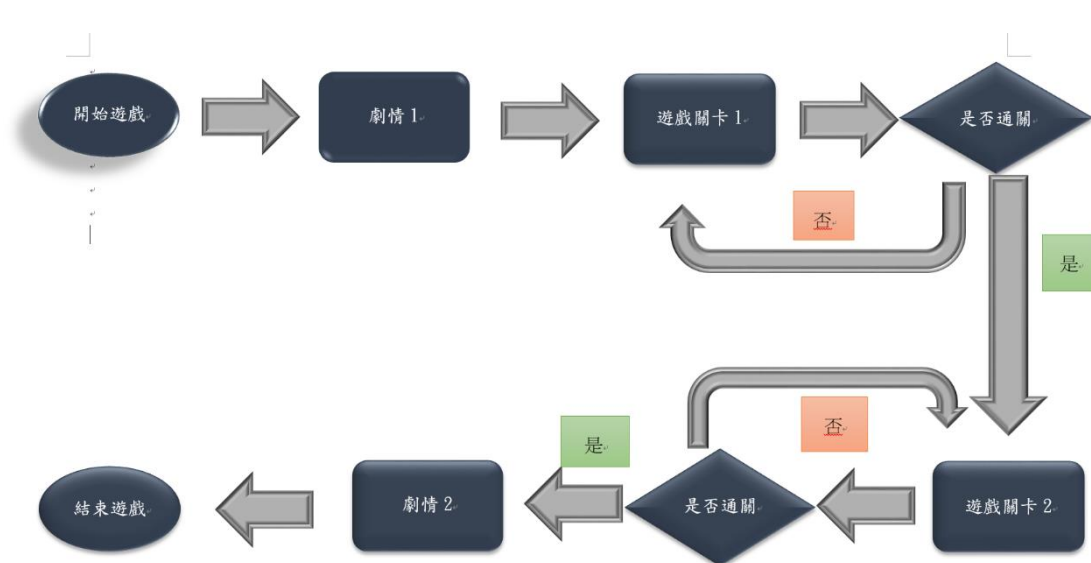


圖 4 物件繪製

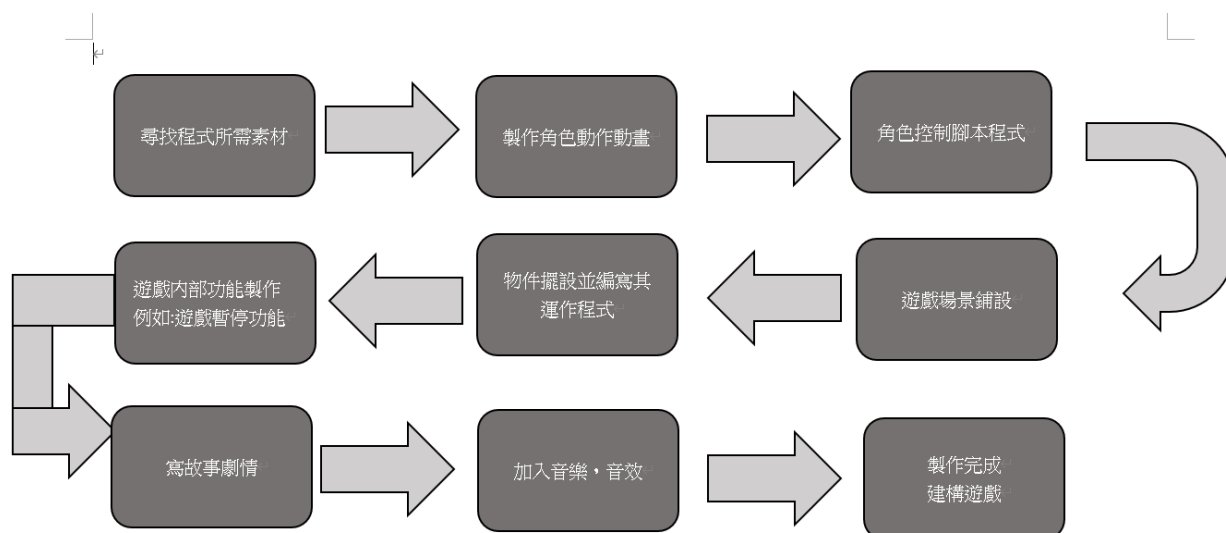
## 六、 背景音樂

我們遊戲開頭的音樂用的是Despair，第一段劇情使用打字聲，遊戲中的場景一用Axero-River，場景二則是無背景音樂，塑造出悄悄的離開魔王城的緊張感，第二段劇情使用Victory，塑造出英雄歸來的壯闊感。

## 七、 整體程式架構



## 八、 程式設計流程



## 肆、 成果展示與討論

### 一、 成果展示

第一部分：進入遊戲時先進到設計好的UI界面（如圖16）



圖 5 開始介面

第二部分：

(1)按下『開始遊戲』則進入劇情畫面



圖 6 劇情演示

(2)按下『遊戲介紹』則進入遊戲說明

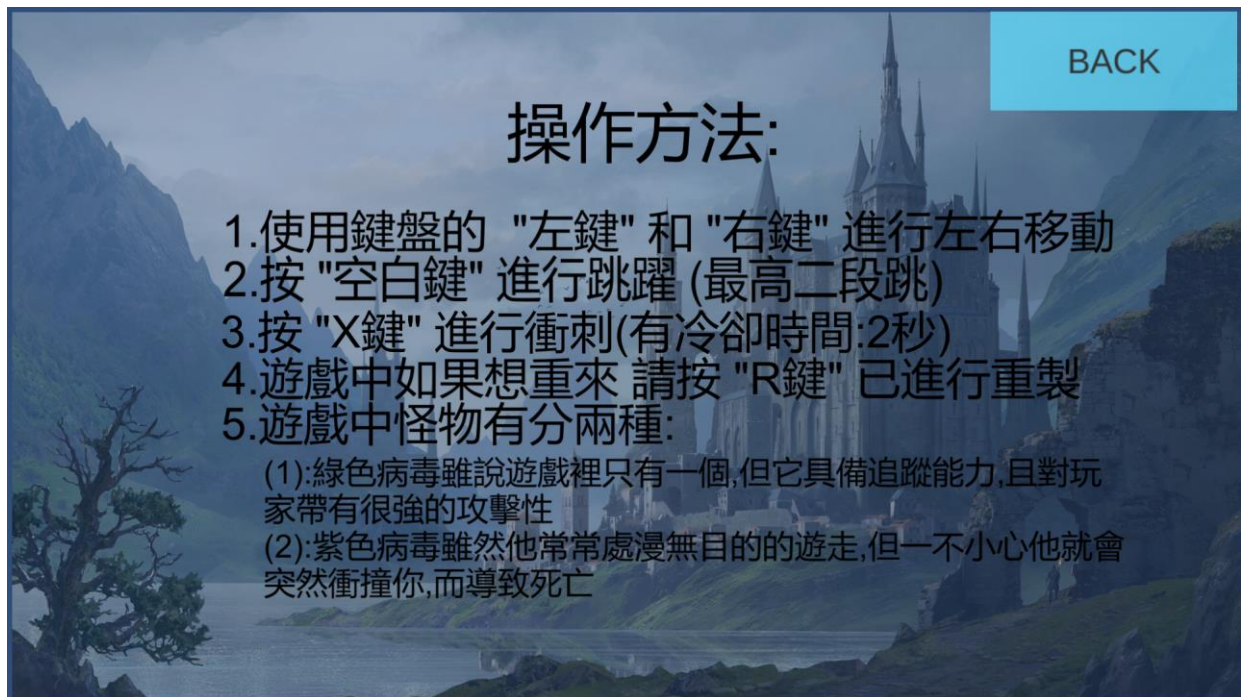


圖 7 操作方法

(3)按下『退出遊戲』則會退回桌面

第三部分：

(1)當玩家走完劇情介紹後則會進入遊戲關卡(一)的主畫面 如圖 19



19遊戲關卡(一)



(2)通過遊戲關卡(一)二後並會進入遊戲關卡(二)的主畫面 如圖 20

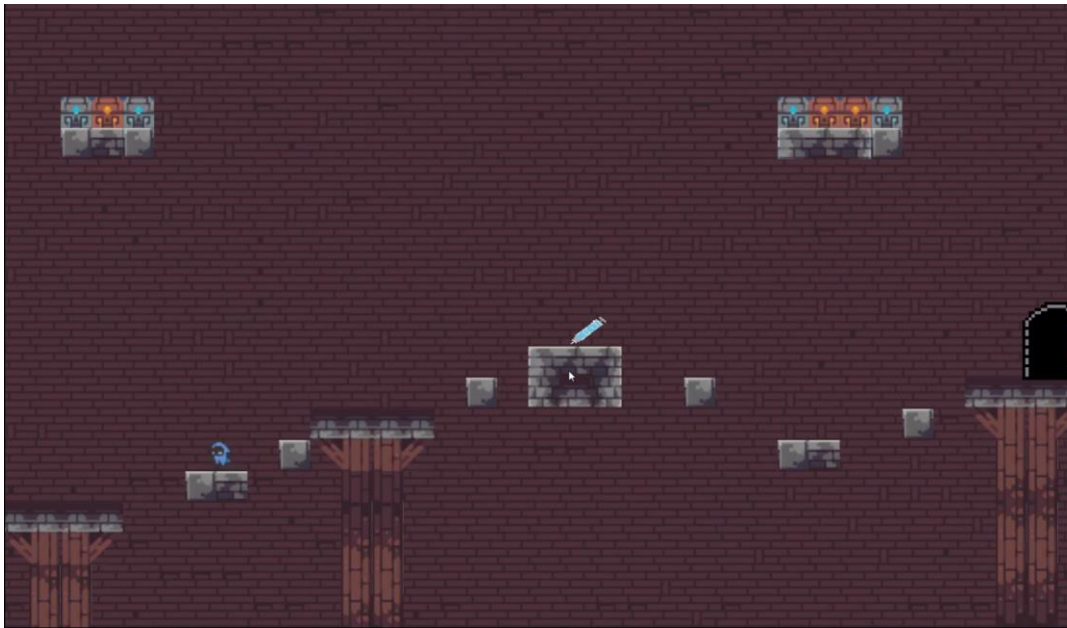


圖 20 遊戲關卡(二)

## 二、問題與討論

1、 問題：我們在做完地圖後做執行，不知道為什麼原本鋪設好的場景出現大面積破圖。

原因：未找出。

解決辦法：只好在重新製作花了不少時間。

2、 問題：在重做地圖時，所以有些小圖片沒辦法填滿格子。

原因：我們的地圖是用圖片分割出一塊一塊的小圖片組裝。

解決辦法：因此我們用了四個圖層去補足。

3、 問題：程式編寫上遇上一些瓶頸。

原因：原本就對C#語法不熟所以大家都得重頭學起。

解決辦法：參考網路上的各個資料綜合起來寫出自己的版本。



## 伍、 結論

我們發現我們做的程度其實跟理想有很大的差距，我們因為時間的限制和遊戲方面的專業知識不足，所以我們不能把所有想要做的東西給做出來，這讓我們有點遺憾。

這次專題讓我們意識到，製作遊戲很辛苦，裡面的分工一項都不能少，而且「失之毫釐，謬以千里」只要有一個小步驟忘記或者失誤，就會導致最後整個遊戲無法運行。

在這次專題的過程中，我們還學到一些控制人物和怪物還有切換場景的程式，以及 UI 界面的製作，更重要的事是我們學到了分工的重要性，也提前擁有了製作遊戲的經驗，讓我們的將來可以有更多的選擇。

## 陸、 參考資料

- 1、 M-Studio 的影片教學

[https://www.youtube.com/playlist?list=PL\\_Pb2I110MfGAsoqtDs8-6kEU55wU8CnE](https://www.youtube.com/playlist?list=PL_Pb2I110MfGAsoqtDs8-6kEU55wU8CnE)。

- 2、 Unity 官方網站

<https://unity.com/>。

## 柒、 部分專題素材

- 1、 遊玩方法的背景

<https://wall.alphacoders.com/big.php?i=1043513&lang=Chinese>。

- 2、 玩家、遊戲中的背景

<https://o-lobster.itch.io/platformmetroidvania-pixel-art-asset-pack>。

- 3、 遊戲開始的背景

<https://szadiart.itch.io/pixle-castle-2d>。

- 4、 遊戲開頭音樂

<https://www.youtube.com/watch?v=4nweeToBM14>