



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系

Unity 2D 益智類遊戲研究與開發

指導教授：林孟源 教授

組員名單：吳承叡 A78C022

陳宥成 A78C063

中 華 民 國 1 1 1 年 0 4 月



嶺東科技大學
LING TUNG UNIVERSITY

資訊管理系專題口試委員審定書

Unity 2D 益智類遊戲研究與開發

指導教授： 林孟源 教授
組員名單： 吳承勸 A78C022
 陳宥成 A78C063

指導教授： _____
口試委員： _____

中 華 民 國 1 1 1 年 0 4 月

謝 誌

本專題報告得以順利完成，首先要感謝恩師林孟源老師細心引導我們，耐心的協助我們，克服研究過程中所面臨的困難，給予我們最大的協助，使本專題得以順利完成。

研究報告口試期間，感謝謝淑玲老師、陳健忠老師不辭辛勞細心審閱，不僅給予我們指導，並且提供寶貴的建議，使我們的專題內容以更臻完善，在此由衷的感謝。

最後，感謝系上諸位老師在各學科領域的熱心指導，增進商業管理知識範疇，在此一併致上最高謝意。

吳承歡 謹誌
陳宥成

中華民國111年04月於嶺東

摘 要

本遊戲以 Unity 來使用懷舊風的倉庫番為主題製作，在原本推箱子的玩法上，結合了陷阱、怪物導致遊戲失敗，來提高難度，讓自己在重玩時可以看到自己的進步。

本遊戲的遊戲規則有三。第一:只能推動一個箱子。第二:不能去拉動箱子。第三:不能同時推動或移動超過一個箱子。遊戲的玩法是把所有的箱子推至指定的點上，本關卡共製作了八個關卡，隨著越進階的關卡，陷阱及怪物也會變多，挑戰難度會變的更高，如果碰到陷阱或是怪物就會導致遊戲失敗。

關鍵詞：益智遊戲、倉庫番、Unity

目 錄

摘 要	I
目 錄	II
表目錄	III
圖目錄	IV
第壹章 緒論	1
1.1 研究動機	1
1.2 研究目的	1
第貳章 文獻回顧與探討	2
2.1. 遊戲介紹	2
2.2 遊戲規則	2
2.3 文獻探討	2
2.3.1 Patrick's Parabox (箱中有箱)	3
2.3.2 Helltaker	4
2.3.3 地城迷蹤	5
2.4 軟體介紹	7
2.4.1 Unity 介紹	7
2.4.2 C#介紹	7
2.4.3 Adobe Illustrator 介紹	8
2.4.4 Adobe Photoshop 介紹	8
第參章 研究方法	9
3.1 研究流程	9
3.2 甘特圖	10
3.3 研究人員分配	10
3.4 開發環境	11
3.5 遊戲流程圖	12
第肆章 系統分析與設計	13
4.1 介面設計	13
4.2 關卡設計	14
4.3 怪物的動畫製作	18
第伍章 結論與未來展望	19
參考文獻	20
附錄	
綜合輔導紀錄表	

表目錄

表 2.1 分析比較表	6
表 3.1 人力分配表	10
表 3.2 開發環境表	11

圖目錄

圖 2.1 以原型及三個模板來自動產生遊戲盤面.....	2
圖 2.2 Patrick's Parabox 遊戲畫面1	3
圖 2.3 Patrick's Parabox 遊戲畫面2	3
圖 2.4 Helltaker 遊戲畫面1	4
圖 2.5 Helltaker 遊戲畫面2	4
圖 2.6地城迷蹤遊戲畫面1.....	5
圖 2.7 地城迷蹤遊戲畫面2.....	5
圖 3.1 研究流程圖	9
圖 3.2 甘特圖	10
圖 3.3 遊戲流程圖	12
圖 4.1 遊戲主畫面.....	13
圖 4.2 遊戲選擇關卡畫面	13
圖 4.3 遊戲第一關地圖	14
圖 4.4 遊戲第二關地圖	14
圖 4.5 遊戲第三關地圖	15
圖 4.6 遊戲第四關地圖	15
圖 4.7 遊戲第五關地圖	16
圖 4.8 遊戲第六關地圖	16
圖 4.9 遊戲第七關地圖	17
圖 4.10 遊戲第八關地圖	17
圖 4.11 怪物動畫製作圖1.....	18
圖 4.12 怪物動畫製作圖2.....	18
圖 4.13 怪物動畫製作圖3.....	18
圖 4.14 怪物動畫製作圖4.....	18
圖 4.15 怪物動畫製作圖5.....	18

第壹章 緒論

1.1 研究動機

由於近年來，在這個無法避免小孩接觸遊戲的年代，如果能夠開發出一款在玩的同時也能增強腦力的遊戲，在玩遊戲放鬆的同時，也能夠加強大腦的思考能力，進而達到一箭雙鵰的目的。與其不讓孩子們玩，不如讓他們接觸一些開發大腦、有助於提升智力及邏輯判斷能力的益智遊戲。

1.2 研究目的

本專題使用Unity來製作2D遊戲，設計一款既好玩又能鍛鍊腦力的遊戲，本遊戲以傳統倉庫番的概念為基礎，在其之上加入了陷阱、怪物等額外內容來達到關卡的多元性，使玩家在通關時能以不同方法來達到關卡目標，進而達到玩家可以在通關後繼續研究其他不同的解法，以增加遊戲的可玩性。

第貳章 文獻回顧與探討

2.1. 遊戲介紹

「倉庫番」是一款經典的電子遊戲，在1982年由日本 Thinking Rabbit 公司首次發行，之後其他開發者爭相製作仿製或衍生產品，致使倉庫番成為此類遊戲的代名詞，而本次專題內容即為在原始的倉庫番上，添加更多元素後衍生出來的遊戲。[1]

2.2 遊戲規則

最初的遊戲規則為遊戲要求玩家在二維地圖上把箱子推至指定的地點，已觸發進入下一關的條件，而本專題則在初始的推箱子玩法上結合了陷阱、怪物等其他條件進一步干擾玩家從而達到更加靈活的玩法。

2.3 文獻探討

從1980年代，日本遊戲公司開發了倉庫番之後，不斷有新的遊戲盤面被開發出來，1990年代開始有人試圖以資訊科學的角度來分析倉庫番問題，其中有人試圖讓電腦自動產生遊戲盤面，如圖所示，作者構想利用一個原型(prototype)及三個模板(Template)來自動產生遊戲盤面，對於電腦所產生的新盤面，其中一部分的研究在於如何評量電腦產生的遊戲盤面是否有解，所採取的方法是以廣度優先搜尋法(Breadth-furst search)來尋找解答，將無解的盤面剔除。但是明顯的，已廣度優先研究法來尋找答案並不是個好方法，但卻是以電腦來解決倉庫番問題的一個前瞻性研究。

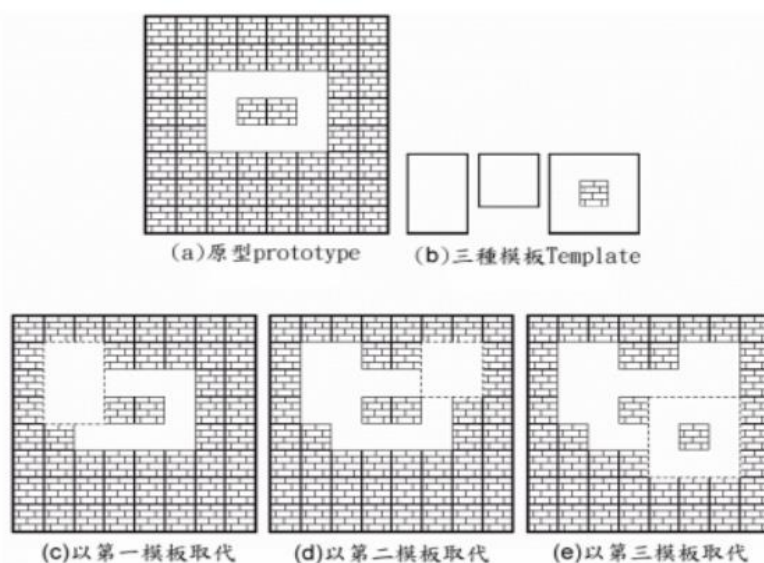


圖 2.1以原型及三個模板來自動產生遊戲盤面

2.3.1 Patrick's Parabox (箱中有箱)



圖 2.2 Patrick's Parabox 遊戲畫面1



圖 2.3 Patrick's Parabox 遊戲畫面2

在遊戲中，採用類似經典益智遊戲《倉庫番》那樣的設計玩法，玩家們需要將關卡上的箱子推到指定場所才行。不過特別的是，在箱子裡還會有箱子存在，而要推動這個箱子裡的箱子，就必須更深入到箱子裡的箱子中的箱子裡來移動裡面的箱子—獨特的遊戲系統，加上箱中有箱的無限組合玩法，以及多達 350 關以上的益智挑戰，讓本作自公開以來就受到眾多玩家注目。
[2]

2.3.2 Helltaker

遊戲中玩家扮演男主前往地獄與各式各樣的知名神話地獄惡魔擬人女性角色追求愛慕，並將其納入後宮，遊戲玩法類似於《倉庫番》，玩家必須透過推動石塊或者打倒骷髏兵，並在有限的步數下清空道路到達目標（惡魔擬人女性角色）身邊，玩家達到目標後還必須選擇適當的選項來說服目標愛慕玩家，遊戲後期有閃躲反應遊戲機制，若玩家選擇錯誤而導致說服失敗，就必須重新開始遊戲關卡。



圖 2.4Helltaker 遊戲畫面1

遊戲本身玩法並不長或是說多特別，是以推方塊為主的益智解謎玩法，並在後期才會有變成反應閃躲遊戲，但最特別的是遊戲的主題非常無厘頭，身為男主的玩家必須通過上面提到的玩法，在地獄中將所有知名的神話地獄角色收入後宮當中。遊戲在推出後，有趣的角色互動與個性，很快就讓遊戲成為網路上的新熱門話題。[3]

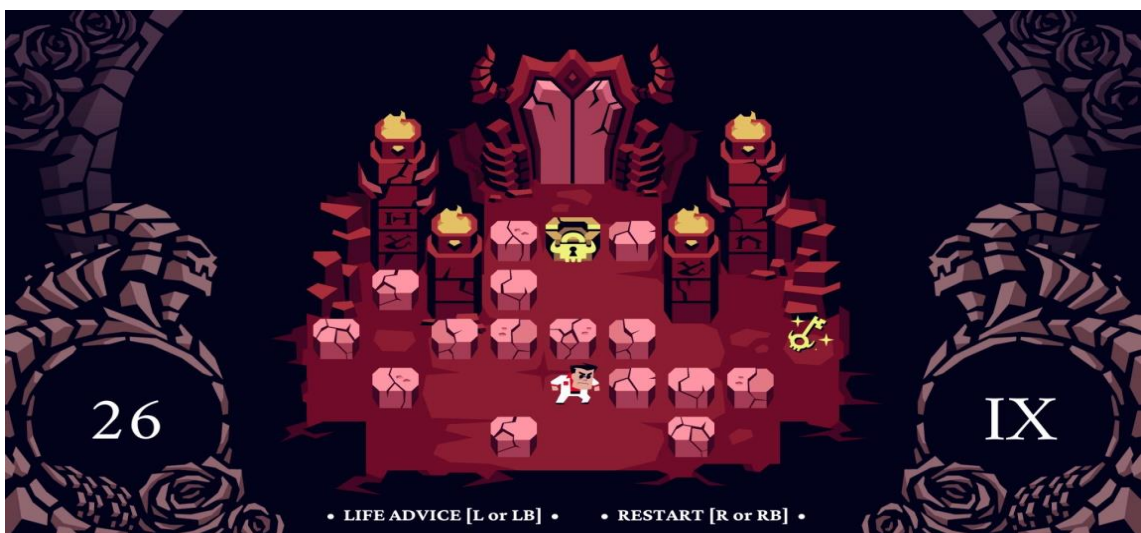


圖 2.5Helltaker 遊戲畫面2

2.3.3 地城迷蹤

《地城迷蹤》在開發初期時，僅由 HuMoFish 單獨製作，融合了傳統倉庫番以及 RPG 裡裝備的概念，是一款考驗玩家考驗 2D 空間邏輯，走可愛像素風格的地城主題解謎遊戲，冒險者的能力會隨著拾取裝備的切換而有所改變，所以每一步都需要深思熟慮。過關之後，你可以繼續嘗試不同的攻略路線並最佳化解法。

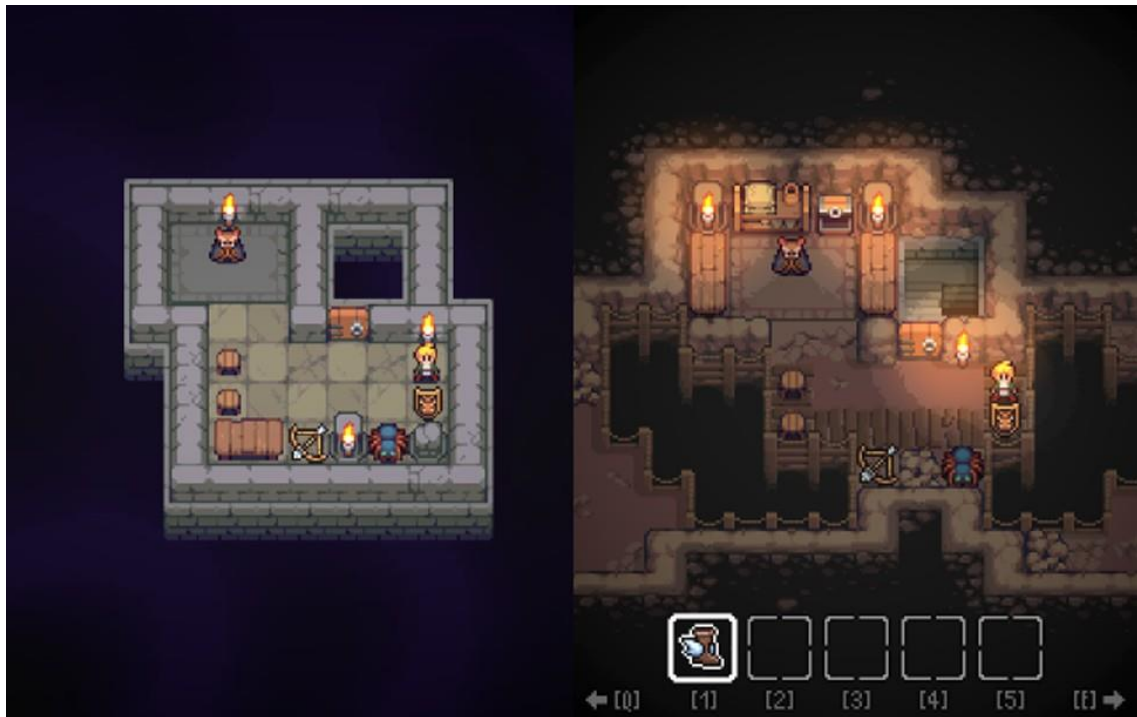


圖2.6地城迷蹤遊戲畫面1

挑戰最佳步數是《地城迷蹤》中的特色之一，每個關卡中都有 HuMoFish 所安排的標準步數，當然、這不一定會是最佳的步數，因此在過關卡之餘，所花費的步驟比預設的少也成為了遊戲中的另一項挑戰。[9]

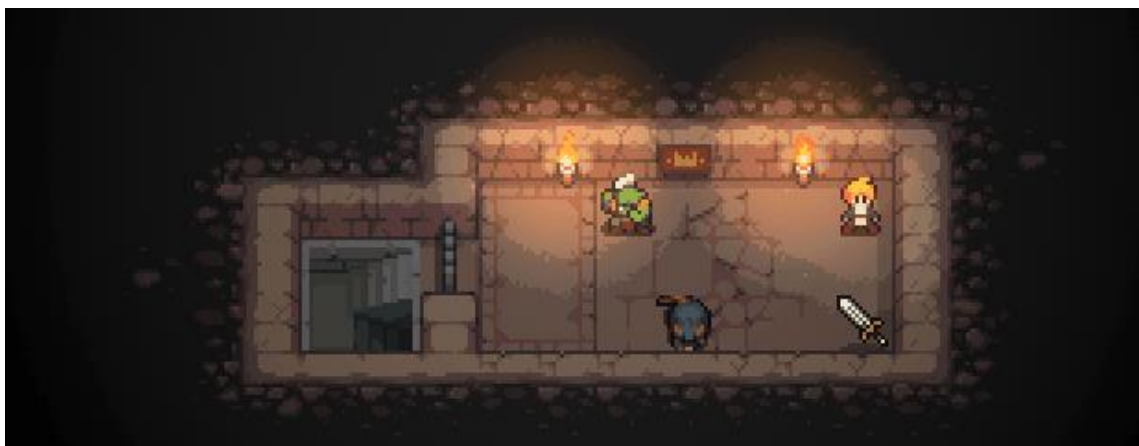


圖2.7地城迷蹤遊戲畫面2

2.3.3 分析比較

表 2.1 分析表

功能 遊戲	選擇關卡	額外陷阱	故事性	限制步數
箱中有箱	O	X	X	X
Helltaker	X	O	O	O
地城迷蹤	O	O	X	X

2.4 軟體介紹

2.4.1 Unity介紹

Unity 是一套跨平台的遊戲引擎，可開發 Windows、MacOS、Linux 單機遊戲或 iOS、Android 行動裝置遊戲。Unity 也可開發網頁遊戲，訪客的瀏覽器支援 WebGL 技術即可遊玩 Unity 開發的遊戲。

Unity 支援 PhysX 物理引擎、粒子系統，並且提供網路多人連線的功能，無需學習複雜的程式語言，符合遊戲製作上的各項需求。Unity 的推出降低遊戲開發的門檻，即使是個人或小型團隊製作遊戲也不再是夢想。對於遊戲公司而言，選擇使用 Unity 引擎也可以縮短遊戲的開發時間。[4]

Unity 的主要特色[5]

(1) 跨平台

遊戲開發者可以通過不同的平台進行開發。遊戲製作完成後，遊戲無需任何修改即可直接一鍵發佈到常用的主流平台上。Unity 遊戲可發布的平台包括 跨平台開發可以為遊戲開發者節省大量時間。

(2) 綜合編輯

Unity 的用戶界面具備視覺化編輯、詳細的屬性編輯器和動態遊戲預覽特性。Unity 創新的可視化模式讓遊戲開發者能夠輕鬆構建互動體驗，當遊戲運行時可以實時修改參數值，方便開發，為遊戲開發節省大量時間。

(3) 資源導入

項目可以自動導入資源，並根據資源的改動自動更新。Unity 支持幾乎所有主流的三維格式，如 3ds Max、Maya、Blender 等，貼圖材質自動轉換為 U 格式，並能和大部分相關應用程序協調工作。

(4) 一鍵部署

Unity 只需一鍵即可完成作品的多平台開發和部署，讓開發者的作品在多平台呈現。

2.4.2 C#介紹

C#是微軟推出的一種基於.NET框架的、物件導向的進階程式語言。C#是一種由 C 和 C++衍生出來的物件導向的程式語言。它在繼承 C 和 C++強大功能的同時去掉了一些它們的複雜特性，使其成為 C 語言家族中的一種高效強大的程式語言。C#以.NET框架類別庫作為基礎，擁有類似 Visual Basic 的快速開發能力。C#由安德斯·海爾斯伯格主持開發，微軟在2000年發布了這種語言，希望藉助這種語言來取代 Java。C#已經成為 Ecma 國際和國際標準組織的標準規範。[6]

2.4.3 Adobe Illustrator 介紹

Adobe Illustrator，簡稱「AI」，是 Adobe 系統公司推出的基於向量的圖形製作軟體。最大特徵在於貝茲曲線的使用，使得操作簡單功能強大的向量繪圖成為可能。現在它還整合文書處理、上色等功能，不僅在插圖製作，在印刷製品（如廣告傳單、小冊子）設計製作方面也廣泛使用。[7]

2.4.4 Adobe Photoshop 介紹

Adobe Photoshop 是一個由 Adobe 開發和發行的影像處理軟體。該軟體提供對 Windows 和 macOS 的支援。Photoshop 主要處理以像素所構成的數位影像。使用其眾多的編修與繪圖工具，可以更有效的進行圖片編輯工作。獨特的歷史紀錄浮動視窗和可編輯的圖層效果功能使使用者可以方便的測試效果。對各種濾鏡的支援更令使使用者能夠輕鬆創造出各種奇幻的效果。[8]

第參章 研究方法

3.1 研究流程

1. 規劃：

從各種倉庫番遊戲為例，參考不同類型的盤面要如何配置，並規劃路線及物件的擺放位置。

2. 分析：

根據其他遊戲的心得結論，從中得到各玩家們不同的喜好、以及優缺點，從而得到該如何進一步優化我們的系統功能。

3. 設計：

以益智遊戲方面的角度，尋找需要的系統功能，並從中擷取必要的元素來設計我們遊戲的介面。

4. 建置：

從各個影片中學習 UNITY 及 C#的應用，並一點一點建構出整個遊戲的架構，最後實踐出目標中的遊戲畫面。



圖 3.1研究流程圖

3.2 甘特圖

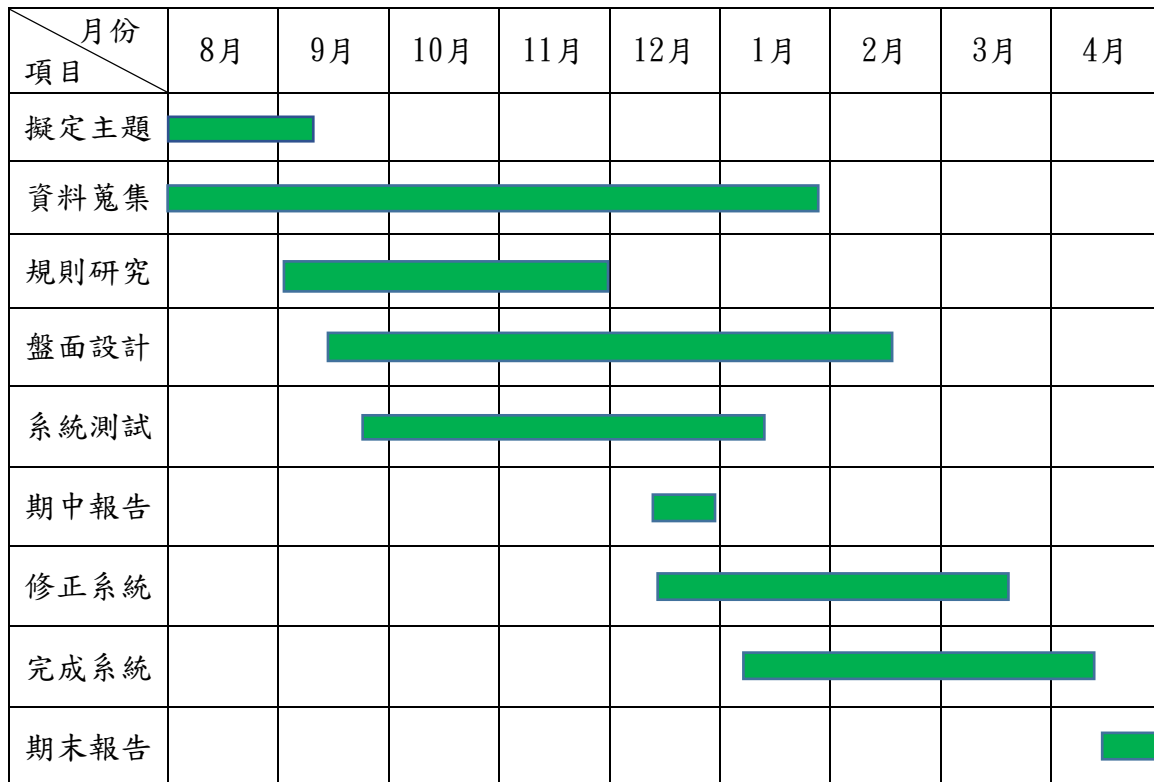


圖 3.2甘特圖

3.3研究人員分配

表 3.1 人力分配表

姓名 工作項目	吳承叡	陳宥成
尋找素材	P	
尋找參考文獻	P	
程式碼撰寫	P	
程式測試	P	M
書面報告製作	P	M
簡報製作	P	

註:P(Primary)為主要負責人，M(Minor)為次要負責人

3.4 開發環境

表 3.2 開發環境

項目	使用環境	
作業系統	Windows 10	 Windows 10
開發遊戲	Unity3D	
程式語言	C#	
圖形設計軟體	Adobe Photoshop	
圖形設計軟體	Adobe Illustrator	

3.5 遊戲流程圖

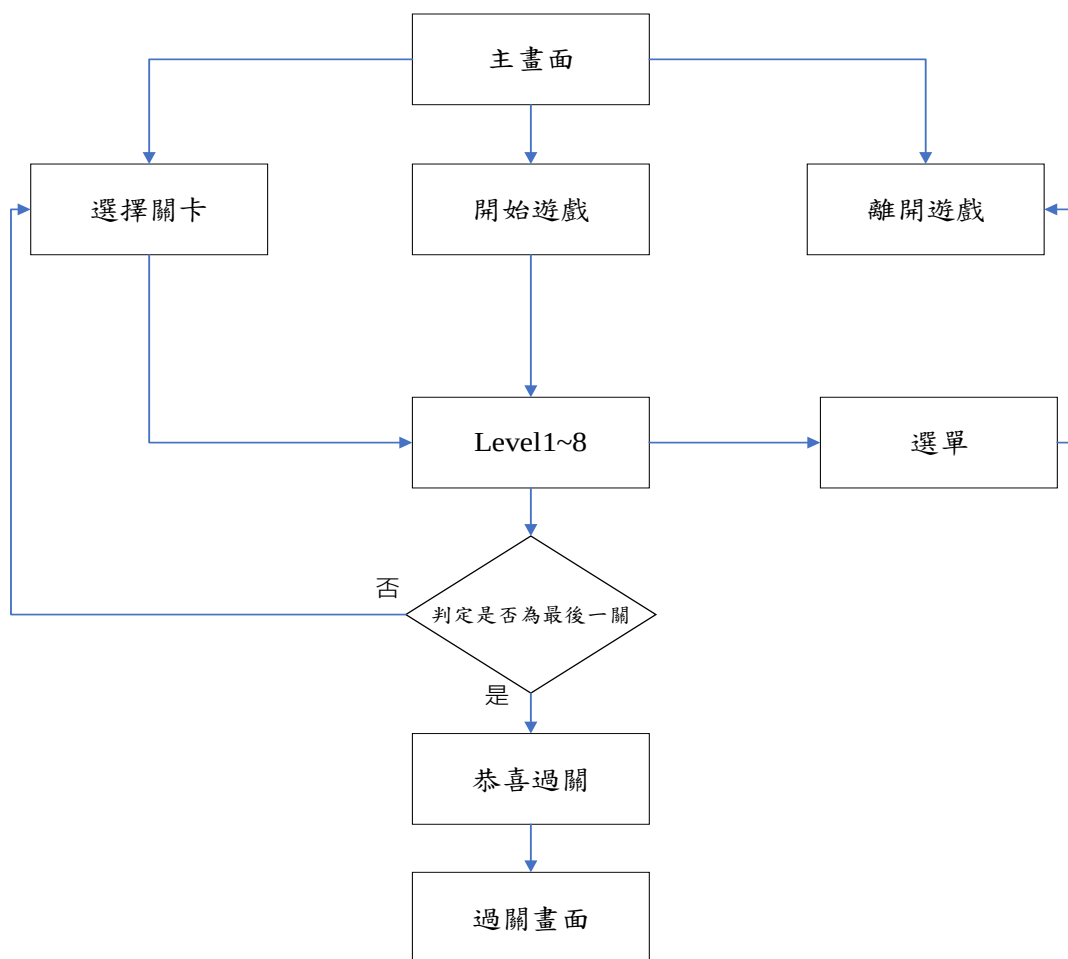


圖 3.3遊戲流程圖

第肆章 系統分析與設計

4.1 介面設計

點開遊戲後第一個進入的畫面為遊戲的主介面，一開始只有開始遊戲、選擇關卡與結束的按鈕。



圖 4.1 遊戲主畫面

進入關卡選擇後若是尚未通關，選擇關卡的按鈕會呈現灰色狀態，沒有辦法點擊進入關卡，而在通關關卡之後會逐一使按鈕轉變成白色，使玩家可以不用每次都從第一關重新進行遊戲。

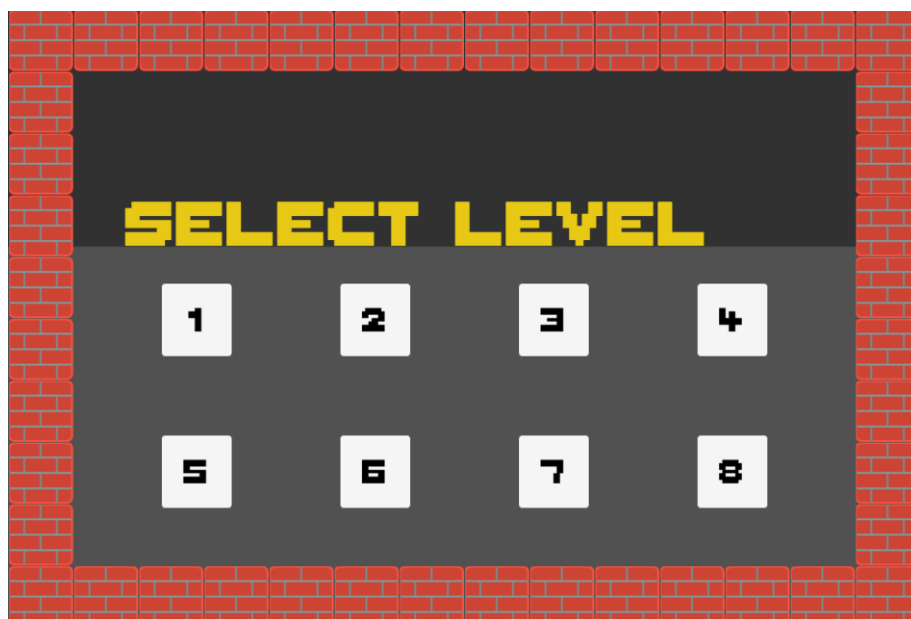


圖 4.2 遊戲選擇關卡畫面

4.2 關卡設計

整個遊戲的關卡盤面為我們手動逐一設計，牆壁，箱子，目的地都是以我們自己的構想來慢慢填上。

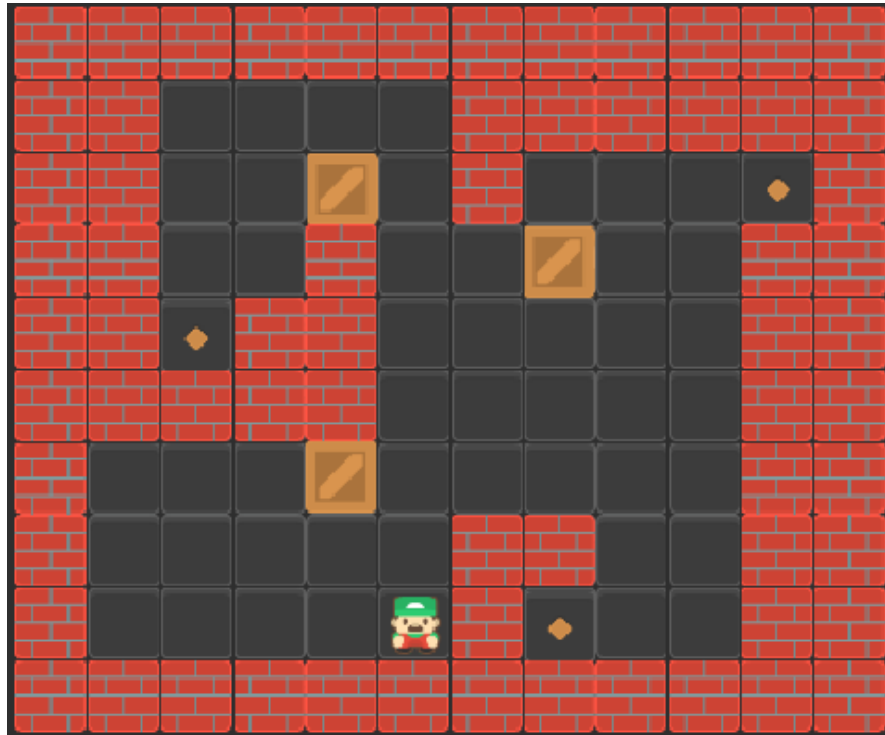


圖 4.3 遊戲第一關地圖

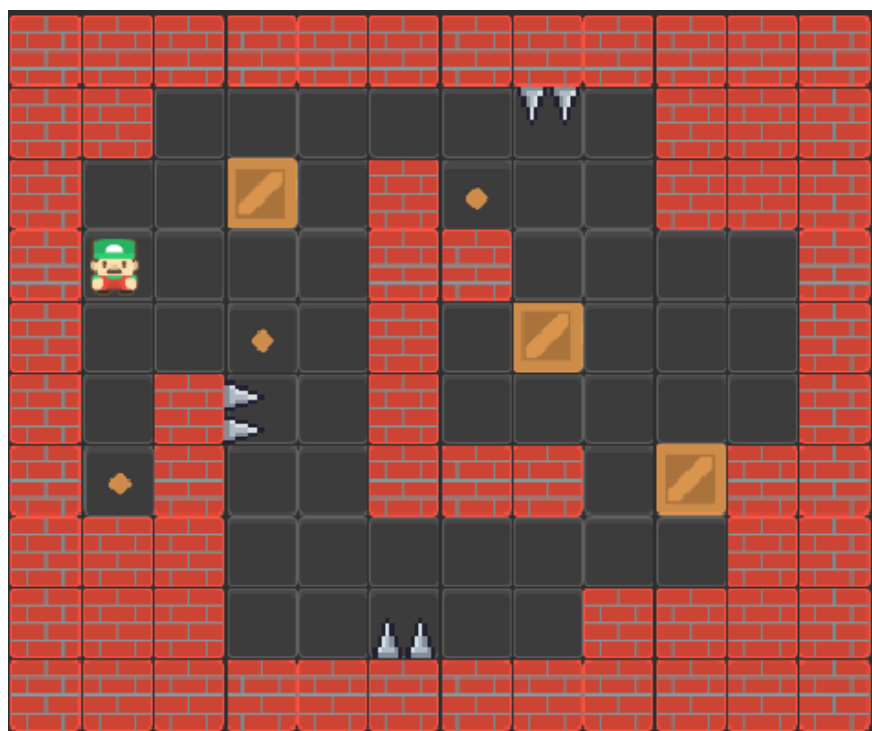


圖 4.4 遊戲第二關地圖

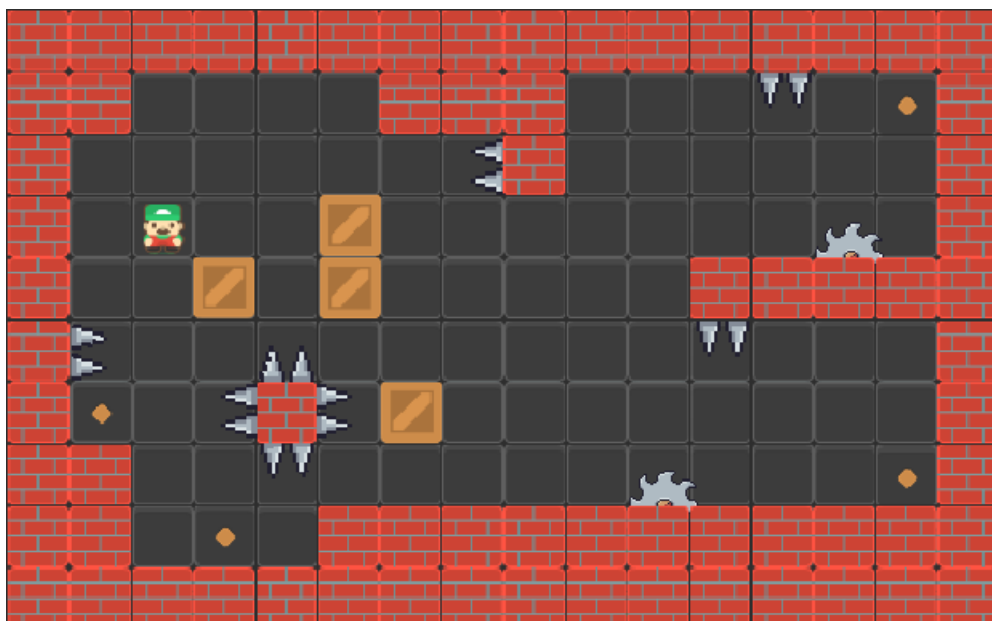


圖 4.5遊戲第三關地圖

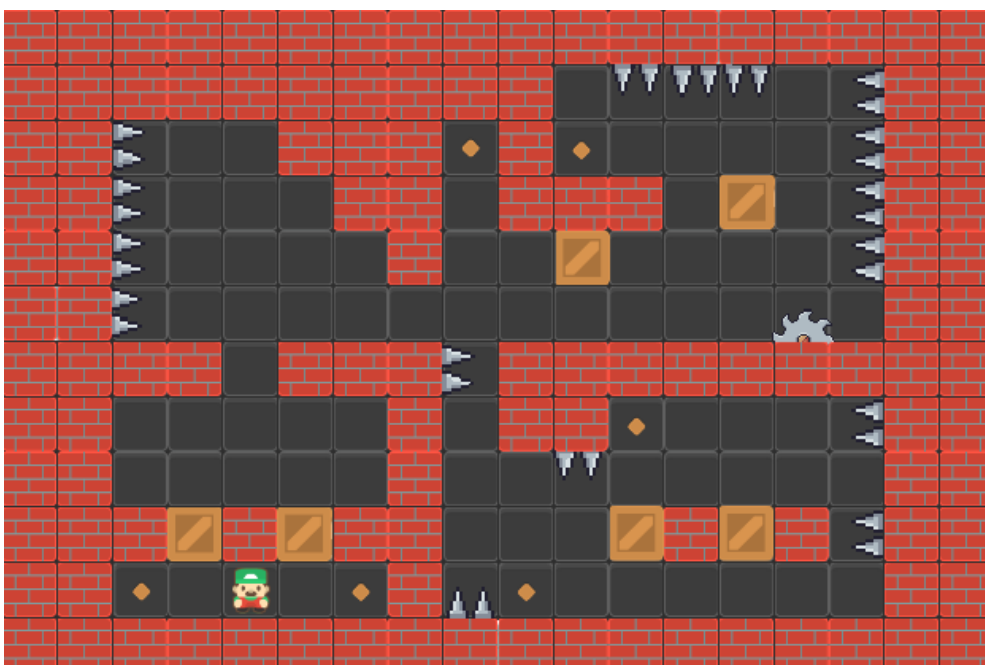


圖 4.6遊戲第四關地圖

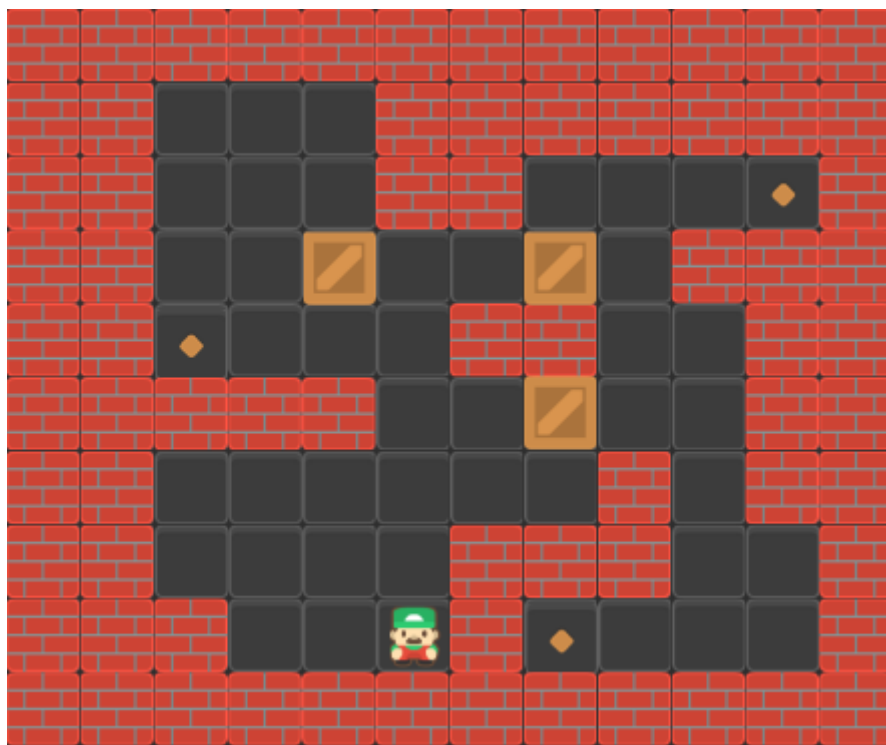


圖4.7遊戲第五關地圖

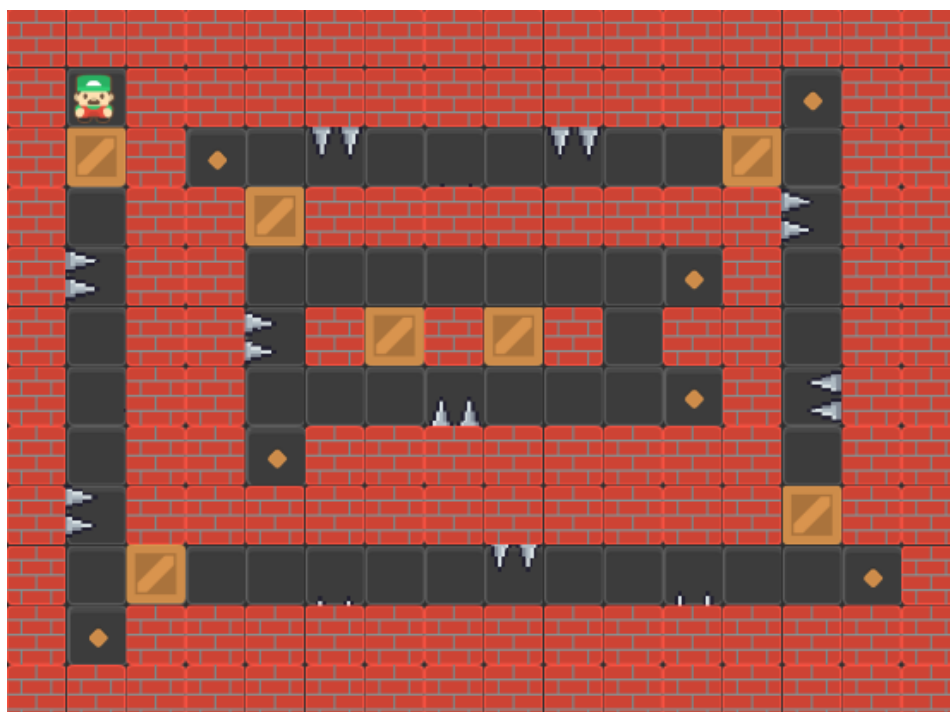


圖4.8遊戲第六關地圖

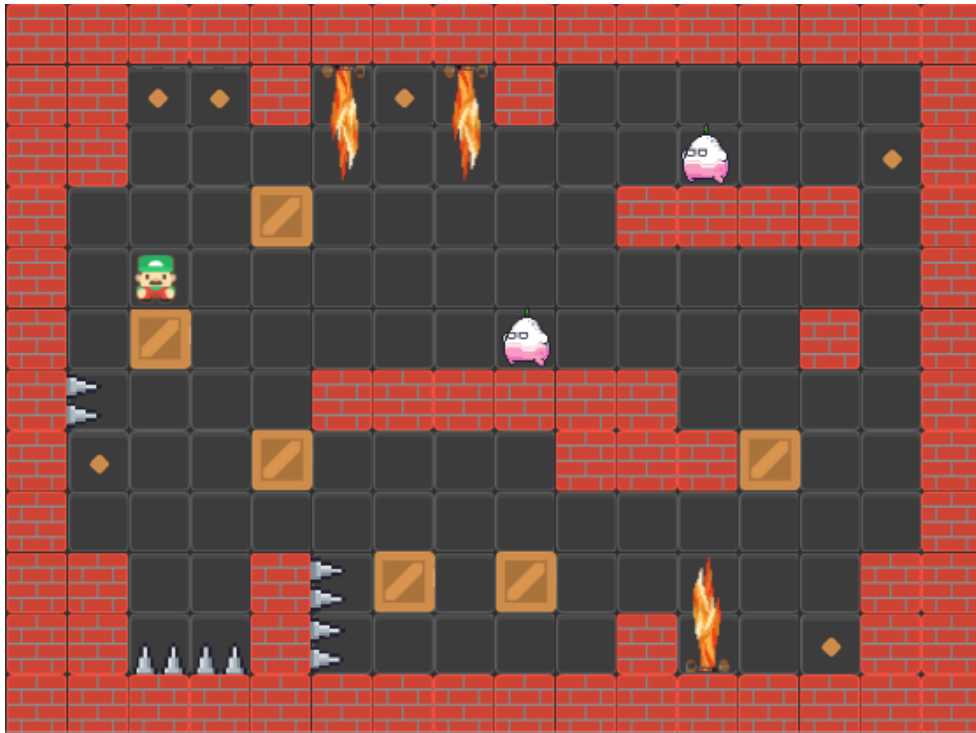


圖4.9遊戲第七關地圖

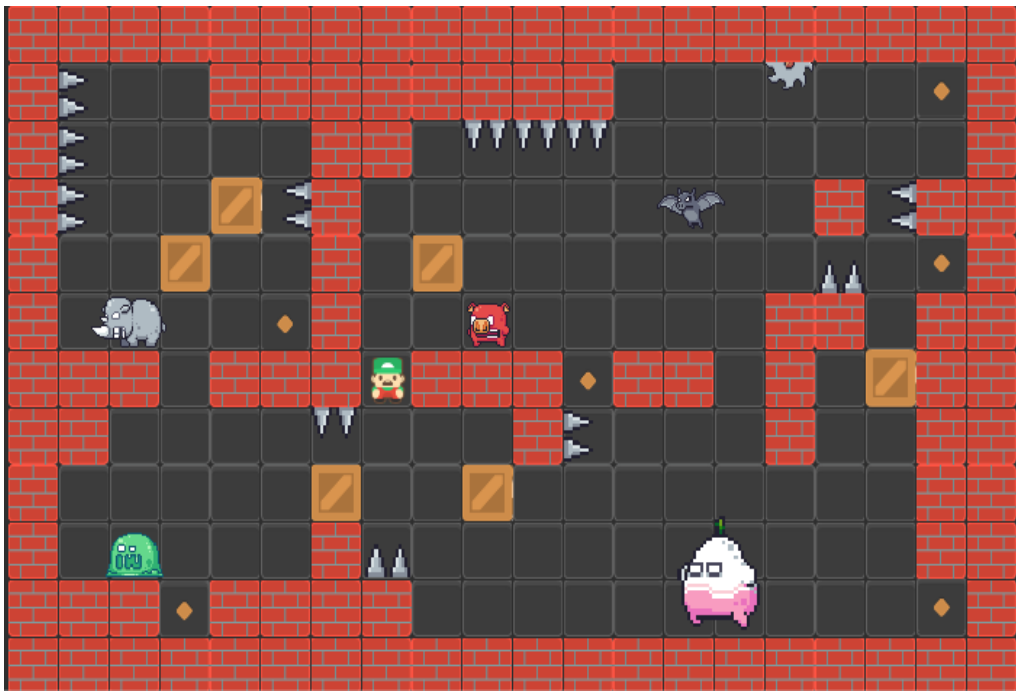


圖4.10遊戲第八關地圖

4.3怪物的動畫製作

Animator 是 Unity 負責動畫播放的元件，製作者可以透過多張靜態圖片製做出一個流暢的動畫。每一個 Animator 大部分都是由多個 animation 所組成。

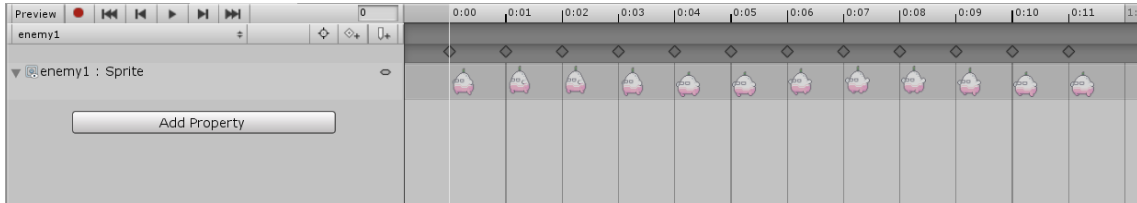


圖4.11怪物動畫製作圖1

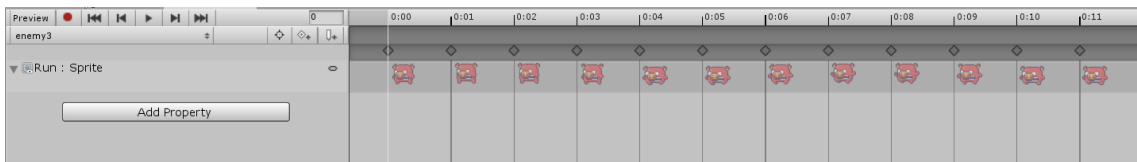


圖4.12怪物動畫製作圖2

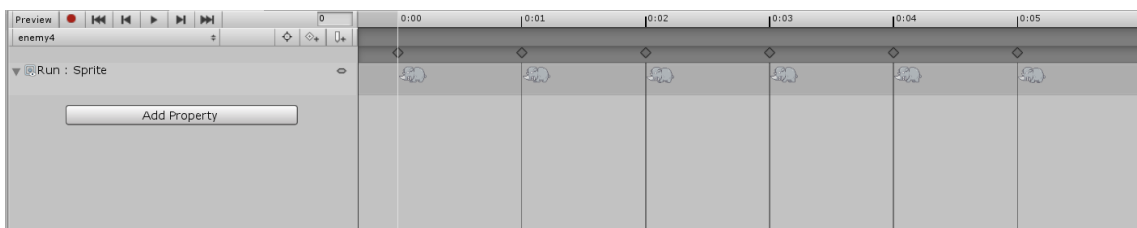


圖4.13怪物動畫製作圖3

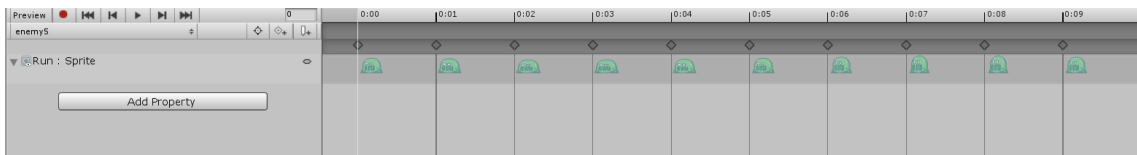


圖4.14怪物動畫製作圖4

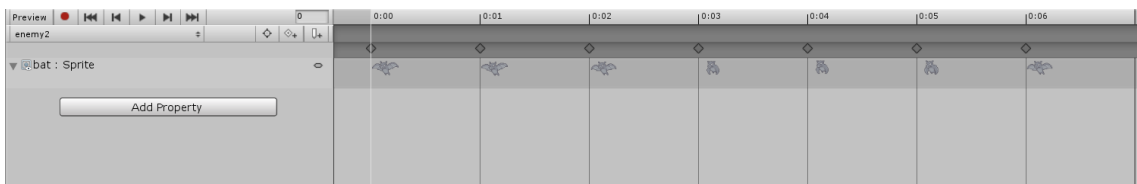


圖4.15怪物動畫製作圖5

第伍章 結論與未來展望

結論:

本專題使用 Unity 開發，AI、Photoshop 及 Unity Asset Store 內的素材來製作，使遊戲畫面更加豐富。而遊戲共設計了八個關卡，八個關卡都有各自的特色及通關的難度，讓人去思考關卡如何破解。

透過本次專題，我們了解到整個遊戲製作流程的架構，透過實際製作提升程式語言能力，我們不斷思考如何才能製作出受到喜愛的遊戲，在過程中遇到許多問題，像是設定物件的碰撞及陷阱移動上的錯誤都有發生，其實部分功能看似簡單，卻需要仔細思考才能實作，而我們現在仍有不少足或缺漏之處有待進步。

未來展望:

- 1.能夠設計出多變化的關卡，或是新增更多角色來讓關卡更有趣
- 2.希望能夠找些測試者來玩遊戲請他們提供想法或是建議，讓我們能夠做的更好
- 3.如果時間充足，我們會更加完善遊戲系統跟遊戲上的優化。

參考文獻

- [1] 倉庫番維基百科
<https://zh.wikipedia.org/zh-tw/%E5%80%89%E5%BA%AB%E7%95%AA>
- [2] 遊戲基地－Patrick's Parabox
<https://news.gamebase.com.tw/news/detail/99396224>
- [3] Healltaker維基百科
<https://zh.wikipedia.org/wiki/Helltaker>
- [4] CG數位網
http://www.cg.com.tw/Unity/Content/Unity_01.php
- [5] Unity3D的特色
<https://bbs.huaweicloud.com/blogs/detail/298015>
- [6] C#維基百科
<https://zh.wikipedia.org/wiki/C%E2%99%AF>
- [7] Adobe Illustrator維基百科
https://zh.wikipedia.org/wiki/Adobe_Illustrator
- [8] Adobe Photoshop維基百科
https://zh.wikipedia.org/wiki/Adobe_Photoshop
- [9] 巴哈姆特GNN新聞－地城謎蹤
<https://gnn.gamer.com.tw/detail.php?sn=208701>

附錄

本研究之主要功能的程式碼包含下列：

● 箱子射線檢測

```
public class box : MonoBehaviour
{
    public Color finishColor;
    Color originColor;

    [Unity Message | 0 個參考]
    private void Start()
    {
        originColor = GetComponent<SpriteRenderer>().color;
        FindObjectOfType<GameManager>().totalboxes++;
    }

    [1 個參考]
    public bool CanMoveToDir(Vector2 dir)
    {
        RaycastHit2D hit = Physics2D.Raycast(transform.position + (Vector3)dir * 0.4f, dir, 0.5f);

        if (!hit)
        {
            transform.Translate(dir);
            return true;
        }

        return false;
    }

    [Unity Message | 0 個參考]
    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("target"))
        {
            FindObjectOfType<GameManager>().finishedboxes++;
            FindObjectOfType<GameManager>().CheckFinish();
            GetComponent<SpriteRenderer>().color = finishColor;
        }
    }

    [Unity Message | 0 個參考]
    private void OnTriggerExit2D(Collider2D collision)
    {
        if (collision.CompareTag("target"))
        {
            FindObjectOfType<GameManager>().finishedboxes--;
            GetComponent<SpriteRenderer>().color = originColor;
        }
    }
}
```

● 角色移動及射線

```
public class playercontroller : MonoBehaviour
{
    Vector2 moveDir;
    public LayerMask detectLayer;

    0 個參考
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.RightArrow))
            moveDir = Vector2.right;

        if (Input.GetKeyDown(KeyCode.LeftArrow))
            moveDir = Vector2.left;

        if (Input.GetKeyDown(KeyCode.UpArrow))
            moveDir = Vector2.up;

        if (Input.GetKeyDown(KeyCode.DownArrow))
            moveDir = Vector2.down;

        if (moveDir != Vector2.zero)
        {
            if (CanMoveToDir(moveDir))
            {
                Move(moveDir);
            }
        }

        moveDir = Vector2.zero;
    }

    1 個參考
    bool CanMoveToDir(Vector2 dir)
    {
        RaycastHit2D hit = Physics2D.Raycast(transform.position, dir, 1.3f, detectLayer);

        if (!hit)
            return true;
        else
        {
            if (hit.collider.GetComponent<box>() != null)
                return hit.collider.GetComponent<box>().CanMoveToDir(dir);
        }

        return false;
    }

    1 個參考
    void Move(Vector2 dir)
    {
        transform.Translate(dir);
    }
}
```

●重置畫面及通關判定

```
public class GameManager : MonoBehaviour
{
    public int totalboxes;
    public int finishedboxes;

    Unity Message | 0 個參考
    private void Update()
    {
        if (Input.GetKeyDown(KeyCode.R))
            ResetStage();
    }

    1 個參考
    public void CheckFinish()
    {
        if(finishedboxes == totalboxes)
        {
            print("YOU WIN!");
            StartCoroutine(LoadNextStage());
        }
    }

    1 個參考
    void ResetStage()
    {
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex);
    }

    1 個參考
    IEnumerator LoadNextStage()
    {
        yield return new WaitForSeconds(2);
        SceneManager.LoadScene(SceneManager.GetActiveScene().buildIndex + 1);
    }
}
```

● 載入遊戲及離開遊戲

```
public class menucontroller : MonoBehaviour
{
    0 個參考
    public void LoadScene(string _name)
    {
        SceneManager.LoadScene(_name);
    }

    0 個參考
    public void QuitGame()
    {
        Application.Quit();
    }
}
```

●怪物的移動及圖片變換方向

```
public class Enemy : MonoBehaviour
{
    private Rigidbody2D rb;

    public Transform leftpoint, rightpoint; //設定左右點
    public float Speed; //輸入速度
    private float leftx, rightx;

    private bool Faceleft = true; //一開始敵人是面向左

    // Unity Message | 0 個參考
    void Start()
    {
        rb = GetComponent<Rigidbody2D>();
        transform.DetachChildren();
        leftx = leftpoint.position.x;
        rightx = rightpoint.position.x;
        Destroy(leftpoint.gameObject);
        Destroy(rightpoint.gameObject);
    }

    // Unity Message | 0 個參考
    void Update()
    {
        Movement();
    }

    // 1 個參考
    void Movement()
    {
        if (Faceleft) //判斷面向左走到點上後,轉向右
        {
            rb.velocity = new Vector2(-Speed, rb.velocity.y);
            if (transform.position.x < leftx)
            {
                transform.localScale = new Vector3(-1,1,1);
                Faceleft = false;
            }
        }
        else
        {
            rb.velocity = new Vector2(Speed, rb.velocity.y);
            if (transform.position.x > rightx)
            {
                transform.localScale = new Vector3(1, 1, 1);
                Faceleft = true;
            }
        }
    }
}
```


●彈出視窗及三個按鈕

```
public class PauseMenu : MonoBehaviour
{
    public static bool isGamePaused = false;

    [SerializeField] GameObject pauseMenu;

    @ Unity Message | 0 個參考
    void Update()
    {
        if (Input.GetKeyDown(KeyCode.Escape))
        {
            if (isGamePaused)
            {
                ResumeGame();
            }
            else
            {
                PauseGame();
            }
        }
        //按下Esc 暫停遊戲
        1 個參考
        public void ResumeGame()
        {
            pauseMenu.SetActive(false);
            isGamePaused = false;
        }

        1 個參考
        void PauseGame()
        {
            pauseMenu.SetActive(true);
            isGamePaused = true;
        }
        //繼續遊戲按鈕
        0 個參考
        public void LoadMenu(string _name)
        {
            SceneManager.LoadScene(_name);
        }
        //回到選擇關卡按鈕
        0 個參考
        public void QuitGame()
        {
            Application.Quit();
        }
        //結束遊戲按鈕
    }
}
```

綜合輔導紀錄表



110學年度 第2學期

教師綜合輔導紀錄表

填表日期：111年3月7日

教師姓名		林進源 教師		所屬系所	資訊管理系	
1	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)	吳永毅	資管4A	111年3月7日	共 1 時 0 分	
輔導項目	■ 專題 Unity 2D益智類遊戲 戲與開發		個人工作進度回報與任務指派： 製作地圖陷阱		討論結果： ☐ 1. 缺席 ☐ 2. 分數：83	
2	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)	陳冠成	資管4A	111年3月7日	共 1 時 0 分	
輔導項目	■ 專題 Unity 2D益智類遊戲 與開發		個人工作進度回報與任務指派： 尋找功能		討論結果： ☐ 1. 缺席 ☐ 2. 分數：83	
3	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)		資管4A	年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：		討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
4	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)		資管4A	年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：		討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
5	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)		資管4A	年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：		討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
輔導學生人次合計		2人	輔導時間合計	1 時 0 分	教師簽名	林進源

備註：

- 一、本表以月為單位。任課教師請於次月5日前將本表繳交至系(所)辦彙整，由系(所)辦彙整統計表後繳交教學發展中心。
- 二、本表留存各系評鑑備查，並請受輔導學生於專題複審前至系辦領回影本，放至專題報告書附錄之中。
- 三、學生簽名欄，由受輔導學生簽名。

教師綜合輔導紀錄表

填表日期：111年3月16日

教師姓名		林孟源 教師		所屬系所		資訊管理系	
1	輔導課程		學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)		吳永豪	資管4A	111年3月16日	共 1 時 30 分	
輔導項目	■ 專題 Unity 2D 益智類遊戲開發		個人工作進度回報與任務指派： 重新製作關卡，添加地面陷阱			討論結果： ☐ 1. 缺席 ☒ 2. 分數：82	
2	輔導課程		學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)		陳清成	資管4A	111年3月16日	共 1 時 30 分	
輔導項目	■ 專題 Unity 2D 益智類遊戲開發		個人工作進度回報與任務指派： 尋找功能			討論結果： ☐ 1. 缺席 ☒ 2. 分數：82	
3	輔導課程		學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)			資管4A	年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：			討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
4	輔導課程		學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)			資管4A	年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：			討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
5	輔導課程		學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)			資管4A	年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：			討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
輔導學生人次合計		2 人		輔導時間合計		1 時 30 分	
				教師簽名		林孟源	

備註：

- 一、本表以月為單位。任課教師請於次月5日前將本表繳交至系(所)辦公室，由系(所)辦公室統計表後撥交教學發展中心。
- 二、本表留存各系評鑑備查，並請受輔導學生於專題複審前至系辦領回影本，放至專題報告書附錄之中。
- 三、學生簽名欄，由受輔導學生簽名。



4 月份

學 年 度 第 學 期

教師綜合輔導紀錄表

填表日期：111 年 4 月 18 日

教師姓名		林益源		所屬系所	資訊管理系	
1	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)	吳承毅	資管 4A	111 年 4 月 18 日	共 1 時 40 分	
輔導項目	■ 專題 Unity 2D 益智類遊戲 研究與開發		個人工作進度回報與任務指派： 討論 word、ppt 及回報進度		討論結果： ☐ 1. 缺席 ☒ 2. 分數：88	
2	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題(三)	陳嘉成	資管 4A	111 年 4 月 18 日	共 1 時 40 分	
輔導項目	■ 專題 Unity 2D 益智類遊戲 研究與開發		個人工作進度回報與任務指派：		討論結果： ☒ 1. 缺席 ☐ 2. 分數：8	
3	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題()			年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：		討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
4	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題()			年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：		討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
5	輔導課程	學生簽名	班級	輔導日期	輔導時間(分)	
	實務專題()			年 月 日	共 時 分	
輔導項目	■ 專題		個人工作進度回報與任務指派：		討論結果： ☐ 1. 缺席 ☐ 2. 分數：_____	
輔導學生人次合計		1 人	輔導時間合計	1 時 40 分	教師簽名	林益源

備註：

一、本表以月為單位。任課教師請於次月 5 日前將本表繳交至系(所)辦彙整，由系(所)辦彙整統計表後繳交教學發展中心。

二、本表留存各系評鑑備查，並請受輔導學生於專題複審前至系辦領回影本，放至專題報告書附錄之中。

三、學生簽名欄，由受輔導學生簽名。