

致謝

專題當中首先要感謝的是專題指導老師 蕭俊杰老師。在大二的時候就選我們為專題學生，專題製作的期間定期開會關注我們專題進度，就連暑假期間也撥出時間與我們線上開會。老師也從題目開始引導我們構想，在專題製作的過程當中，遇到的程式問題也指導我們許多，這也讓我們從一開始遇到問題會卡關很久，慢慢地變成能自己思考並解決問題，讓我們的程式能力加強許多。

組員也是很重要的，在每位組員都會把各自所該做的事情努力完成，同時也在要求的時間內做好，並且能符合老師所要求的進度才能讓我們專題很順利進行，製作的過程各個組員發生困難時，其他組員也都盡心盡力的去幫忙解決，讓我們遊戲製作可以如期完成。

還要感謝游博捷學長，在我們一開始學習Unity的時候，指導了我們很多，讓我們在剛開始不熟悉的狀態下能快速上手。也謝謝同班同學幫我們進行遊戲的試玩，提供了我們遊戲的問題，讓我們進行更改，並且也在專題製作的期間大家互相勉勵。

最後要感謝龍華科技大學，除了有著優秀的師資教導我們各項專業能力，讓我們能從不懂任何程式，變成能製作出屬於我們的一款遊戲，還提供了研究室給我們製作專題使用。

摘要

本研究的動機是因應高齡化社會的邁進，老年人比例的增高，如果能延緩老年人退化的現象，那麼我們的遊戲構想將會是很好的設計，成品會以軟體遊戲呈現。如上述動機，在遊戲構想方面，主要以反應力、專注力及認知能力的遊戲，讓老年人能夠在娛樂的同時，也能延緩高齡化社會中，多個能力的退化。主題是為老年人設計，從中有許多復古懷舊的畫面設計。這次使用的軟體以Unity為主體，伴隨著C#程式及3Ds Max的建模，搭配小畫家及線上簡單去背去完成的復古軟體設計。再給予家中長輩、系上同學及周遭朋友試玩遊戲的回饋下，有許多配置更偏向認知能力及反應力，考量是專注力對於老年人的體力受限。

遊戲分為主線遊戲及五個副線遊戲，遊玩時分為兩個關卡，主線遊戲是希望玩家能夠從頭玩到尾，過程中給予各個副線遊戲過關的線索，或是給予額外的加分。主線遊戲為「關卡遊戲」，以玻璃彈珠球為玩家主體，以物理引擎利用鍵盤控制轉動方向，度過重重難關，直到遊玩結束；其中五個副線遊戲分別分佈到兩個關卡，第一關的遊戲為：「(1)陀螺遊戲：『以舊磁磚為底山丘野草為背景，然後控制我們的彈珠，將陀螺頭上數字順序從小到大一一觸碰完畢即可過關，以便增進反應力及認知能力。』」、(2)骰子遊戲：『分為4個小遊戲，有些許玩法有做變更，增進認知能力。』及(3)象棋遊戲：『從黑到紅，從大到小開始依序選擇，直到遊戲結束。』」，第二關的遊戲為：「(4)猜燈謎遊戲：『以燈籠為觸發，各點都有題目讓玩家思考，給予玩家腦筋急轉彎』及(5)圖片問答遊戲：『讓玩家辨別圖片並選取答案的正確性，給予玩家有想當的認知能力。』」。

關鍵字:高齡化社會、老年人、退化、復古、反應力、認知能力、軟體、遊戲

Abstract

The motivation of this study is to advance the aging society, Increase the proportion of the elderly, if it can delay the degradation of the elderly, then our game concept will be a very good design, The finished product will be presented as a software game. The above motivation, In terms of game concept, the game is mainly responsive, focused and cognitive, Allow older people to entertain while, It can also delay the aging society, Degradation of multiple capabilities. Themes are designed for the elderly, so there are many retro vintage photo designs. The software used this time is roughly based on Unity. with the C# program and 3Ds Max modeling, with a small artist and online simple backs to complete the retro software design. Give home elders and classmates and friends around under trial game feedback, there are many configurations that are more cognitive and reactive, considering that concentration is limited to the elderly.

The game is divided into main lines and five small levels, The main mission is to hope that players can play from beginning to end, Ever since gave each small game clue, Or give extra points. The main line is the "Roll game", with glass marbles as the control element, and the physics engine uses the keyboard to control the direction of rotation and encounters heavy difficulties until the end of the game; Five of the small levels are (1) Gyro Games: Use old tiles as background for hills and weeds, Then control our marbles, The numerical order of the gyro heads from small to large all touched can be cleared, In order to increase responsiveness and cognitive ability.(2) Divided into 4 mini-games, some changes in gameplay are made to improve cognitive ability.(3) Chess game: From black to red, choose from big to small until the end of the game.(4) Guess the riddle game: Triggered by lanterns, each point has a title to allow players to think and give players a brain teasers.(5) Picture quiz game: Allows the player to identify the correctness of the options and options in the diagram, giving the player the desired cognitive ability.

Keywords: Ageing society, Seniors, Degradation, Nostalgia, reaction, cognitive ability, software, game

目錄

致謝	I
摘要	II
ABSTRACT	III
目錄	1
圖目錄	3
程式目錄	7
表格目錄	8
第一章、前言	9
1-1 研究動機與背景	9
第二章、文獻探討	10
第三章、使用技術	11
3-1 UNITY	11
3-1-1 Animator	11
3-1-2 Audio	14
3-1-3 Camera	16
3-1-4 Collider	18
3-1-5 Materials	20
3-1-6 GameObject	22
3-1-7 GUI	27
3-1-8 Light	35
3-1-9 Scenes	37
3-1-10 Transform	39
3-1-11 PlayerPrefs	41
3-1-12 Terrain	43
3-2 3Ds MAX	47
3-2-1 模型建設	47
3-2-2 可編輯多邊形	49
3-2-3 連接	52
3-2-5 Loft	55
3-2-6 布林計算運用	57
3-2-7 材質編輯器	59
3-3 語音處理辨識及輸出	62
3-3-1 語音辨識	62
3-3-2 語音輸出	68
第四章、專題成果簡介	71
4-1 第一關簡介	71
4-2 第二關簡介	74
第五章、第一關遊戲	75
5-1 關卡遊戲	75
5-2 陀螺遊戲	82
5-3 骰子遊戲	84
5-4 象棋遊戲	89
第六章、第二關遊戲	90

6-1 關卡遊戲	90
6-2 猜燈謎遊戲	91
6-3 圖片問答遊戲	92
第七章、結論 94	
7-1 工作時程	94
7-2 研究結果	95
7-3 專題製作心得	96
7-3-1 黃緯皓心得	96
7-3-2 張家寶心得	96
7-3-3 楊宏隆心得	96
7-3-4 張簡信成心得	96
7-4 未來展望	98
7-5 注意事項	98
參考文獻	99

圖目錄

第三章

圖3- 1門開關動畫編輯	11
圖3- 2動畫編程	12
圖3- 3動畫參數指向圖	12
圖3- 4 AUDIO 音效匯入圖	14
圖3- 5 CAMERA與PLAYER位置圖	16
圖3- 6視角差別圖	16
圖3- 7錢幣圖	18
圖3- 8湖泊圖	18
圖3- 9創建材質球	20
圖3- 10空白材質球	20
圖3- 11匯入貼圖材質	21
圖3- 12 CREATE EMPTY	22
圖3- 13 GAMEOBJECT[]介面	23
圖3- 14螢幕座標	23
圖3- 15螢幕座標版本	24
圖3- 16預設位置	25
圖3- 17陀螺顯示	26
圖3- 18 GUI	27
圖3- 19 TEXT(INSPECTOR)	28
圖3- 20 IMAGE(INSPECTOR)	28
圖3- 21 圖片丟入UNITY ASSET資料夾內	29
圖3- 22選取圖片後的(INSPECTOR)	29
圖3- 23 選取圖片後 (INSPECTOR IMAGE(SCRIPT))	29
圖3- 24 BUTTON(INSPECTOR)	30
圖3- 25 BUTTON的INSPECTOR中常用功能	31
圖3- 26解析度差別	31
圖3- 27 CANVAS(INSPECTOR)	32
圖3- 28 CANVAS (INSPECTOR中的[CANVAS SCALER(SCRIPT)])	32
圖3- 29 UNITY光影	35
圖3- 30 LIGHT選擇	35
圖3- 31 POINT LIGHT使用	36
圖3- 32增加光源呈現效果	36
圖3- 33 SCENES匯出匯入	38
圖3- 34 TRANSFORM介面	39
圖3- 35 數字視角相反	40
圖3- 36 排行榜顯示	42
圖3- 37 TERRAIN介面	43
圖3- 38地形座標旋轉前	44
圖3- 39地形座標旋轉後	44
圖3- 40地圖應該旋轉的方向	45
圖3- 41匯入高度圖之後旋轉的地形	45
圖3- 42匯出的高度圖	45
圖3- 43 IMPORT HEIGHTMAP	46
圖3- 44硬幣拆UV結果	47
圖3- 45硬幣貼UV結果	47
圖3- 46可編輯多邊形	49
圖3- 47頂點	49
圖3- 48邊	49
圖3- 49可編輯多邊形功能表	50

圖3- 50尚未切片前的多邊形	50
圖3- 51切片後的多邊形	50
圖3- 52多邊形	50
圖3- 53多邊形功能表	51
圖3- 54擠出後的多邊形	51
圖3- 55連接列表	52
圖3- 56沒有連接物件的列表	53
圖3- 57沒有連接的物件圖	53
圖3- 58連接後的物件圖	53
圖3- 59起始關鍵點	54
圖3- 60關鍵點結尾	54
圖3- 61設置關鍵點	54
圖3- 62放樣	55
圖3- 63放樣功能	55
圖3- 64線與斷面	56
圖3- 65獲取圖形	56
圖3- 66圖形A與B	57
圖3- 67差集	57
圖3- 68交集	58
圖3- 69材質編輯器	59
圖3- 70空白材質球	59
圖3- 71偏移選項	60
圖3- 72導出選項	61
圖3- 73嵌入式媒體勾選	61
圖3- 74套件範例	62
圖3- 75範例辨識	62
圖3- 76 關卡初始圖	65
圖3- 77 辨識成功圖	65
圖3- 78 辨識錯誤圖	66
圖3- 79第二關圖	67
圖3- 80 GOOGLE TRANSLATE TTS 網址	68
圖3- 81 ANDRIOD模式	69
圖3- 82 WINDOWS 模式	69
圖3- 83 其他API程式 WINDOWS 模式	70
圖3- 84其他範例程式ANDROID模式	70
圖3- 85 GOOGLE TTS MP3檔	70

第四章

圖4- 1第一關俯瞰圖	71
圖4- 2陀螺遊戲	72
圖4- 3象棋遊戲	72
圖4- 4骰子遊戲介面	73
圖4- 5第二關	74
圖4- 6懷舊問答	74

第五章

圖5- 1第一關俯瞰圖	75
圖5- 2初始畫面	76
圖5- 3提示字串出現	77
圖5- 4吃金幣計分	77
圖5- 5水橋過河機關	78
圖5- 6尖刺機關	79
圖5- 7寶箱圖	79

圖5- 8搭船渡河	80
圖5- 9熊	80
圖5- 10鐵柵欄和骰盅	81
圖5- 11竹蜻蜓	81
圖5- 12簡單模式	82
圖5- 13一般模式	83
圖5- 14困難模式	83
圖5- 15骰子介面	84
圖5- 16原構想	84
圖5- 17遊戲介面	85
圖5- 18 懷舊畫面設計	87
圖5- 19骰子畫面設計	87
圖5- 20畫面設計	88
圖5- 21最後懷舊成品	88
圖5- 22象棋遊戲	89

第六章

圖6- 1第二關俯瞰圖	90
圖6- 2第二關初始	91
圖6- 3答完爬上高處	91
圖6- 4遊戲初始畫面	92
圖6- 5遊戲中畫面	93

第七章

圖7- 1專題行程甘特圖。	94
圖7- 2專案基本流程圖。	95

程式目錄

第三章

程式3- 1 開門動畫程式	12
程式3- 2 開門動畫程式2	13
程式3- 3 AUDIO CLIP音效程式圖	14
程式3- 4 調用音檔圖	15
程式3- 5 AUDIO調用副程式圖	15
程式3- 6 CAMERA與PLAYER相對位置程式圖	17
程式3- 7 錢幣觸發程式	18
程式3- 8 PUBLIC GAMEOBJECT	22
程式3- 9 BTSTART	24
程式3- 10 預設物件位置	25
程式3- 11 陀螺小遊戲	26
程式3- 12 宣告公共函式寫法	30
程式3- 13 TOGGLE及SLIDER連動程式碼	31
程式3- 14 判斷執行次數	33
程式3- 15 題目宣告(SWITCH CASE)	33
程式3- 16 給予骰子轉動的力	34
程式3- 17 過場不刪除	37
程式3- 18 TAG與刪除	37
程式3- 19抓取場景	38

程式3- 20物件新座標	39
程式3- 21 轉變物件座標	40
程式3- 22 陀螺轉動	40
程式3- 23 起始存取排行榜	41
程式3- 24 計算排行榜	42
程式3- 27原按鍵程式	63
程式3- 28修改按鍵後	63
程式3- 29字串顯示	64
程式3- 30修改後字串顯示	64
程式3- 31題目答案陣列	67
程式3- 32觸發亂數取題目	67
程式3- 26 TEXT TO SPEECH	68

表格目錄

第三章

表3- 1 碰撞器即觸發器說明

19

第1章、前言

1-1 研究動機與背景

隨著醫療技術日益進步，人們的平均壽命也不斷延長，高齡人口也持續增加，再加上少子化的衝擊，高齡人口比例不斷攀升。根據世界衛生組織WHO之定義，「高齡化社會(Ageing Society)」為65歲以上人口佔總人口比例達7%、「高齡社會(Aged Society)」為比例達14%、「超高齡社會(Super-Aged Society)」則為比例達20%。自1993年起，台灣便已開始邁入了高齡化社會，隨著時代演進，台灣人口結構失衡的問題持續放大。國家發展委員會也推估，2018年台灣將會跨入「高齡社會」，甚至預估到2026年(民國115年)，台灣的老年人口將突破20%門檻，走向超高齡社會。

生老病死，是生為人皆無法避免的嚴肅課題。而人類身心狀況可視為「健康、亞健康、失能」的連續體，前二階段有賴積極的「預防照顧」保持健康並延緩老化，最後的失能階段則得依靠消極的「長期照顧」維持生活。為因應民眾的照護需求，政府也提出建立長照服務體系、居家照護等對策，以減少城鄉差距，同時提出強化社區照顧據點，適時依據點位置提供老人服務，提高服務效能，並建立整合性的服務網絡及其他輔助方案。

相對的，「預防照顧」較不需國家角色直接介入，一如「世界衛生組織」(WHO)在二〇〇二年提出的「活躍老化」(Active Ageing)概念：藉由參與家庭、社區、同儕、社會等社交活動，延緩銀髮族退化速度，維持身體與心靈雙重活力。

為維持銀髮族之身體與心靈活力，本研究計畫提出以設計適合銀髮族使用之多重模式遊戲，來增進銀髮族之認知能力並預防失智。

第2章、文獻探討

由於少子化及醫療技術之日益進步，高齡人口比例不斷攀升。據國家發展委員會之推估[1]，2018年台灣將會跨入「高齡社會」，甚至預估到2026年（民國115年），台灣的老年人口將突破20%門檻，走向超高齡社會。內政部統計處之資料亦顯示[2]，老化指數（即每100個65歲以上人口對14歲以下人口之比）已經於2017年2月正式跨越100%，再再顯示我國人口老化之嚴重性。

人口老化造成之問題眾多，其中最重要的，不外乎若長者快速老化而生病，將需要「長期照護」，因而造成家庭、社會及國家之沉重負擔。針對延緩老化，全球已經有許多研究團隊相繼投入研究。如[3-12]之研究中，便提出延緩長者之認知能力下降，有助於避免或延遲失智症之罹患。而為延緩長者之認知能力下降，許多研究團隊亦提出相對應之方法，如國際標準交際舞(International Ballroom Dancing)[13]、音樂治療(Music Therapy)[14]、機器人治療(Robot therapy)[15-17]及認知行為治療(Cognitive Behavioral Therapy, CBT)[18]等，皆可刺激認知，並進而延緩認知能力下降及避免失智症。

有更多之研究團隊，對長者玩遊戲是否能延緩認知能力下降，做了深入的探討。如Gamberini, Luciano, et al [19]中，便指出每日適當玩遊戲，可帶來心理健康。而Wang, Ping, et al [20]中，更發現長者（60歲以上）玩遊戲，能有效延緩認知能力下降。體感遊戲(Exergames [21-24] or Somatosensory Games [25-26])亦被證實對延緩認知能力下降有幫助。輔助學習太極拳之嚴肅遊戲(Serious Games)[27]，亦顯示該遊戲有助保持長者認知能力。

Wang, Ping, et al [20]中亦發現，一般之電動遊戲(Video Games)，對銀髮長者之重要認知能力如處理速度(Processing Speed)及視覺空間處理能力(Visuospatial Processing)，有較大之幫助，對執行功能(Executive Function)及注意力(Attention)之幫助次之，而對記憶力(Memory)則較無幫助。Orange, Joseph B., and Angela Colton-Hudson [28]亦提出，語言溝通(Communication)對維持認知能力有很大的幫助。

本研究計畫擬針對銀髮族之五項重要認知能力（處理速度、視覺空間處理能力、執行功能、注意力及記憶力），設計開發一套適合銀髮族使用之多重模式遊戲，並整合Google 語音辨識及語音合成技術 [29]，誘導使用者多使用語音與系統互動，以延緩其認知能力退化，並避免失智症。

第3章、使用技術

3-1 Unity

Unity是一個遊戲製作引擎，能製作出2D與3D的遊戲，並且匯出成不同平台上的遊戲，所能匯出的平台有Windows、Linux、iOS、Android、Xbox、PS Vista、PS4、Universal Windows Platform、WebGL、Facebook，在我們初期原本是製作Android平台的手機遊戲，由於考量到效能的問題，變成了製作成Windows的PC遊戲，Unity大多使用C#與JavaScript兩種程式語言製作遊戲，本專題遊戲使用C#製作，也能匯入從其他軟體製作的模型，本專題使用3Ds Max製作模型並且匯出成FBX檔案套用至Unity。

3-1-1 Animator

在Unity中我們使用Animator功能製作遊戲動畫，我們在遊戲物件中編輯動畫可在Clips剪輯多個動畫且可以設定不同時間節點做不同的動作如圖3- 1為門由關閉變為打開的畫面，製作完成後放進Animator內做編成如圖3- 2，編程完後可設定細節如速度、時間等，完成後在Parameters新增參數放置Conditions如圖3- 3，即可在程式中指向功能。

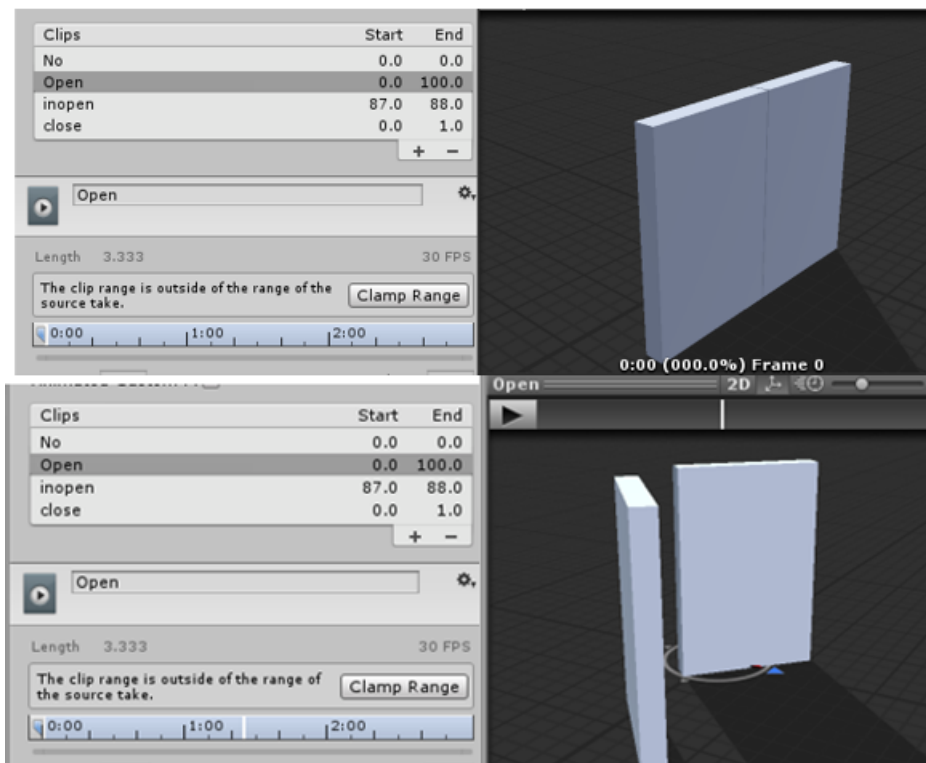


圖3- 1門開關動畫編輯

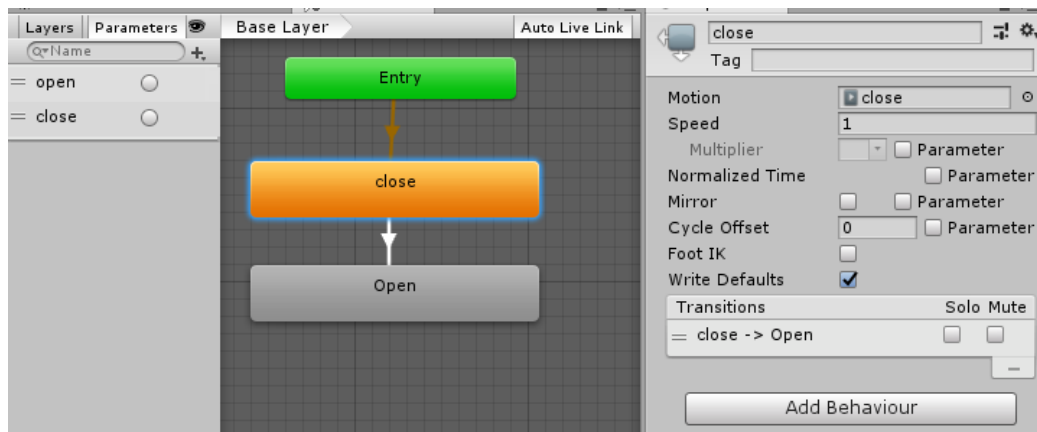


圖3- 2動畫編程

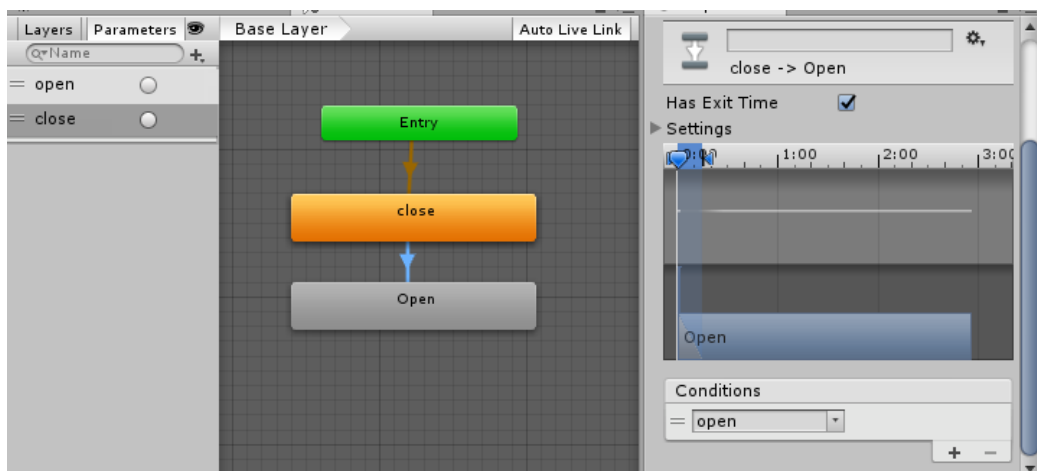


圖3- 3動畫參數指向圖

在程式3- 1第24行程式碼『public Animator doorPlay』, 並在第123行程式碼『doorPlay.SetTrigger(" open ")』, 完成動畫執行設定, 另一種寫程式的方法如程式3- 2第124行程式碼『GetComponent<Animator>().SetTrigger(" open ")』即可完成動畫執行設定。

兩者差別於第一種程式碼必須公開繼成動畫物件後套用於程式碼中, 雖然程序比較多, 但是較為方便簡短好管理且清楚明瞭, 第二種程式碼直接觸發參數 PlayBoat, 雖然只有單單一行指令, 但沒有寫是哪個動畫物件指引, 如果在多人共同執行的專案中如有多個相似指令不易察看功能或排除故障。

```

24      public Animator doorPlay;
25      public Animator doorPlay2;

121     if (Example._speechRecognitionResult.text == "芝麻開門")
122     {
123         doorPlay.SetTrigger("open");

```

程式3- 1 開門動畫程式

```

122     if (Example._speechRecognitionResult.text == "芝麻開門")
123     {
124         GetComponent<Animator>().SetTrigger("open");

```

程式3- 2 開門動畫程式2

3-1-2 Audio

Audio功能是用來播放聲音的物件，我們使用來播放背景音樂及音效，在Main Camera中會自動產生Audio Listener組件，這是必備的聽聲器，就如同人類的耳朵一般，我們將Audio Source設置在Player上，讓Player走到哪裡音樂就放到哪裡。如果只有一個聲音來源可以直接將音源拖入物件上，物件也會自動產生一個Audio Source，由於專案中同時存在多個Audio Source也不能在程式碼中控制選擇播放哪個聲音，所以我們使用Audio clip 來做為音效的播放，在程式3- 3第9行公共繼承音頻片段陣列來儲存多個音效檔案，如圖3- 4中Clips 內使用兩個音效，可以在第186行調用AudioSource播放片段陣列1於三軸的位置並且以100%的聲音大小播放出來，播放完畢後會直接移除遊戲中的音效，就不會與Audio Source直接播放的音樂產生衝突。



圖3- 4 Audio 音效匯入圖

```

9      public AudioClip[] clips = { };
186    AudioSource.PlayClipAtPoint(clips[1], new Vector3(65, 12, 20), 10.0f);

```

程式3- 3 Audio clip音效程式圖

我們也對於Audio Source播放音檔做一個調用程式以方便主要程式選取播放音檔，如程式3- 4 在第63行以簡便的方式調用music Script，播放名為『BGM1』的音檔，則music檔中PlayMusic的副程式如程式3- 5 第6行公共繼承常數字串MusicPath 路徑名為『audio/』的資料夾，在第7行公共繼承靜態的Audio Source名為InMusic等於空值來做調用，在第8行做公共繼承靜態副程式名為PlayerMusic(字串名稱)，字串名為音源檔名，在第10行判斷InMusic是否為空，執行InMusic等於在Main Camera物件新增Audio Source組件，第14行為InMusic的片段設為參照目錄為MusicPath資料夾內的音檔名稱，第15行為InMusic音源的聲音設為10%，在第16行播放音檔。

```

61      void Start()
62      {
63          music.PlayMusic("BGM1");
64          GameObject DontDestroyOnLoad(gameObject);

```

程式3- 4 調用音檔圖

```

5  public class music : MonoBehaviour {
6      public const string MusicPath = "audio/";
7      public static AudioSource InMusic = null;
8      public static void PlayMusic(string name)
9      {
10         if (InMusic == null)
11         {
12             InMusic = Camera.main.gameObject.AddComponent<AudioSource>();
13         }
14         InMusic.clip = Resources.Load<AudioClip>(MusicPath + name);
15         InMusic.volume = 0.1f;
16         InMusic.Play();
17     }
18 }

```

程式3- 5 Audio調用副程式圖

3-1-3 Camera

Camera是Unity呈現非常重要的環節，可以設定多個Camera來做切換或是套用於物件，Camera在3D專案中以球狀一般可以轉向任何視角，我們將Camera套用於Player物件斜上方60度角的位置如圖3- 5，設定位置後我們寫程式將Camera與Player一直保持於相對位置，讓Camera可以一直跟隨Player移動做為遊戲視角，如程式3- 6 中 第7行公共繼承物件Player以套用於程式中，在第8行公共繼承三軸offset以抓取位置，第11行設定程式執行初始三軸位置等於攝影機的坐標減去玩家的坐標，變成相對位移位置，在程式執行當中第17行將攝影機的坐標等於玩家移動的坐標加相對位移的坐標，攝影機就會跟著移動，第20行為判斷按下按鈕Z之後攝影機相對位置就會更改為第22~24行寫入的三軸位置，以便攝影機近遠控制如圖3- 6中顯示左邊為按下X鍵後視角較近圖，右邊為原始執行圖。

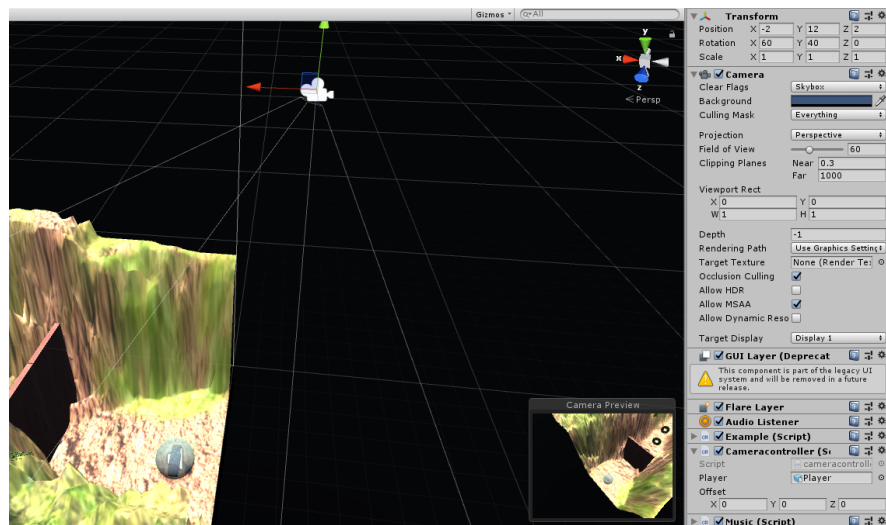


圖3- 5 Camera與Player位置圖

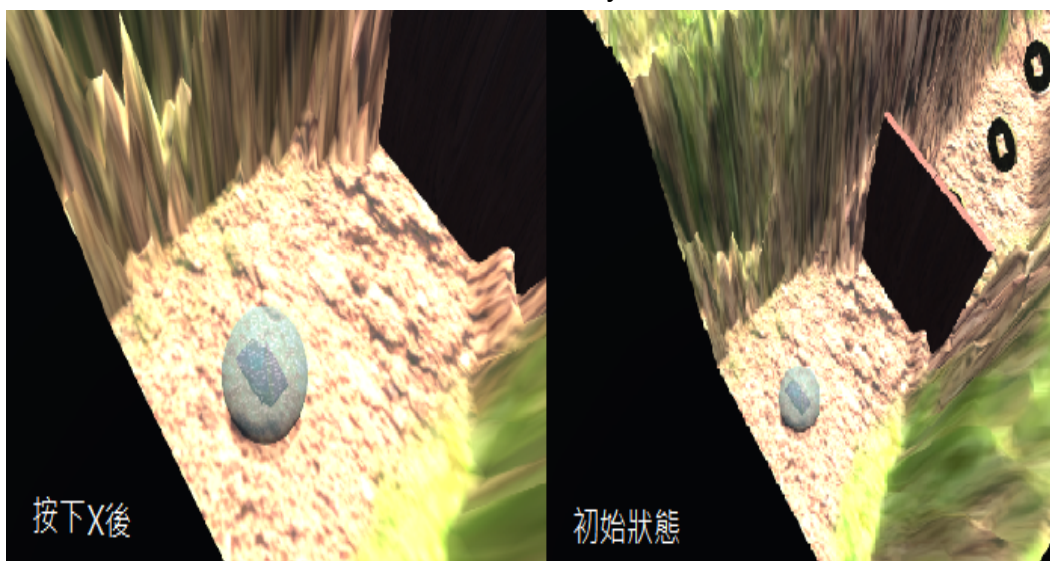


圖3- 6視角差別圖

```

7 public GameObject player;
8 public Vector3 offset;
9 void Start()
10 {
11     offset = transform.position - player.transform.position;
12     //相對位移=攝影機的座標-玩家的座標;
13     GameObject.DontDestroyOnLoad(gameObject);
14 }
15 void Update()
16 {
17     transform.position = player.transform.position + offset;
18     //攝影機的座標=玩家的座標+相對位移;
19
20     if (Input.GetKeyDown(KeyCode.Z))
21     {
22         offset.x = -3;
23         offset.y = 9.5f;
24         offset.z = -3;
25     }
26     if (Input.GetKeyDown(KeyCode.X))
27     {
28         offset.x = -2;
29         offset.y = 5.5f;
30         offset.z = -2;
31     }

```

程式3- 6 Camera與Player相對位置程式圖

3-1-4 Collider

Collider是我們在Unity最常使用的功能，我們在金幣、湖泊、陀螺、關卡道具等都使用到此功能如圖3- 7，利用Collider中再勾選Is Trigger變成觸發器，在此使用Is Trigger的狀態是為了讓玩家物件能直接穿越碰撞物體，如果沒有Is Trigger玩家物件會因為物理引擎的效果往後彈，並設定Tag來做為程式控制內容，如程式3- 7中第26行為宣告觸發器、並在第28行設如果玩家碰撞到錢幣則執行以下功能，圖3- 8 中如墜落湖泊也是利用觸發器製作，不同的是墜入湖泊還有使用Transform功能定位。



圖3- 7錢幣圖

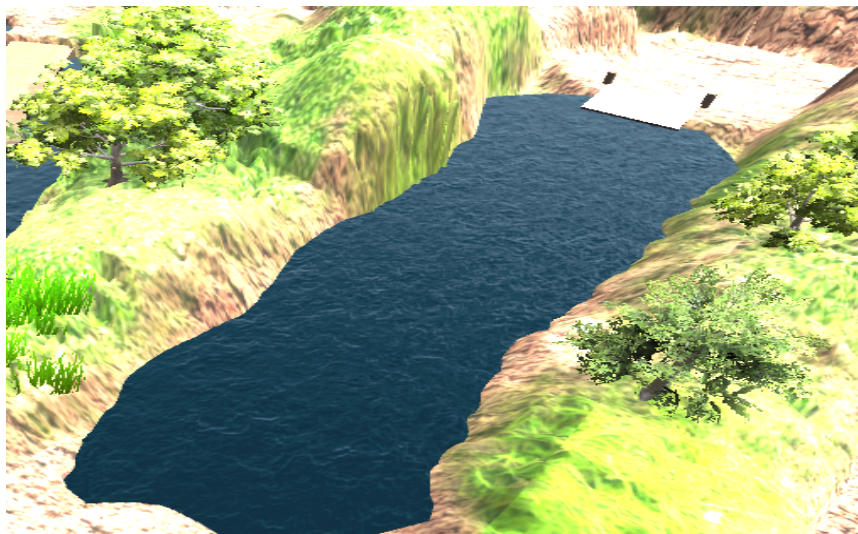


圖3- 8湖泊圖

```
26 void OnTriggerEnter(Collider player)
27 {
28     if (player.gameObject.CompareTag("Money"))
29     {
30         player.gameObject.SetActive(false);
31         Count = Count + 1;
32         SetCountText();
33         AudioSource.PlayClipAtPoint(clips[0], new Vector3(5, 12, 5), 5.0f);
    }
```

程式3- 7 錢幣觸發程式

在程式碼中使用Collider 有三種觸發方式及三種碰撞方式如表3- 1，我們大多

都使用第四種OnTriggerEnter, 可依照遊戲設計方式來選擇使用的方式。

表3- 1 碰撞器即觸發器說明

OnCollisionEnter	當碰撞體『開始』接觸另一個碰撞體時會被調用
OnCollisionExit	當碰撞體『停止』接觸另一個碰撞體時會被調用
OnCollisionStay	當碰撞體一直持續接觸另一個碰撞體時會被調用
OnTriggerEnter	當碰撞器對方『進入』觸發器時會被調用
OnTriggerExit	當碰撞器對方『停止』觸碰觸發器時會被調用
OnTriggerStay	當碰撞器持續在觸發器內時會被調用

3-1-5 Materials

Materials就是材質的意思，在Unity許多物件都會使用到材質，通常材質都會在3Ds Max裡面會好才會上傳，但是有時候材質會跑掉，這時候就要在Unity裡面編輯或者創建材質球如圖3- 9，剛創建的材質球是沒有顏色的，如果只是需要單色的話點選Albedo右邊如圖3- 10就可以選擇需要的顏色，但是我希望可以加上貼圖材質，就點選Albedo左邊然後選取貼圖如圖3- 11即可。

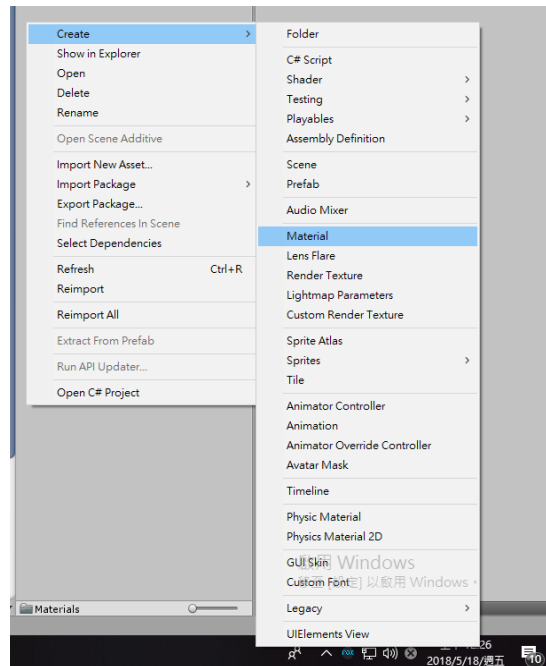


圖3- 9創建材質球

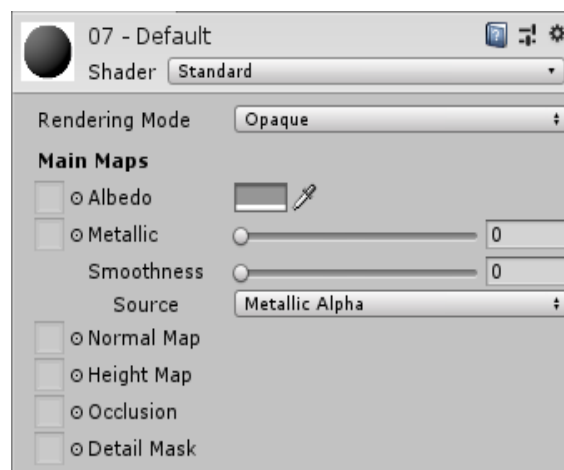


圖3- 10空白材質球

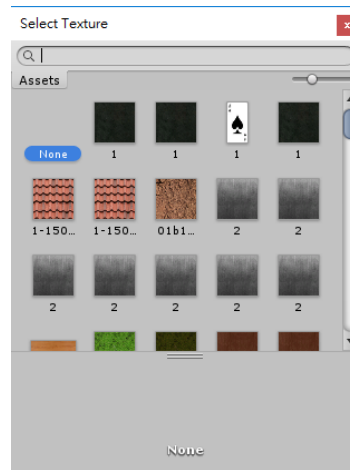


圖3- 11匯入貼圖材質

3-1-6 GameObject

GameObject就是Unity當中的物件，GameObject可以是任何的物件，例如：3Ds Max模型、內建產生的方塊、UI Button等等，在我們的專題當中用了許多的GameObject，在程式當中宣告GameObject去控制物件也是非常方便的，可以讓我們直接用程式控制許多的事情，在專題的陀螺與象棋兩個遊戲當中就用了程式去控制。在設計界面當中Creat Empty創立一個空物件，這個空物件可以用在許多的不同用途上，如圖3- 12創立後更改名稱為GameCtrl並且加入程式用來控制我們整個小遊戲，

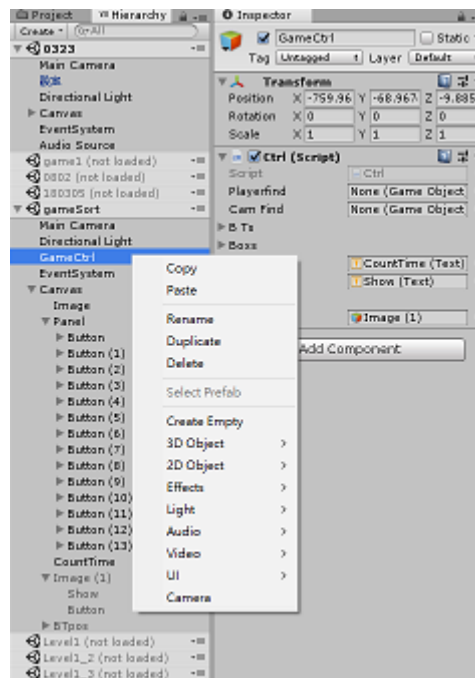


圖3- 12 Create Empty

我們首先在程式碼宣告物件名稱如程式3- 8，GameObject可以單獨宣告也可以宣告成List，這兩種在專題當中都有使用到。宣告以後設計介面會如圖3- 13，此時就可以依造需求放入物件，如果是List的話就要先設定大小後再放入，圖3- 13當中我們就放入了Button、Image、Text及模型。

```
10 public GameObject Playerfind;  
11 public GameObject CamFind;  
12 public GameObject Iset;  
13 public GameObject[] BTs;  
14 public GameObject[] Boxs;  
15 public GameObject[] BTposall;  
16
```

程式3- 8 public GameObject

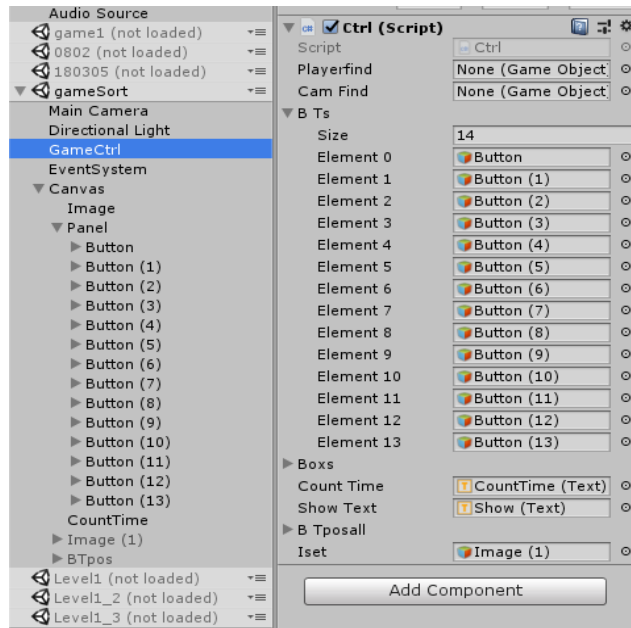


圖3- 13 GameObject[]介面

象棋總共有14個物件，首先是把物件放到棋盤上隨機的位置，並且符合我們的解析度大小。專題當中使用螢幕座標轉換成UI座標的方式，螢幕座標的範圍是以左下角(0,0)開始、右上角的座標為(Screen.width,Screen.height)如圖3- 14，由程式3- 9中第177行隨機給randomPos一個符合解析度大小的位置，首先變換第一個物件的位置，並且將隨機值加入到List當中，再次抓取一個隨機值，並且去比較randomPos是否存在於List當中，如果存在的話再次給另一隨機值，並且設定物件的位置與再次加入List當中。成果顯示如圖3- 15螢幕座標版本。

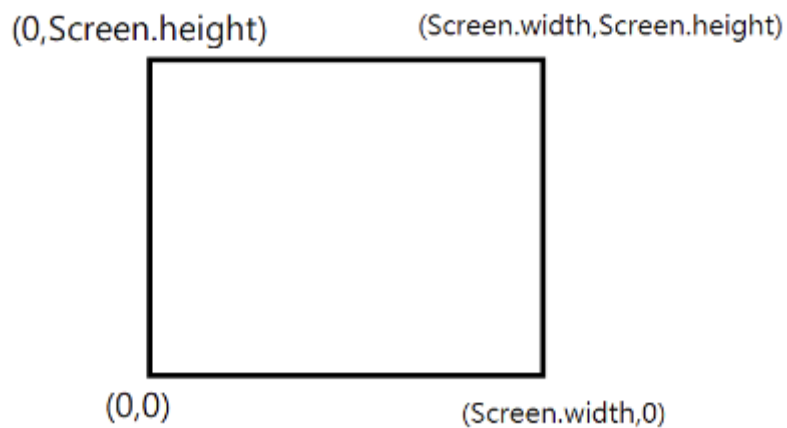


圖3- 14螢幕座標

```

174 public void BTstart()
175 {
176     Ctrlpoint = 0;
177     randomPos = new Vector3(Random.Range(0, Screen.width), Random.Range(0, Screen.height), 0);
178     BTs[0].transform.position = randomPos;
179     BTvector.Add(randomPos);
180     for (int i = 1; i <= 13; i++)
181     {
182         randomPos = new Vector3(Random.Range(0, Screen.width), Random.Range(0, Screen.height), 0);
183         while (BTvector.Contains(randomPos))
184         {
185             randomPos = new Vector3(Random.Range(0, Screen.width), Random.Range(0, Screen.height), 0);
186         }
187         BTs[i].transform.position = randomPos;
188         BTvector.Add(randomPos);
189     }

```

程式3- 9 BTStart

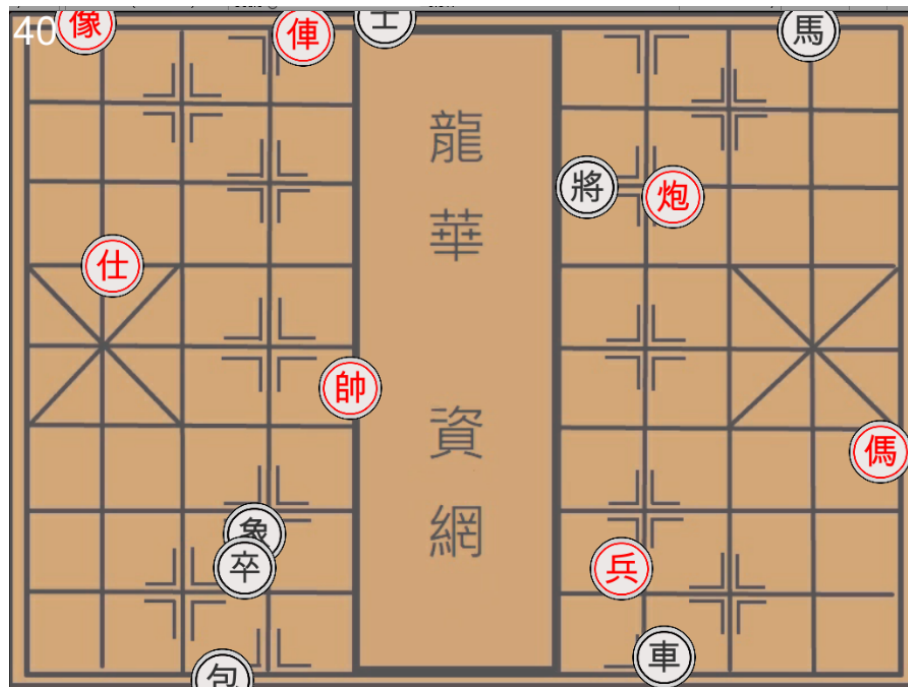


圖3- 15螢幕座標版本

由圖3- 15看到象棋有機會重疊與超出一點解析度範圍，象棋也不像現實當中在點或是格子當中，而我們想更符合現實與讓介面更好，使用了另一種方法解決，起先將點上都預設如圖3- 16，再將我們所預設好的按鈕位置隱藏並且更改我們的程式碼。

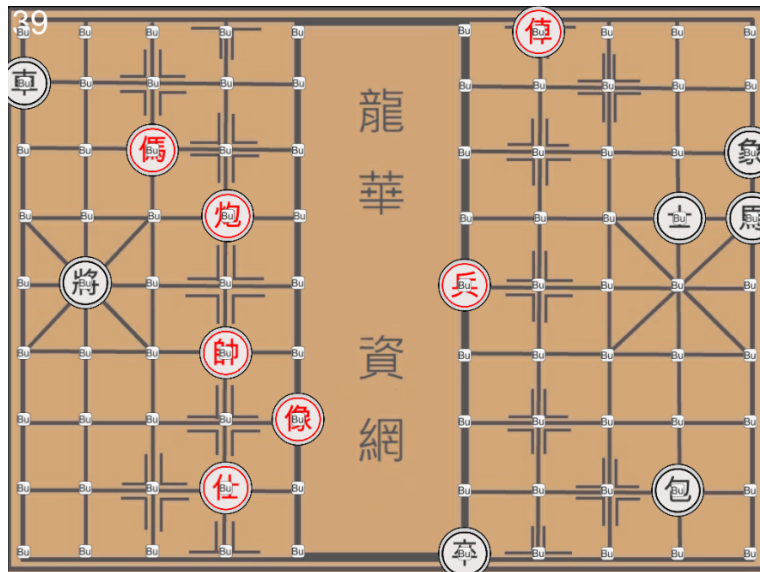


圖3- 16預設位置

由程式3- 10可以看到這幾乎是與前版本相同的程式碼，不過我們將第177行的randomPos由螢幕座標更改成我們預設物件的座標位置，如此一來象棋就不會重疊也不會發生超出解析度的問題。

```

174     public void BTstart()
175     {
176         Ctrlpoint = 0;
177         randomPos = BTposall[Random.Range(0, 89)].gameObject.transform.position;
178         BTs[0].transform.position = randomPos;
179         BTvector.Add(randomPos);
180         for (int i = 1; i <= 13; i++)
181         {
182             randomPos = BTposall[Random.Range(0, 89)].gameObject.transform.position;
183             while (BTvector.Contains(randomPos))
184             {
185                 randomPos = BTposall[Random.Range(0, 89)].gameObject.transform.position;
186             }
187
188             BTs[i].transform.position = randomPos;
189             BTvector.Add(randomPos);
190         }
191     }

```

程式3- 10 預設物件位置

陀螺小遊戲當中使用相同的方式去執行隨機不重複，但在2D與3D遊戲的差別上，使用不同的手法呈現出來。遊戲當中設置10*10的紅磚地板與陀螺，先顯示紅磚地板與把陀螺隱藏。由程式3- 11陀螺小遊戲的第92行看到先將一個紅磚地板顯示消失，在進行隨機不重複，但我們希望增加遊戲的困難度，不要讓陀螺過於的相近，第98行使用到了Vector3.Distance這是Unity的API去取兩個物件的位置並且算出兩個物件的距離，第109行取得地板再去做陀螺生成，才不會出現陀螺懸空的狀況，再把陀螺顯示出來，第116行給NumberPos一個在此次顯示的陀螺上方高2.5的三維位置，第119行照順序生成我們預設好的數字物件。而最後成果就會顯示如圖3- 17。


```

91     int randomInt = UnityEngine.Random.Range(minRan, maxRan);
92     crBoom[randomInt].SetActive(false);
93     setRandom.Add(randomInt);
94     for (int i = 0; i < 3; i++)
95     {
96         randomInt = UnityEngine.Random.Range(minRan, maxRan);
97         getRandom = setRandom[i];
98         dis = Vector3.Distance(crBoom[randomInt].transform.position, crBoom[getRandom].transform.position);
99         while (setRandom.Contains(randomInt) && dis <= 6)
100         {
101             randomInt = UnityEngine.Random.Range(minRan, maxRan);
102         }
103         crBoom[randomInt].SetActive(false);
104         setRandom.Add(randomInt);
105     }
106
107     for (int i = 0; i < 3; i++)
108     {
109         BonusRandom = setRandom[setRandom.Count - 1];
110         dis = Vector3.Distance(crBonus[randomInt].transform.position, crBonus[BonusRandom].transform.position);
111         while (setRandom.Contains(randomInt) && dis <= 6)
112         {
113             randomInt = UnityEngine.Random.Range(minRan, maxRan);
114         }
115         crBonus[randomInt].SetActive(true);
116         NumberPos = new Vector3(crBonus[randomInt].transform.position.x,
117                                 Convert.ToSingle(crBonus[randomInt].transform.position.y + 2.5f),
118                                 crBonus[randomInt].transform.position.z);
119         Instantiate(NumberText[i], NumberPos, Quaternion.identity);
120         setRandom.Add(randomInt);

```

程式3- 11 陀螺小遊戲

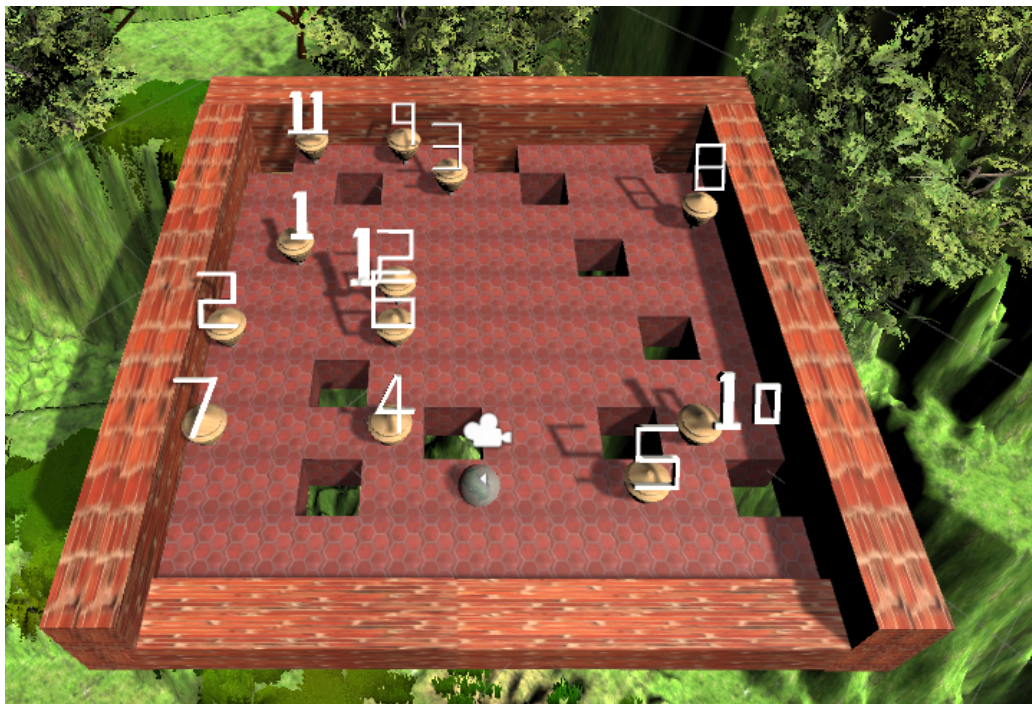


圖3- 17陀螺顯示

3-1-7 GUI

圖形使用者介面 (Graphical User Interface)

為大眾所習慣的介面，通常使用簡單的文字加上顏色圖形，放在視窗的位置上擺設，讓畫面更美觀且更吸引人。

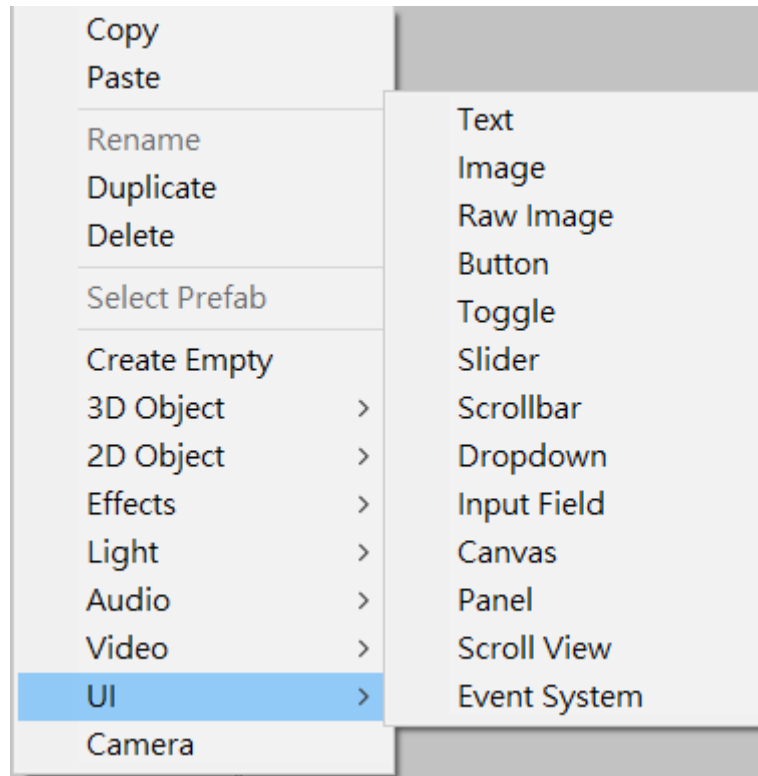
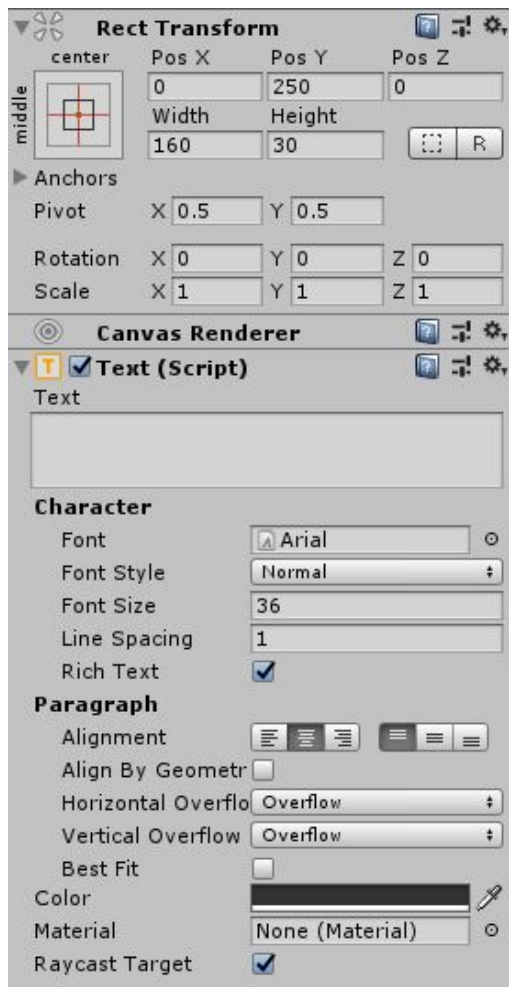


圖3- 18 GUI

如圖3- 18 所示，我們常用得UI可分為圖中各細項，其中此專案有用到的為Text、Image、Button、Toggle、Slider及Canvas。



XYZ設定此物件在畫面的位置

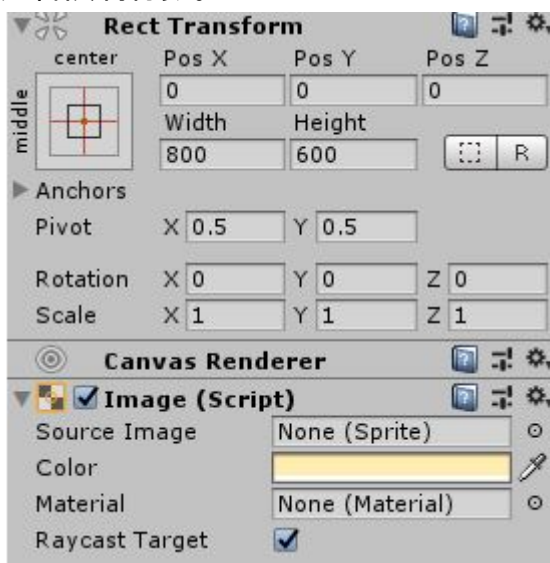
2D的UI則只會用到XY軸

Text(Script)

此區為文字編輯區

圖3- 19 Text(Inspector)

Text 文字 如圖3- 19可以搭配Image變成類似公告的造型，或是遊戲介紹的文字敘述；其中Horizontal Overflow 代表X軸是否限制字數， Vertical Overflow則是Y軸是否限制行數。



Rect Tranform

位置及畫面大小設置

Image(Script)

主要配置

圖3- 20 Image(Inspector)

Image 圖片如圖3- 20 在Source Image 可以丟自己設置好的圖片或材質，如果

沒有圖片能放, 也可以單純改Color給它顏色。
設置圖片成UI圖:



圖3- 21 圖片丟入Unity Asset資料夾內

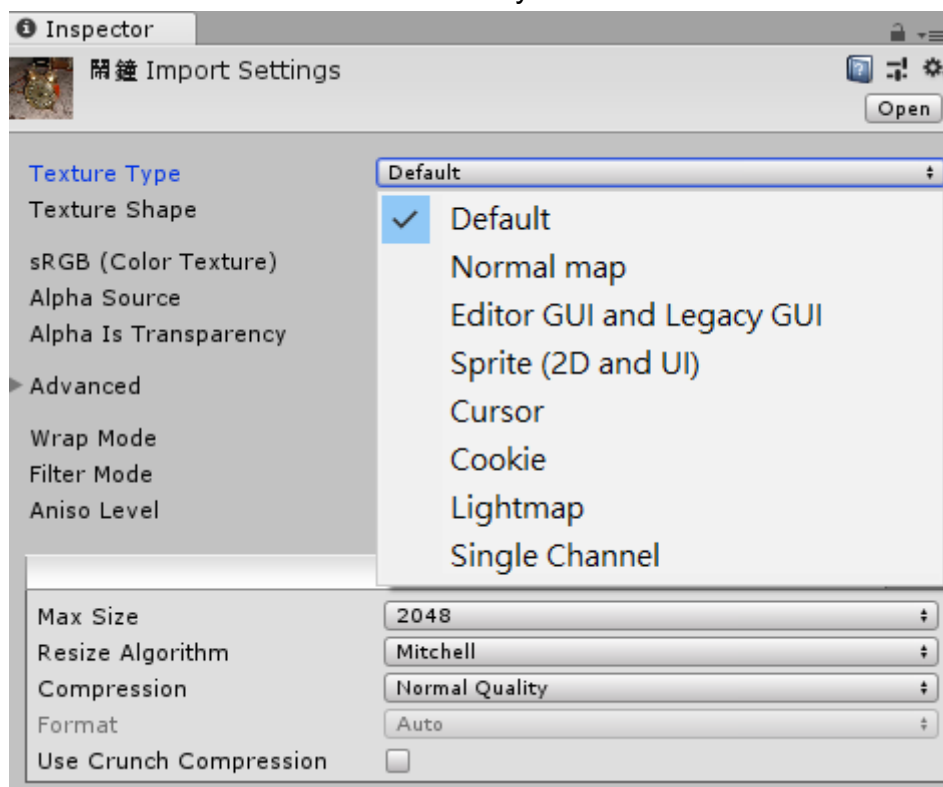


圖3- 22選取圖片後的(Inspector)

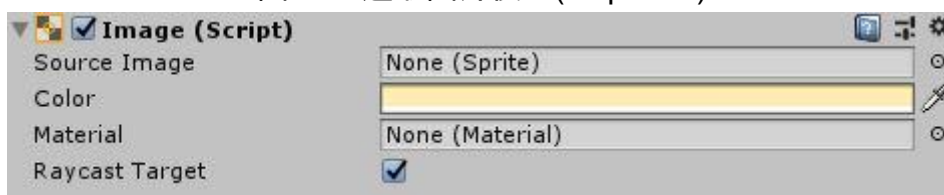
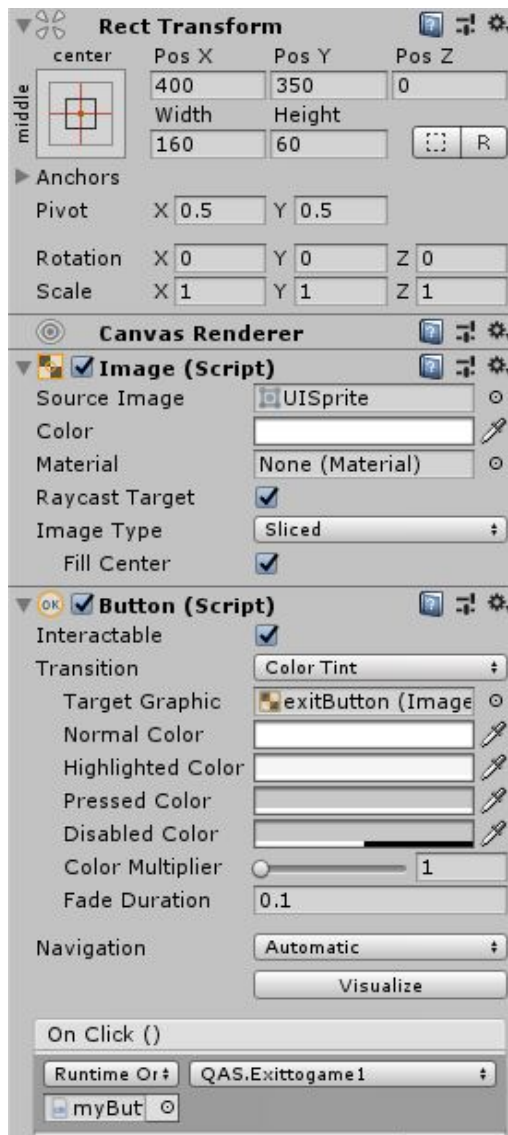


圖3- 23 選取圖片後 (Inspector Image(Script))

原先放入的圖面都會像是圖3- 21 右下角那樣右側沒有箭頭, 我們只需將圖3- 22 Inspector->Texture Type -> Default 改成 Sprite(2D and UI) 就可以拉到圖3-

23Image->Source Image 裡面套用。



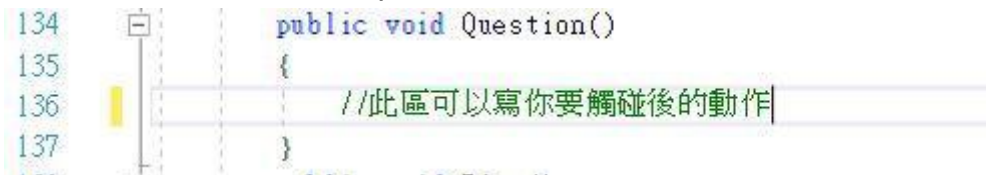
按鈕可以當成能按壓的圖片

此區為Button主要設計頁面

圖3- 24 Button(Inspector)

Button 按鈕 是我們最常用的物件，如圖3- 24中，最常用的功能則是On Click(), 能夠撈程式碼中寫好的Function，然後輕鬆關聯。

Button設置On Click 與 C# Script關聯



程式3- 12 宣告公共函式寫法

如程式3- 12在程式內宣告一個公共變數的void 宣告名() {}，之後將觸碰動作寫在大括弧裡面。

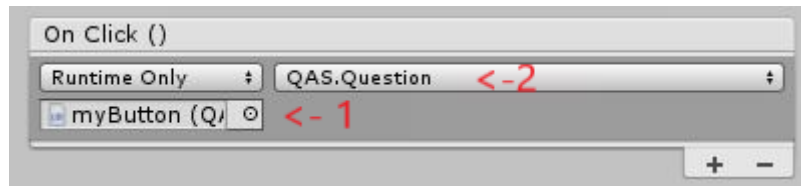


圖3- 25 Button的Inspector中常用功能

在Button->Inspector 內可以找到如圖3- 25， 首先將你寫好的script 或是有放script的元件拉到圖中第一步， 之後調整圖中第二步， 通常點開會有一行是script的名稱， 點開後再選寫好的Function。

Toggle 切換方塊 只有按下打勾 及不按空白， 此功能用在背景音樂是否靜音， 如程式3- 13第43-46行， 當按下toggle之後， 音樂音量改為0。

```

42 myMusic.volume = voice.value; //滑條與音量大小
43 if (toggle.isOn == true)
44 {
45     myMusic.volume = 0;
46 }

```

程式3- 13 Toggle及Slider連動程式碼

Slider 滑塊條 數值介於0至1， 此處用在背景音樂的大小聲， 如程式3- 13第42行的程式碼隨著slider給予音樂音量的大小。

Canvas 帆布 布局用的主元件， 主要是用來收納所有UI物件的(子元件)。其中有遇到畫面比例會因為解析度不同而有畫面上的誤差， 如圖3- 26， 解決方法則是選取Canvas → 看到Inspector → Canvas Scaler(Script) → UI Scale Mode → 改成Scale With Screen Size， 之後將Reference Resolution X,Y比例打成4:3， 或是如圖的解析度1024*768

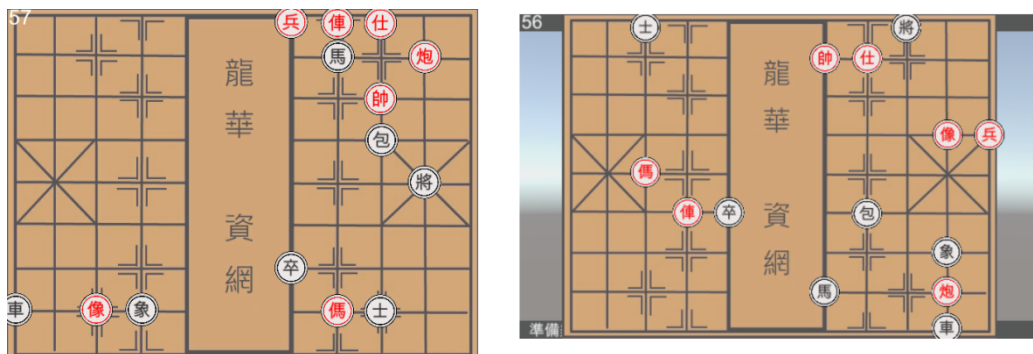


圖3- 26解析度差別

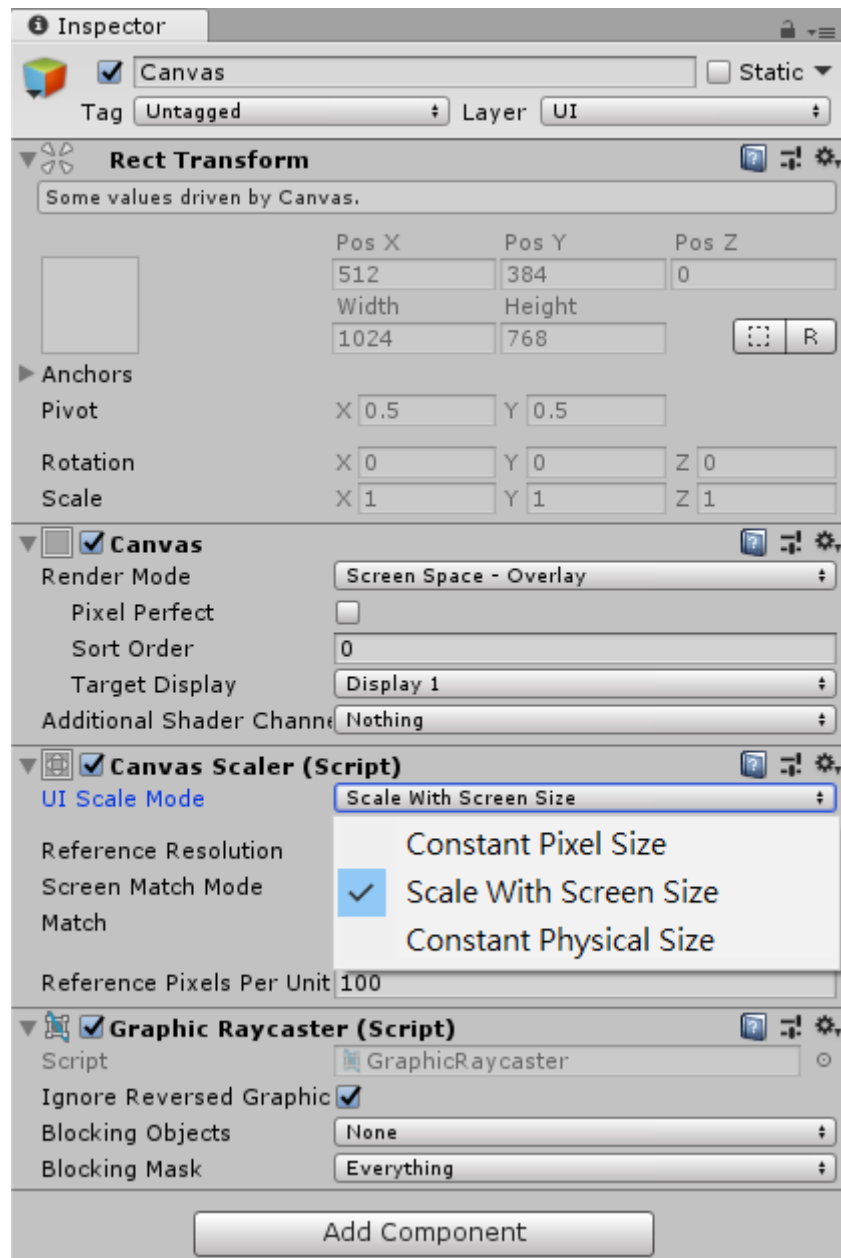


圖3- 27 Canvas(Inspector)

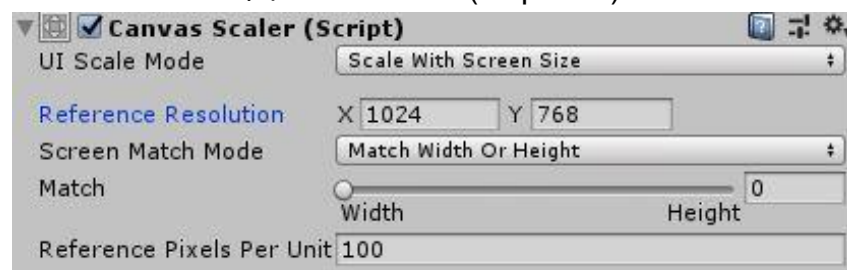


圖3- 28 Canvas (Inspector中的[Canvas Scaler(Script)])

問答遊戲程式內容，分為亂數選題(存陣列判斷重複)、宣告題目、判斷答案。
亂數選題：


```

114 RandomNumber = Random.Range(1, 24); //起始數值到最後數值-1
115 checkeq[count] = RandomNumber;
116 if (count > 0)
117 {
118     for (int i = 0; i < count; i++)
119     {
120         if (RandomNumber == checkeq[i])
121         {
122             Randomcheck();
123         }
124         else
125         {
126             /*for (int j = 0; j < count; j++)
127             {
128                 scoretext.text += "\n"+checkeq[j].ToString() + ", ";
129             }*/
130         }
131     }
132 }
133

```

程式3- 14 判斷執行次數

如程式3- 14中陣列由0開始存取，所以迴圈會從陣列checkeq[0]開始到總共存了幾幾次。

if(count) 是因為如果還沒存取過，會直接做導致錯誤。

宣告題目：

```

141 switch (RandomNumber)
142 {
143     case 1:
144         Text_text.text = "請問圖片中的物品名稱叫什麼?";
145         mybutton.image.overrideSprite = Camera;
146         RButton_text.text = "投影機";
147         LButton_text.text = "相機";
148         break;
149     case 2:
150         Text_text.text = "請問圖片中的房子叫什麼?";
151         mybutton.image.overrideSprite = threejoin;
152         RButton_text.text = "三合院";
153         LButton_text.text = "高樓大廈";
154         break;
155     case 3:
156         Text_text.text = "請問圖片中的道具用在什麼場合?";
157         mybutton.image.overrideSprite = boodiesee;
158         RButton_text.text = "歌仔戲";
159         LButton_text.text = "布袋戲";
160         break;
161     case 4:
162         Text_text.text = "請問圖片中的建築偏向哪國風格?";
163         mybutton.image.overrideSprite = jph;
164         RButton_text.text = "日式";
165         LButton_text.text = "美式";
166         break;
167     case 5:
168         Text_text.text = "請問圖片中的表演為何?";
169         mybutton.image.overrideSprite = gerzaisee;
170         RButton_text.text = "布袋戲";
171         LButton_text.text = "歌仔戲";
172         break;
173     case 6:
174         //

```

程式3- 15 題目宣告(Switch case)

如程式3- 15用switch case 個別宣告的原因是，如果都用亂數給予答案選項，就會造成答案與另一個選項會天差地遠。

判斷答案:

在按下右邊按鈕或是左邊按鈕時都會給SelectTrueAns變數一個數值，1為正確，0為錯誤。

原先是給予左邊按鈕及右邊按鈕個別用switch case 給予個別答案，但是後來用規律的方式排答案，因此縮短許多程式碼。

亂數題目為奇數時，左按鈕會使變數SelectTrueAns=1，反之右按鈕變數SelectTrueAns=0。

偶數題目則反之。

最後則是按下中間的送出，則可以判斷變數SelectTrueAns的數值，順便幫count計數；當count達到10時，代表題目做了10題，便會將題目設為唯獨，並跳出返回主遊戲的按鈕。

骰子遊戲程式設計:

程式寫法主要是分兩大類別，第一大類別是對子類別，另一大類別是單支類別。

對子類別:一對、兩對、三條、葫蘆、鐵支、大滿貫。

單支類別:單支、順子

對子類別:

想法是有一個對子計數多加一，一對計數1，兩對計數2，三對計數3，葫蘆計數4，鐵支計數6，大滿貫計數10。

單子類別:

想法是先存兩個陣列為000000及 123456，如果骰子值有與陣列相同，此想對區塊0改為1，最後總和 000000 的值，當值達到5，代表此區就是單支或順子，最後在判斷，如果最前面或是最後面為0，代表為順子，反之為單支。

其中遇到骰子站立問題，是以轉動解決如程式3- 16。

```
278 public void Yroll()
279 {
280     ran = Random.Range(3.1f, 6.9f);
281     rb.AddTorque(0, ran, 0);
282 }
```

程式3- 16 給予骰子轉動的力

3-1-8 Light

Unity新專案當中都會直接預設一個Main Camera與Directional Light, Directional Light是一個全場景的燈光, 在3D遊戲的狀況下模型就會產生光影, 不過當有些材質放入Unity的時候搭上燈光, 產生光影之後不會呈現我們當初看到的樣子如圖3- 29。

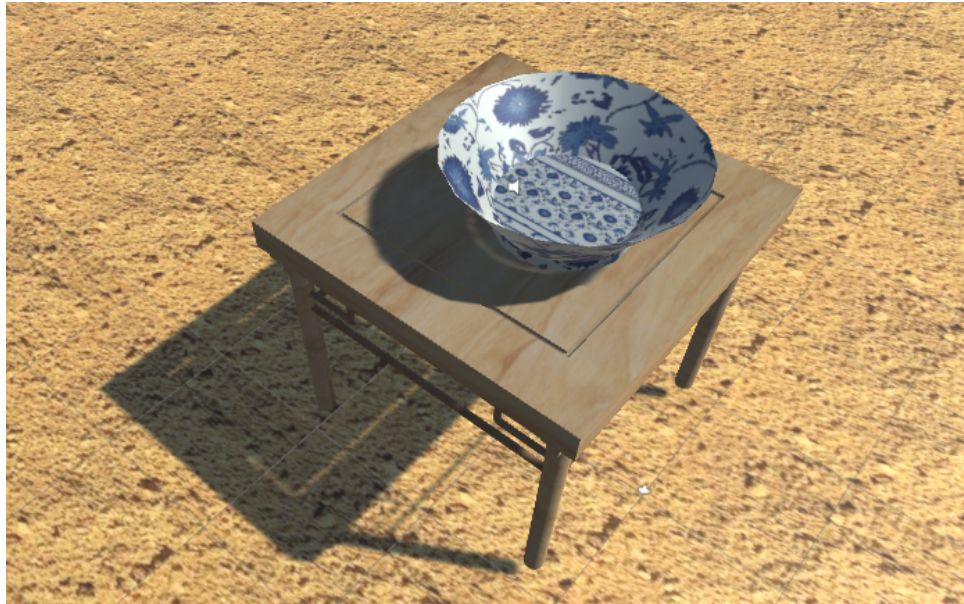


圖3- 29 Unity光影

當產生光影的時候, 瓷碗所貼上類似青花瓷效果的材質並沒有那麼的鮮豔, 而是因為有陰影進而偏黑了。此時我們就可以加入Unity當中的Light, 也提供了不同的燈光效果如圖3- 30。我們使用了兩個point light如圖3- 31, 將瓷碗的材質打上光, 進而顯示成原本鮮豔的材質效果, 另一個放在桌子下方, 讓光影效果更佳的明顯, 並且使桌子讓使用者能更清楚地觀看到效果。最後光影的呈現就會如圖3- 32增加光源呈現效果。

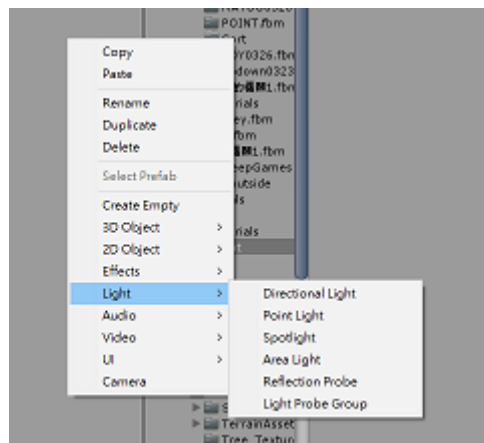


圖3- 30 Light選擇

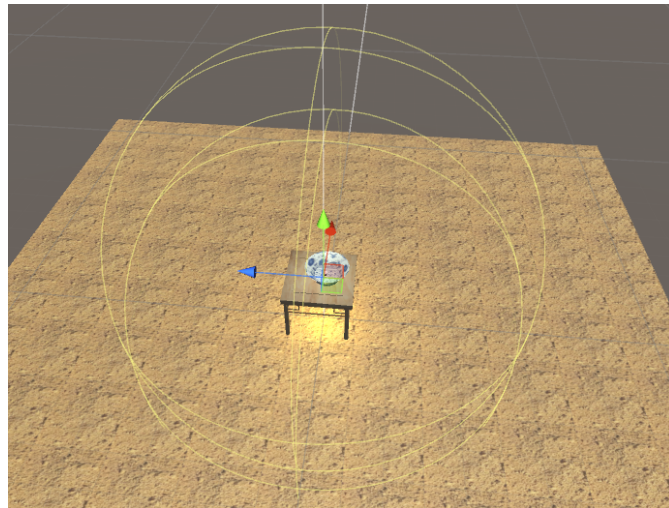


圖3- 31 Point light使用



圖3- 32增加光源呈現效果

3-1-9 Scenes

Unity一開始的場景拉入的物件會在遊戲開始的時候直接產生出來，在進行場景轉換的時候，場景、物件、程式都會一同刪除並且轉換到新的場景，程式可以使物件在轉換場景的時候不被刪除如程式3- 17，專題當中Player與Main Camera兩個物件都有使用不被刪除，讓兩者進到小遊戲中可以繼續使用並且一直存在著，我們在第一關的場景預設這兩個物件，所以在兩者都不會被刪除情況下從小遊戲回來第一關時，我們要先刪除物件，才不會使第一關出現了兩個Player的狀況。

```
68 | | | | GameObject.DontDestroyOnLoad(gameObject);
```

程式3- 17 過場不刪除

由程式3- 18看到，我們先宣告PlayerFind與CamFind，但是在設計介面當中並沒有放入預設物件，因此我們從程式裡面去抓取如程式3- 18第47行與第48行，先將Player與Camera物件的Tag設定完成，在程式當中就能利用Tag去找到物件。第54行就是執行過場的動作，會到所設定的取名畫面。第55行與第56行Destroy刪除我們Tag的物件。如此一來把物件刪除後回去第一關，就不會發生科幻電影當中從未來回來過去的人互相碰面的狀況。

```

45 void Start()
46 {
47     Playerfind = GameObject.FindGameObjectWithTag("Player");
48     CamFind = GameObject.FindGameObjectWithTag("MainCamera");
49     Hard();
50     RankStart();
51 }
52 public void back()
53 {
54     Application.LoadLevel("game1");
55     Destroy(Playerfind);
56     Destroy(CamFind);
57     PlayerCtrl.ball = 2;
58 }

```

程式3- 18 Tag與刪除

在Unity不知道目前進行到什麼場景，我們也可以直接用程式去抓取目前的場景，這樣可以方便我們直接去轉換場景。在陀螺小遊戲當中有分三個難度，所以每個難度的控制程式並不會知道玩家進行到哪關，但是我們又要執行碰到界外的時候從新開始，所以就使用到了如程式3- 19抓取場景的第250與251行給一個字串此場景的名稱，再進行讀取場景。

```
247     if (player.gameObject.CompareTag("OutSide"))
248     {
249         // Application.LoadLevel("Level1");
250         string nowScene = Application.loadedLevelName;
251         Application.LoadLevel(nowScene);
252     }
```

程式3- 19抓取場景

Unity可以創建許多的專案與Scenes，並且可以將這些場景匯出與匯入。我們就將此方法應用在我們專題的分工上面，這樣才不會發生需要等待其他組員先完成某些項目才能繼續之後的工作，美工組也能先建造場景與模型再傳給程

式組套用程式。如圖3- 33看到選取Export Package就是進行將場景匯出的動作，選取之後可以再決定所要匯出的物件。而匯入的功能就要點選Import Package -> Custom Package，如此一來只要做好每個小遊戲後再進行整合專案就可以完成

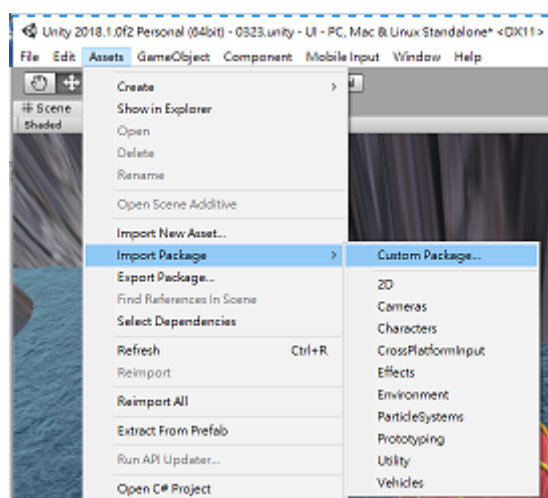


圖3- 33 Scenes匯出匯入



圖3- 35 數字視角相反

```
8      transform.Rotate(new Vector3(-45, 225, 0));
```

程式3- 21 轉變物件座標

為了讓玩家能回味以前玩童玩時候的滋味，我們也將陀螺製作成了能轉動的，玩家遊玩的時候就能感受到猶如自身在打陀螺的感受。由程式3- 22可以看到，我們在Update放入讓陀螺每秒都會變更Z軸的方向轉動，而Z軸設定成正數的話物件是以由右向左轉動的方式呈現。

```
13      void Update () {  
14          transform.Rotate(new Vector3(0, 0, 90) * Time.deltaTime * 2);  
15      }
```

程式3- 22 陀螺轉動

3-1-11 PlayerPrefs

進入第三關會先執行RankStart(), 在我們的排行榜當中使用Unity的PlayerPrefs, PlayerPrefs可以將遊戲當中的資訊儲存到本機當中用來做遊戲存檔讀取的功能, 在這我們拿來應用在本機排行榜。由程式3- 23所看到, 第211行開始if(!PlayerPrefs.HasKey("RankBall"))如果沒有RankBall這筆資料, 將會產生一個RankBall, 並且在設定一個Float的型態名稱設置為RankBallA都儲存著初始值0, 而如果真的話就進行取得動作就可以, 並且將取得的值存在List當中, 第233行把所有的設定儲存好。

```
209 void RankStart()
210 {
211     if (!PlayerPrefs.HasKey("RankBall"))
212     {
213         PlayerPrefs.SetString("RankBall", "");
214         for (int i = 0; i < 3; i++)
215         {
216             rankstring += "[" + (i + 1).ToString() + "] " + fRank[i].ToString("F1") + "\n";
217             showText.text = rankstring;
218             PlayerPrefs.SetFloat(i + "RankBallA", 0);
219         }
220         print("no key");
221     }
222     if (PlayerPrefs.HasKey("RankBall"))
223     {
224         rankstring = "排行榜:\n";
225         for (int i = 0; i < 3; i++)
226         {
227             fuck = PlayerPrefs.GetFloat(i + "RankBallA", setff);
228             fRank[i] = fuck;
229             rankstring += "[" + (i + 1).ToString() + "] " + fRank[i].ToString("F1") + "\n";
230             showText.text = rankstring;
231         }
232         print("Has key");
233         PlayerPrefs.Save();

```

程式3- 23 起始存取排行榜

在初始的值都設定好以後, 到破關的時候就會產生分數, 由程式3- 24第169行當我們判斷兩個List的數量、值、順序都相等的時候為true, 並且就會顯示排行榜計算玩家的分數, 當獲取分數以後使用氣泡排序法去把玩家的分數與先前的排行榜做排序, 第188行開始將此次的排行本存回我們的RankBallA, 在將排行榜加入一個String當中, 第195行開始判斷此次的分數是否有進入排行榜, 有的話就會找此次分數的位置, 並將分數換成紅色以利玩家知道自己此次遊玩的結果如圖3- 36。

```

169         if (NumberA.SequenceEqual(NumberB) && Ctrlpoint == 0)
170         {
171             Iset.gameObject.SetActive(true);
172             Ctrlpoint += 1;
173             countRank = (1 / Level1Time) * 500;
174             fRank[3] = countRank;
175             for (int i = 0; i < fRank.Length; i++)
176             {
177                 for (int j = 1; j < fRank.Length; j++)
178                 {
179                     if (fRank[j - 1] < fRank[j])
180                     {
181                         temprank = fRank[j - 1];
182                         fRank[j - 1] = fRank[j];
183                         fRank[j] = temprank;
184                     }
185                 }
186             }
187             rankstring = "排行榜:\n";
188             for (int i = 0; i < 3; i++)
189             {
190                 setff = fRank[i];
191                 PlayerPrefs.SetFloat(i + "RankBallA", setff);
192                 rankstring += "[" + (i + 1).ToString() + "]" + fRank[i].ToString("F1") + "\n";
193                 showText.text = rankstring;
194             }
195             if (rankstring.Contains(countRank.ToString("F1")))
196             {
197                 firstkey = rankstring.IndexOf(countRank.ToString("F1"));
198                 setrank = rankstring.Insert(firstkey - 5, "<color=red>");
199                 setrank = setrank.Insert(firstkey + 15, "</color>");
200                 showText.text = setrank;
201             }
202             PlayerPrefs.Save();
203         }

```

程式3- 24 計算排行榜

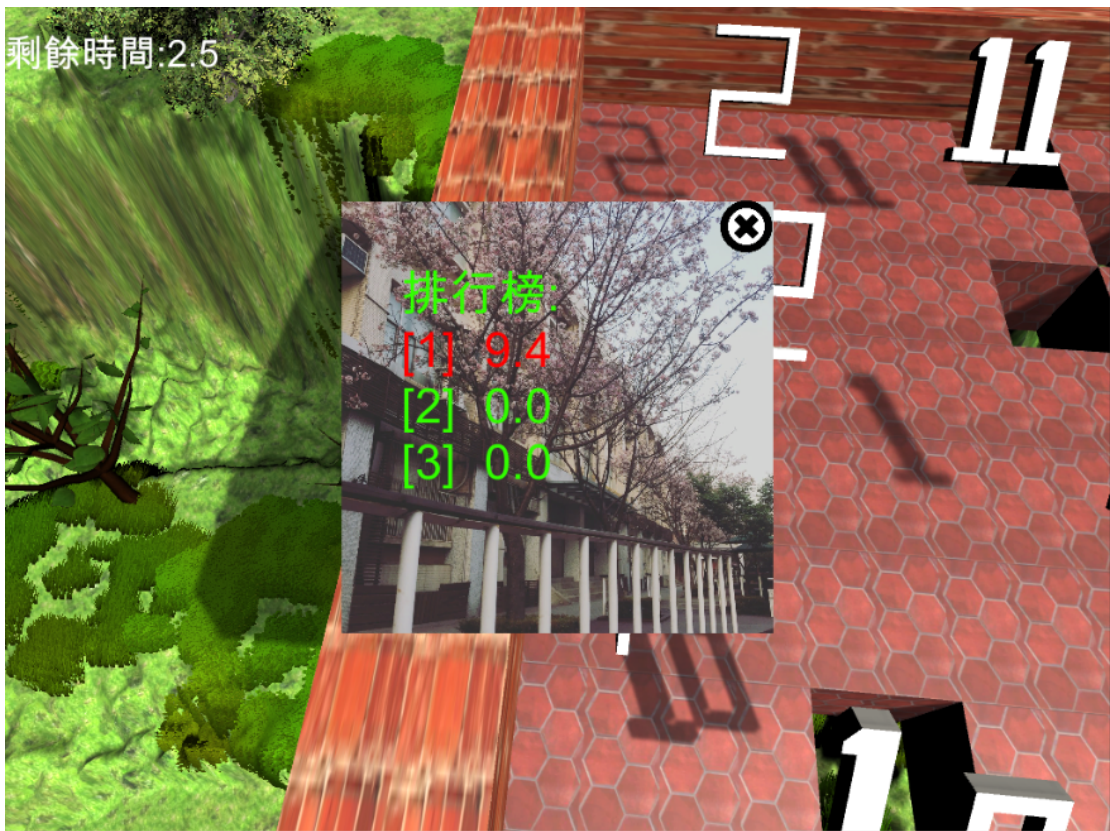


圖3- 36 排行榜顯示

3-1-12 Terrain

遊戲主題主要為闖關，需要大範圍的地圖讓玩家移動闖關，為了讓地圖看起來更逼真，所以使用Unity內建的地形來創建遊戲所需要的地圖。

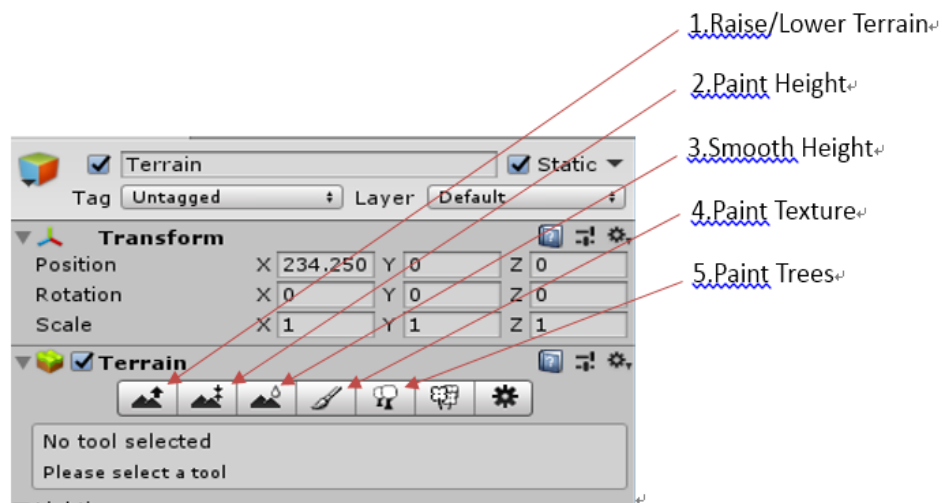


圖3- 37 Terrain介面

圖3.37-1 Raise/Lower Terrain

這項功能是在地圖上控制山的隆起跟凹陷，自行調整繪製的筆刷樣式、筆刷大小與筆刷繪製的強度。

圖3.37-2 Paint Height

這項功能大部分與Raise/Lower Terrain相同，但是多了一個可以限制高度的功能。

圖3.37-3 Smooth Height

這項功能是可以將Raise/Lower Terrain與Paint Height繪製出來的地形給平滑化，讓地圖不會看起來那麼菱菱角角的。

圖3.37-4 Paint Texture

這項功能可以幫地圖上材質，可以將圖片貼在地型上面，讓地形看起來更逼真。

圖3.37-5 Paint Trees

這項功能可以將做好的樹木以及草材質，貼到地圖上面。

因為遊戲內是由多個地形組成的，當初我們是由不同專案總合而成的，所以在整合的時候有發生一些問題，例如第一關跟第二關的前後左右不一樣如圖3-40，導致我們自己測試的時候方向感很容易錯亂，但是地形不能像物件一樣直接使用旋轉來改變方向，如果使用旋轉的話只會改變軸心而已如圖3- 38為旋轉前圖3- 39為旋轉後。

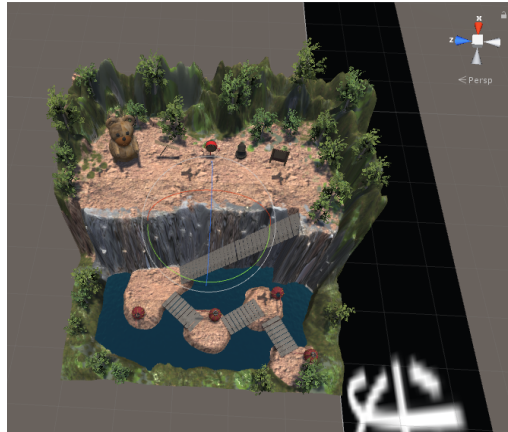


圖3- 38地形座標旋轉前

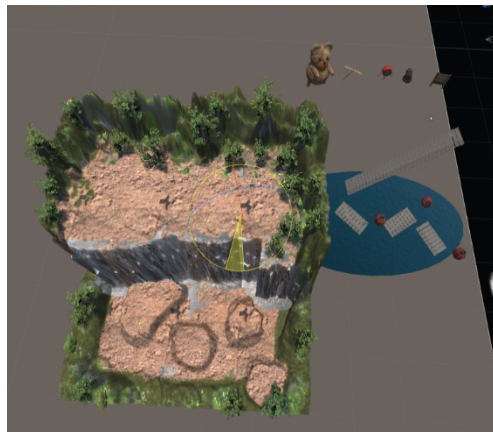


圖3- 39地形座標旋轉後

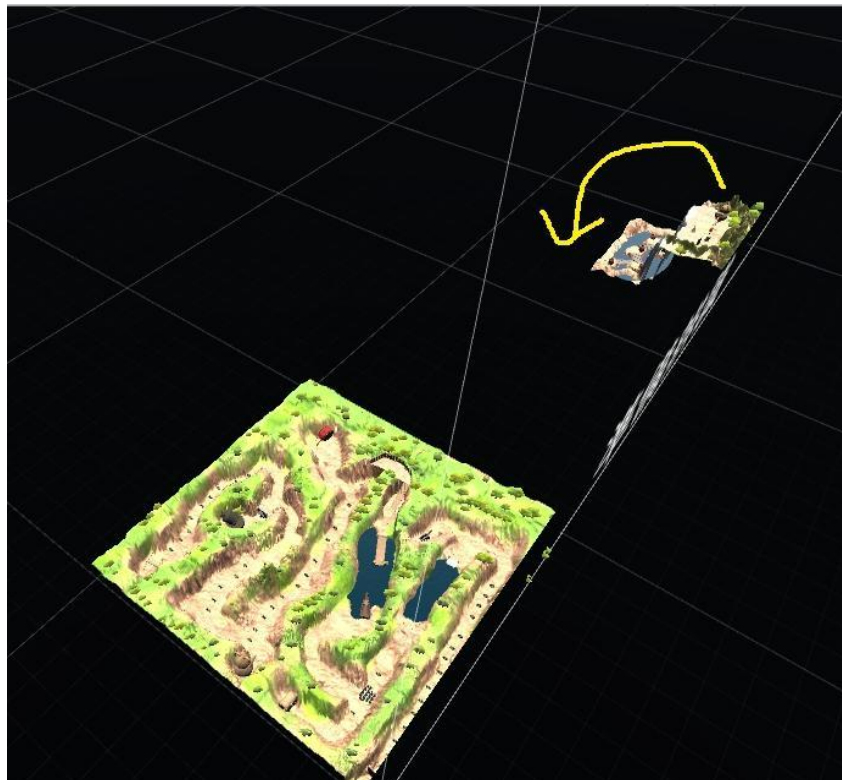


圖3- 40地圖應該旋轉的方向

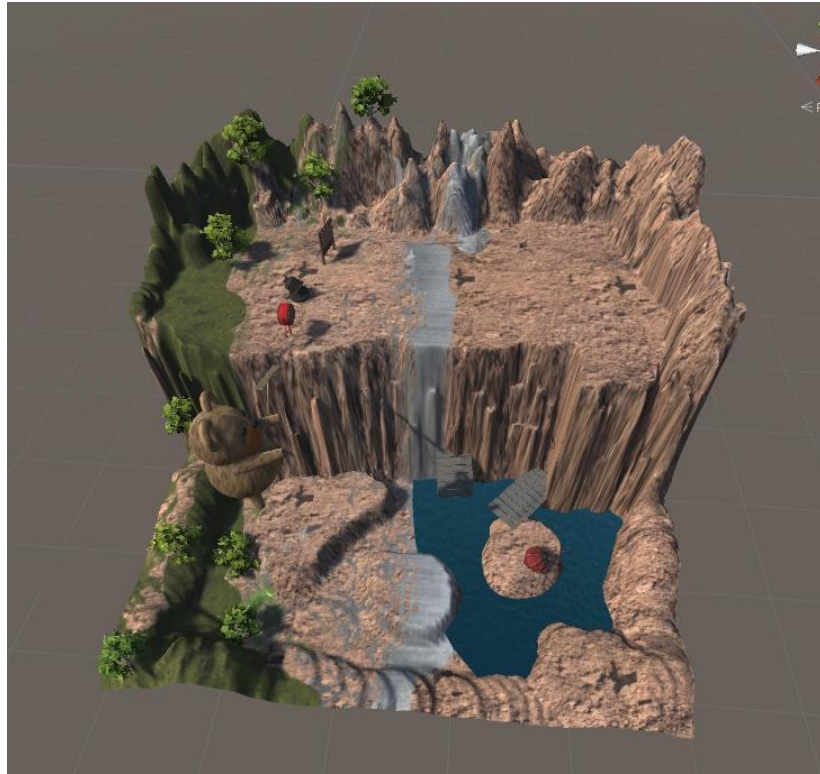


圖3- 41匯入高度圖之後旋轉的地形

這時候就要使用Height map來匯出高度圖3- 42, 因為高度圖是特殊格式, 所以必須使用Photoshop來進行轉向, Export之後轉向之後再Import進去如圖3- 43, 地形就會跟著轉向如圖3- 41, 只是地形旋轉原本加的材質不會旋轉所以必須重新上色。

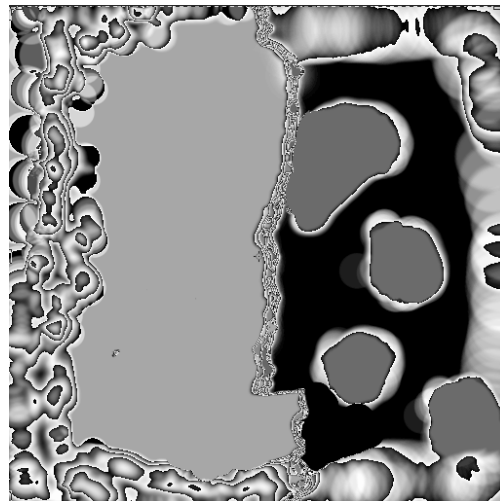


圖3- 42匯出的高度圖

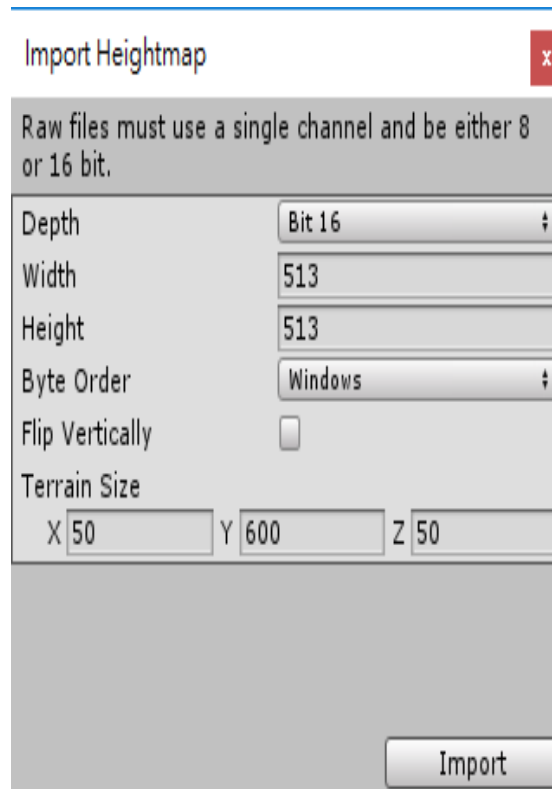
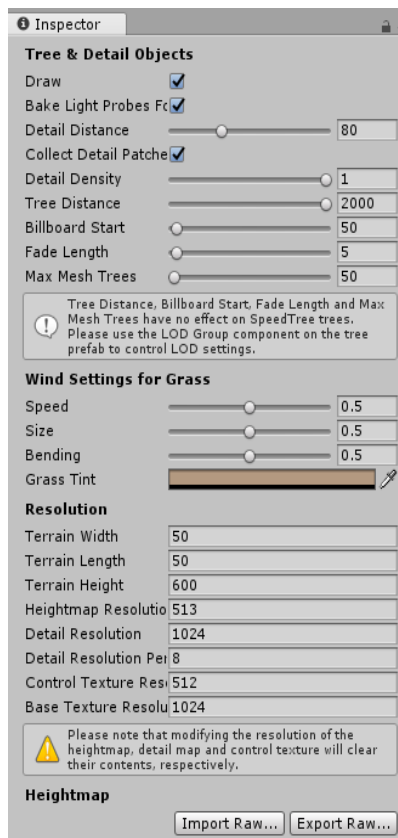


圖3- 43 Import Heightmap

3-2 3Ds Max

3Ds Max是一個具有強大功能的建模工具，可以建立許多專業的3D模型還有動畫，主要介面有4個不同視角的方位圖，可以讓使用者用不同的方位去拉取模型，在動畫上有許多方式來製作動畫，在設計家具、遊戲、材質上面都會用到，本次專題主要是設計3D遊戲，故使用3Ds Max來製作裡面的3D物件。

3-2-1 模型建設

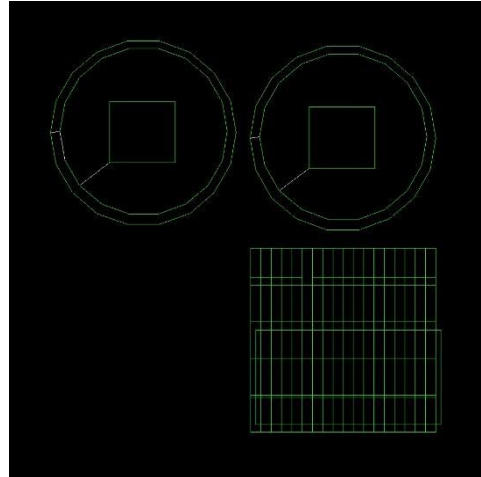


圖3- 44硬幣拆uv結果

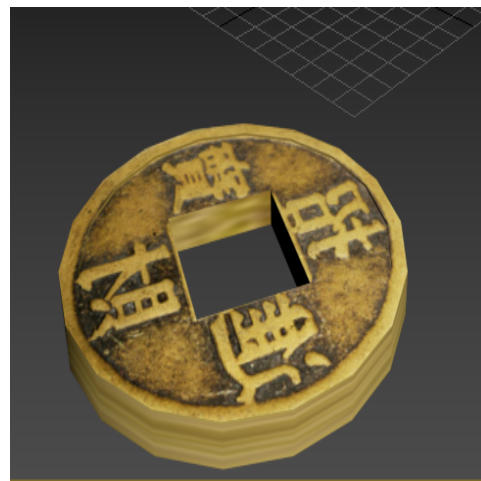


圖3- 45硬幣貼uv結果

為了配合遊戲內的各種物件及需求，使用3Ds Max來製作物件還有動畫，遊戲中有許多物件在觸發之後會開啟動畫，就必須要用到3Ds Max來錄製以及製作模型，使用3Ds Max製作完模型之後，必須要幫物件上各種材質，比較複雜以及多面的物件就必須要拆uv，來達到一個物件可以準確地把圖貼在各個面上面。

假設以第一關的硬幣來說，如果直接把材質拉進去物件的話，圖片上的字就會無法準確地出現在硬幣上面，所以把硬幣拆uv之後如圖3- 44再使用Photoshop來貼材質，完成在貼回去3Ds Max上，就可以達成目的了如圖3- 45。

3-2-2 可編輯多邊形



基於遊戲內所需要模型會有各個的形狀, 不可能只會用到內建的圓形、正方形、長方形等等, 所以如 REF

_Ref515153077 \h 圖3- 46這個功能主要是將原始不可編輯的物體轉換為可以編輯的狀態, 裡面子級別有許多功能, 方便做出不同的轉換以及變形, 例如:點、線、面以及多邊形。

圖3- 46可編輯多邊形

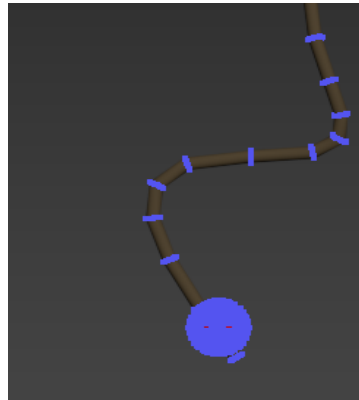


圖3- 47頂點

如圖3- 47頂點可以移動模型上面各個點來拉取圖形, 可以選取一個或者多個頂點, 在我需要改變模型形狀的時候這個功能是最常用到的, 他也可以生出多個等距的點來創造出等比例的面, 來讓我在做多邊形的時候更方便。

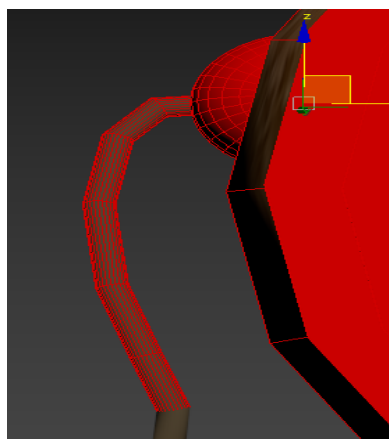


圖3- 48邊

如圖3- 48邊可以選擇一個或者多個邊, 並使用標準的方式移動他們, 這個功能我比較少用到, 但是在單獨移動一個線的時候也是蠻方便的。

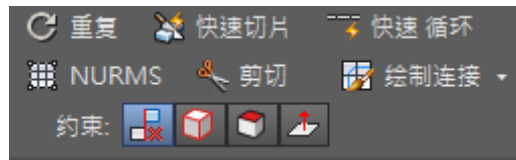


圖3- 49可編輯多邊形功能表

可編輯多邊形裡面有一個功能叫「快速切片」如圖3- 49, 他的用意是把多邊形自由地畫出分割線, 讓使用者可以切出想要的多邊形來進行編輯如圖3- 50尚未切片前的多邊形在使用快速切片後如圖3- 51為切片後的多邊行, 圖3- 52為多邊形圖。

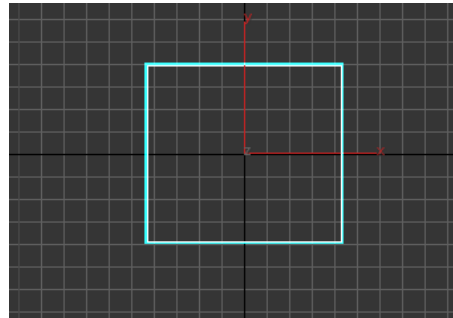


圖3- 50尚未切片前的多邊形

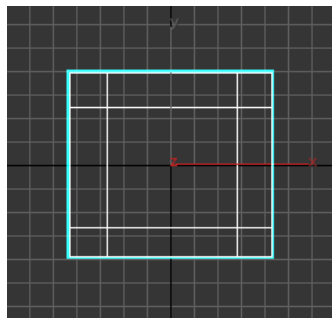


圖3- 51切片後的多邊形

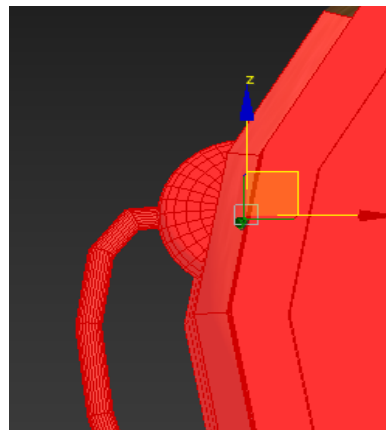


圖3- 52多邊形



圖3- 53多邊形功能表

多邊形可以選擇一個或者多的面，進行拉取跟移動，功能表裡面有一個功能叫做「擠出」如圖3- 53，我在做圖形的時候常常會使用到這個功能，像是硬幣周遭的突起，就可以選取圓形邊緣來做出擠出的動作，讓他選取的面都均勻長高如圖3- 54。

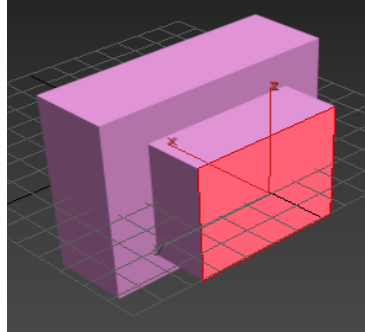


圖3- 54擠出後的多邊形

3-2-3 連接



圖3- 55連接列表

這個功能可以把多個多邊行組成的模型組成一個物件如 REF_Ref515153354 \h 圖3- 55, 當初會使用到這個功能是因為當我做好模型的時候, 需要匯入到 Unity裡面去, 在Unity拉開物件會顯示出多個物件如 REF_Ref515153438 \h 圖3- 56, 不但會讓人眼花撩亂, 也會讓寫程式的同學覺得不方便而且更佔資源, 一開始是採用「群組」的方式, 但是這個方式治標不治本, 雖然在3Ds Max裡面把物件匯集在一起了, 但是匯入到 Unity裡面一樣會顯示多個物件, 所以後來才找到這個方式。

圖3- 57是沒有連接物件使用幾個物件, 如果連接了就會顯示成一個物件如圖3- 58, 這樣會更節省資源, 也不會有多个物件的問題, 簡單的物件就還好, 可是像這種多個物件組成的模型, 有沒有連接就會差很多。

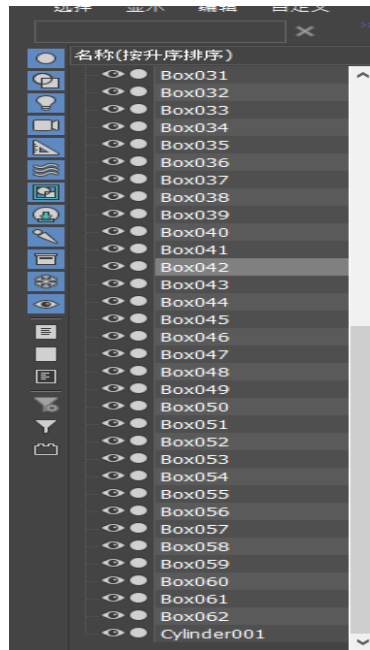


圖3- 56沒有連接物件的列表

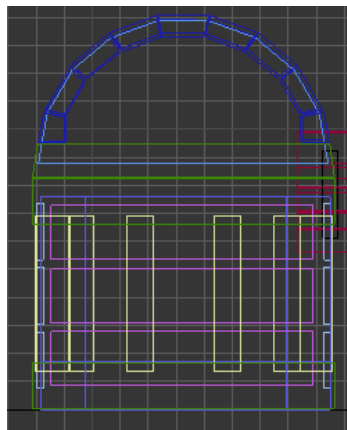


圖3- 57沒有連接的物件圖

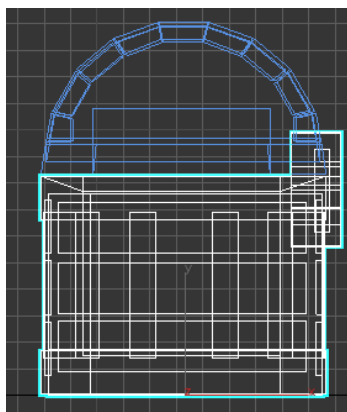


圖3- 58連接後的物件圖

3-2-4 動畫



圖3- 59起始關鍵點



圖3- 60關鍵點結尾



圖3- 61設置關鍵點

在遊戲中的物件會加入一些動畫，增加使用者體驗，最基本的動畫就是使用『設置關鍵點』來製作動畫如圖3- 61，通常都是使用旋轉或者平移來讓物體達到開啟的效果，例如：開門、箱子開啟、電梯上升...之類的。

預設值數是100，可以在0-100之間設置自己要的關鍵點如圖3- 59為起始關鍵點、圖3- 60為關鍵點結尾，假如要一直讓動畫移動不同的位置可以在0-20、20-40、40-60設置不同的動畫，他就會依序的移動，一個模型通常會有很多物件組成，可以讓多個物件同時移動，讓動畫看起來更豐富。

3-2-5 Loft

Loft是一個很好用的功能，一開始在繪製一些模型的時候，常常遇到一些無法解決的問題，例如一些曲面無法做得很自然，像是燈籠的牽繩或者寶箱的曲面，如果使用『可編輯多邊形』拉取就會變得很費時，而且又很不自然，後來在網路上尋找各種教學跟功能才發現這個方法。首先要繪製出一個路徑在繪製出一個斷面，在路徑上獲取斷面圖形就可以了。(複合對象>放樣)如圖3- 62。



圖3- 62放樣

Loft有分兩種方式一種是『獲取圖形』跟『獲取路徑』如圖3- 63，兩種都可以形成物件，不一樣的是物件形成的位置。



圖3- 63放樣功能

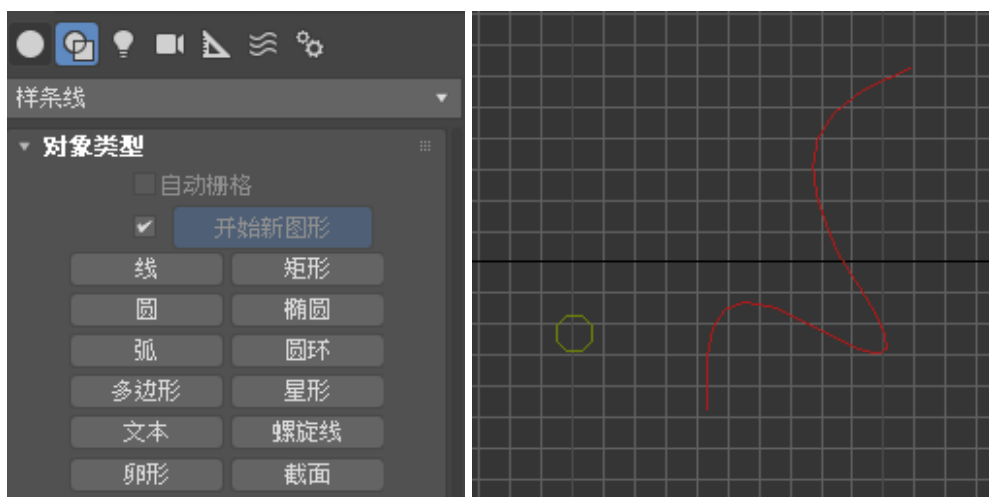


圖3- 64線與斷面

Loft要先繪出一條路徑跟斷面，假設我要畫一條圓線，先在樣條線繪出線跟圓，因為我要獲取圓的斷面，線為路徑，所以要選取線，獲取圓的路徑如圖3-64，獲取成功就會如圖3- 65一樣。

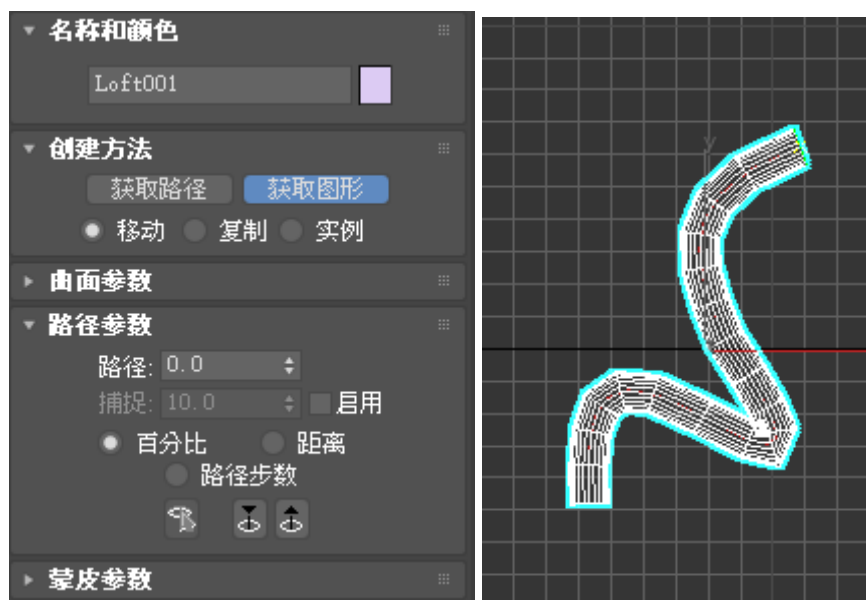


圖3- 65獲取圖形

3-2-6 布林計算運用

一開始在繪製模型的時候，有遇到一個問題，到底要怎麼把物件用出一個圓洞，像是初期學習時繪製起司跟骰子的時候就有遇到這個問題，布林可以將物件相加、相減或相交的計算式，在物件結合或者凹洞處理的特殊造型上面，常常會用到。

假設以一個立方體跟圓形當範例，立方體為A圓形為B如圖3- 66，若要結合使用交集($A+B$)可以變成圖3- 68，可以用來結合物件，若是使用差集就可以畫出骰子所需的凹洞如圖3- 67。

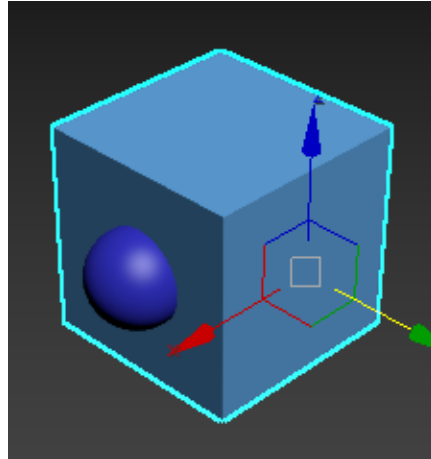


圖3- 66圖形A與B

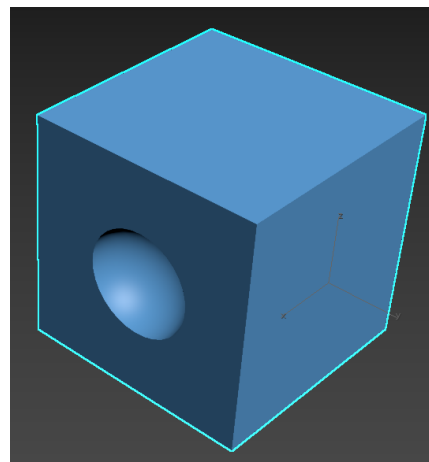


圖3- 67差集

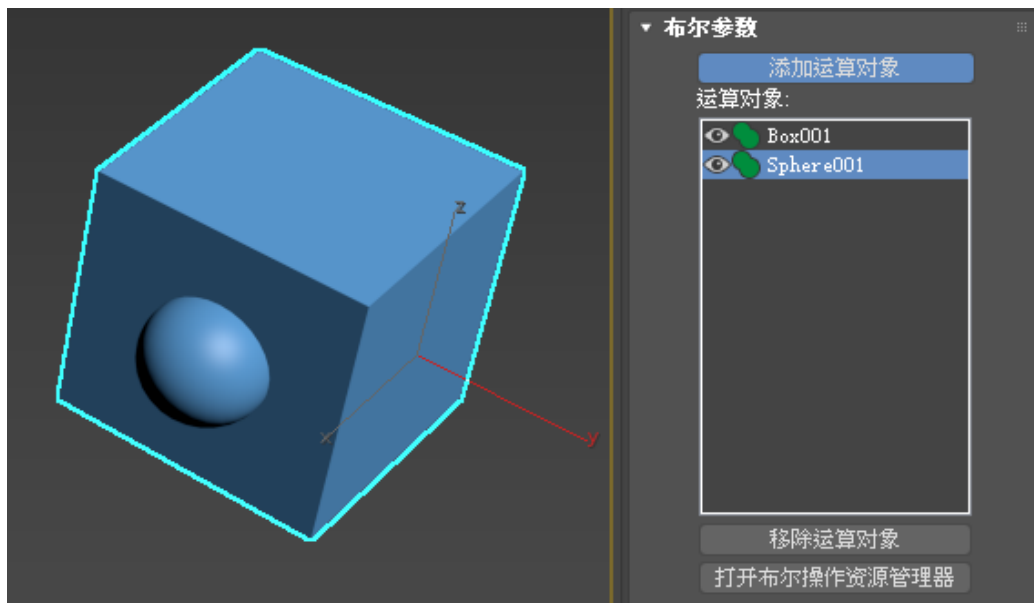


圖3- 68交集

3-2-7 材質編輯器

當初使用3Ds Max的時候，物件的顏色都是使用最簡單的單色來上色，都沒有甚麼問題，因為只需要做一些簡單機關，還有方塊跟路徑，但是當要做一些地板、房子、動物之類的，如果只用單色就會顯得太單調，最重要的是如果使用裡面的顏色來上色，轉成fbx檔匯入Unity裡面就會變成沒有顏色，物件全部都是灰色的，變成在做遊戲的同學，還要再Unity裡面上色，非常的麻煩，後來就上網找要怎麼解決，就發現這個功能了，不但可以把圖的材質貼上去模型上，也可以加入許多反光之類的，可惜後來發現在3Ds Max裡面的光影跟反光，在Unity裡面都無法顯示，但是貼圖還是可以正常顯示的，所以只要加貼圖，都會使用這個功能如圖3- 69。



圖3- 69材質編輯器

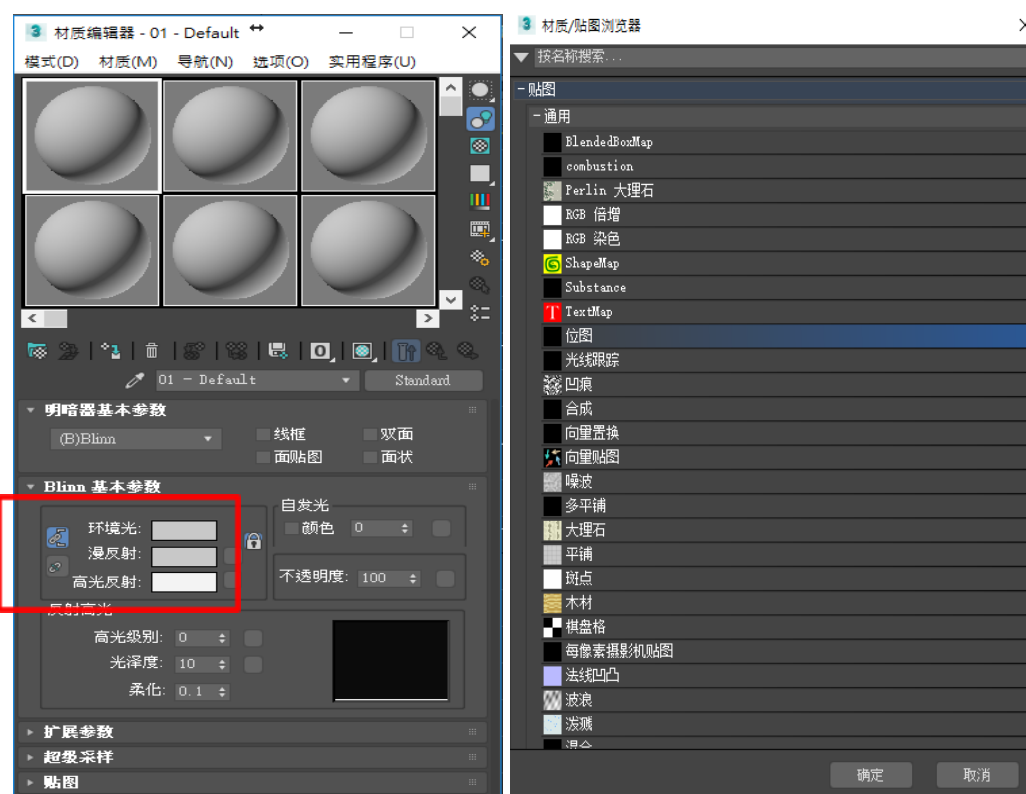
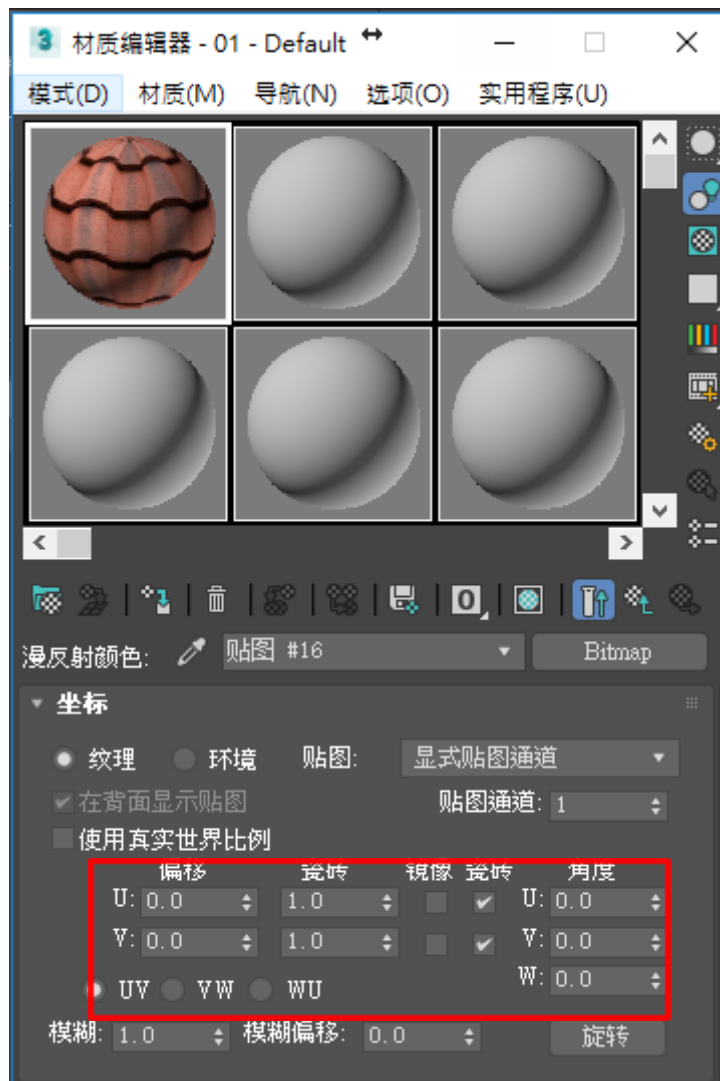


圖3- 70空白材質球

選取 REF_Ref514965057 \h 圖3- 69的材質編輯器，進入之後選取要匯入的空白材質球，點選之後如果要匯入的材質是貼圖，就要選取漫反射旁邊的小灰色方塊，選取『位圖』如 REF_Ref514971823 \h 圖3- 70，選取要繪出的圖片創造出材質球在拉入模型上面。



REF_Ref514971525 * MERGEFORMAT 圖3- 71
偏移選項在紅色方框裡面，可以控制物件上面的貼圖數量還有位置，可以旋轉貼圖或者改變位置來讓其他物件貼圖對準，如果貼圖材質太大也可以增加數量來讓它變小還有解析度變高。

圖3- 71偏移選項

在使用這個編輯器的時候，匯成fbx轉入Unity有發生過問題，就是不管在這裡的材質有沒有貼上去，只要轉到Unity就會發生物件都是灰白色的，沒有任何的材質在上面，後來的解決方法就是在匯出的時候勾選『嵌入的媒體』，才可以生成一個fbm的貼圖資料夾到Unity，不然在Unity上面重新貼材質會很麻煩！

若要把模型丟入Unity若使用原始max檔案會無法讀取，所以要轉成Unity可以讀取的fbx檔案(文件>導出>導出文件)如圖3- 72，勾選『嵌入的媒體』如圖3- 73，在Unity就可以正常顯示了。



圖3- 72導出選項

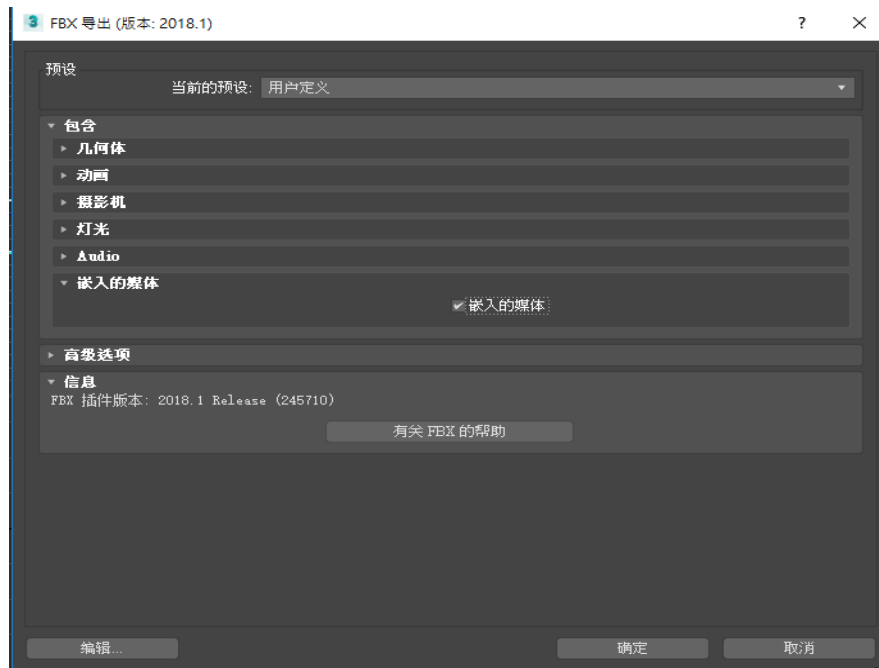


圖3- 73嵌入式媒體勾選

3-3 語音處理辨識及輸出

語音處理辨識及輸出是人說話經由麥克風設備輸入再透過電腦辨識為文字或是由文字轉換為聲音再由喇叭設備輸出的技術，可用於手機軟體及電腦軟體中，現在許多智慧型手機都有此功能，如Apple手機的Siri、Android手機的語音、Windows語音等，都是語音辨識技術的軟體，我們套用語音處理辨識及輸出於遊戲中，增加遊戲趣味性。

3-3-1 語音辨識

Google Cloud Speech Recognition是Unity的跨平台語音紀錄與辨識的套件，支援Android、iOS、Windows、Mac、Linux等，由於語音辨識和API的製作以我們的程度無法做出來，我們套用Google Cloud Speech Recognition套件並且修改我們需要使用的部分來使用，如圖3- 74為套件範例程式，可以選擇自己國家的語言做語音辨識成文字，如圖3- 75為套件範例程式做語音辨識的動作後呈現的樣子，字串中會顯示辨識最佳的選項做為辨識字串，也會顯示可能為辨識字串的替代字串，但範例程式的比較麻煩是先按下Start來執行收音，收完音之後再按下Stop來停止收音並辨識。

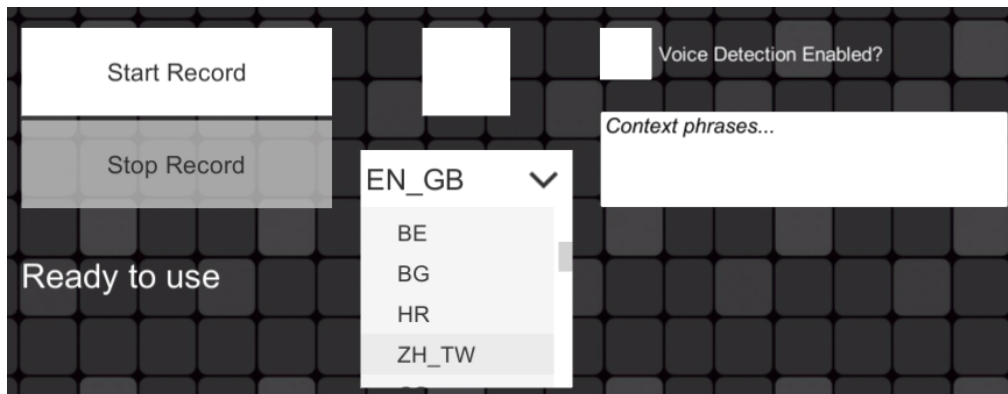


圖3- 74套件範例

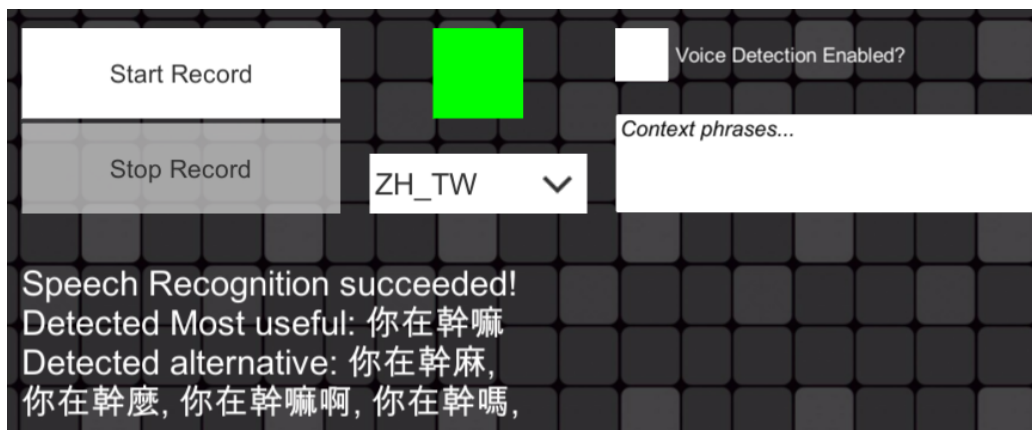


圖3- 75範例辨識

如程式3- 27為範例程式所使用的程式碼，第71~78行為Start的程式碼，第80~85行為stop的程式碼，我們修改為如程式3- 28中第98行以按鍵F3來控制語音辨識，當按住F3可以做輸入聲音的動作，第104行放開F3則停止輸入聲音。

```

71 private void StartRecordButtonOnClickHandler()
72 {
73     _startRecordButton.interactable = false;
74     _stopRecordButton.interactable = true;
75     _speechRecognitionState.color = Color.red;
76     _speechRecognitionResult.text = string.Empty;
77     _speechRecognition.StartRecord(_isRuntimeDetectionToggle.isOn);
78 }
79
80 private void StopRecordButtonOnClickHandler()
81 {
82     _stopRecordButton.interactable = false;
83     _speechRecognitionState.color = Color.yellow;
84     _speechRecognition.StopRecord();
85 }

```

程式3- 27原按鍵程式

```

98 if (Input.GetKeyDown(KeyCode.F3))
99 {
100     Debug.Log("按下F3");
101     _speechRecognitionResult.text = string.Empty;
102     _speechRecognition.StartRecord(_isRuntimeDetectionToggle.isOn);
103 }
104 else if (Input.GetKeyUp(KeyCode.F3))
105 {
106     _speechRecognition.StopRecord();
107 }

```

程式3- 28修改按鍵後

我們修改按鍵方式後並且將字串設成只會顯示辨識最佳的字串，並不會顯示出可能替代的字串，如程式3- 29為範例程式的字串顯示程式碼第124~125行會告訴我們最佳字串，第137行會顯示可能的替代字串，第141行為辨識失敗的字串，我們修改為如程式3- 30中第80行直接顯示辨識出最佳的字串，在第84行直接顯示辨識錯誤的字串。

```

114 private void SpeechRecognizedSuccessEventHandler(RecognitionResponse obj, long requestIndex)
115 {
116     if (!_isRuntimeDetectionToggle.isOn)
117     {
118         _startRecordButton.interactable = true;
119         _speechRecognitionState.color = Color.green;
120     }
121
122     if (obj != null && obj.results.Length > 0)
123     {
124         _speechRecognitionResult.text = "Speech Recognition succeeded! Detected Most useful: "
125             + obj.results[0].alternatives[0].transcript;
126
127         string other = "\nDetected alternative: ";
128
129         foreach (var result in obj.results)
130         {
131             foreach (var alternative in result.alternatives)
132             {
133                 if (obj.results[0].alternatives[0] != alternative)
134                     other += alternative.transcript + ", ";
135             }
136         }
137         _speechRecognitionResult.text += other;
138     }
139     else
140     {
141         _speechRecognitionResult.text = "Speech Recognition succeeded! Words are no detected.";
142     }
143 }

```

程式3- 29字串顯示

```

76 private void SpeechRecognizedSuccessEventHandler(RecognitionResponse obj, long requestIndex)
77 {
78     if (obj != null && obj.results.Length > 0)
79     {
80         _speechRecognitionResult.text = obj.results[0].alternatives[0].transcript;
81     }
82     else
83     {
84         _speechRecognitionResult.text = "無法辨識語音!";
85     }
86 }

```

程式3- 30修改後字串顯示

我們將語音辨識套件使用於第一關及第二關的闖關地圖中，如圖3- 76是在遊戲初始畫面中左下為辨識文字輸出樣子，右下為遊戲提示字串，如圖3- 77是當按住F3說出正確的通關密語，則門會開啟並且關閉提示字串，如圖3- 78是當說出錯誤的通關密語，則門不會開啟，並且辨識文字輸出的字串會變成你說出的字串並且提示字串會持續顯示。

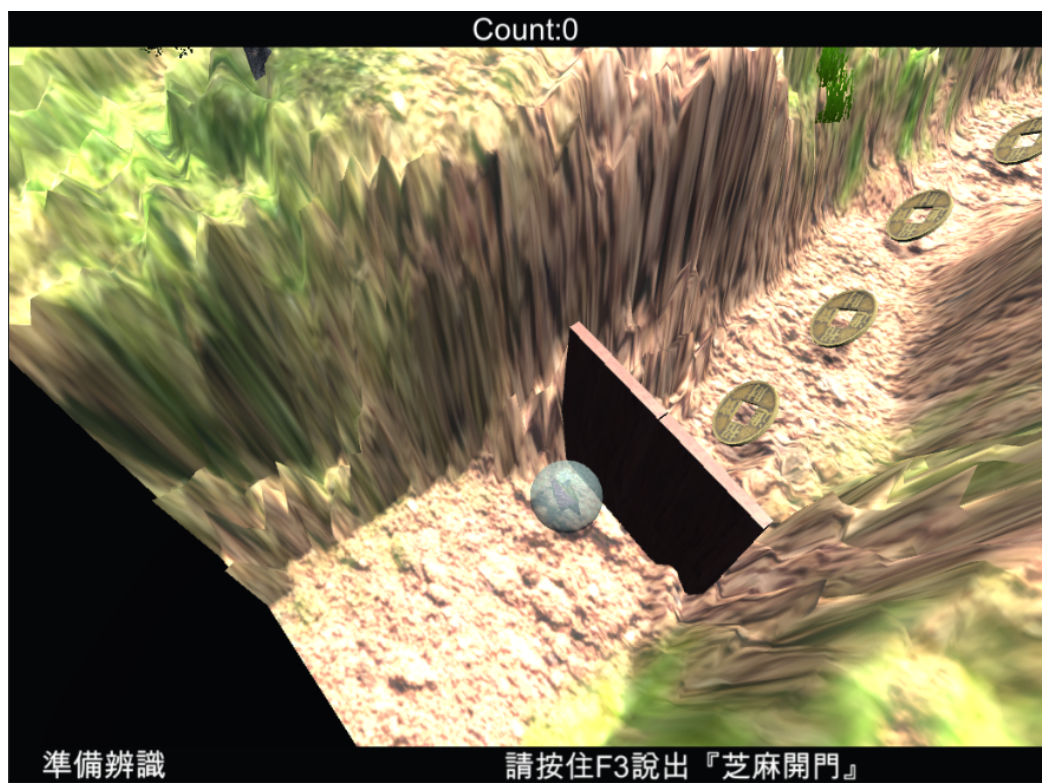


圖3- 76 關卡初始圖

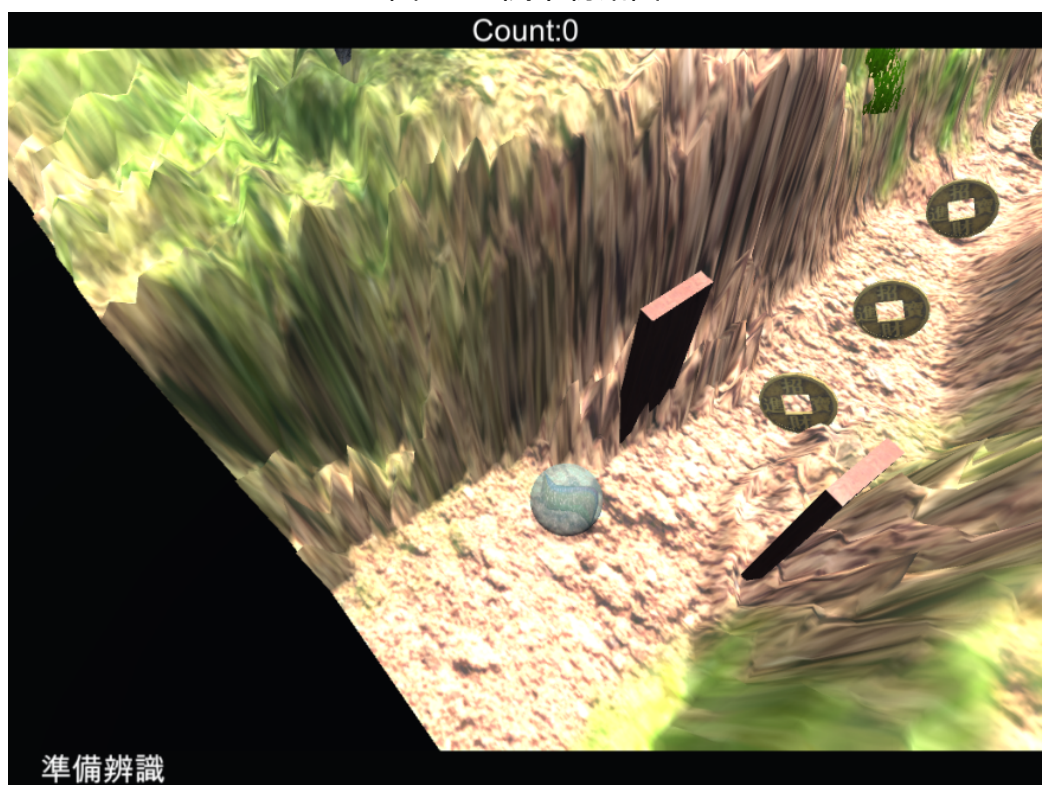


圖3- 77 辨識成功圖

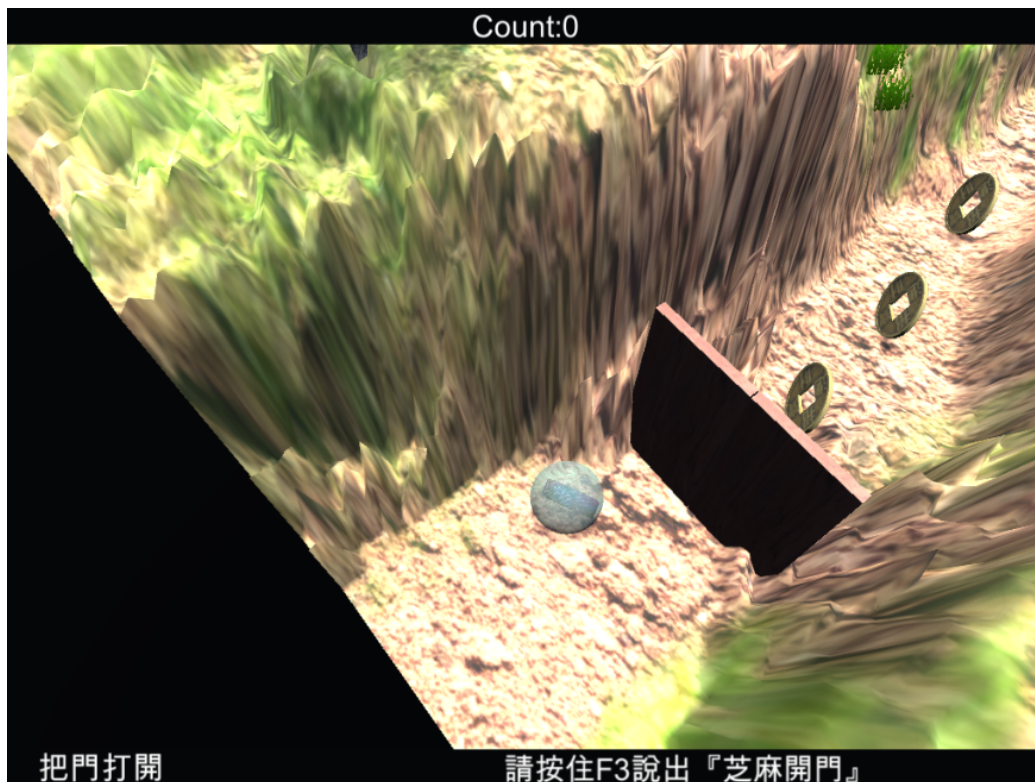


圖3- 78 辨識錯誤圖

在第二關中我們也使用語音辨識的功能來破關卡，如圖3- 79我們以猜謎的方式做問答辨識，程式碼以陣列取亂數選擇題目字串顯示於右下提示字串區，如程式3- 31中第35~40行為題目字串的陣列，第41~46行為答案字串的陣列，當到達第一個燈籠的土地時會觸發亂數取題目如程式3- 32，並且不會重複出現一樣的題目來做問答。

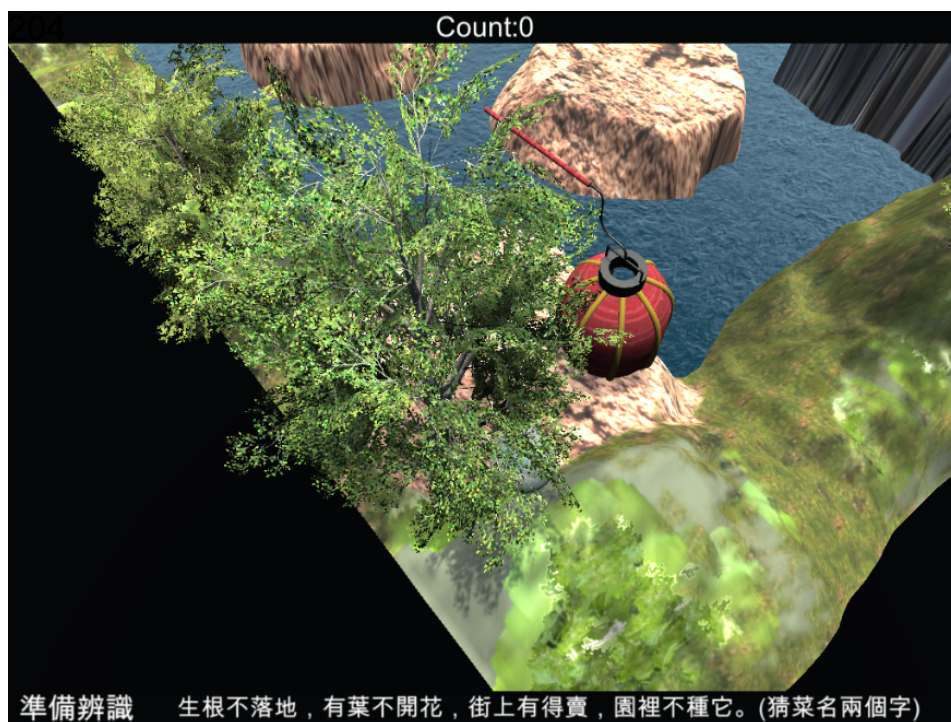


圖3- 79第二關圖

```

35 string[] title = {"二個男人光著屁股坐在石頭上。(猜成語四個字)",
36                  "地裡走，溝裡串，捎著針，忘了線。(猜動物兩個字)",
37                  "學狗坐，沒狗高，又沒耳朵又沒毛。(猜動物兩個字)",
38                  "個兒高又大，脖子似吊塔。和氣又善良，從來不打架。(猜動物三個字)",
39                  "麻屋子，紅帳子，裡面睡個白胖子。(猜豆類兩個字)",
40                  "生根不落地，有葉不開花，街上有得賣，園裡不種它。(猜菜名兩個字)"};
41 string[] Ansl = {"一石二鳥",
42                  "刺蝟",
43                  "青蛙",
44                  "長頸鹿",
45                  "花生",
46                  "豆芽"};

```

程式3- 31題目答案陣列

```

255 if (player.gameObject.name == "Cubel")
256 {
257     InvokeRepeating("timer1", 1, 1);
258     int randomInt = UnityEngine.Random.Range(0, 6);
259     while (RandomList.Contains(randomInt))
260     {
261         randomInt = UnityEngine.Random.Range(0, 6);
262     }
263     RandomList.Add(randomInt);
264     NowText.text = title[randomInt];
265     Ans.text = Ansl[randomInt];
266     Checkpoint = 1;
267     timmer_i = 0;
268 }

```

程式3- 32觸發亂數取題目

3-3-2 語音輸出

在Google翻譯有個文字轉語音的功能，我們嘗試以這種方式做出文字轉成語音的功能，原本在Google使用的網址為如圖3- 80，第1行為舊式網址列，進入會出現『存取的要求遭到拒絕』，但以第2行增加『client=tw-ob』即可使用，我們以此製作一個程式。

如程式3- 26第20行將Google translate的輸出網頁音檔在第27行設定成AudioSource.clip，在第35行播放出來，但是Unity在Windows的模式GetAudioClip不支援MPEG檔案格式，在ios、Android模式執行可以成功如圖3- 81，在Android模式執行沒有錯誤提示，但是在Windows模式執行會有不支援的錯誤提示如圖3- 82。

```
1 https://translate.google.com/translate_tts?ie=UTF-8&tl=zh&q=%E4%BD%A0%E5%A5%BD
2 https://translate.google.com/translate_tts?ie=UTF-8&tl=zh&client=tw-ob&q=%E4%BD%A0%E5%A5%BD
```

瀏覽器預設編碼ie=utf-8 語言tl=zh 客戶端client=tw-ob 文字q=你好

圖3- 80 Google translate tts 網址

```
10 public string url;
11 public AudioSource source;
12 void Start()
13 {
14     source = gameObject.GetComponent<AudioSource>();
15     StartCoroutine(DownloadTheAudio());
16 }
17 IEnumerator DownloadTheAudio()
18 {
19     //不可讀取
20     //url = "https://translate.google.com/translate_tts?ie=UTF-8&tl=zh&client=tw-ob&q=%E4%BD%A0%E5%A5%BD";
21     //url = "http://billor.chsh.chc.edu.tw/sound/thund.wav"; //可讀取
22     url = "http://taira-komori.jp/sound/os/daily01/knocking_an_iron_door1.mp3"; //不可讀取
23
24     WWW www = new WWW(url);
25     yield return www;
26
27     source.clip = www.GetAudioClip(false, false, AudioType.MPEG);
28     source.clip = www.GetAudioClip(false, false, AudioType.WAV);
29 }
30
31 void Update()
32 {
33     if (Input.GetKeyDown(KeyCode.F10))
34     {
35         source.Play();
36     }
37 }
```

程式3- 26 Text to speech

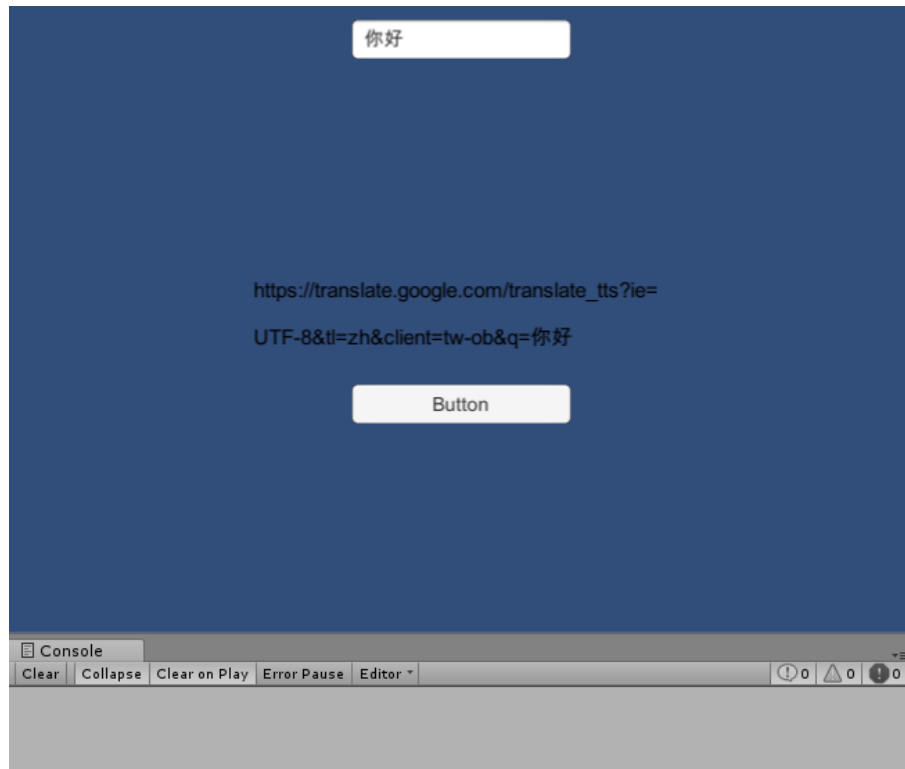


圖3- 81 andriod模式

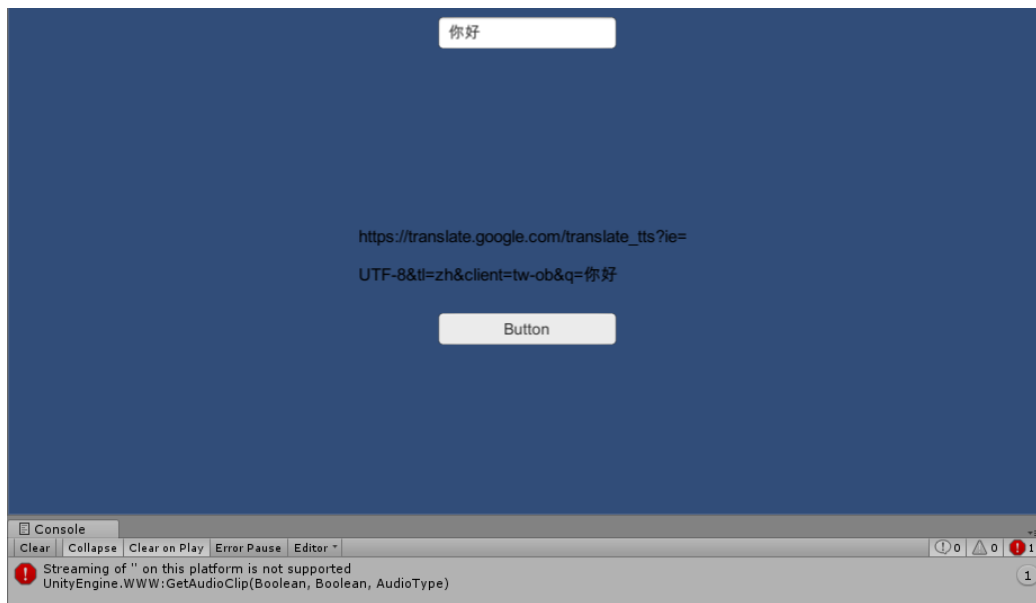


圖3- 82 Windows 模式

我們使用過其他的API程式也是Windows模式不支援MPEG的檔案格式，如圖3- 83中以Windows模式執行，也有不支援的情形，在圖3- 84中Android模式中可以正常執行，這個程式可以使用其他音檔格式來執行，但礙於速度非常慢，所以我們不採用。

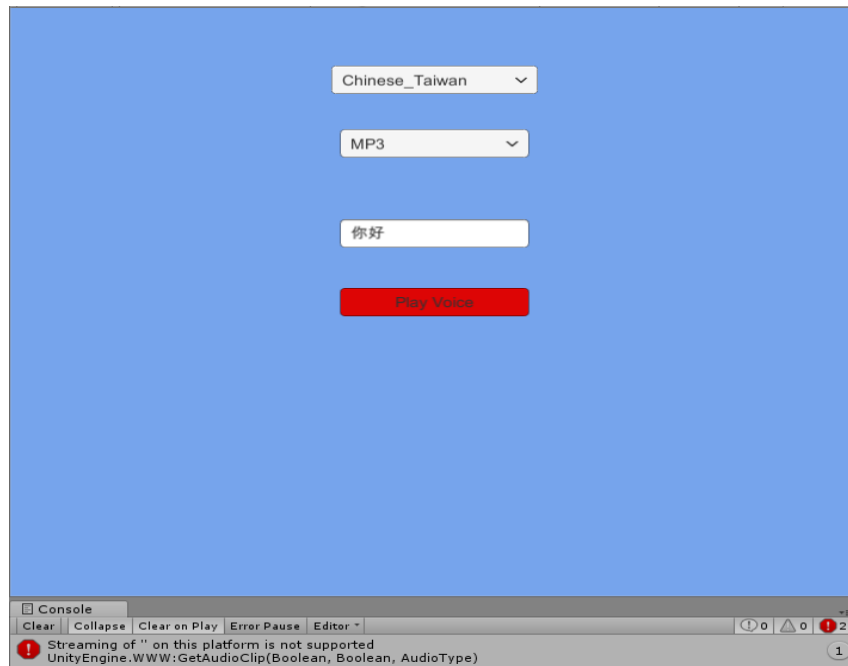


圖3- 83 其他API程式 Windows 模式

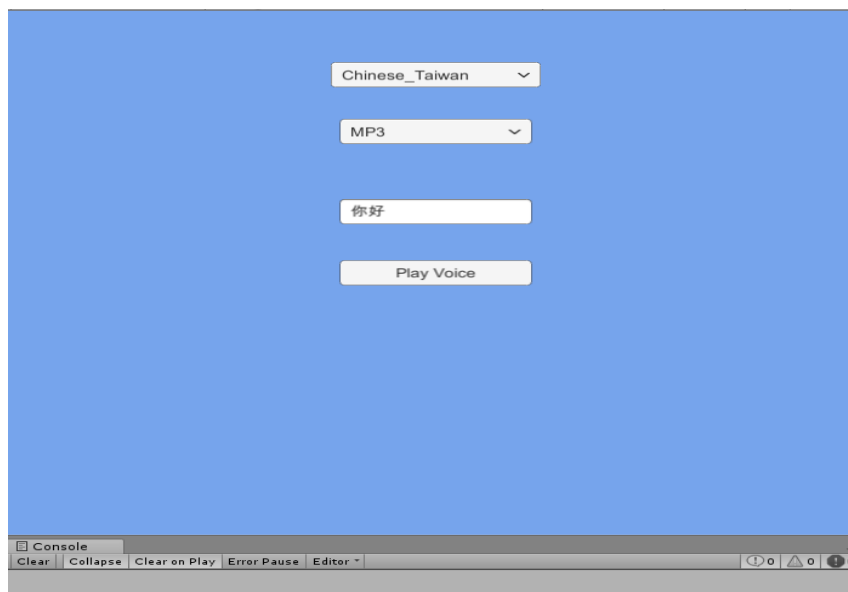


圖3- 84其他範例程式android模式

最後我們不使用Text to Speech，以輸出後直接下載MP3檔案放置專案的方式來套用，不會發生上列問題，在任何模式都可以直接播放MPEG音檔如圖3- 85。



圖3- 85 Google tts mp3檔

第4章、專題成果簡介

此遊戲是一款復古風格遊戲，並且加入我們的創意，主要推廣的群眾是老年的玩家，在現代逐漸高齡化的社會，希望能藉由此遊戲預防老年的一些問題，與年長的玩家互動，讓玩家能因為此遊戲喚起自己以往的故事與回憶，增加與年輕有更多良好的互動。

4-1 第一關簡介

遊戲中有許多的元素，當中結合語音互動、空間概念、反應力、記憶力。在第一關開始的時候會播放經典歌曲「望春風」勾起玩家以往的記憶，關卡當中有著許多的機關等著玩家，在突破機關的過程中會有語音與玩家互動，關卡就像是個迷宮如圖4- 1，玩家也必須要具備認路的能力，遊玩的過程當中，玩家會經由古早童玩與創意小物進入我們所準備的小遊戲，第一關有三個小遊戲分別具備著不同的挑戰。



圖4- 1 第一關俯瞰圖

玩家會先進入第一個小遊戲「陀螺遊戲」如圖4- 2，此遊戲從以往三合院的概念設計，並且設置在清幽的山谷當中，遊戲具有空間概念與認知能力，把不同的難度依序破關，也要記憶陀螺的位置才能輕鬆取勝。

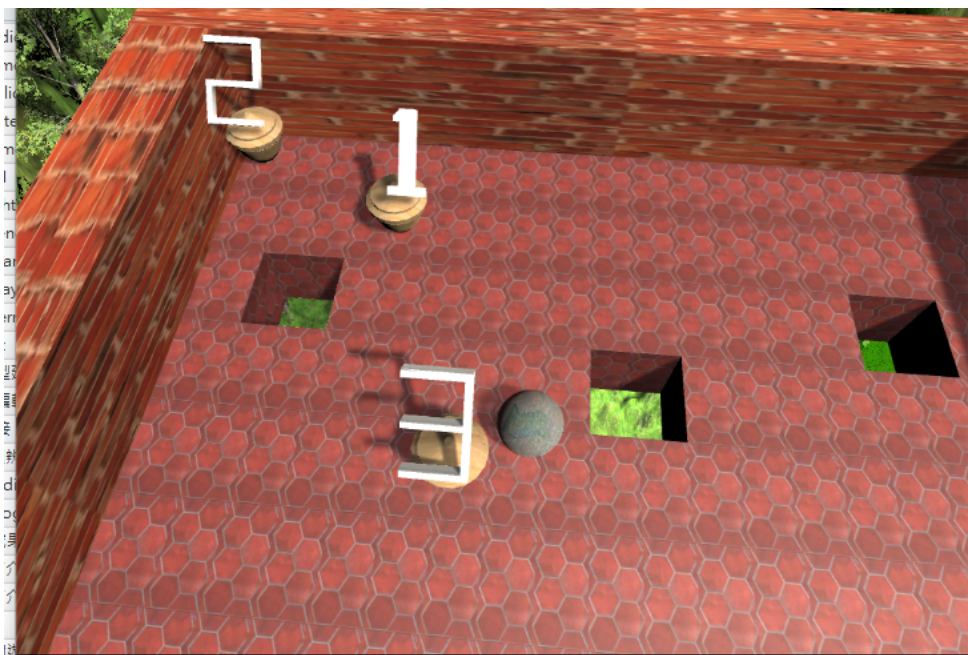


圖4- 2陀螺遊戲

象棋遊戲如圖4- 3是一個挑戰玩家反應力的小遊戲，由年長玩家喜愛遊玩的象棋啟發並且變換成我們風格的遊戲，玩家要知道象棋的大小排序，並且刺激玩家的眼與手，挑戰玩家的反應。

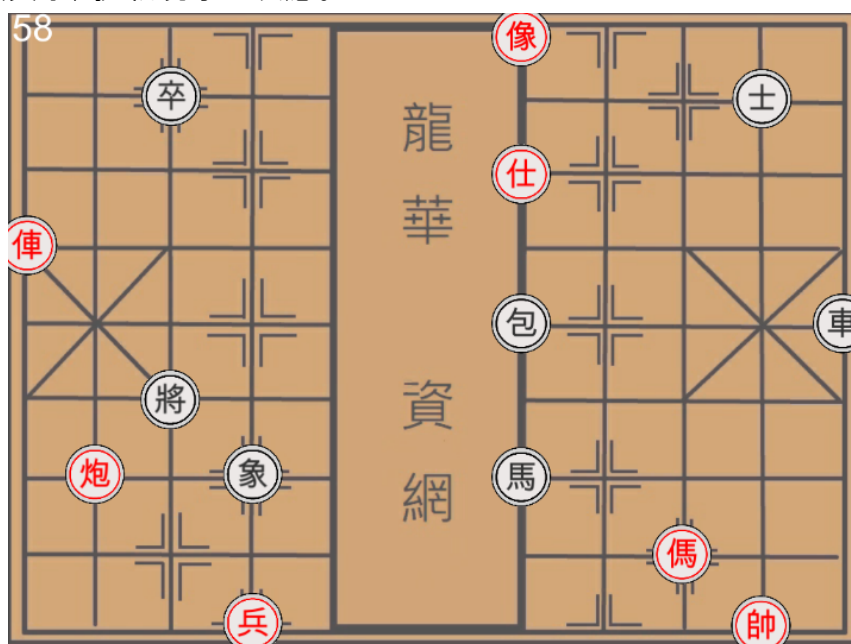


圖4- 3象棋遊戲

骰子遊戲由古早時候香腸攤會有的十八仔骰子遊戲啟發，但不鼓勵玩家以賭博的心態遊玩，我們將十八仔變換風格，又分成了四種模式如圖4- 4，在有著些許的運氣成分當中，添加了不同的遊玩模式，使玩家必須要去動腦思考如何讓自己取得勝利的方式。

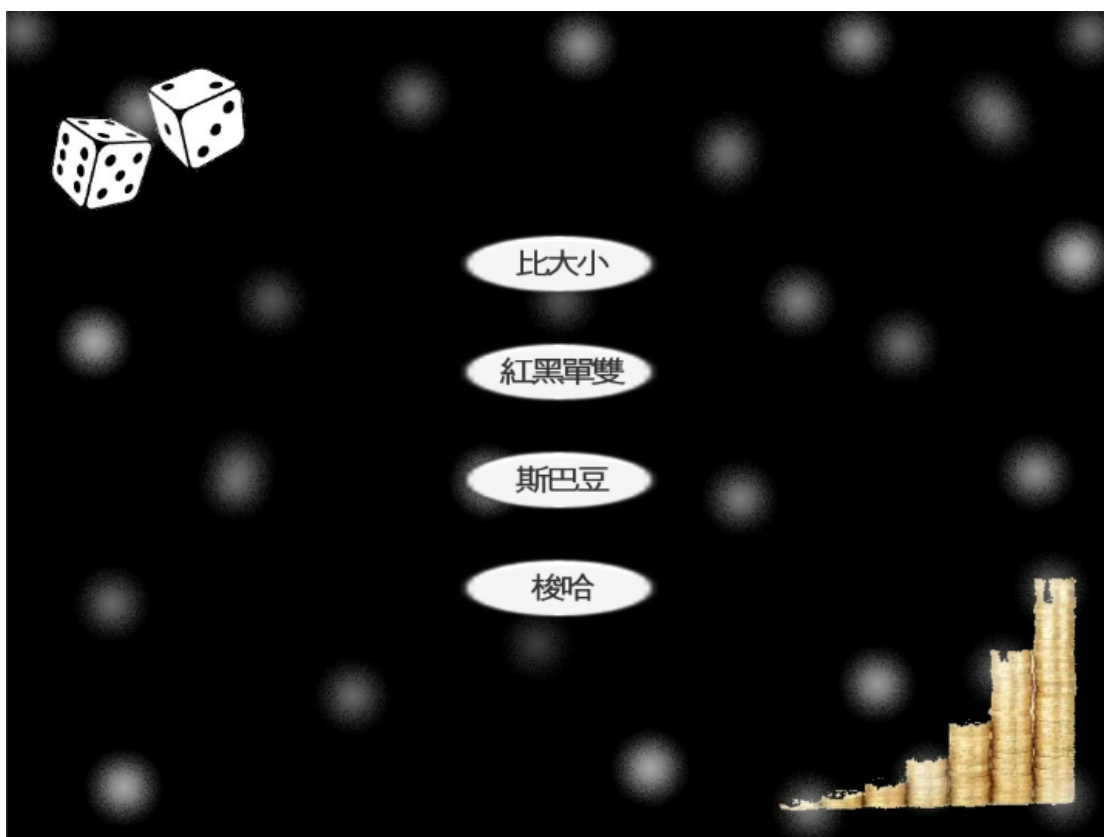


圖4- 4骰子遊戲介面

4-2 第二關簡介

玩家將第一關闖蕩完成的時候會進入到第二關如圖4- 5, 第二關提供了兩種不同風格的機智問答遊戲給玩家, 首先玩家會在玄武岩懸崖下方的起點, 進行台灣傳統文化猜燈謎活動, 並且有著燈籠配合著景色, 讓玩家腦筋急轉彎, 以語音回答出解答才能過關登頂進行最後的小遊戲。

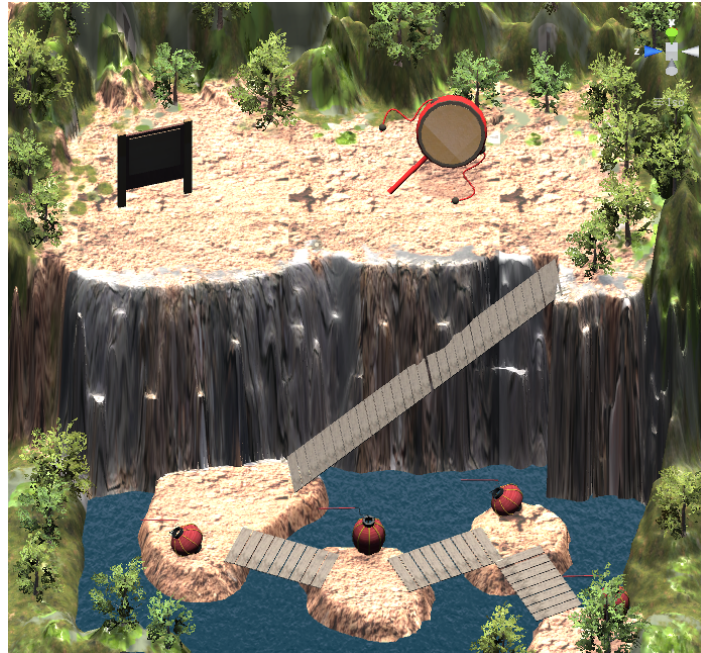


圖4- 5第二關

在最後進入的圖片問答小遊戲如圖4- 6, 是一個讓玩家勾起回憶的重要關卡, 在遊戲當中會出現照片並且要點選正確的答案, 回答越多正確的題數可以獲得更高的分數, 當中的題目有許多懷舊的事物, 也加入了當年火紅的布袋戲, 讓玩家可以回味那傳統的台灣文化。



圖4- 6懷舊問答

第5章、第一關遊戲

5-1 關卡遊戲

第一關關卡遊戲為初始主體，遊戲地圖俯瞰如圖5- 1。



圖5- 1第一關俯瞰圖

遊戲控制方法：

W、A、S、D或Up、Down、Lift、Right為上下左右的控制。

ESC為關閉程式。

F3為語音辨識，按住F3可以說出通關密語或謎語，放開F3後會辨識答案是否正確。

遊戲初始畫面如圖5- 2，左下方為語音辨識的字串，中間正上方為吃金幣的

計分板, 右下方為提示字串的位置, 當向前碰到門會出現提示字串如圖5- 3, 按住F3答出正確的答案放開後門會打開得以繼續通行, 通行後如圖5- 4碰到金幣後金幣會消失且得到分數, 吃越多金幣分數越高。

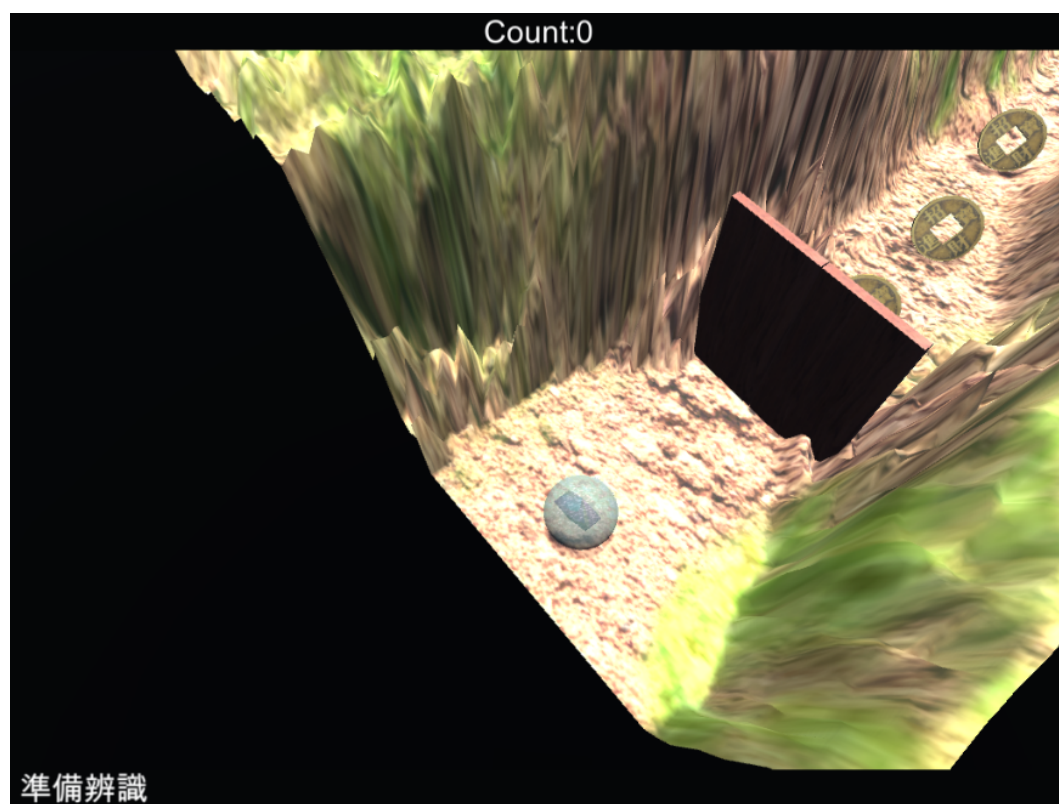


圖5- 2初始畫面

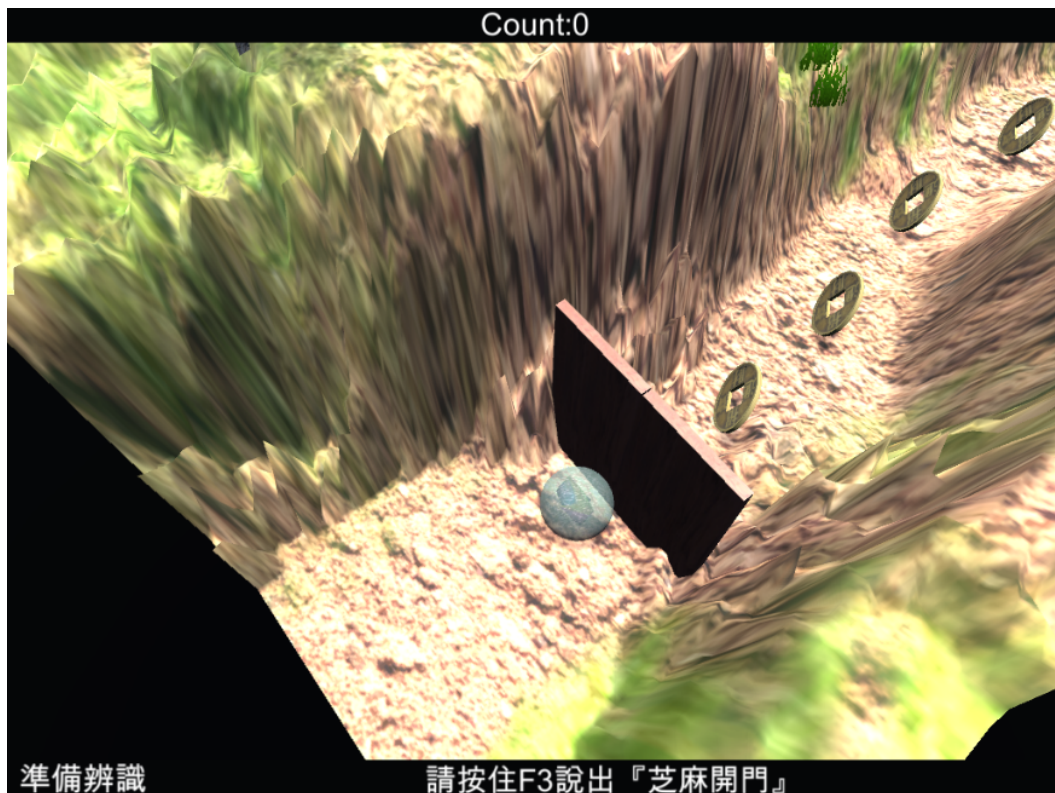


圖5- 3提示字串出現

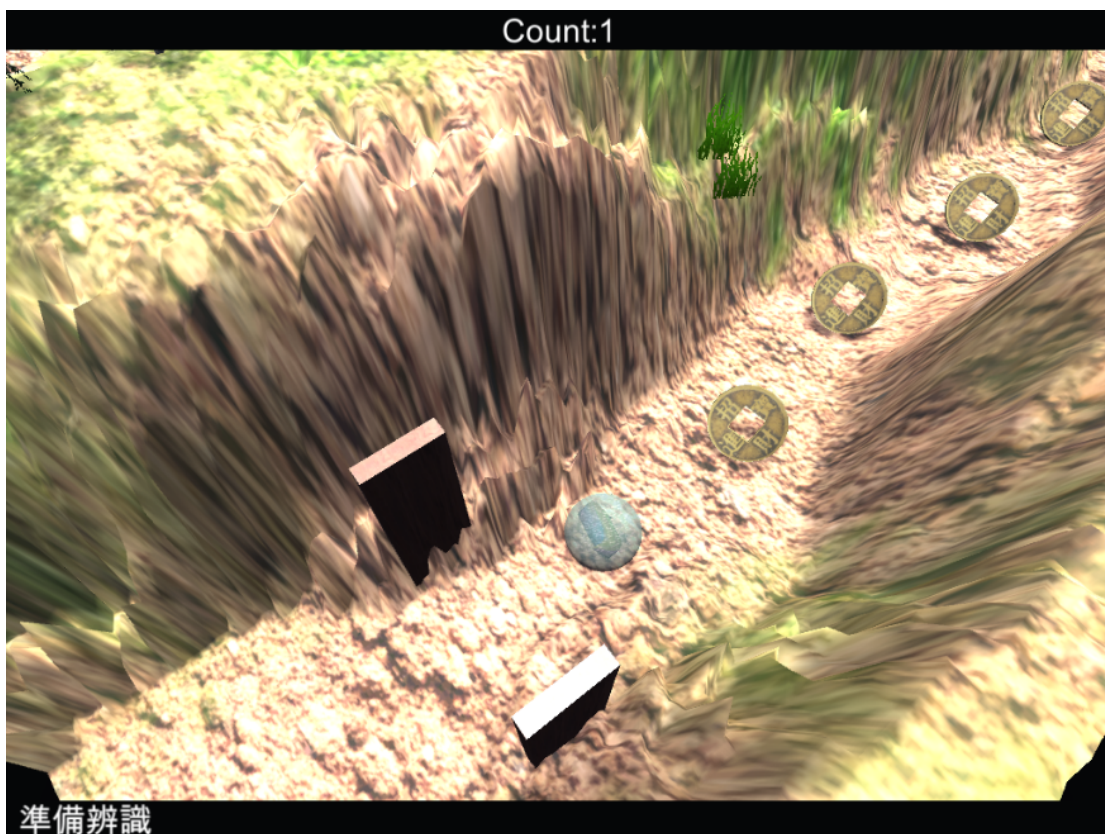


圖5- 4吃金幣計分

繼續行走會碰到機關如圖5- 5, 要碰到上方的拉桿才能夠觸發橋的機關得

以通行，如果掉到水裡則會回到橋前無法通行，過橋後會遇到尖刺機關會上下移動如圖5- 6，如同掉到水裡一樣觸碰到會回到關卡前無法通行，碰到寶箱如圖5- 7則會進入陀螺遊戲，並且給出關卡提示，完成遊戲後繼續行走則會遇到船必須過河如圖5- 8，掉下船碰到水一樣會回到船前並且重新再來，接下來會陸續遇到熊如圖5- 9則會進入象棋關卡，出來繼續行走會看到鐵柵欄，必須答出通關密語才能進入毆虫如圖5- 10，則鐵柵欄的通關密語會再進入陀螺遊戲時顯示，如果有記住則能順利通關，如果沒記住或不知道是這個關卡的則無法進入，最後走到竹蜻蜓如圖5- 11則恭喜突破第一關，會進入第二關關卡進行下一關的試煉。

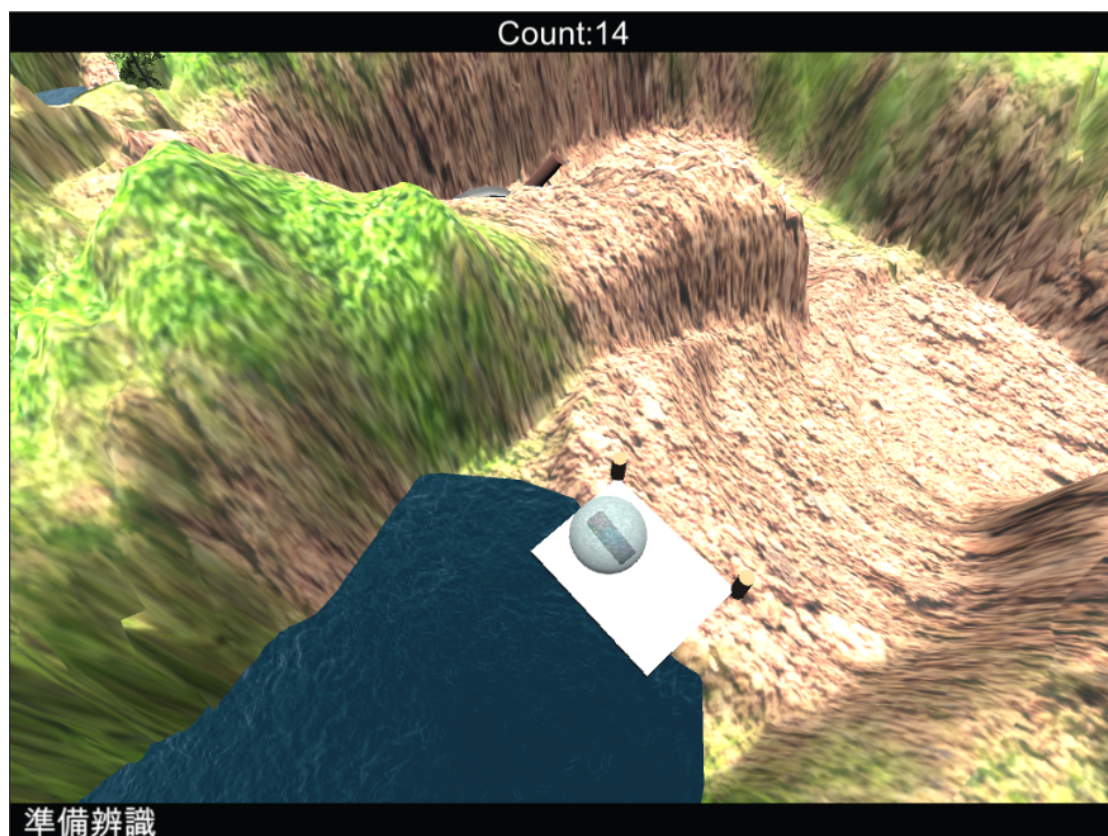


圖5- 5水橋過河機關



圖5- 6尖刺機關



圖5- 7寶箱圖



圖5- 8搭船渡河



圖5- 9熊

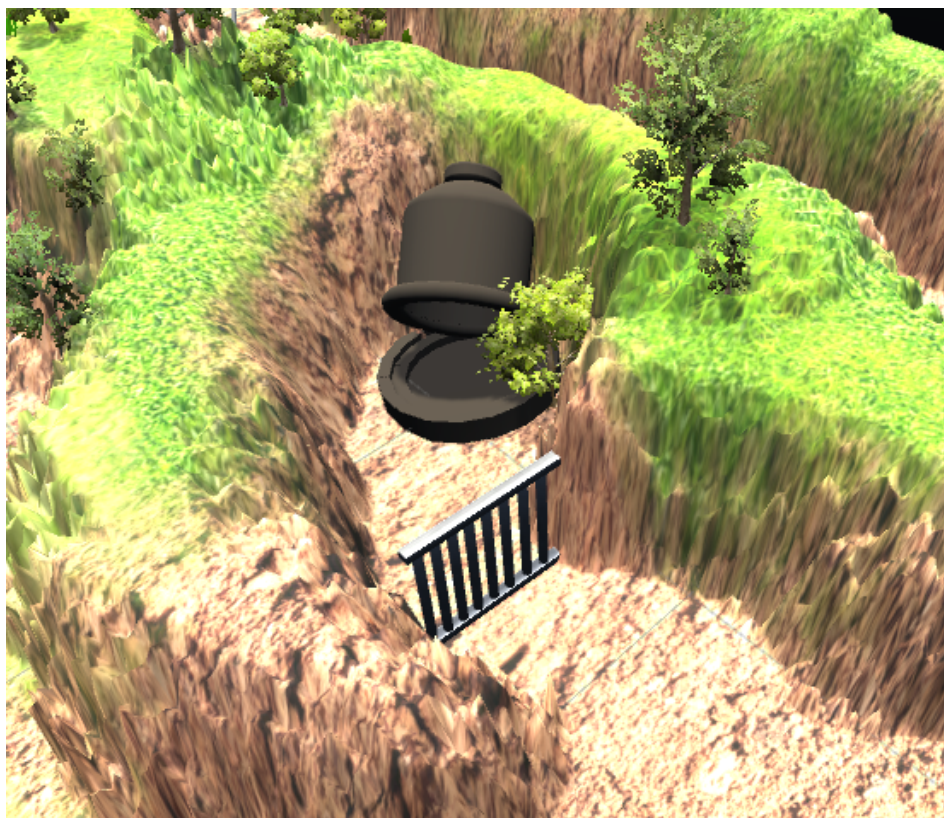


圖5- 10鐵柵欄和骰盅

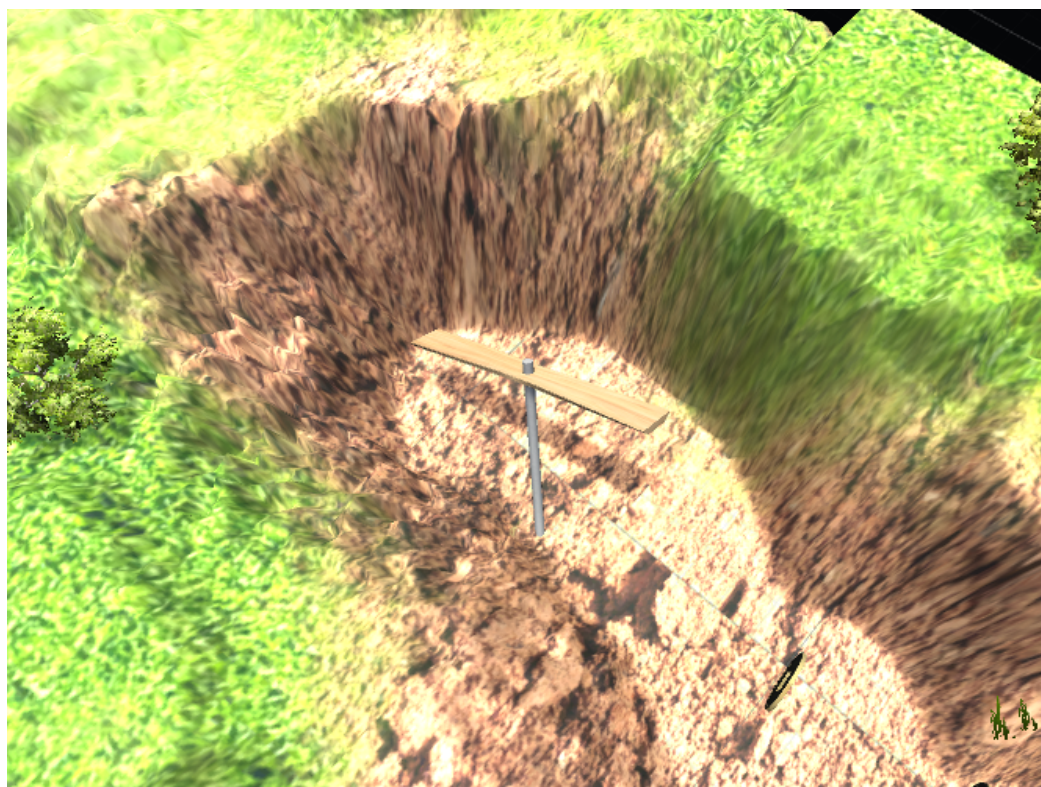


圖5- 11竹蜻蜓

5-2 陀螺遊戲

陀螺小遊戲分成簡單、一般、困難三個難度，讓玩家能慢慢地接受挑戰，玩家必須依照在陀螺上方的數字去依序碰撞才能破關，當玩家觸碰順序錯誤或掉入紅磚地板區域至界外區，將會在當次遊玩模式重新開始，如圖5- 12紅磚地板與陀螺各會顯示消失三個，並且不會限制玩家的遊玩時間，讓玩家能先輕鬆的遊玩，並且理解規則與熟悉操作，當玩家正確破關後進入一般模式如圖5- 13，以倍數成長陀螺與紅磚地板變成各顯示消失6個，並增加倒數時間30秒，讓玩家在稍有壓力的情況下進行遊玩，如圖5- 14困難模式變成了各12個陀螺與紅磚地板顯示消失，時間限制也延長至一分鐘，不過玩家如果沒能即時的熟悉操作的話，在有限的時間內是無法將困難模式破關的，如果玩家以較快的時間將困難模式完成，玩家將能獲得較高的分數。

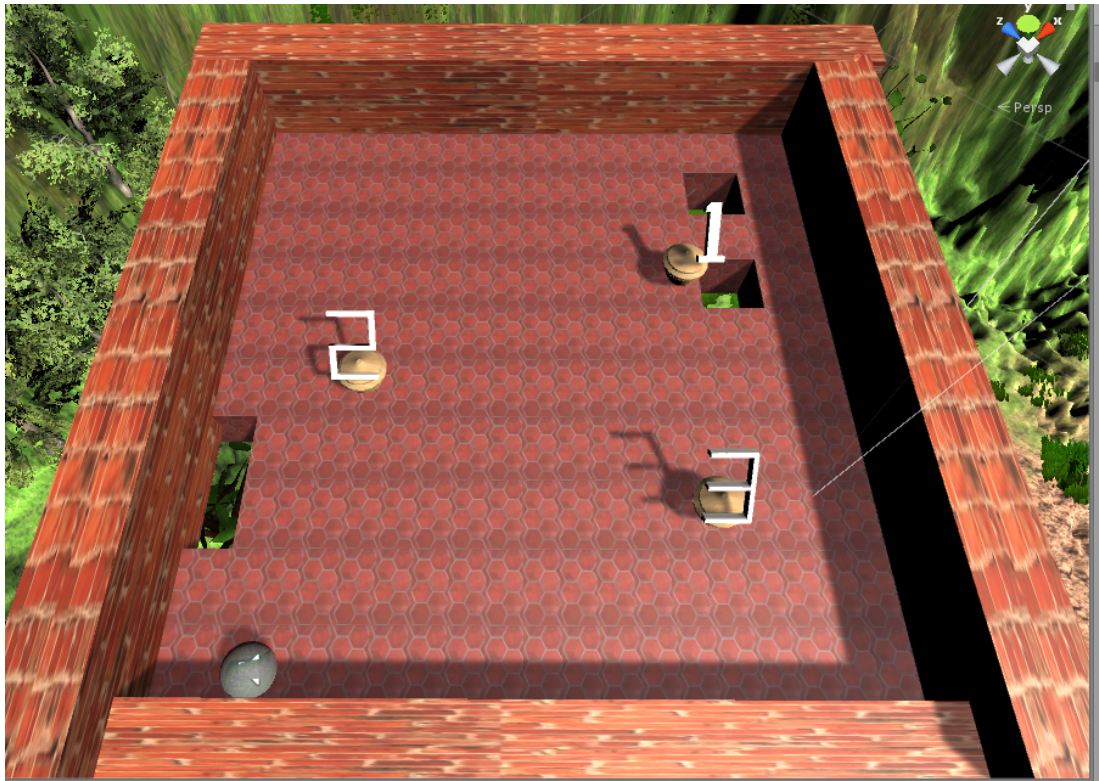


圖5- 12簡單模式

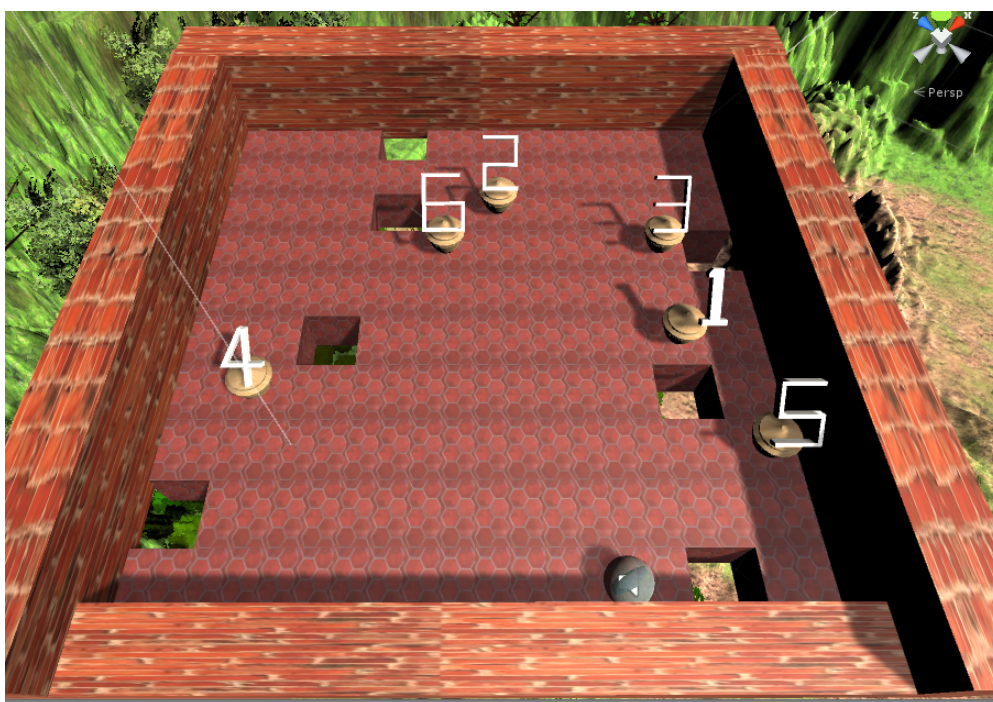


圖5- 13一般模式

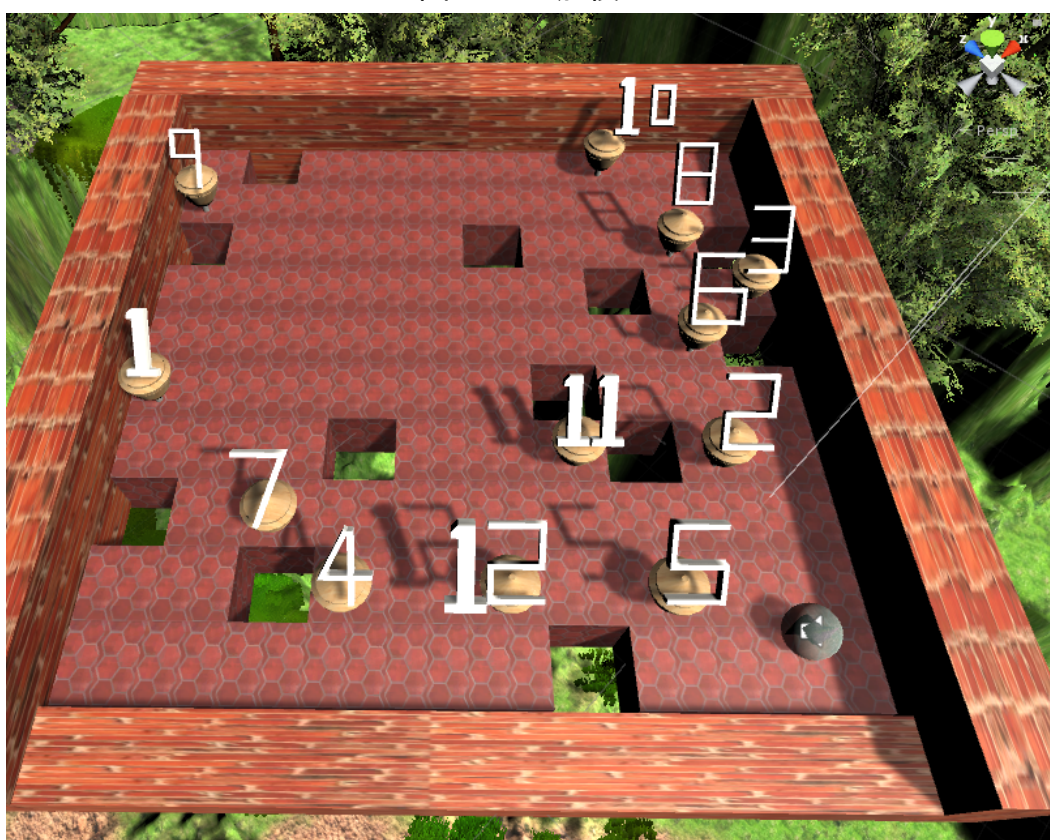


圖5- 14困難模式

5-3 骰子遊戲

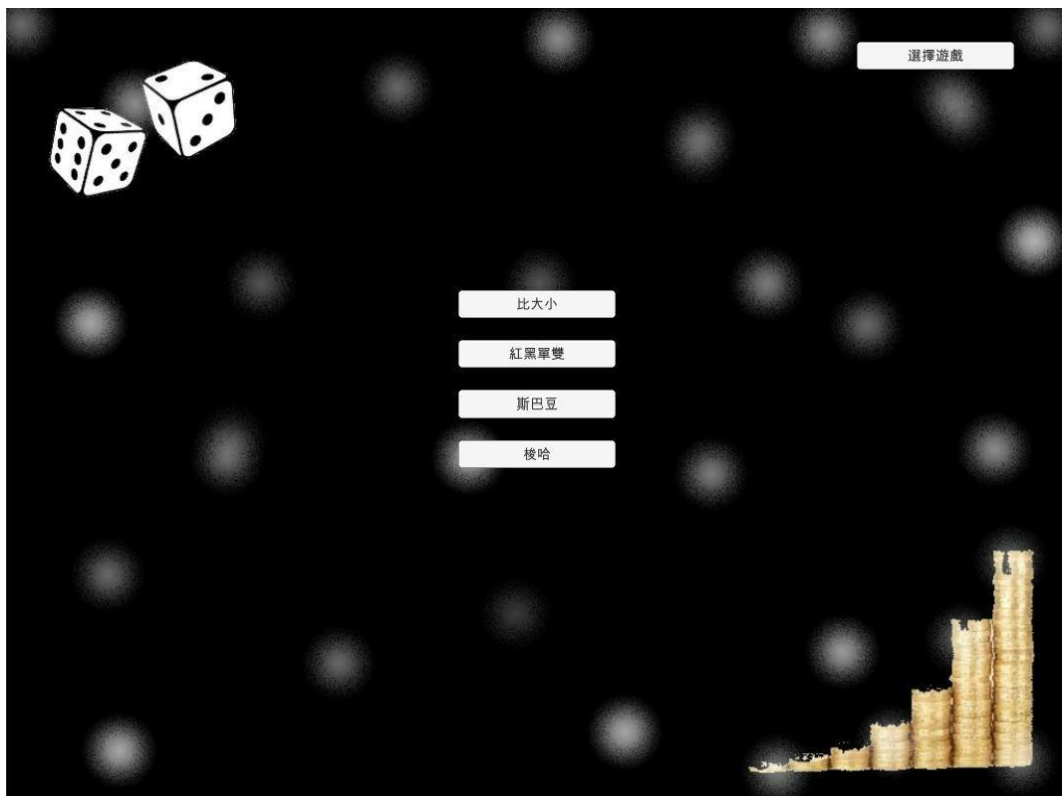


圖5- 15骰子介面

遊戲分為比大小、紅黑單雙、斯巴豆、梭哈等常見遊戲的變化，如圖5- 15。構想是以簡單的文字方塊UI-Text、按鈕UI-Button、3D物件、物理作用的鋼體 Rigidbody、程式C#Script給予物理力量以及遊戲的發想。

原先是想用2D製作，如圖5- 16，但是考慮時間非常充足，於是開始使用3D的構想來製作。

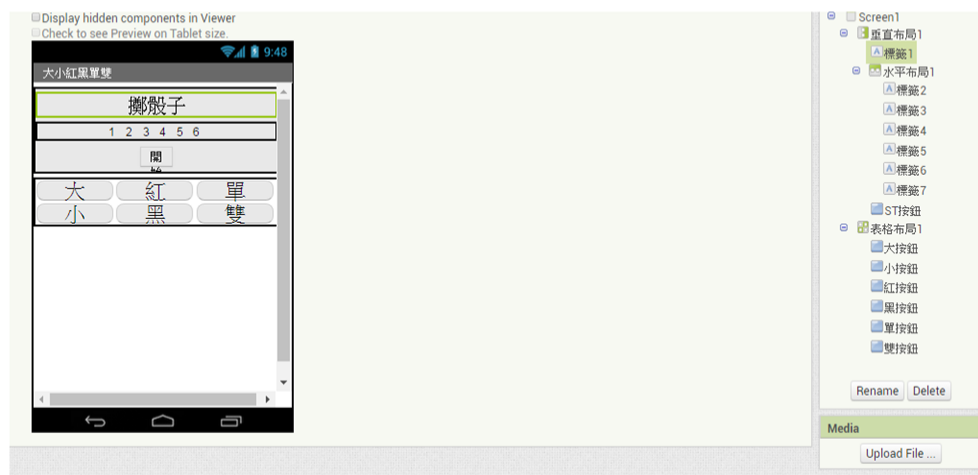


圖5- 16原構想

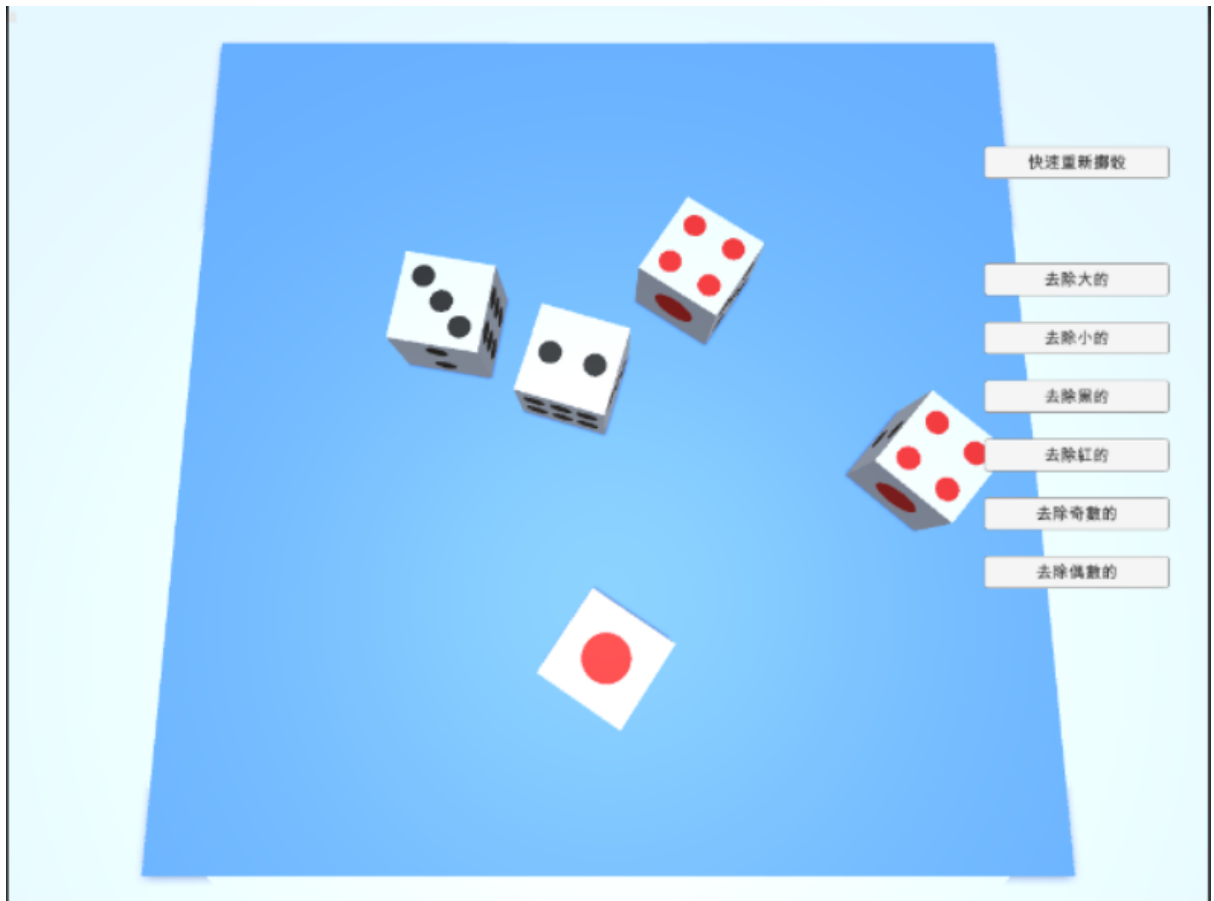


圖5- 17遊戲介面

後來是以簡單的3D方塊、地板plane物件，還有按鈕UI-Button來做雛形，如圖5-17。

比大小：

多了比較紅以及比較黑的選項，此部分的程式比較很單純，比大小只是簡單判斷，比紅和比黑則是要了解紅色點數為1和4，所以了解此概念的話，判斷也是十分簡單。

紅黑單雙：

此區的程式碼與比大小的寫法雷同，都是簡單的判斷，只是玩法是希望玩家使用兩個選項去除，最後取總數和，骰子總和越高，分數越高，反之總和如果越小就代表玩家並沒有思考，所以總分也更低。

斯巴豆：

玩法有分為4顆骰子和5顆骰子的差別，這次專題是使用4顆骰子的玩法；對於玩法的發想，把骰完後的點數，造成結果的原因找出，比如 1 1 2 3 總和5，你要找出兩個一樣的數字，代表你知道怎麼玩。

梭哈：

此區規則是最複雜的，因為判斷上分為8種，大小排序分別為大滿貫>鐵支>順子>葫蘆>三條>兩對>一對>單支。

大滿貫:

五顆骰子數值皆相同，判斷極為簡單。

Ex: 6 6 6 6 6

鐵支:

四顆骰子數值相同，一顆數值不相同。

Ex: 2 2 2 2 1

順子:

骰子數值連號。

Ex: 1 2 3 4 5 or 2 3 4 5 6

葫蘆:

三條配一對。

Ex: 3 3 3 1 1

三條:

三顆骰子數值相同即可。(如果剩下兩顆相同，即為葫蘆。)

Ex: 2 2 2 3 1

兩對:

兩個一對配上一個不相同的數。

Ex: 5 5 6 6 1

一對:

兩數相同為一組。

Ex: 1 1 2 3 4

單支:

沒有任何對子。

Ex: 1 3 4 5 6

最後考量老人視力問題，將選項做微調如圖5- 18。

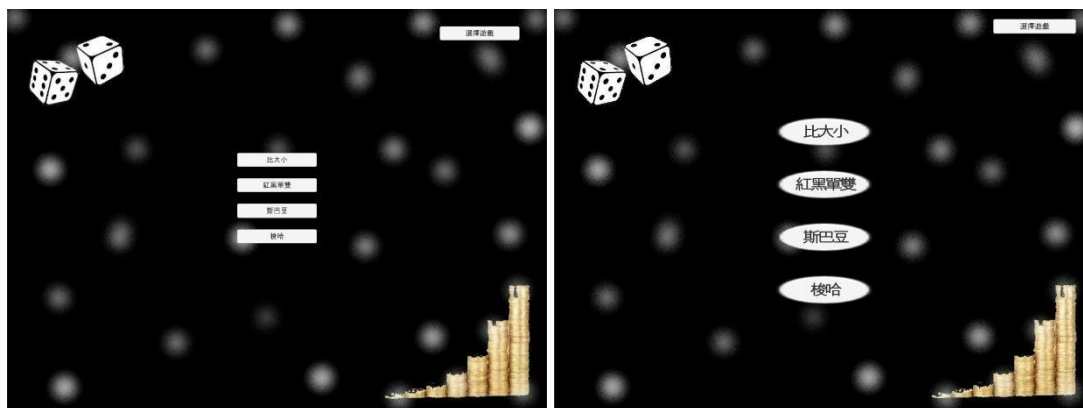


圖5- 18 懷舊畫面設計

原先遊戲畫面是以能夠完成為原則，等到有動作才去製作相關主題，如圖5- 19，之後經過設計及配色改為圖5- 20，最後由於走向主題及美工支援，就完成了如圖5- 21。

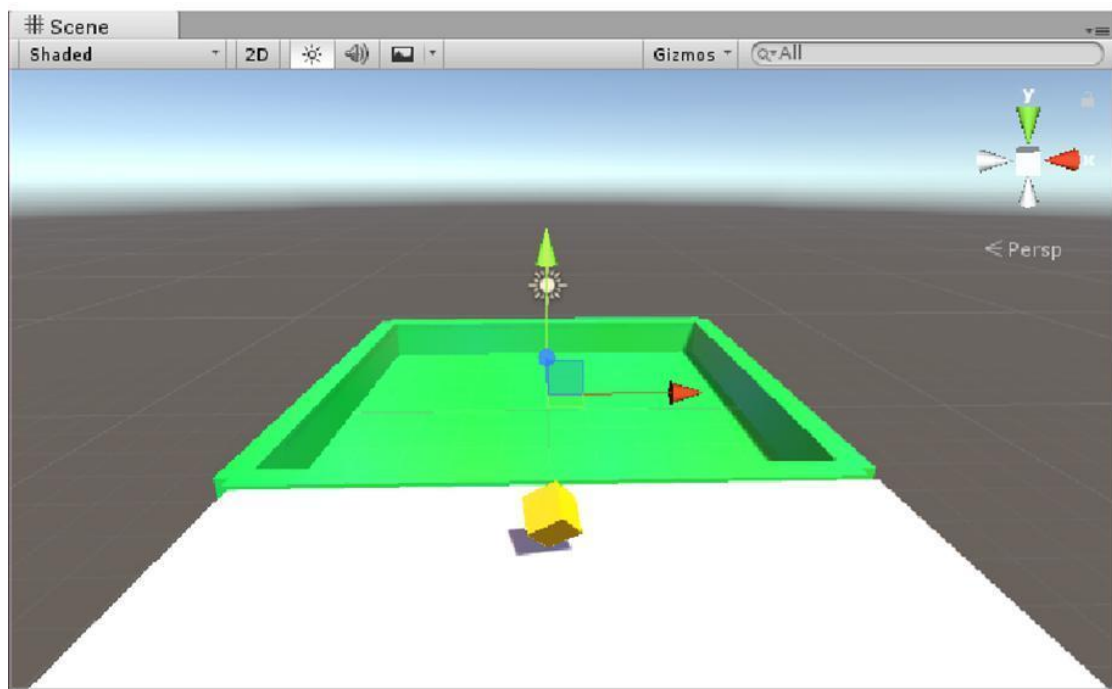


圖5- 19骰子畫面設計

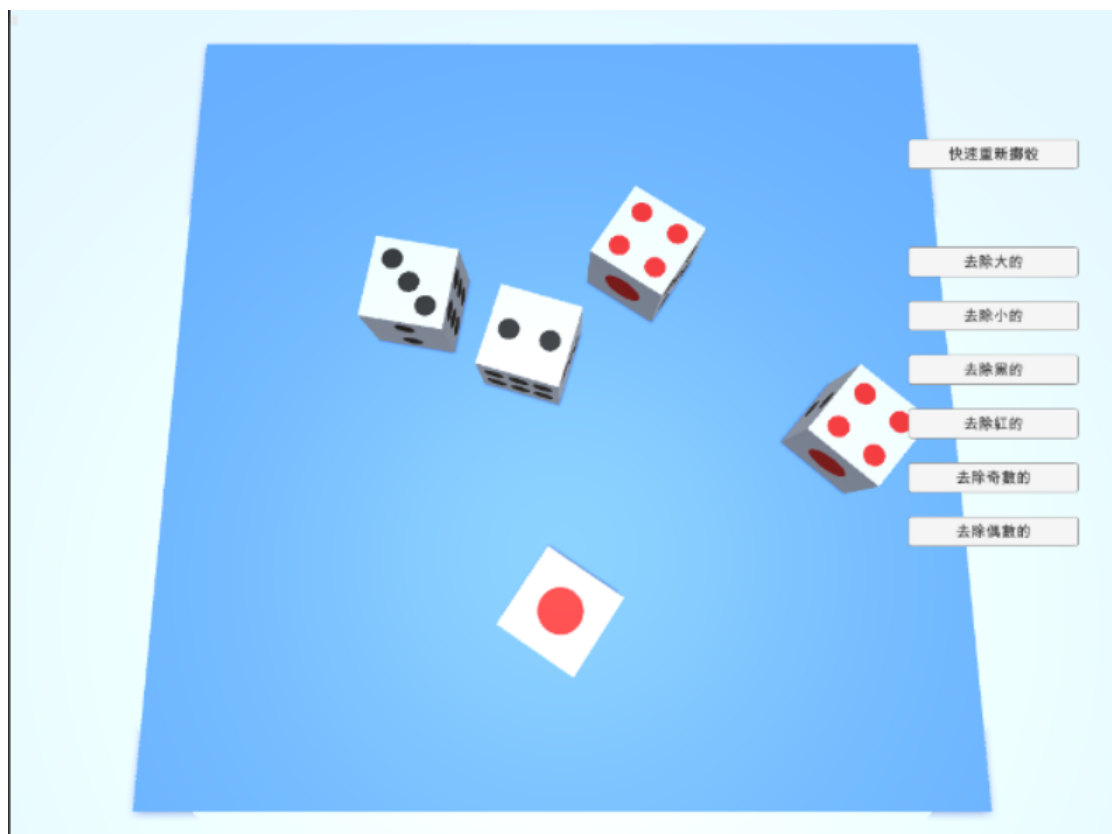


圖5- 20畫面設計

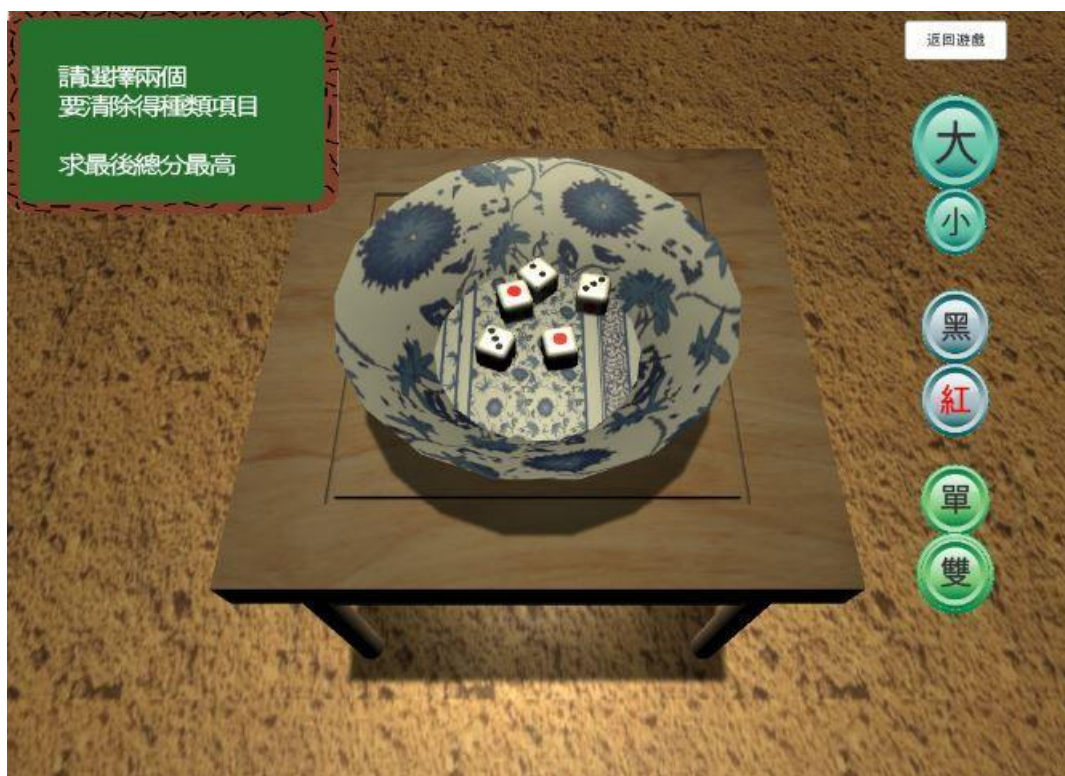


圖5- 21最後懷舊成品

5-4 象棋遊戲

如圖5- 22玩家必須在一分鐘內依照象棋的大小順序去點選象棋，在這模式下是以黑棋先紅棋後，並以由大到小的順序，順序是：將>士>象>車>馬>包>卒>帥>仕>相>俥>傴>炮>兵，每兩秒會有一個象棋變換位置，玩家必須要眼明手快，挑戰玩家的反應力，以越快的速度完成便獲得越高的分數。

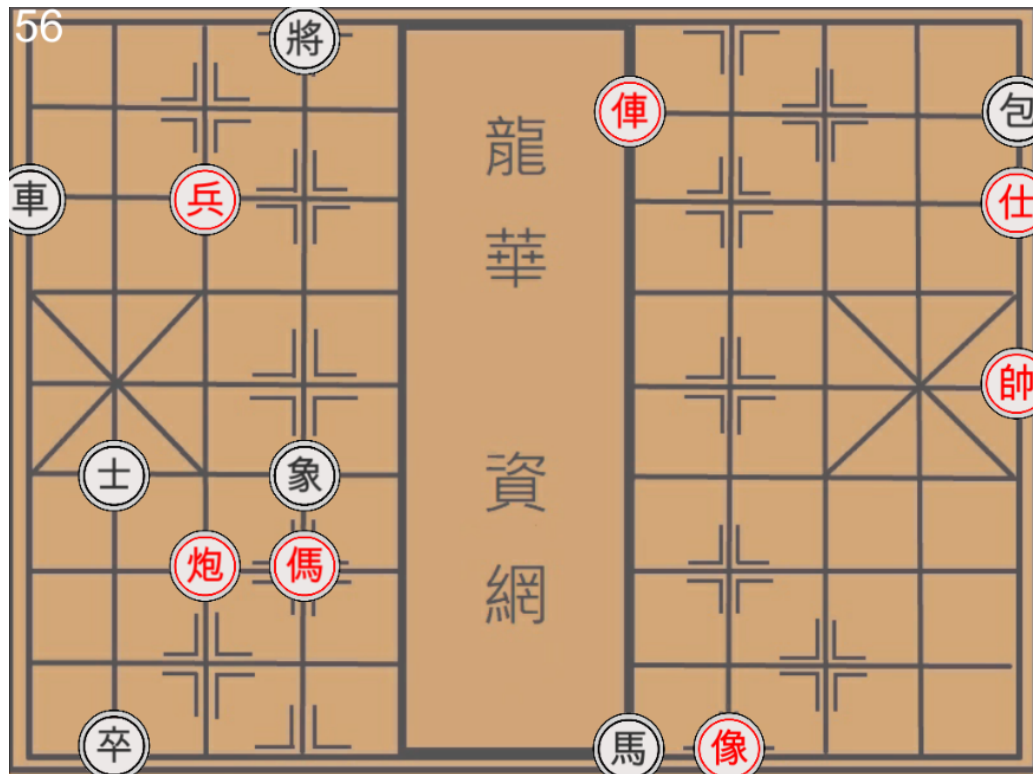


圖5- 22象棋遊戲

第6章、第二關遊戲

6-1 關卡遊戲

第二關為猜燈謎遊戲，地圖俯瞰圖如圖6- 1。

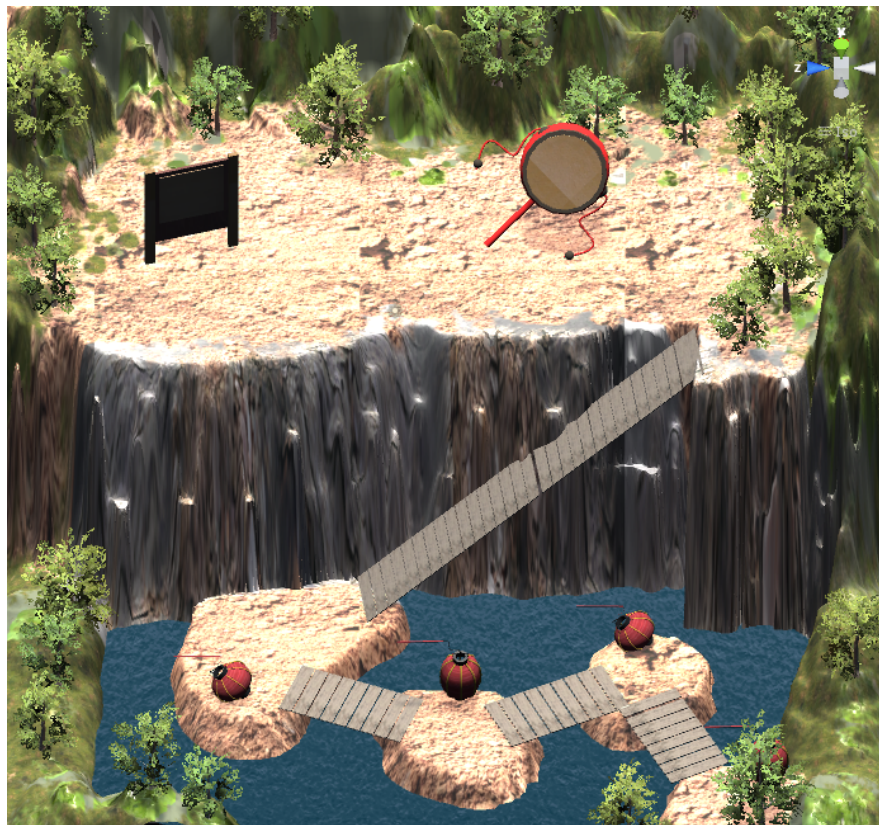


圖6- 1第二關俯瞰圖

遊戲控制方法：

W、A、S、D或Up、Down、Left、Right為上下左右的控制。

ESC為關閉程式。

F3為語音辨識，按住F3可以說出通關密語或謎語，放開F3後便會辨識答案是否正確。

移動時會遇到四個猜燈謎遊戲，題目以亂數出題方式，按住F3說出題目並放開F3，猜對就會出現橋可以通行，當答完題目後爬上高處即可進行圖片問答遊戲，完成題目問答遊戲後繼續行走 碰到木板時會出現遊戲結束字樣。

6-2 猜燈謎遊戲

當進入第二關時，右下方提示字串如圖6- 2會顯示謎語，按住F3說出謎語答案時會出現橋能夠進入下一關，如果答錯則會顯示說出來的字串並且要繼續做答，當秒數到答60秒，則會重新出題目讓玩家做答，當走完四個關卡後則能夠爬到高處如圖6- 3，可以選擇進行圖片問答遊戲繼續進行。



圖6- 2第二關初始

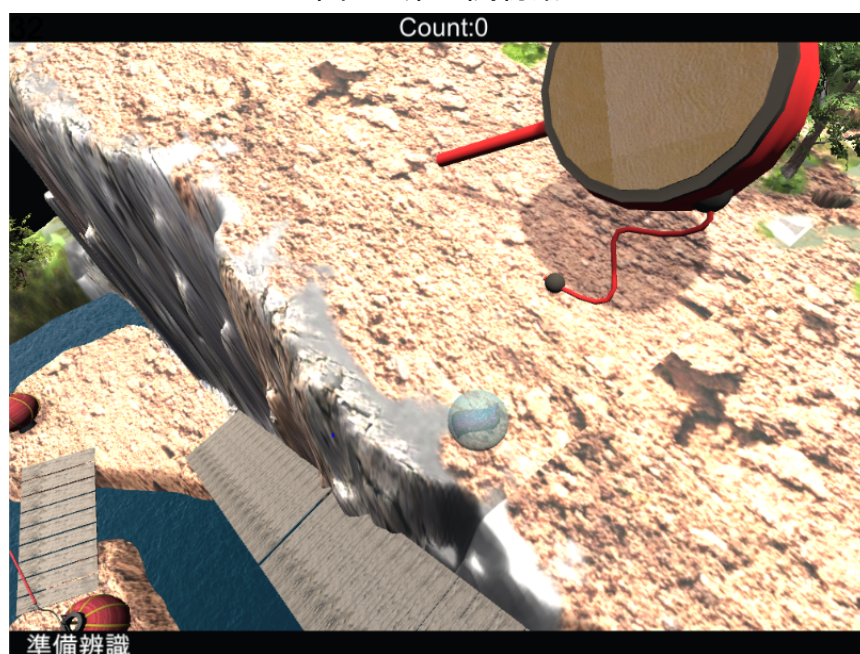


圖6- 3答完爬上高處

6-3 圖片問答遊戲

給予懷舊的題材，但是又需要辨別問題來得分，再回味的同時需要動腦，這就是這關產生的意義，由於老舊畫面現在周遭不易取得，所以在網路上查詢創用CC再找無版權或無商業用途的圖片使用。

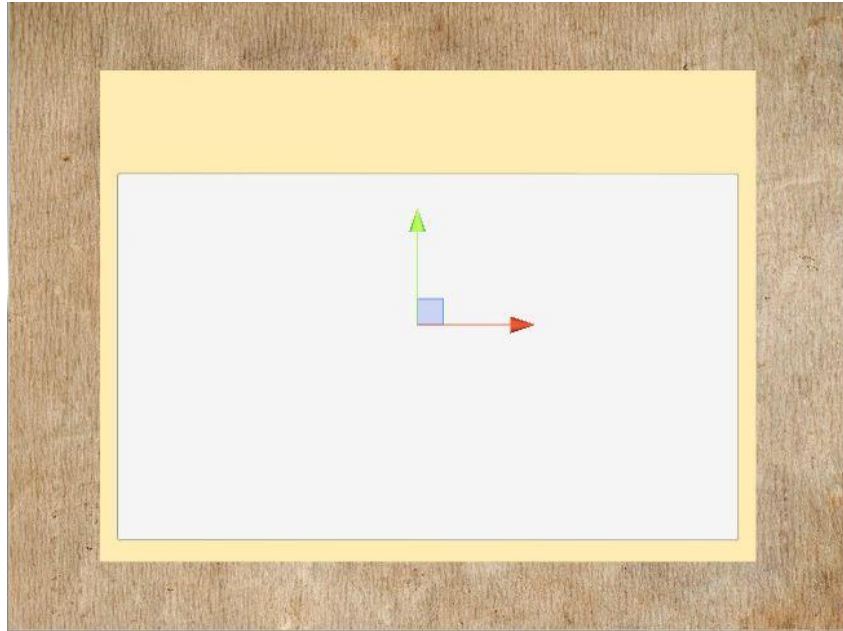


圖6- 4遊戲初始畫面

主畫面分為三層如圖6- 4，底層(最外圍)此區給予老舊木頭材質，中間層(橘色區塊)此區上部分用來顯示問題的文字，其他部分則是放頂層的圖片。遊戲畫面如圖6- 5，在選擇完選項後，在選項中間會有送出，為最後確認後的結果。



圖6- 5遊戲中畫面

第7章、結論

7-1 工作時程

2017/05-2018/06

找到老師之後就開始討論上課期間固定開會及暑假每周晚上固定開會，在分組時就分配好美工組以及程式組，在2017/03/28開始有會議記錄及PowerPoint，以及創立專用的雲端讓我們可以及時看進度及備份；訂好題目之後，前期我們各組都在網路上看範例，然後自己使用Unity做模仿練習，練習到2017年8月之後我們漸漸走向主題，美工組開始從簡易的單色3D建模，變成了加入材質的3D建模，3D物件也逐漸有懷舊復古的感覺，地形也不是原本那種方塊地形，已經變成了材質包拉出來的泥土、山丘、小樹及雜草地形；而程式組不單單只是模仿他人的遊戲，而是有自己所設計出來的小遊戲，直到2018年1月我們開始走向懷舊主題以及延緩老化的機制，漸漸有了新的畫面設計，以及更確切的主題；到了今年的3月，我們開始做遊戲整合，發現到整合後有許多問題，不論是命名的問題，或是遊戲切換物件的問題，原先預計一個禮拜可以整合完畢，卻不知不覺弄了一個月；而文案是從這學期的第10周開始撰寫，對於Word的使用上，我們也稍微小小下了功夫，雖然文筆不是這麼的好，但是老師都會給我們一些修改建議。

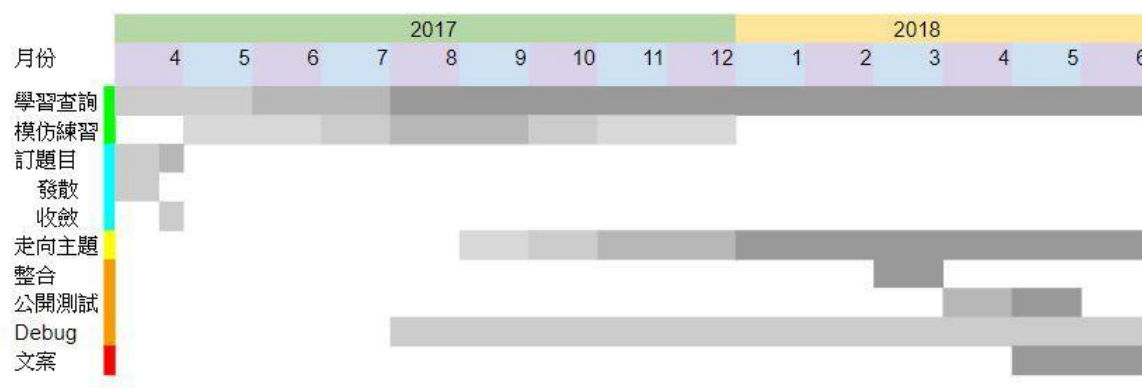


圖7- 1專題行程甘特圖。

7-2 研究結果

以反應力、專注力、運氣、認知能力及思考設計出的這款懷舊遊戲。遊戲內容不複雜，但是能從中細細品嘗，有些關卡會有些難度，但是又不會讓老人家保持放空狀態。

對於我們的成品，或許不足以上架，但是我相信這次的構想，能夠讓大家都更了解老人化社會的趨勢；隨著老年化的趨勢我們這種類型的成品，也會漸漸露出檯面。

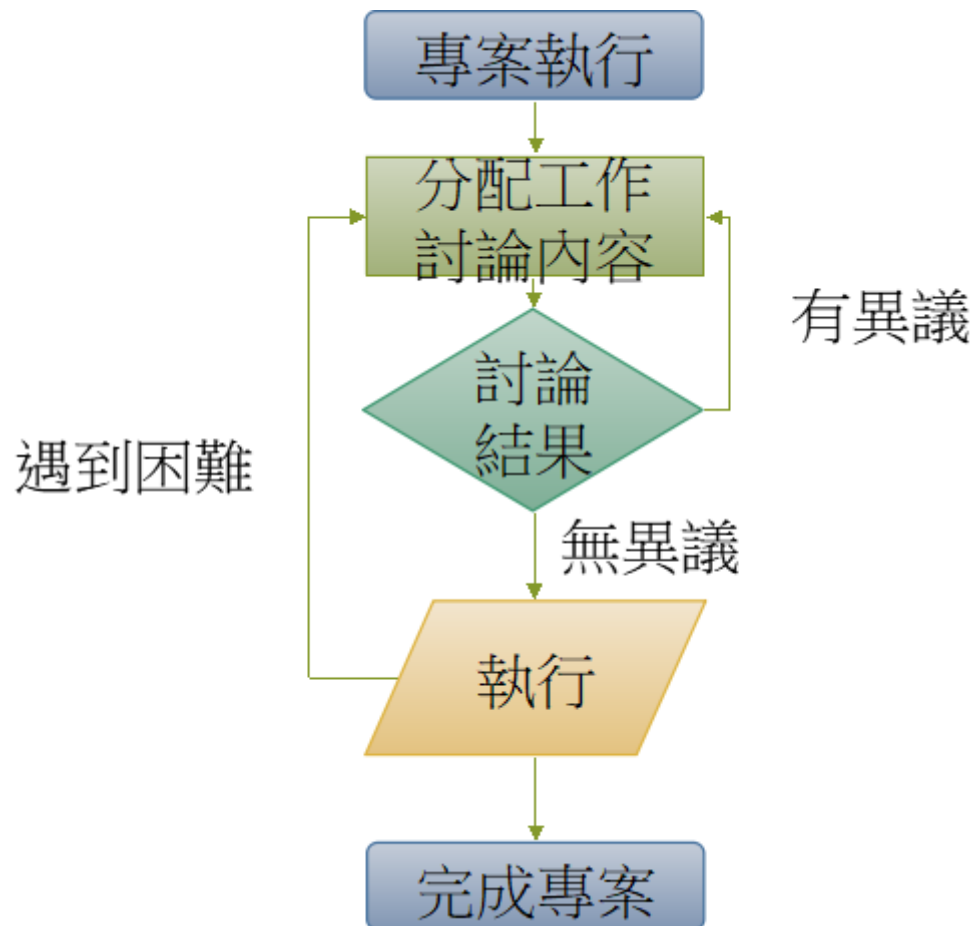


圖7- 2專案基本流程圖。

7-3 專題製作心得

7-3-1 黃暉皓心得

在大二開始做專題的時候懵懵懂懂不會使用Unity，我們一開始自學時模仿網路上的教學範例，並練習語法和物件、模型、播放器、動畫等，過程中會遇到許多問題和技術上的困難，我們也上網自己尋找解決方式或和組員討論並解決困難，到後來我們能夠看影片就知道怎麼做出一樣的小遊戲，並且模仿製做出來做為練習，熟練Unity基本操作後，大三我們開始討論製作我們自己的遊戲，過程中老師指導我們如何想出大家都喜歡而且也想做的題目，到最後我們能夠自己創造出遊戲，以前我們都是玩別人做的遊戲到現在我們能夠寫遊戲給別人玩，我覺得這感覺非常的棒，雖然技術還不夠純熟，沒辦法做到非常好的遊戲但是做出我們自己的遊戲就是最大的喜悅。

7-3-2 張家寶心得

此次的專題是大學以來第一個製作最久的專案與程式，與以往課堂上的作業經驗完全不同，是個非常好的體驗，雖然課堂上學習過C#，但是Unity又是一個完全不同的開發經驗，從零到自己在網路上學習完成，當中有遇到了很多的困難，不過經由老師的意見更改，與將問題除錯完成，也體驗到了與朋友共同做一個專案必須要去配合與解決很多的問題，最後製作出屬於自己的一個遊戲，這對於從小玩遊戲長大的我是非常開心的事情。

7-3-3 楊宏隆心得

在這次的專題過程中，我學到的不僅僅是對於沒使用過的程式Unity，有進一步的了解，而且是能夠在網路上，自己找到方法解決，自己處理事物的能力，也能夠從團隊討論中，發現每個人可能都有不同的想法，但是最重要的是能夠在討論後有個大家都接受的答案，這也是一門說不出的學問，同時也能在與老師的開會之中發現，很多自己與同組想不到的想法，都能夠從老師口中蹦出新的概念，讓我們的專題能夠越來越多樣，而不是死板的完成一項很普通的專題。

研究的過程中，不會只是各自做自己的事情，而是最先會分配內容，以及每周開會討論進度，這些東西看似不太重要，但是其實這就是專題該學到的團隊合作，在平常上課期間，我們學到的都只是學術的知識，但是在專題中，我們能學到的還有自學能力，還有督促及幫助隊友的輔助能力，在團隊不會只有組長要做事，而是大家發現有缺什麼都會提醒，有問題都會一起討論。

7-3-4 張簡信成心得

這次的專題製作的時間很長也學到了很多東西，尤其是使用的軟體都是我從來沒有學過的東西，我必須上網自己找教學跟影片，但是教學往往都是英文，也造就了我在學習上面的困難，但是還是一一克服了，原本我們是要做2D的遊戲使用illustrator繪圖，但是老師覺得2D不夠創新，所以叫我們嘗試製作3D的遊戲，在接觸3Ds Max的時候，我真的覺得很困難，因為上面有一堆功能都是我們從來沒有接觸過了，還有Unity上面也有許多建造模型跟地形的功能，這些都是我以往沒學過的東西，在摸索跟學習的時間往往超過實際操作的時間，一開始我們是以模仿別人遊戲為基礎，做的跟影片內的遊戲一模一樣來增進我們的能力跟熟悉軟體的操作，一開始建造的模型真的是慘不忍睹，越來越上手之後，以往要繪製一兩小時的模型，在後來只要花半小時就可以完成了，在這專題路上沒有

很順利，中間有遇到許多的困境，尤其是除錯跟整合問題，在使用Unity的時候光是模型材質還有地形就發生過很多的問題了，專題上主要是跟朋友之間的溝通以及合作，不只程式跟專案要克服還有團隊裡面的問題，但是完成這個遊戲還是讓我覺得是一件很值得回憶的經驗。

7-4 未來展望

這個遊戲目前有許多小遊戲，都是以喚醒銀髮族回憶還有防癡呆設計的，由於考慮到手續效能問題，所以採用電腦的方式，鍵盤操作上會有比較大的難度，未來會往手機上面發展，增加陀螺儀還有搖桿功能，會使銀髮族更好操作，並能隨時隨地就遊玩到遊戲，我們希望這個遊戲不只是遊戲，是可以讓銀髮族回憶童年並且防止老化，現在年輕人越來越沉迷於3C，對老人疏於陪伴，在未來希望能增加更多玩家，讓年輕人可以在家陪伴老人遊玩，增加年輕玩家與老年玩家之間的話題性，科技越來越發達，希望能以科技的方式幫助銀髮族們。

7-5 注意事項

遊戲是專門設計給銀髮族遊玩的，由於部分遊戲是採用3D畫面建議使用視窗化，視窗化可以改善暈眩感，遊戲有可能會發生3D暈眩症，此症狀會有許多的併發症，像是噁心、頭痛、發燒等，如果發生，請暫時將視線避開螢幕，閉眼休息等狀況緩和許多，再繼續遊玩。

參考文獻

- [1] 國家發展委員會, 中華民國人口推估(105至150年)
https://www.ndc.gov.tw/Content_List.aspx?n=84223C65B6F94D72
- [2] 內政部統計處, 老化指數, <https://www.moi.gov.tw/stat/chart.aspx>
- [3] Shatenstein, Bryna, and Pascale Barberger-Gateau. "Prevention of Age-Related Cognitive Decline: Which Strategies, When, and for Whom?" *Journal of Alzheimer's Disease* 48.1 (2015): 35-53.
- [4] Stern, Yaakov. "Cognitive reserve in ageing and Alzheimer's disease." *The Lancet Neurology* 11.11 (2012): 1006-1012.
- [5] Fratiglioni, Laura, Stephanie Paillard-Borg, and Bengt Winblad. "An active and socially integrated lifestyle in late life might protect against dementia." *The Lancet Neurology* 3.6 (2004): 343-353.
- [6] Niederstrasser, Nils G., et al. "Approaches to cognitive stimulation in the prevention of dementia." (2016).
- [7] Min, Jung-Ah, Chang-Uk Lee, and Chul Lee. "Mental health promotion and illness prevention: a challenge for psychiatrists." *Psychiatry investigation* 10.4 (2013): 307-316.
- [8] Brodziak, Andrzej, et al. "Guidelines for prevention and treatment of cognitive impairment in the elderly." *Medical science monitor: international medical journal of experimental and clinical research* 21 (2015): 585.
- [9] Rebok, George W., et al. "Ten-year effects of the advanced cognitive training for independent and vital elderly cognitive training trial on cognition and everyday functioning in older adults." *Journal of the American Geriatrics Society* 62.1 (2014): 16-24.
- [10] Buschert, Verena, Arun LW Bokde, and Harald Hampel. "Cognitive intervention in Alzheimer disease." *Nature Reviews Neurology* 6.9 (2010): 508.
- [11] Sawami, Kazue, Himeyo Nakagawa, and Tetsuro Kitamura. "Relationship between cognitive function, vascular age and stress." *Int J clinical & case 1* (2017): 4-83.
- [12] Sawami, Kazue. "Intervention for Elderly People to Prevent Stress and Cognitive Decline in Subjects." *Current Opinions in Neurological Science* 1 (2017): 44-52.
- [13] Lazarou, Ioulietta, et al. "International Ballroom Dancing Against Neurodegeneration: A Randomized Controlled Trial in Greek Community-Dwelling Elders With Mild Cognitive impairment." *American Journal of Alzheimer's Disease & Other Dementias*, 32.8 (2017): 489-499.
- [14] Sawami, K., et al. "Verification of Preventive Effect of Dual-Task and N-Back Task-Incorporated Music Therapy against

Dementia." *Neurochem Neuropharm Open Access* 3.115 (2017): 2.

- [15] Hamada, Toshimitsu, et al. "Robot therapy as for recreation for elderly people with dementia-Game recreation using a pet-type robot." *Robot and Human Interactive Communication*, 2008. RO-MAN 2008. The 17th IEEE International Symposium on. IEEE, 2008.
- [16] Chan, Jeanie, Goldie Nejat, and Jingcong Chen. "Designing intelligent socially assistive robots as effective tools in cognitive interventions." *International Journal of Humanoid Robotics* 8.01 (2011): 103-126.
- [17] Agrigoroaie, Roxana M., and Adriana Tapus. "Developing a healthcare robot with personalized behaviors and social skills for the elderly." *Human-Robot Interaction (HRI)*, 2016 11th ACM/IEEE International Conference on. IEEE, 2016.
- [18] Reid, Larry D., Faith E. Avens, and Alicia A. Walf. "Cognitive behavioral therapy (CBT) for preventing Alzheimer's disease." *Behavioural brain research* 334 (2017): 163-177.
- [19] Gamberini, Luciano, et al. "A game a day keeps the doctor away: A short review of computer games in mental healthcare." *Journal of CyberTherapy and Rehabilitation* 1.2 (2008): 127-145.
- [20] Wang, Ping, et al. "Age-Related Cognitive Effects of Videogame Playing Across the Adult Life span." *Games for health journal* 6.4 (2017): 237-248.
- [21] Brox, Ellen, et al. "Exergames for elderly: Social exergames to persuade seniors to increase physical activity." *Pervasive Computing Technologies for Healthcare (PervasiveHealth)*, 2011 5th International Conference on. IEEE, 2011.
- [22] Garcia, Jaime Andres, et al. "Exergames for the elderly: Towards an embedded Kinect-based clinical test of falls risk." *HIC*. 2012.
- [23] Gerling, Kathrin Maria, Jonas Schild, and Maic Masuch. "Exergame design for elderly users: the case study of SilverBalance." *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology*. ACM, 2010.
- [24] Lamothe, Claudine JC, Rolinde Alingh, and Simone R. Caljouw. "Exergaming for elderly: Effects of different types of game feedback on performance of a balance task." (2012): 103-107.
- [25] Chen, Shang-Ti, Yueh-Guey Laura Huang, and I-Tsun Chiang. "Using somatosensory video games to promote quality of life for the elderly with disabilities." *Digital Game and Intelligent Toy Enhanced Learning (DIGTEL)*, 2012 IEEE Fourth International Conference on. IEEE, 2012.

- [26] Shang-Ti, C. H. E. N., et al. "Effects of improvement on selective attention: Developing appropriate somatosensory video game interventions for institutional-dwelling elderly with disabilities." TOJET: The Turkish Online Journal of Educational Technology 11.4 (2012).
- [27] De Moraes, Wagner O., and Nicholas Wickström. "A serious computer game to assist Tai Chi training for the elderly." Serious Games and Applications for Health (SeGAH), 2011 IEEE 1st International Conference on. IEEE, 2011.
- [28] Orange, Joseph B., and Angela Colton-Hudson. "Enhancing communication in dementia of the Alzheimer's type." Topics in Geriatric Rehabilitation 14.2 (1998): 56-75.
- [29] Google, CLOUD SPEECH API,
<https://cloud.google.com/speech/?hl=zh-tw>
- [30] UI畫面的固定比例
<https://godstamps.blogspot.tw/2015/04/unityugui.html>
<https://jerrard-liu.blogspot.tw/2015/05/canvas1.html>
- [31] Unity Scripting API
<https://docs.unity3d.com/ScriptReference/index.html>
- [32] 骰子轉動汲取數值
<https://www.youtube.com/watch?v=LVKqTzK9-AI&t=619s>
- [33] Unity Scripting API: <https://docs.unity3d.com/ScriptReference>
- [34] Google Cloud Translate API:
<https://cloud.google.com/translate/?hl=zh-tw/>