內容

[一、簡介 3](#_Toc42780607)

[動機 4](#_Toc42780608)

[分工 4](#_Toc42780609)

[遊戲說明 5](#_Toc42780610)

[遊戲圖形 7](#_Toc42780611)

[遊戲音效 9](#_Toc42780612)

[程式設計 10](#_Toc42780613)

[程式架構 11](#_Toc42780614)

[程式類別 12](#_Toc42780615)

[程式技術 15](#_Toc42780616)

[結語 17](#_Toc42780617)

[間題及解決 18](#_Toc42780618)

[時間表(不含上課時間) 21](#_Toc42780619)

[貢獻比例 23](#_Toc42780620)

[自我檢核表 23](#_Toc42780621)

[收獲 25](#_Toc42780622)

[心得及感想 26](#_Toc42780623)

[附錄 27](#_Toc42780624)

# 一、簡介

## 動機

本遊戲原型為一款解謎類RPG魔塔，日語為魔法の塔，也稱50層魔塔（遊戲有50層地圖的原因）。是由日本Caravan Software設計師渡部直於1996年製作，發行於MS-DOS作業系統。是解謎類RPG的始祖。

魔塔（魔法の塔）操作簡單易明、有故事劇情、可以跟NPC互動、多地圖、遊戲設定多變、有陷阱關卡、怪物種類豐富、怪物的數據多變豐富、有很大卡關的機會、遊玩時間持久等，到現今RPG都有這此元素，所以喜愛謎類RPG的亞洲玩家，都會知道這個遊戲。

此遊戲直到現時，仍然有很多玩家包括我們在外，都很熱愛這個遊戲。現在有大量玩家重新設計遊戲地圖，出現了不同的改版。而且我們其中一個組員吳軍樂，更斷續遊玩這個遊戲5年，都還沒過關。所以對這個遊戲有一份執著和熱情，加上我們都對此遊戲有一定程度的了解，所以決定實做此遊戲作為本學期的目標。

## 分工

吳軍樂：程式編寫、整體遊戲設計、程式架構設計、音效設計、素材準備

劉學逸：程式編寫、遊戲測試、程式漏洞修補、動畫設計、素材處理

## 遊戲說明

遊戲簡介：

原版魔塔是50層地圖的版本，最快平均破關時間（不包括卡關和思考時間）為2小時，因為遊玩時間過長，所以我們選擇採用一個新手的改版地圖樣（21層魔塔）。

遊戲玩法及規則：

遊戲以拯救被囚禁於塔內最高層的公主為首要目標。但此版本的塔經歷過原版的戰爭（50層魔塔）後救援失敗後，魔塔倒塌剩21層，加上時間已久，裡面的增益道具都失效了，而且守著這座塔的魔法傀儡和怪物的能力也減弱了不少，是復仇救出公主的好機會，可以比較簡單地救出公主。塔裡有大量上鎖的門，玩家要透過撿取地上的鑰匙，或者打敗魔法傀儡或怪物賺取金幣，然後拿金幣跟一樣被困在魔塔裡的平民賺買鑰匙，開啟上鎖的門，才能通往下一層，從而到逹頂層救出公主。

本地圖有3個Boss的關卡，位於第10層、第15層和第20層。玩家打敗第10層Boss得以找到通往第11層的樓梯、打敗第15層Boss得以賺取大量金幣，第20層為終極Boss，打敗後得以開啟通往得21層的門，救出公主。

第2層為副本關卡（平民囚牢），可以用金幣買鐵鑰匙救出平民，有機會獲取一些重要物資，為打開第11層上鎖的白色門。

當人物血量不足應對怪物時，會有提示表示能力不足，但不會死掉，玩家可以撿取武器或裝備、拿金幣透過商店增加能力值，重新裝備人物逆轉殘局。否則會卡關收場。

當進入位於第10層和第20層封閉式Boss關卡階段時，人物會被鎖在一個空間，不能逃離，直到裡面的Boss死亡，挑戰Boss成功時，上鎖的門才得以打開。如果人物血量不足應對Boss階段怪物時，人物會死亡，然後轉到Game Over畫面，遊戲失敗，遊戲結束，跳回主畫面。

玩家成功到達第21層，救出公主，並跟公主對話後，最後會轉到Win Game畫面，遊戲勝利，遊戲結束，跳回主畫面。

操作、特殊功能及密技：

人物操作（鍵盤）：

|  |  |
| --- | --- |
| [↑]：往上行走 | [↓]：往下行走 |
| [←]：往左行走 | [→]：往右行走 |

對話框操作（鍵盤）：

|  |  |
| --- | --- |
| [Space Bar]：功能鍵（退出／繼續） | [N]：是 |
| [Y]：否 | [S]：遊戲統計功能鍵 |

密技（鍵盤）：

|  |  |
| --- | --- |
| [A]：畫面轉至上一層 | [D]：畫面轉至下一層 |
| [U]：得到1個鐵鑰匙 | [I]：得到1個黃鑰匙 |
| [O]：得到1個藍鑰匙 | [P]：得到1個紅鑰匙 |
| [Q]：得到100點生命值 | [W]：得到50點防禦力 |
| [E]：得到50點攻擊力 | [C]：得到50個金幣 |

## 遊戲圖形

門：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\d2.png |  |  |  |  |

有動畫的貼圖：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |

怪物及NPC：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |
|  |  |  | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e9.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e10.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e11.png |  |
|  |  |  |  |  |  |  |
|  |  |  | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e8.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e13.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e14.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e16.png |
| C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e17.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\e18.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\n1.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\n2.png | C:\Users\Sam\AppData\Local\Microsoft\Windows\INetCache\Content.Word\n3.png |  |  |

無動畫的貼圖：

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |

道具

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |

人物：



## 遊戲音效

|  |  |
| --- | --- |
| bgm1.mp3 | 主要遊戲背景音樂 |
| bgm2.mp3 | 打Boss時之遊戲背景音樂 |
| attack.mp3 | 攻擊怪物音效 |
| door.mp3 | 開啟黃、藍、紅、白門音效 |
| item.mp3 | 拾取物品或道具之音效 |
| gate.mp3 | 開啟鐵門之音效 |
| floor.mp3 | 走樓梯到下一層樓之音效 |
| tryOpen.mp3 | 嘗試開啟門，但失敗之音效 |

# 程式設計

## 程式架構

## 程式類別

|  |  |  |
| --- | --- | --- |
| 類別名稱 | .js 檔行數 | 說明 |
| animationWall | 57 | 動畫牆的動畫 |
| bat | 147 | 蝙蝠怪物物件／動畫及功能 |
| blueKeyItemInventory | 40 | 藍色鑰匙背包計數 |
| bombMan | 122 | 玩家物件及移動動畫 |
| consoleBoard | 58 | 遊戲資訊欄顯示及功能 |
| constants | 76 | 物件編碼代碼常數 |
| door | 105 | 門物件／動畫及功能 |
| doorPickAxe | 59 | 開門鎬物件／動畫及功能 |
| femaleLevel | 74 | 女生角色地圖資源檔讀取及數據處理 |
| gameOver | 92 | 遊戲結束頁面 |
| gems | 53 | 寶石道具物件／動畫及功能 |
| genderMenu | 144 | 選擇角色頁面 |
| guard | 169 | 衛兵怪物物件／動畫及功能 |
| ironKeyItemInventory | 40 | 鐵色鑰匙背包計數 |
| items | 76 | 裝備道具物件／動畫及功能 |
| keys | 68 | 鑰匙道具物件／動畫及功能 |
| loadGame | 118 | js檔執行函式載入 |
| mainGame | 18 | 遊戲level定義 |
| maleLevel | 70 | 男生角色地圖資源檔讀取及數據處理 |
| map | 1618 | 遊戲的核心，處理遊戲的所有事件、畫出貼圖及功能總合處理 |
| mapTile | 40 | 地板物件 |
| MonsterHpDisplay | 25 | 怪物血量顯示顏色判定功能 |
| myAbout | 157 | 關於遊戲頁面 |
| myMenu | 161 | 遊戲主頁面 |
| npc | 143 | NPC物件／動畫及功能 |
| npcMessageBoard | 56 | NPC對話框物件 |
| playerState | 60 | 玩家狀態數據 |
| potions | 54 | 藥水道具物件／動畫及功能 |
| priest | 145 | 法師怪物物件／動畫及功能 |
| redKeyItemInventory | 40 | 紅色鑰匙背包計數 |
| settingBoard | 28 | “S”鍵統計框物件 |
| showLevelBoard | 38 | 地圖樓層數框物件 |
| skeleton | 183 | 骷髏人怪物物件／動畫及功能 |
| slime | 223 | 史萊姆怪物物件／動畫及功能 |
| specialEnemys | 194 | 特殊怪物物件／動畫及功能 |
| Squid | 253 | 大烏賊Boss怪物物件／動畫及功能 |
| stage | 56 | 樓梯物件／動畫及功能 |
| stoneWall | 57 | 牆物件／動畫及功能 |
| store | 9 | 商店價錢計算功能 |
| terrain | 477 | 地圖資源檔 |
| winGame | 106 | 遊戲勝利畫面 |
| yellowKeyItemInventory | 40 | 黃色鑰匙背包計數 |
| zombie | 143 | 獸人怪物物件／動畫及功能 |
| **總行數** | 5892 |  |

## 程式技術

1. 地圖原始資料儲存
   * 把地圖所有的物件編碼，然後使用一個物件（terrain.js），以物件的編碼來存儲全部21張地圖的所有物件（地板、牆、道具、怪物）的原始位置，還有儲存人物初始位置，是遊戲地圖的原始資源物件。
2. 地圖換層
   * 整個遊戲在遊玩階段，都是在同一個level裡進行。換地圖時在同個畫面中重畫地圖，以實現地圖換層。更換樓層的實做是從物件（terrain.js）中讀取所有樓層的物件數據（物件類別、位置等），然後放在一個陣列（map）裡。而到該樓層時，會從陣列（map）裡讀取該層的數據，從而生成和畫出該層物件，然後把物件存在陣列（mapArray）裡。當該樓層的物品有更變時，會將更改陣列（map）裡的數值。所有樓層數據都會保留，直到程式關閉為止。
3. 更換地圖的記憶體管理
   * 為了節省記憶體存儲空間，程式碼會盡量減少new新的物件。
   * 在更換地圖時，會重新從陣列（map）裡讀取該層的編碼數據，然後清空陣列（mapArray），重新new出新的物件。
   * 為了減少浪費記憶體資源，會重覆使用有用而且相同的物件。在清空陣列時，會判定下一層要輸出的地圖中的數據，是否有跟現在有的陣列（mapArray）裡相同的物件。如果有的話會保留不會清空。保在下一層數輸出時使用。重新利用物件。
4. 人物狀態及人物持有物品
   * 遊戲的人物物件有存當前血量、防禦力、攻擊力和剩餘金幣。更有儲存人物持有鑰匙數量的資料。
5. 打怪系統
   * 所有怪物物件都存有血量、防禦力、攻擊力和掉落金幣的變數。然後依照公式來算出回合數、人物扣血量的數據。
6. NPC聊天系統及商店系統
   * 使用畫出圖層的方法，顯示對話框、然後針對對話框會使用到的按鍵，實做人物可以跟NPC聊天、跟NPC買東西等功能。商店系統可以使用公式算出購買價格，然後透過對話框跟玩家互動，實做購物的功能。

# 結語

## 間題及解決

1. 未使用及學習過JS
   * 在網路上自學JS的語法、架構和用法，再實作在專題裡
2. JavaScript版本Game Framework的說明文件缺少
   * 一開始是參考範例程式，學習他的語法來實作。但到最後有些功能在範例程式中寫的方法有點亂或者沒有用到有關的功能，例如是BGM的宣告、動畫、按鈕功能等，所以直接看src裡的原始碼來自我理解和使用。
3. 遊戲人物和物品大小單位的定義
   * 遊戲的顯示像素為1920 x1080，而遊戲是以一格為單位，整個版面有26 x 13格。為了固定物件的大小，我們有定義一個(objectX, object)，這樣就能定義了物件的大小。所以我們整個遊戲都以「格」來做單位，而不是以像素來定義
4. JavaScript版本的Game Framework裡缺少或者找不能相關的功能來做想做的功能
   * 例如我們的遊戲的人物能在21層地圖中自由穿梭，遊處會保存21層地圖的所有物件。在Game Framework裡有. goToLevel的功能，但這個功能會new出一個新的Level出來，而導致上一層的東西沒有保存，不符合我們的遊戲架構。所以我們使用在同一個level裡，以重新畫圖的方法來處理多層地圖的穿梭，把地圖裡的所有物件放一個陣列裡，然後可以隨時提取使用。但這個做法也引起了另一問題。
5. 程式運行中會面臨記憶體不足
   * 上面提及說我們把21張地圖裡的所有物件存在一個陣列裡。一張地圖有26 x 13個物件，這樣導致我們的遊戲有嚴重的記憶體不足問題。所以我們選擇把物件以整數來編碼，然後一個陣列存了21張地圖的編碼，一個陣列是存當前層數的物件。這樣就可以大幅節省記憶體存儲空間。但這樣也再引發了另一問題。
6. JavaScript記憶體釋放有缺陷，導致遊戲有機會記憶體溢出問題。
   * JavaScript的記憶體釋放不同於C++。據這次實驗的經驗發現，在JS的環境下，是沒有一個明文的記憶體釋放的寫法。Google Chrome的環境下，當一個變數或物件，它的值是null或undefined時，在一段時間裡沒有用到當個變數或物件時，而且是在一個不規則的時間，不規定的情況下，Google Chrome才會把這個記憶體釋放出來。所以這樣JS的程式必然會有記憶體溢出問題。釋放記憶體的最有效方法是把Google Chrome給關掉，記憶體就會釋放。因此最佳的解決方法是做好記憶體管理，減少浪費使用不必要的記憶體，把程式優化是目前最佳的解決方法。
7. 記憶體管理
   * 有以上的經驗後，除了做了第5項，避免一次new太多的物件之外，我們的程式更要盡量減少new新的物件：第一、在每一次畫新地圖時，都會檢查有沒有相同的物件，從而重用上一層地圖共有的物件。第二、在必要時在物件上宣告其他物件都有的東西。例如地板、怪物都是一個物件，當打死怪物後顯示地板。如果用正常的物件時，當打死怪物後會把怪物的物件delete，再new地板物件。但在JS的話這樣new了一個新的物件，就會有記憶體不足或溢出問題。所以我們會打地板也宣告在怪物裡，這樣可以只改圖片，不改物件。然後等換地圖時，再統一考慮要不要把怪物的物件null掉，new成地板的物件。否則下一地圖也有用到同一個怪物的物件，就會重用物件，以記憶體舒緩記憶體的問題。

## 時間表(不含上課時間)

|  |  |  |  |
| --- | --- | --- | --- |
| 週次 | 吳軍樂 | 劉學逸 | 說明 |
| 0 | 18小時 | 7小時 | 做出牆壁物件、地板物件和第1層的地圖。人物可以自由走路，不會穿牆。 |
| 1 | 24小時 | 4小時 | 做出多層地圖、門物件、鑰匙物件、用密技切換上層下層地圖。修改人物在換地圖後不能移動的Bug。 |
| 2 | 7小時 | 6小時 | 加入道具物件，增加怪物物件、記憶體管理。 |
| 3 | 8小時 | 5.5小時 | 完成所有怪物物件。加入訊息欄顯示玩家對戰相關訊息。修改貼圖無法正常顯示。 |
| 4 | 7小時 | 8小時 | 加入NPC物件。排好每個地圖所有的物件。 |
| 5 | 5小時 | 3.5小時 | 加入商店物件、UI物件與UI功能。 |
| 6 | 4小時 | 4小時 | 加入大烏賊boss、完成編排所有聊天NPC、完善聊天NPC對話框功能。 |
| 7 | 12小時 | 7小時 | 加入背景音樂和音效。 |
| 8 | 7小時 | 4小時 | 增加動畫效果。 |
| 9 | 5小時 | 6小時 | 增加目錄和死亡畫面、結束畫面。 |
| 10 | 5小時 | 4小時 | 加入怪物扣血量顯示功能。 |
| 11 | 5小時 | 4小時 | 增加方使玩家的功能：顯示地圖能賺多少金幣。 |
| 12 | 5小時 | 5小時 | 增加依照怪物扣血量顯示不同的字體顏色，優化遊戲，測試程式、記憶體管理。 |
| 13 |  |  | 增加選擇角色（男／女）功能、遊戲關於的頁面。 |
| 14 |  |  |  |
| 15 |  |  |  |
| 合計 |  |  |  |

## 貢獻比例

吳軍樂：50%

劉學逸：50%

共100%

## 自我檢核表

|  |  |  |  |
| --- | --- | --- | --- |
|  | 項目 | 完成否 | 無法完成的原因 |
| 1 | 解決Memory leak | ☑已完成 □未完成 |  |
| 2 | 自定遊戲 Icon | ☑已完成 □未完成 |  |
| 3 | 全螢幕啟動 | ☑已完成 □未完成 |  |
| 4 | 有About晝面 | ☑已完成 □未完成 |  |
| 5 | 初始畫面說明按鍵及滑鼠之用法與密技 | ☑已完成 □未完成 |  |
| 6 | 上傳setup / apk / source 檔 | ☑已完成 □未完成 |  |
| 7 | 報告字型、點數、對齊、行 距、頁碼等格式正確 | ☑已完成 □未完成 |  |
| 8 | 報告封面、側邊格式正確 | ☑已完成 □未完成 |  |
| 9 | 報告封面、側邊格式正確 | ☑已完成 □未完成 |  |
| 10 | 報告附錄程式格式正確 | ☑已完成 □未完成 |  |

## 收獲

吳軍樂：

在這次的專案中，我們使用HTML5的環境做開發。而寫的語言是JavaScript。HTML5只需要用瀏覽器就可以編譯執行（例如這次使用Google Chrome），是一個蠻方使的語言，也讓我們在這次開發時，少了很多有關編譯器的問題。但同時，因為JavaScript不像是C++等語言有編譯器來做debug和報錯，如果程式有語法錯誤，程式會編譯不了，而且會顯示一定跟語法錯誤有關的訊息。但JavaScript是跑到哪，編譯到哪。當出現錯誤時，往往都是因為程式抓不到值才會報錯，而且不會有語法錯誤的提示。所以我們要用土法debug法，用console.log()放在程式的不同位置，然後運行程式，去判定程式跑到哪裡會出現錯誤。這樣也讓我們每次有錯誤時，debug時因為都在找哪裡有錯，而導致浪費很久的時間。但當做久了以後，有了經驗，就可以預測到哪邊有錯誤，debug的時間也變短了，我們的進度也加快了很多。

還有就是學到怎樣做記憶體管理。JavaScript非常消耗記憶體，記憶體不足或者泄漏問題的嚴重性都比其他語言大，所以我們不得不做記憶體管理。瀏覽器中也缺少記憶體使用的顯示面板。所以我們學到了在工作管理員中查看記憶體的使用狀況，來判定是否有記憶體泄漏，再想辦法做減少記憶體消耗的措施，讓程式能穩定地跑。

## 心得及感想

吳軍樂：

HTML5的執行編譯環境，和JavaScript的語法，都跟以前寫C、C++、python等語言有很大差別。例如是程式會跑到一半的時候突然報錯、程式越跑越慢等情況，在以前都沒有遇過，要花時間去找去查怎麼去解決，挑戰性高之餘，也能學到自我學習。

另外在JavaScript編程的感想，我覺得JavaScript有優點也有缺點。優點就是JavaScript跑到哪編譯到哪，而且它幾乎沒有資料型態，單是這個優點，就可以讓我們做出可以把一個不同類別的物件，push到同一個陣列裡，然後就把陣列當成是一個容器，可以把任何物件都push進去，有需要的時候可以指到某一個位置抓值，在放到適合的function，就不會有錯誤。讓我們可以把一層地圖的所有物件都放在同一個陣列裡，非常方便。另外JavaScript的語法自由，function可以任意宣告，連一個var或者一個變數，都可以是一個function。就可以實現到引用變數時，那個變數就會呼叫一次function重新算值，可以讓程式更有直視感。

缺點就是因為太自由了，沒有一個有規則，固定的語法，所以有錯誤時難以debug。還有就是記憶體的問題，我們有迫切性去做記憶體管理，也讓我有機會去學習把程式寫得更好。

附錄