

國立臺北科技大學

2020 Spring 資工系物件導向程式實習

期末報告

HELLTAKER



第 32 組

108590013 郭建麟

108590040 林誠祐

目錄

一、 簡介	3
1. 動機	3
2. 分工	3
二、 遊戲介紹	4
1. 遊戲說明	4
2. 遊戲圖形	4
3. 遊戲音效	6
三、 程式設計	7
1. 程式架構	7
2. 程式類別	8
3. 程式技術	8
四、 結語	10
1. 問題及解決方法	10
2. 時間表	10
3. 貢獻比例	11
4. 自我檢核表	11
5. 收穫	11
6. 心得、感想	12

一、 簡介

1. 動機：

HELLTAKER是一款動腦闖關並體驗劇情的免費遊戲，當初會選這款遊戲是因為裡面的關卡設計非常有挑戰性、角色非常可愛、劇情也很棒，雖然我們沒有做的遊戲裡沒有加入劇情，但是遊戲的體驗非常好，因此我們才會選擇製作這款遊戲。

2. 分工：

郭建麟：負責主要的程式碼撰寫、邏輯分析、演算法撰寫、以及撰寫報告。

林誠祐：協助程式碼撰寫、尋找專案所需的音效和圖片等素材、及撰寫報告。



二、 遊戲介紹：

1. 遊戲說明：

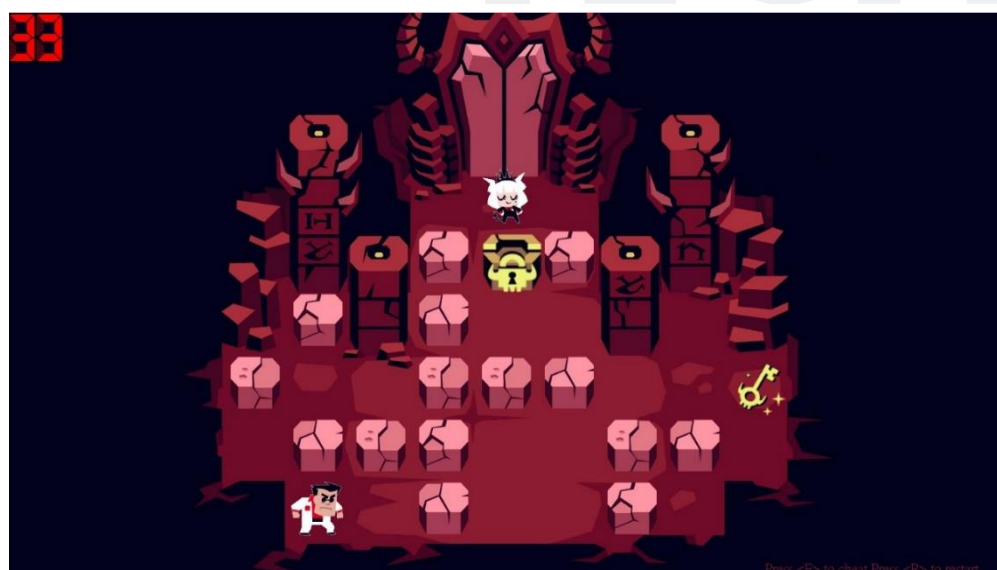
本遊戲主要以鍵盤的上下左右鍵或 W,A,S,D 鍵進行角色移動，遊戲規則為控制主角並透過解謎的方式移動至公主的所在位置即可過關，每個關卡有各自的移動步數限制，步數用盡會重製關卡，地圖中也會有障礙物、敵人、鎖阻礙主角通關，主角必須擊殺敵人、移動障礙物、取得鑰匙解開鎖，並且在消耗最少的步數下到達公主的所在地，特殊功能為按R鍵可重新此關卡，密技為按E鍵可將障礙物及敵人消失。

2. 遊戲圖形：

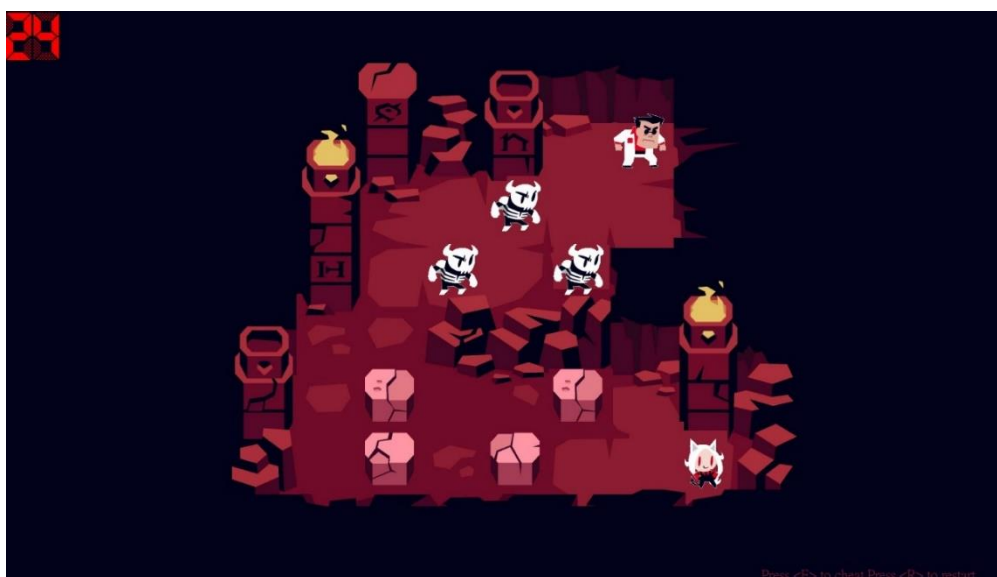
遊戲開始畫面：



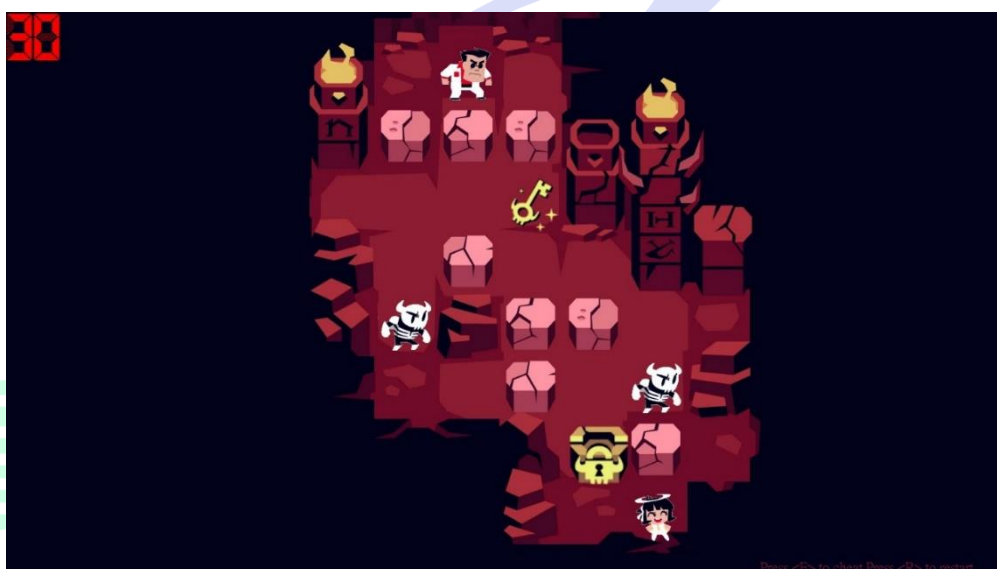
第一關遊戲畫面：



第二關遊戲畫面：



第三關遊戲畫面：



Game over 畫面：

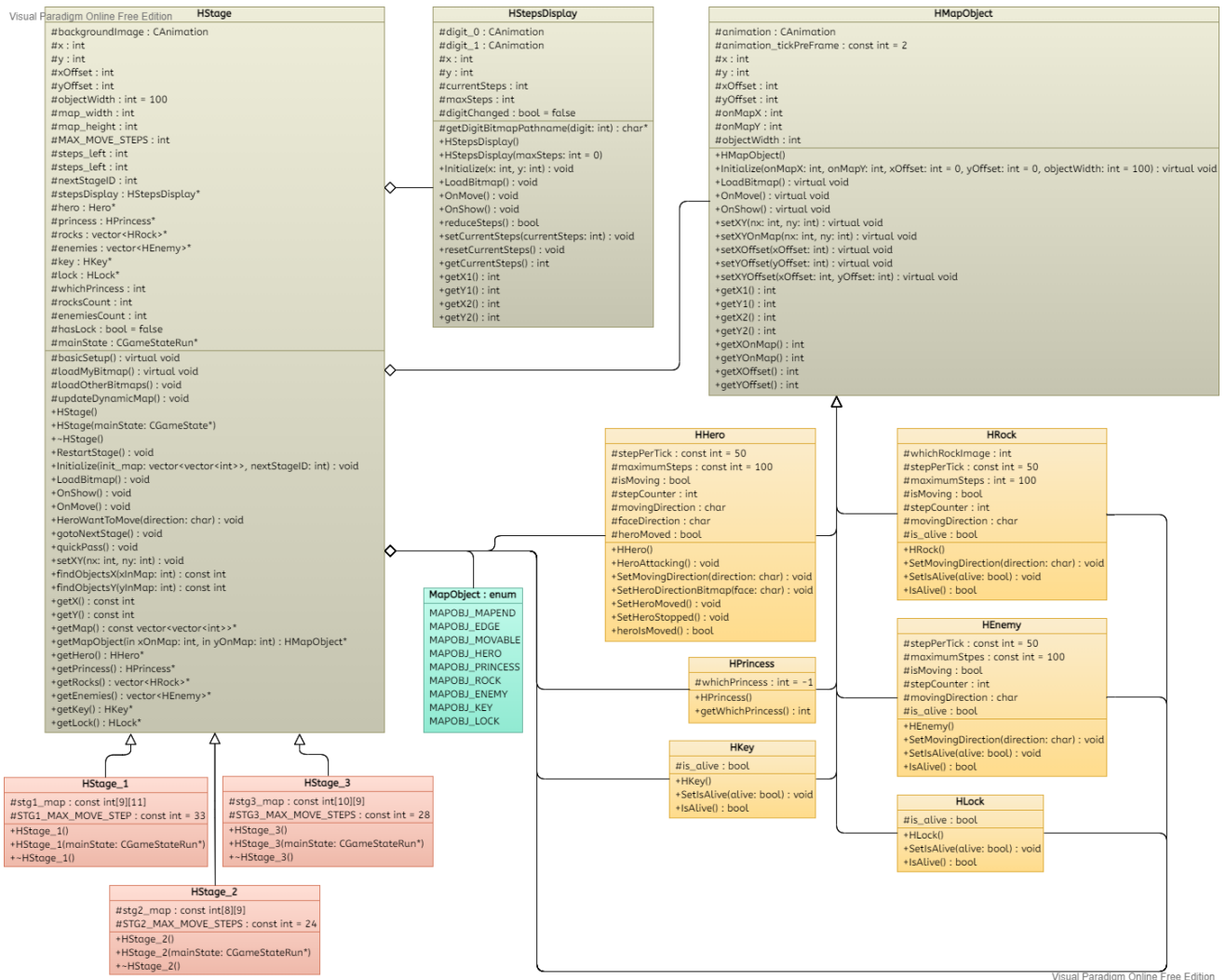


3. 遊戲音效：

- bgm1.mp3：遊戲背景音樂
- character_move_01.wav：主角移動音效
- stone_kick_02.wav：主角踢石頭的音效
- stone_move_01.wav：石頭移動
- enemy_kick_02.wav：敵人移動
- enemy_die_02.wav：敵人死亡
- key_pick_up_01.wav：撿起鑰匙
- door_opening_01.wav：打開鎖
- door_closed_kick_02.wav：被鎖阻擋的音效
- gameover.mp3：遊戲結束音樂

三、 程式設計

1. 程式架構：



2. 程式類別：

類別名稱	.h 檔行數	.cpp 檔行數	說明
HStage	147	491	以下關卡的父類別，包含每個關卡會需要用到的方法和變數
HStage_1	39	109	第一關
HStage_2	38	103	第二關
HStage_3	40	103	第三關
HStepsDisplay	72	115	顯示剩餘的步數，每個關卡都有各自獨立的計數器
HMapObject	44	57	以下地圖物件的父類別，包含每個物件會需要用到的方法和變數
HHero	68	158	主角
HPrincess	30	82	通關目標
HRock	45	116	障礙物－石頭
HEnemy	41	112	障礙物－敵人
HKey	25	55	鑰匙
HLock	25	47	障礙物－鎖

3. 程式技術：

A. 同類型的物件繼承共同父類別

我們做的遊戲中，有許多物件擁有共同的屬性和方法。但是若將這些同性質的物件各自獨立開來寫的話，會有以下幾點問題：

1. 如果需要修改某個元素的演算法或名稱時，需要較為繁瑣及重複的修改。如：HHero和HPrincess這兩個class都需要地圖上的x, y座標及實際顯示的x, y座標四個變數，而如果我需要將地圖上的x, y座標的名稱修改為xOnMap和yOnMap，則我需要開啟這兩個檔案後再各自對這兩個檔案中的這兩個變數進行修改。如果只有兩個檔案則不成問題，但是當物件非常多時就會影響到開發效率。

2. 當這些同性質的物件擁有一模一樣的功能時，如果需要對其中的內容進行修改，則會相當費時。尤其是當程式執行錯誤後發現是這些物件都有的某個一模一樣個方法時，則需要一個一個對其修正。

3. 在新增此類型的物件時，除了需要從其他物件複製程式碼過來以外，根據實際用途不同，還需要額外新增些許功能。這樣在後續檢閱程式碼時，較不容易看出個物件間的差異。

綜上原因，我們決定將這些同性質的物件都共用同一個父類別，來解決上述問題。如 HHero 和 HPrincess 都繼承 HMapObject、HStage_1 和 HStage_2 都繼承 HStage。

B. 獨立的地圖座標系統

當角色移動時，程式操作的並不是角色的實際座標位置，而是角色在獨立的地圖上的位置。而玩家每次移動時，地圖都會重新演繹一次，以確保角色在正確的位置上。而顯示時，每個地圖上的物件都會根據自己在地圖上的位置，加上該地圖座標系統位於視窗上的偏移量(Offset)後，才得出實際上的顯示位置，並交由 OnShow 顯示。

C. 可讀性較高的移動系統

傳統上，當我們需要移動角色時，總共需要四個 if 判斷式來處理角色分別在上、下、左和右移動時的內容。但我們為了增加可讀性，我們利用了類似於向量的參數，讓角色移動時，只須根據角色的移動方向來更改移動向量，再根據角色在地圖上的位置加上移動向量後，即可進行針對該移動方向的移動判斷。此舉不僅增加了程式碼的可讀性，還一定程度的增加了執行上的效能，後續要新增功能(如音效)時也可以較好的判斷在哪一種移動模式下需要播放哪一種音效。

四、 結語

1. 問題及解決方法：

在遊戲製作過程中，我們遇到的問題主要都偏向對C++本身使用上的熟練程度。如指標傳遞問題、繼承後override問題。而如果有什麼語法上的問題，我們除了會參考cplusplus.com和stack overflow以及其他網路上的資料與教學外，我們還會向其他同學或助教請教，互相討論。

對於此framework的問題則更多的是透過我們本身對程式開發的經驗及理解，以及借助Visual Studio強大的搜尋功能，來找到我們所需要的各種功能，如調整視窗大小、新增或刪減GameState等。

當然在開發過程中也很常出現各式各樣的bug，在debug時出現的各種錯誤代碼也是非常令人摸不著頭緒。但再次借助Visual Studio強大的除錯能力，我們透過逐行執行及中斷點來糾錯，終於讓程式可以順利運行。

2. 時間表：

週次	組員-郭建麟	組員-林誠祐	說明
1	2	0	尋找遊戲
2	1	0	Git 練習
3	4	0	了解範例程式碼
4	5	5	找遊戲資源
5	5	5	地圖座標規劃
6	3	3	主角移動設定
7	9	9	障礙物設定
8	3	6	通關條件設定
9	3	3	各物件碰撞設定
10	3	3	移動步數設定
11	3	3	修正 bug
12	3	3	新增類別並優化
13	3	3	新增類別並優化
14	3	3	新增快速通關
15	3	3	新增重製按鈕
16	3	6	新增新關卡
17	2	6	修改開始與結束畫面
總時數	58	61	

3. 貢獻比例：

- 108590013 郭建麟：50%
- 108590040 林誠祐：50%

4. 自我檢核表：

週次	項目	完成否	無法完成原因
1	解決 Memory leak	已完成	
2	自訂遊戲 Icon	已完成	
3	有 About 畫面	已完成	
4	初始畫面說明按鍵及滑鼠	已完成	
5	之用法與密技	已完成	
6	上傳 setup/apk/source 檔	已完成	
7	setup 檔可正確執行	已完成	
8	報告字型、點數、對齊、行	已完成	
9	距、頁碼等格式正確	已完成	
10	全螢幕啟動-改列加分項目		

5. 收穫：

● 郭建麟

我在本學期得到的最大收穫是弄熟了C++的指標，以及C++的繼承與法。我主要學的程式語言是Java，但是因為Java不存在指標，到了C++之後反而不太適應。而經過了一年物件導向程式設計課程的洗禮，我已經能夠正確使用及分辨C++的指標了。

● 林誠祐

運用上學期所學的知識以及了解範例遊戲的程式碼後，對一個遊戲的架構以及遊戲功能的實現有了一定的了解，在範例遊戲的程式碼中學到了如何實現各個物件的動作判定與人物移動的相關功能實作方法，最主要還是類別的運用，各個類別的繼承寫好，就能讓程式碼變得更簡單、更好維護。

6. 心得、感想：

● 郭建麟

小時候我曾經去過一個專門教授寫程式的補習班。而我第一次接觸到程式語言後，便夢想著要寫一個屬於自己的遊戲。上了大學後，此課程雖然是透過以前的老師所留下來的framework來撰寫，但我也收穫不少，例如實際體驗一個遊戲的工作流程、理解遊戲執行時具體來說會有什麼動作等，讓我離我的夢想更近了一步。

● 林誠祐

透過這次做遊戲才知道原來製作一個遊戲需要花非常多的時間與精力，要寫上千甚至上萬行的程式碼才能完成一個小遊戲，雖然我們目前做的遊戲規模比不上市面上大部分的遊戲，但是能了解製作一個遊戲背後需要多少的心力，除了思考該如何實現遊戲所需功能之外，還要能夠在如此龐大的程式碼中尋找錯誤並除錯，這並不是一件簡單的事，實際做過遊戲後才知道這是一份非常辛苦的工作。

