

國立臺北科技大學

2023 Spring 資工系物件導向實習

期末報告

**Undertale**



第 19 組

## 目錄

### 一、簡介

1. 動機： .....1
2. 分工： .....1

### 二、遊戲介紹

1. 遊戲說明 .....2
2. 遊戲圖形 .....5
3. 遊戲音效 .....9

### 三、程式設計

1. 程式架構 .....10
2. 程式類別 .....10
3. 程式技術 .....13

### 四、結語

1. 問題及解決方法： .....19
2. 時間表 .....20
3. 貢獻比例 .....22
4. 自我檢核表 .....23
5. 收穫 .....23
6. 心得或感想 .....24
7. 對於本課程的建議 .....24

## 一、簡介

### 1. 動機：

這個遊戲我們這組的兩個人都玩過而且也都很喜歡，尤其是對於作者製作遊戲的細節以及設計的關卡玩法感到十分敬佩，所以想說來重現一下這個遊戲裡面的一些遊玩內容。

### 2. 分工：

我們一開始先設計架構，接著分配誰該做哪個部分，該部分的人要負責除了能讓程式碼順利地動起來以外，若有圖片需要處理的部分，也該自行處理完畢。

較大項目大致可以分為：

內容	負責人
act barrage Character enemy frame game_manager game_text greater_dog items migosp monster_frame move papyrus	110590017

text user_frame	
BarrageMode ButtonFrame Fight GreaterDogRound PapyrusRound TextContent menu	110590015

## 二、遊戲介紹

### 1. 遊戲說明

遊戲的通關方式不只一種，玩家可以選擇與它 fight、act、item、mercy，是個自由度較高可以隨心探索的遊戲。

玩家選擇行動後，系統顯示出該行動的結果，怪物也會給玩家一個行為，也就是彈幕攻擊。

可以透過查看怪物對於玩家行為的回饋，來進行下一個行為，嘗試讓怪物接受和平的解決這場戰鬥，以達成通關條件。

目前過關路徑大致有三種

1. 透過 fight 對怪物造成攻擊，將其殺害以結束遊戲。
2. 透過 act 討好怪物，讓他願意放過你，則選擇目標時怪物的名字會呈現黃色。  
這代表玩家可以使用 mercy 中的 spare，以結束遊戲。
3. 該怪物特殊的判定，有可能會需要先 spare 才會觸發條件或是該怪物特殊的 act 會觸發某種 event(下方攻略會提到)

結束遊戲進到結束畫面後，按 Q 可以回到選單進行下一場戰鬥，且玩家等級會因剛才完成關卡後拿到的 exp 而升等，最大血量上升。

注意：

壹、遊玩前請安裝.\Resources\font 內的字體

貳、Stage 01 因為是小怪 所以怪物圖片旁邊的對話框會隨時間自動消失，但 Stage 03 是 boss 關所以怪物圖片旁邊的對話框需要按下 enter/z 才會進入下一個階段

遊玩規則：

這個遊戲基本上是彈幕遊戲。針對不同的怪物，通關的方式會不一樣。在主頁按下 Q 可以查看教學，按 ESC 回到主頁。

←→選擇 FIGHT、ACT、ITEM、MERCY

↑↓←→選擇框中的行動

↑↓←→進行彈幕閃躲

ENTER/Z 下一步/做選擇

SHIFT/X 上一步

G GodMode

如果想查看怪物的彈幕攻擊模式，但又不想一直躲避，可以按下 G，會觸發 god mode，無論如何都不會受傷，直到再次按下 G 取消這個模式為止。

攻略 : (請反白)

Stage 01 Migosp 破關分支：

talk -> mercy -> 結束

fight -> hp = 0 -> 結束

Stage 02 GreaterDog 破關分支：

beckon -> pet -> play -> pet -> pet -> mercy -> 結束

ignore -> ignore -> ignore -> ignore -> 結束

fight -> hp = 0 -> 結束

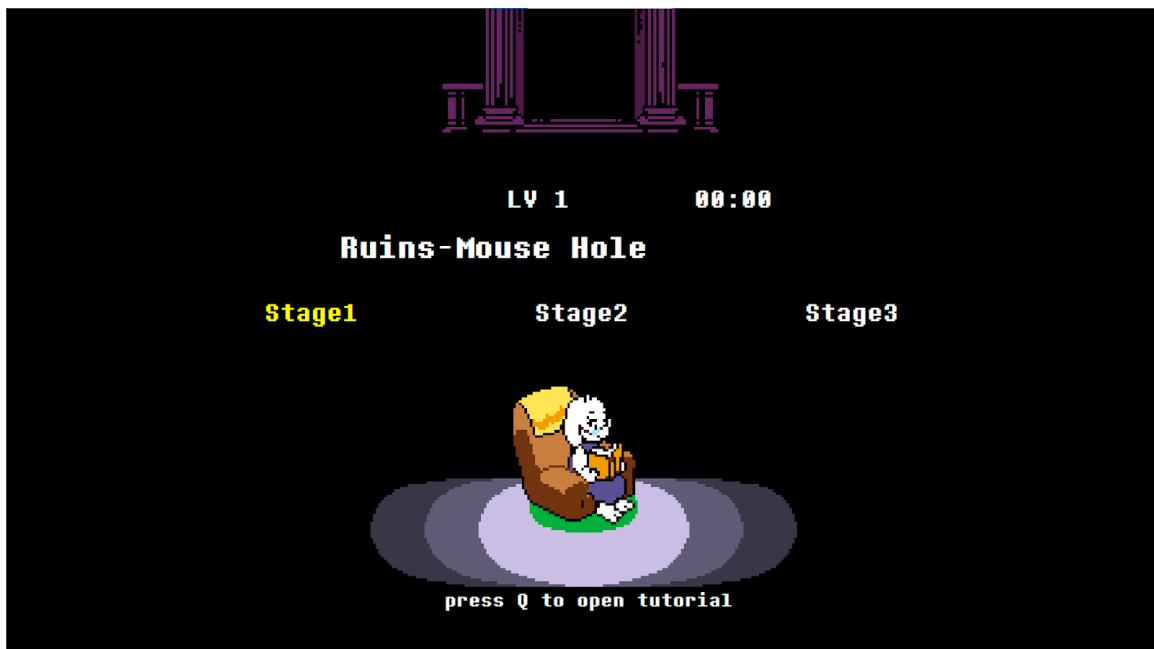
Stage 03 Papyrus 破關分支：

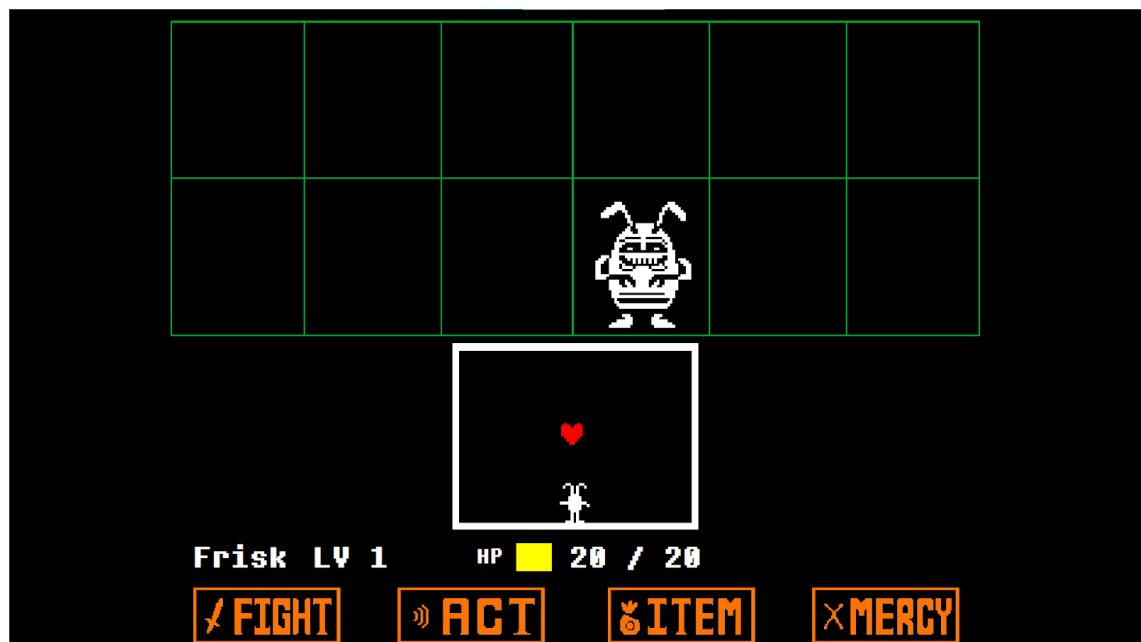
loop round -1 -> if mercy or fight round 0 ->

round 1 -> round 2 -> round ... -> round 21 -> mercy -> 結束

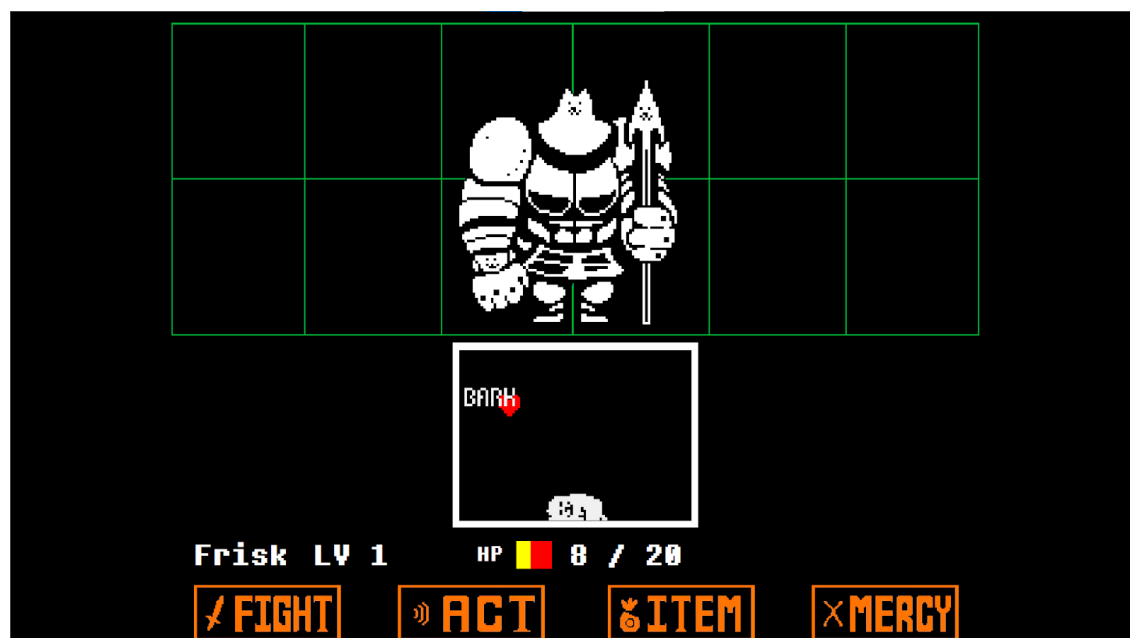
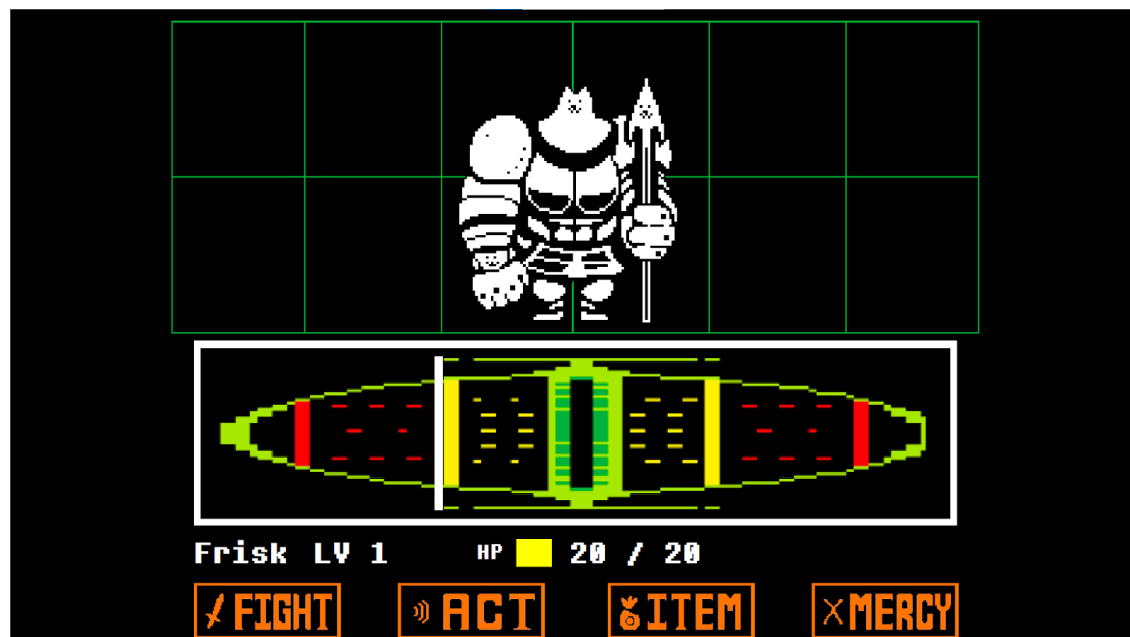
fight -> hp = 0 -> 結束

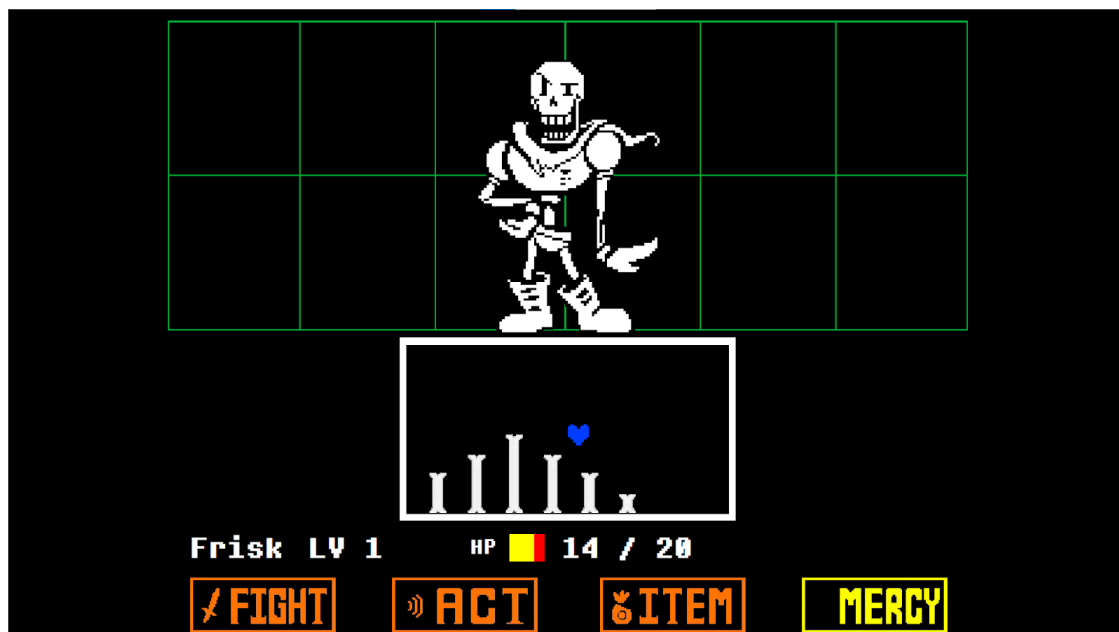
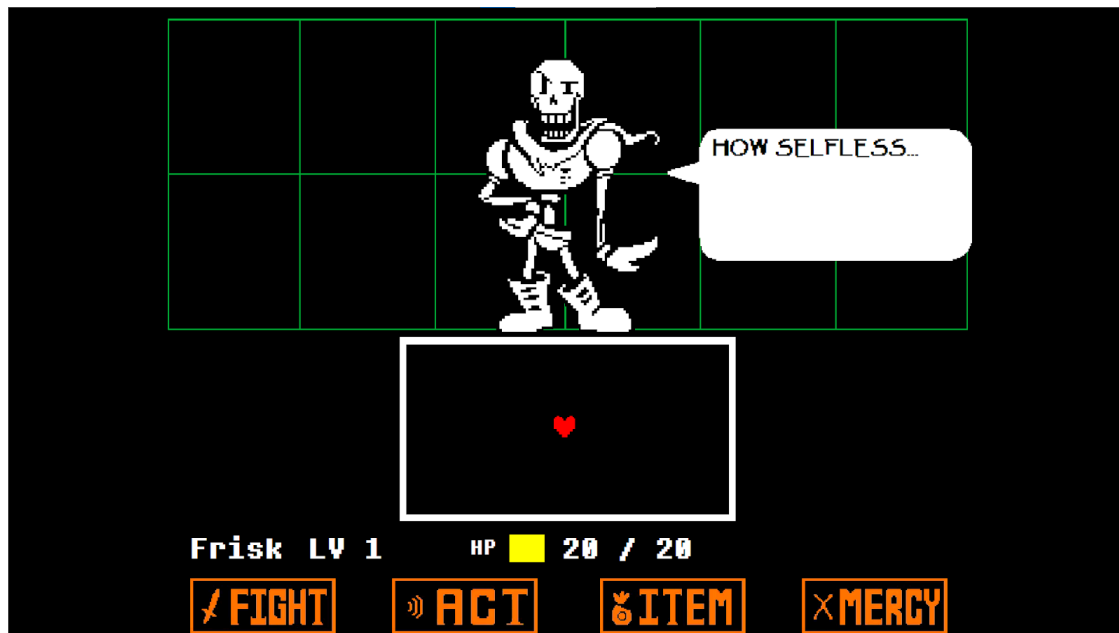
## 2. 遊戲圖形











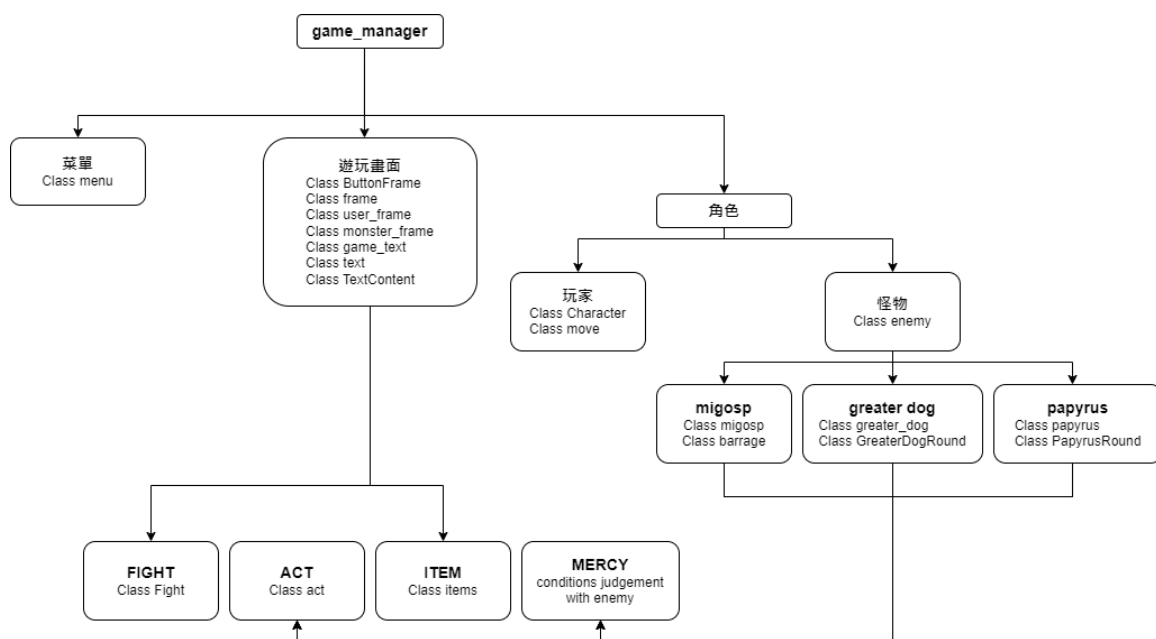
### 3. 遊戲音效

資料夾	音效類別	音效檔案
Resources\Background	Stage 01 背景音樂	01.mp3
Resources\Background	Stage 02 背景音樂	02.mp3
Resources\Background	Stage 03 背景音樂	03.mp3
Resources\Background	meun 背景音樂	start.mp3
Resources\Background	GameOver 背景音樂	end.mp3
Resources\Effects	進入 Stage 音效	battlefall.MP3
Resources\Effects	結束戰鬥怪物消失	endfight.MP3
Resources\Effects	fight 斬擊聲	fight.MP3
Resources\Effects	使用 item 回血	heal.MP3
Resources\Effects	彈幕戰鬥我方紅心受傷	hurt.MP3
Resources\Effects	左右選擇 fight,act,item,mercy	select.MP3
Resources\Effects	在框裡選擇的音效	select_1.MP3
Resources\Effects	怪物文字特殊效果音效 (音效後有空白)	txt.MP3
Resources\Effects	怪物文字特殊效果音效 (加大 txt.mp3 聲音)	txt_2.MP3
Resources\Effects	怪物文字特殊效果音效	txt_3.MP3

	(切除 txt.mp3 空白)	
Resources\Effects	Stage03 切換愛心模式音效	tempbell.MP3

### 三、程式設計

#### 1. 程式架構



#### 2. 程式類別

類別名稱	標頭檔 (.h)	原始碼 (.cpp)	說明
act	16	29	處理 ACT 的行為
barrage	61	148	處理單個彈幕的行為

BarrageMode	138	253	處理一群彈幕是如何一起行動的系統 barrage manager
ButtonFrame	64	144	控制四個主要行為的按鈕
Character	38	101	處理玩家所有資訊與行為
enemy	188	298	處理怪物的系統 控管所有敵人需要的東西
Fight	157	294	處理 FIGHT 的行為
frame	56	67	不同種類 frame 的大架構
game_manager	53	231	與流程互相配合的系統 管理流程
game_text	37	80	讓文字可以一個一個跑出來的動畫系統 text manager
greater_dog	55	351	greater dog 這個怪物的控制系統
GreaterDogRound	40	214	greater dog 這個怪物的彈幕攻擊模式系統
items	59	206	處理 ITEM 的行為
menu	29	121	菜單系統

migosp	39	188	migosp 這個怪物的控制系統
monster_frame	32	57	怪物說話的框(上方白背景框)系統
move	62	250	玩家移動系統
papyrus	54	324	papyrus 這個怪物的控制系統
PapyrusRound	75	767	papyrus 這個怪物的彈幕攻擊模式系統
text	43	71	處理文字顯示的系統
TextContent	34	56	從 json 讀取資料的系統
user_frame	64	277	繼承 frame 處裡白邊黑底的框的移動 與顯示內部會的文字
總行數	1394	4527	

### 3. 程式技術

110590015:

我提一個花最久時間處理與修正的地方，就是我在處理每個怪物的彈幕攻擊模式時發現每個怪物的模式都不一樣，而且會出現多少彈幕根本不知道，因為數量不是固定的。而每個彈幕會有許多數值需要設定，如圖片、速度、攻擊力、初始位置等等。所以，如果想在怪物的 class 加入所有彈幕的 member，數量不固定、數值也不固定，如果無腦把所有彈幕加總，全放在怪物 class 的話讀取速度會太慢，整個程式在執行的時候也會很卡。

我想到其實一個回合應該只要有該回合需要出來的彈幕就可以了，但數量一樣是不固定的，每個怪物的回合模式數量也不一樣。

於是我決定寫一個 Class BarrageMode，主要做的事情是：

1. 控制什麼時候會需要哪種模式
2. 哪種模式需要哪些彈幕
3. 判斷及處理這個回合的狀況（扣血、結束、顯示的狀況等等）

但是，彈幕的數量跟數值這邊還需要處理。如果直接在程式指定的話，一個 function 需要的 parameter 會很多，於是我決定將每個回合的彈幕資料儲存在 json 中，並寫了一個讀取 json 資料的 Class，主要做的事情是：

針對我現在需要的狀況，讀取該狀況的資料，並處理一些例外狀況

所以我的 Class BarrageMode 變成：

1. 控制什麼時候會需要哪種模式
2. 哪種模式需要哪些彈幕
3. 讀取所需的數值資料、處理並儲存
4. 讓每個回合的每個彈幕正常的動起來，移動軌跡正常
5. 判斷及處理這個回合的狀況（扣血、結束、顯示的狀況等等）

即使我最後一個敵人有 20 幾個回合，有超過 100 個的彈幕，我也不需要一次讀完，我可以只在我需要什麼資料的時候讀該有的資料，這樣可以節省記憶體空間、同時執行的速度也不會那麼卡。

110590017:

#### ▼流程控管(game\_manager)

透過狀態機將遊戲分階段

利用 enter/z 控制進入下一個狀態

利用 shift/x 控制回到上一個狀態

每個階段設置是否能自行回到上一狀態或是自行往前下一個階段

持續 loop 某段流程直到，通關進入流程 enter 後進到 my\_game\_over

設定每個階段所有功能的開啟與關閉(enable)

與某些功能需要在一個流程完整跑完後重製

#### ▼文字系統(text, game\_text)

包裝原始 CTextDraw 的文字功能成新的 obj text.cpp，text 物件控管該字串印出、該 text 特效(Type 效果、正常模式)、顏色、位置、字體、寬。

開發 game\_text(text)，管理 text 組成的 vector，負責批次印出、控管印出時機 (enable)、依照設定的 mode 自動 format text 們的位置。

#### ▼道具系統(items)

設定 Struct 紀錄 item 數據，名稱、使用次數、恢復血量，使用道具扣除該道具使用次數，for each 所有道具如果使用次數為零，則將該道具 delete 出道具清單。

game\_manager 傳入選擇到的道具，道具系統(items)回傳該道具的文字與回血量。

有的道具能使用兩次，有的道具使用後會改變名稱。

有想過需不需要 ID 但後來只有 4 個道具就採取 ID 跟 item 物件，而是直接使用 items

#### ▼玩家系統(character)



紀錄玩家基本資料，名稱、level、exp、最大 hp、當前 hp、接收 game\_manager 傳進的回血/扣血修改玩家當前 hp，遊戲結束紀錄 exp 與改變 level。

#### ▼ 玩家本體移動系統(move)

負責控管戰鬥中的可操控閃躲彈幕的玩家本體。

透過 game\_manager 接收 user\_frame(框)所傳進的角落座標，去限制玩家本體的移動範圍。

以向量(0,0),(0,1),(0,-1),(1,0),(-1,0),(-1,1)...代表方向性，控制玩家本體移動方向。

以 WindowsAPI 的 GetKeyState 去偵測持續按下按鍵，並非使用 OnKeyDown 邊緣觸發。

切換玩家本體移動模式：

▽紅心模式：

預設模式，預設向量為(0,0)靜止不動，設定移動 x,y 係數，依照按鍵控制向量，正規化使斜方向移動距離與上下方向移動距離相同。

▽藍心模式：

特殊模擬重力模式，預設向量(0,1)，玩家本體依照 x,y 係數下降，透過計數按下↑的時間計數，到了一定的時間會自動掉落，依照計數器數字(高度)計算重力加速度更改 y 係數，著地前都無法再次跳躍。

#### ▼ 彈幕系統基底(barrage)

基本彈幕單位，含設定位置、基本移動、切換模式、呼叫玩家(character)扣血功能  
抓取玩家本體(move)的紅心位置偵測是否與彈幕圖片重疊，接著做碰撞時處理，消失/不消失/扣血/閃爍。

#### ▼ 敵人圖片旁邊的框系統(monster\_frame)

在該顯示的階段 game\_manager 打開 enable enemy 將文字(game\_text)與模式傳入、按照模式印出

敵人旁邊的模式指定，分別有三種模式

自動消失：隨著時間自動消失(Stage 01)，偵測計數器時間進到下一階段

不會出現：該敵人不會說話(Stage 02)，偵測 pass 直接進到下一階段

Enter/z 下一步：該敵人說話需要透過 enter/z 才可以進到下一步(Stage 03)

### ▼下面長框系統(frame, user\_frame)

#### ▽移動部分

本體有四張圖片，分別為四個邊，框的移動是透過移動四邊圖片而形成動畫效果。

主要核心為 frame 底下的 create\_frame 所構成，此 function 能夠透過輸入想要的寬高、以及框的左上角(x,y)，去形成任意大小的且左上角定位在 x,y 的框。

且計算了該框的四邊角落的位置，主要功能為給玩家本體與彈幕抓取移動範圍。

透過 create\_frame 去個別寫出四邊移動的方法例如

左右等速向中間縮減框的寬度

左右等速向外增加框的寬度

將上邊往下移動

將上邊往上移動

只向右增寬，左上角基準點不動

只向右簡寬，左上角基準點不動

基本上都是呼叫 create\_frame 加減裡面的數值一幀一幀去移動，形成動畫。

再透過每個怪物特有的框去客製範圍停下移動。

目前整合了幾個會用到的框移動動畫，用 enum 命名，包成 function 只要給 command 就可以一行完成

預設模式→方形框(Stage 1/2)

預設模式→偏低長方形框(Stage 2)

所有框→預設模式

預設模式→大長方形框(Stage 3)

大長方形框→右邊拉長非中心位置框(Stage 3 後期 使用)

(雖然 Stage 03 超級攏長但建議不要跳過台詞，特殊演出也會被跳過)

右邊拉長非中心位置框→大長方形框(Stage 3 後期 使用)

#### ▽選擇及文字部分

框需要移動外同時須要顯示文字，以及選擇功能。

需要在框中印出的文字透過 `game_manager` 傳入 `game_text(text_manager)`。

傳入的東西會在框中被 `print` 出。

`print` 出的東西則會被選擇功能引用，

依照傳入 `game_text` 內的 `text vector` 數量去判斷能夠選擇的數量，

或是該 `game_text` 的設定上並不是能夠被選擇的。(系統文字等等...)

#### ▼敵人系統(enemy)

大雜燴 `class`，因為本遊戲的敵人都過於個性化，

只能大概統整出一個 `enemy class` 去讓敵人們繼承，

但因為內部設定過於相異，所以每個敵人又分別有個 `class`。

內部設置了敵人基本數值，與各別需要用到的條件判斷。

大略分為以下功能

設定該 `enemy` 可行動項目

該項目回傳的台詞透過 `game_manager` 傳給框顯示

設定該敵人當前狀態系統提示字(戰鬥結束後回去的那個地方)

該項目回傳的台詞透過 `game_manager` 傳給框顯示

設定選擇的行動會顯示什麼文字

該項目回傳的台詞透過 `game_manager` 傳給框顯示

設定該怪物因為行動影響怪物圖片旁邊的文字內容

該項目回傳的台詞透過 `game_manager` 傳給框顯示

設定該怪物破關條件及通關文字...同上

輸出彈幕設定、決定攻擊結束時機、轉變玩家本體模式信號、等等特殊判定.....

紀錄行動造成的數據，去更改敵人狀態

#### ▼大致台詞處理流程

透過 json 讀檔取得當前敵人/道具所有文字資訊，存成 unordered\_map

在需要 print 出的時候讀出 map 內的字，enemy/item 會回傳台詞

框會在需到的時候 print 出來

(如果有隨機台詞會在讀出 map 的時候 rand 傳回)

enemy/item 會記錄該次 enter 次數去叫出下一段文字

叫出來的文字會建立一個 text vector 拿去建造一個 game\_text

game\_manager 需要用到的時候會去 get enemy/item 的 game\_text

框拿到字後透過 game\_manager 去控制 print 出的時機

#### ▼雜項：

menu 的計數器用了 google 教的 CString 的 format 自動補零再轉 string 蠻魔法的。

印出文字的特效是 CAudio 同個音效設了很多個通道(id)播放才不會有卡住問題，蠻魔法的。

## 四、結語

### 1. 問題及解決方法：

110590015:

花最久時間處理的第一個是在處理動畫的問題。在 FIGHT 當中，攻擊敵人的白條與扣血的顯示都是漸進的，不會說取得值之後一瞬間做完，因為要讓他慢慢做，所以要設置計數器，在時間內一步一步的做，產生漸進的感覺。同時，因為這個框架是每固定時間自動刷新，所以我取攻擊值後攻擊的行為一直一直重複發生，敵人的血無限的被減下去，所以也要設定一個判斷值，當我執行一次後，直到回合結束前，就不執行了。新回合時，這個判斷會重置。這樣才可以正常執行應該要執行的功能。

第二個是，怪物在攻擊時，通常不會只出現一個彈幕。如果在怪物的 class 中放置幾十個幾百個彈幕當作他的 member，會太冗長、不好控制而且一次讀這麼多的東西會太花時間，執行也會很卡，所以我在想該怎麼處理這個很多東西一起動的問題。就如我程式技術那邊提到到的，我做了一個處理一群東西一起動的 class，只讀取需要的數值以及讓程式跑起來不會那麼卡。

110590017:

設計控制流程的狀態機，包裝原始的 print()成文字系統之類的

敵人內部機制都有微妙的不同，內部設定幾乎都長得不太一樣，需要在 Enemy 設置系統上下點功夫。

因為自己設計的架構做的很破，所以天天重構天天快樂，有問題就是不知道自己在寫甚麼，無情重構是最好的選擇，動態斷點人生好夥伴。

還蠻重要的點，這次所有命名幾乎都是用駝峰式，中途才把 enum 的改成較常見的命名格式，下次要記得注意不同類型的東西跟命名規則要有區分。

## 2. 時間表

週次	110590015(小時)	110590017(小時)	說明
1	說明		
2	17	3	熟悉框架，架構討論，做長頸鹿練習
3	8	10	15:處理動畫以及線條伸展問題，延遲移動問題等等 17:開發 frame&設計 api
4	12	20	15:修復只做一次的功能無限的做下去的 bug 17:開發 user_frame&開發 text&開發 game_text&設計 api
5	16	15	15:BOSS 關與怪物攻擊的彈幕移動模式 17: 開 發 move& 開 發 monster_frame&開發 charactor&設計 api
6	11	15	15:刷新速度不夠快該怎麼處理例外的問題 17:開發 items&處理 mygame_run 狀態機跟 game_maneger
7	10	13	15:修正彈幕大框架的 bug 與增加細項功能

			17:開發 act&stage1 完成
8	demo1		
9	10	14	15:把第三個敵人 23 回合的 22 個回合寫完 17:重構 enemy&開發 barrage
10	9	11	15:處理第三個敵人最後一回合，與修前面回合的 bug 17:stage2&重構
11	7	10	15:修復無敵時間 bug，處理第二個敵人的彈幕攻擊模式 17:重構
12	7	10	15:繼續處理第二個敵人的彈幕攻擊模式與修復 bug 17:重構
13	demo2		
14	13	17	15:完成第三個敵人最後一回合、修復第二個敵人 bug 17:重構台詞導入系統&stage3&到處加音效
15	9	10	15:搭配流程作出彈幕修正 17:修 bug&重構

16	12	16	15:處理細小 bug，重構一些程式碼 17:開發文字特效&重構
17	10	7	15:重構程式碼、準備收尾 17:期末養老&吃飽閒閒加細節
18	demo3		
總 時 間	151	171	

### 3. 貢獻比例

110590015：50%

110590017：50%



#### 4. 自我檢核表

	項目	是否完成	為何不能完成
1	解決 Memory Leak 的問題	是	
2	自訂遊戲 Icon	是	
3	About 畫面	是	
4	遊戲中說明如何操作	是	
5	發佈(Release)版本	是	
6	承上項，該版本可以正常運作	是	
7	報告字型、點數、對齊、行距、頁碼等格式正確。	是	

#### 5. 收穫

110590015:

寫了這次專案以後，對於 C++ 與一些物件導向的特性更加熟悉了。對於繼承、多型、一些物件的處理方式、如何解決問題、如何在有限的條件下做到想做的事情、如何重構與修改讓我的程式碼變得更好等等...實在是有很多學到的東西。從以前只有 terminal，只看得到文字變成真的有畫面，真的有互動時，真的很感動。

110590017:

雖然設計了一些系統，但除了數學以外沒有覺得特別學到什麼，就寫遊戲蠻開心的，頂多就寫了一學期的 cpp 稍微熟了一點但其實都沒用到甚麼 cpp 技術。  
寫遊戲就是寫數學。

## 6. 心得或感想

110590015:

開發遊戲的過程雖然一直遇到 bug、程式常常不知為何 crush 掉，當下雖然頭很痛，但是當完成看他動的很正常的時候真的很感動。與上學期不同，不是為了寫作業而寫 c++，而是真的想完成什麼，做出什麼才寫 c++，目的不同，感覺思考的方向與目標也不同。而且做的還是一直很喜歡的遊戲，在玩的時候完全不會注意到這麼多，真的自己做了一次發現作者把一切都掌握得太好了，不論是細節或是玩法。這份專案讓我學習到不止是程式碼的知識，還有一些設計遊戲的東西，比我預期中學到了更多的東西。

110590017:

我寫了垃圾架構狀態機，我寫的重力也是垃圾，陽春產物堆在一起，常常在改內部架構，可能有慢慢變比較好，但還是寫了很多爛 code。

## 7. 對於本課程的建議

110590015:

沒有什麼特別的想法，謝謝助教與老師。

110590017:

oop 過了不代表會寫 oop