物件導向期末專題-第二組

迷離符文RunesBlur

1. 介紹(動機與發想)

起初我們在討論的過程中，林彧頎(組長)提出了，3個主題，分別是小朋友下樓梯、跑酷遊戲、與類似RPG的戰鬥遊戲，其中有部分組員對於RPG遊戲持贊同意見，另外潘妤揚組員有提出要不要做橋牌遊戲，而我們最後決定將這兩種所擁有的一些特徵結合，最終形成了RunesBlur迷離符文這款遊戲!

而我們也開始自行思考與設計!

一張含有 室內 的圖片

自動產生的描述一張含有 文字, 個人, 桌, 室內 的圖片

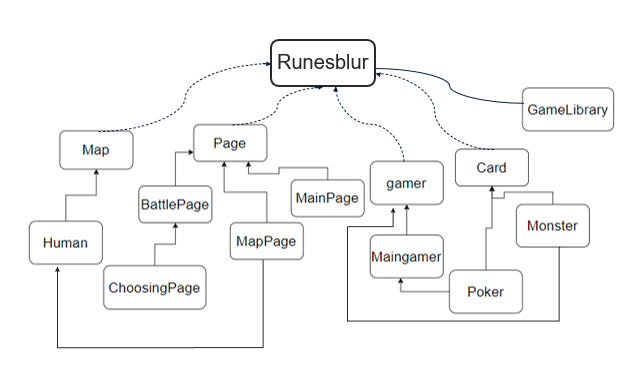
自動產生的描述

Figure 1 RPG遊戲示意圖 Figure 2 橋牌示意圖

1. 架構

在我們遊戲的基礎架構中，希望可以透過讓玩家在RunesBlur的遊戲故事架構下進行遊玩，因此我們的遊戲一開始進入後會有故事介紹(林彧頎發想)，並透過基礎的遊戲流程(陳于瑞發想)，玩家基本上進入遊戲後，需要在迷離符文大地(地圖)中行走，並會在沿途中碰到怪物進行對戰，對戰過程中，會選擇腳色進行攻擊，而這些腳色也對應到卡牌的五種屬性(有動畫)，在攻擊完全部的怪物後，會回到主頁面，也代表著主角一行人逃離了迷離符文大地!

以下是我們code的架構，也可以說是UML的簡單地封裝、繼承、多型之間各個class的關係!



從最基礎來說，我們的Map就是在定位每一個點，而Human，則是可以繼承Map的位置，進而達到人物可以在上面走動的效果。另外MapPage分別又繼承了Page、Human，繼承Page是因為地圖也需要音樂的引進，Human是因為main呼叫的時候不想讓使用者看到Map的內容，所以只會讓使用者呼叫MapPage的東西，丟進去的東西之後會連結Map的函式進行設定，而Choosing Page則是可以做到輔助Battle Page的一些腳色上的選擇等等...因此會從BattlePage繼承，而Battle則是做到展示戰鬥畫面，因此會先從Page繼承最基本的音樂要素，另外MainPage與BattlePage一樣都會從Page繼承音樂要素，並主要進行遊戲一開始主畫面的部分!

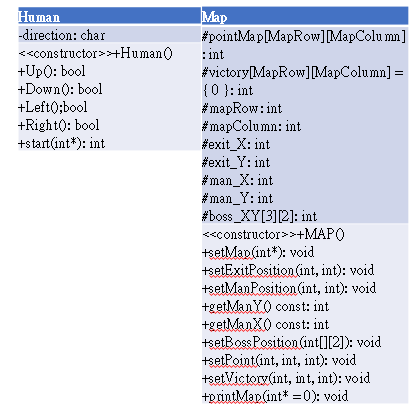
再來是玩家與怪物的部分，玩家與怪物是由最基礎的gamer進行abstract class的設定，而會由繼承的Maingamer與Monster來對攻擊、等級…等需求進行實作，另外玩家的Maingamer還有Poker會額外幫助主角建立個別腳色(如:king、queen…等等)的來對這些設定進行運算與實作，而同時前面所提到的Poker與Monster皆會需要額外繼承Card來對應其後面實作所對應的不同腳色或怪物之間的資料庫設定!

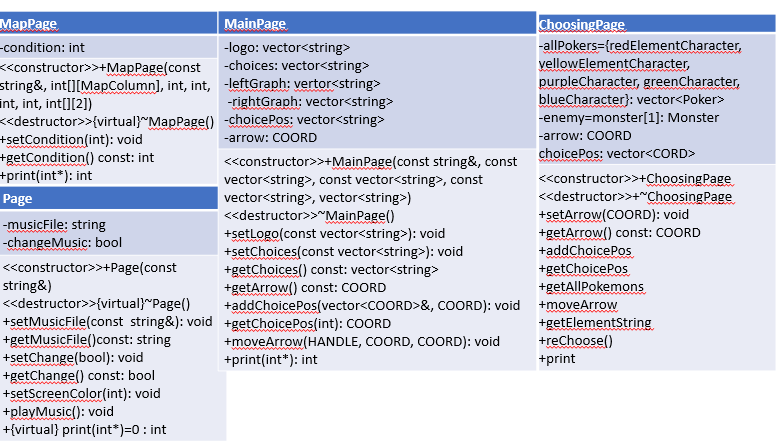
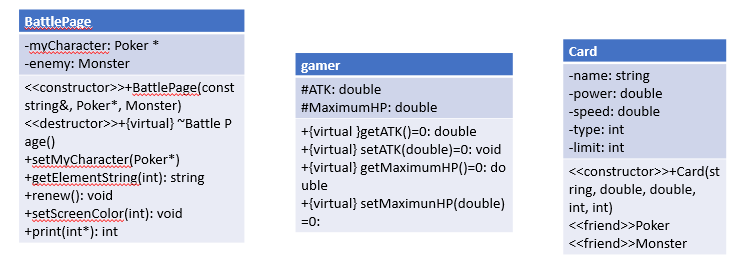
最後以上這種種，會於GameLibrary進行純粹的資料庫建立，像是圖形、地圖、腳色、怪物設定等等…，最後會於RunesBlur進行實作!

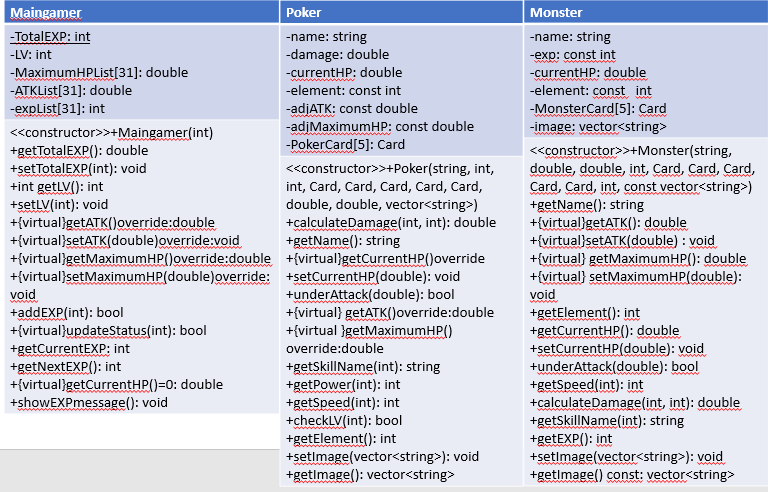
(UML於下一頁)

1. UML

這裡將展示上面對UML的架構解說的個別UML!







1. CODE(三大OOP原則)

下面將解說Code的三大OOP基本原則與主要運作介紹!

1. 玩家(封裝、繼承、多型)

玩家的部分，玩家是由最基礎的gamer進行abstract class的設定，而會由繼承的Maingamer來對攻擊、等級…等需求進行實作，另外玩家的Maingamer還有Poker會額外幫助主角建立個別腳色(如:king、queen…等等)的來對這些設定進行運算與實作，而同時前面所提到的Poker會需要額外繼承Card來對應其後面實作所對應的不同腳色之間的資料庫設定!

綜合以上，看到程式碼中，玩家有運用到封裝、繼承、多型，以下圖片解說。

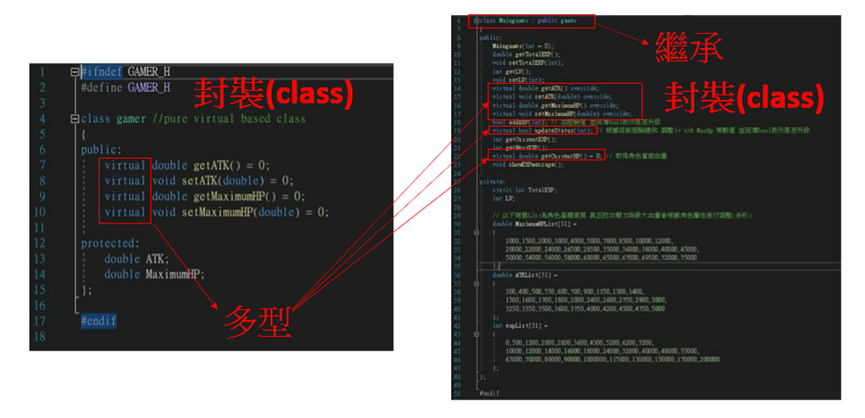


Figure 1(左)gamer.h(abstract class用於被繼承，並展現多型(virtual)

(右)Maingamer.h(繼承gamer.h並展現多型(virtual override) 並且這兩者 所對應的不同功能由個別的class封裝起來!)

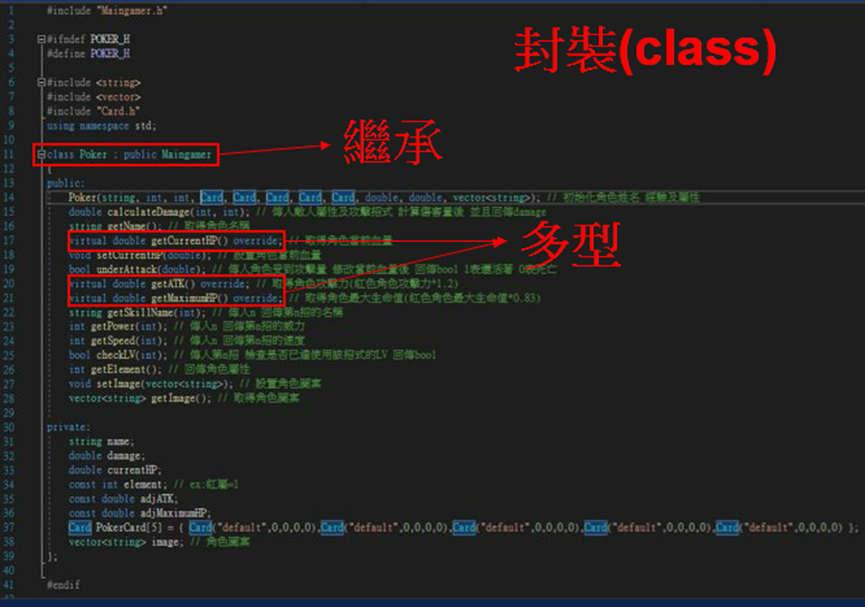


Figure 2Poker.h(繼承Maingamer並且多型於virtual，最後本身class將這些封裝。)

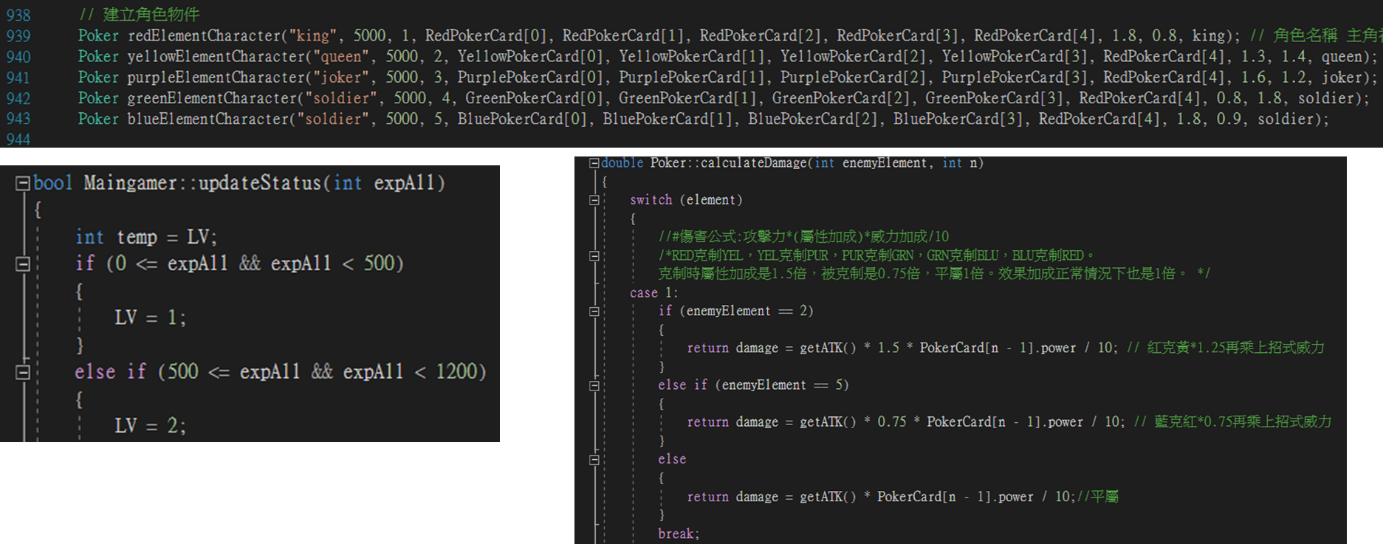


Figure 3玩家的實作示範

1. 怪物(封裝、繼承、多型)

怪物是由最基礎的gamer進行abstract class的設定，而會由繼承的Monster來對攻擊、等級…等需求進行實作，Monster會需要額外繼承Card來對應其後面實作所對應的怪物之間的資料庫設定!

綜合以上，看到程式碼中，怪物有運用到封裝、繼承、多型，以下圖片解說。

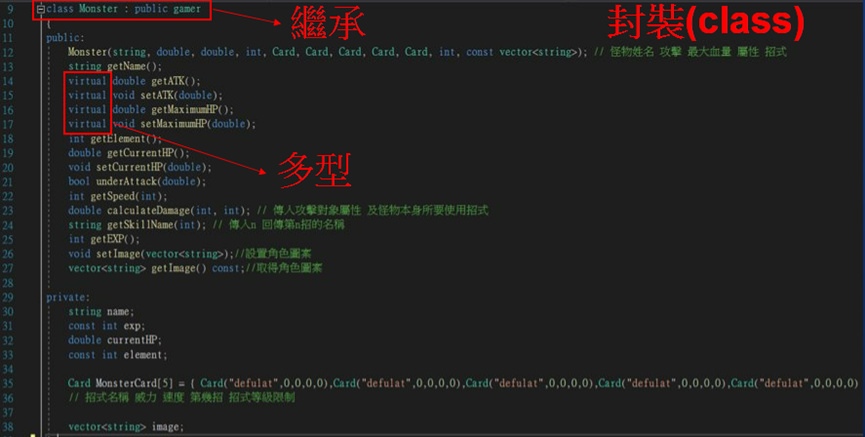


Figure 4Monster.h(會繼承gamer、card 並展現多型(virtual)最後本身有封裝class)

一張含有 文字, 綠色, 室外, 匾額 的圖片

自動產生的描述

Figure 5 Monster實作程式

1. 卡牌(封裝)

作為主要設定基本卡牌所對應功能的abstract class會給玩家與怪物繼承!



Figure 6 Card作為abstract class有封裝的作用!

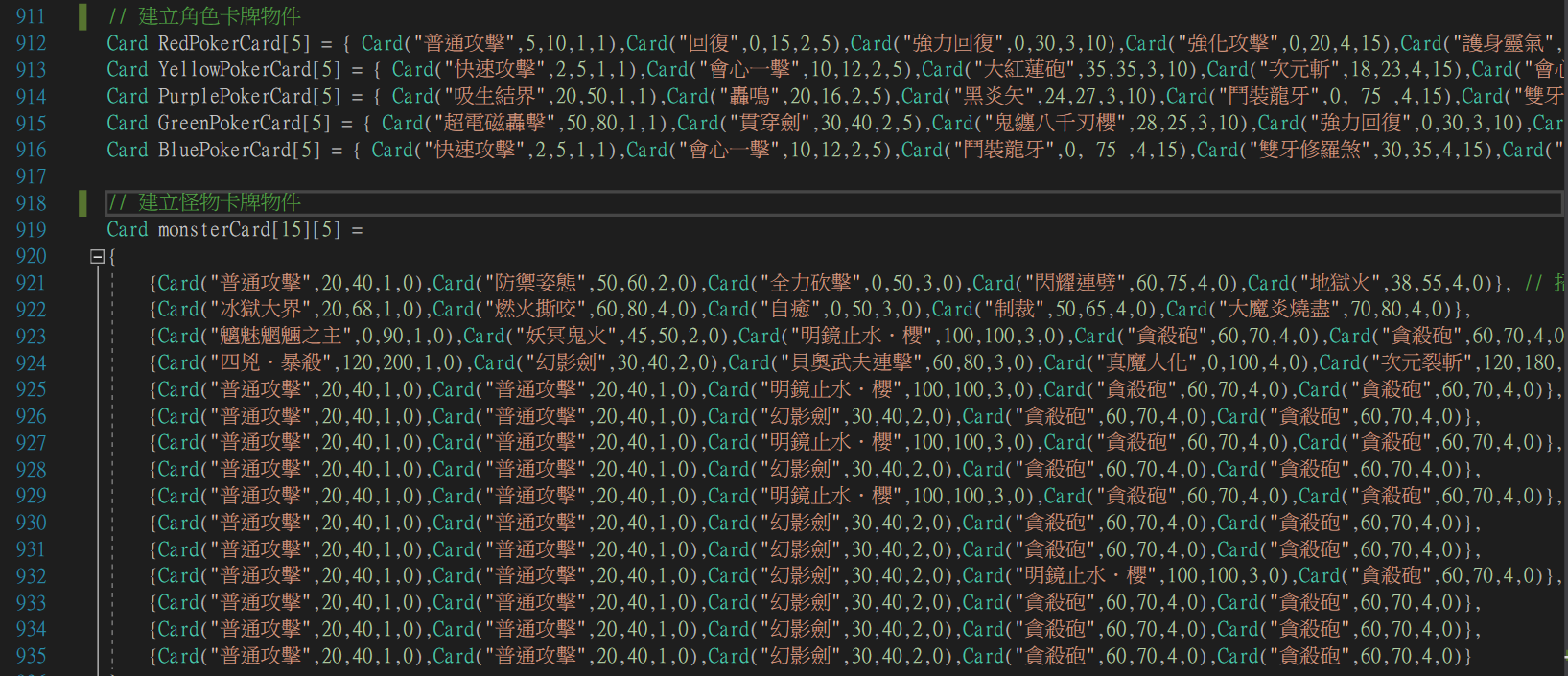


Figure 7 Card被poker與monster繼承後的實作程式

1. 地圖(封裝、繼承)

Map就是在定位每一個點，而Human，則是可以繼承Map的位置，進而達到人物可以在上面走動的效果。另外MapPage分別又繼承了Page、Human，繼承Page是因為地圖也需要音樂的引進，Human是因為main呼叫的時候不想讓使用者看到Map的內容，所以只會讓使用者呼叫MapPage的東西，丟進去的東西之後會連結Map的函式進行設定，而Choosing Page則是可以做到輔助Battle Page的一些腳色上的選擇等等...因此會從BattlePage繼承，而Battle則是做到展示戰鬥畫面，因此會先從Page繼承最基本的音樂要素，另外MainPage與BattlePage一樣都會從Page繼承音樂要素，並主要進行遊戲一開始主畫面的部分!

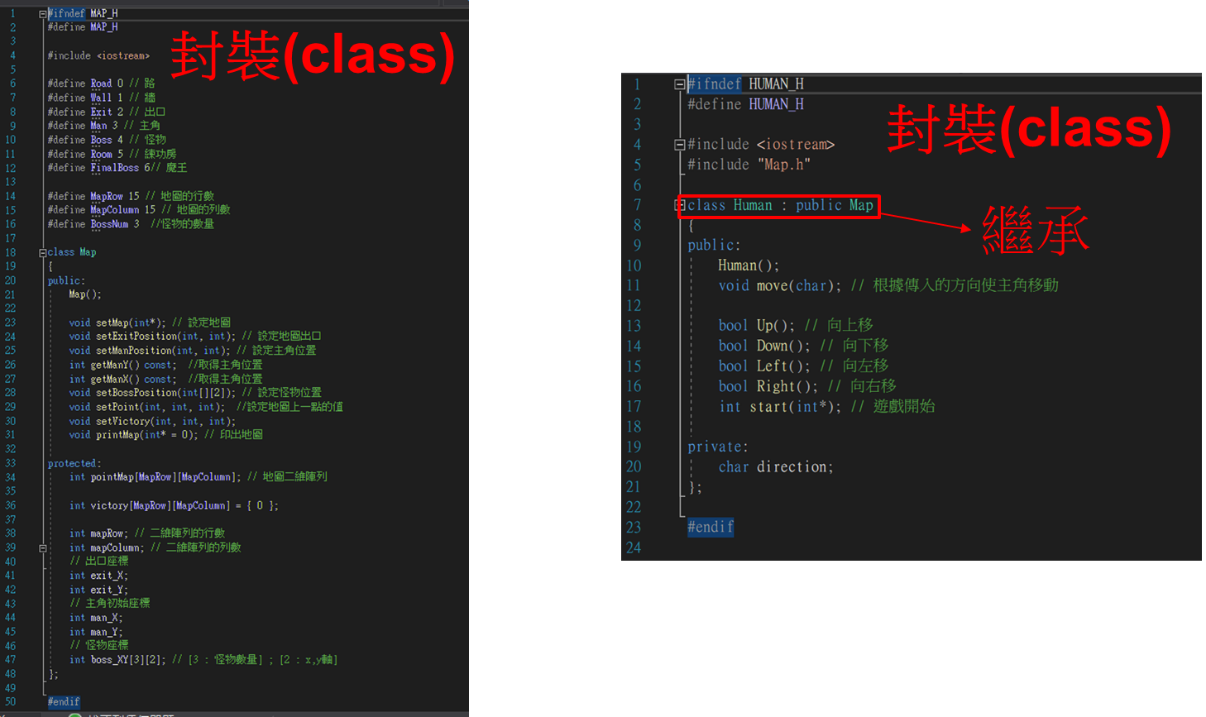


Figure 8 地圖本身的繼承關係與class本身的封裝

一張含有 文字 的圖片

自動產生的描述

Figure 9 地圖實作程式

1. 補充(成果展示code示範)

統合以上的勇者 怪物 卡牌 地圖 後我們另外有建立程式的library透過配合Runesblur.cpp執行出最終程式。

一張含有 文字 的圖片

自動產生的描述

一張含有 文字 的圖片

自動產生的描述

1. RunesBlur(成果展示)

(因為是在程式運行下錄音，因此遊戲音樂品質較不佳，另外也會有操作的聲音!盡請見諒!)

[](https://www.youtube.com/embed/QWBEx1vD_W4?feature=oembed)

1. 分工

|  |  |  |
| --- | --- | --- |
| 林彧頎(A1105505) | 潘妤揚(A1105545) | 杜佩真(A1105549) |
| 1.玩家CLASS相關撰寫  2.POKER-CLASS主要撰寫  3.主要統整CODE  &邏輯運算  (垂直整合)  4.PPT製作  5.WORD報告撰寫  整合與補充  6.遊戲故事設計人員  7.PPT報告 | 1.地圖相關CLASS撰寫  2.主要統整CODE  &邏輯運算  (垂直整合)  3.PPT製作  4.PPT報告 | 1.遊戲架構設計人員  2.主要統整CODE  &邏輯運算  3.遊戲畫面設計與實作  4.PPT製作 |
| 陳于瑞(A1095547) | 王重鈞(A1105531) | 磨文傑(A1105509) |
| 1.主要遊戲架構設計人員  2.協助統整CODE  3.PPT報告 | 1.怪物CLASS相關撰寫與怪物OBJECT建立  2.PPT報告 | 1.CARD CLASS相關撰寫與建立怪物與POKER之間卡牌物件  2.CARD PPT製作  3.PPT報告 |

(心得於下一頁)

1. 心得

**A1105505林彧頎**

在這次專題中，我是組長，因此可以說專題的過程中，每一個步驟我都有參與，或幫忙校正，一開始是由我來找組員，接著訂定我們要討論的時間，由於有些組員要上班或是有事要忙，因此基本上我們都是線上討論並完成，我一開始有提出3種方案，最終我們決定將橋牌與rpg的某些性質結合創造出如上面報告介紹的RunesBlur。另外我們一開始的分工是先各自作各自所需要製作的class，但或許是大家是第一次做，因此沒有人提出其他的分工方法，並且每一次討論完都會詢問組員是否有問題，即使沒有問題，我們也會互相詢問，或有時我會直接詢問組員是否產生問題，並幫忙解決，接著當我們打算全部基礎class都設好後，再由林彧頎(我)、潘妤揚、杜佩真進行主要的垂直整合，並有陳于瑞學長協助幫忙，我覺得在過程中最困難的是，必須要在大家很繁忙的時間中，抽出時間討論、製作，還有我個人較為麻煩的部分，即為統整大家的成品，舉凡:程式碼、PPT、WORD我都需要整理，並進行製作，但還好在過程中有組員的幫忙，與盡可能的配合，我認為我們才能如此順利的完成!

接著我想談到程式部分，我在水平製作中主要負責玩家，並於之後有主要進行玩家、怪獸、卡牌、的垂直整合，另外也有幫忙進行遊戲畫面的設定，與故事設定，以及美編圖形的協助!

另外說明一下玩家的部分，玩家與怪物是由最基礎的gamer進行abstract class的設定，而會由繼承的Maingamer與Monster來對攻擊、等級…等需求進行實作，另外玩家的Maingamer還有Poker會額外幫助主角建立個別腳色(如:king、queen…等等)的來對這些設定進行運算與實作，而同時前面所提到的Poker與Monster皆會需要額外繼承Card來對應其後面實作所對應的不同腳色或怪物之間的資料庫設定!

最後以上這種種，會於GameLibrary進行純粹的資料庫建立，像是圖形、地圖、腳色、怪物設定等等…，最後會於RunesBlur進行實作!

**A1105549 杜佩真**

在這次的專題中，我和其他組員一起負責統整大家的程式碼，實作出我們的遊戲。因為是第一次和其他人一起分工完成程式碼，所以對我來說是一個蠻大的挑戰，實際在整合時也遇到了許多的困難，除了要讀懂別人撰寫的程式碼意思，還有要知道如何呼叫到實作的地方，在呼叫的過程也遇到了許多的問題，常常出現很多無法呼叫或是呼叫出來不是我們要的樣子的狀況，而我們只能不斷地換方式撰寫程式碼，以達到我們的需求，做到最後時，也深深的感受到牽一髮而動全身的感覺，常常在我們加了一個額外的東西時，整個遊戲畫面就變得很不一樣，而且只要報錯，在debug的過程中也是困難重重，有很多要注意的小細節，但在這次實作的過程中，也讓我更加了解物件導向的概念。

**A1105545 潘妤揚**

很慶幸我是在這個組別，因為有大家的幫忙這個作品才能成功的完成。在Runesblur當中，我是負責裡面地圖的部分，所謂的地圖其實是人物在尋找怪物的那個頁面，因此Map就是在定位每一個點，而Human，則是可以繼承Map的位置，進而達到人物可以在上面走動的效果。另外MapPage分別又繼承了Page、Human，繼承Page是因為地圖也需要音樂的引進，Human是因為main呼叫的時候不想讓使用者看到Map的內容，所以只會讓使用者呼叫MapPage的東西，丟進去的東西之後會連結Map的函式進行設定。但是這些部分做好之後還需要大家協助一同整合，在整合的過程中我們可以說是沒日沒夜，不停地在想辦法要怎麼把大家四散的程式碼拼湊在一塊，雖然或許有些人的程式碼有小幅的被修改，但是以大方向而言，我們所描繪的藍圖仍是一致的。最後有成功做出來真的無敵開心，感覺努力有所回報，振奮人心。真的非常感謝團隊的大家，有你們才會有Runesblur，謝謝你們。

**A1105531王重鈞**

在初始階段，我們是想利用C++的SFML函式庫來進行遊戲的圖像化，但學著學著似乎發現要從零開始學難度似乎有點大，花了很多時間在看官方文檔，釐清函式回傳的類型，而且有大量的比較運算子(::)，無法直觀的表現出來該物件、函式的具體作用。最後我們選擇不使用圖形化的介面，轉而在終端機上面用字元繪畫。

這次物件導向期末專題，我負責怪物的類別宣告，由於我的部分較簡易，只需處理怪物的名稱、數值、掉落經驗等基礎設定，仍有心力可以幫助其他組員，我便一起想卡片類別該如何實作，從該如何呼叫使用的卡片，再到如何修該玩家、怪物的私有變數(基礎數值)。

統合階段是我覺得整份專題最難的部分，因為要結合每個人撰寫程式碼的風格，使用公共的成員函式進行物件資料的增修，無法像自己一個人那樣用著自己的邏輯下去寫程式。下一次遇到需要多人協作的專題時，得在前期擬定好想要實現的功能，呼叫方式，甚至是類別中成員的權限。

**A1105509磨文傑**

原本我們這組打算使用SFML製作專題的遊戲，但SFML似乎有點複雜，短時間內難以做出甚麼成果，因此只好遷就於dos小視窗。製作專題的過程中，我也與其他組別的同學互相討論和學習，覺得我們的成果似乎略遜一籌，可知還有許多的進步與學習空間。

我覺得這次專題的困難之處在於「和他人一起寫程式」，在毫無軟體工程的概念之下，不斷地切磋、討論、磨合。對於我負責撰寫的卡片類別程式碼，在最後關頭時，我發現可以大幅度優化，感謝組員願意讓我修改程式，最終大約減少了300～350行。

我認為我們這組有許多問題需要檢討。首先，劃分哪個類別該有哪些data member以及member function不夠果斷，導致冒出多餘的程式碼；再者，對於物件導向的特性不夠了解，以致於在程式的溝通上有所障礙；最後，由於我們這組的分工合作偏向水平整合，因此幾乎沒有垂直整合，經過多次的討論，總算成功整合專題。分析檢討，讓我能夠未雨綢繆未來的挑戰。

總而言之，我覺得有點誤打誤撞地成功完成期末專題，雖然過程不太順利，但能完成就是好事，謝謝每位組員的貢獻與互助。

**A1095547陳于瑞**

這次的物件導向專題我們這組做的是遊戲，我負責的部分是遊戲總體構想、設計數值及程式技術支援，開發過程是有遇到不少困難，靠著與組員的溝通協調還有合作才能一一解決問題，在查詢程式相關資料的過程我也吸收了很多知識，原來程式還有這種寫法。遇到問題時會想自己解決，但常常還是需要集思廣益詢問組員意見才能處理問題，每次看到原本不能正常運作的功能經過數次改寫最後成功執行都讓人有很大成就感。透過這次專題我學到跟別人合作的方式，也讓我對物件導向有更深層的了解，體會到合作的益處與重要性。