

Program Design II



遊戲說明：

《PROGRAM DESIGN II》是由部政佑、江婕滢、黃若慈自主研發的一款全新回合制戰鬥RPG。遊戲發生在一個被稱作「國立成功大學」的幻想世界。在這裡，被選中的人們將會被授予「Java大法」，獲得物件導向之力。你將扮演一位名為「大學生」的神秘角色，在自由的旅行中邂逅性格各異、能力獨特的同伴們，和他們一起擊敗強敵、找回失散的學分——與此同時，逐步發掘「程式設計二」的真相。

Requirements

Java == 1.8.0_401 (Oracle JDK) / 1.8.0_402 (OpenJDK)

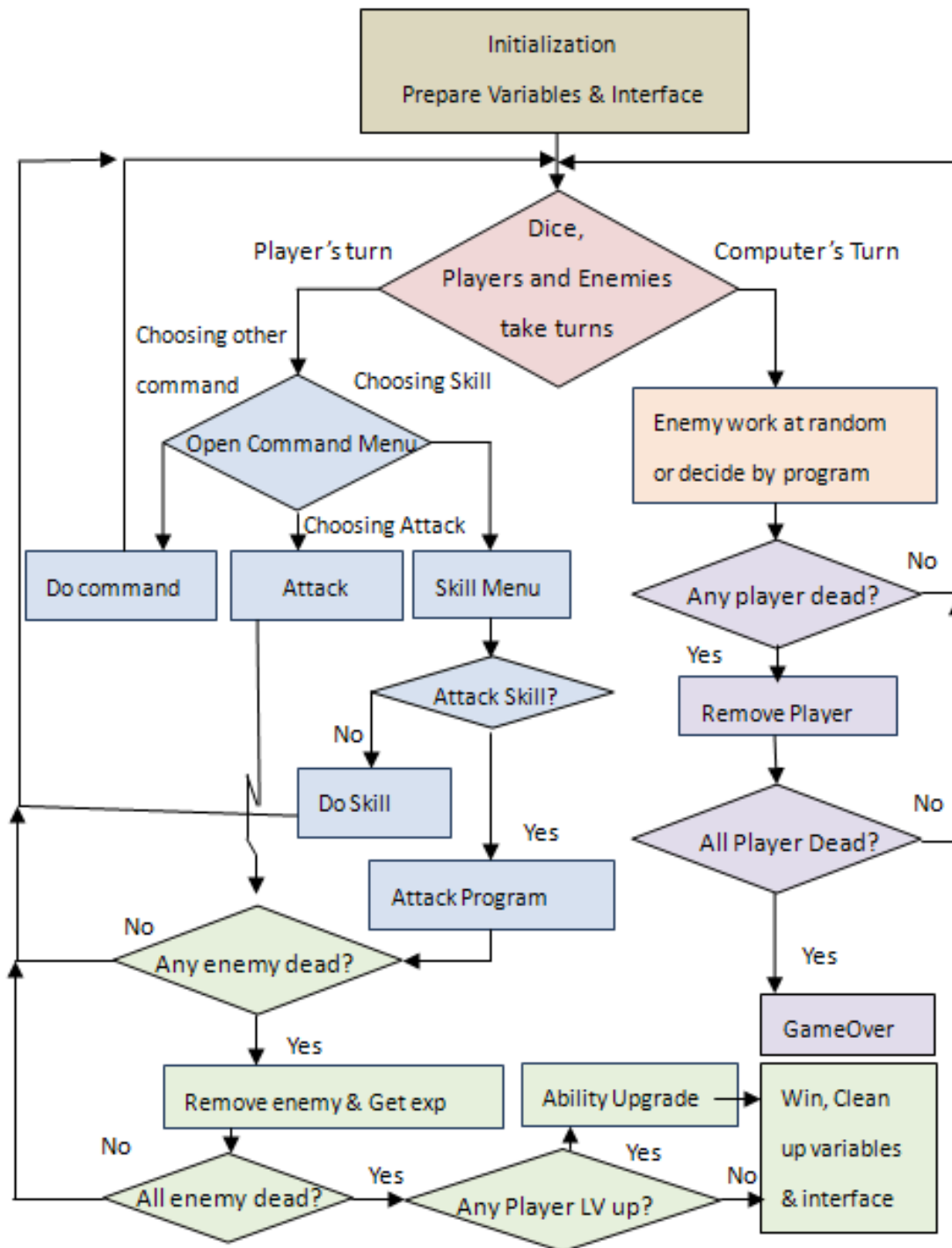
遊戲簡介：

跟敵人戰鬥時消耗MP使用技能、透過回答成功Java教材題庫的問題來對敵人造成傷害。

遊戲玩法：

《PROGRAM DESIGN II》為回合制戰鬥RPG，戰鬥系統流程圖如下：

畫面說明 (Flow Chart of a Battle System):



過關條件：

- 將敵人的HP全數歸零。

失敗條件：

- 我方角色HP全數歸零。

遊戲操作：

W:選項向上。

A:選項向左。

S:選項向下。

D:選項向右。

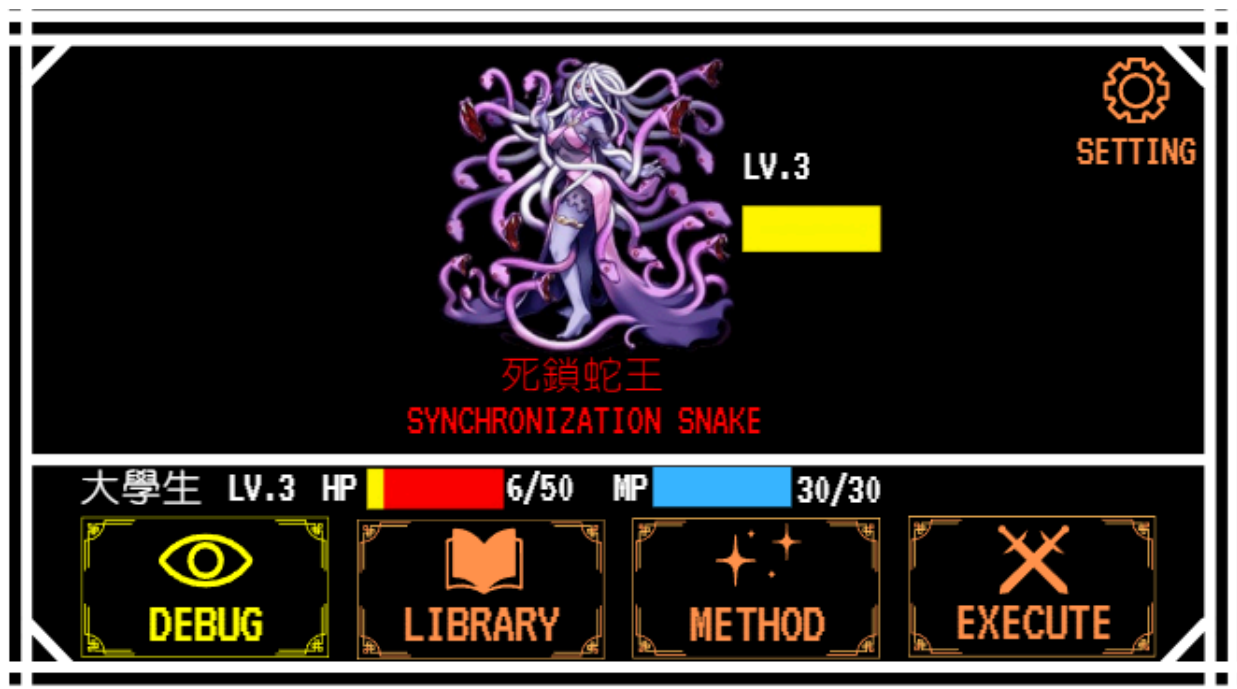
滑鼠左鍵:選擇選項。

遊戲機制：

我方回合(Player's turn):

當輪到我方回合，使用WASD在Command Menu中移動，當移動到Command Menu的某個選項，那個選項會顯示黃色。此時點擊滑鼠左鍵便可觸發該選項的功能，當使用完該選項的功能回合便會結束。

畫面說明(Battle screen diagram):



Command Menu:

1. Skill Menu:

- **DEBUG**: 又稱偵錯，戰鬥選項之一。在程式開發中，Debug表示調試過程，通過這一過程可以查看程式的運行狀態和變數值等。在遊戲中代表消耗一回合在遊戲中查看敵人的各項數據(HP、ATK、DEF、技能)詳細內容將在後續補充。。
- **LIBRARY**: 又稱書庫，戰鬥選項之一。Library在程式設計中指的是函式庫，這些通常是一組已編寫好的功能，可以被其他程序調用來增加功能或改進性能。在遊戲中代表玩家的道具庫，可以使用道具來獲得某些優勢或補正，詳細內容將在後續補充。
- **METHOD**: 又稱方法，戰鬥選項之一。在程式設計中，Method 指的是類別中的一個功能實現，用於完成特定的操作。代表該角色使用的技能池，可消耗MP使用技能對敵人或我方產生特殊效果，詳細內容將在後續補充。

使用完Skill Menu的選項後，該我方角色的回合便會結束。

2. Attack:

- **EXECUTE**: 又稱執行, 戰鬥選項之一。Execute在計算機術語中常用來描述執行一段代碼或命令。在遊戲中代表對敵人發動攻擊。

在玩家選擇發動攻擊時, 必須先回答題庫的問題, 並透過答對題數的多寡進行傷害判定, 答對越多題目就能累積更多傷害。

題庫: 由開發者出題, 出自國立成功大學程式設計二課程的教材、作業以及考試。

題庫的題目會分成三個難度: ★、★★、★★★。

★是簡單的觀念題、★★是中等的判斷題、★★★是高難度的挑戰題。每個難度都有至少數十題的份量。當發動攻擊時便會隨機從題庫抽題, 但等級越高的敵人題目的平均難度也越高。

此外, 每位敵人的出題範圍會跟等級有關, 每提高一個等級就會多一本講義的內容。

舉例: 由於小蟲(Small Bug)屬於LV.0的敵人, 因此出題範圍只有包含*PD2 Lecture 0 - Course Overview*以及*PD2 Lecture 1 - Introduction*; 但異常皇帝(Exception Emperor)屬於LV.3的敵人, 因此出題範圍包含*PD2 Lecture 0 - Course Overview*一直到*OOP - Representation of OO Design and Powerful Libraries/Functionalities*, 並且越後面的章節, ★★★的題目就越多。
最後, 有些敵人比較特殊, 因此有獨立的專屬題庫。

使用完Attack後, 該我方角色的回合便會結束。

基礎傷害公式(若DEF為0的情形):

- **ATK**: Attack, 攻擊力, 越高代表能造成越多傷害。
- **HP**: Hit Point, 生命值, 或也有Health Point的說法。代表死亡前能最多受到多少傷害。
- 當B被A攻擊, $B \text{的HP} = B \text{的HP} - A \text{的ATK} * A \text{答對的題數}$ (若A為敵人則是1)

傷害減免公式(若DEF大於0的情形):

- **DEF**: Defence, 防禦力, 可以抵銷ATK的數值, 越高代表越難對其造成傷害。

- **DMG**: Damage, 實際傷害, 也就是ATK扣除傷害減免後, 最終造成的傷害值。
- 當B被A攻擊, A對B造成的DMG = A的ATK - B的DEF
- B的HP = B的HP - A的DMG * A答對的題數(若A為敵人則是1)
- 當B的DEF比A的ATK高或相同時, A對B造成的DMG固定為1。

敵方回合(Computer's turn):

- **Skill**: 每位敵人都有其專屬技能, 當使用完技能該敵人的回合便會結束, 詳細內容將在後續補充。
- **Attack**: 敵人將對一名或多名我方角色發動攻擊, 扣除我方角色的HP, 發動完攻擊該敵人的回合便會結束, 詳細內容將在後續補充。
- 敵方的傷害公式與我方相同。

遊戲術語:

- **強健性(Robustness)**: 是指軟體在面對預期之外的錯誤、輸入不確定性和異常條件時, 能夠維持一定程度的功能運作的能力。遊戲中代表玩家的HP(生命值), 當受到敵人攻擊歸零時則遊戲失敗。
- **適應性(Adaptability)**: 是指它能夠在不同的環境或情況下正確運行。這包括能夠處理不同的輸入、操作系統、硬體設備等。遊戲中代表玩家的DEF(防禦力)跟ATK(攻擊力)。
- **重用性(Reusability)**: 是指軟體的一部分(例如, 類別或方法)可以在不同的程序中使用, 而不需要進行大量的修改或重寫。遊戲中代表玩家的MP(魔力量), 可消耗來使用技能對敵人造成傷害。
- **學分(Credits)**: 為主角必須獲得的強大力量。代表玩家的LV(等級), 從最低的LV.0到最高的LV.3, 每提升一個學分便可獲得全方面的提升。
- **錯誤 (Errors)**: 為玩家每回合進行攻擊時所需解決的問題數目, 通常等級越高的敵人錯誤會越多, 難度也越高。
- **封裝(Encapsulation)**: 物件導向之力之一, 為主角的METHOD技能, LV.1時解鎖。它賦予程式設計師自由實現一個元件的細節, 而不用考慮其他程式設計師寫的元件的內部實現方式。在遊戲中使用該技能可啟動封裝護盾, 隔絕指定敵人的核心技能。在接下來的一個回合內, 敵人無法使用其技能。
- **繼承(Inheritance)**: 物件導向之力之一, 為主角的METHOD技能, LV.2時解鎖。在遊戲中使用該技能可臨時繼承敵人的一項隨機技能並立即使用。

- 多型(Polymorphism): 物件導向之力之一，為主角的METHOD技能，LV.3時解鎖。它允許我們使用一個共同的介面來操作不同的對象。在遊戲中使用該技能可以一次使用兩個不同夥伴的技能。

地圖說明：

- 寬 1000 pixel; 高 600 pixel。
- 每格 25 * 25 pixel, 可放置一個物件。

物件設定：

《PROGRAM DESIGN II》的主要物件包含：敵方角色、我方角色、NPC、武器、防具、道具，以下將逐一說明。

敵方角色：

畫面說明：

數據需求	英文名	類型	數據範圍	備註
怪物ID	monsID	int流水號		唯一
怪物名稱	monsNam	string	256字元	
怪物等級	monsLV	int	1 ~ 100	目前僅開放到100
怪物類型	monsType	bool	0或1	0：小怪，1：Boss
怪物血量	monsHP	int		
怪物攻擊	monsATK	int		
怪物防禦	monsDEF	int		

