

Fundamental Object Oriented Programming

Final Project

B07902037 蔡沛勳 B07902039 曾曄翔 B07902115 陳致元

目錄

1. 遊戲介紹及玩法
2. Class 間的互動關係 (UML)
3. 設計優點
4. 設計缺點
5. 使用素材
6. 分工

一、遊戲介紹及玩法

1. 遊戲介紹

我們設計了一款以棋盤進行的回合制戰棋遊戲，在 20*30 格的地圖中，玩家操控英雄以打敗敵人為目標。遊戲流程為由英雄先攻，依序進入以下四個階段：

準備階段：耐力回復為最大值，處理狀態效果並對狀態做更新。

移動階段：消耗耐力值進行移動，移動到目的地後可選擇跳過移動階段。

行動階段：可選擇技能進行施放，電腦會判定玩家技能是否使用成功。

結束階段：回合結束，判定遊戲勝負是否分曉。

而隨後輪到敵人的回合，由電腦決定四階段的行為，雙方重複以上流程直到一方血量歸零，遊戲結束。

2. 遊戲玩法

移動階段時，玩家可使用 w、s、a、d 來進行上、下、左、右方向的移動，以 h 或空白鍵來跳過移動階段。

行動階段則按照 GUI 指示以 0 ~ 9 來施放對應技能。

技能列表：

共通技能：

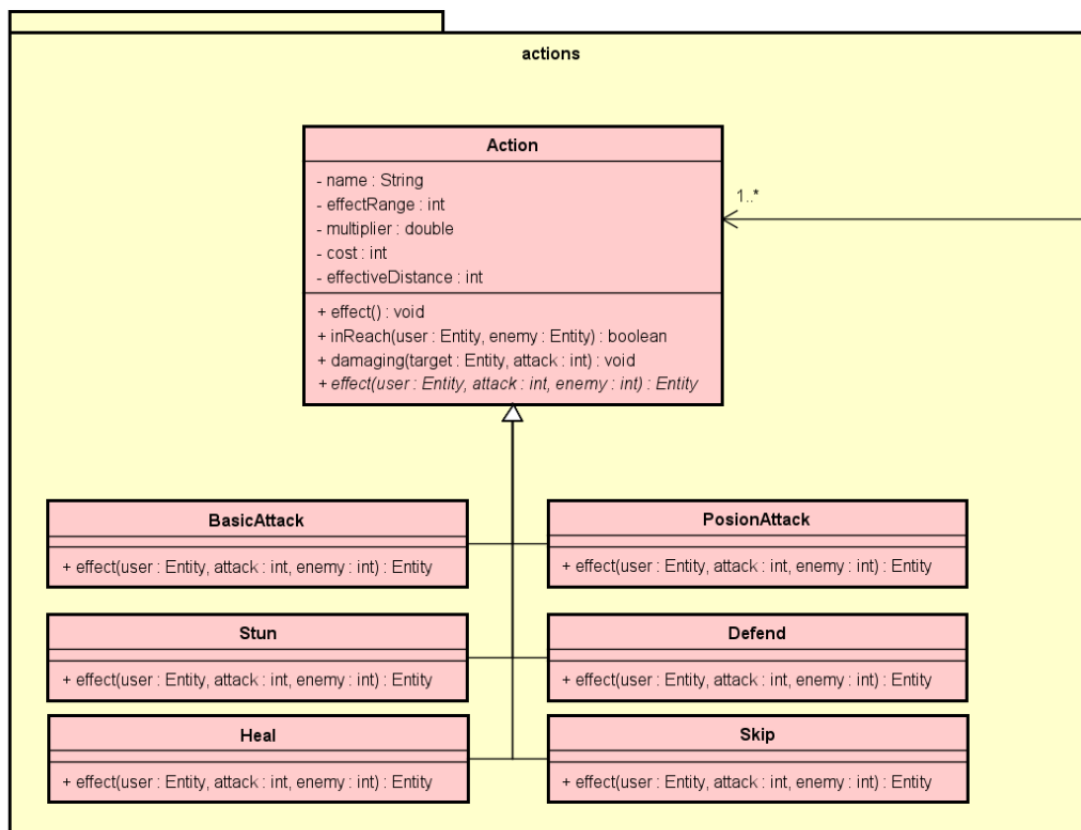
1. 放空 (Skip)：不施放技能
2. 基本攻擊 (Basick Attack)：距離 1 以內的敵方受到 1 倍攻擊傷害

英雄特有技能：

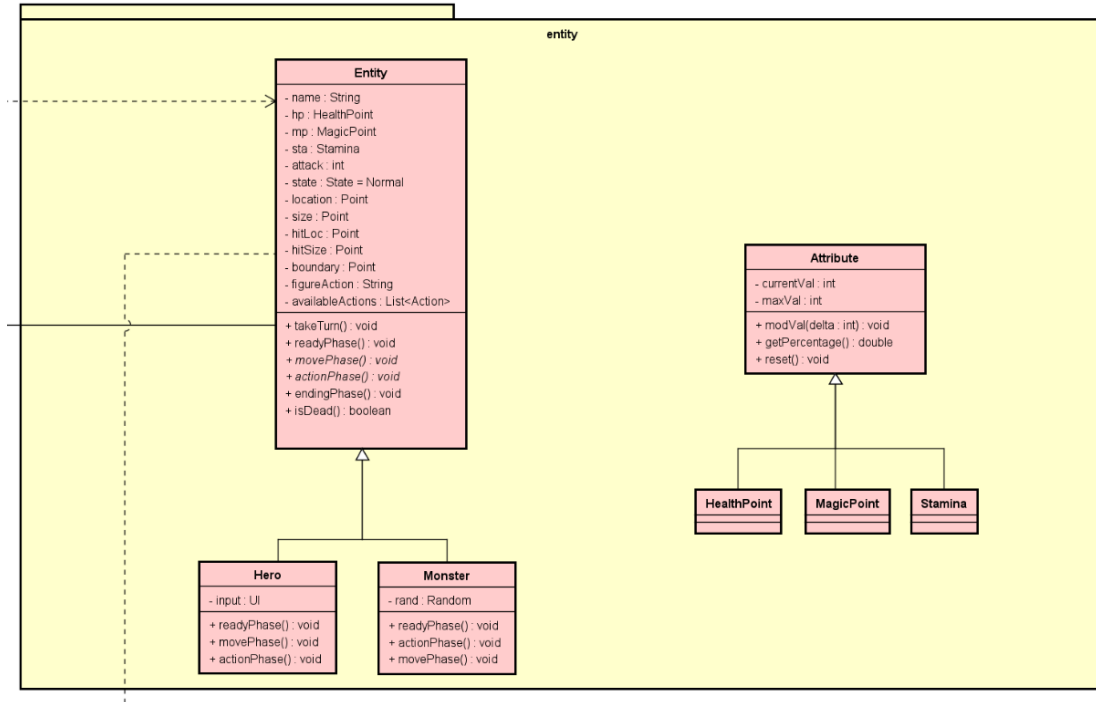
3. 毒害 (Poison)：距離 1 以內的敵方受到 0.25 倍攻擊傷害並中毒 (Poisoned)
4. 暈眩 (Stun)：距離 1 以內的敵方受到 0.25 倍攻擊傷害並暈眩 (Dizzy)
5. 治療 (Heal)：自身恢復 10 點生命
6. 防禦 (Defend)：自身進入防禦 (Defending) 狀態

二、Class 間的互動關係 (UML)

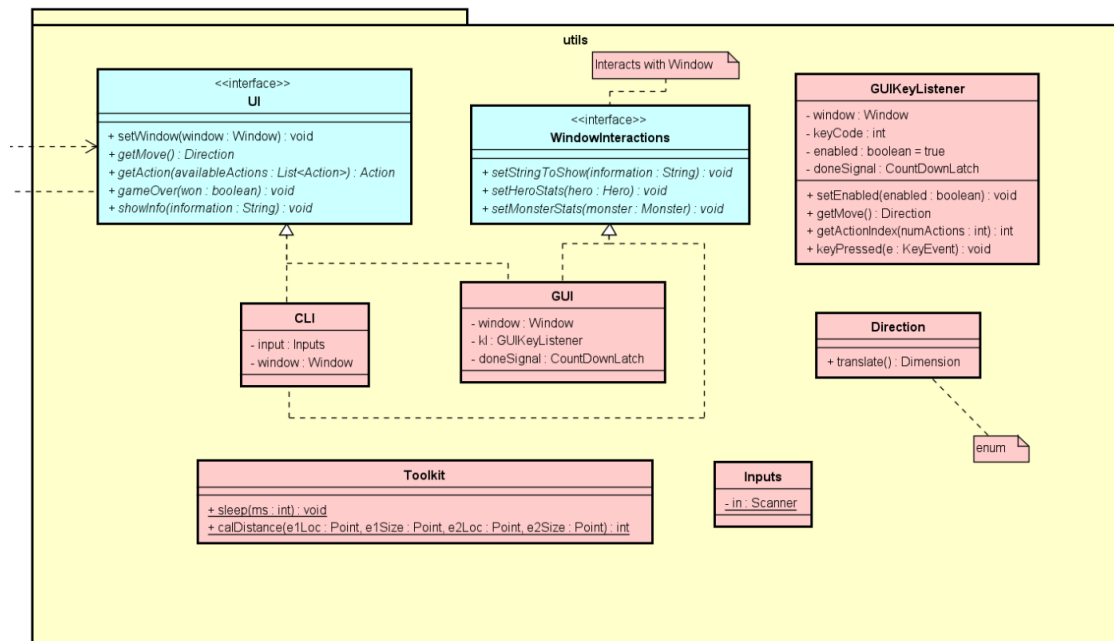
Action：



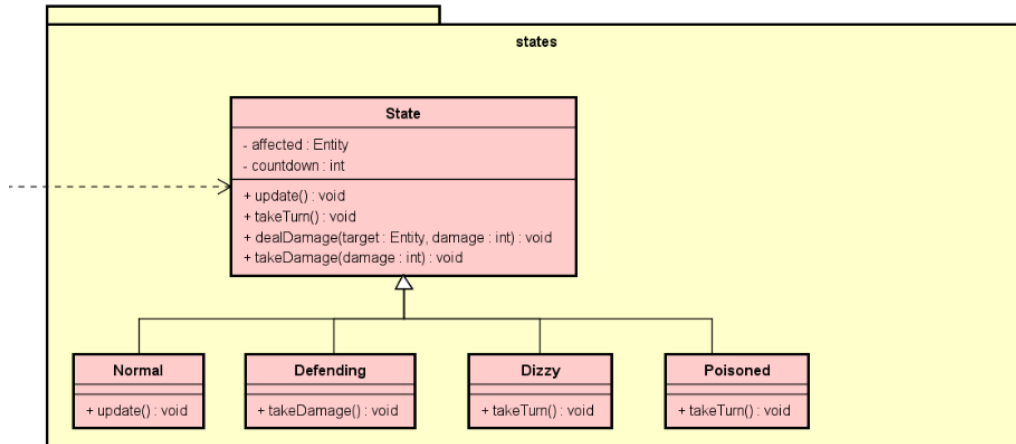
Entity :



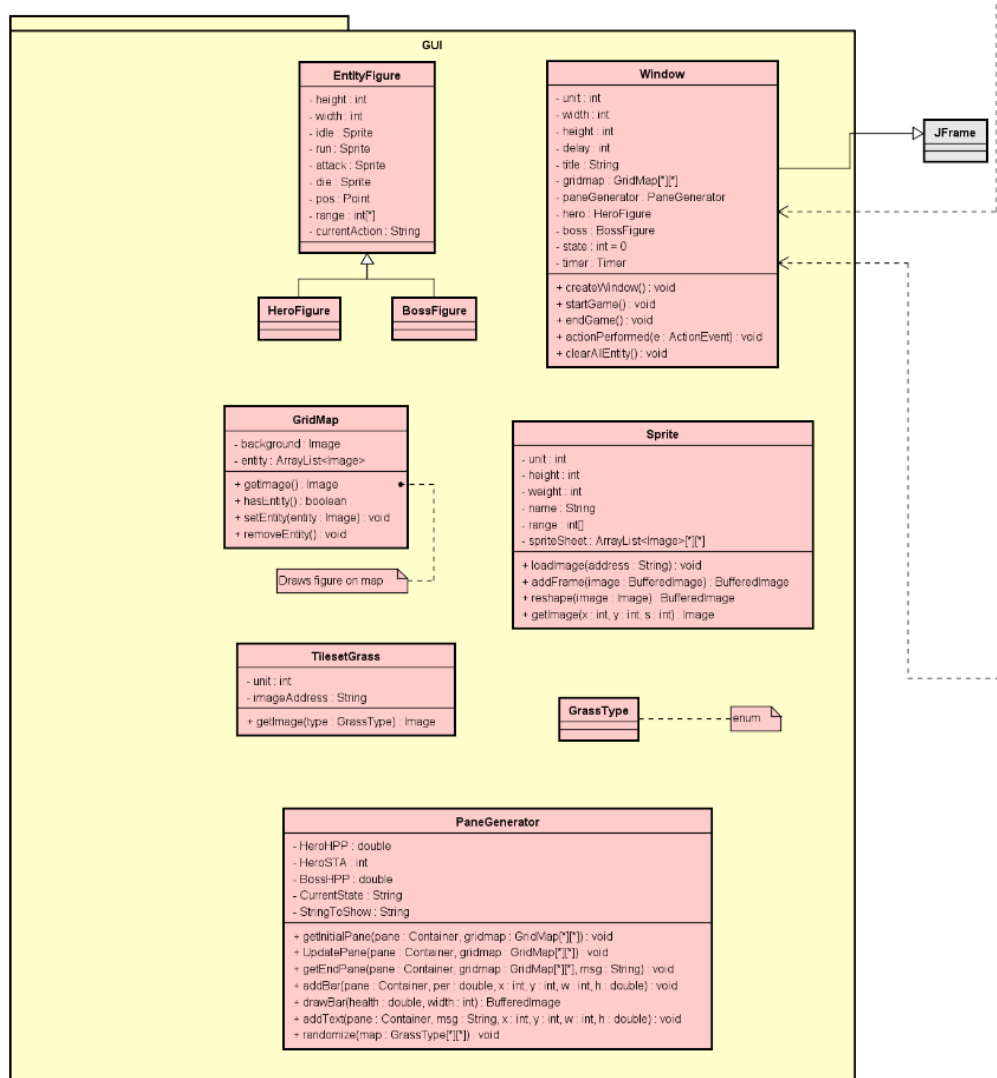
Utils :



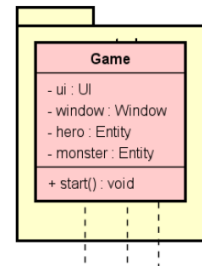
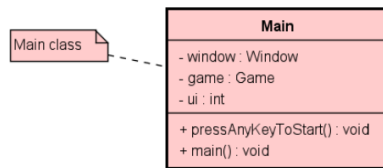
States :



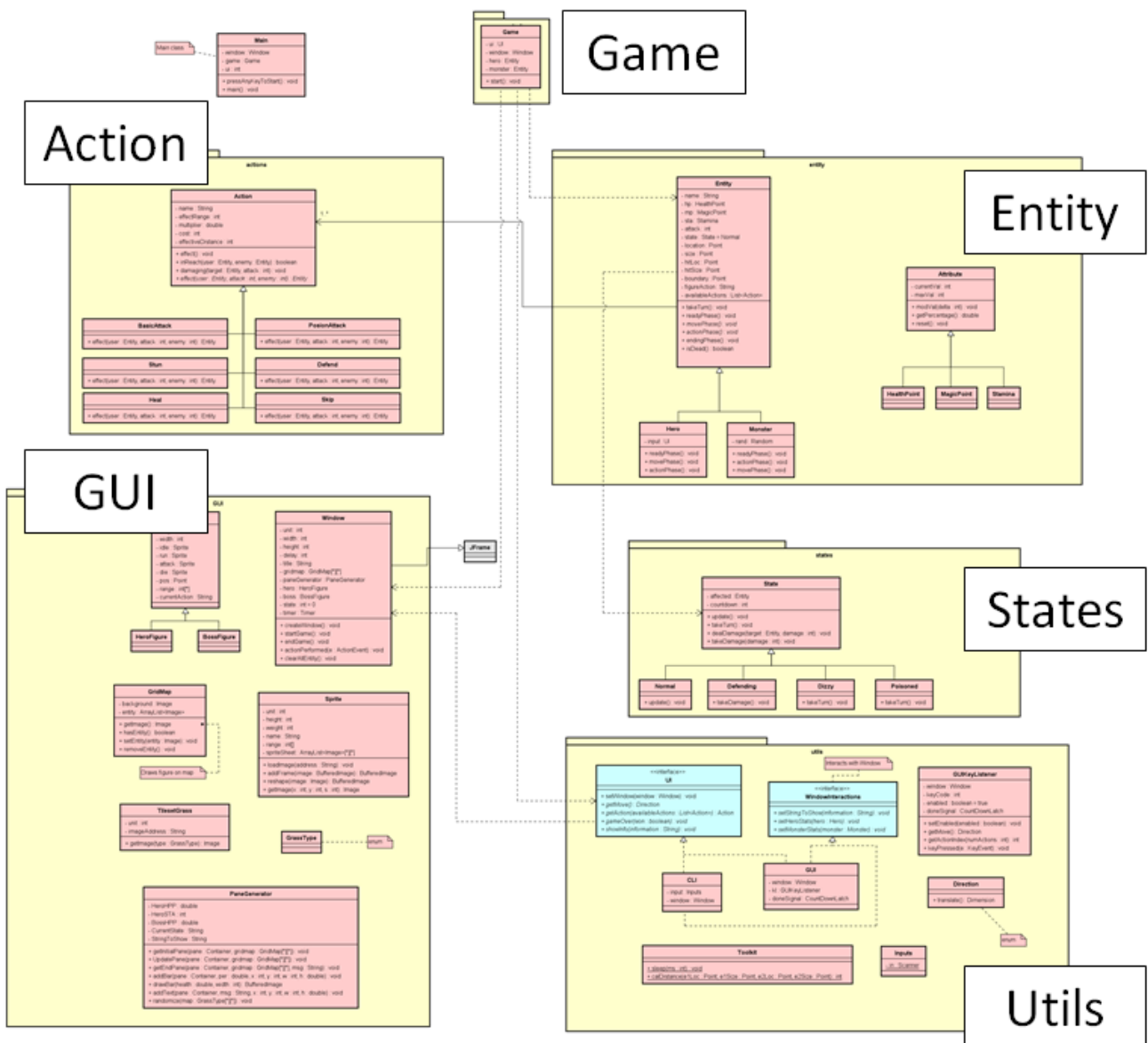
GUI :



Main 及 Game :



總和：



三、設計優點

1. 分割圖形與流程控制

實作時我們將圖形與流程控制分別管理，主要在相對應的輸入輸出與介面互動，因此我們能夠把圖形部分移除，遊戲仍能運作。

2. 技能與狀態

技能與狀態的實作容易擴充。只要符合 Interface 的要求，便可將新的技能及狀態加入並使用。

3. 可支援 CLI 輸入及 KeyboardInput

可選擇使用 command line，依照文字題示輸入文字進行遊戲；或是使用鍵盤，依照圖形介面上的簡化提示輸入按鍵進行遊戲。

4. 地圖顯示物的可疊加性

地圖上每一格 grid 於每一個 frame 重新繪製，並將該格上的所有圖片由背景至 entity 依序畫上，因此不必擔心 sprite 範圍過大導致破圖，日後若要增加地圖物件也不必擔心被覆蓋掉。

5. 像素風格與 java 本身輸出風格契合

以 graphics 依百分比繪出的血量表與文字風格與我們使用的素材圖片與背景契合，HUD 的設計也因此不會顯得突兀。

四、設計缺點

1. 受限於 sprite 限制

由於圖片素材皆為網路上找尋的，使用上便受限於現成素材。像是我們紀錄一個 sprite 的 hitbox 需使用三組座標，依序紀錄 sprite 的座標、hitbox 的座標、及 hitbox 的範圍，對移動的判定及範圍也因此受到限制。

2. 地圖設計的彈性及擴張性不足

無論是 Window 的大小顯示及地圖的大小，受限於畫出來的初始地圖，我們皆沒有將其做成可隨時更改的形式，而是直接寫死，這影響我們之後做設計時不時要重新規劃地圖以及 HUD 介面，以顯示最小限度的必要資訊。

五、使用素材

本次專題我們沒有使用到 java 外部的 package，而我們使用的圖片素材包含以下：

1. Hero's sprite：<https://rvros.itch.io/animated-pixel-hero>
2. Boss's sprite：<https://clembo.itch.io/bringer-of-death-free>
3. Tileset map：<https://cainos.itch.io/pixel-art-top-down-basic>

以上素材圖片，我們都儲存在 graph 資料夾中使用。

六、分工

遊戲設計及報告為三人共同討論並完成，以下以實作部分進行分工。

1. 蔡沛勳：找尋圖片素材，GUI Package內的 code 實作。
2. 曾曄翔：control/entities/states/actions/utils 部分實作。
3. 陳致元：整合 gameflow 與 GUI，Keyboard input 的實作，繪製 UML。

七、參考資料

1. GridMabLayout Introduction：
<https://docs.oracle.com/javase/tutorial/uiswing/layout/gridbag.html>
2. Sprite Introduction：<https://zetcode.com/javagames/movingsprites/>
3. 助教上課簡報：
https://docs.google.com/presentation/d/1GKWInQmnXNLm3WchXOUOrWWb6g0lsDh8JNJVIWeEt8I/edit#slide=id.gd7bf225fd9_0_537