

Object Oriented Software Design

– Final Project

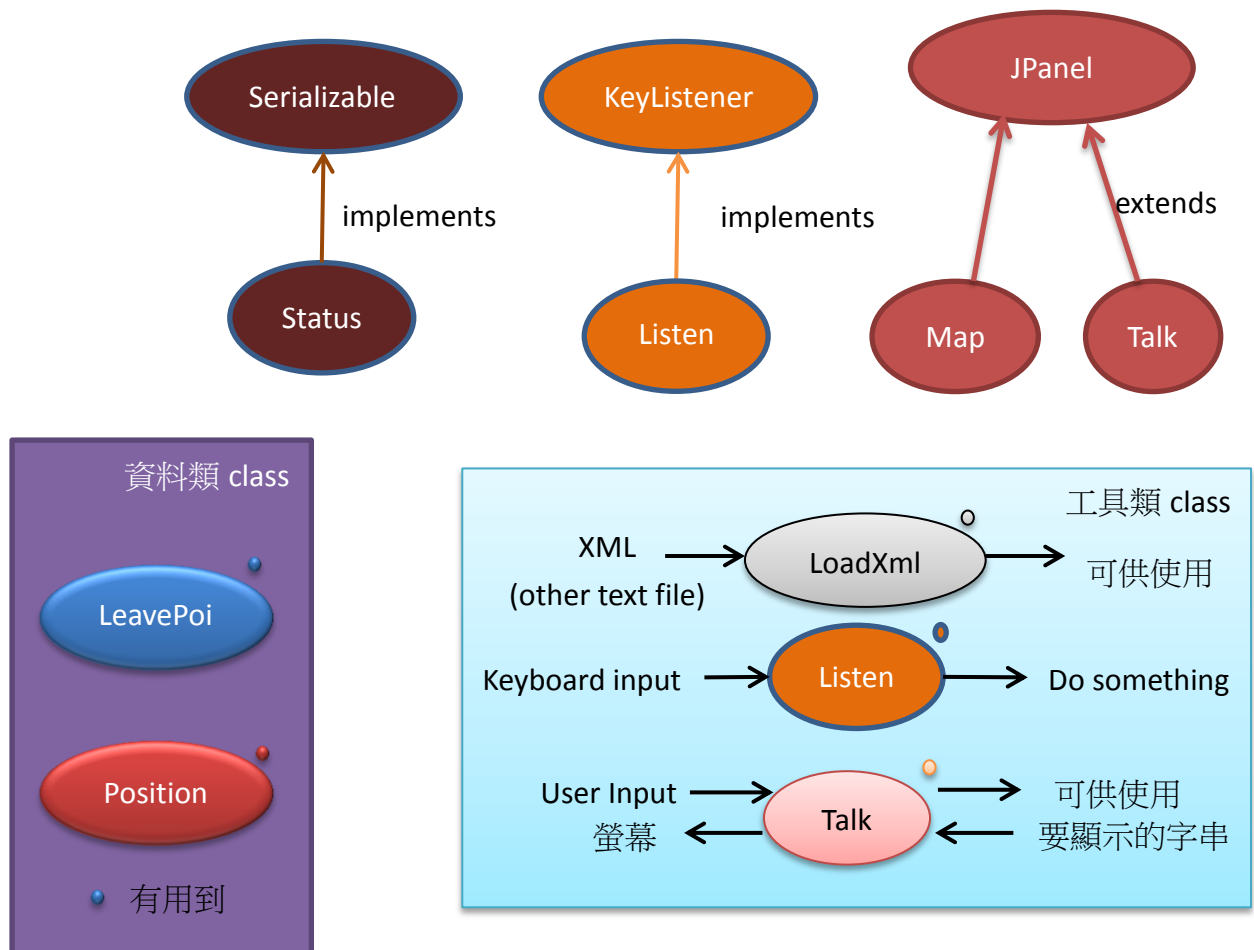
組員名單與分工：

B99902020 謝蕙蘭：畫圖、寫小遊戲、Report

B99902110 許毓軒：主架構、Report

B99902045 許恩碩：地圖相關、主要劇情、Report

類別關聯與介紹：



Game1: (對打遊戲)

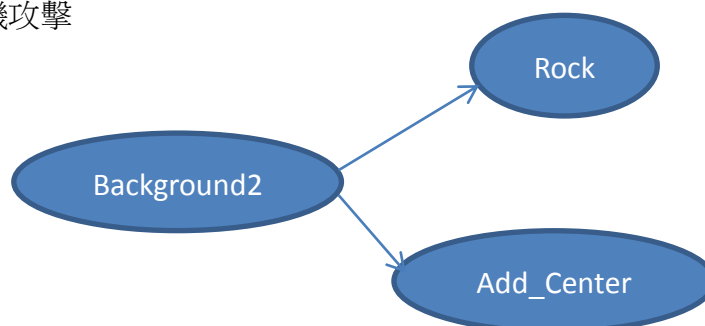
Background: 主要 class，判斷攻擊與移動，檢查遊戲的終止條件並回傳 win 或 lose 訊息

Mainrole: user 的移動，load 子彈與攻擊

Professor: 簡單的 AI 並隨機攻擊

Tear: 子彈物件

Weapon: professor 的武器

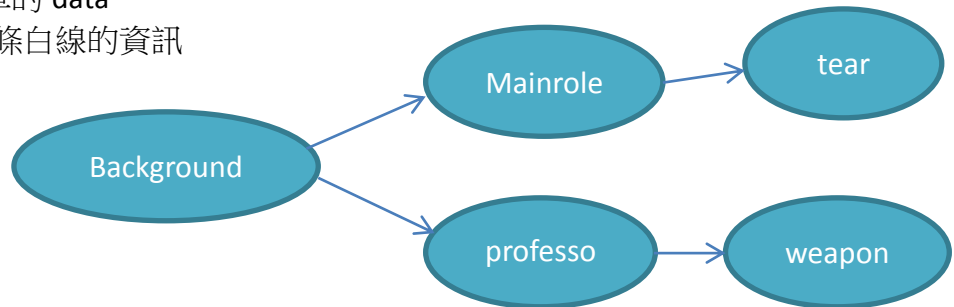


Game2: (穿梭車陣遊戲)

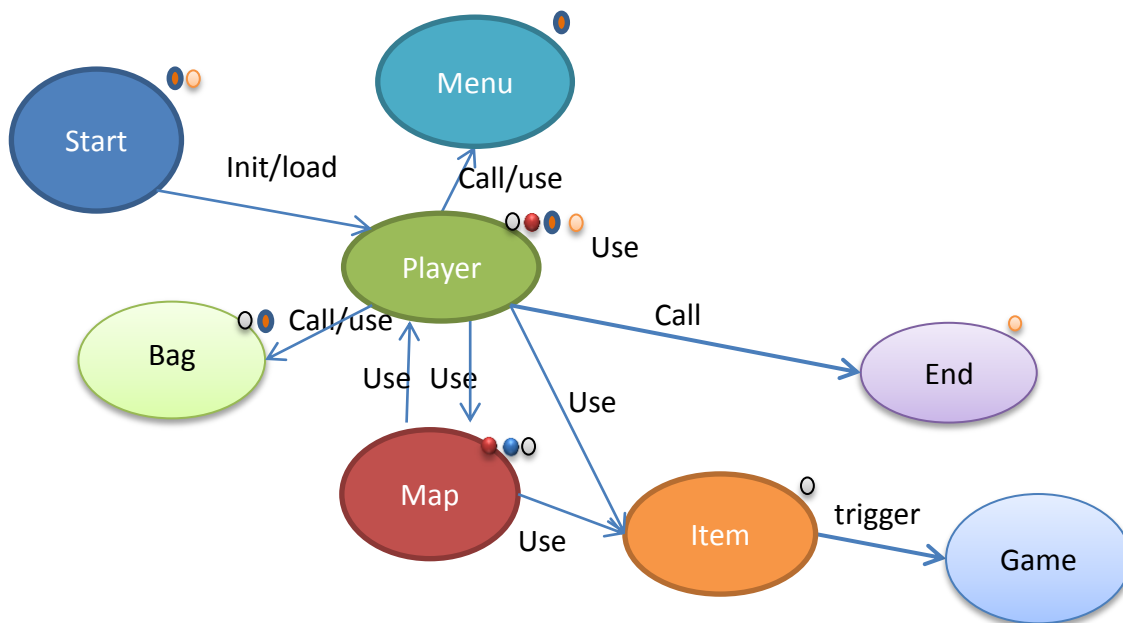
Background2: 主要 class，判斷車子的移動與畫白線，檢查是否相撞，回傳 win 或 lose 訊息

Rock: 車子物件及每台車的 data

Add_center: maintain 每條白線的資訊



整體



Start：遊戲的開始主畫面，控制 Player 要重新開始、載入進度或離開

End：判斷 Player 的 status 是否滿足結局的條件，並負責顯示結局畫面

Player：主要的控制 class，接收使用者的輸入，控制角色的行動，與其他 class 互動

Menu：選單列的 class，負責告訴 Player class 玩家選擇的是選單列中的哪一項，由 Player class 進行處理

Talk：負責控制對話框的 class，用來顯示訊息及取得玩家輸入的文字

Map：負責控制地圖相關的事還有一些畫面的繪製，其他元件

LeavePoint：儲存地圖中場景切換點需要資料的 class，包括有要該點在地圖的位置及，要切換到哪個地圖的哪個點。

Position：儲存地圖上 X,Y 座標的 class

Bag：道具欄，用來顯示道具，管理道具的使用情形

Item：地圖上的可互動元件，包含道具、NPC

Listen：鍵盤的 listener，在 Player 中用於取得玩家的操作動作

LoadXML：由 XML 或其他的文字檔中讀取資訊的工具 Class

Game：負責控制小遊戲的啟動、載入

設計優點：

1. 如 item、bag、map 內的物件資訊(如:NPC 對話、劇情觸發條件、物品在地圖的位置、圖案、map 的轉換點等)都是由 XML 檔或其他類型的文字檔中直接讀取，所以要在新增 map、item、劇情等，只需要更改 XML 後重新 load 即可完成，不需要更改程式或是重新 compile。
2. 由於小遊戲都是由 Game class 進行載入，要新增新的小遊戲十分容易。
3. 每個 class 之間的責任劃分都很清楚，不會有一件事情可能由兩個 class 都可以做到。
4. 玩家可採取的行為都被規定好了，不太可能讓程式崩潰。
5. 與 npc 的對話跳脫選項式的互動，雖然找尋正確的關鍵字可能較難，卻也多了一些趣味性，甚至可能意外找出一些設計者埋在其中的隱藏梗或劇情。
6. 劇情完善且多變化，不同的支線所可能產生的結局都不相同，甚至還埋藏一些無用、作為誤導或娛樂功能的支線。

設計缺點：

1. 有些元件之間的互動(如:Player 與 Map)是直接藉由互相傳彼此的 reference。
2. Item 中並沒有區分，讓 NPC 與地圖上的道具成為繼承 item 的不同 class，在使用上雖然方便，有時可能會有一些混淆。
3. 並沒有善用 extend 或 implement，導致之後的開發如果要新增新的元件時，可能會比較無所適從。
4. 小遊戲的畫面跟趣味性及和劇情的關聯性都還有很大的進步空間
5. 要了解 XML 的寫法以修改、新增新的道具、劇情，需要把有些 class 的運作方式看完，對於細節上（事件的觸發、地圖上道具的消失）才能夠掌握。

遊戲介紹：

遊戲執行說明

直接執行資料夾中的 final.jar，或輸入 java -jar final.jar，但記得，該 jar 檔需與 image、Game、mapInfo、XML 放在一起

操作按鍵介紹

↑ ↓ ← → - 控制遊戲人物
ESC - 打開選單
ENTER - 與地圖上的道具、NPC 互動，選項的選擇

選單：

選單中可以選擇開啟道具欄、存檔、讀取、離開（回到主畫面）。

而在道具欄中以 ENTER 使用道具，可能會得到該道具的說明、提示，在滿足某些條件下，該道具甚至可能和其他道具轉變為新的道具，比較特別的是，當面對著某些地點或 NPC 使用道具，可能會將該道具給予該 NPC 或是觸發些特殊的事件。

互動

比較特別的是，除了與 NPC 進行對話外，在某些情況下，NPC 會詢問你問題，你需要以打字的方式進行回覆，只有回答到關鍵的字詞，才有辦法推進劇情或得到有用的資訊。

劇情

在劇情方面總共有五個結局，還有加上些純屬娛樂（誤導）的支線，玩家可以由與 NPC 的對話、道具的提示、道具與 NPC 的關聯找出要怎麼做，才能夠將劇情進展下去。

我們設計的 RPG Game 主要為劇情取向，在對話設計上花了不少心力，往往線索就藏在對話或道具中，玩的時候需要細心的閱讀並加上一些想像力就能輕鬆過關。

這裡概略以圖介紹一下我們的主要劇情走向、流程。

