

FOOP Final Project Report

B06902023 林恩廷

B06902047 陳彥

B06902115 袁才育

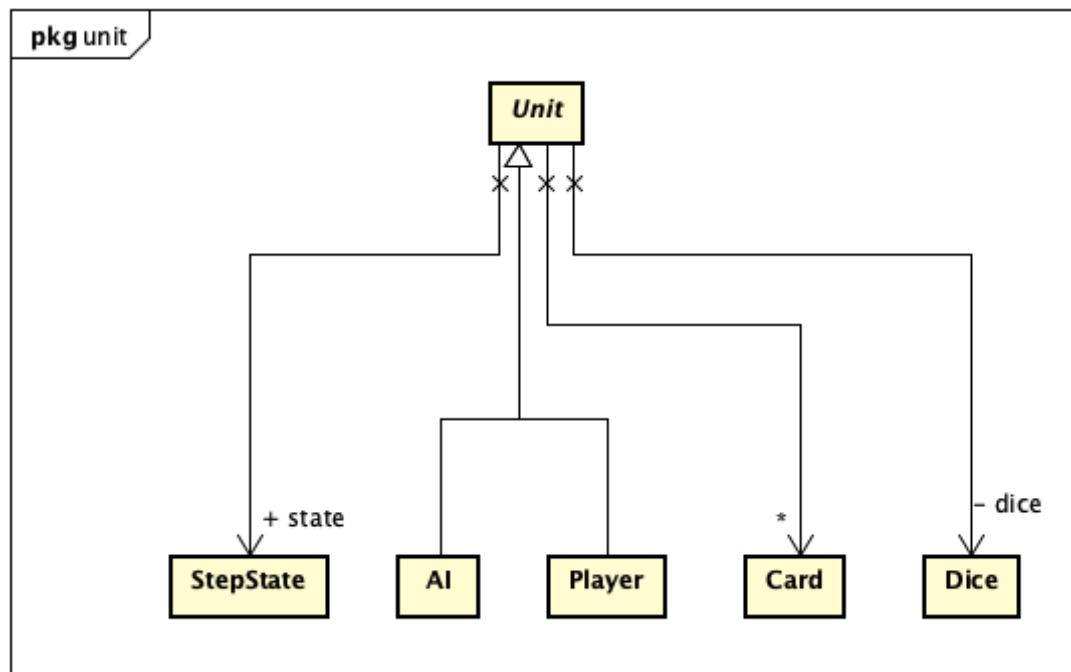
B06902119 張原豪

1. Responsibilities of the team members

- 林恩廷：Controller 銜接、部分 Map 功能
- 張原豪：遊戲 UI 設計、實作
- 袁才育：設計 model 中 unit、state、dice 等功能
- 陳彥：實作 model 裡的 map、location、card 等

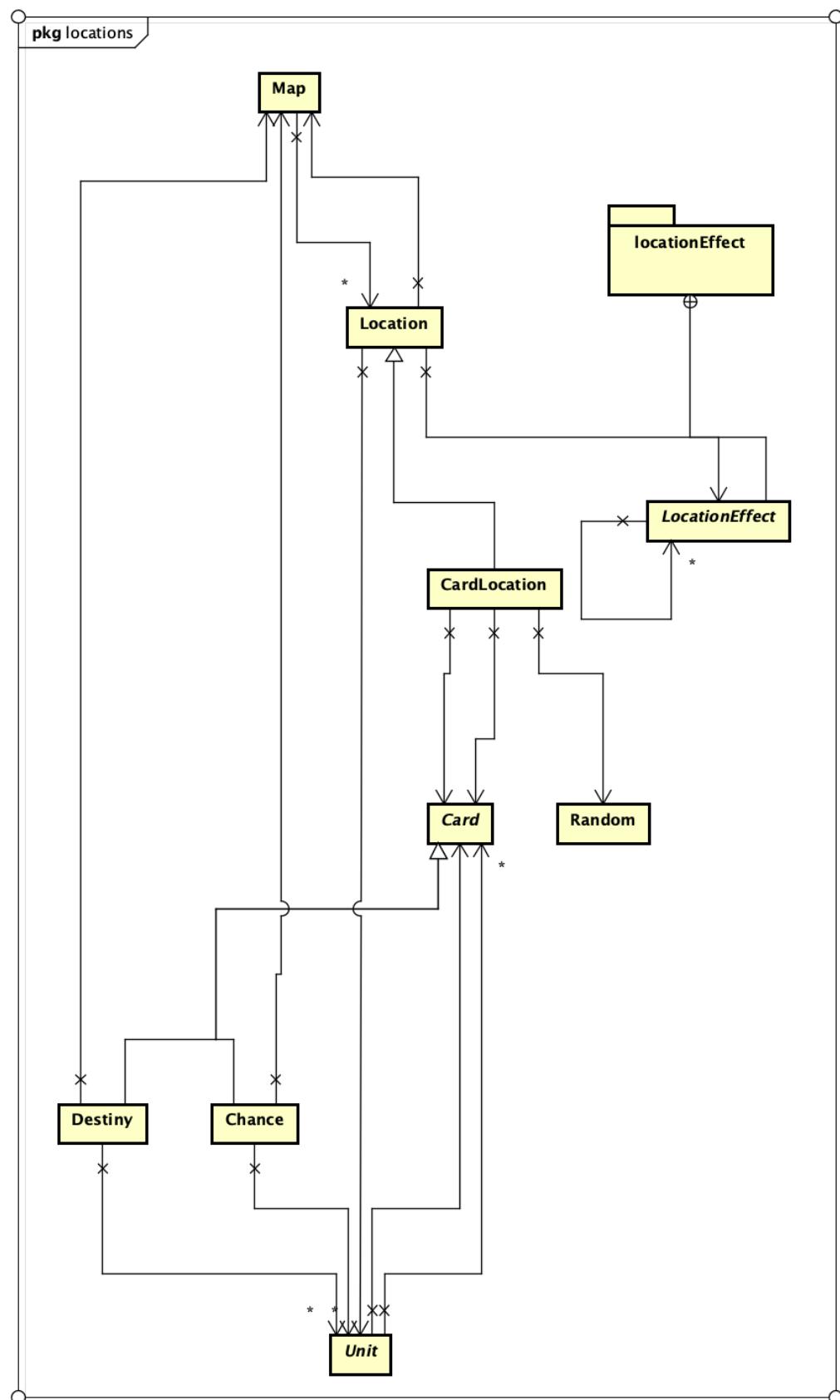
2. Class relations

- Model
 - unit



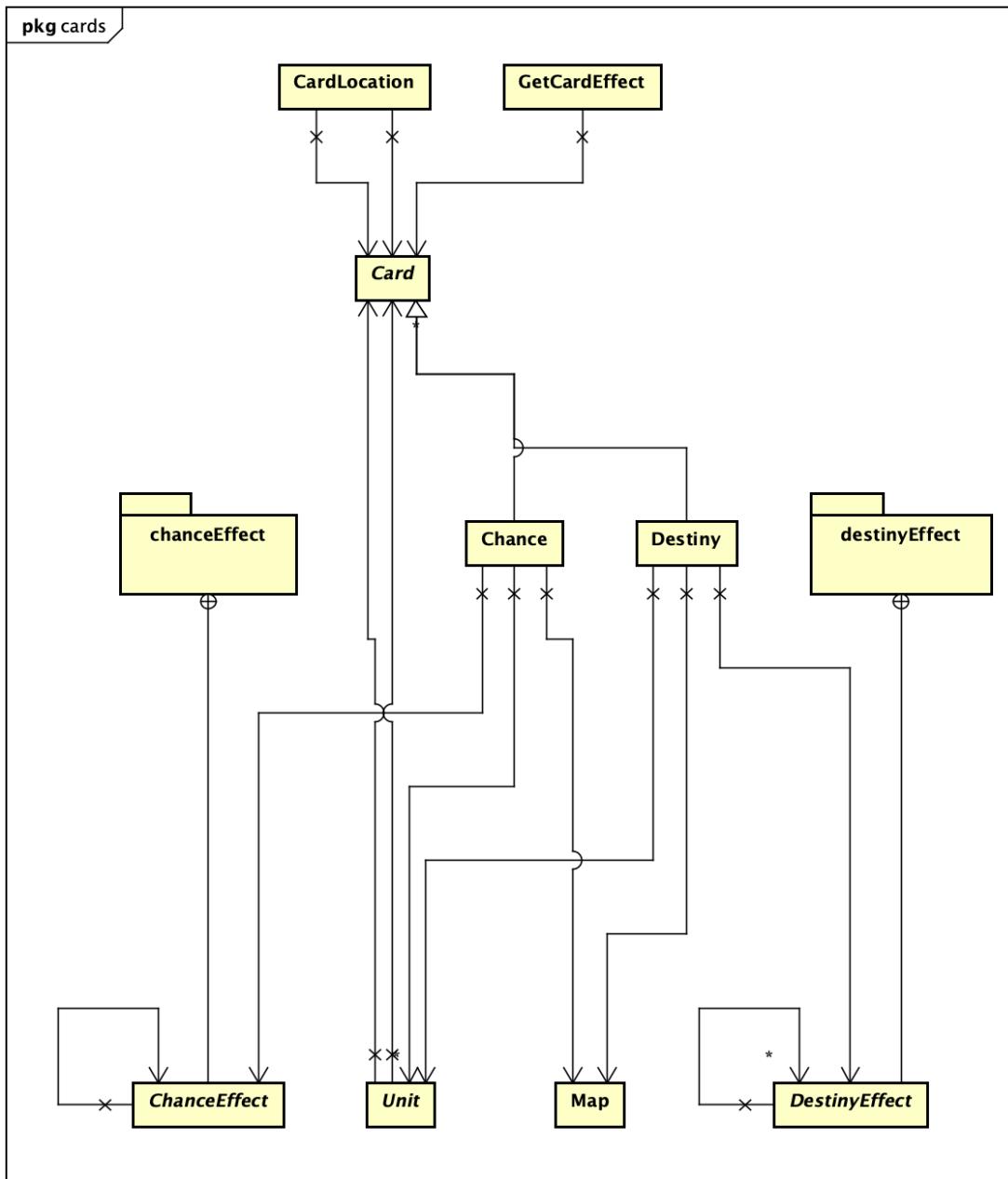
- Unit: 紀錄玩家的所有數據，如名字、數值、當前位置等。
- Player: 遊戲中的玩家，為 Unit 的擴充。
- AI: 遊戲中的電腦，自動設置勝利條件與性別。
- states
 - Stepstate: 紀錄目標玩家當前的步伐大小以及持續回合。
- dice
 - Dice: 遊戲中的骰子，返回投擲後得到的點數。

o locations



- Location: 記錄地點的所有資訊，包含 ID、名稱、是否在角落等
- CardLocation: 記錄需要抽卡的地點的基本功能，為 Location 的擴充
- Map: 包含一個放所有 Location 的 List
- locationEffect: 實作 Location 會給玩家帶來的影響，包含 NormalEffect 和 SpecialEffect

- cards



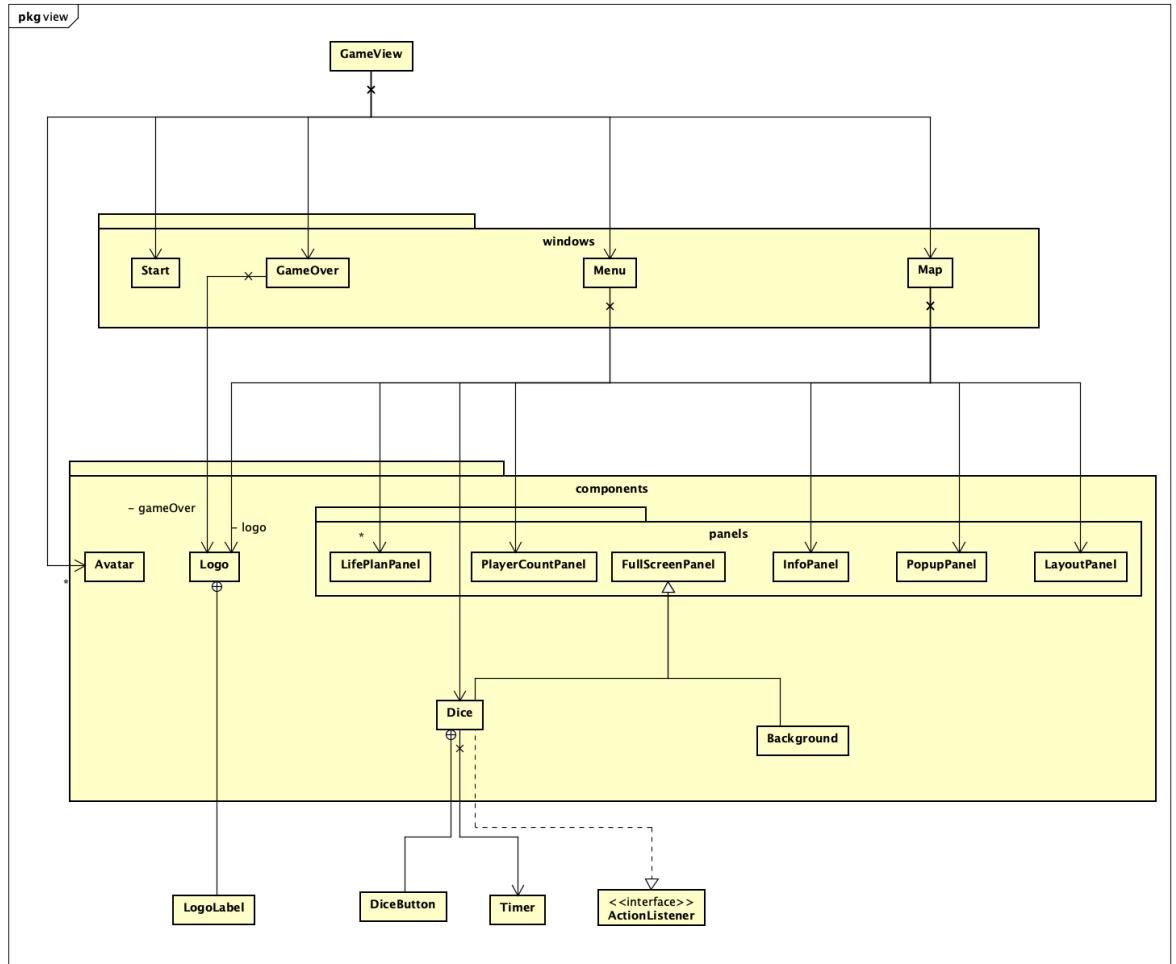
- Card: 記錄卡片的基本功能，如 takeEffect
- Chance: 機會牌，記錄卡片的數據，如會加減多少 happiness, knowledge, money 等，為 Card 的擴充

- Destiny: 命運牌，記錄卡片的數據，如會加減多少 happiness, knowledge, money 等，為 Card 的擴充
- chanceEffect / destinyEffect : 實作 Card 會給玩家帶來的影響，包含 NormalEffect 和 SpecialEffect

- View

- GameView Class

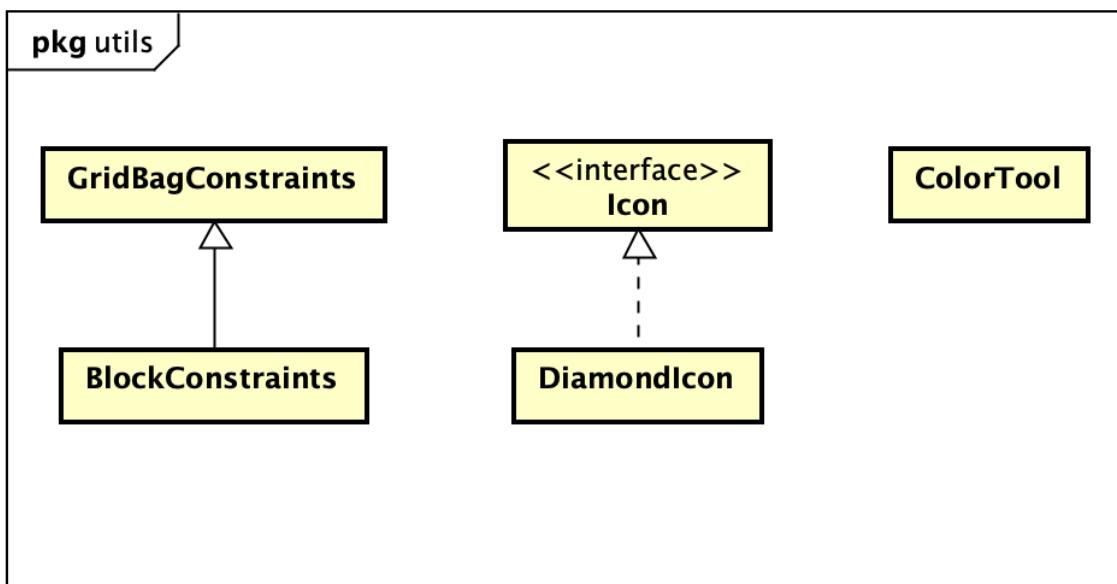
- 集合遊戲共有一些設定，如字體、畫面大小等等
 - 包含所有遊戲畫面，launch()後會依據遊戲流程更換頁面



- components

- Card: 遊戲畫面中央的機會命運區
- Location: 遊戲畫面的校園地區
- Avatar: 玩家頭像物件
- Background: 遊戲背景物件
- Dice: 遊戲畫面中的骰子物件，控制擲骰子動畫及是否鎖住骰子。
- Logo: 遊戲中的標題物件
- FullScreenPanel: 設置 Panel 為全螢幕
- PlayerCountPanel: Menu 中設置玩家人數的 panel，使用 JComboBox 實作

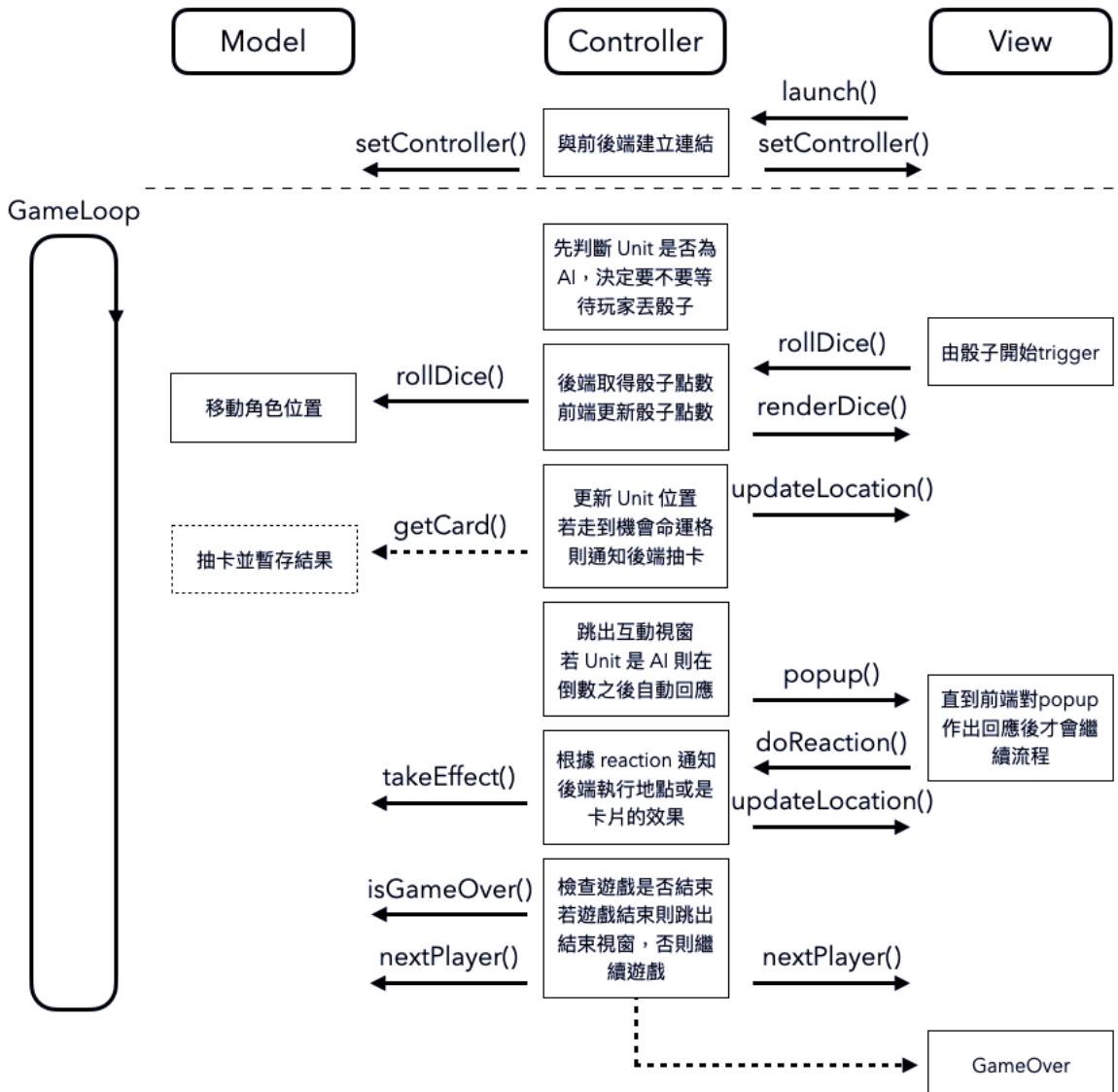
- LifePlanPanel: Menu 中玩家設定目標的 panel，每個 attribute 由一個 private class AttributePanel 管理。
- LayoutPanel: 設定 Map 中個物件的位置及其他屬性，從 config 中讀取設定檔
- PlayerInfoPanel: 顯示玩家目前的各項數值進度，輪到該玩家擲骰子時會改變顏色。
- InfoPanel: 遊戲顯示各種通知訊息的區塊
- PopupPanel: 遊戲的彈跳式區塊，玩家可與之互動進行選擇
- windows
 - Start: 遊戲的第一個畫面，包含一個 JButton。
 - Menu: 負責遊戲人數、玩家勝利條件等等設定，主要使用 components.panels 中的 PlayerCountPanel 和 LifePlanPanel
 - Map: 主要遊戲畫面，使用 components.panels 中的 InfoPanel、PopupPanel 和 LayoutPanel，還有一個 Dice。
 - GameOver: 遊戲結束畫面，用 components 中的 Logo 及 JButton 組成
- utils



- BlockConstraints: 簡化 GridBagConstraints 的設定
- ColorTool: 調整顏色深淺、透明度
- DiamondIcon: 菱形花邊

• Controller

- 由於 Controller 的 package 只有兩個 Class，一個為處理事件的 Timer 子類別，另一個是 Controller 主體，因此以下透過遊戲流程的邏輯來解釋 Controller 的工作。



3. Advantages of the design

- 採用 MVC 設計模式，以利於程式整體的開發與組員之間的分工
- 由於遊戲整體動畫部分皆可以由簡單的 timer 完成，並不需要做到像是 FPS 遊戲一樣不斷刷新畫面，因此最後採用 Event Driven 的設計方式，藉以避免多餘的運算
- 遊戲中的文本部分都是從 config 讀入，將來需要增加其他語言時只需要新增對應的 config 再加上排版即可完成
- 在執行流程中將不可互動的物件鎖定，減少來自使用者的不確定因素

4. Disadvantages of the design

- 由於採用 Event Driven，因此若未來想要做更複雜的動畫效果，就需要重新架構 controller 的部分

- 有些部分並沒有寫到非常抽象，因此若要加入變化幅度較大的異動(臨時加入互動小遊戲等等)時會變得較為艱難

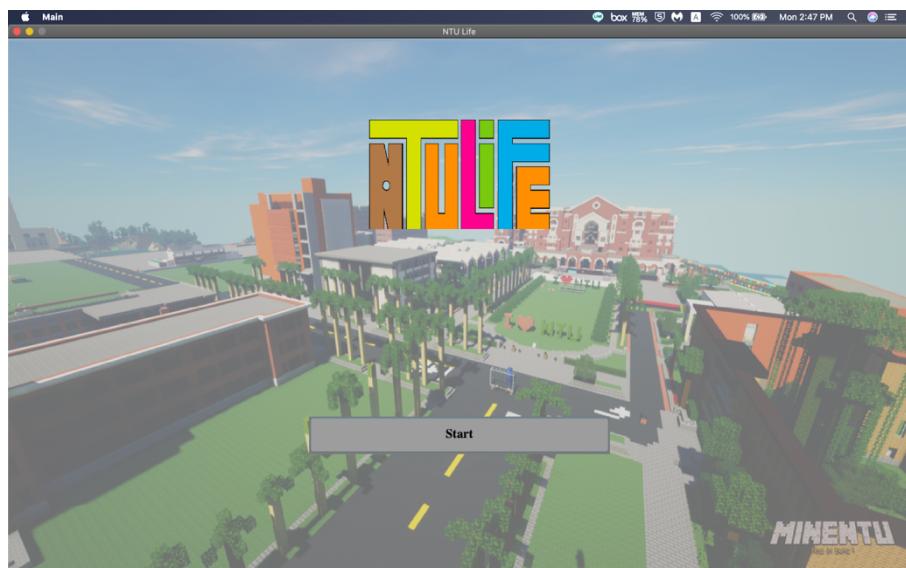
5. Other packages used

- Swing
- AWT

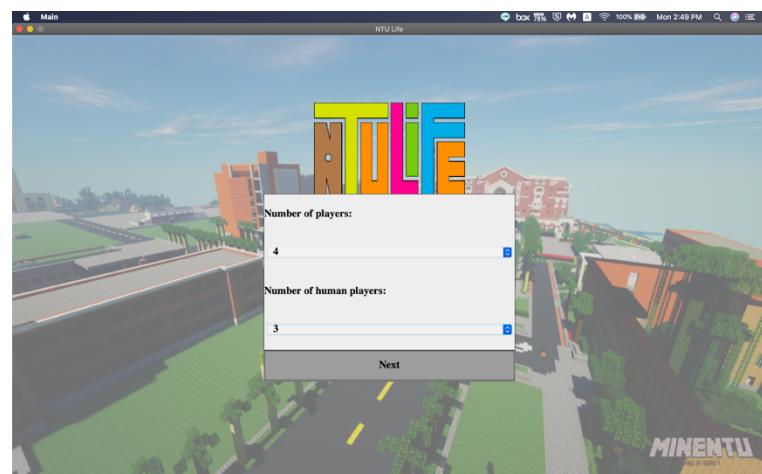
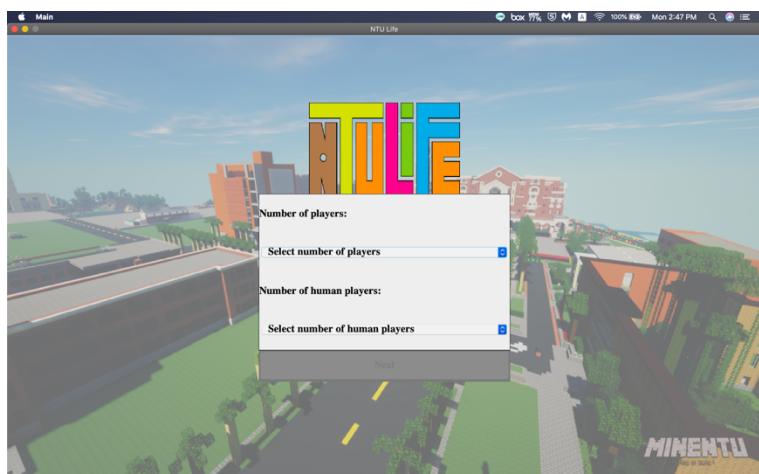
6. How to play the game

a) 遊戲畫面及規則解說

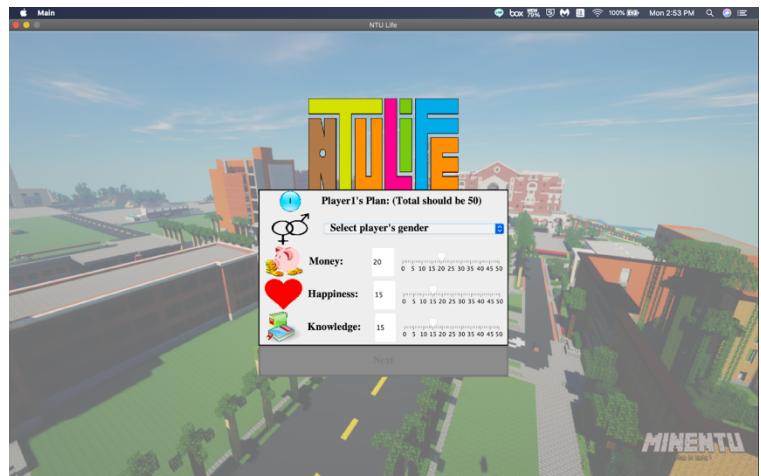
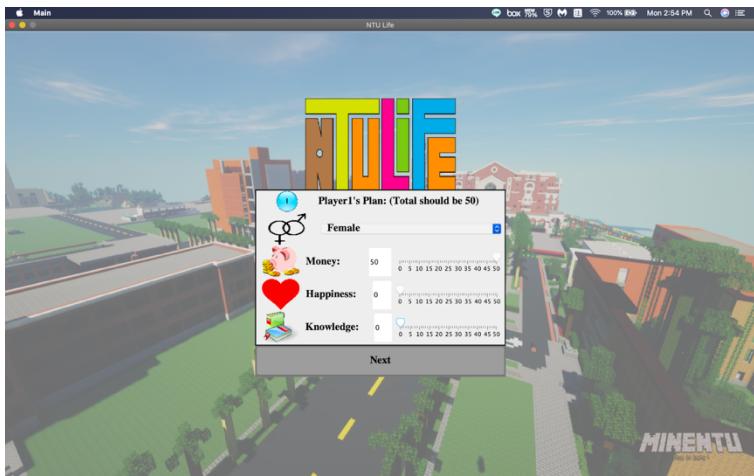
i) 啟動遊戲後進入主頁面，按下 Start



ii) 選擇總玩家人數及人類玩家數，按下 Next



iii) 所有人類玩家選擇遊戲中的性別及訂定自己的勝利條件，資產、快樂、知識的總和需等於 50。



iv) 完成後即可開始遊戲，玩家請在輪到自己時擲骰子決定前進步數並按 OK 確認。遊戲勝利者為最快達成自己所訂定的勝利目標的玩家。



v) 若任一玩家達成自己的勝利條件，則遊戲結束。按下 Close 關閉遊戲畫面。

