

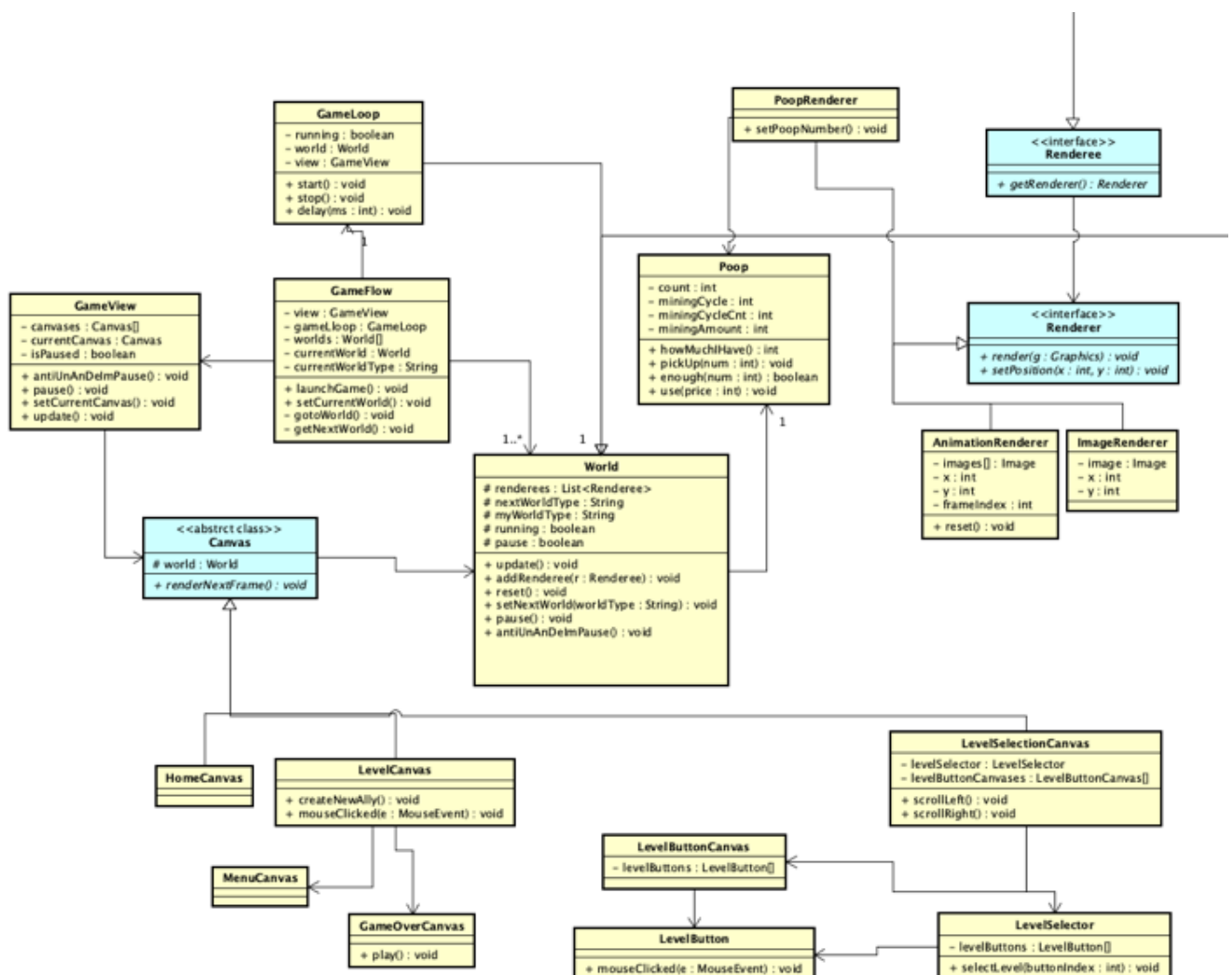
FOOP Final Report

組員與分工

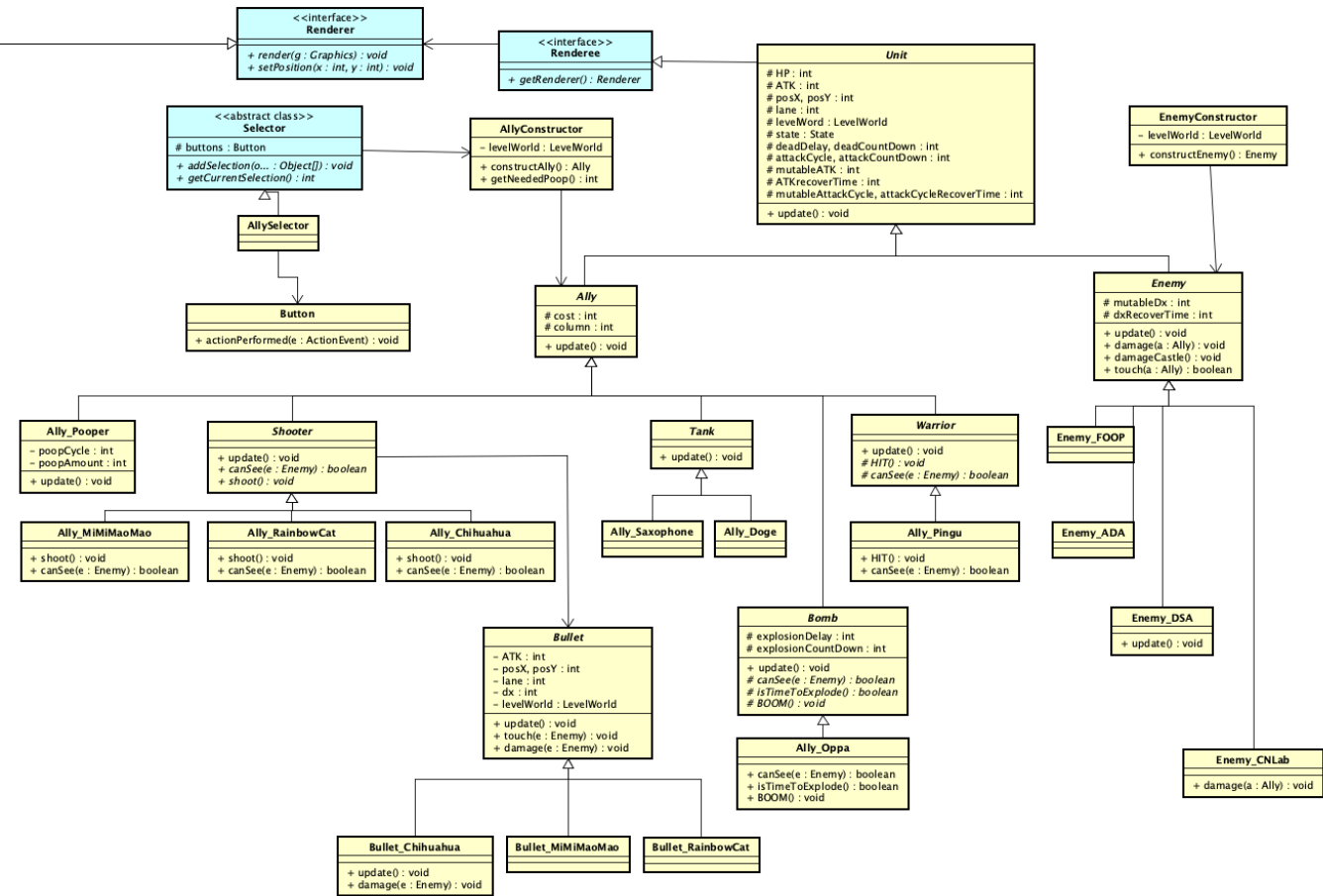
學號	姓名	工作
B07902004	陳品臻	遊戲圖像圖像設計、遊戲平衡、角色設計
B07902010	陳柏鴻	GUI、動畫、遊戲迴圈與流程、merge 和 debug
B07902110	張漢芝	遊戲紀錄讀寫檔、關卡設計、遊戲迴圈與流程、角色設計

Design Pattern

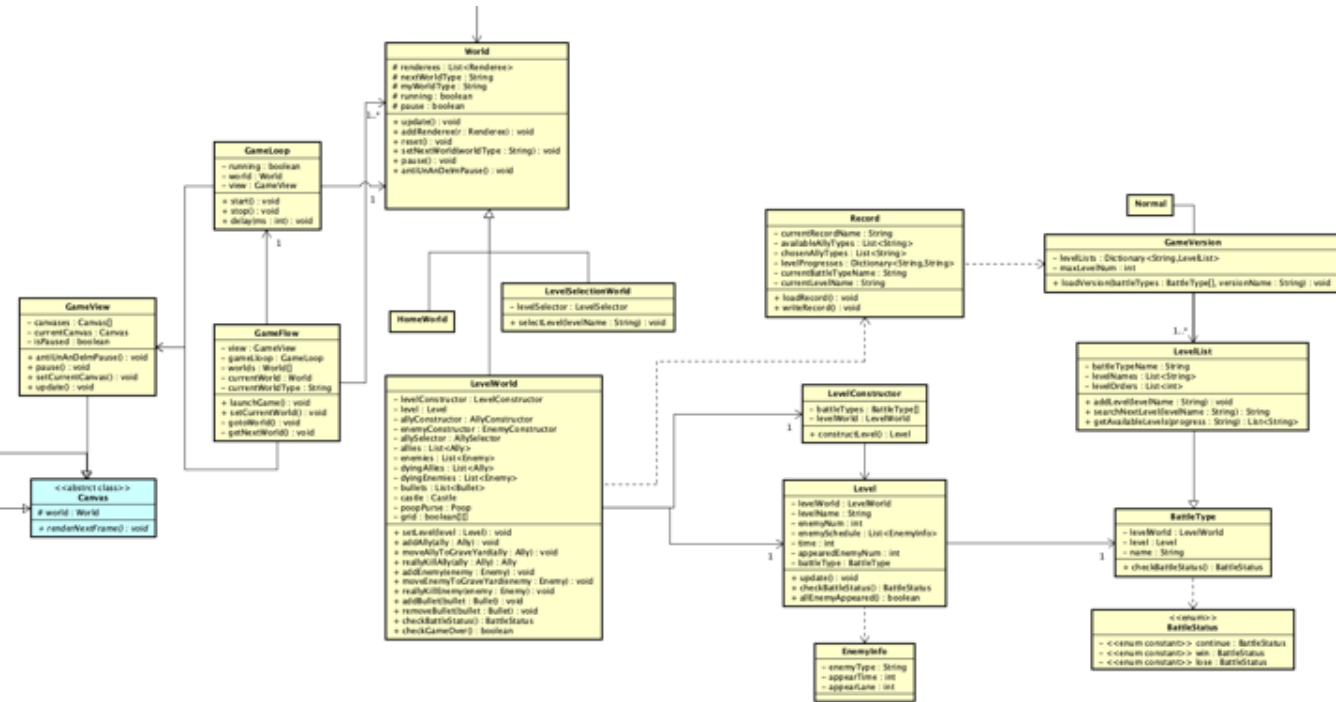
GUI



Game Unit



Game Controller



- **GameView**
 - 讓 `GameFlow` 可以根據 `world` 來切換當前畫布 (`Canvas`)
 - 讓 `GameLoop` 可以更新當前畫面 (`view.update()`)
- **Canvas**
 - 綁定一個 `world`
 - 依據畫布的狀況 (e.g. 滑鼠點擊) 來切換下一個 `world` (`world.setNextWorld()`)
 - 依據 `world` 狀況更新畫布
 - 綁定一些 `Button` 或 `implement MouseEventListener` 來接收玩家資訊
 - 從 `AllyConstructor` 來新增或得知角色資訊
- **Renderree**
 - 綁定一個 `Renderer` 使其可以被畫在 `Canvas` 上
- **Renderer**
 - 將 `Renderree` 的資訊 (圖片、位置等) 畫在 `Canvas` 上
- **World**
 - 持有所有在螢幕上呈現的物件並 `update` 他們，每一種 `World` 都有自己判斷停止時機的方法
- **GameLoop**
 - 執行一個 `World` 直到其停止
- **GameFlow**
 - 選擇現在要執行的 `World` 放入 `GameLoop`
- **Level**
 - 儲存一個關卡的資訊，包括該關卡的種類、每個敵人的類型、出現時機、出現位置 (第幾個 `lane`)
- **LevelConstructor**
 - 讀入檔案製作成一個 `Level` 物件，並傳入 `LevelWorld`
- **BattleType**
 - 定義一個關卡玩法，能夠判斷目前戰場狀態
- **BattleStatus**
 - 戰場狀態的 `enum`，包含 `win`, `lose`, `battleContinue`
- **Castle**
 - 我方堡壘，血量歸零代表我方輸
- **Poop**
 - 戰鬥中的資源，用來買 `Ally`
- **Unit**
 - 定義遊戲中的角色的基本性質，如血量、攻擊力等
 - `subclass` 必須實作 `update()`

- Ally
 - extend Unit ，有 Pooper ， Shooter ， Tank ， Bomb ， Warrior 五種不同的類型
- Pooper
 - extend Ally ，會生產 Poop
- Shooter
 - extend Ally ，最基本的攻擊單位，會發射 Bullet 攻擊 Enemy
- Tank
 - extend Ally ，血量厚但攻擊力低的角色，可以拖延 Enemy 的進攻
- Bomb
 - extend Ally ，能夠炸死周圍的 Enemy ，爆炸後立即死亡
- Warrior
 - extend Ally ，近距離作戰的角色
- Bullet
 - Shooter 由 Shooter 射出攻擊 Enemy ，攻擊完立即消失
- Enemy
 - extend Unit ，會對 Ally 進行攻擊
- Record
 - 遊戲紀錄，包含關卡進度、可用角色、選擇角色
- GameVersion
 - 遊戲版本，包含所有類型的所有關卡
- LevelList
 - 紀錄某種 BattleType 所有關卡名稱和順序的資料結構

Advantages of our design

- 角色和背景皆為親手設計，跟資工系有所關聯、很多彩蛋
- 動畫精緻，每個 Unit 在走路、攻擊和死亡時都有不同的動畫
- 有儘量做到 OCP，在新增功能或是 debug 時都很方便，未來能夠繼續開發
- extend LevelWorld 和 BattleType 可以新增遊玩模式
- 容易新增關卡
- 容易新增角色
- 刺激好玩
- 只需要用滑鼠，鍵盤壞掉也可以玩
- 遊戲參數更新與畫面更新分工明確

Disadvantages of our design

- 所有的視窗無法自由縮放，遊戲顯示的大小必須為全螢幕（1440 * 900）
- 相同、類似功能的按鈕或 Canvas，必須重複設計一個 Class
 - 由於遊戲的按鈕和彈出視窗不多，因此並沒有很完整的模組化
- 未有足夠大量的測試樣本，因此不確定遊戲平衡是否完善
- 加入其他系統（例如：道具）可能會更改到其他 class

Additional Packages

GUI

- java.swing
- java.awt
 - 主要用於接收滑鼠點擊、按鈕點擊事件，以及 GUI 的版面呈現

How to play

遊戲介面



Glossery


角色	參數	功能
 電腦	HP	玩家主要的堡壘，如果城堡 HP 小於 0 則為玩家輸
 鍵盤	HP	放置朋友的位置
 Selector	-	玩家可以選擇多個朋友放在鍵盤的一個鍵帽上
 你的朋友	HP, Cost, 傷害力, 攻擊速度, ... 等	每種朋友有不同的功能
 作業	HP, 傷害力, 移動速度... 等	每個關卡會有不同組合的作業，會攻擊你的朋友或電腦，若將關卡中所有作業解決則玩家獲勝
 PoopPurse	Poop 的數量	可以購買 Selector 中的角色

遊戲規則

參考塔防遊戲，目標是保護城堡不要死掉。

1. 當朋友的 cost 大於 poop 數量時，不可以選擇該朋友
2. 當已經有朋友放在鍵盤的某一個鍵帽上時，不可以在該鍵帽上放其他朋友，直到其死亡
3. 擊敗場上所有作業即勝利！
4. 反之，若電腦被作業攻佔（HP \leq 0），即輸了這場遊戲

遊戲流程

1. 進入主畫面，可以選擇從當前紀錄直接開始 ，或是從 Level 中選擇一個關卡進行



- 必須破前面的關卡，可以解鎖後面的關卡！
- 也可以退出遊戲 



2. 開始後，依照 遊戲規則 來進行遊戲



3. 若想要重新開始遊戲，可以點選 ，來回到主畫面或重新開始此關卡

