

# FOOP Final Project Report

---

## Team Members and Responsibilities

- B07902003 資工三 吳雲行
  - 設計病毒的隨機生成以及擴散的規則
  - 設計病毒數量、染疫人口及死亡人口的關係
  - 設計遊戲的開始、結束畫面
  - 設計 info panel 的內部排版
  - 研究如何對JSplitpane、JLayeredpane進行更好的排版
  - 繪製遊戲結束圖
- B07902013 資工三 陳建豪
  - 設計遊戲各個 panel 間的排版
  - 設計暫停遊戲、繼續遊戲的功能
  - 設計 message panel 的跑馬燈以及遊戲中各時機點顯示的訊息
  - 處理各元件因應不同遊戲狀態的不同行為
  - 設計遊戲運行的邏輯
  - 設計各項道具與感染人數的實際連結
- B07902048 資工三 李宥霆
  - 發想 project 的整體架構
  - 改善整體程式碼的架構
  - 設計 map panel 以及地圖上的分區
  - 設計 toolbar panel、detail panel
  - 設計各項道具的功能以及各項道具的詳細資訊
  - 蒐集道具圖、繪製遊戲開始圖

## Relationship between Classes

- package components
  - Area
    - 紀錄地圖上各分區包含的 grid 的編號
    - 提供地圖上各像素對應到的分區編號
    - 提供把某個分區內的所有 grid 塗上顏色的函式
  - Virus

- 記錄病毒的位置以及該位置所屬分區的編號的 JLabel
- package frame
  - JFrame
    - 遊戲畫面的基底 JFrame，遊戲中的其他 panel 都會放在 JFrame 上
- package game
  - Gameflow
    - 負責初始化版面上的各個 JPanel，包含排版等
    - 維護數個 Timer，定時讓 StateManager 更新遊戲內各個狀態，如金錢、時間、病毒生成、感染、康復、死亡等
    - Implement GameStateListener，會隨著 GameState 的變化做出不同的流程
  - GameState
    - **遊戲本身的狀態**的 enum，包含 init、pause、in game、win 和 lose
  - GameStateListener, MapStateListener, ItemStateListener, GoldListener, VirusListener, DateListener, ItemNumListener, PeopleNumListener
    - Interface，負責在遊戲狀態發生變化時做出改變
    - 如果一個元件會根據某個遊戲狀態不同而做出不同的行為，那麼該元件就需要 implement 相對應的 interface，並且把自己加入 StateManager 中相對應的 Listener List，並且 override interface 裡的函式來做出自己的行為
  - StateManager
    - 負責管理遊戲的所有狀態，包含 GameState、金錢、病毒數、道具、各區人口等
    - 負責遊戲邏輯的運作，如病毒生成的量和地點等
    - 維護不同的 Listener 的 List，當某個 state 發生變化時就會呼叫 List 內的所有元件相對應的 interface method
- package panels
  - DetailPanel
    - 畫面右下角的 JPanel，其上又有三個不同的 Panel，包含各區域資訊、各道具資訊、各區域使用道具情況，根據滑鼠的行為會顯示不同的 Panel，藉此來顯示不同的資訊
    - Implement ItemStateListener，如果滑鼠停留在物品的圖片上，就會顯示該道具的功能
    - Implement MapStateListener，如果滑鼠停留在地圖上的某一個位置，就會顯示該位置所屬分區的詳細資訊
  - InfoPanel
    - 畫面右上角的 JPanel，顯示遊戲當下的時間、金錢以及確診人數
    - Implement GoldListener，只要金錢數量有變化就會顯示新的金錢數量
    - Implement DateListener，只要遊戲時間有變化就會顯示新的遊戲時間

- 備有暫停鈕，在使用者暫停遊戲時會改動game state，進而讓gameflow的timer停下
- MapPanel
  - 畫面左下的 JPanel，顯示遊戲的地圖以及地圖上的病毒
  - Layout manager 使用 GridLayout，會偵測滑鼠停留在哪一個 grid 上，並且利用 Area 中的函式把該 grid 所屬的分區塗上顏色
  - 如果滑鼠從 a 分區的 grid 移動到 b 分區的 grid，就會讓 StateManager 去通知所有的 MapStateListener 做出改變
  - Implement VirusListener，遊戲中的病毒增加或減少都會即時在 MapPanel 上顯示出來
- MenuPanel
  - 遊戲剛開始時顯示的 JPanel，按下 Start game 的 button 後就會被隱藏起來
- MessagePanel
  - 畫面左上角的 JPanel，會顯示當前遊戲的訊息（如某區疫情嚴重）
  - GameFlow 中會定時檢查特定的數值（如某分區的病毒數量、死亡人數等），並且讓 MessagePanel 顯示相對應的訊息來提醒玩家
  - Implement GameStateListener，當 GameState 變成 pause 時跑馬燈顯示的文字會變成“遊戲已暫停”，當 GameState 變成 in game 時跑馬燈會繼續顯示遊戲的訊息
- ToolbarPanel
  - 畫面右邊中間的 JPanel，顯示遊戲內的道具的圖片
  - 會偵測滑鼠停留在哪個道具的圖片上，如果滑鼠從 a 道具的圖片移動後停留在 b 道具的圖片上，就會讓 StateManager 去通知所有的 ItemStateListener 做出改變
- package utils
  - Date
    - 用來讓 InfoPanel 方便顯示日期的 class
  - Utils
    - 包含一些在開發過程中常用到的函式，例如建構一個新的按鈕或 JLabel 等

## Advantages of Our Design

- 遊戲的狀態統一由 StateManager 來管理，各元件只需要告訴 StateManager 希望如何改變遊戲狀態，不需要自己改變遊戲狀態，且由於遊戲狀態改變的邏輯判斷都寫在 StateManager 中，較不容易發生不同元件間修改遊戲狀態導致衝突的情況
- StateManager 的所有變數和 method 都是 static，在各個 class 中都可以方便的使用 StateManager 中的 public method
- 在狀態的種類不變的情況下，新增元件只需要 implement 相對應的 Listener interface、把自己加入到 StateManager 中相對應的 List 中並且定義好自己的行為就可以正確地運作，不需要改動 StateManager 或是其他原本的程式碼

- 我們的各項道具是寫死在 StateManager 裡面的 array，這樣在操作各項道具時邏輯會比較簡單，程式也會跑比較快
- 各個panel各自有class來實際設計版面，而沒有直接將整個遊戲版面統一設計，可以更方便做各自細節的微調
- 病毒擴散、人口感染、死亡等等，都是套用現實社會中可能出現的規則，例如 $R_0$ 值、感染致死率等等，更有現實感
- 遊戲理念上貼近時事，且選用實際地區與實際背景，能夠使玩家感受到身歷其境

## Disadvantages of Our Design

- 遊戲的狀態統一由 StateManager 來管理，StateManager 的負荷會比較大且日後延伸較為困難
- StateManager 的所有變數和 method 都是 static，這代表這個遊戲只能是一個單機遊戲
- 雖然新增元件能夠不改動到過去的程式碼，但如果要新增一些遊戲狀態就必須改動 StateManager，讓 StateManager 去維護新的 Listener List
- 我們的各項道具是寫死在 StateManager 裡面的，因此在道具方面做擴充會比較麻煩
- 在病毒render和滑鼠掃過各區所造成的render時，病毒數量若多一點時有可能有lag現象，為計算量和render量較大導致
- 在病毒生成與傳播規則上因為參考現實世界 $R_0$ 值的規則，有較多參數與複雜的計算，若日後視遊戲狀況要調整參數會需要重新計算傳染人口比例是否合理
- 各個元件都是使用swing基本元件，如button等，沒有做太多style上的處理，整體看起來可能稍微粗糙而不吸引人

## Other Packages We Used

在這次 project 中我們只有使用到 java 的 swing 和 awt，沒有使用到其他的 package。

## How to Play Our 2D Game

*“當病毒大軍來襲，身為指揮官，你是否能做出對的決策...”*

這個遊戲叫做疫情模擬器，可怕的病毒會不定時隨機出現，如果不適時作出防疫措施的話病毒會以可怕的速度蔓延。妥善運用手上的金錢，針對不同區域的情況使用不同的道具以求減緩疫情的發展速度。

進入遊戲後，右上角會有模擬的時鐘、擁有的金錢以及目前整張地圖上的感染人數及比例，如果把滑鼠移到地圖上的某個位置，畫面右下角就會顯示該區的總人口數、感染人數、死亡人數以及已部署的各項防疫用品，如果把滑鼠移到畫面右邊的道具圖片上，就會顯示部署該道具所需要的金額以及的該道具帶來的效果。如果要在某區部署防疫用品，就以滑鼠左鍵點擊該分區後在畫面右下角選擇要部署的道具及數量後按下 confirm 即可。

注意！道具有一定數量的限制，不是有錢就可以瘋狂購買。必須審慎考慮哪一區疫情比較嚴重，適當進行分配，才能夠正確減緩疫情。

只要能夠連續控制住疫情 14 個週期，就能獲得遊戲的勝利。如果死亡人口數大於總人口數的 20% 或是死亡人口加上染疫人口大於總人口數的三分之一，遊戲就會宣告失敗。

“沒有可以馬上完全消滅的疫情，只求控制住它...”