

專案題目：資工殺

小組名稱：浩全大帥哥別插/*入註解*/

小組成員：103062116 楊澤仁 103062122 鄭鈺敏

103062224 劉哲宇 103062240 蔡宗宇

背景與目的

我們經常會遇到一些很難抉擇的問題，像是一群人要出去吃飯的時候，常常會聽到的一句話就是：「我都可以。」於是一群人像無頭蒼蠅一般地往外走而不知道要吃什麼。雖然這裡是號稱美食沙漠的新竹，但以學餐的選擇還是有蠻多的，甚至到了學校外面也是有能填飽肚子的店。但是，就是因為選擇如此繁多，一個人吃午餐還好，但是如果有四、五人要一起吃飯呢？要吃小吃部、水木、風雲？甚至是校外？校外的選擇也分有東門、西門呢！

就是因為上述原因，所以我們想到讓「因為人太多而難以抉擇的問題」能獲得簡單且快速的解決方式：每個人都在電腦輸入自己的名字跟想做的事情，在一個名為「資工殺的」遊戲上作生死搏鬥，讓勝出的那一方即為團體的最終決定，讓大家不再困擾類似的問題。

設計概念

遊戲化設計：此專案是將眾人難以抉擇的問題，化身成一個卡牌遊戲，讓使用者透過遊戲中的廝殺來決定一個團體最後的抉擇。遊戲流程設計為：讓玩家先輸入自己的暱稱以及想要做的事情（像是想要吃哪家店，如果有人填隨便就能優先送他上路），之後登入伺服器等待其他玩家連線，等到大家都加入了之後，即可開始遊戲。遊戲開始玩家可立刻獲得三張牌，每次輪到自己回合時可抽一張牌，而場上只有被輪到的玩家可進行出牌。若出到沒牌、或不想繼續出牌時，可以結束你的回合（牌的各種效果附在卡牌實作的說明底下）。遊戲過程中，亦可使用聊天室聊天，你可以透過聊天室遊說大家，圍剿那些提出很爛的決策的人，或是說服意見相近的人先跟你聯手，對付那些你們都反對的人。在經過廝殺後的最後倖存者，就會是團體的最後決策人！

技術細節

一、Server and Client 之間的溝通內容

Server 會 maintain 三個 vector，一個專門用來記錄遊戲，一個是用來記錄單一場遊戲的連線 threads，一個是用來記錄單一場遊戲的聊天 threads。當 Server 按下 Start 按鈕，就會把當前所有在等待遊戲開始的 clients 丟到一個 Game 裡面並由紀錄 Game 的 vector 來

繼續作處理，以達到同樣一個 Server 可以開啟多場遊戲的架構。在遊戲中 Server 主要負責牌堆的維護，像是洗牌、發牌等，當玩家出牌後會將他所出的牌的資訊送給 server，由 server 判斷這張牌所造成的效果並以類似指令的方式通知 clients 更新顯示資訊，例如某位玩家少了 X 滴血、被丟了什麼牌，當遊戲結束時（定義為只剩一名玩家存活），通知所有 clients 並顯示贏家及他所做的決定。

關於遊戲訊息的傳送，由於我們都是用字串的形式來傳遞遊戲資訊，為了避免字串太多太混亂易造成寫程式的困難，我們把那些會用來傳送遊戲訊息的特殊字串，彙整到同一個檔並定義成 static final String，並且為每一個字串寫了詳盡的 document，每次使用時都可以作為參考，出錯的機率就小了很多（可參考 GameMessage.java）

二、 遊戲畫面

本專案的畫面呈現都是利用 JFrame 包住 Processing 這個函式庫所提供的 PApplet 來作處理。主要分成以下幾種技術：

1. 選擇牌：在滑鼠移到牌的上方，牌就會凸出來，利用牌的四個點座標，再加上滑鼠的座標，以及 Ani 所提供的 to 函式，就可以達到這個效果。如果滑鼠進一步點下，牌也會跟著滑鼠一起移動。
2. 顯示卡牌以及角色資訊：承上，在選擇牌的過程中，可以利用內建的函式來顯示卡牌的資訊，例如用 text() 以及 rect() 來顯示姓名、意圖、剩餘血量以及手牌數。此外，利用圓心以及滑鼠的座標算距離，如果小於角色圓圈的半徑，就代表滑鼠指向它，就可以顯示出角色資訊。
3. 牌的移動動畫：其他閒置中的角色可以藉由 Server 知道現在正在出牌的角色出了什麼牌，並且利用 Ani 的 to 函式把該張牌移動到畫面右上方，且如果卡牌效果會讓某玩家失去牌，也會利用同樣的方式顯示失去的牌。另外，在自己的回合開始時，也會由主畫面中間偏左方出現新的卡牌，並且移動到手牌區。
4. 指定型卡牌：在選擇卡牌以後，在滑鼠所指的地方，會出現一個準心，可以選擇讓此張卡牌的效果發動至任何一名玩家，也是利用內建的 ellipse 以及 line 函式搭配 mouseX，mouseY 來作繪畫顯示。
5. Minim：音樂的撥放等，都是利用 Minim 這個函式庫來處理。包含 mp3 的讀取、撥放、回至音訊檔原點等。
6. ControlP5：這個函式庫主要是用來顯示所有按鈕、聊天室的聊天訊息以及聊天室的打字區域等。

三、遊戲資訊的存放：

每個 client 端的遊戲，都會存有該場內的所有玩家的血量、卡牌數量，當然還有每名玩家的名子與意見，而對自己則可以知道每張牌分別是什麼牌。而 server 其實除了牌庫（負責洗牌、發牌、回收牌）之外，並不存有場上玩家的資訊，server 只處理發牌、處理玩家的出牌、詢問被指定效果的玩家是否要出某張牌來回避、處理遊戲結束，非常節省空間資源。

四、牌庫

server 針對每一場遊戲都會 maintain 一個獨立的牌庫（底下有牌庫的初始牌量），牌庫本身有 drawArea, discardArea 兩區，每次進入遊戲時將所有牌放入 drawArea 中，會先使用 `Collection.shuffle(drawArea)` 來洗牌。牌庫本身提供 `drawTop()` 來抽出 drawArea 中最上面那張牌，可以給 server 發牌用。

每當 player 把牌用掉，或是因為別人的效果而損失手牌時，server 會知道它丟了什麼牌，然後用牌庫的 `discardCard()` 把被丟棄的牌放到 discardArea 中。

如果在使用 `drawTop()` 時發現 drawArea 是空的，就會把 discardArea 的所有牌移到 drawArea 中並且洗亂，因此在正常的遊戲底下不會有牌庫被抽光的問題，且場上的牌加牌庫的牌，各種牌的數量與組合永遠會一樣。

五、卡牌實作

本遊戲的牌可分成兩種：基本牌、以及錦囊牌。首先先創立一個最上層的類別：Card。Card 會囊括所有種類的牌都有的性質，例如卡片分類、卡名、描述、圖片路徑、卡片 ID 等。也因為在我們的架構下，server 與 client 是用字串進行溝通，所以有個 `effectString()` 函式，透過填入該卡的目標後，此函式會產生發生的效果，server 只需要把該字串廣播給所有 client，即可在 client 端透過該字串更新場上玩家的資訊（如卡牌量、血量等）。

基本牌如殺、閃、金蘋果，直接繼承 Card 來實作。

而由於本遊戲每張錦囊牌的效果都十分獨立，因此對於錦囊牌的使用上有不小的困難。所以我們又創了一個類別：JinCard。Jincard 會繼承 Card，並且多了以下的屬性（皆為 boolean），這使得出牌時的判定能簡單許多：

1. AreaEffective:

會影響到全體（輪流被影響）。反之，則效果只會生效於一人。

2. SelfExclusive:

如為 !AreaEffective 牌種，該張牌不能用在自己身上。

如為 AreaEffective 牌種，該張牌效果會跳過出牌者本身。

反之，則可以對自己使用 / 效果會發生在自己身上。

3. EffectiveNow:

在這個回合就會產生效果。反之，則該牌會輪到被使用的角色該回合時才生效。

4. Conditional:

只在某些條件成立下生效。反之，則無論無何都會生效。

每張錦囊牌類別都會繼承 JinCard，並且獲得不同的屬性。透過這四種屬性，可進一步知道 player 出某張錦囊牌時，是否必需要求 player 指定某個玩家，且是否可以指定自己（若不行就要跳出通知提醒 player）。且 server 知道某個 player 出了哪張牌時，也能透過牌的屬性，知道該牌是否需要進行條件判定，以及會對誰生效等等資訊。

以下為牌的種類與資訊（×N 代表該排在牌庫中有幾張）

基本牌：

1. 金蘋果 × 15：在自己的回合內為自己回復一點體力值。

2. 閃 × 15：當受到【殺】的攻擊時，可以使用一張【閃】來抵消【殺】的效果。

3. 殺 × 30：出牌階段，對除自己以外的一名角色使用，效果是對該角色造成 1 點傷害。

錦囊牌：

1. 狂歡宴會 × 8：出牌階段，對所有角色使用。按照行動順序結算，所有角色各回復 1 點體力。

2. 點石成金 × 8：出牌階段，對自己使用，摸兩張牌。

3. 盜取 × 8：出牌階段，對除自己以外，選擇並獲得該角色的隨機

一張牌。

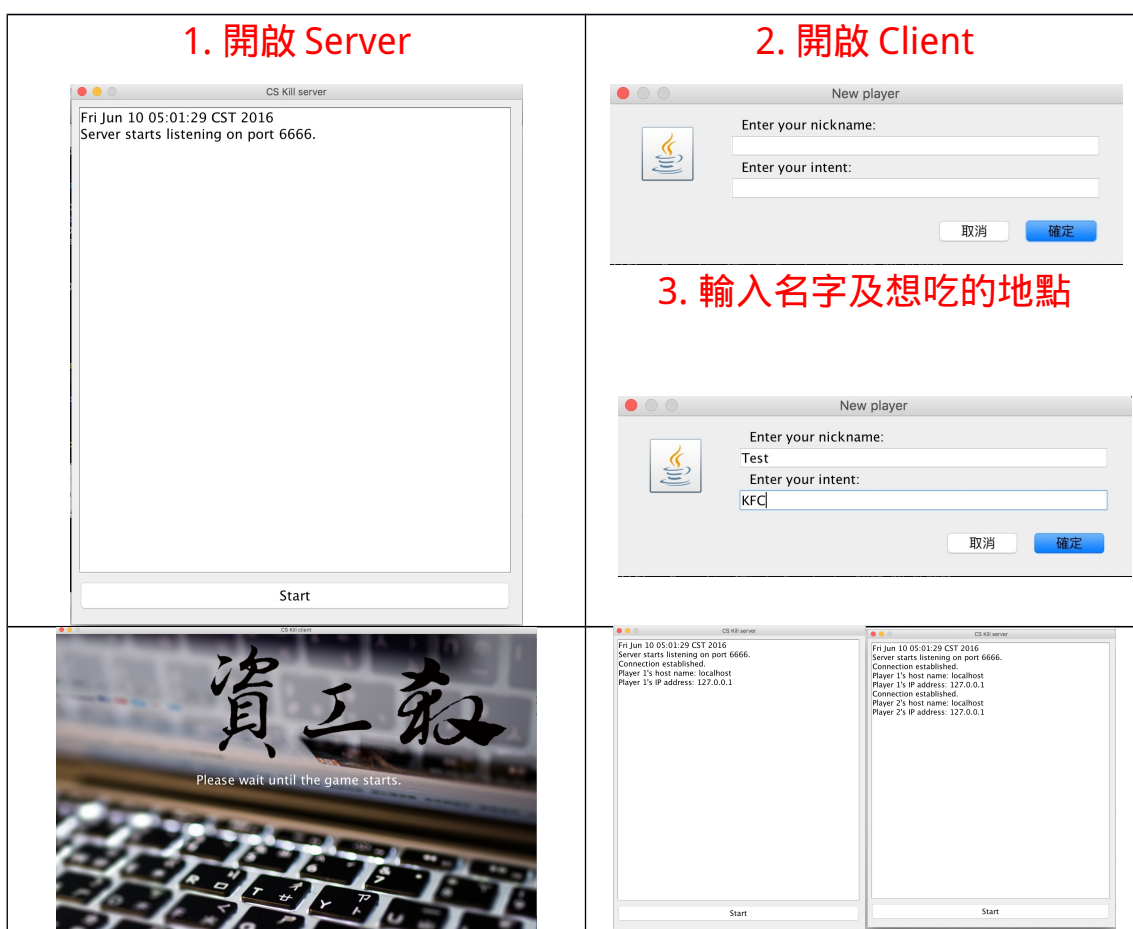
4. 繳械 × 8：出牌階段，對除自己以外任意一名角色使用。棄掉該角色的隨機兩張牌。

5. 萬箭齊發 × 12：出牌階段，對自自己以外的所有角色使用，按行動順序結算，除非目標角色打出一張【閃】，否則該角色受到【萬劍齊發】對其造成的 1 點傷害。

6. 鐵騎突襲 × 12：出牌階段，對除自己以外的所有角色使用，按行動順序結算，除非目標角色打出一張【殺】，否則該角色受到【鐵騎突襲】對其造成的 1 點傷害。

7. 五穀豐登 × 8：出牌階段，對所有角色使用。自己獲得三張牌，其他人獲得一張牌。

實作成果



4. 遊戲開始畫面	5. 此時 Server 會顯示連線人數
	
6. 遊戲進行畫面	
	
7. 遊戲結束畫面	

遇到的困難

1. 遊戲的選擇: 其實我們一直都不太確定到底要做出什麼遊戲，因為在與助教討論時有提到一個重點是遊戲時間不能太長，否則會導致為了做決定玩這個遊戲後反而花更多時間的問題，最後就半推半就地選擇模仿一些三國殺的規則來寫一個卡牌遊戲，以後應該要更有創意會比較好些。

2. ControlP5 的 bug: 在實作聊天室的時候，發現 Client 在結束回合後，會無法繼續聊天，原本以為是自身程式碼所導致的 block 的問題，但是後來發現是 ControlP5 這個函式庫在 lock 物件的時候，會把其他物件也都 lock 起來，幸好原本的程式碼就可預防「結束回合鈕」被重複多按的情況，所以只要不使用 lock 即可。

3. Minim 的路徑問題: 我們沒有想到程式 export 成 jar 檔之後，

Minim 會疑似因為路徑問題而找不到檔案，導致我們在 demo 的時候，因為讀不到音訊檔而造成了錯誤，使得 Server 在 Client 當掉之後整個 crash 掉。後來我們根據錯誤訊息把音訊檔放至適當位置才讓 export 出來的 jar 檔運作成功。

工作分配

楊澤仁(103062116)：Proposol、卡片建立、美術

鄭鈺敏(103062122)：程式基本架構、聊天室、研究少見函式庫的使用方法、上台報告、Final Report。

劉哲宇(103062224)：錦囊卡屬性分類、卡片效果實作、上台報告 Partner、Final Report

蔡宗宇(103062240)：程式基本架構、卡牌資訊顯示，遊戲間卡牌特效顯示

檔案連結

<https://github.com/taffy128s/SoftwareStudioFinalProject>

Server: java -jar server.jar <Port>

Client: java -jar client.jar <Server IP> <Server Port>

參考資料

<https://docs.oracle.com/javase/8/docs/api/>

<https://processing.org/reference/>

<http://benedikt-gross.de/libraries/Ani/>

<http://www.sojamo.de/libraries/controlP5/>

<http://code.compartmental.net/tools/minim/>