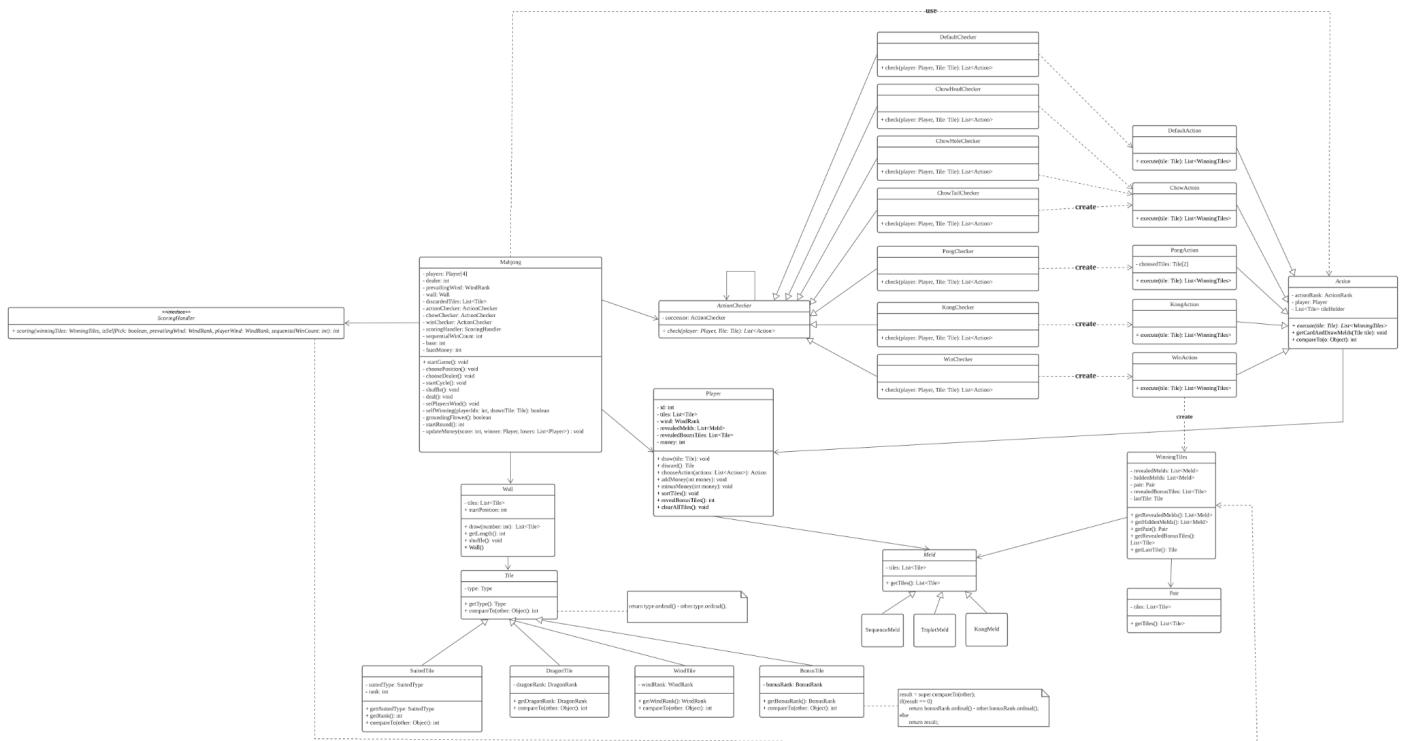


1. the team members' names and school IDs
 - 林詩敏 b06902075
 - 吳柏成 b06902037
 - 鄭人愷 b06902127
 2. how you divide the responsibilities of the team members
 - a. 林詩敏
 - i. 台數計算
 - b. 吳柏成
 - i. Initial Design
 - ii. Game Flow
 - c. 鄭人愷
 - i. 遊戲介面
 - ii. Action
 3. the relations between the classes that you design



a. Mahjong

- 負責建立Wall, Player, ActionChecker等Class
 - 負責遊戲流程
 - 需要使用者執行操作時調用player.chooseAction()或player.discard()
 - 呼叫ActionChecker以確認玩家可以執行的Action, 並從所有玩家執行的Action中選擇順位最高的執行(ex. user1選擇吃, user2選擇碰, 則最後會執行碰的Action)
 - 呼叫Action.execute已執行特定Action

- 胡牌時使用scoringHandler計算台數
- b. Player
- 負責記錄玩家相關參數(ex.UID, 手上有的牌)
 - Mahjong透過player輸入使用者操作
 - ActionChecker檢視player是否可以執行Action
 - scoringHandler根據玩家狀態計算台數
- c. Wall
- 牌堆
 - 負責發牌給玩家
- d. Tile
- 分為SuitedTile, DragonTile, WindTile, BonusTile四種
 - SuitedTile可以再細分為Dot, Bamboo與Character
 - ActionChecker根據玩家與牌面上的Tile判斷是否可以執行Action
 - scoringHandler根據牌型計算台數
- e. ActionChecker
- Mahjong藉由ActionChecker判斷玩家可以執行的Action
 - 有WinChecker, PongChecker, KongChecker與ChowChecker
 - 判斷可以執行後，生成Action物件回傳Mahjong
- f. Action
- 紀錄並執行特定玩家選定的操作
- g. scoringHandler
- 根據遊戲狀態計算台數
4. the advantages of your design
- Open Close Principle
 - i. Mahjong透過ScoringHandler這個Interface使用算台功能，物件只要implement該Interface就可以讓Mahjong算台。如需加入本遊戲所設計的TaiwanScoringRules以外的算台規則，就只需符合該Interface即可，滿足Open Close Principle。
 - Chain of Responsibility
 - i. ActionChecker的設計使用Chain of responsibility，若確認使用者可以執行某一種Action，ActionChekcer就會將該Action加到回傳的list中，最後回傳的list就包含了所有使用者可以執行的Actions。這樣設計的好處是如果想要取消支援某一種Action，我們可以透過拔除對應的ActionChecker和Action來達成。但Action的部分因為使用到Enum，沒有完美達成open close design principle，在第五點我們會更詳細提到，並提出解決方案。
5. the disadvantages of your design
- 沒有MVC，若使用MVC的話就可以將視圖和業務邏輯分離，降低class之間的耦合性，並使GUI的設計更為容易。

- 使用者的Action有可能彼此衝突，比如說有一位玩家打了一筒，剩下三位玩家分別可以吃、碰、胡，而這些Action有優先順序在，胡的順位大於碰，而碰的順位大於吃。目前我們的處理方式是使用enum讓不同action有不同的順位，這樣並無法滿足open close design principle。
 - 可能的解決方法：同樣使用Chain of Responsibility，每一種Action會對應到特定的ActionExecutor，並將所有的ActionExecutors依照牌型的優先順序串接。之後將所有使用者執行的Action傳入，如果可以胡就執行WinAction並回傳，反之則往傳給successor判斷是否執行。沒有使用此種方法主要是因為太晚想到了，程式碼重構會需要花太多時間。

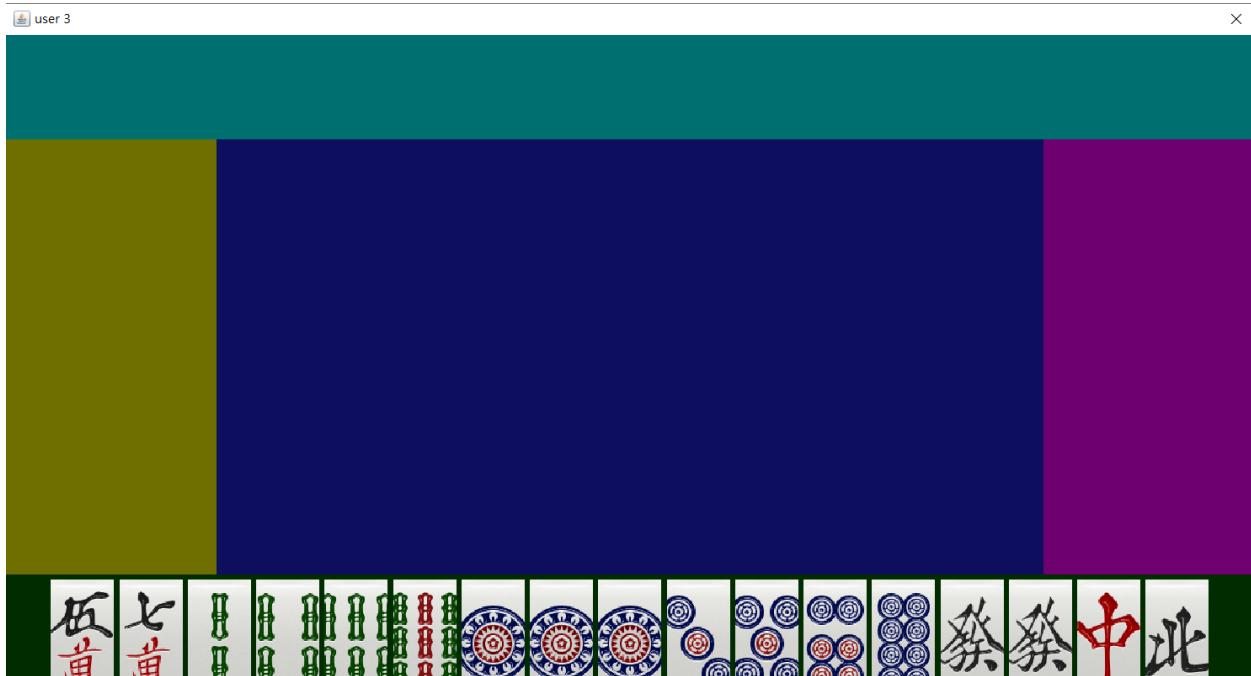
6. other packages that you have used

- Swing for the GUI

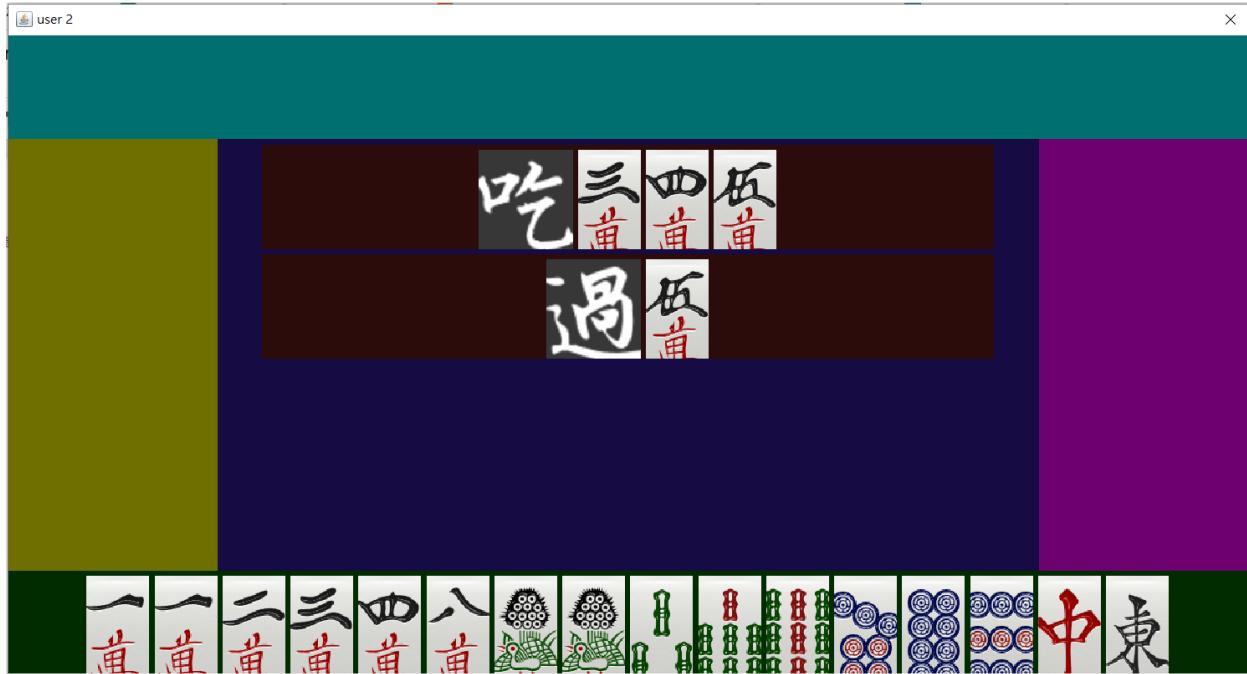
7. how to play your 2D game

- ### ● 遊戲流程

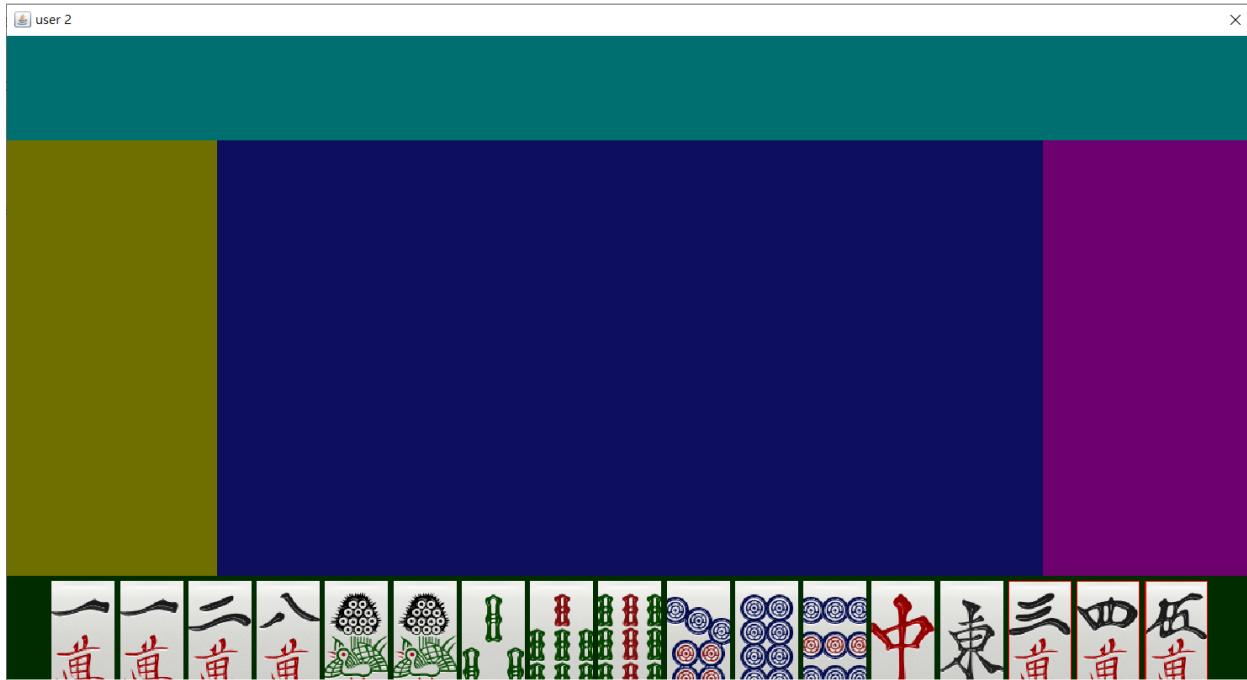
- 出牌階段，玩家點擊麻將以出牌，視窗左上角會顯示目前玩家



- 當可以吃時畫面中間跳出選項，點擊棕色區塊表示選取Action



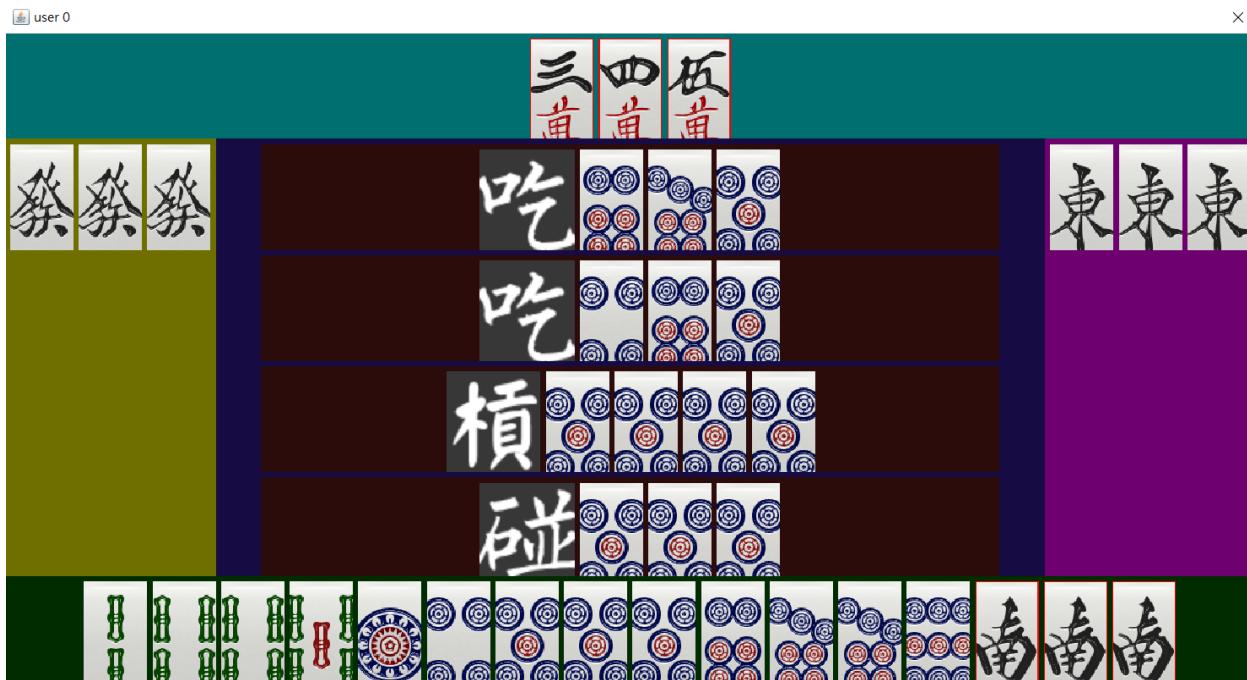
- 在吃牌過後，進牌放置右手邊，並以紅框表示



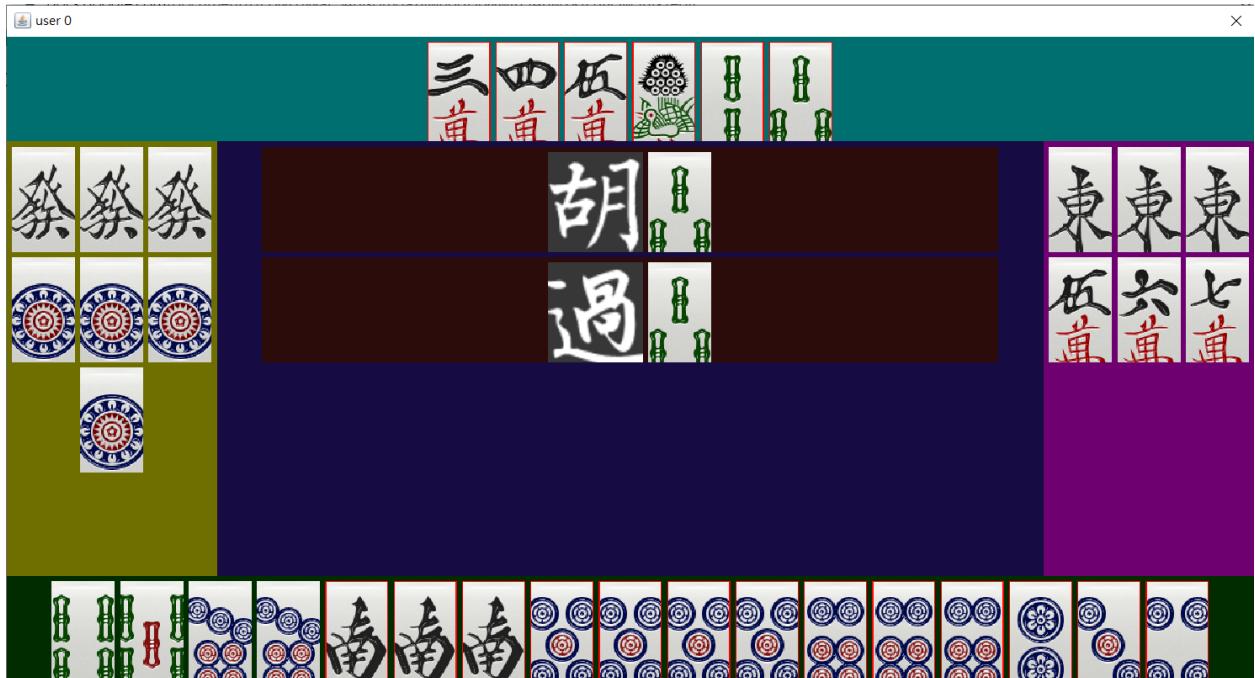
- 玩家可以碰時，畫面中間出現選項



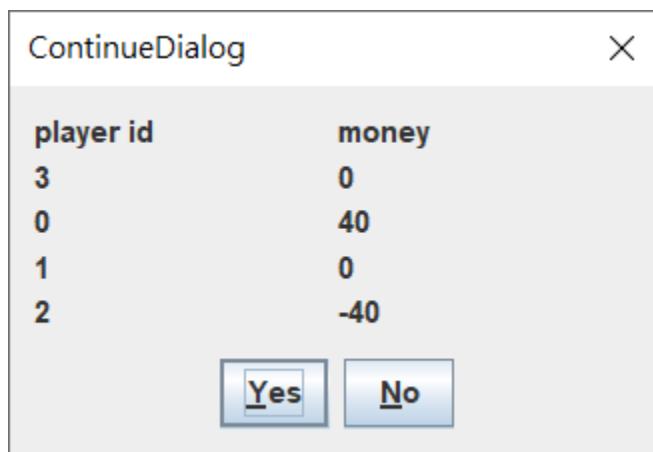
○ 玩家可以槓時，畫面中間出現選項



○ 玩家可以胡牌時，畫面中間出現選項



- 一局結束後會跳出結算畫面，並詢問是否繼續遊戲



- 台數計算

麻將遊戲的台數計算由 ScoringHandler interface 負責。在此我們設計 TaiwanScoringRules implements ScoringHandler, 以稍微簡化過的台灣麻將規則計算台數。TaiwanScoringRules 在 scoring method 中, 檢查是否符合 ScoringPattern 條件, 若條件符合則更新台數並移除相依牌型(如三/四/五暗刻)。以下列出我們採用的 ScoringPattern :

牌型	Class	台數	條件
莊家連 N 拉 N	SequentialWin	2N+1	無。

自摸	SelfPick	1	自己摸到胡牌的牌。
門清	WinFromWall	1	胡牌前沒有吃、碰、明槓。
不求人	AllFromSelf	1	門清自摸。
單邊	OneSide	1	12 聽 3、89 聽 7。
中洞	Middle	1	聽順子的中間序牌。
單吊	OneShot	1	只聽將牌缺少的那張牌。
全求人	AllFromOthers	2	胡牌者只剩1張手牌。 如自摸不計單吊。
正花	Flowers Seasons	1	東家拿到一花(春、梅)，南家拿到二花(夏、蘭)，西家拿到三花(秋、菊)，北家拿到四花(冬、竹)。 實作上將四季及四君子分兩個 class。
花槓	FlowerKong SeasonKong	2	拿齊四季或四君子。 不計正花。 實作上將四季及四君子分兩個 class。
七搶一	SevenBonuses	8	持有 7 張花牌，搶別家 1 張花牌湊滿所有花牌而胡牌。 不計正花、花槓。
八仙過海	EightBonuses	8	取得全部 8 張花牌而胡牌。 不計正花、花槓。
平胡	CommonHand	2	胡牌者無字、無花、無刻、不是偏崁獨、非自摸。
三暗刻	ThreeSelfTriplets	2	胡牌時，手牌有3副刻子(包括暗槓)。
四暗刻	FourSelfTriplets	4	胡牌時，手牌有4副刻子(包括暗槓)。 不計三暗刻。
五暗刻	FiveSelfTriplets	8	胡牌時，手牌有5副刻子(包括暗槓)。 不計三暗刻、四暗刻、五暗刻。
碰碰胡	AllInTriplets	4	胡牌時，除了將牌外全為刻子(或槓)。
混一色	MixedOneSuit	4	胡牌由一種花色的序牌及字牌所組成。

清一色	AllOneSuit	8	胡牌由一種花色的序牌所組成。
字一色	AllHonorTiles	16	胡牌由字牌的刻子(槓)所組成。 不計碰碰胡。
三元刻	Dragons	1	中、發、白任何一組刻子或槓子。
小三元	SmallDragons	4	胡牌時有三元牌的 2 副刻子及將牌。 不計三元刻。
大三元	BigDragons	8	胡牌時有三元牌 3 副刻子。 不計三元刻。
圈風刻	PrevailingWind	1	持有附和圈風的任何一組刻子或槓子。
門風刻	SeatWind	1	持有附和門風的任何一組刻子或槓子。
小四喜	SmallWinds	8	胡牌時有風牌的 3 副刻子及將牌。 不計圈風刻、門風刻。
大四喜	BigWinds	16	胡牌時有風牌的 4 副刻子。 不計圈風刻、門風刻。

下圖為台數計算部分的 class diagram。ScoringHandler interface 及 abstract class ScoringPattern 可以讓之後在不同計分規則(如香港麻將規則)或牌型的選用以及修改上更有彈性。然而，有許多牌型之間即使高度相似，卻受制於 ScoringPattern 的架構分為不同 class，使程式碼有些重複，像是三/四/五暗刻、大/小三元等。

