

# 大老二遊戲說明

## 一、 遊戲初始介面、設定

遊戲的初始界面有三個按鈕：start、exit、setting。點擊 setting 會跳出 set player number 和 set mode 的選項，點擊 set player number 則可以選擇 2~4 個 player 的選項，點擊 set mode 可以選擇 player vs player 的選項，點擊 start 開始遊戲，點擊 exit 則離開遊戲。

## 二、 遊戲開始

### ● 洗牌、梅花三、發牌、手牌排序:

首先，透過四種花色和十三種數字的雙層迴圈產生一副 52 張的牌，存進 Deck(QList<Card>) 中。

洗牌：

接下來透過隨機數引擎類別，使用給定的種子值初始化引擎物件，再使用隨機數引擎對 deck 容器中的元素進行隨機排序，達到洗牌的效果。

找梅花三：

先透過迴圈遍歷 Deck 的每個項目，在每次迴圈中，檢查當前卡牌是否為梅花三，即檢查卡牌的花色是否為梅花且點數是否為 3，如果找到梅花三，將其索引值存儲到 index 變數中，並使用 break 跳出迴圈。接著檢查 index 的值是否為-1，若不是-1，則表示找到了梅花三。將梅花三從原來的位置取出（takeAt()），然後將其插入到容器的最前面（push\_front()），這樣就將梅花三移動到了 Deck 的開頭。

發牌：

接著我們根據不同的玩家數量發牌，如果玩家人數為 4 人，則將牌均勻分為 4 堆，每堆 13 張牌。如果玩家人數為 3 人，則將牌分為 3 堆和 1 堆，前三堆各有 17 張牌，最後一堆為空。如果玩家人數為 2 人，則將牌分為 2 堆和 2 堆，前兩堆各有 26 張牌，後兩堆為空。如此一來在後續的設定上將更加便捷。

手牌順序：

設定顯示第一堆牌堆，即含有梅花三的那一組牌，將第一、二、三、四堆牌分別分給 player1、2、3、4，並設定 currentPlayer 為 player1，即從第一個玩家開始。

### ● 玩家遊玩順序(含 pass):

遊戲開始由第一個玩家出牌，成功出牌(或是 pass)後，畫面將顯示 player2 的手牌，而 player1 的位置移動到原 player4 的位置，也就是玩家的牌堆和頭像順時針移動，即達到逆時針出牌的效果。

- 出牌驗證

在 `isSelectionValid` 的函式中，判斷玩家滑鼠點選的牌是否為合理的組合，並且用 `presenttype` 記錄目前遊戲輪次為哪種型態(單張: 0、對子: 1、順子: 2、葫蘆: 3、鐵支: 4、同花順: 5)如果為不合理的組合則須重新選擇。

首先，`isSelectionValid` 函數會檢查傳入的向量是否為空的。如果向量為空的，函數就回傳 `false` 判定為無效牌組。如果並非空的向量，就將傳入的牌進行排序，並產生一個 `map` 來統計每個數字的牌有多少張。接著根據選擇的牌數量進行不同的檢查：

一張牌:

因一張牌不會有錯誤的組合，故函數回傳 `true`，`presenttype = 0`。

兩張牌:

檢查這兩張牌是否數字相同為一對。如果是回傳 `true` 否則返回 `false`，`presenttype = 1`。

三張或四張牌:

如果選擇了三張或四張牌直接回傳 `false`，因為在這個遊戲規則中，三張或四張牌不能形成有效的組合。

五張牌:

首先檢查這五張牌是否為同花順(由於同花順的組合不包含 JQKA2, QKA12, KA123, 故第一張牌的數字只能在 3~10 之間)，接著檢查是否為特定組合的同花順(如 "23456", "A2345")，如果是同花順就回傳 `true`，`presenttype = 5`，否則接續判斷是否為鐵支。檢查是否有四張相同數字和一張不同數字的牌組(鐵支)，如果是鐵支就回傳 `true`，`presenttype = 4`，否則接續判斷是否為葫蘆。檢查是否有三張相同數字和兩張相同數字的牌組(葫蘆)，如果是葫蘆就回傳 `true`，`presenttype = 3` 否則接續判斷是否為順子。檢查是否為順子時，由於要排除特定的組合(JQKA2, QKA12, KA123)，故第一張牌的數字只能在 3~10 之間，如果皆不符合再接著檢查是否為特定組合的順子(如 "23456", "A2345")，如果是順子就回傳 `true`，`presenttype = 2` 否則回傳 `false`。

在 `isPutValid` 的函式中，透過 `switch` 語法和多個 `case` 檢查手牌是否能夠在當前遊戲的狀況下合理出牌，如果合理就回傳 `true` 否則回傳 `false`。`presenttype` 為當前要出的牌的牌型，而 `lasttype` 則是上一家出的牌型。

以下為各個 `case` 的簡單說明：

case 0(單張):

如果目前要出的手牌為同花順、鐵支則回傳 true，否則如果上一手牌和目前要出的牌都是單張，那麼比較牌的大小和花色來決定是否可以出牌。

case 1(對子):

如果目前要出的手牌為同花順、鐵支則回傳 true，否則如果上一手牌和目前要出的牌都是兩張，那麼先比較牌的大小，如果大小相同則比較花色來決定是否可以出牌。

case 2(順子):

如果目前要出的手牌為同花順、鐵支則回傳 true，否則如果上一手牌和目前要出的牌都是順子，那麼先比較順子數字大小，如果順子數字大小都相同再比較數字最大的花色來決定是否可以出牌，但如果上一副牌與目前要出的牌都是"23456"，則比較 2 的花色大小。

case 3(葫蘆):

如果目前要出的手牌為同花順、鐵支則回傳 true，否則如果上一手牌和目前要出的牌都是葫蘆，則比較三張一樣的牌數字大小來決定是否可以出牌。

case 4(鐵支):

如果目前要出的手牌為同花順，則回傳 true，否則如果上一手牌和目前要出的牌都是鐵支，那麼比較這四張牌的數字大小來決定是否可以出牌。

case 5(同花順):

如果上一手牌和目前要出的牌都是同花順，那麼先比較同花順數字大小，如果同花順數字大小都相同再比較數字最大的花色來決定是否可以出牌，但如果上一副牌與目前要出的牌都是"23456"，則比較 2 的花色大小。

如果 isPutValid 和 isSelectionValid 函式都是 true，就將選到的牌丟到畫面中間，並且在手牌中移除。

- 分數計算:

在出牌的 playCards2 函式中，刪除玩家手牌的同時檢測是否該玩家的手牌大小為 0，若為 0，表示該玩家的手牌出完了，即跳入計算分數的環節。

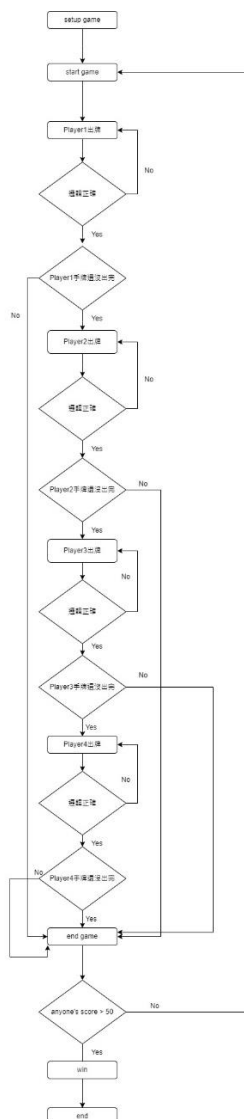
計算：

首先，使用一個迴圈遍歷所有玩家，其中 player 是當前玩家，在迴圈中，對於除了當前玩家之外的其他玩家，檢查是否有數字為 2 的牌（hasCardTwo()函式檢測）。若有數字為 2 的牌，將該玩家手中的牌數乘以 2 加到 score 中，否則只將手中的牌數加到 score 中。完成迴圈後，將 score 的值加到 player 的總得分中（addScore 函式）。然後檢查 player 的總得分是否大於等於 50。如果是，調用 winGame 函式（傳入獲勝的 player）以宣布該玩家獲勝。否則，調用 reGame() 函數以開始新的一局。

### 三、 得分畫面

### 四、 獲勝贏家

### 五、



## 六、 遇到的困難

- 出牌無法顯示在畫面中:

在我們要顯示出出去的牌時，牌的圖片並沒有正確顯示，無論如何利用 qDebug 檢查，確認各個環節都有順利進行，圖片仍然沒有顯示。最後，我們改變呼叫顯示出手的牌的位置，從原本的 `playCards2()` 移到每一輪更新畫面顯示的函式 `updateVisible()` 中，就解決了問題。

- 畫面不順暢:

在製作遊戲的過程中，我們發現幾個畫面不順、卡頓的原因。首先是聲音的使用，在開啟遊戲時的第一聲音效要出來的時候會有較明顯得卡頓。此外，由於我們切換玩家的畫面是不斷刪掉舊的畫新的，所以在切換的瞬間會有一點延遲時間而無法達到流暢的遊戲體驗。

- 時間分配:

這次的 project 與選設的作業撞期，不擅長時間管理的我們可說是相當痛苦，好在最後助教跟老師還是有延期提供補交的機會讓我們能順利完成。