

國立台灣師範大學資訊工程研究所
碩士論文

指導教授：林順喜 博士

電腦麻將溝通程式 MJTalk 的設計與實作及
電腦麻將程式 MahJongDaXia3 的演算法探討

研究生： 林世和 撰

中華民國 108 年 06 月

目錄

摘要.....	3
ABSTRACT.....	4
致謝.....	5
第一章 緒論.....	6
1.1 研究背景.....	6
1.2 研究目的.....	6
1.3 論文架構.....	8
第二章 文獻探討.....	9
2.1 台灣麻將介紹.....	9
2.2 比賽規則簡介.....	11
2.3 相關論文及程式介紹.....	12
第三章 MJTALK 程式實作與 MAHJONGDAXIA 的演算法探討.....	15
3.1 麻將平台與人工智慧程式溝通指令.....	15
3.2 MJTALK 程式流程圖.....	20
3.3 麻將平台程式 MJTALK 所需功能及實作.....	20
3.4 比賽局數期望值計算.....	35

3.5 MAHJONGDAXIA3 的演算法探討	37
3.6 MAHJONGDAXIA3 規則改動	43
第四章 實驗與成果.....	49
第五章 結論與展望.....	53
附錄	54
參考文獻.....	56



摘要

電腦麻將溝程式 MJTalk 的設計與實作及
電腦麻將程式 MahJongDaXia3 的演算法探討

林世和

由於目前麻將程式比賽時，都需依賴提供的遊戲平台，該平台的好處是圖形化介面及完整的功能，但還是有許多不足的地方，像是無法重複開桌測試，每次測試完後都需重啟伺服器。導致需要大量測試時，時間上的浪費，因此我們決定自行研發一套電腦麻將溝程式 MJTalk。在這篇論文前半段將會討論到 MJTalk 的設計與實作。在本論文所研發的電腦麻將溝程式 MJTalk，採用 `stdin/output` 方式溝通，使用隨機亂數來模擬洗牌及發牌，而其他程式間的溝通指令則依照交大林正宏所撰「麻將平台與人工智慧程式溝通協定」中的規定來傳輸，其餘的功能像是胡牌判斷、特殊牌型判斷、代打模式等設計都會在論文中做探討。

2016 年由於 AlphaGo 打敗了人類的世界棋王李世石，造成人工智慧、大數據、深度類神經網路等該領域科技蓬勃發展。而在人機對戰的棋類遊戲中，麻將由於多人遊戲、不完全資訊跟機率問題，屬於運氣成分比較重的遊戲，其他遊戲常用的演算法較難適用於麻將，因此一直以來都不容易跟人類玩家抗衡，相關算法以及論文討論也相對上比較少量，而在這篇論文後半段將會討論到關於電腦麻將程式 MahJongDaXia3 的演算法探討。

由於現今電腦麻將程式的主流都是在進攻方面做研究，如何最快速的達成胡牌，才是它們所追求的，因此忽略了防守方面的演算法探討。如何在進攻與防守間達成最佳的平衡也是一個重要的課題，而在論文的後半部則會對於電腦麻將程式 MahJongDaXia3 的演算法做防守方面的探討及改進。

關鍵字:人工智慧、麻將平台與人工智慧程式溝通協定、麻將對局

ABSTRACT

The current Mahjong program competitions rely on the game platform provided by National Chiao Tung University. The platform benefits are graphical interfaces and complete functions, but there are still many deficiencies, such as the inability to repeat the table test, the server needs to be restarted after each test, and so on. This leads to a waste of time when a large number of tests are required, so we decided to develop our own game platform MJTalk. The design and implementation of MJTalk will be discussed in the first half of this thesis. The computer Mahjong communication program MJTalk uses the standinput/standoutput method to communicate and uses random-number generation to develop a card shuffling and dealing. The communication commands between other programs are written according to the protocol defined by Cheng-Hung Lin of National Chiao Tung University. The remaining functions such as claiming a tile to win judgment, special card type judgment, and the pinch-hitting mode will be discussed.

In 2016, AlphaGo beat Lee Sedol, one of the world's top Go players. This resulted in the blooming of artificial intelligence, big data, and deep neural network researches. Mahjong is a multiplayer game and an imperfect information game with probability problem that makes it very unpredictable. Most conventional gaming algorithms are difficult to be applied on Mahjong. Therefore, most computer Mahjong programs are not easy to compete with human players. The related algorithms and paper discussions are relatively few. In the second half of this thesis, we will discuss the algorithms of a computer Mahjong program MahjongDaXia3.

Since the mainstream of today's computer Mahjong programs is to conduct research on offense, how to achieve claiming a tile to win is what they are pursuing. Therefore, most of them ignore the defensive issue. How to achieve the best balance between offense and defense is also an important topic. In the second half of the thesis, the defensive aspects of the computer Mahjong DaJiDiaXia3 program will be discussed and improved.

致謝

在這段碩士學位求學過程中，首要感謝林順喜教授在這段時間的關懷與指導，當改善麻將程式遇到瓶頸效果不如預期時，給了我開發麻將溝通平台的建議，讓我有新的研究方向，使得 MJTalk 誕生了。

再感謝家人的支持，讓我能專心在課業上，不用為了錢煩惱，能心無旁騖的完成碩士學位。

感謝實驗室的各位同學的付出與支持，志宏、俊緯學長給了許多程式研發上的建議跟幫助。課業上則是彥吉、品儒、昌龍、羽恆跟我組隊一起過關或著幫我解惑，彼此相互陪伴鼓勵。測試方面則要感謝沅燁學弟幫我抓出許多功能上的不足跟程式 Bug，讓我的 MJTalk 趨於完善。在這段求學的過程中許多問題，都有賴各位的建言與參與，感謝各位。

最後再次感謝林順喜教授跟沅燁學弟，是因為教授，所以才有 MJTalk 的誕生。沅燁學弟則是讓我在研發麻將程式這領域內不用孤軍奮戰，才有這份論文的完成。

第一章 緒論

1.1 研究背景

目前在台灣舉辦電腦對局的研討會，均有舉辦麻將對局比賽。TCGA(Taiwan Computer Game Association)與 TAAI(Taiwanese Association for Artificial Intelligence)，分別在每年六月及十一月時舉辦，為台灣電腦對局的兩大盛會。而在國際上有 ICGA(International Computer Games Association)，也有比賽麻將項目，是讓有志於研發麻將程式的研究者站上國際舞台的機會。

由於目前麻將程式測試時，都須仰賴交大的 cycgame 比賽平台，每測試完一個回合都須要重啟平台，造成測試上的困難及時間上的浪費。俗話說的好工欲善其事，必先利其器，連測試工具都要靠他人，又如何好好的研發程式呢?因此才有了研發麻將溝程式-MJTalk 的念頭。

麻將規則各個地區都不盡相同，而交大的 cycgame 平台則是以一般的台灣麻將為基礎設計的。而 MJTalk 則是客製化，自己從無到有設計出來的，想加哪種功能只要想的出來演算法都能自己增加跟修改，因此並不只限於台灣麻將的測試。舉例來說，像是日本麻將的手牌張數只有 13 張跟胡牌條件一定要有特殊牌型才能胡牌等規則上的不同，只要改變 MJTalk 中的玩家手牌陣列大小及在胡牌判斷演算法中加入「特殊牌型才能胡牌的條件判斷」，就能適用於日本麻將的測試。簡單來說，MJTalk 有很高的彈性自己就可以去改良成適用各區域麻將遊戲的測試平台。

1.2 研究目的

麻將可謂本國的國粹，是中華民族的傳統技藝，原稱作「麻雀」；麻將自中國流傳至各國，並且隨著各國、各地的風俗民情漸漸地在規則上有所變化。台灣麻將是民國初年時傳入，當時流傳的是廣東十三張，而後才演變成現今的台灣十六張，規則和台數計算皆已和廣東十三張不同，大多數的台灣人會打的都以台灣十六張為基準。

MJTalk 的開發以比賽平台 cycgame 所提供的指令標準為準，採 standinput/standoutput 方式與其他麻將 program 作溝通。規則採用台灣十六張麻將的規則，並且去除掉花牌八張，所以實際上比賽時只有：1~9 萬各四張、1~9 筒各四張、1~9 條(索)各四張、東西南北中發白七張字牌各四張，合計 136 張。勝利條件為達成「胡牌」的牌型，「胡牌」要有五「組」與一「眼」的組合共 17 張。「組」分兩種，一種為連續序數牌三張的「順」，另一種為三張相同的「刻」。「眼」則為兩張相同的牌，分數採計以 1000 分為胡牌底分，特殊牌型將有加分的「台」，每台 500 分(此分數為比賽規定的)。

此研究目的就是希望能提供另一種簡單的測試工具給他人使用，由於是非圖形化介面，比賽時還是以比賽平台 cycgame 為主。但研發麻將程式期間如果需要測試時，就可以使用 MJTalk 來測試，可以省去大量設定及操作時間，簡單來說 MJTalk 就是一個簡化版的 cycgame 測試平台。

至於麻將程式防禦策略方面，由於目前大部分的麻將程式對於進攻的研究都大於防守，以至於普遍放槍率都偏高。因此想在放槍率跟胡牌率中取得一個較佳的平衡，嘗試透過一些人類玩家的經驗與技巧，來提升麻將程式的效能，開發出一套具有能和人類玩家競爭的麻將電腦程式。

在許多電腦對局的程式中，審局函式往往是最為重要關鍵的部份，它會決定程式對於牌型好壞或者是盤面優劣的認知，設計一個良好與正確的審局函式將會是最為困難的部分。由於麻將必須依靠現有的資訊去決定捨哪一張牌以取得最高的效益，由於是不完全資訊因此許多參數的設定上，往往是毫無依據的猜測或者是由實驗調教出來的數據。這些猜測可以是由人類的經驗法則來套用，數據則是多次模擬情況後所歸納出來的，因此如何找出最好的經驗法則作為規則或著期望值最高的數據來提升麻將程式的效能，是我們要好好探討的。

1.3 論文架構



圖 1-1 論文架構

第二章 文獻探討

2.1 台灣麻將介紹

以下取自維基百科介紹「台灣麻將」：

十六張牌：開局時間家取 16 張牌，莊家則取 17 張牌。

吃牌的擺放：兩張中間。例如三四萬吃五萬，把五萬放在三萬和四萬中間。（讓其他人知道吃的是哪張。）

連莊&拉莊：連莊，即連續做一次莊。每連莊一次便加上一台。「拉莊」，即是連莊數加倍，例如：

連 1 拉 1+莊家=3 台，連 2 拉 2+莊家=5 台，連 3 拉 3+莊家=7 台，連 N 拉 N+莊家= $2 \times N + 1$ 台。

搬風：四人先隨機決定。再從牌中挑出東、南、西、北各一張，蓋起來打亂。玩家以丟三顆骰子的方式，決定由誰最先來抽牌，最先抽牌的人的位子是東風位，再從該位玩家逆時針方向輪流抽牌。

5、9、13、17 點，是由丟骰者自己開始抽牌。

6、10、14、18 點，是由丟骰者下家開始抽牌。

3、7、11、15 點，是由丟骰者對家開始抽牌。

4、8、12、16 點，是由丟骰者上家開始抽牌。

抽到「東」的人坐東風位。

起莊：堆牌完成後，由坐在東風位的玩家丟三顆骰子，決定誰是第一個莊家。並且由莊家開始，從自家面前的十八幢牌的最右邊開始數（順時針），留下骰子所指幢數，一次取兩幢（四張）牌，逆時針

四家輪流取四次完成配牌。此時每人手上各有十六張牌。例：若骰子擲出 11 點，則由西風位玩家當莊，並且由莊家開始，從面前取走第十二、十三幢牌（留下右邊十一幢），下家以此類推。

補花：開門後，莊家手裡共有十七張牌，檢查牌中是否有「花牌」，若有則要一次全部出示於「手牌前方」，再從嶺上牌取走相對應的數量讓自己手牌繼續保持十七張牌，稱做「補花」。莊家補牌結束後，示意閒家可以依序補花。若補牌補到「花牌」，則要「後補」，程序跟補花相同。

流局：若牌局進行到剩下「八幢牌」（十六張牌）留在牌牆上還沒有人胡牌，則此局結束，俗稱「鐵八墩」或「黃一莊」。流局後莊家連莊數加一。

計分方式：以負方看勝方的視角計算：1 底+台數

若莊家涉及輸贏則必須加上 $[2n+1]$ 台， n =連莊次數 故正常情況有以下特性：

1. 榮胡時非放槍者不需要賠（本身無關輸贏）
2. 自摸時勝方所贏的錢是同樣牌型榮胡的三倍起跳（不計莊家連莊情況）
3. 被自摸時沒有分攤賠問題，所有輸家皆賠 $[1 \text{ 底} + \text{台數}]$ 起跳（不計莊家連莊情況）
4. 底、台數皆與負方所賠分數成「加減法計算」而非倍數、指數計算

吃牌：上家丟出來的牌和自己手上二張牌湊成順子，則可吃牌。

碰牌：任何一家所丟出來的牌與自己的對子湊成刻子，則可碰牌。

槓牌：分為明槓和暗槓，明槓又分為大明槓(碰槓)和小明槓(加槓)，如下。

碰槓：任何一家所丟出來的牌與自己的刻子湊成槓子，則可碰槓。

加槓：若自己已經持有一組刻子，隨後又摸進該牌，讓刻子成為槓子，則可加槓。

暗槓：若自己已經持有一組槓子，則可暗槓。

胡牌：達成 5 組 1 眼即為胡牌，其中組包含「順」、「刻」兩種形式。

胡牌優先順序大於碰（槓）牌，碰（槓）牌的優先順序大於吃牌。

面子：三張牌組成的刻子或順子，胡牌需要五面一眼。

刻子：三張同樣的牌組成的面子。

順子：三張連續的數牌組成的面子。如四五六萬。

搭：差一張牌即可成為面子的兩張牌。已完成的面子也稱為搭。

台數計算：見附錄。

2.2 比賽規則簡介

以下引用吳俊緯「電腦麻將程式 MahJongDaXia 的設計與實作」中的說明。

現行於 TCGA、TAAI、ICGA 進行的電腦麻將對局比賽皆採用 Cycgame 所提供的比賽平台，採用台灣十六張麻將規則，在此之上做了一些變動。1. 比賽不採用花牌機制，因此實際總牌數為 136 張。2. 比賽無連莊設定，莊家輪流擔任，莊家採計一台。3. 分數採計胡牌以 1000 分為基本底，特殊牌型將有加分的「台」，每台 500 分，自摸收三家分數，放槍為一對一賠分。目前比賽賽制以四人互為上下家共六種排法(如表 2-1)，每種排法打 16 局，合計為 96 局。換一組牌再打 96 局，所以四人比賽要打 192 局，而為求公平，每 16 局中的所有起手牌型每位玩家都會拿到一次。

東	A	A	A	A	A	A
南	B	D	B	C	C	D
西	C	C	D	D	B	B
北	D	B	C	B	D	C

表 2-1 四人互為上下家的六種排法

而關於比賽的流程，若比賽隊伍為 4 隊，則取分數最高前三名為冠、亞、季 軍。若比賽隊伍為 6 隊，則抽籤分 A、B 兩組，各取分數最高兩名進入 C 組賽事 決定冠、亞軍，A、B、C 組剩餘玩家進入 D 組賽事，取最高分為季軍。

2.3 相關論文及程式介紹

電腦麻將的研究相關論文大多以國內為主，201 年由國立交通大學林正宏所撰「麻將平台與人工智慧程式溝通協定」，平台對於麻將牌張定義方式為將每一張牌定義一個唯一的號碼，作為平台與程式溝通之用；麻將平台和人工智慧程式溝通方法則是使用 `stdin`/`stdout` 的重定向方式來溝通；麻將平台與人工智慧程式溝通範例時序圖如下圖 2-1。最後則是麻將平台對於各種動作指令的設定。



圖 2-1 麻將平台對於各種動作的溝通優先順序

再來是 2013 年由國立高雄應用科技大學許綸洲所撰「麻將人工智慧設計」，其中提到蒙地卡

羅法應用於麻將時，由於計算量過大，需時過長，不適用於短時間思考的麻將遊戲上，也間接驗證了許多遊戲中使用的搜尋演算法並不適用於麻將。其中提到的棄牌系統是為了找出最佳的安全牌，因此提出一個扣分機制，來解決安全牌判斷問題，如下圖 2-2。實驗結果並無法有效減少輸牌機率，可見麻將在防禦這部分，還沒有有效的一套公式來減少放槍率。

非字牌(萬筒條)公式:

$$\text{應扣分數} = 2 \frac{\text{本身棄牌張數} * \text{最近一張牌位置}}{\text{棄牌總數}} + (2 \frac{\text{相關棄牌張數} * \text{最近一張牌位置}}{\text{棄牌總數}} / 2)$$

字牌公式:

$$\text{應扣分數} = 2 \frac{\text{本身棄牌張數}}{\text{棄牌總數}}$$

圖 2-2 扣分機制公式

最後是 2015 年由國立台灣師範大學吳俊緯所撰「電腦麻將程式 MahJongDaXia 的設計與實作」，其中以各張牌的進牌機率來取代傳統的參數權重，也是少數的規則為導向的麻將程式。其主要決策分為：拆分-將牌型拆解出全部的組合方式、整合-將牌型各組合的可能性整合成一物件資訊、比較-將各牌型物件比較優劣，得到該丟出的牌。這是一支完全以最快達到胡牌為主要目標的程式，不管放槍等因素，防禦決策完全沒有，因此規則設計的好壞非常重要。下圖 2-3 為 MahJongDaXia 的簡易程式流程圖。

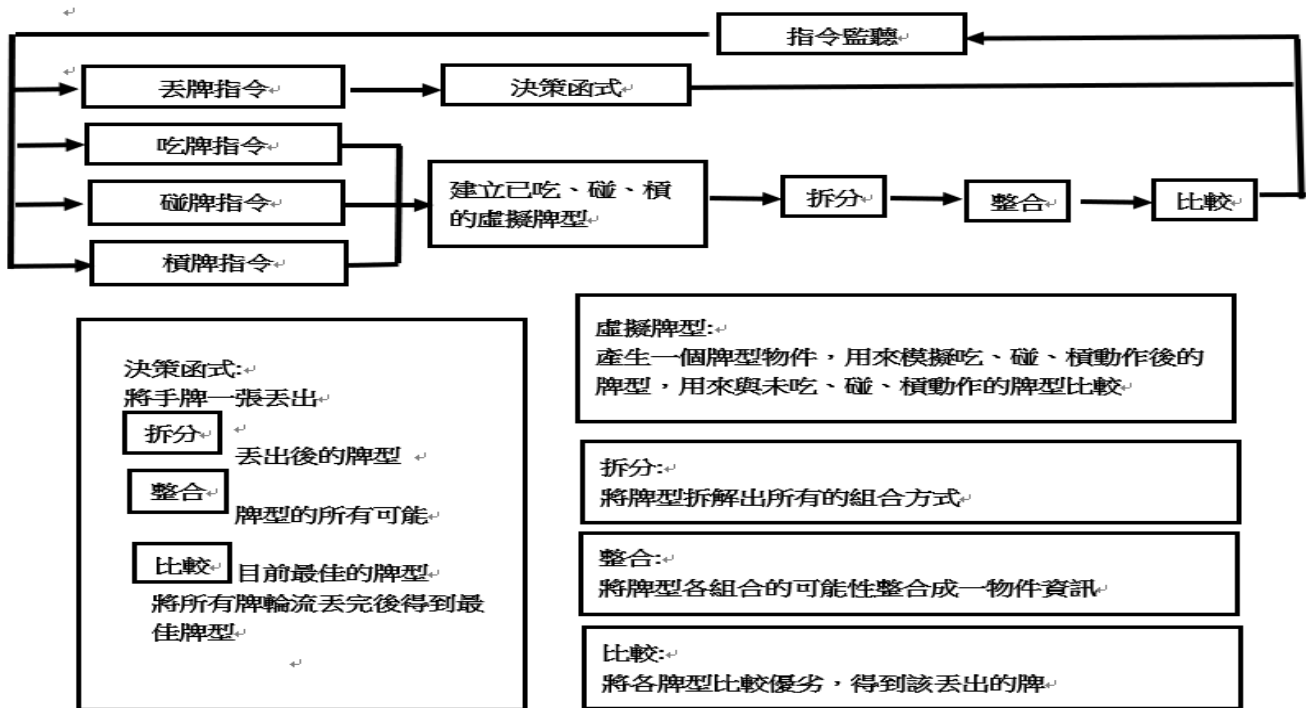


圖 2-3 MahJongDaXia 的簡易程式流程圖



第三章 MJTalk 程式實作與 MahJongDaXia 的演算法探討

由於是第一次撰寫溝通程式，其設計原理完全是依照「麻將平台與人工智慧程式溝通協定」中的指令為基礎來研發，此章中會分節介紹麻將平台與人工智慧程式溝通指令、麻將平台程式執行流程、麻將平台程式所需功能及實作、比賽局數期望值計算。最後再介紹對於麻將程式 MahJongDaXia3 規則上的改動。

3.1 麻將平台與人工智慧程式溝通指令

本小節引用自「麻將平台與人工智慧程式溝通協定」中的指令說明。

麻將牌張定義共 144 張牌，包括萬、條、筒、字和花牌，將每一張牌定義一個唯一的號碼，作為平台與程式溝通之用。如表 3-1。

1. 定義花牌

牌張	春	夏	秋	冬	梅	蘭	菊	竹
號碼	10	20	30	40	50	60	70	80

表 1: 花牌定義

2. 定義萬子牌

牌張	一萬	二萬	三萬	四萬	五萬	六萬	七萬	八萬	九萬
號碼	110~113	120~123	130~133	140~143	150~153	160~163	170~173	180~183	190~193

表 2: 萬子牌定義

3. 定義條子牌

牌張	一條	二條	三條	四條	五條	六條	七條	八條	九條
號碼	210~213	220~223	230~233	240~243	250~253	260~263	270~273	280~283	290~293

表 3: 條子牌定義

4. 定義筒子牌

牌張	一筒	二筒	三筒	四筒	五筒	六筒	七筒	八筒	九筒
號碼	310~313	320~323	330~333	340~343	350~353	360~363	370~373	380~383	390~393

表 4: 筒子牌定義

5. 定義字牌

牌張	東風	南風	西風	北風	紅中	青發	白皮
號碼	410~413	420~423	430~433	440~443	450~453	460~463	470~473

表 3-1：字牌定義

人工智慧程式使用 `stdin` 接收麻將平台 Server 傳送的指令；並使用 `stdout` 傳送指令給麻將平台。如圖 3-1。

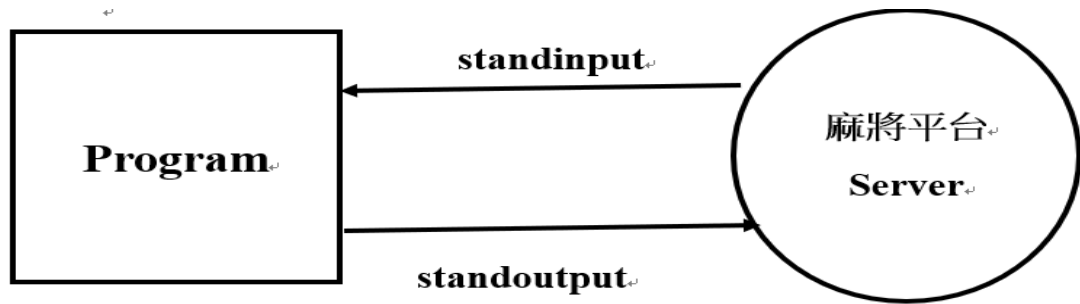


圖 3-1 麻將平台和人工智慧程式溝通方法

麻將平台傳給人工智慧程式的指令如下：

1. `/start MJ <position> <ProgramPath>`

遊戲開始時，Server 以 `/start MJ` 告知 Program 遊戲開始，並給予 Program 的位置和 Program 程式的名字。Program 的名字代表被呼叫程式的檔名為 `ProgramPath.exe`。

舉例：`/start MJ 2 Program_1` 此例表示：名為 `Program_1.exe`，座位為 2 號。

2. `/initGame <wind><game><open><banker><continue>`

送出 `/start` 指令後，Server 以 `/initGame` 告知 Program 該局的訊息，包括幾風、幾局、開門位置、莊家位置和目前連莊數目。風 `<wind>` 和局 `<game>` 為 `east`、`south`、`west`、`north` 其中之一。

舉例：`/initGame east east 3 2 0` 此例表示：風東局，3 號玩家開門，2 號玩家為莊家，連莊數為 0。

3. `/initCard <card1> <card2>...<card16>`

送出 `/initGame` 指令後，Server 以 `/initCard` 告知玩家起始拿到哪些牌。

若為莊家則拿 17 張牌；若非莊家則拿 16 張牌。

舉例：`/initCard 121 122 142 143 152 220 241 260 270 342 351 370 383 390 462 463` 此例表示：Program 拿到 121 122 142 143 152 220 241 260 270 342 351 370 383 390 462 463。

4. /mo <card>

遊戲進行時，Server 傳/mo 指令給 Program，告知 Program 摸到哪張牌。

舉例：/mo 410 此例表示：Program 摸到牌 410。

5. /ask throw

遊戲進行時，Server 要求玩家丟牌，會傳送/ask throw 要求玩家丟牌。可能發生在玩家摸、吃、碰、槓牌後。

6. /ask pong

遊戲進行時，Server 要求玩家碰牌，會傳送/ask pong 詢問玩家要不要碰牌

7. /ask gong

遊戲進行時，Server 要求玩家槓牌，會傳送/ask gong 詢問玩家要不要槓牌。

8. /ask eat

遊戲進行時，Server 要求玩家吃牌，會傳送/ask eat 詢問玩家要不要吃牌。

9. /ask hu

遊戲進行時，Server 要求玩家胡牌，會傳送/ask hu 詢問玩家要不要胡牌。

10. /throw <position> <card>

遊戲進行時，玩家丟出牌後，Server 將廣播/throw 訊息給四個玩家知道。

舉例：/throw 3 410 此例表示：3 號玩家打出牌 410。

11. /eat <position> <card1> <card2> <card3>

遊戲進行時，有玩家吃牌，Server 將廣播/eat 訊息給四個玩家知道。

吃來的牌放中間，且小於<card1>、<card3>。

舉例：/eat 4 280 271 292 此例表示：4 號玩家吃 280、271、292 三張牌。

12. /pong <position> <card1> <card2> <card3>

遊戲進行時，有玩家碰牌，Server 將廣播/pong 訊息給四個玩家知道。

碰牌順序為<card1>、<card2>、<card3>由左至右遞增。

舉例：/pong 2 320 321 323 此例表示：2 號玩家碰 320、321、324 三張牌。

13. /gong <position> 4 <card1> <card2> <card3> <card4>

遊戲進行時，有玩家槓牌，且為明槓，Server 將廣播/gong 4 訊息給四個玩家知道。

槓牌順序為<card1>、<card2>、<card3>、<card4>由左至右遞增。

舉例：/gong 2 4 160 161 162 163 此例表示：2 號玩家明槓 160、161、162、163 四張牌。

14. /gong <position>0

/gong <position> 0 <card1> < card2> < card3> < card4>

遊戲進行時，有玩家槓牌，且為暗槓，Server 將廣播/gong <position> 0 訊息給四個玩家知道。

槓牌順序為<card1>、<card2>、<card3>、<card4>由左至右遞增。

若 Program 位置為暗槓時的玩家位置，則 Server 會傳「/gong <position> 0 <card1> <card2> <card3> <card4>」給 Program。

若 Program 位置不是暗槓時的玩家位置，則 Server 會傳「/gong <position> 0」給 Program。

舉例：/gong 2 0 160 161 162 163 此例表示：Server 傳給 2 號玩家，自己暗槓 160、161、162、163 四張牌。/gong 2 0 此例表示 Server 傳給非 2 號玩家，2 號玩家暗槓四張牌。

15. /gong <position> 1 <card>

遊戲進行時，有玩家槓牌，且為補槓，Server 將廣播/gong 1 訊息給四個玩家知道。

舉例：/gong 1 1 154 此例表示：1 號玩家補槓牌 154。

16. /hu <position> <hucard> <card1> <card2>...

遊戲進行時，有玩家胡牌，Server 將廣播/hu 訊息給四個玩家知道。其中<hucard>為敵家放槍或自摸的牌，<card1> <card2>...為胡牌玩家還沒現出來的牌。

舉例：/hu 3 232 112 120 132 192 193 231 233 372 382 392 471 472 473 此例表示：3 號玩家胡 232 這張牌，3 號玩家未現出來的手牌為 112 120 132 192 193 231 233 372 382 392 471 472 473。

17. /exit <score1> <score1> <score3> <score4>

Server 要求 Program 中斷時傳給 Program 的指令。通常發生在一局結束時，也可能發生在 Program 超時、Program 傳送指令錯誤、Server 異常結束等狀況。<score1>、<score2>、<score3>、<score4> 分別為座位 1、2、3、4 的玩家該局得到的分數。

人工智慧程式傳給麻將平台的指令如下：

1. /throw <card>

遊戲進行時，Server 送出/ask throw 指令後，Program 丟牌使用/throw 指令。

舉例：/throw 112 此例表示：Program 丟出牌 112。

2. /eat <card1> <card2>

遊戲進行時，Server 送出/ask eat 指令後，Program 吃牌使用/eat 指令。<card1>、<card2>為 Program 自己擁有的牌，用來吃上家的丟出的牌。

舉例：/eat 132 145 此例表示：Program 用 132、145 兩張牌吃上家丟出的牌。

3. /pong <card1> <card2>

遊戲進行時，Server 送出/ask pong 指令後，Program 碰牌使用/pong。<card1>、<card2>為 Program 自己擁有的牌，用來碰上家的丟出的牌。

舉例：/pong 112 113 此例表示：Program 用 112、113 兩張牌碰敵家丟出的牌。

4. /gong 4 <card1> <card2> <card3>

遊戲進行時，Server 送出/ask gong 指令後，Program 明槓牌使用/gong 4 指令。明槓時使用指令「/gong 4 <card1> <card2> <card3>」，其中 card1、card2、card3 為 Program 自己擁有的牌，用來槓敵家丟出的牌。

5. /gong 0 <card1> <card2> <card3> <card4>

遊戲進行時，Server 送出/ask gong 指令後，Program 暗槓牌使用/gong 0 指令。暗槓時使用指令「/gong 0 <card1> <card2> <card3> <card4>」，其中<card1>、<card2>、<card3>、<card4>為 Program 自己擁有的牌。

6. /gong 1 <card>

遊戲進行時，Server 送出/ask gong 指令後，Program 補槓牌使用/gong 1 指令。補槓時使用指令「/gong 1 <card>」，其中 card 為 Program 自己擁有的牌，用來和自己已經碰出的刻子型成槓子。

7. /hu

遊戲進行時，Server 送出/ask hu 指令後，Program 胡牌使用/hu 指令。

8. /pass

遊戲進行時，Server 送出/ask eat、/ask pong、/ask gong、/ask hu 指令後，若 Program 不想吃、碰、槓、

胡，則使用/pass 指令。

3.2 MJTalk 程式流程圖

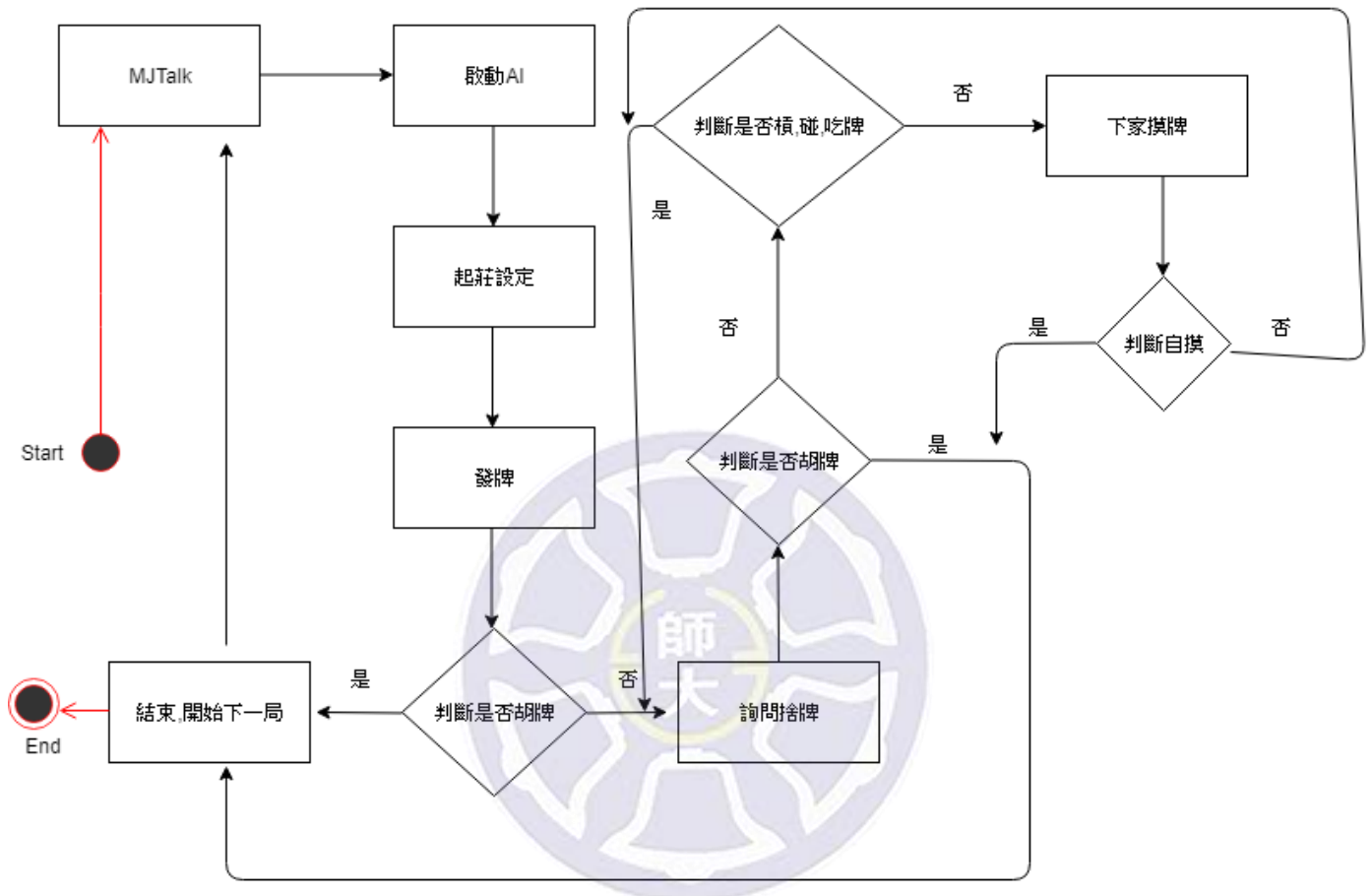


圖 3-2 MJTalk 簡易流程圖

流程如圖 3-2 所示，如下：

- 1.輸入 Program 編號及位置->啟動 Program。
- 2.平台隨機起莊後傳送指令給 Program。
- 3.亂數產生手牌傳送給 Program。
- 4.監聽指令。

3.3 麻將平台程式 MJTalk 所需功能及實作

1.起莊：

i.隨機產生 1-4 的亂數，設定莊家及閒家的位置後，再隨機產生 3~18 的亂數用來模擬莊家擲骰子取牌動作，並依東南西北順時針的順序去數，最後依莊家位置去計算開門位置。

EX：假設隨機亂數取得 1，表示 Program1 為莊家其他 Program 為閒家，記錄莊家 Program 編號，以便之後發牌時莊家的 Program 手牌多發一張，之後亂數取得 15(骰子數 15)，則 15 數到 Program3，表示從 Program3 的位置開始拿牌，拿完牌後取得開門位置。如下圖 3-3。



圖 3-3 起莊

2.發牌

i.依照每張牌的編號，隨機打亂順序排好，共 136 張牌。一次 16 張牌的方式，分別傳給 Program1~Program4，莊家多發一張，並儲存各個程式的手牌，以便後面判斷吃碰槓胡的情況。

ii.利用指標記錄下一張要發的牌的位置。

3.丟牌

i.傳送丟牌指令給莊家，並取得 Program 丟牌後，傳送給所有 Program。

EX：莊家為 Program1，傳送/ask throw 給 Program1，取得/throw 330，傳送/throw 1 330 指令給所有 Program。

ii.尋問是否對所丟的牌依序做胡、槓、碰、吃等動作，如有則進入胡、槓、碰、吃等動作函式。

iii.進入摸牌。

4.摸牌

- i.將指標所指的牌發給下一個玩家，並更新手牌。
- ii.發送丟牌指令，接收 Program 回傳的丟牌指令，知道丟哪張牌後，更新玩家手牌。
- iii.發送丟牌資訊給其他玩家。

EX：指標目前所指為 413，傳送/mo 413 指令給 Program1，並更新 Program1 的手牌，發送/ask throw 指令給 Program2，取得/throw 440，更新玩家手牌後，發送丟牌資訊給其他 Program。如下圖 3-4。



圖 3-4 摸牌

5.吃牌/pass

- i.依照接收到的丟牌資訊，去判斷下家是否能吃牌。

判斷方法：依照丟的牌去搜尋下家手牌是否有此張牌+(-)2、+(-)1 的牌。

EX：丟 3 萬，搜尋下家手牌是否有 1 萬、2 萬或者 4 萬、5 萬或者 2 萬、4 萬。如下圖 3-5。

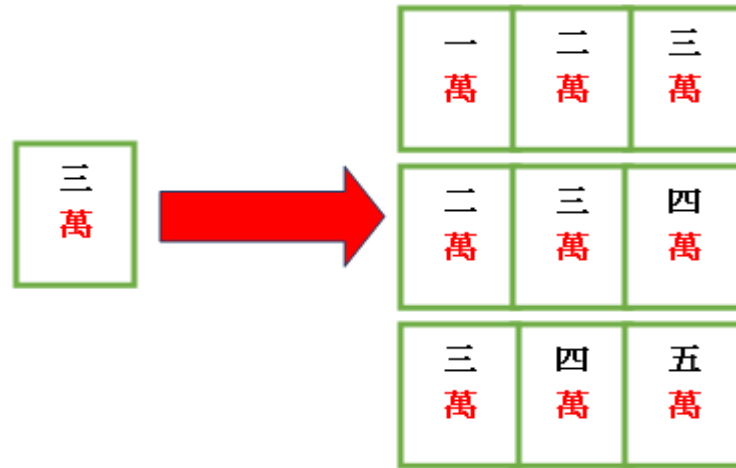


圖 3-5 吃牌 i

ii.如可以吃牌，發送吃牌指令給 Program，接收 Program 回傳的指令，如吃牌則更新手牌，並將吃牌的資訊發送給所有 Program 後存入吃牌陣列內，以便後續作胡牌的判斷。

EX：假設 Program1 丟牌(/throw 131)，發送/ask eat 指令給 Program2，取得/eat 140 150，更新 Program2 手牌，並將/eat 2 140 131 150 指令傳給所有 Program，將吃牌資訊 131 140 150 存入吃牌陣列。如圖 3-6。



圖 3-6 吃牌 ii

iii.如果回傳 pass 則不動作。

6.碰牌/pass

i.依照接收到的丟牌資訊，去判斷其他三家是否能碰牌。

判斷方法：依照丟的牌去搜尋各家手牌是否有同張牌 2 張以上。

EX：丟一萬，搜尋三家手牌，哪家有 2 張以上的一萬。如圖 3-7。

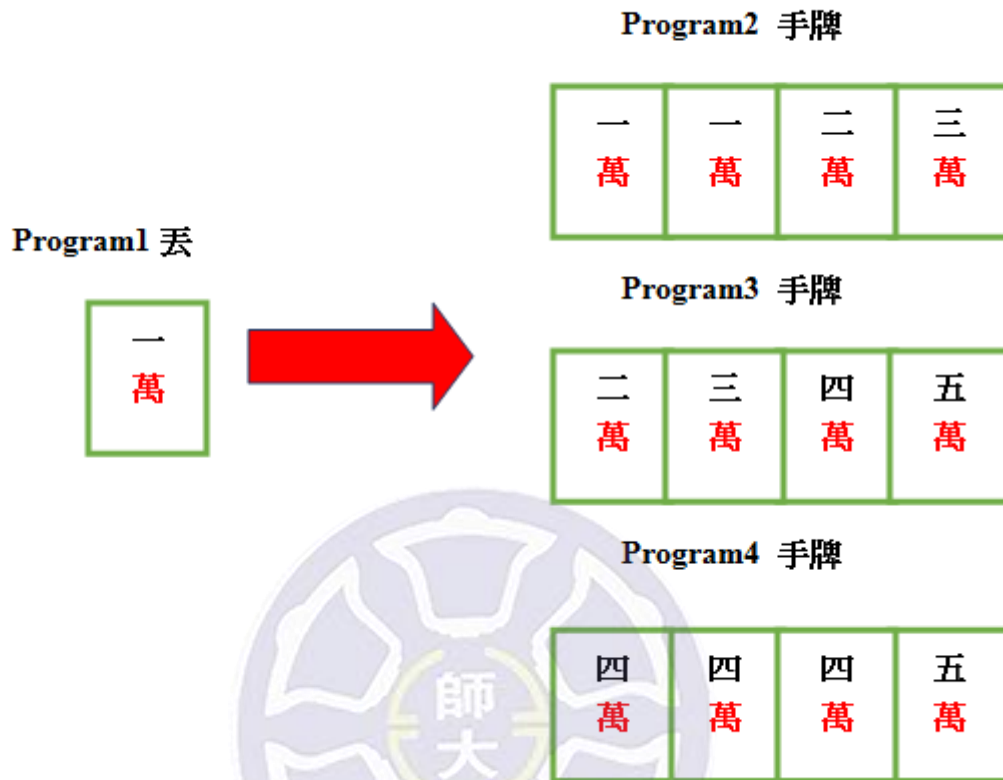


圖 3-7 碰牌 i

ii.如可以碰牌，則發送碰牌指令給那家 Program，接收 Program 回傳的指令，如碰牌則更新手牌，並將碰牌的資訊發送給所有 Program 後存入碰牌陣列內，以便後續作胡牌的判斷。

EX: 假設 Program1 丟牌(/throw 131), 發送/ask pong 指令給 Program2, 取得/pong 130 132, 更新 Program2 手牌, 並將/pong 2 130 131 132 指令傳給所有 Program, 將碰牌資訊 130 131 132 存入碰牌陣列。如圖

3-8。

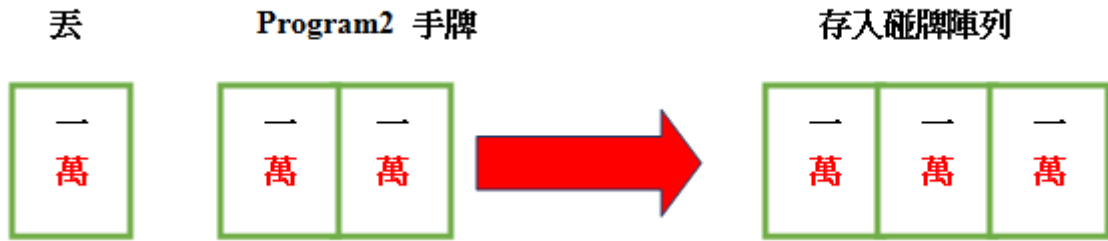


圖 3-8 碰牌 ii

iii. 如果回傳 pass 則不動作。

7. 槓牌/pass

暗槓：

i. Program 依照接收到的摸牌資訊，去判斷是否能暗槓牌。

判斷方法：Program 摸牌時判斷目前 Program 手牌是否有同樣牌 3 張。

EX：摸北風，搜尋自己手牌是否有 3 張北風。

ii. 如果有則傳送暗槓牌指令，平台接收 Program 回傳的指令，如暗槓牌則更新手牌並將暗槓牌的資訊發送給所有 Program 後，存入槓牌陣列內，以便後續做胡牌的判斷。

EX：Program1 摸牌(/mo 110)，發送/ask gong 指令給 Program1，取得/gong 0 110 111 112 113 指令，更新 Program1 手牌，並將/gong 1 440 441 442 443 指令傳給 Program1，將/gong 0 指令傳給其它 Program 後，將槓牌資訊 110 111 112 113 存入槓牌陣列內，以便後續作胡牌的判斷。如圖 3-9。

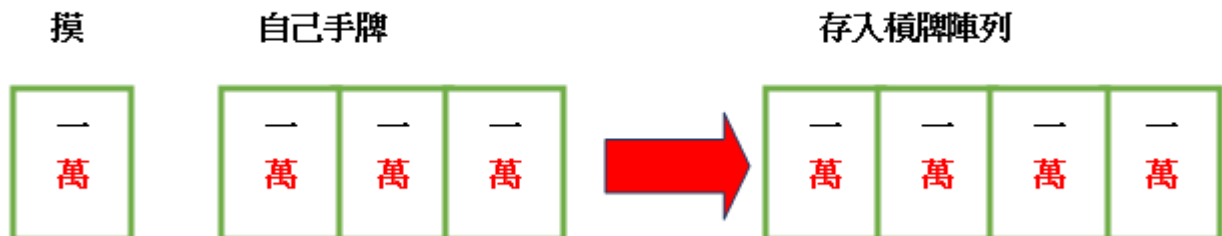


圖 3-9 暗槓

iii. 暗槓後平台傳送/mo 122 指令告知 Program1 程式摸到牌 122，再傳送/ask throw 指令要求 Program1

程式丟牌，而 Program1 程式傳送/throw 412 指令告知平台丟出牌 412。

iv.如果回傳 pass 則不動作。

明槓：

i.依照接收到的丟牌資訊，去判斷其他兩家能否槓牌(規則:不可槓上家的牌)。

判斷方法：依照丟的牌去搜尋另外兩家手牌是否有同張牌 3 張。

EX：丟 1 萬，搜尋其他家手牌是否有 3 張 1 萬。如圖 3-10。

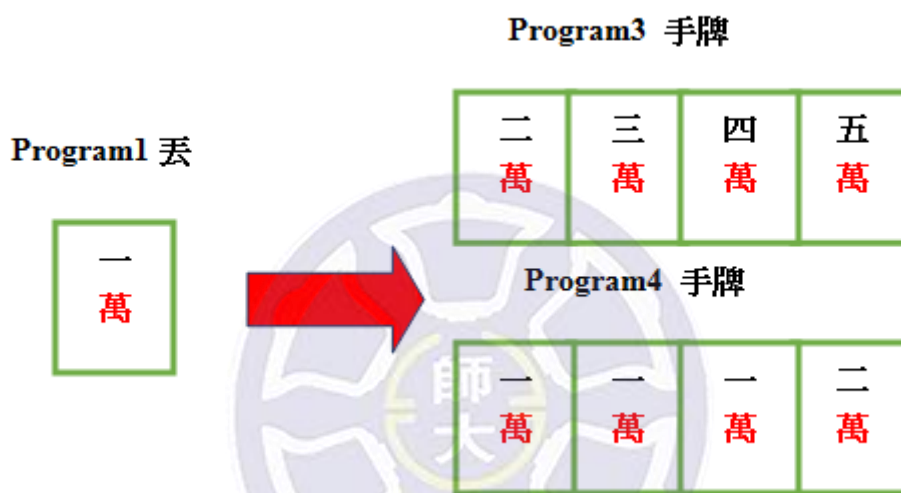


圖 3-10 明槓 i

ii.如有則發送槓牌指令，接收 Program 回傳的指令，如槓牌則更新手牌並將槓牌的資訊發送給所有 Program 後存入槓牌陣列內，以便後續做胡牌的判斷。

EX：Program1 丟 111，發現 Program4 可以槓，發送/ask gong 給 Program4，取得/gong 4 110 112 113 指令，更新 Program4 手牌，並將/gong 4 110 111 112 113 指令傳給所有 Program 後，將槓牌資訊 110 111 112 113 存入槓牌陣列內，以便後續作胡牌的判斷。如圖 3-11。

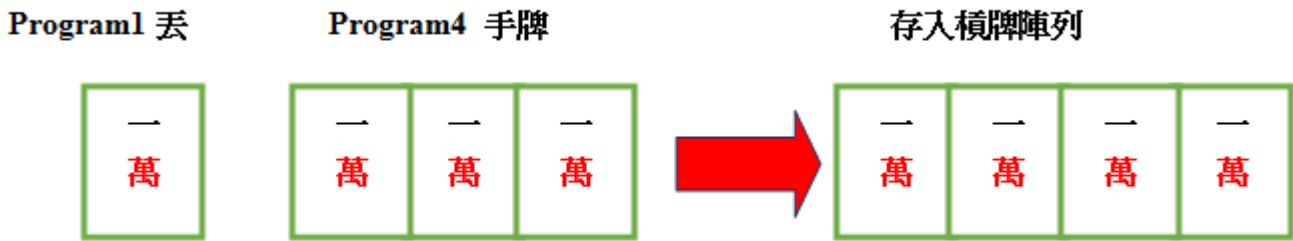


圖 3-11 明槓 ii

iii.明槓後平台傳送/mo 410 指令告知 Program3 程式摸到牌 410，再傳送/ask throw 指令要求 Program3 程式丟牌，而 Program3 程式傳送/throw 410 指令告知平台丟出牌 410。

iv.如果回傳 pass 則不動作。

補槓：

依照接收到的摸牌資訊，去判斷是否能補槓牌。

判斷方法：Program 摸牌時判斷 Program 的碰牌陣列是否有同樣的牌。

EX：摸 112，搜尋碰牌陣列內是否有 110 111 113。

如果有則傳送補槓牌指令，接收 Program 回傳的指令，如補槓牌則更新手牌並將補槓牌的資訊發送給所有 Program 後存入槓牌陣列內後將碰牌陣列內的值刪除，以便後續做胡牌的判斷。

EX：Program1 摸牌(/mo 112)，發送/ask gong 指令給 Program1，取得/gong 1 112 指令，更新 Program1 手牌，並將/gong 1 112 指令傳給所有 Program，將槓牌資訊 110 111 112 113 存入槓牌陣列內並將碰牌陣列內的 110 111 113 刪除，以便後續作胡牌的判斷。如圖 3-12。

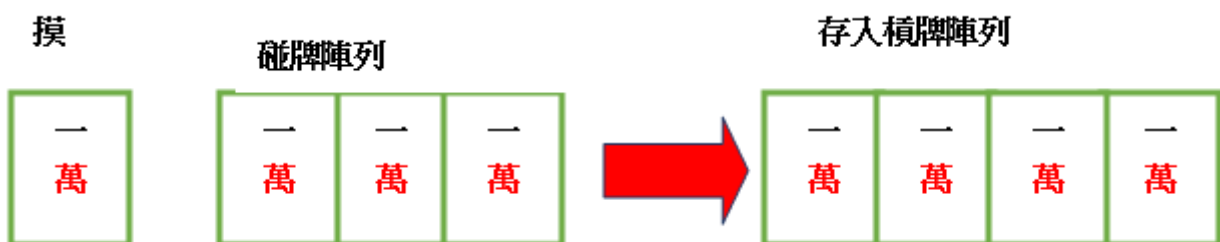


圖 3-12 補槓

iii. 槓牌後平台傳送/mo 220 指令告知 Program1 程式摸到牌 220，再傳送/ask throw 指令要求 Program1 程式丟牌，而 Program1 程式傳送/throw 450 指令告知平台丟出牌 450。

iv. 如果回傳 pass 則不動作。

8. 胡牌

胡牌判斷定理:以下引用 2017 年，威廉斯堡大學數學系的李志光教授的判斷胡牌演算法(參考打麻將的數學冷知識 (二): 如何一眼就知道胡牌了沒 <https://www.thenewslens.com/article/100657>)。

一副牌 P，若把一個對子（俗稱眼睛）拿掉後，假設此時數字最小的牌是 x，

- 若 x 的張數是 3 張以上，則拿掉 3 張 x（一刻）後，剩下牌為 Q。
- 否則拿掉 x, x+1, x+2（一順）之後，剩下的牌為 Q。（若無法拿，則 P 沒胡）

則「P 胡」若且唯若「Q 胡」。

舉個例子說明：

例 1：判斷 P = 33345678 是否胡牌？

P 中唯一能當眼睛的地方，就是 3，因此把 33 一對眼睛拿掉，變成 Q = 345678，Q 是胡牌型，所以 P 也是。

例 2：判斷 P = 55666777889 是否胡牌？

其中 55、66、77、88 都可以當眼睛，每種情況都試一次。

- 若 55 當眼，剩下 666777889；最小的 6 有三張，直接拿掉 666，剩 777889，再拿 777，得 889，所以沒胡。
- 若 66 當眼，剩下 556777889；最小的是兩張 5，但沒有辦法拿掉兩個 567 了，所以沒胡。
- 若 77 當眼，剩下 556667889；最小的是兩張 5，但沒有辦法拿掉兩個 567 了，所以沒胡。

各種能拿掉眼睛的方式都試過，都無法胡，所以 P 不是胡牌型。

i. 依照接收到的丟牌資訊，去判斷其他三家是否能胡牌。

判斷方法：判斷時先將丟的牌加入手牌，然後依序將手牌內的眼睛拿出後，將手牌內的順、刻組數計

算出，再計算吃、碰、槓陣列內的順、刻組數，最後相加看是否總數為 5 組順、刻配 1 組眼，如達成條件則表示胡牌。

EX：Program1 丟 9 筒，Program3 的手牌為 7 筒，8 筒，一萬，一萬，加入 9 筒後，將眼睛拿掉後(1 組眼)，計算出手牌有 1 組順，吃牌陣列內存(456 條、345 筒)2 組順，碰牌陣列內存(東東東)1 組刻，槓牌陣列內存(9999 萬)1 組刻，最後相加滿足胡牌條件總數為 5 組順、刻配 1 組眼，則發送胡牌指令/ask hu 給 Program3 尋問是否胡牌，如回傳胡牌指令，則傳送胡牌資訊給所有 Program，程式結束。如圖 3-13。



圖 3-13 胡牌

EX: 平台廣播/hu 3 390 370 380 111 112 指令告知四個 Program, Program3 胡了 390 這張牌, 且 Program3 程式未現出來的牌為 111、112、370、380。

ii. 自摸: 發牌給 Program 時, 判斷其是否有達成胡牌條件(5 組順、刻配 1 組眼), 如有則傳送胡牌指令給此 Program, 程式結束。

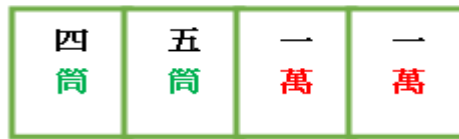
EX: Program1 摸 330, Program1 的手牌為 110 111 340 350, 加入 330 後, 計算出手牌有 1 組順、1 組眼, 吃牌陣列內存(240 250 260(456 條)、130 140 150(345 萬))2 組順, 碰牌陣列內存(180 181 182(888 萬))1 組刻, 槓牌陣列內存(190 191 192 193 (9999 萬))1 組刻, 最後相加滿足胡牌條件總數為 5 組順、刻配 1 組眼, 則發送胡牌指令/ask hu 給 Program1 尋問是否胡牌, 如回傳胡牌指令, 則傳送胡牌資訊給所有 Program, 程式結束。如圖 3-14 自摸。



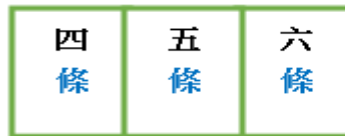
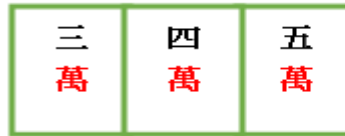
Program1 摸



Program1 手牌



Program1 吃牌陣列



Program1 碰牌陣列



Program1 槓牌陣列

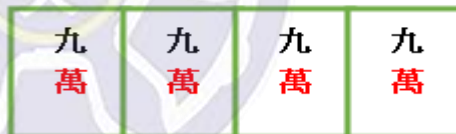


圖 3-14 自摸

iii.槓上自摸

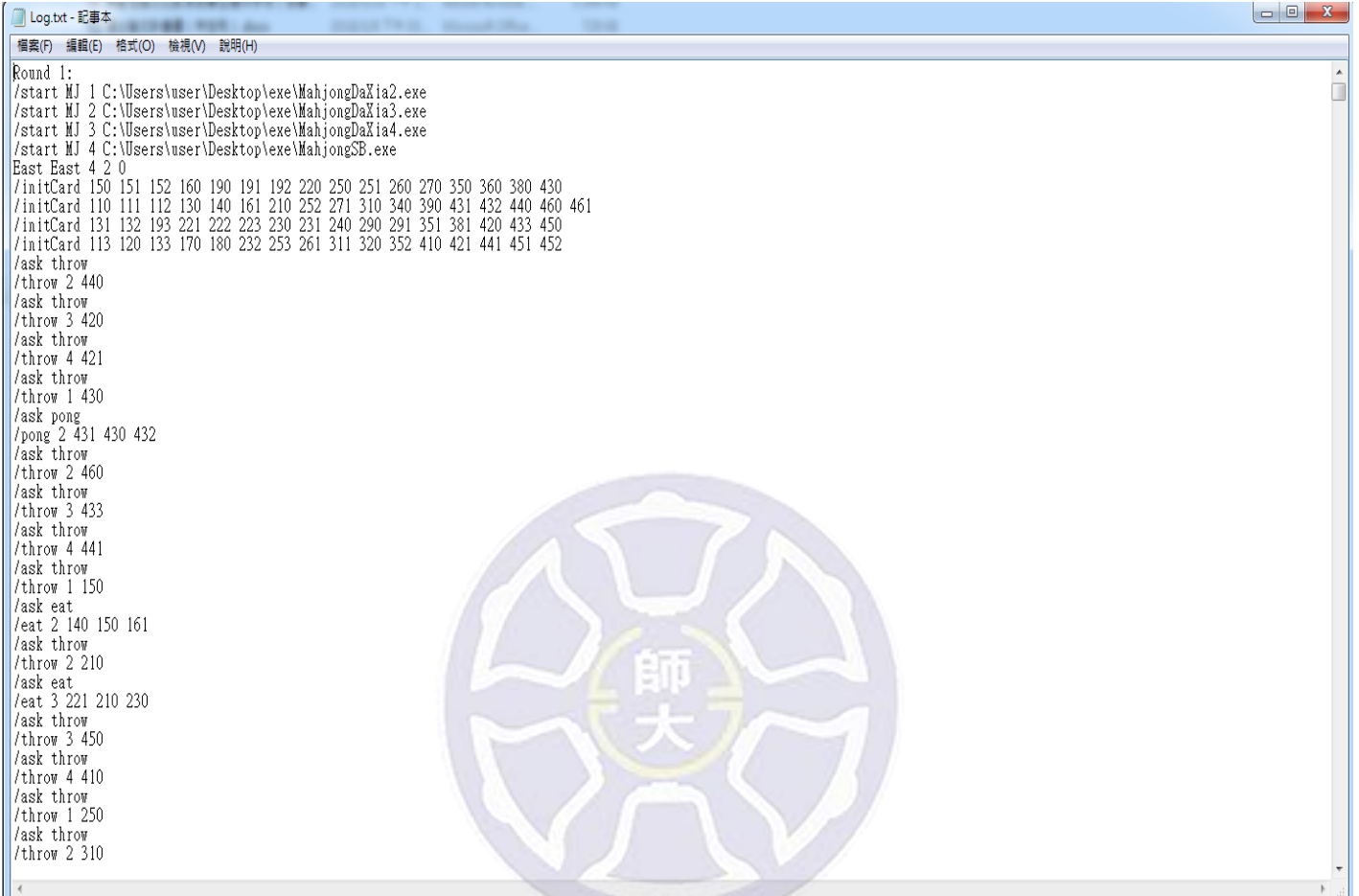
暗槓、補槓後摸的牌去判斷是否自摸，如果自摸則台數+1。

IV.一炮三響

胡牌有優先順序下家>對家>上家，所以胡牌時時會判斷誰家先胡，如果三家都胡則都算。

9. 記錄 Log 檔

會將整局的程式執行過程中的指令存入 Log 內，如圖 3-15。



```
Log.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Round 1:
/start MJ 1 C:\Users\user\Desktop\exe\MahjongDaXia2.exe
/start MJ 2 C:\Users\user\Desktop\exe\MahjongDaXia3.exe
/start MJ 3 C:\Users\user\Desktop\exe\MahjongDaXia4.exe
/start MJ 4 C:\Users\user\Desktop\exe\MahjongSB.exe
East East 4 2 0
/initCard 150 151 152 160 190 191 192 220 250 251 260 270 350 360 380 430
/initCard 110 111 112 130 140 161 210 252 271 310 340 390 431 432 440 460 461
/initCard 131 132 193 221 222 223 230 231 240 290 291 351 381 420 433 450
/initCard 113 120 133 170 180 232 253 261 311 320 352 410 421 441 451 452
/ask throw
/throw 2 440
/ask throw
/throw 3 420
/ask throw
/throw 4 421
/ask throw
/throw 1 430
/ask pong
/pong 2 431 430 432
/ask throw
/throw 2 460
/ask throw
/throw 3 433
/ask throw
/throw 4 441
/ask throw
/throw 1 150
/ask eat
/eat 2 140 150 161
/ask throw
/throw 2 210
/ask eat
/eat 3 221 210 230
/ask throw
/throw 3 450
/ask throw
/throw 4 410
/ask throw
/throw 1 250
/ask throw
/throw 2 310
```

圖 3-15 記錄 Log 檔

10. 特殊牌型計算及聽牌數計算

特殊牌型計算：

i. 計算胡牌時所有牌的張數。

EX：123456789 萬中中中發發發白白。9 張萬子 3 張紅中 3 張發財 2 張白板。

ii. 再由張數去判斷各種牌型。

EX：只有萬子跟字牌達成混一色條件，由中發白組成的兩刻一眼達成小三元條件。

iii. 最後為依照達成的牌行計算所得台數。

EX：混一色台數+4，小三元台數+4。如圖 3-16。

一 萬	二 萬	三 萬	四 萬	五 萬	六 萬	七 萬	八 萬	九 萬
中	中	中	發	發	發	白	白	

混一色、小三元

圖 3-16 特殊牌型計算

IV.全求、海底撈月、地聽判斷

全求：胡牌時計算手牌數，如果手牌數為，1 則達成全求條件，台數+2。

海底撈月：自摸時確認摸牌的指標，如果指標為最後一張牌，則達成海底撈月條件，台數+1。

地聽：正常規則下，地聽後就只能執行摸打動作，無法改變手牌，但麻將程式會根據最佳決策去改變手牌，目前平台方面無法解決這個問題，因此沒有地聽判斷。

聽牌數計算：

由於聽牌數為 1 時(獨聽)，台數會加 1，因此需計算出聽牌張數。

i.將所有牌依序加入手牌後判斷是否胡牌。

ii.如胡牌則將加入的牌存入聽牌陣列。


iii.聽牌陣列內個數為一時表示獨聽台數+1，不是獨聽台數不變。

11.得分計算

i.設定底 1000 分、台 500 分。

ii.會先將胡牌台數算出後，再依底 1000 分、一台 500 分的公式算出當局得分，再將各局得分跟總分情況寫入 Log 檔，如圖 3-17。

EX：當局胡牌為平胡，台數為 2，當局得分為 $1000+500\times 2 = 2000$ 分。



```
Score.txt - 記事本
檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)
Player4: -28000
Hu: 58 65 55 51
Mo: 15 15 13 12
Boom: 46 65 59 59
Round 300:
當局得分
Play2: +2000
Play3: -2000
目前得分:
Player1: 47500
Player2: 13000
Player3: -32500
Player4: -28000
Hu: 58 66 55 51
Mo: 15 15 13 12
Boom: 46 65 60 59
```

圖 3-17 得分 Log 檔

11. 防呆機制

i. 當 Program 回傳的指令有誤時，啟動此機制，將 Program 剔除平台代打，當輪到此 Program 動作時，就不作任何吃、碰、槓、胡動作判斷，只執行摸打動作。

判斷方法：當 Program 有執行吃、碰、槓等動作時，檢查回傳的指令是否有錯誤，有錯誤則記錄此 Program 編號後，關閉連線。當下一回合輪到此 Program 時，只作摸打動作，直到此局結束。當下局開始重新連線此 Program，此時 Program 又可以開始正常運行。

EX：當 Program2 丟 110，平台檢測到 Program3 可以吃後，發出 /ask eat 指令後，接收回來的指令卻是 /eat 120 310，指令有誤，則不執行更新手牌與發送吃牌指令給其他 Program 的動作，直接跳入 mo 牌函式，執行摸打動作，直到此局結束。如圖 3-18。

```

file:///C:/Users/user/Documents/Visual Studio 2015/Projects/RandomSample/RandomSample/bin/Debug/RandomSample.EXE
/ask throw
/throw 3 383
/ask eat
/pass
/mo 213
/ask throw
/throw 4 213
/mo 162
/ask throw
/throw 1 141
/ask pong
Get error, AI4 into dummy!
/mo 132
/ask throw
/throw 2 132
/mo 123
/ask throw
/throw 3 123
/ask eat
/pass
/mo 163
/ask throw
/throw 4 163
/mo 111
/ask throw
/throw 1 111
/mo 432
/ask throw
/throw 2 432
/ask pong

```

圖 3-18 防呆啟動，執行摸打動作

3.4 比賽局數期望值計算

由於麻將遊戲的運氣成分蠻重的，有可能在棋力相差不大的情況下，最後幾把被翻盤，因此究竟該設多少局為比賽基準，將運氣成分最小化也是一個課題。經由以下實驗可以看出來當比賽至 100 局時棋力較弱的已經平衡，分數變動不大如下圖 3-19。

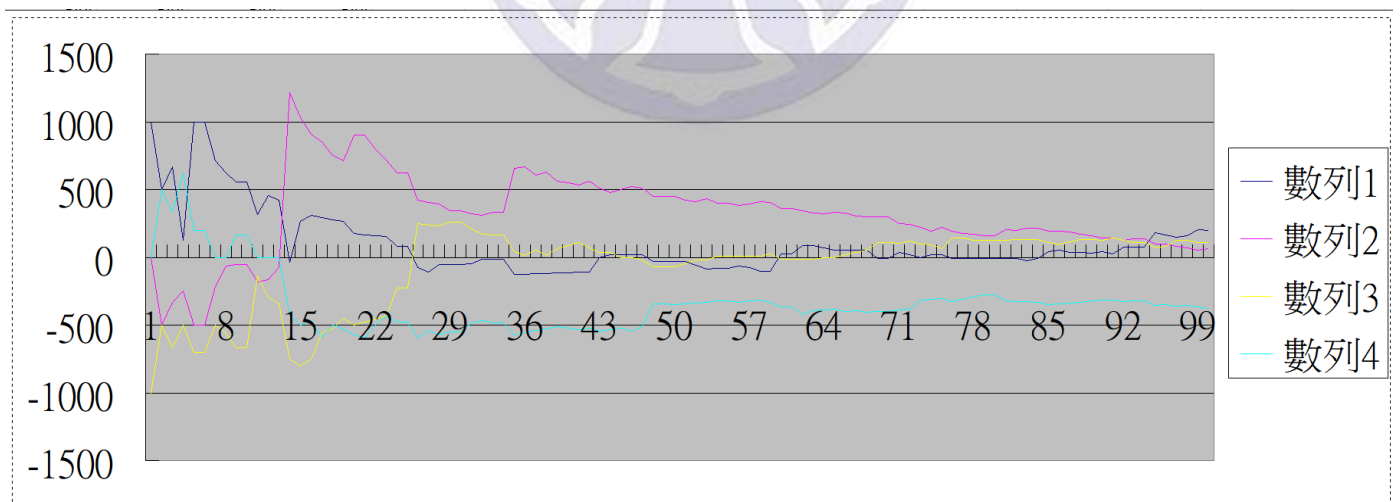


圖 3-19 100 局的程式平均得分曲線

至 200 局時棋力較強的還是有小幅變動，如下圖 3-20。

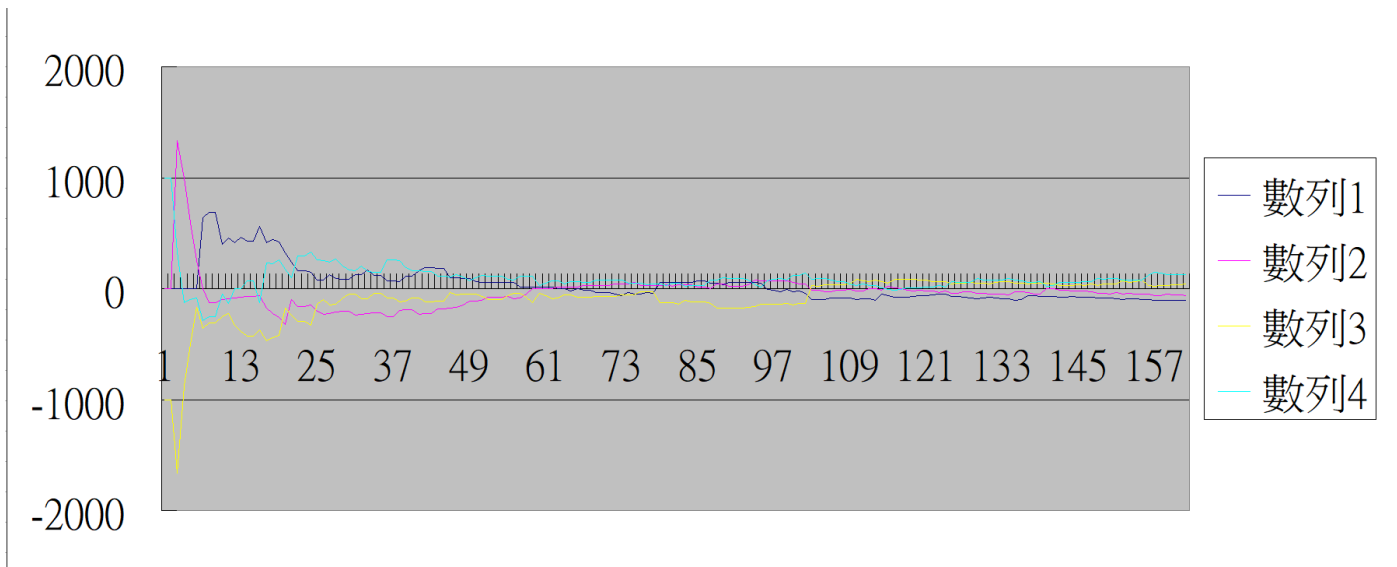


圖 3-20 200 局的程式平均得分曲線

當至 300 局時 4 隻程式都已達到平衡，分數變動基本固定如下圖 3-21。

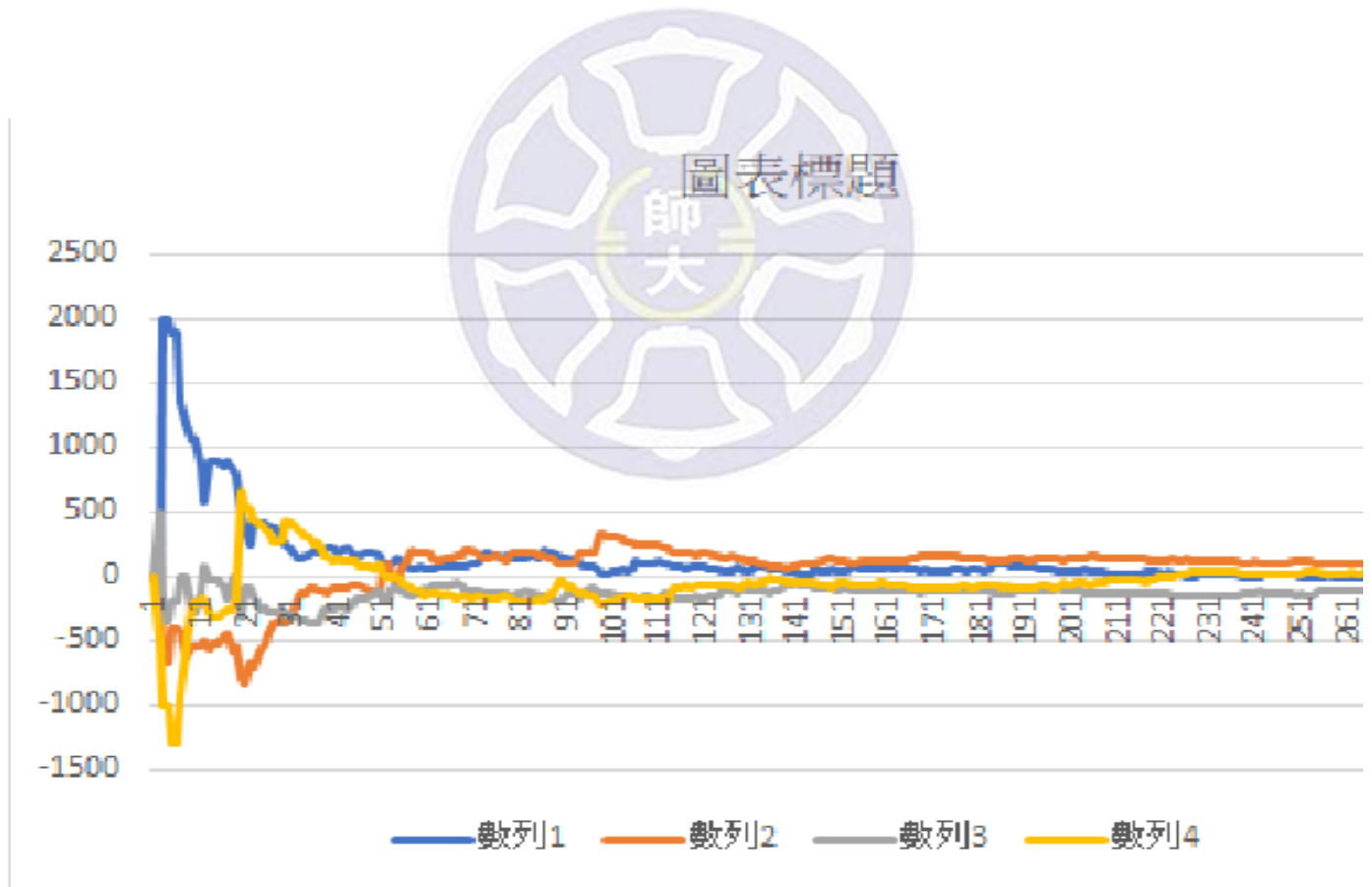


圖 3-21 300 局的程式平均得分曲線

由此可見最佳的比賽盤數應設為 250 至 300 為妥。

3.5 MahJongDaXia3 的演算法探討

由於此節要介紹 MahJongDaXia3 的演算法改動，因此要先介紹此程式原本的設計原理，此程式核心分為拆分、整合、比較。以下引用自吳俊緯「電腦麻將程式 MahJongDaXia 的設計與實作」論文中的圖片說明：

拆分：

簡單來說就是將手牌拆成各種情形後，比較其各種牌型的完成組數、未完成組數、眼睛組數，找出最佳的牌型。

完成組定義：由刻、順等 3 張或以上組合成的牌型。

未完成組：由搭子、中洞、眼等兩張組合成的牌型。

眼：由兩張相同的牌組成的牌型。

EX：33345 萬 44455 筒，可以拆成：

1.333 萬 45 萬 444 筒 55 筒，2 組完成組，2 組未完成組。

2.33 萬 345 萬 44 筒 45 筒 5 筒，1 組完成組，3 組未完成組，1 孤張。

則第一種牌型優於第二種，則選擇拆成牌型 1。

牌型物件：

MahJongDaXia3 審局用的物件結構以雙層 48 格作為已完成的組的數量儲存用，第一層存刻，第二層存順，如圖 3-22 所示。假使有三張五筒編碼為 350、351、352，就會在第一層的第 35 格加 1 的值。若有一筒、二筒、三筒，編碼為 310、320、330，則會在第二層的第 33 格加 1 的值。未完成組以雙層 48 格來儲存，存的值意義為缺的牌。假使有兩張五筒編碼為 350、351，就會在第一層的第 35 格加 1 的值，表示缺的牌為五筒。若有二筒、三筒，編碼為 320、330，則會在第二層的第 31 格與第 34 格加 1 的值。孤張以單層 48 格儲存，用來存孤張張數。如圖 3-22。

完成組表		0	1	2	3	4	5	6	7	8	9
萬	草										
萬		0	1	2	3	4	5	6	7	8	9
條											
筒											
字											

未完成組表		0	1	2	3	4	5	6	7	8	9
萬	草										
萬		0	1	2	3	4	5	6	7	8	9
條											
筒											
字											

單張表		0	1	2	3	4	5	6	7	8	9
萬											
條											
筒											
字											

圖 3-22 MahJongDaXia3 審局用的物件結構

整合：

整合是在 MahJongDaXia3 才採用的動作，其目的是使程式能更清楚的了解，在捨出一張牌後，剩下的牌型其進牌的多種可能性。當一個物件 S 進統整函式時，會與現階段最佳的牌型物件 B 做比較統整，以下列出其流程：

1. 比較兩者組數，若 S 組數大於 B，則回傳 S，反之亦然，若兩者相等進入下一個比較。如圖 3-23。

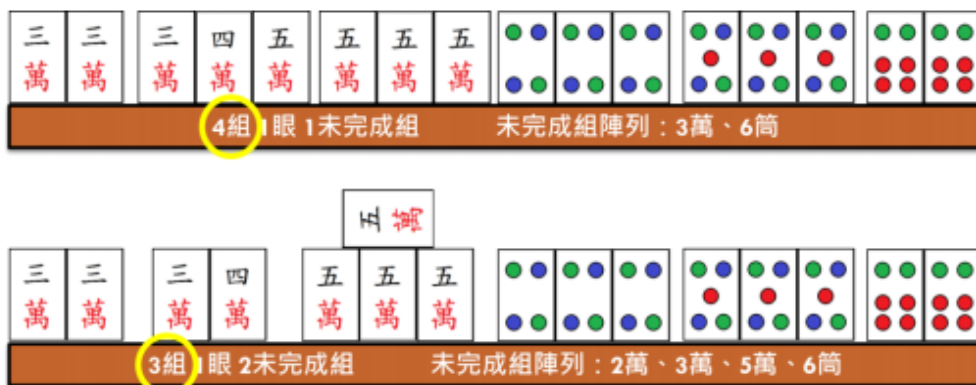


圖 3-23 選擇組數較多的拆解：4 組大於 3 組，因此選擇上面的拆解

2. 比較兩者未完成組數，若 S 組數大於 B，則回傳 S，反之亦然，若兩者相等進入下一個比較。

如圖 3-24。

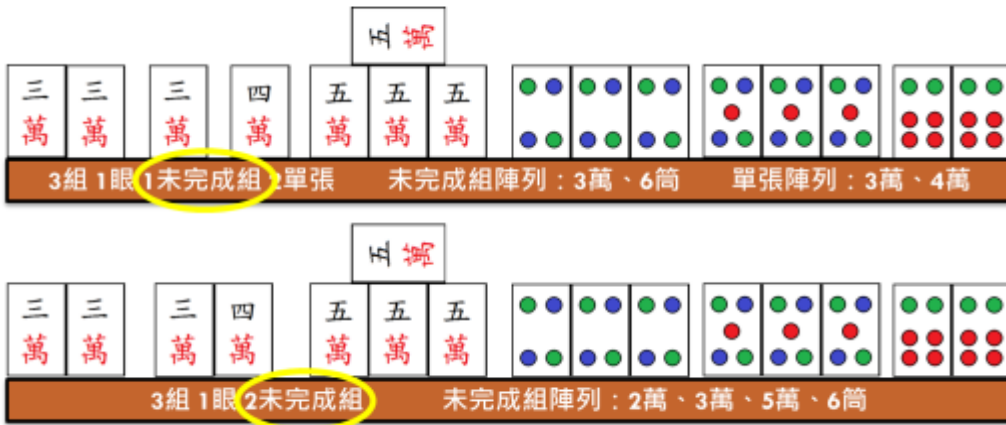


圖 3-24 選擇未完成組數較多的拆解：1 未完成組小於 2 未完成組，因此選擇下面的拆解

3. 比較兩者未完成的機率總和，以較大的作為基底，從機率和較低的物件將基底所沒有的未完成組加入其陣列。以圖 3-25 作範例，假設有一牌型二萬、三萬、三萬，我們將有兩種取法。一為取二萬、三萬為搭，等一萬、四萬吃牌，以三萬為單張。另一取法為取三萬、三萬為眼，等三萬碰牌，以二萬為單張。在牌型整合中，若一萬、四萬的取法機率和較高，則以此取法作為基底，並將等待三萬碰的機率加到基底的陣列中。

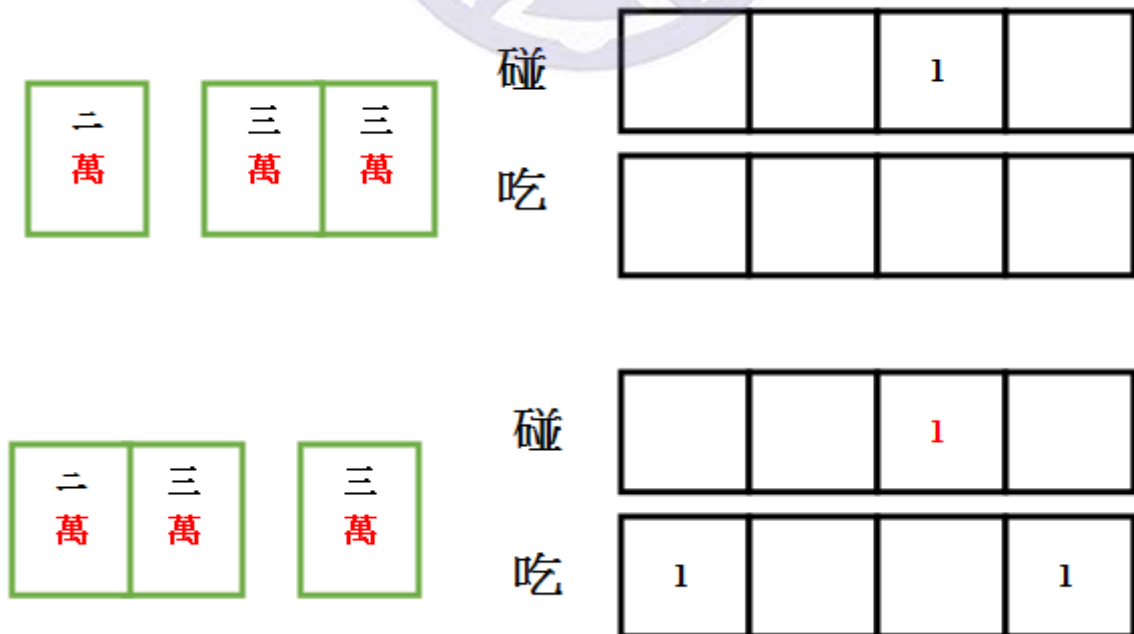


圖 3-25 機率總和範例

4. 單張部分做相同動作將單張陣列整合到基底的單張陣列中。

做完所有分支整合即可回傳牌型物件，最終得到捨去一張牌後的牌型優劣，包含最大的完成組數、等待完成的組數、機率總和等等。

比較：

在這部分是將各個丟出牌後的牌型做比較優劣的動作，也是麻將程式決策的最核心部分。由於 MahJongDaXia3 本身是基於規則導向的程式，會列出依序遵循比較的規則，也會說明兩個在設計時的關鍵決策點。

首先列出規則：

規則 1. 選擇已完成組數較多的，若相同則看規則 2。

規則 2. 選擇未完成組數較多的，若相同則看規則 3。

規則 3. 選擇未完成機率總和較高的，若相同則看規則 4。

規則 4. 檢測捨出的這張牌是否與其他牌相依，優先捨出沒相依的牌，若兩者同時相依或未相依則看規則 5。

規則 5. 檢測捨出的這張牌是否造成獨立單張，優先捨出造成獨立單張的牌，若兩者皆不會或皆會造成獨立單張則看規則 6。

規則 6. 選擇單張成組機率總和較高的，若相同則看規則 7。

規則 7. 檢測單張曝光度，優先捨出曝光度較高的單張。

由於上述規則 1-3 已在整合說明，這邊就介紹規則 4-7 為主。

規則 4：單張牌相依部分開始作為單張牌捨棄的規則。

首先檢查單張是否與其他手牌相依，在此相依的定義為與其他手牌的數值差異在 2 之內。如圖 3-26，以三萬為例，一萬、二萬、三萬、四萬、五萬皆為其相依牌，只要手牌有這五種牌其一，那三萬就是有相依的單張牌。此規則是避免將相依單張隨意的捨棄，由於拆分的做法會產生部分的單張，假設遊戲剛開始且沒人丟牌，牌型為三萬、四萬、五萬、六萬、五筒。由於取完成組會造成三萬或六萬形成單張，然而捨棄三萬或六萬與捨棄五筒的價值將不相同。在單純的單張成組機率的狀況下，三

萬或六萬的機率會低於五筒，因為單張成組機率的計算與相依範圍相同，然而手上已有四萬、五萬的情況下，會造成可進牌較少，因此機率較低，於是使用此規則進行校正。

相依的定義

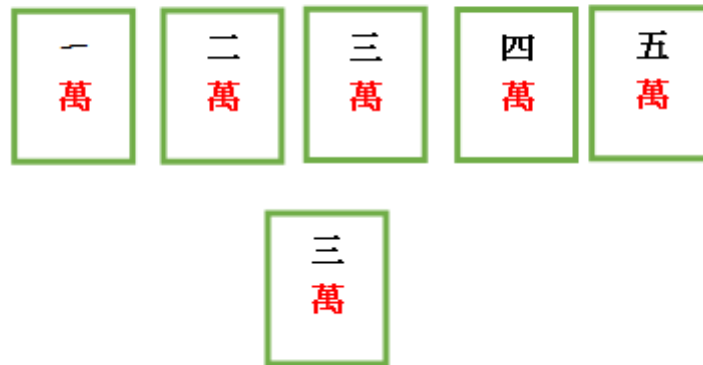


圖 3-26 規則 4 範例：相依的定義

規則 5： 檢測捨出的這張牌是否造成獨立單張，優先捨出造成獨立單張的牌。

部分牌在捨棄時會造成獨立單張，此策略目的希望先捨棄不會造成獨立單張的單張牌。以圖 3-27 中二萬、四萬、五萬、五萬、五萬為範例，並且剩餘牌型為 2 組 1 眼 2 未完成組。其中所有未完成組進牌機率最低為三萬，因此程式判斷捨二萬、四萬其中之一。對於二萬、四萬皆為互相相依，然而捨棄四萬會造成二萬成為獨立單張，而捨二萬使四萬成單張。然而四萬與五萬相依也會有發展成進三萬或六萬使五萬當眼的情況，因此就先捨棄不會造成獨立單張的牌。

不相依

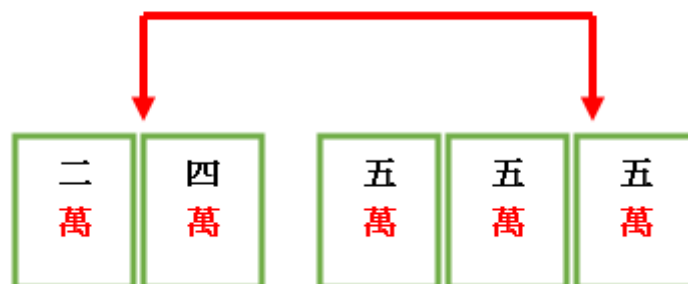


圖 3-27 規則 5 範例：選不造成獨立單張

規則 6：選擇單張成組機率總和較高的

單張成組機率規則主要是留下在相依範圍內進牌機率較高的單張牌。如圖 3-28 為例，當有一張三萬的單張牌時，單張成組機率的計算將會是(摸到一萬的機率)×(吃到二萬的機率)+(摸到二萬的機率)×(吃到一萬的機率+吃到四萬的機率)+(摸到三萬的機率)×(碰到三萬的機率)+(摸到四萬的機率)×(吃到二萬的機率+吃到五萬的機率)+(摸到五萬的機率)×(吃到四萬的機率)。在牌型沒有眼時會額外加上組成眼的機率，規則將捨掉機率較低的捨牌。

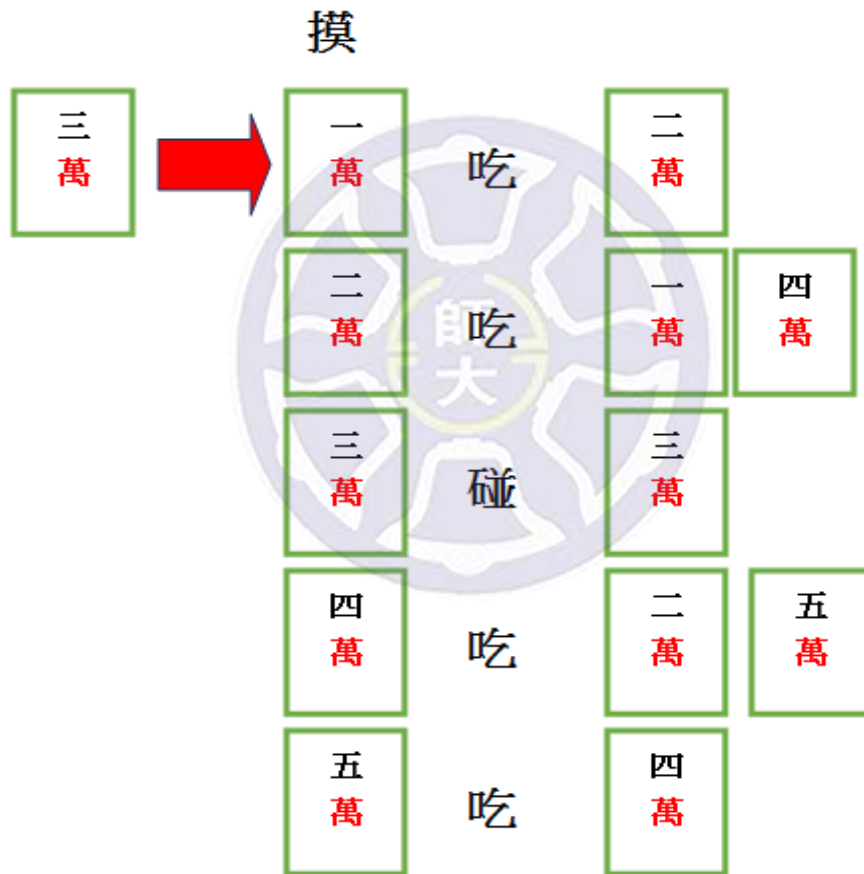


圖 3-28 規則 6 範例：單張成組機率

規則 7：檢測單張曝光度，優先捨出曝光度較高的單張。

最後一個規則偏向防守規則，是在不影響進攻的情況下訂定的。首先說明曝光度，曝光度是假設其餘玩家能得到的牌面資訊。如圖 3-29，若手牌有三張三萬，沒有人丟出三萬，則曝光度為 0，表示對於其他玩家來說，沒有看到任何一張牌，此為曝光度。在規則的選擇上，單張牌相依範圍內的曝

光度越高，則優先捨棄，原因為當牌曝光度高則表示需求較低，牌較安全，因此做此策略且不影響進攻的部分。



圖 3-29 規則 7 範例：曝光度比較

機率公式：

在程式計算的過程中會用到的機率有摸、吃、碰、胡四種，下列依序列出其算法，而其中的符號，S 表示可摸牌數量，其上限為 56，x 表示其他三家未攤開的牌數量，其上限為 48，y 表示上家未攤開的牌數量，其上限為 17，i 表示某一種牌其座落位置為未知的數量，其範圍為 0 到 4。自己取得到某一種牌的機率 P 算法如下：當 i=1，表示這種牌僅餘 1 張尚未能確定其位置，則它可能在可摸牌堆(有 S 張)或其他三家未攤開的牌(有 x 張)中。則 $P = 1/S+x$ 。當 i=2，表示這種牌僅餘 2 張尚未能確定其位置，則摸的時候有 2 倍機率摸到這種牌。故 $P = 2/S+x$ 。當 i=3，表示這種牌有 3 張尚未能確定其位置，則摸的時候有 3 倍機率摸到這種牌。故 $P = 3/S+x$ 。當 i=4，表示這種牌有 4 張尚未能確定其位置，則摸的時候有 4 倍機率摸到這種牌。故 $P = 4/S+x$ 。

3.6 MahJongDaXia3 規則改動

由於 MahJongDaXia3 並沒有防守策略，採用純規則導向為主要演算法，是一支純粹以最快胡牌為主要目標的程式，且一旦聽牌後就不做任何吃碰槓動作，持續摸打直到胡牌或放槍，因此會造成許多人類看起來不合理的決策。以下設計了一些規則希望藉此使其行為更趨於合理。

1. 下車

i.由於大約 80%的麻將遊戲都會在剩餘 32 張牌結束，也就是在第 10 巡左右，因此設定在第 10 巡時就會判斷手牌是否聽牌或者低於 1 進聽。如果大於一進聽就開始防守，跟上家或莊家打或者依同線牌原理打放槍率低的牌，以達到防禦的目的。

ii.記錄對手吃碰槓的搭子數，假如對手三搭時我方還需 2 進聽則依同線牌原理打安全張，放棄此局。

何謂同線牌?「同線牌」是指差距正好為二的一組數字，分別是：「一、四、七」、「二、五、八」及「三、六、九」，如下圖3-30。



圖3-30 同線牌

iii.預測對手需要的手牌，優先捨去分數相差較遠的對手所需的牌。

步驟1：將捨棄某張牌時的大致手牌情況列舉出來，經由模擬產生手牌，算出各種手牌的出現機率，進而預測其所需的牌。

EX：捨棄5萬時，可能的手牌情況大致為以下幾種：

1. 23萬5萬78萬，需要14萬、69萬。
2. 12萬5萬89萬，需要3萬、7萬。
3. 5567萬，不需萬子。如下圖3-31。

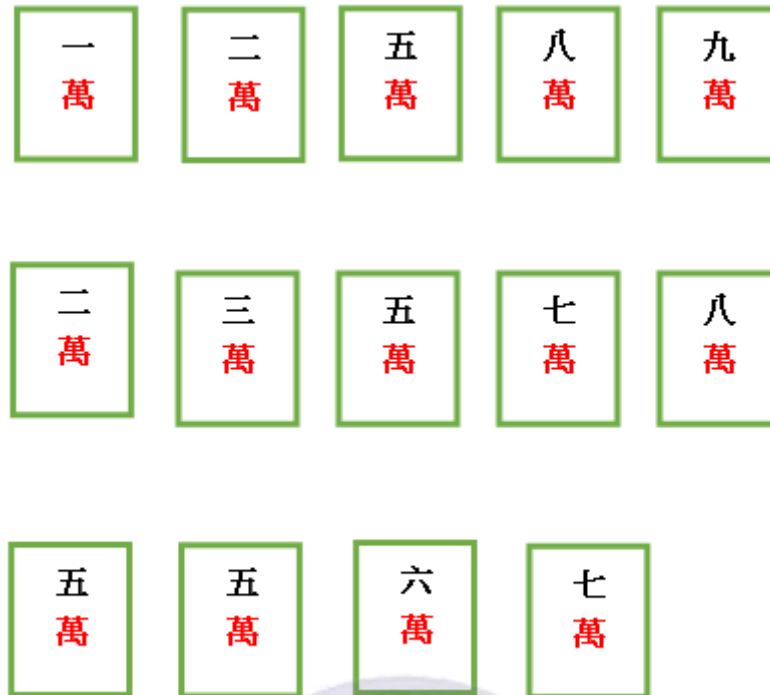


圖3-31 手牌可能情況

模擬1000次後，假設各個手牌出現次數分別為1：500次、2：300次、3：200次，其出現機率分別為50%、30%、20%。

步驟2：計算各個玩家的分數，排序出各個名次。

EX：目前排名為Program1：5000分、Program2：1000分、Program3：-4000分、Program4：-2000分。

步驟3：當達成下車條件時，以目前分數差距最遠的玩家為目標，優先捨去他想要的牌

EX：假設己方為Program3，而目前分數差距最遠的是Program1，因此優先考慮Program1想要的牌去捨給他，假設當局他捨5萬，依模擬計算出他可能要的牌由高至低為14萬、69萬、3萬、7萬，不需萬子，則將手牌內有的牌依14萬、69萬、3萬、7萬，隨機等順序去捨給他。

簡單來說就是當手上沒有同線牌可以防禦時，優先放槍給分數影響性最低者。

2.轉聽-碰

由於MahJongDaXia3程式是以最快聽牌為主要決策，對於胡牌機率部分只要不是死局(胡牌機率=0)就會搶聽，假使聽的牌機率不為0卻很低(EX：12萬聽3萬，檯面上3萬已出現3張)最後一張出現在鐵八墩或其餘玩家手裡機率很高，雖不是死局卻跟死局相似，此時，做轉聽的動作是不錯的選擇。

策略：當所聽的牌數低於某個張數時，假如將眼睛碰掉後，新轉聽的牌的張數比原本多，則將眼睛碰掉，使胡牌機率變高。如圖3-32。



圖3-32 轉聽-碰

3.轉聽-吃

策略：假設目前已經為聽牌狀態，假如吃下上家所打的牌後，能使聽牌數增加，並計算聽的牌所剩張數，如果有增加則吃。

EX：11萬(眼)57789萬(此時只聽6萬)，上家打8萬，吃後聽69萬，聽牌數增加且69萬張數比6萬多時，則吃。如圖3-33。

此時聽六萬

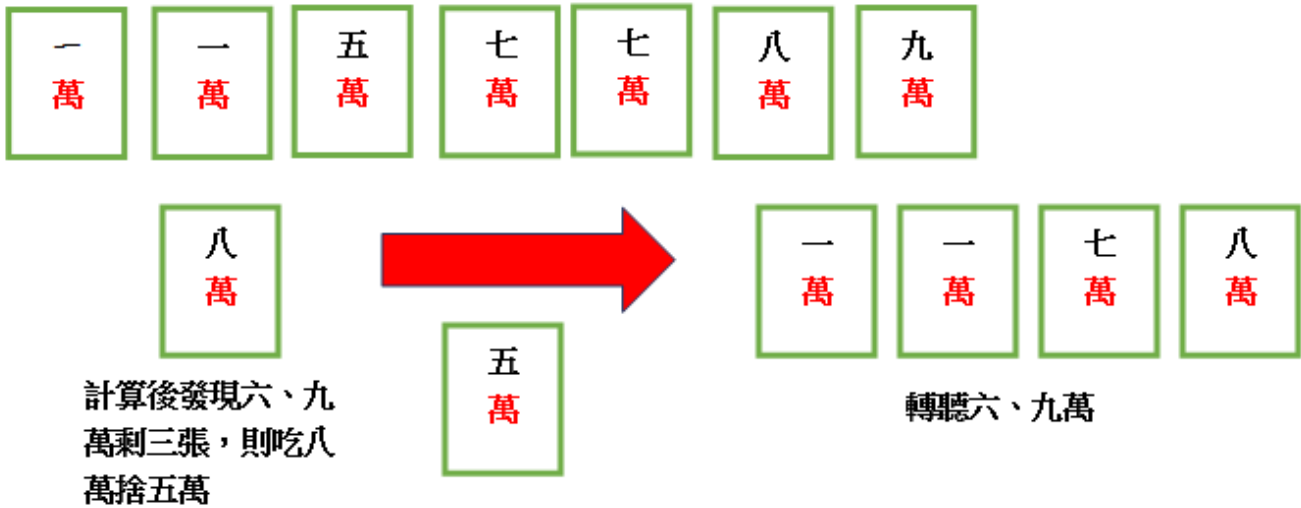


圖3-33 轉聽-吃

4. 搭子保留

原本MahJongDaXia3是會保留進牌機率最高的搭子，但現在假設我們將目標鄰邊的牌抓的越多，則對手將目標視為孤張的機率就越高，所以目前將鄰邊的牌狀況也考慮進去，再做加權總合去決定捨去哪些搭子。

EX：假設手牌89萬，11233萬，此時進7萬的機率較高(4張)進2萬機率較低(3張)，但由於2萬鄰邊牌較多(4張)，此時做加權 $3+4=7$ 大於 $4+2=6$ ，則保留13萬這搭捨去89萬。如圖3-34。



圖3-34 搭子保留

5.眼睛不固定

目前觀察到程式一開始會將眼睛固定後就不對其做動作了，造成進有效牌機率降低。

想法：假設目前是一進聽且未完成組中有兩個單一進牌組(EX：79萬89條11萬)時，如果將眼睛碰掉後，

有效進牌張數大於原本，則將眼睛碰掉，增加進牌機率。如圖3-35。

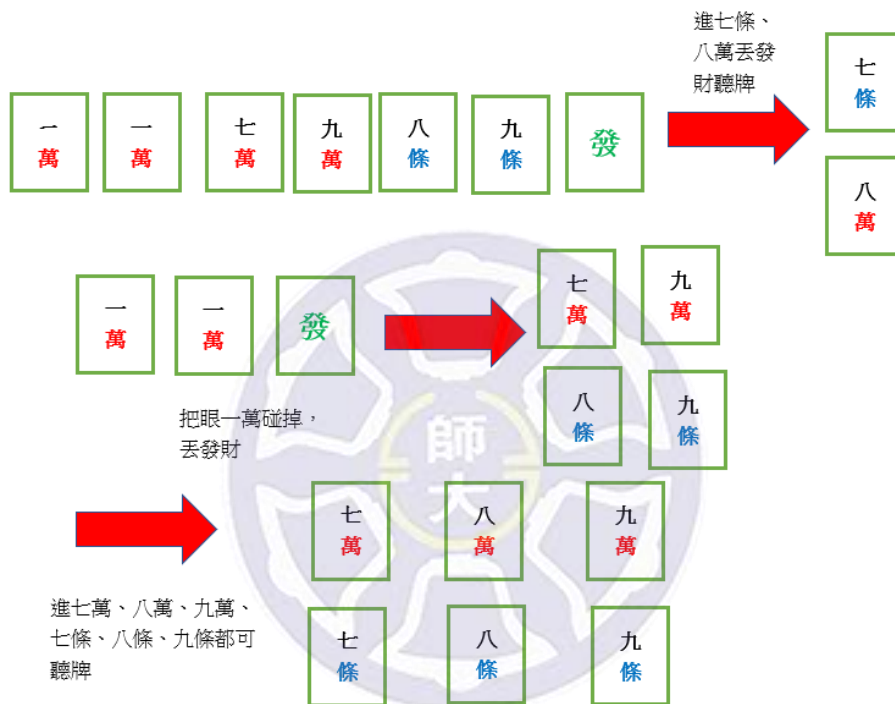


圖3-35 眼睛不固定

第四章 實驗與成果

程式開發環境：圖4-1為MahJongDaXia3的開發環境，圖4-2為MJTalk的開發環境。兩者的開發軟體都是visual studio 2015。

品項	內容
作業系統	Windows7企業版64bit(6.1,Build7601) Service Pack1
處理器	Intel(R)Core(TM)i5-2400CPU @ 3.10GHz(4CPUs),~3.1GHz
記憶體	8192MB RAM
開發環境	Microsoft Visual Studio Community 2015 版本14.0.25123.00 Update 2 Microsoft .NET Framework 版本4.6.01055

圖4-1 MahJongDaXia3的開發環境

品項	內容
作業系統	Windows 10家用版64bit
處理器	Intel(R)Core(TM)i7-7600HQ CPU @ 2.60GHz 2.60GHz
記憶體	16.0 GB
開發環境	Microsoft Visual Studio Community 2015 版本14.0.25123.00 Update 2 Microsoft .NET Framework 版本4.6.01055

圖4-2 MJTalk的開發環境

目前實驗成果：

目前 MJTalk 已能完整測試，並將對局情形顯示出來，將對局內容寫入 log 檔內以便後續解盤。

執行情況如圖 4-3、4-4。

```

file:///C:/Users/user/Documents/Visual Studio 2015/Projects/RandomSample/RandomSample/bin/Debug/Randc
/exit 2500 0 -2500 0
剩餘局數:1
目前得分:
Player1:47500
Player2:11000
Player3:-30500
Player4:-28000
/start MJ 1 C:\Users\user\Desktop\exe\MahjongDaXia3.exe
/start MJ 2 C:\Users\user\Desktop\exe\MahjongSB.exe
/start MJ 3 C:\Users\user\Desktop\exe\MahjongDaXia3.exe
/start MJ 4 C:\Users\user\Desktop\exe\MahjongSB.exe
/initGame South North 1 2 0
/initCard 113 171 173 210 213 232 282 370 391 393 413 442 453 460 463 472
/initCard 112 123 132 161 162 193 211 231 241 260 280 293 320 321 341 342 473
/initCard 110 163 170 181 233 251 252 261 262 271 291 322 362 420 431 471
/initCard 111 133 140 153 190 223 230 242 250 292 312 352 363 411 440 450
/ask throw
/throw 2 473
/mo 273
/ask throw
/throw 3 471
/mo 192
/ask throw
/throw 4 411
/mo 270

```

圖 4-3 MJTalk 執行情況

```

file:///C:/Users/user/Documents/Visual Studio 2015/Projects/RandomSample/RandomSample
/throw 1 323
/ask pong
/pong 2 320 323 321
/ask throw
/throw 2 211
/ask pong
/pass
/mo 272
/ask throw
/throw 3 271
胡:2
放槍:3
莊家:2
連莊數:0
牌型: 獨聽一台;
/hu 2 271 112 123 132 161 162 221 231 241 260 280 341 342 343
pong:320 323 321
p1: 210 213 220 232 270 282 370 382 460 463
pong:391 392 393
pong:171 170 173
p3: 143 163 172 181 233 251 252 261 262 272 273 322 330 340 360 362
p4: 111 133 140 153 190 192 212 223 230 242 250 312 313 351 352 363
/exit 0 2000 -2000 0
剩餘局數:0
目前得分:
Player1:47500
Player2:13000
Player3:-32500
Player4:-28000

```

圖 4-4 MJTalk 執行情況

至於 MahJongDaXia3 的演算法探討，經由 2018 年 7 月份的 ICGA 比賽獲得銅牌，只贏過交大的一支 RunCat 程式，結果不盡理想。但經由比賽數據可知放槍率在前期有明顯的降低。圖 4-5 為 ICGA

2018 銅牌(正面)、圖 4-6 為 ICGA 2018 銅牌(反面)。



圖 4-5 ICGA 2018 銅牌(正面)



圖 4-6 ICGA 2018 銅牌(反面)

表 4-1 為 160 場比賽數據。



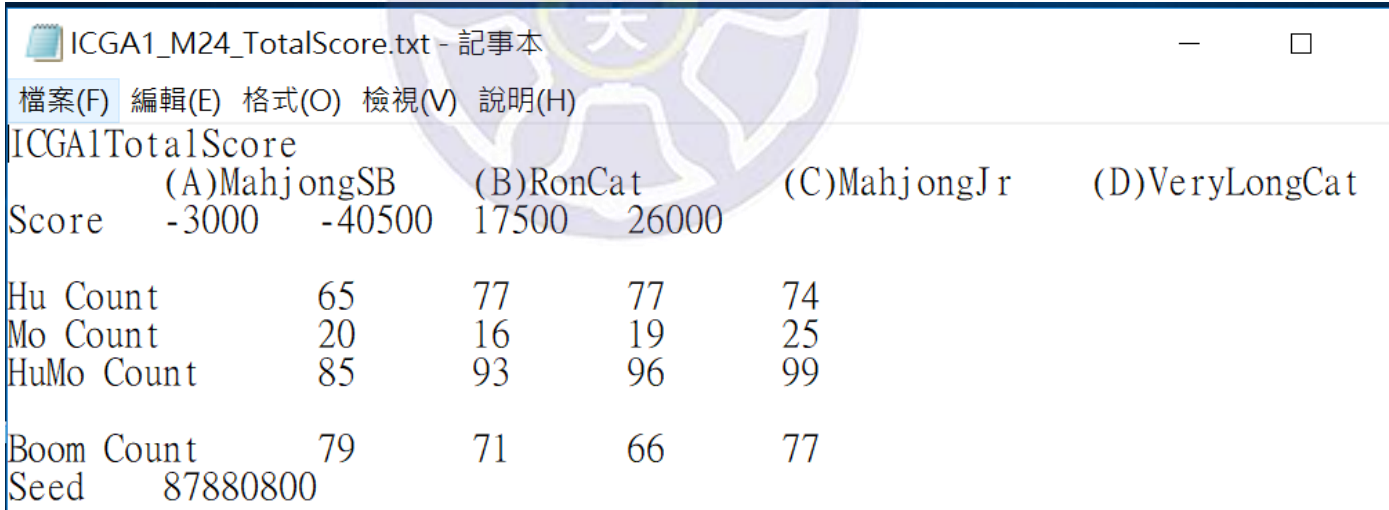
ICGA1_M10_TotalScore.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

ICGA1TotalScore	(A)MahjongSB	(B)RonCat	(C)MahjongJr	(D)VeryLongCat
Score	41500	-39500	12500	-14500
Hu Count	29	34	34	28
Mo Count	9	4	10	8
HuMo Count	38	38	44	36
Boom Count	24	30	35	36
Seed	87880800			

表 4-1 160 場比賽數據

表 4-2 為最終比賽數據。



ICGA1_M24_TotalScore.txt - 記事本

檔案(F) 編輯(E) 格式(O) 檢視(V) 說明(H)

ICGA1TotalScore	(A)MahjongSB	(B)RonCat	(C)MahjongJr	(D)VeryLongCat
Score	-3000	-40500	17500	26000
Hu Count	65	77	77	74
Mo Count	20	16	19	25
HuMo Count	85	93	96	99
Boom Count	79	71	66	77
Seed	87880800			

表 4-2 最終比賽數據

第五章 結論與展望

MJTalk 使用時要先確保麻將程式能執行，由於麻將程式開發環境的因素，目前測試時都以在 Windows7 底下為主。

MJTalk 能有效的測試各個麻將程式的運作情形，雖然比賽時仍然使用交大提供的平台，但至少在自行研發麻將程式的測試過程中能使用屬於我們師大自己的測試工具，不用再仰賴他人的工具，以免每次想測試時都得麻煩對方重啟 server，浪費許多不必要的時間。

雖然目前沒有圖型化介面，但該有的基本功能都已齊全，不影響測試跟解盤。日後有時間也可以自行研發介面後再將 MJTalk 整合進去就行了，這樣一來我們師大也有一個屬於自己不輸給交大提供的平台了。至於功能方面可以後續加入門風刻、搶槓、地聽、地胡等功能。目前跟交大平台不同的功能有防呆功能，他們出問題就被踢掉，而我們的平台只踢除當局，後面的遊戲還是可以繼續。還有連莊台計算，交大是莊家輪流當，所以沒有連莊的設定。

至於 MahJongDaXia3 的研發，以規則導向來說，進攻方面已經很厲害了，不輸使用深度搜尋配合蒙地卡羅法的交大 VeryLongCat，所以只能在防禦面下苦功，如何使防禦的改善所得分數優於胡牌率下降所失分數是值得好好研究的。如果還想提升胡牌率，可以嘗試跟交大一樣，也使用蒙地卡羅法或者使用深度學習去訓練模組，具體如何實現就需要未來再去努力了。

附錄

麻將台數列表介紹(紅色為平台已有功能)

台類名稱	台數	說明
莊家連 N 拉 N	2N+1 台	做莊者胡牌多計 1 台，但放槍時也要多賠 1 台。每連莊一次再增加 2 台
自摸	1 台	自己摸到胡牌的牌。
門清	1 台	胡牌前沒有吃、碰、明槓，但可暗槓。
門清自摸	3 台	胡牌時為門清的情況下自摸。(合稱門清一摸三)
三元台	1 台	中、發、白任何一組刻子或槓子。
圈風台	1 台	字牌東、南、西、北，任何一組刻子或槓子，附和圈風即有 1 台。 例：該場為東風圈，若持有東風刻(槓)即有 1 台。
門風台	1 台	字牌東、南、西、北，任何一組刻子或槓子，附和門風即有 1 台。 例：該場為北家，若持有北風刻(槓)即有 1 台。
正花	1 台	春、夏、秋、冬(四季)，梅、蘭、菊、竹(四君子)，拿到正花即有 1 台，東家為一花(春、梅)，南家為二花(夏、蘭)，西家為三花(秋、菊)，北家為四花(冬、竹)。
獨聽	1 台	有三種。1.邊張：胡的牌為邊張牌。例：持有一萬、二萬，胡三萬。2.嵌張：胡的牌為中洞牌。例：持有四筒、六筒，胡五筒。3.單騎：單吊將牌。例：持有一張九索、聽九索。
搶槓	1 台	別人加槓的牌，正是自己能胡的牌，則可搶槓胡。
槓上開花	1 台	暗槓、加槓、補花後所摸進的牌正是自己能胡牌。可加計自摸。
河底撈魚	1 台	胡他家打出的河底牌。

海底撈月	1 台	摸進海底牌後剛好胡牌。可加計自摸。
花槓	2 台	拿齊四季或四君子即有兩台。不計正花。
全求人	2 台	胡牌者手牌皆已吃、碰，槓只剩 1 張手牌，胡牌時由其他家放槍。
平胡	2 台	胡牌者，手牌要符和：「無字、無花、無刻、先有將、聽兩面、非自摸」的規定（可自行約定是否需門清）
三暗刻	2 台	胡牌時，手牌中有 3 副刻子（包括暗槓）。
對對胡	4 台	胡牌時，除了將牌外全為刻子（或槓）。
混一色	4 台	由一種花色的序數牌及字牌組成的牌型。
小三元	4 台	胡牌時有三元牌的 2 副刻子及將牌。不計三元台
四暗刻	5 台	胡牌時，手牌中有 4 副刻子（包括暗槓）。
五暗刻	8 台	胡牌時，手牌中有 5 副刻子（包括暗槓）。五暗刻時必定與對對胡複合，可另計。
清一色	8 台	由一種花色的序數牌組成的牌型。
小四喜	8 台	胡牌時有風牌的 3 副刻子及將牌。不計風台。
大三元	8 台	胡牌時有三元牌 3 副刻子。不計三元台。
七搶一	8 台	持 7 張花牌可以搶別家 1 張花牌湊滿花牌而胡牌。須補花後表明花胡，被搶花者視同放槍，莊連拉照算。未補牌而叫胡，則花胡不算。不計正花、花槓
八仙過海	8 台	取得全部 8 張花牌而胡牌。須補花後表明花胡，視同自摸，莊連拉照算。未補牌而叫胡，則花胡不算。不計正花、花槓。
地聽	8 台 (或 4 台)	四家在無發生吃、碰、明槓的情況下，第一巡打出第一張牌即聽。不計門清
字一色	16 台 (或 8 台)	由字牌的刻子（槓）所組成的牌型。必定與對對胡複合。會與大四喜、小四喜、大三元或小三元複合，可加計。
大四喜	16 台	胡牌時有 4 副風刻組成的牌。不計風台。
地胡	16 台	閒家專屬牌型，配牌後，第一巡內，無發生吃、碰、槓之情況，閒家於此巡內第一次摸牌即自摸胡，不計自摸、門清、不求人。其餘另計。
天胡	16 台	莊家專屬台型，其於起手配牌補花後 17 張完美自摸。不可暗槓。不計門清、自摸、不求人、獨聽、槓上開花，其餘另計
人胡	16 台	閒家專屬台型，配牌後，第一巡內未摸牌前，無發生吃、碰、槓之情況，胡他人牌，不計門清，其餘另計。

參考文獻

- [1] 林正宏，麻將平台與人工智慧程式溝通協定。2011，國立交通大學研究所碩士論文。
- [2] 許綸洲，麻將人工智慧設計。2013，國立高雄應用科技大學研究所碩士論文。
- [3] 陳新颺，電腦麻將程式 ThousandWind 的設計與實作，2013，國立臺灣師範大學資工所碩士論文。
- [4] 莊立楷，麻將人工智慧之研究。2014，國立交通大學研究所碩士論文。
- [5] 沈庭璋，電腦麻將程式 TaKe 的設計與實作，2014，國立臺灣師範大學資工所碩士論文。
- [6] 吳俊緯，電腦麻將程式 MahJongDaXia 的設計與實作。2015，國立台灣師範大學資工所碩士論文。
- [7] 維基百科-麻將規則介紹
(<https://zh.wikipedia.org/wiki/%E5%8F%B0%E7%81%A3%E9%BA%BB%E5%B0%87>)。

