



新北市板橋區埔墘國小資優班獨立研究論文

研究題目

骰子吹牛的必勝祕訣

作者：

李泓杰。新北市埔墘國民小學。六年三班

林楷恩。新北市埔墘國民小學。六年四班

李宙恩。新北市埔墘國民小學。六年五班

張煒杰。新北市埔墘國民小學。六年六班

指導老師：
黃靜蘋 老師
黃舜煒 老師

中華民國115年5月

謝誌

能完成這個研究除了靠我們自己的努力之外，也有很大一部分是靠老師們的協助，所以我們要在這此謝謝幫過我們而且提供專業建議的黃舜煒老師和李典霖老師，以及陪我們練習的黃靜蘋老師，還有支持我們做這個研究的家長，多虧有他們的幫助，我們才能順利完成這個研究。

摘要

骰子吹牛是結合機率與心理博弈的不完全資訊賽局。由於遊戲狀態數龐大，難以透過樹狀圖分析，因此改採數學期望值與統計對弈結果來研究策略。針對人類易受情緒干擾而無法理性判斷的弱點，我們利用生成式AI設計不同等級的電腦策略進行驗證。實驗結果顯示，透過機率模型與手牌動態切換策略，能顯著提升勝率，並證明最佳對局策略需兼具機率準確性與不可預測的隨機性。

壹、前言

一、研究動機

因為我們的老師在課堂上向我們介紹這款桌遊時，我們覺得非常好玩，也想要找出能夠百戰百勝、百分百擊敗對手的策略。後來上網去查，找到骰子吹牛跟數學當中的可能性有關，所以我們發現：在試玩時有些同學百戰百勝、也有人百戰百敗，但還是不清楚這個遊戲跟數學的統計與可能性有甚麼關聯，也不知道甚麼時候比較適合抓；甚麼時候比較不適合抓。我們希望能透過這個研究提升我們在這個遊戲中贏的可能性。所以，我們想要研究骰子吹牛的技巧與方法找到有效的策略並提升勝率。

二、研究目的

本研究的主要目的是研究吹牛骰子遊戲，透過數學方法以及對局結果找尋最佳的對局策略。

（一）吹牛遊戲中的數學分析

1. 算出5顆骰子中的所有組合
2. 將5顆骰子中的組合依照組成進行分類
 - (1) 5顆相同骰子
 - (2) 4顆相同骰子
 - (3) 3顆相同另2顆相同
 - (4) 3顆相同另2顆不同
 - (5) 2組2顆相同另外1顆不同
 - (6) 1組2顆相同，其他3顆不同
 - (7) 5顆骰子皆不同
3. 計算出各種不同類別中，5顆骰子出現組合的可能性
 - (1) 出現5顆相同骰子的可能性
 - (2) 出現4顆骰子相同的可能性

- (3) 出現3顆骰子相同另2顆也相同的可能性
- (4) 出現3顆骰子相同，但另外2顆部相同的可能性
- (5) 出現2組2顆相同的骰子，另1顆相異的可能性
- (6) 出現2顆骰子相同，另外3顆骰子相異的可能性
- (7) 出現5顆骰子皆相不相同的可能性

4. 期望值理論

- (1) 當一作為通用骰時，五顆骰子中期望會出現的顆數是多少
- (2) 當一作為通用骰時，一組骰子中期望會出現的顆數是多少
- (3) 當一不做為通用骰時，五顆骰子中期望會出現的顆數是多少
- (4) 當一不做為通用骰時，一組骰子中期望會出現的顆數是多少

(二) 吹牛電腦策略設計以及對局結果分析

1. 設計AI策略

- (1) Lv0 : 利用遊戲規則設計 策略_Lv0隨機
- (2) Lv1 : 利用基礎觀念及期望值設計 策略_Lv1機率模型
- (3) Lv2 : 利用與L1的對弈結果設計 策略_Lv2機率模型

2. 對弈結果分析

- (1) AI 對 AI 結果分析
- (2) AI對人 結果分析

哪一種骰子組合勝率較高

先後順序對於勝負的影響

三、文獻探討

表一、文獻重點及應用

文獻名稱	文獻摘要	本研究啟發
驚爆骰子樂之吹牛大王	計算在不同顆數的骰子下，出現數字1的可能性是多少，再由不同遊玩人數中，假設1出現的可能性及顆數，判斷對手喊的數字顆數是否為真，決定自己要繼續喊或是準備抓牌。	除了計算通用骰出現的可能性之外，另外探討其他相同出現顆數的可能性。
「欲罷不能」桌遊機率之探討與延伸	將4顆骰子擲出的組合加以分類，4顆相同、3顆相同、2顆相同.....等，並從中探討裡面骰子兩兩一組時，不同骰子總和組合的可能性。	將吹牛骰子組合進行分類，並探討哪一種組合出現的可能性最高。
實現強化學習於多人吹牛骰子遊戲之探討	將5顆骰子的組合進行分類，並計算可能性，運用AI結合賽局理論，模擬在兩人賽局遊戲中，在不同經驗法則、先後手的情況下，經由互相叫牌、抓牌的過程中，判斷在怎麼喊最容易讓對手抓失敗、什麼時候抓可以提高成功的可能性。	1.使用AI進行模擬對戰，並制定加入人性思考時的策略。 2.探討不同牌型的勝率。
看誰最會吹-破解吹牛的秘密	使用窮舉法，將所有骰子的組合在不考慮排序的情況下逐一列出共252種組合，比較簡單隨機抽樣方式及選擇最多組合的總和進行抽樣分析，得出兩者結果相近，只有在極端值有差異。	除了用抽樣之外，進一步延伸以期望值探討在遊戲中出現幾顆骰子的可能性較高。

貳、研究設備與器材

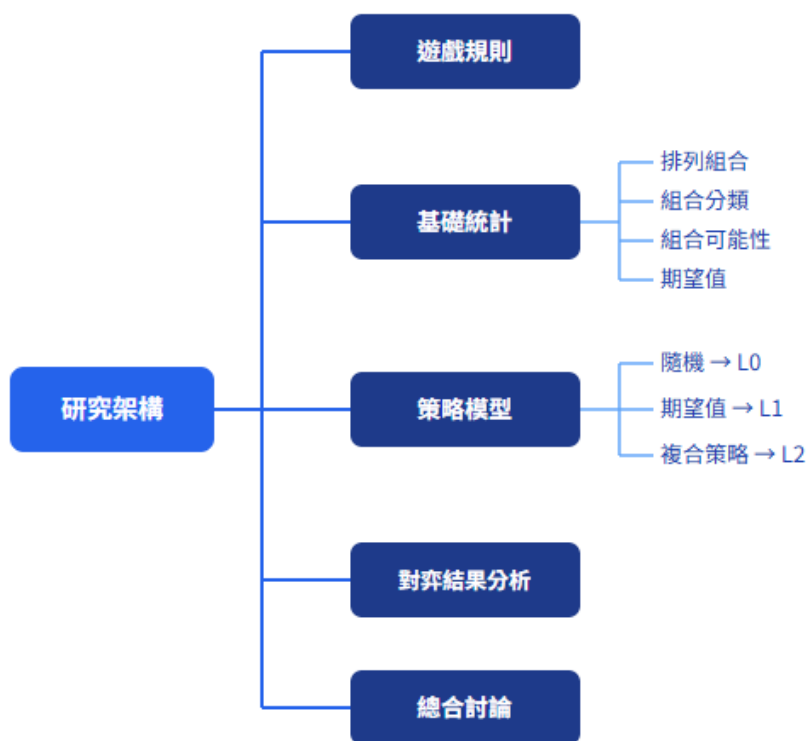
- 骰子、骰盅：進行真人對弈實驗時使用。
- 紀錄表、筆：記錄每局叫牌順序、抓牌時機與勝負結果。
- 電腦：撰寫程式、執行模擬對弈與統計分析。
- 生成式 AI（Gemini-3-Pro）：
 1. 生成圖表將理論可視化
 2. 生成程式加速研究進程
 3. 將策略想法轉換為電腦程式，並協助調整與測試不同策略(下方為程式介面)。



圖一、程式介面

參、研究過程與方法

一、研究架構



圖二、研究架構圖

二、研究方法

(一) 遊戲規則

吹牛（或稱大話骰）為一種多人骰子遊戲，每人擁有5顆骰子。遊戲具備「不完全資訊」特性，玩家僅能看見自己的骰子，需透過叫牌與推理猜測場上總數。

叫牌與抓牌規則：

- 所有玩家同時搖骰，並只看自己的骰子。
- 起始玩家喊出「X個Y」（預測場上至少有X顆Y點數骰子）。
- 下家可選擇「加注」（喊出更大的數量或點數）或「抓」（質疑上家吹牛）。
- 若喊出的數量X相同，則點數Y必須大於上家；若數量X大於上家，則點數Y無限制。
- 一旦有人選擇「抓」，所有玩家就同時打開骰盅，檢查全場是否達到被質疑的叫牌。

若實際數量 小於 被抓所喊的數量，則該叫牌玩家「吹牛失敗」，被抓的人輸。

若實際數量 大於或等於 被抓所喊的數量，則抓牌的人判斷錯誤，抓的人輸。

通用骰 (Wild Card) 機制：骰子點數「1」通常被視為通用骰，可代表任何點數。但在遊戲過程中，若有玩家喊過「X個1」，則「1」失去通用功能，僅能代表1點。



圖三、玩家先手喊 10個6 (場上只有3個不符合) 電腦抓成功獲勝



圖四、電腦先手喊 2個3 (場上有7個3 符合) 玩家抓失敗 電腦獲勝

(二)可能性與期望值

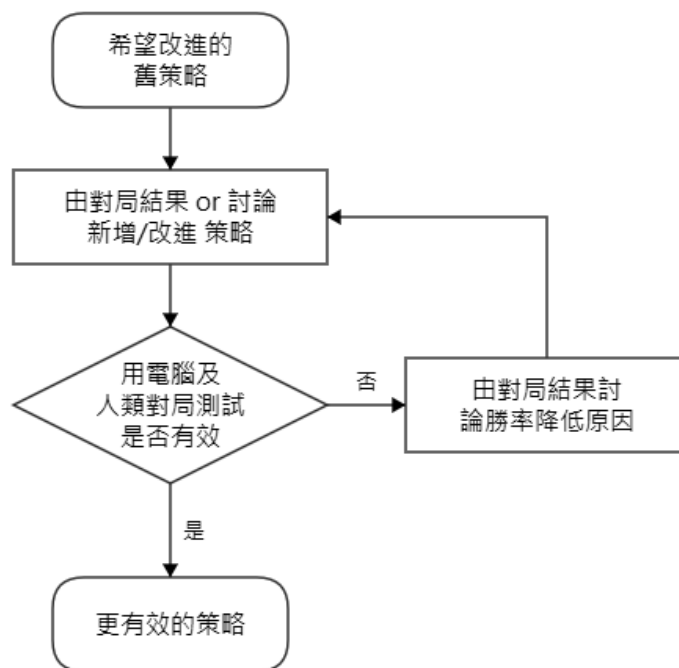
在吹牛遊戲中，知道出現組合的可能性有多大會決定如何叫牌與抓牌的時機，因此，我們將會計算出每組組合出現的機率以及可能出現的期望顆數。

(三)生成式AI模擬

為了比較明顯感受到策略對於勝率帶來的影響，我們利用程式化做成AI，並將計算出的可能性與期望值輸入AI及制定策略讓他們之間互相對弈，最後藉由分析數據及勝率，直觀的看到策略對對弈結果帶來的變化。為了分析不同的策略和思考方式，我們將AI分成3個等級：

- 1.AI等級Lv0:AI僅知道規則(完全隨機)，但如何叫牌及何時抓牌則以隨機方式進行，難以判斷。
- 2.AI等級Lv1:AI會參照自己的手牌，並根據機率與期望值，推測對手叫喊的數字是否為真，並決定應該如何叫牌或何時抓牌。
3. AI等級Lv2:基於Lv1關於機率與期望值的策略對弈結果討論設計，找出僅用期望值會出現的問題，修正常見對局失誤，增加應對更多不同狀況的能力。

這是我們實驗的網址：<https://poe.com/preview/RMmRjOMqEWFFjvTPQEKL>



圖五、調整AI流程圖

(四)策略分析

根據AI模擬遊戲的過程中，我們會根據模擬結果，將所獲得的資料進行分析，依據我AI對弈的結果，修正我們制定的策略，最後，討論出適合的遊戲策略，提高獲勝的可能性。

肆、研究結果

一、吹牛遊戲中的數學分析

(一) 吹牛遊戲一組骰盅共有5顆骰子，每顆骰子共有6個面，因此，所有組合有 6^5 種，共有7776種組合。

(二) 因吹牛遊戲主要是看骰子的顆數，因此，我們將吹牛骰子的進行分組，並計算出每組的組合共有幾組。

- 1.五顆骰子相同:從六面骰子中選擇1個數字，每個骰子都只能是相同的數字。
- 2.四顆骰子相同:選一個點數當作「四顆」的(6種)，選另一顆點數當作「剩下的一個」(5種)，並決定那一個點數出現在哪一個位置(5個位置)。
- 3.三顆相同另兩顆相同: 選一個點數當三顆的(6種)，選另一個點數當兩個的(5種)，再從5個位置選三個放三同，另外兩個放兩同(10種)。
- 4.三顆相同另兩顆不同:選一個點數當三個的(6種)，從剩餘五個點數選兩個為單顆，然後，五個位置中決定三同的位置，剩餘兩個位置為單顆骰子並且可以換位置(20種)。
- 5.兩組兩顆相同另外一顆不同:從六個點數選兩個當作兩對(15種)，從剩餘四個點數選一個當單點(4種)，而位置則是先五個位置選兩個擺放第一組兩同的(10種)，剩下3個位置擺放第二組兩同的(3種)。
- 6.一組兩顆相同，其他三顆不同:先從六個數字選其中一個做為一對的(6種)，剩下五個數字分別為其他三個數字(10種)，位置為五個位置先放兩個相同的，剩下三個位置放其中一顆骰子，接著，剩下兩個位置放另一個骰子排列有(60種)。
- 7.五顆骰子皆不同:從六個數字中選出其中五個(6種)，位置由五個位置逐一放置不同數字的骰子(120種)。

公式: $C(6,X)C(6-X,Y) \times C(5,A) \times C(5-A,B) = \text{組合數}$

紅色部分為骰子數字的組合；黃色部分為位置的組合

紅色部分: X為選擇數字的組數；Y為剩餘數字抽取的組數

黃色部分: A為選擇位置的組合，B為剩餘位置的組合

表一、各組合個數

分類	組合數
五顆骰子相同	6
四顆骰子相同	150
三顆相同另兩顆相同	300
三顆相同另兩顆不同	1200
兩組兩顆相同一顆不同	1800
一組兩顆相同其他三顆不同	3600
五顆骰子皆不同	720

(三)一組骰子出現組合的可能性

當知道五顆骰子共有7776種組合，也知道每個分類共有幾組後，就可知道每種組合出現的可能性，其計算方式為

$$\text{公式: } \frac{\text{每種類別組合的組數}}{\text{所有組數:7776種}} = \text{可能性}$$

表一為各種組合分類的組數，將各組分類的組數除以總組數7776種，即可得到表二所有組合分類的可能性，其中當一作為通用骰時與一不做為通用骰時，出現的可能性具有差異，當一做為通用骰時，一組骰子中，出現三顆相同數字的骰子可能性較高；但當一不做為通用骰時，則一組骰子出現兩顆相同的可能性較高。

表二、各種骰子組合出現的可能性

組合	可能性(百分比)	
	一作為通用骰	一不作為通用骰
五顆相同	2.006%	0.07716%
四顆相同	16.718%	1.929%
三顆相同兩顆相同	6.43%	3.858%
三顆相同兩顆相異	38.58%	15.432%
兩組兩顆相同	11.574%	23.148%
兩顆相同三顆相異	23.148%	46.296%
五顆相異	1.543%	9.259%
總和	100%	100%
一組骰子出現相同 數字的期望值	≈3.09	≈2.05

(四)期望值

期望值在我們應用的場景可以這麼說明：某個指定點數，期望會出現幾顆？

- n 個骰子出現某指定點數的期望值 = 骰子顆數 $n \times$ 點數出現的可能性

在一作為通用骰時，且有決定數字時，每顆骰子的機率為 $\frac{1}{3}$ ，五顆骰子時，期望值是 $5 \times \frac{2}{6} = \frac{10}{6} \approx 1.666$ ，當有指定數字時，五顆骰子可能出現的期望顆數為1.6顆。

在一不作為通用骰時，且有決定數字時，每顆骰子的機率為 $\frac{1}{6}$ ，五顆骰子時，期望值是 $5 \times \frac{1}{6} = \frac{5}{6} \approx 0.833$ ，當有指定數字時，五顆骰子可能出現的期望顆數為0.83顆。

對手骰子較高的機率指定點數的的骰子個數會落在最接近期望值的整數 2顆(一作為通用骰時期望值1.666顆)以及1顆(一不作為通用骰時期望值0.83顆) 附近。

當已知道一組骰子在不同類別的機率時，期望值為

- 每組n個骰子出現某指定點數的期望值 = 骰子顆數 n × 組合出現機率

表三、骰子出現顆數的期望值

	各種組合的期望值			
	當一作為通用骰		一不做為通用骰時	
自己骰盅骰子已知	單顆骰子	一組骰子	單顆骰子	一組骰子
五顆相同	6.67	8.09	5.83	7.05
四顆相同	5.67	7.09	4.83	6.05
三顆相同	4.67	6.09	3.83	5.05
兩顆相同	3.67	5.09	2.83	4.05
一顆相同	2.26	4.09	1.83	3.05
全部不同(自己沒有、對手有)	1.67	3.09	0.83	2.05

由表三得知，當一做為通用骰時與一不做為通用骰時，在兩人遊戲中，期望值約有一顆的差距，而在有指定數字的單顆骰子中，與不指定數字，一組骰子出現相同數字的期望值中，兩者之間約差兩顆骰子。

表四、至少出現的顆數

至少出現顆數	至少出現顆數的可能性	
	一做為通用骰	一不做為通用骰
0顆	100%	100%
1顆	86.8%	59.8%
2顆	53.9%	19.6%
3顆	21%	3.5%
4顆	4.5%	0.3%
5顆	0.4%	≈0.0008%



圖六、一作為通用骰時，相同數字出現剛好與至少的顆數

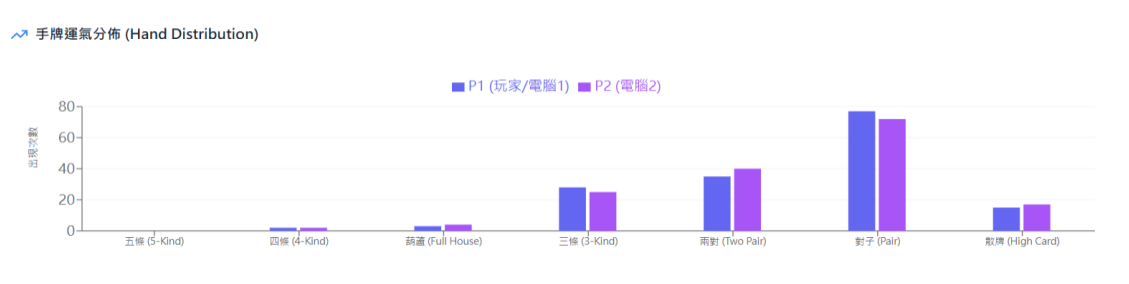


圖七、一不作為同用骰時，相同數字出現剛好與至少的顆數

套用自己骰子出現數量約為3.09顆的數值以及對手出現2顆可能性>50%的機率，可得出喊5顆(或在自己顆數為x時x+2顆)皆在安全範圍內。(條件：1仍作為通用骰)
 可得知若一不再當作通用骰時，喊兩顆是在安全範圍內。

(五) 利用分級找尋最佳策略

1. L0 : 利用基礎觀念設計 策略_Lv0隨機



圖八、玩家與Lv0雙方擲到的骰子組合

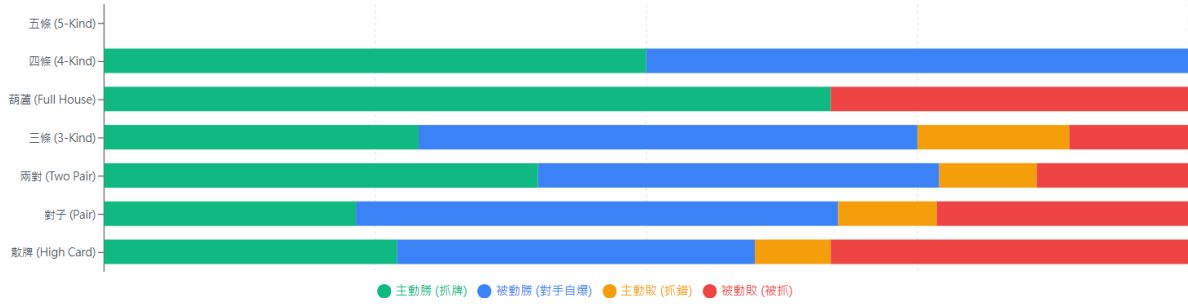
圖七顯示，最常拿到的組合為一個對子有46.56%，其次為兩個對子有23.44%。

表五、玩家與AI:Lv0對弈結果

人V.S.AI:Lv0				
	人		AI	
	勝	敗	勝	敗
五條	0	0	0	0
勝率	無		無	
出現率	0%			
四條	2	0	1	1
勝率	100%		50%	
出現率	1.25%			
葫蘆	2	1	1	3
勝率	67%		25%	
出現率	2.19%			
三條	21	7	7	18
勝率	75%		28%	
出現率	16.56%			
兩對	27	8	14	26
勝率	77%		35%	
出現率	23.44%			
對子	52	25	18	54
勝率	68%		25%	
出現率	46.56%			
散牌	9	6	6	11
勝率	60%		35%	
出現率	10%			
先手	28	22	25	85
勝率	56%		22.7%	
局數	160			
勝率	71%		29%	

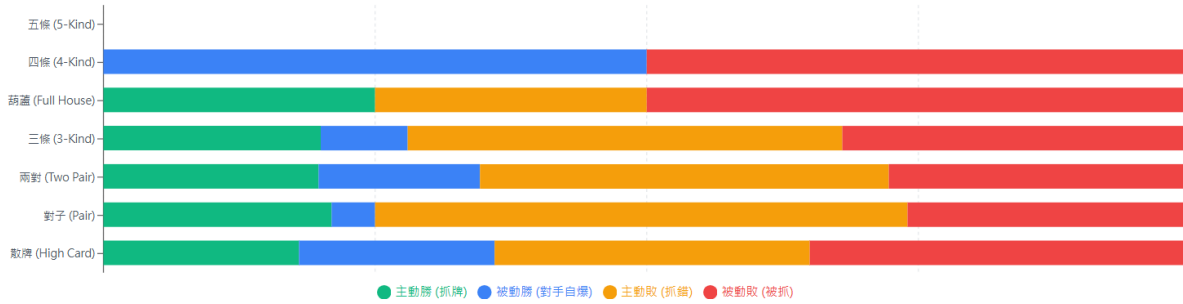
由表五對弈結果得知在與AI:Lv0對戰時，當玩家拿到的組合，有3條組合的骰子時，獲勝的可能性可到70%以上，而葫蘆的組合或是四條的組合，雖然獲勝可能性很高，但出現次數不多，不容易判斷，而AI部分不論哪種組合獲勝的可能性都偏低約在25%~35%之間，四條以上因為次數太少，難以判斷；另外，玩家先手的獲勝可能性在56%，而電腦在先手獲勝可能性為22.7%，根據對弈結果，可獲得與AI:Lv0對弈策略如圖九所示。

Player 1 (玩家/電腦1) 策略

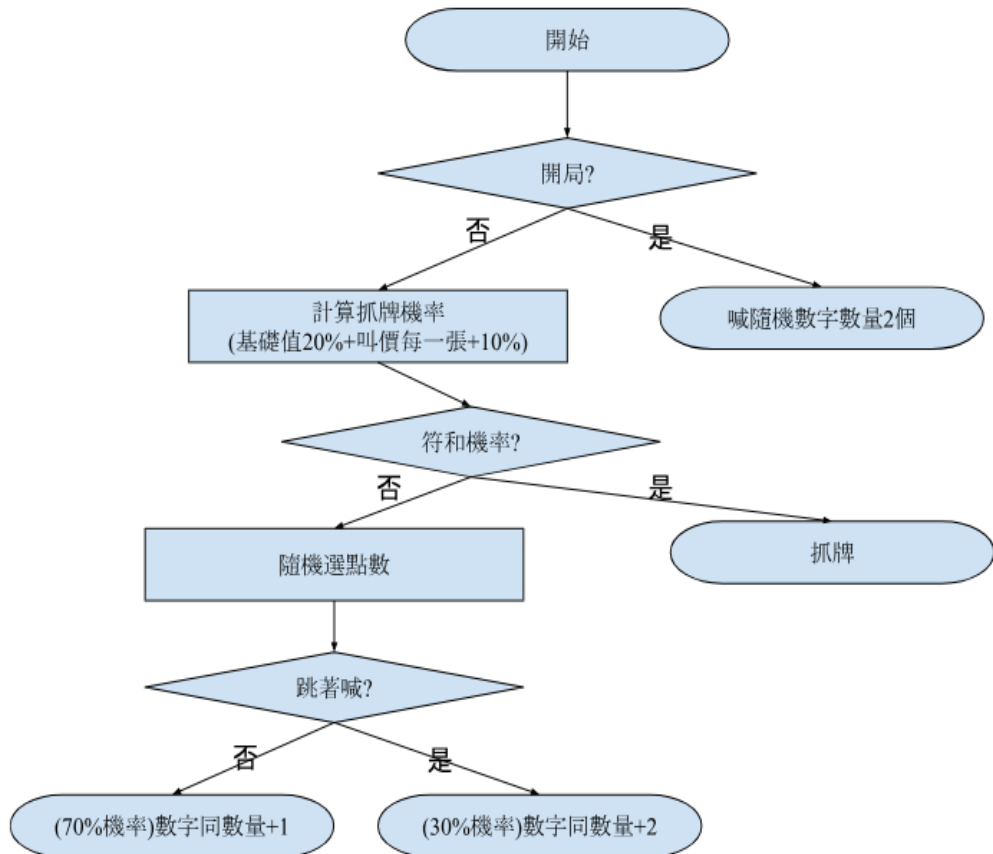


圖九、玩家獲勝及失敗的行動

Player 2 (電腦2) 策略



圖十、電腦AI獲勝及失敗的行動

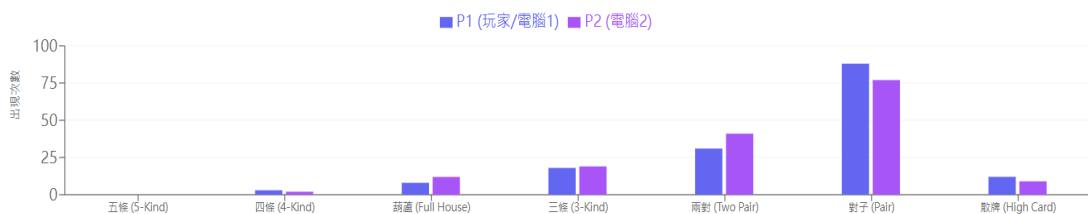


圖十一、AI的Lv0時的策略圖

Lv0的AI是新手入門級，因此，AI叫牌並不會依據自己的手牌開始叫，抓牌僅看喊價數量增加機率(但仍是隨機)。

2. L1：以期望值設計 策略_Lv1機率模型

↗ 手牌運氣分佈 (Hand Distribution)



圖十二、玩家與Lv1雙方擲到的骰子組合

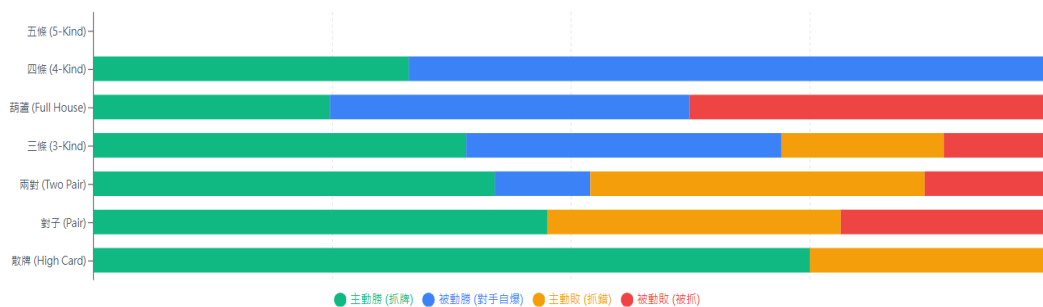
圖十二為玩家與Lv1對戰時，雙方拿到的骰子組合，與Lv0相同，對子的組合數最多，其次是三條。

表六、玩家與AI:Lv1對弈結果

人V.S.AI:Lv1				
	人		AI	
	勝	敗	勝	敗
五條	0	0	0	0
勝率	無		無	
出現率	0%			
四條	3	0	1	1
勝率	100%		50%	
出現率	1.56%			
葫蘆	5	3	9	3
勝率	63%		75%	
出現率	6.25%			
三條	13	5	13	6
勝率	72%		68%	
出現率	11.56%			
兩對	16	15	23	18
勝率	52%		56%	
出現率	22.5%			
對子	42	46	25	52
勝率	48%		32%	
出現率	51.56%			
散牌	9	3	1	8
勝率	75%		11%	
出現率	6.56%			
先手	42	31	42	45
勝率	57.53%		48.27%	
局數	160			
勝率	55%		45%	

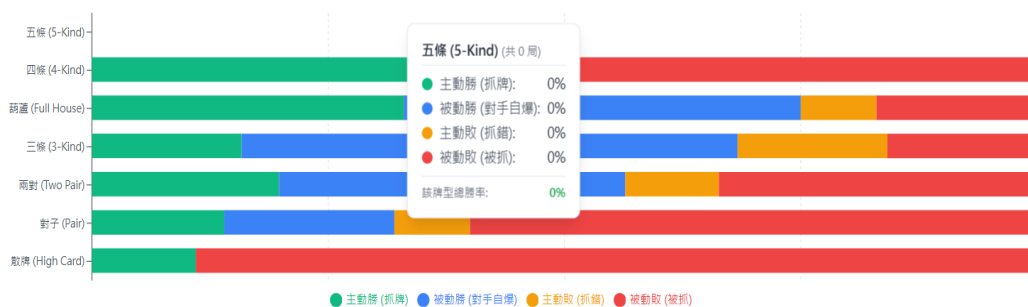
根據表六結果，玩家與AI:Lv1對弈結果，出現三條時獲勝的可能性有72%，葫蘆與四條則是出現次數太少，難以判斷；AI部分則是出現三顆或以上相同骰子獲勝可能性較高，可達到68%，其餘組合或勝率都偏低，未滿50%；另外，雙方先手獲勝可能性在48%~57%之間。

Player 1 (玩家/電腦1) 策略

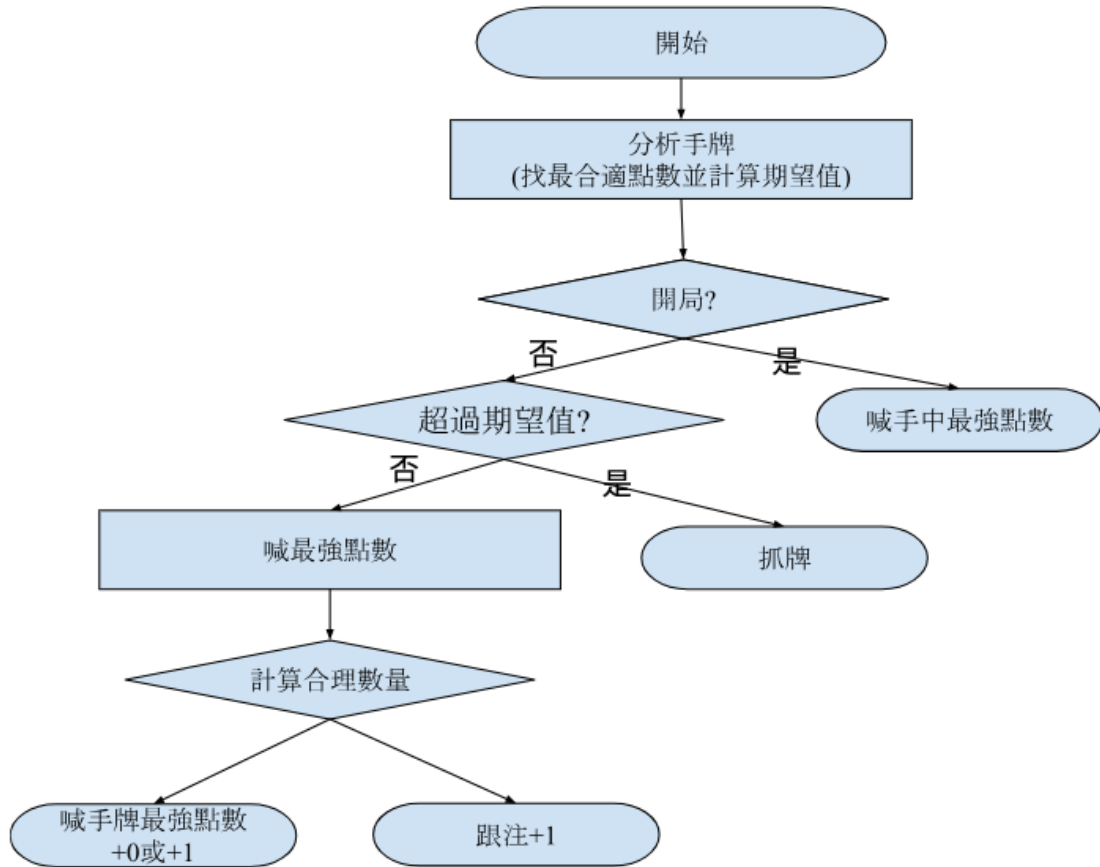


圖十一、玩家獲勝及失敗的行動

Player 2 (電腦2) 策略



圖十二、AI獲勝及失敗的行動



圖十三、AI的Lv1時的策略圖。

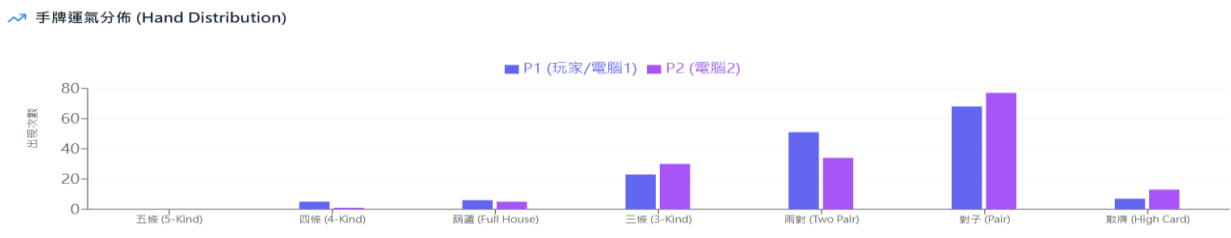
※最強點數：手上持有最多數量骰子點數

在Lv1中，電腦AI已知道可能性與期望值的概念，可以判斷出每一牌局的期望值與可能性，若是叫牌超過期望值，則電腦會抓。而電腦會開始依據自己手牌擬定策略，進行叫牌及抓牌。下圖十四Lv1的抓牌策略 (M:手上有的 Q:對手喊的)

	Q=2	Q=3	Q=4	Q=5	Q=6	Q=7	Q=8	Q=9	Q=10
M=0	喊	喊	抓	抓	抓	抓	抓	抓	抓
M=1	喊	喊	喊	抓	抓	抓	抓	抓	抓
M=2	喊	喊	喊	喊	抓	抓	抓	抓	抓
M=3	喊	喊	喊	喊	喊	抓	抓	抓	抓
M=4	喊	喊	喊	喊	喊	喊	抓	抓	抓
M=5	喊	喊	喊	喊	喊	喊	喊	抓	抓

圖十四、Lv1的抓牌策略

3. L2 : 利用與L1的對弈結果設計 策略_Lv2機率模型，L2會依據自己的牌型調整較牌與抓牌的策略。



圖十五、玩家與Lv2雙方擲到的骰子組合

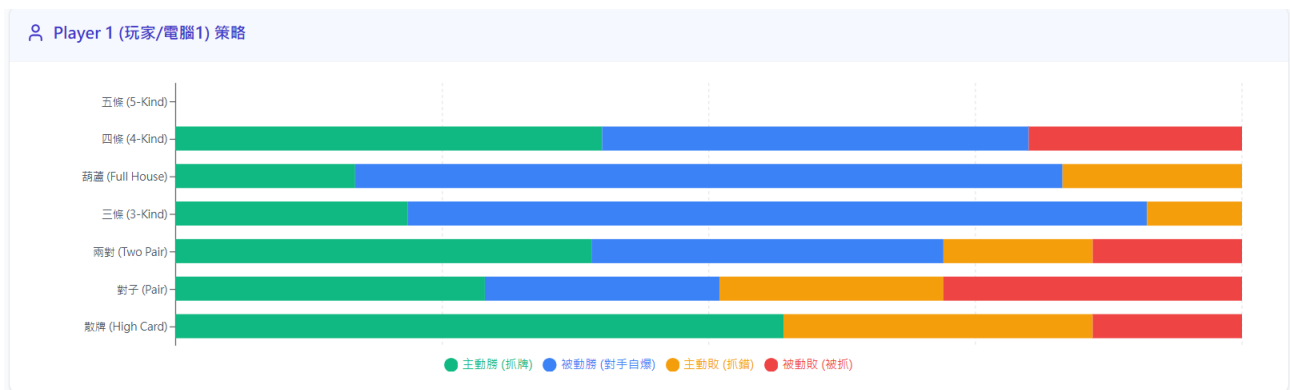
圖十五得知，雙方拿到對子的可能性最高有45.31%，其次為兩對有26.56%，與前兩個AI結果相近。

表七、玩家與AI:Lv2對弈結果

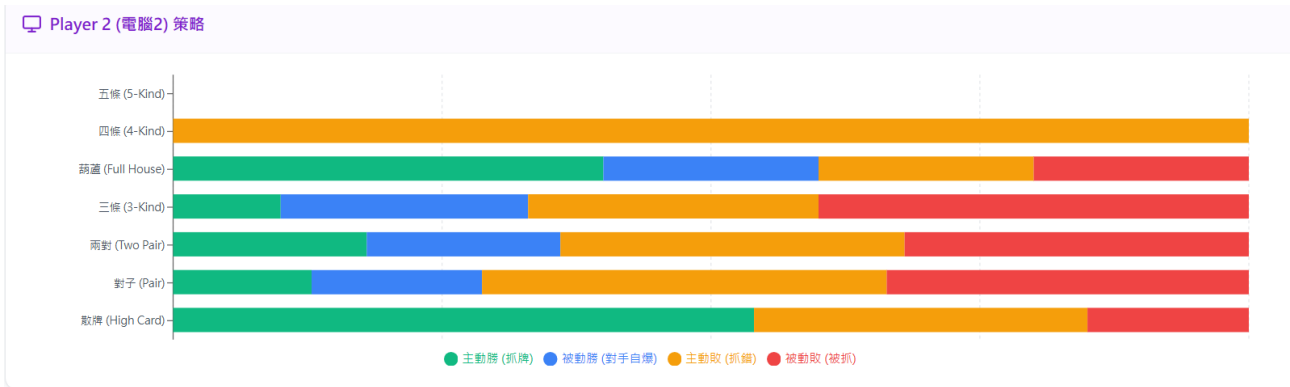
人V.S.AI:Lv2				
	人		AI	
	勝	敗	勝	敗
五條	0	0	0	0
勝率	無		無	
出現率	0%			
四條	4	1	0	1
勝率	80%		0%	
出現率	1.88%			
葫蘆	5	1	3	2
勝率	83%		60%	
出現率	3.44%			
三條	21	2	10	20
勝率	91%		33%	
出現率	16.56%			
兩對	37	14	12	65
勝率	73%		29%	
出現率	26.56%			
對子	35	33	22	55
勝率	51%		29%	
出現率	45.31%			
散牌	4	3	7	6
勝率	57%		54%	
出現率	6.25%			
先手	36	17	37	70
勝率	67.9%		34.6%	
局數	160			

人V.S.AI:Lv2				
	人		AI	
	勝	敗	勝	敗
五條	0	0	0	0
勝率	無		無	
出現率	0%			
四條	4	1	0	1
勝率	80%		0%	
出現率	1.88%			
葫蘆	5	1	3	2
勝率	83%		60%	
出現率	3.44%			
三條	21	2	10	20
勝率	91%		33%	
出現率	16.56%			
兩對	37	14	12	65
勝率	73%		29%	
出現率	26.56%			
對子	35	33	22	55
勝率	51%		29%	
出現率	45.31%			
散牌	4	3	7	6
勝率	57%		54%	
出現率	6.25%			
先手	36	17	37	70
勝率	67.9%		34.6%	
勝率	66%		34%	

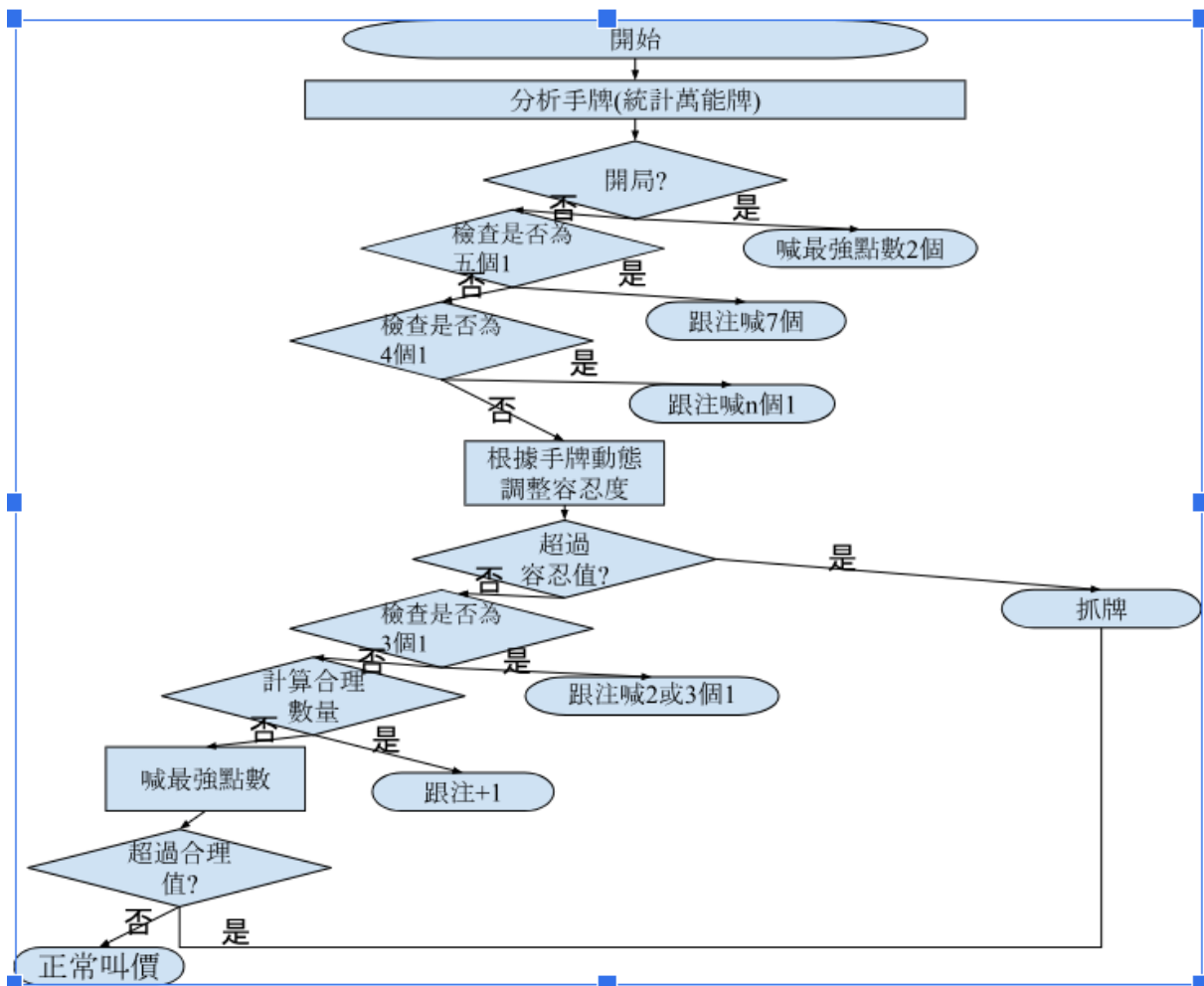
由表七得知，玩家在三條與兩對的或勝可能性最高分別為91%與73%，平均勝率在66%以上，而AI勝率則普遍偏低，獲勝可能性較高的組合因擲出的可能性與次數較少，則不列入參考。在先手部分，玩家獲勝可能性可達67.9%。



圖十六、玩家獲勝及失敗的行動



圖十七、AI獲勝及失敗的行動



圖十八、AI的Lv2時的策略圖

在與Lv1對弈的過程中，發現在很接近期望值的地方Lv1有時會應該抓而沒抓，因此參考前面計算過的機率修正了抓牌策略，如下圖十九，使得喊注更加合理 (增加9%)，另外新增特殊牌型的利用手段，進而更靈活的運用特殊牌型。

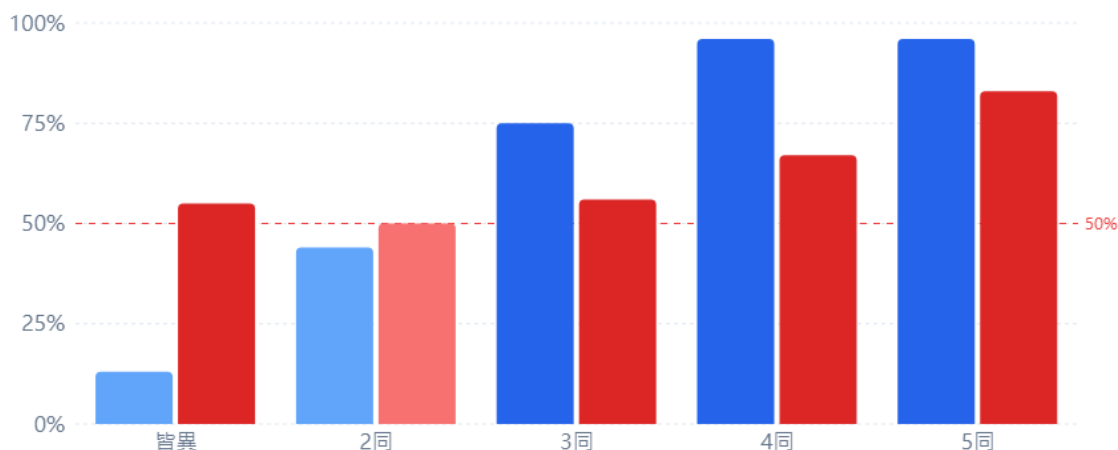


圖十九、修正後的策略

(1) Lv2b及 Lv2c

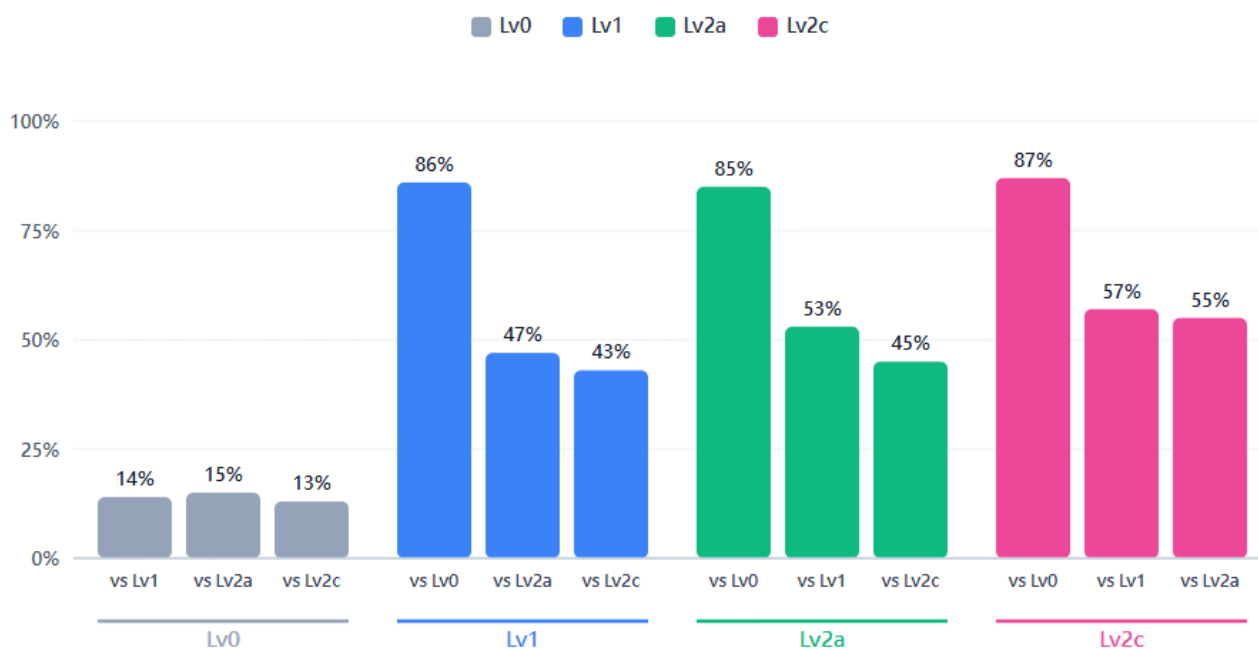
做為比較同時製作了頻繁詐唬的Lv2b，測試中因為勝率及判斷都非常差故沒有加入討論。

測試中發現Lv1藍 和Lv2a紅 (如下圖二十) 在對弈結果上根據骰子類型的優勢差異明顯，可以明顯看到Lv1對於3個及以上相同的骰子勝率高，2個及以下的則是Lv2a占優勢，因此製作了結合兩者優點根據手牌切換策略的Lv2c



圖二十、Lv1 和Lv2a 10000局對弈牌型統計

2. 對弈結果分析



圖二十一、AI對AI 勝率統計 各10000局

可以看到隨著AI的策略優化，勝率也有所提升。

為實驗AI策略的效果，我們作為玩家與四種不同難度的AI進行約900場對局並且用Excel表格統計這其中1972次決策的正確性。分析結果會在第五部分討論。

全部人類決策記錄 (按 AI 等級分區)													
1	等級	玩家	局號	步驟	行動類型	叫牌內容	自己手牌	自己點數	期望總計	超出期望	成立機率	決策判定	勝利者
1949	Lv2c		91	5	Challenge	抓牌	6-6-4-5-3	1			0.045	正確抓牌	人
1950	Lv2c		92	2	Bid	3個5	3-5-5-4-1	3	4.67	-1.67		保守叫牌	電腦
1951	Lv2c		92	4	Bid	5個5	3-5-5-4-1	3	4.67	0.33		合理叫牌	電腦
1952	Lv2c		93	1	Bid	4個3	4-1-3-3-4	3	4.67	-0.67		合理叫牌	人
1953	Lv2c		93	3	Bid	6個3	4-1-3-3-4	3	4.67	1.33		合理叫牌	人
1954	Lv2c		94	2	Bid	2個5	3-3-1-2-5	2	3.67	-1.67		保守叫牌	電腦
1955	Lv2c		94	4	Challenge	抓牌	3-3-1-2-5	2			0.868	誤抓	電腦
1956	Lv2c		95	1	Bid	2個2	5-2-3-6-6	1	2.67	-0.67		合理叫牌	人
1957	Lv2c		95	3	Bid	4個2	5-2-3-6-6	1	2.67	1.33		合理叫牌	人
1958	Lv2c		95	5	Challenge	抓牌	5-2-3-6-6	1			0.045	正確抓牌	人
1959	Lv2c		96	2	Bid	2個5	2-1-5-3-5	3	4.67	-2.67		保守叫牌	電腦
1960	Lv2c		96	4	Bid	5個2	2-1-5-3-5	2	3.67	1.33		合理叫牌	電腦
1961	Lv2c		97	1	Bid	2個2	3-2-3-4-5	1	2.67	-0.67		合理叫牌	人
1962	Lv2c		97	3	Bid	4個4	3-2-3-4-5	1	2.67	1.33		合理叫牌	人
1963	Lv2c		97	5	Challenge	抓牌	3-2-3-4-5	1			0.045	正確抓牌	人
1964	Lv2c		98	2	Bid	3個4	3-5-4-2-1	2	3.67	-0.67		合理叫牌	電腦
1965	Lv2c		98	4	Bid	6個4	3-5-4-2-1	2	3.67	2.33		稍高叫牌	電腦
1966	Lv2c		99	1	Bid	4個2	3-5-2-2-4	2	3.67	0.33		合理叫牌	電腦
1967	Lv2c		99	3	Challenge	抓牌	3-5-2-2-4	1			0.21	正確抓牌	電腦
1968	Lv2c		100	1	Bid	2個3	4-5-6-2-6	0	1.67	0.33		合理叫牌	電腦
1969	Lv2c		100	3	Bid	4個2	4-5-6-2-6	1	2.67	1.33		合理叫牌	電腦
1970	Lv2c		101	1	Bid	6個5	5-1-1-4-5	4	5.67	0.33		合理叫牌	電腦
1971	Lv2c		102	1	Bid	4個6	6-1-5-6-3	3	4.67	-0.67		合理叫牌	人
1972	Lv2c		103	2	Bid	4個5	1-5-3-2-1	3	4.67	-0.67		合理叫牌	電腦
1973	Lv2c		103	4	Challenge	抓牌	1-5-3-2-1	3			0.539	誤抓	電腦
1974	Lv2c		104	1	Bid	5個6	6-6-1-3-6	4	5.67	-0.67		合理叫牌	人
1975	Lv2c		105	2	Bid	4個4	6-1-4-4-2	3	4.67	-0.67		合理叫牌	電腦
1976													

圖二十二、AI對玩家 決策紀錄

四個 AI 等級對弈統計								
指標	Lv0 (最易)		Lv1 (初級)		Lv2a (中級)		Lv2c (進階)	
指標	數值	佔比/說明	數值	佔比/說明	數值	佔比/說明	數值	佔比/說明
總局數	125		240		203		345	
人類勝場	97	77.6%	143	59.6%	127	62.6%	203	58.8%
AI 勝場	28	22.4%	97	40.4%	76	37.4%	142	41.2%
總叫牌次數	137		401		319		608	
合理叫牌(人類)	72	52.6%	260	64.8%	217	68.0%	409	67.3%
總抓牌次數	56		189		101		158	
正確抓牌(人類)	34	60.7%	171	90.5%	95	94.1%	145	91.8%
誤抓	22	39.3%	18	9.5%	6	5.9%	13	8.2%

圖二十三、AI對玩家 對局分析

伍、討論

一、好牌與壞牌

骰子吹牛是一種不完全資訊的賽局遊戲，在遊玩過程中，較好的策略是依據自己的骰子組成加上根據期望值去判斷對手骰子組成的可能性與骰子顆數，並制定適合的策略。

1.當自己骰子組成為全部不同或是重複性低或是沒有擲出的情況下，因為，自己的骰子組成低於期望值，當對手叫喊的顆數若大於或等於期望值的情況下，則抓牌會是比較適合的策略。

2.當自己的骰子出現組合為單一數字重複多顆高於期望值的情況下，可持續喊高，當叫喊的顆數高於期望值，可增加對手抓牌失敗的可能性。

因此，骰子組成的好、壞牌會根據策略是以主動抓牌或是等別人抓而決定，雖然，骰子組成重複性低，或是自己沒有出現相同數字的骰子，但若是調整策略，也會增加獲勝的可能性，但是自己有重複性高的組合，可增加自己的叫牌策略，誘使對手抓牌提高勝率；若是自己牌型散亂，則可以使用的策略會有侷限，在有通用骰的情況下，出現三顆相同骰子會是較好的組合；而在沒有通用骰時，至少出現兩顆會是比較可以掌握局勢，而若出現三顆以上，則是可以提高獲勝的可能性，也就是說出現三顆相同骰子以上的組合獲勝的可能性較高

二、通用骰的兩面性

骰子吹牛當中的數字一，若是滿足條件的情況下可作為通用骰，但是通用骰數量過多，高於期望值的情況下，除了會影響對手判斷，也會干擾自己的決策，因為，對手也可使用通用骰當作任意數，因此，當己方出現較多通用骰時，叫牌跟抓牌的策略擬定要再謹慎。

三、抓牌與叫牌的極限顆數

當我們跟AI進行模擬遊戲時，Lv2時常會叫到接近極限的顆數，若是抓牌則很容易抓失敗，若是繼續往上追加，則容易被抓成功，因此，我們覺得應該會有一個最佳策略的數字存在，根據期望值及至少出現顆數骰子的可能性進行分析，兩人遊戲的極限值約在4~5顆之間，而數字則需要經過來回的叫牌進行資訊蒐集，才能有效判斷。

四、骰子組合的類別與制定策略

雖然吹牛大多以最高顆數為叫牌的主要策略，但若是牌組出現兩個一隊或是葫蘆組合(三顆相同加上另兩顆為一對)，則可以讓自己的叫牌策略更為多元，有較多變化，加上通用骰後，可以叫牌使組合變化更為複雜，達到干擾對手判斷的目的。

五、AI模擬與真人對戰

我們在與AI進行對弈時，發現AI的叫牌及抓牌都有一定的模式可以預測，因此，若是照著模式走，預測AI的行為及叫牌、抓牌，則可以提高與AI對戰的勝率；在與人對弈時，人在反應及調整的速度比較快，若是自己行為被預測的時候，則會及時修改策略，干擾對手判斷，可預測性會比AI低。

六、資訊量與勝負

在與AI對弈過程中，有時候只有2~3步驟，遊戲就結束，有時候需要5~6步驟，才會結束遊戲，而步驟越多，表示所得到的資訊量也越多，可能會影響到勝負，未來研究也可以探討約需要幾個步驟可以獲得足夠獲勝的資訊量。

七、策略綜合討論

這次的策略設計有達到預期的成果，利用機率判斷確實能達到與熟練玩家對局的難度，起初我們的勝率甚至還比不過AI，但是經過多場對局，我們能逐漸找到AI固定的模式，使得容易從AI的喊注由經驗判斷AI的骰子組合，或者可以提前知道AI對於喊注的反應，兩者都能使AI進入玩家的陷阱，經過討論我們認為更優的策略要有以下兩項特性。

1.隨機性：由圖二十三AI對玩家對局分析可以看到 雖然說Lv0勝率低，但是因為其不可預測性常常會導致人類玩家誤判局勢，誤抓的機率明顯高於其他模型，若能增加些許隨機的策略，可以使玩家無法精準預測AI的骰子組合。

2.記憶性：與人類玩家相同，如果AI也能透過對局了解人類玩家的習慣，可以避免被落入重複的陷阱，甚至可以詐唬對方。

陸、結論

根據我們的實驗、計算與統計，我們得出結論：1 還是通用骰的時候，對手出現 2 顆相同數字的機率超過一半。所以手上有 x 顆，喊 $x+2$ 顆是安全的。一旦有人喊過 1，這個安全範圍縮小，喊 $x+1$ 顆才安全。一旦對手超過這個範圍，就可以考慮抓牌。

手牌好不好，決定用哪種打法。你拿到 3 顆以上相同，就主動加注，讓對手自己判斷失誤。牌型散亂，就保守跟牌，等對手叫超期望值再出手抓牌。

AI 對弈的數據也支持。Lv1 加入期望值判斷，對 Lv0 勝率升到 86%。Lv2c 根據手牌切換策略，對 Lv1 和 Lv2a 都保持 55% 以上的勝率。策略愈細，勝率愈高。

不過我們也發現，純靠機率的 AI 有一個弱點，對局多了會被看穿規律。要讓對手真的無法預測，策略裡需要一些不按牌理出牌的空間。這是我們下一步想繼續研究的方向。

柒、參考文獻

何承蓉、盧俊憲、鄭宇軒、葉益甫 (2006)。驚爆骰子樂之吹牛大王。臺北縣新莊市昌平國民小學。

宋奇恩、張予馨(2025)。看誰最會吹-破解吹牛的秘密。新北市林口區新林國小獨立研究。

周宗穎(2023)。實現強化學習於多人吹牛骰子遊戲之探討。第二十二屆旺宏科學獎 成果報告書

黃晟瑋、呂承穎、劉祐瑋、陳以芯、蘇瑱、范裕紘(2020)。「欲罷不能」桌遊機率之探討與延伸。嘉義市東區崇文國民小學。

附錄.1 吹牛遊戲五顆骰子的可能性程式

```
import itertools

from collections import Counter

def analyze_dice_probabilities():
    # 設定骰子的面數和顆數
    faces = [1, 2, 3, 4, 5, 6]
    num_dice = 5

    # 1. 算出五顆骰子中，有幾種「排列」 (Order Matters)
    # 這是計算機率的分子，因為 11112 和 11211 在物理投擲上是不同的事件
    all_permutations = list(itertools.product(faces, repeat=num_dice))
    total_permutations = len(all_permutations)

    # 2. [新增功能] 算出五顆骰子中，有幾種「組合」 (Order Doesn't Matter)
    # 這裡 11112 和 11211 會被視為同一種情況
    unique_combinations = list(itertools.combinations_with_replacement(faces, num_dice))
    total_unique_combinations = len(unique_combinations)

    print(f"--- 基礎數據 ---")
    print(f"骰子顆數: {num_dice}")
    print(f"1. 總排列數 (考慮順序): {total_permutations} (計算式: 6^5)")
    print(f"說明: 用於計算機率的分子，11112 與 11211 視為不同。")
    print(f"2. 唯一組合數 (不看順序): {total_unique_combinations}")
    print(f"說明: 這是你要求的「盤面種類」，11112 與 11211 視為相同。")
    print("-" * 30)

    # 初始化計數器
    counts = {
        "五顆相同 (豹子)": 0,
        "四顆相同 (鐵支)": 0,
        "三顆相同+兩顆相同 (葫蘆)": 0,
        "三顆相同+兩顆相異 (三條)": 0,
        "兩組兩顆相同+一顆相異 (兩對)": 0,
        "兩顆相同+三顆相異 (一對)": 0,
        "五顆皆相異 (散牌/順子)": 0
    }
```

```

# 3. 將五顆骰子中的組合依照組成進行分類
# 注意：計算機率時，我們仍然必須使用 all_permutations (7776種) 來跑迴圈
# 因為「散牌」出現的物理機率遠高於「豹子」，不能用 unique_combinations 來算機率
for combo in all_permutations:
    counts_obj = Counter(combo)
    freqs = sorted(counts_obj.values(), reverse=True)

    if freqs == [5]:
        counts["五顆相同 (豹子)"] += 1
    elif freqs == [4, 1]:
        counts["四顆相同 (鐵支)"] += 1
    elif freqs == [3, 2]:
        counts["三顆相同+兩顆相同 (葫蘆)"] += 1
    elif freqs == [3, 1, 1]:
        counts["三顆相同+兩顆相異 (三條)"] += 1
    elif freqs == [2, 2, 1]:
        counts["兩組兩顆相同+一顆相異 (兩對)"] += 1
    elif freqs == [2, 1, 1, 1]:
        counts["兩顆相同+三顆相異 (一對)"] += 1
    elif freqs == [1, 1, 1, 1, 1]:
        counts["五顆皆相異 (散牌/順子)"] += 1

# 輸出結果
print(f"{'類別名稱':<20} | {'次數':<8} | {'機率 (%)':<12} | {'機率 (分數)}")
print("-" * 65)

check_sum = 0

for category, count in counts.items():
    probability_percent = (count / total_permutations) * 100
    # 保持小數點後 5 位
    print(f"{'category':<20} | {'count':<8} | {'probability_percent':<12.5f}% | {'count'} / {'total_permutations'}")
    check_sum += count

print("-" * 65)
print(f"驗證加總次數: {check_sum} (應等於 {total_permutations})")

```

```
if __name__ == "__main__":
    analyze_dice_probabilities()
```

執行結果

```
--- 基礎數據 ---
骰子顆數: 5
1. 總排列數 (考慮順序): 7776 (計算式: 6^5)
   說明: 用於計算機率的分母, 11112 與 11211 視為不同。
   說明: 用於計算機率的分母, 11112 與 11211 視為不同。
2. 唯一組合數 (不看順序): 252
   說明: 這是你要求的「盤面種類」, 11112 與 11211 視為相同。
-----
類別名稱          | 次數      | 機率 (%)      | 機率 (分數)
-----
五顆相同 (豹子)   | 6         | 0.07716 %    | 6/7776
四顆相同 (鐵支)   | 150      | 1.92901 %    | 150/7776
三顆相同+兩顆相同 (葫蘆) | 300     | 3.85802 %    | 300/7776
三顆相同+兩顆相異 (三條) | 1200    | 15.43210 %   | 1200/7776
兩組兩顆相同+一顆相異 (兩對) | 1800   | 23.14815 %   | 1800/7776
兩顆相同+三顆相異 (一對) | 3600    | 46.29630 %   | 3600/7776
五顆皆相異 (散牌/順子) | 720     | 9.25926 %    | 720/7776
-----
○ 驗證加總次數: 7776 (應等於 7776)
```

附錄.2 AI對弈程式v5.0(2 / 1版本)

<https://poe.com/preview/RMmRjOMqEWFFjvTPQEKL>



附錄.3 對弈複盤分析器

<https://poe.com/preview/1b7e144tQvOGygk5gBqw>

研究心得

李宙恩：

原本我們想要做偏向遊戲、人文類的題目，但是我們的家長要求我們做科展，正好老師上課時間提到了吹牛，我們就選擇了這個題目。做這個研究我覺得非常有趣，也讓我多認識了一種新遊戲，只不過在和AI試玩時，剛開始總找不到AI的規律，所以就一直輸，只不過玩到後面就愈來愈得心應手，所以勝率也就愈來愈高。只不過對戰多種AI之後就容易將AI之間的策略搞混，勝率也無法穩定維持。

張煒杰：

一開始我們因為家長要求做科展而為了題目煩惱很久，之後老師建議我們可以以骰子吹牛為題目，一開始大家都認為很無聊，可是更深入後發現其實有百戰百勝的方法，而不是完全靠”11111”的運氣而已。這個研究最困難的部分是骰子的組合，因為如果想要統計骰子點數出線機率的話，是必須找出所有可能的，而如果只用自己投出的來算的話，是會有誤差的，可使用AI則會有正確性的疑慮。

李泓杰：

我因為對骰子和機率有興趣，所以決定和組員們一起做這個研究。我們的研究有用到AI，一開始我們一直無法調整出一個會所有規則的AI，所以總是發現AI再亂喊(叫)，如骰子顆數小於玩家人數(喊0個加數字)、不過也在讓AI從很菜到比我們強的過程中得到不少成就感，而且也從實驗過程中增強吹牛技巧，成為組裡最強的人。(其中一些技巧很多都是採用AI的策略)除了吹牛技巧外，也從老師那裏學到了很多數學知識，尤其是和機率有關的，讓我很慶幸做了這個實驗，以後也想繼續做實驗。

林楷恩：

我們原本是要做minecraft的研究，但發現科展不行使用這個主題，於是老師建議我們做吹牛的研究，一開始玩的時候，我覺得這就是一個平凡無奇的遊戲，直到後來深入研究，才發現可以用多種方式來達到百戰百勝的效果，而不是沒有目的性的亂喊，完全靠運氣獲勝的那種方式。在研究的過程中我們也遇到了一些困難，像是找出骰子的所有組合；推算期望值；設計AI……等等的問題，我們花了好一段時間才完成。做完這份研究後，我的數學能力也提升了不少，更讓我知道如何寫生成式AI的程式，可說是收穫滿滿。