

投稿類別：專題報告類

篇名：巴什博弈的數學分析及不完全資訊博弈的應用

作者：

醫管系、班級、姓名

指導老師：

壹、前言

一、研究動機

從《天才基本法》這部劇的第四集中，研究者首次接觸到了巴什博弈的特殊玩法，進而對這款看似簡單卻蘊含策略的遊戲產生興趣。隨後在學習高中數學的排列組合這個單元及近期流行的解壓玩具「滅鼠先鋒」中，發現都是相似的原理，卻有各種變形的形式與玩法。使研究者不禁對原版遊戲的核心原理產生好奇，也欲用Python 程式將原理實體化，使其更可觀易理解。並在原有的規則上設計加入不完全資訊的新玩法，使其不只是策略玩法，更加入運氣成分，使遊戲加入不確定性以增加趣味。

二、研究目的

- (一) 了解巴什博弈的規則與原理，掌握其勝負策略
- (二) 利用 python 程式實體化遊戲原理，並藉此證明安全局面與其預測性
- (三) 將遊戲規則改良為不完全資訊戰局，並計算其勝負機率與最適策略

貳、文獻探討

一、巴什博弈基本規則與原理

(一) 規則

巴什博弈(Bash Games)是一個簡單且具有策略性的兩人博弈遊戲。遊戲規則是：兩個玩家在一堆石頭中輪流拿取，一次最多拿規定之上限，不可不拿，將場上石頭取完者勝。由於規則直觀、操作容易，常常被當作入門的程式設計練習（Leetcode，無日期）或是高中數學的基礎邏輯推演。

巴什博弈也被視作一款經典數學遊戲拈博弈(Nim Game)的簡化。Buton(1901)提出拈博弈這款數學遊戲。而巴什博弈的特點是在拈博弈的基礎上加入限制，將拈博弈中多堆石頭改為一堆石頭，並在每回合加入「拿取上限 m 」的限制，使巴什博弈的推導過程更具直觀性，策略更集中。

(二) 基本原理

本文中 s 為總棋子數， m 為每回合最多可拿之上限， k 為任意正整數，分析巴什博弈的必勝策略。

首先，當場上剩下 $(m + 1)$ 顆棋子時，不論一方拿走 1 至 m 顆，都無法一次拿完，因此必會剩下 1 至 m 顆供對方操作。反之，若場上剩下的棋數小於或等於 m ，則該方可一次拿完並獲勝。（ACM Book，無日期）

本研究進一步將 $(m + 1)$ 擴大，總數 s 設為 $k(m + 1)$ ，此時後手只要根據先手拿走的數量 x ，拿取 $(m + 1 - x)$ ，使場面上保持 $k(m + 1)$ ，直至拿完，後手勝利。此種後手勝利的局面為 $s = k(m + 1)$ 。反之，若 s 不等於 $k(m + 1)$ ，則先手只要拿走 s 除以 $(m + 1)$ 的餘數顆棋子，使場上剩下 $s = k(m + 1)$ ，重複上述策略，即可獲勝。以 $(m + 1)$ 作為基準雙方交替控制局面，進而取得勝利。

二、安全局面

拈博弈和巴什博弈的共通點是兩者皆存在「安全組合(Lsafe combinateion)」，是指一方玩家在場上留下一組數字，且之後不出任何差錯，則另一位玩家無法獲勝。這樣的一組數字被稱為「安全組合」(Bouton,1901)。

而在巴什博弈中，當出現 $s = k(m + 1)$ 時，只要後手不出現失誤，則無論先手如何拿取，後手必勝，此類局面即為安全局面。由此可知，只要玩家掌握總數以及拿取上限，便可將場面控制為安全局面，勝負從一開始便已定。

三、安全局面與完全資訊賽局

依照玩家可掌握的資訊情況，可分為完全資訊賽局及不完全資訊賽局，如象棋、西洋棋和圍棋都屬於完全資訊賽局；而德州撲克及橋牌皆屬於不完全資訊賽局（葉俊廷，2009）。李松沛（2020）則指出「**而完全資訊賽局不用考慮機率問題，也就是說所有可以拿來參考及運算的資訊都呈現在棋面上。**」

因巴什與拈博弈皆有安全組合的特性，因此均具有高度的可預測性。只要雙方都掌握遊戲規則並能進行正確的計算，比賽的勝負在遊戲初期便可被推論出來。而要出現安全組合的情況，必須建立在完全資訊的情況下。雖然拈博弈與巴什博弈同屬完全資訊博弈，但構成「完全資訊」所需掌握的關鍵資訊卻有所不同。在拈博弈中，玩家必須明確知道每一堆物品的數量，才能正確計算尼姆和，判斷局勢優劣（ByteGather，2021）。而在巴什博弈中，玩家則需知曉剩餘總數與每回合的最大拿取上限，才能辨識是否處於安全局面。

四、Python 程式語言

Python 是一款被廣泛使用的程式語言，具有兩大優勢。一為它可提升開發人員的效率，相較於其他程式語言，Python 可使用較少的程式碼來編寫程式，並使閱讀程式碼更加簡便；其次，由於其語法結構接近英文，因此開發人員可較容易理解與閱讀程式碼，對於討論與修正程式時更順暢。（AWS，無日期）

綜上所述，巴什博弈是一款具備數學邏輯且有可預測性的完全資訊賽局，其核心在於玩家掌握剩餘總數與每回合的最大拿取上限後，並以將場面控制為安全局面為目標制定策略，則勝負從開局便定。由於其規則簡潔、邏輯清晰，適合以程式語言進行模擬與分析。而 Python 語言具備語法簡潔與易理解的特點，因此筆者將其作為工具進行模擬對戰。

參、研究方法

一、研究方法

- （一）文獻分析法：藉由上網收集文獻資料，了解巴什博弈的規則與原理。
- （二）Python 程式模擬：撰寫巴什博弈對戰程式，模擬玩家與對手交替拿取棋的過程，歸納出四種情況的流程函式圖，並藉由程式還原，證明安全局面與其預測性。
- （三）將變形規則以樹狀圖分析與表格統整：在傳統巴什博弈中加入以抽卡決定上限的規則，使其變成不完全資訊賽局，並分析總數與拿取上限組合的勝率與最佳策

略，並透過樹狀圖與表格統整結果。

二、研究流程

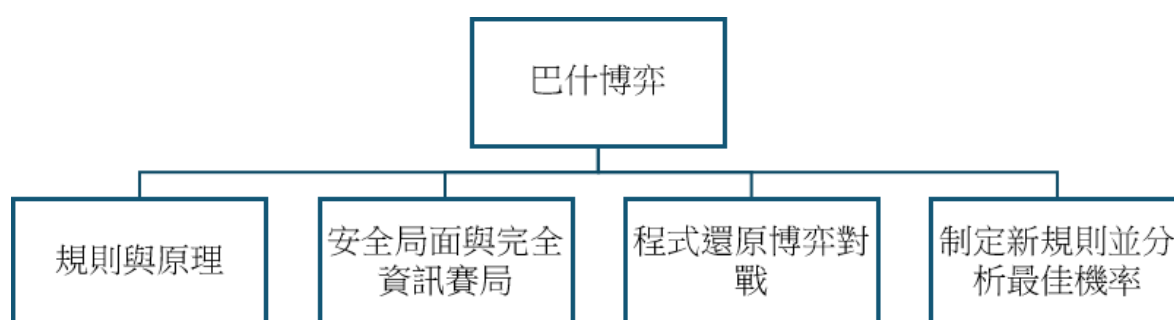
圖一：研究流程圖



圖一資料來源：研究者自行繪製

三、研究架構

圖二：研究架構圖



圖二資料來源：研究者自行繪製

肆、研究分析與結果

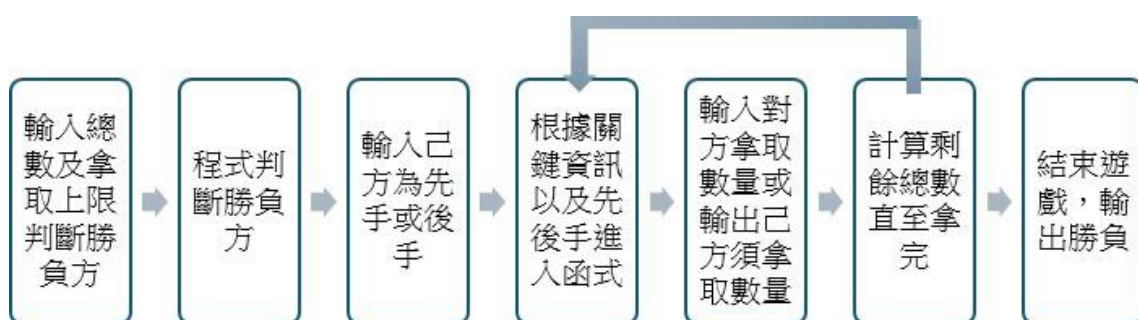
一、規則訂立

由於巴什博弈玩法變化性大，因此在此先明定所採用之規則。遊戲由兩人操作對戰，雙方輪流從一堆總數 s 的棋子中取 1 至 m 個數，取完者勝。此外，將先手玩家稱為 A ，後手玩家稱為 B 。不論己方為先後手， x 表示己方回合拿取的數， y 為對方回合拿取的數。且 s 除以 $(m + 1)$ 的餘數之後將以 $s \% (m + 1)$ 表示。

二、程式還原

本研究使用Python 撰寫巴什博弈模擬對戰程式，將理論中的安全局面以實際對戰的模式呈現。透過此程式，玩家可輸入總數及拿取上限，程式會自動判斷勝負方，再由玩家選擇先後手並進行模擬實戰。實戰中，只要輸入對方拿取的數量，程式會計算剩餘總數以及拿取上限，並輸出己方需拿取的數量。以下是程式流程圖：

圖三：程式進行流程圖



圖三資料來源：研究者自行繪製

(一) 判斷勝負

根據文獻指出巴什博弈只要掌握完全資訊，便可輕易判斷勝負。其核心理論為：當 $s = k(m + 1)$ 時進入安全局面，則只要不出現差錯，則後手獲勝。因此透過程式判斷，即可利用 $s \% (m + 1)$ 是否等於 0，輸出正負方，如圖四。

圖四：判斷勝負程式碼

```

#----- 遊戲開始 -----
s = int(input("總數："))
m = int(input("拿取上限："))

if s % (m + 1) == 0:
    print("後手勝")
else:
    print("先手勝")
    
```

圖四資料來源：研究者自行製作

(二) 四種情境函式

圖五：玩家選擇先後手程式碼

```

z = input("先手為A，後手為B")
    
```

圖五資料來源：研究者自行製作

如圖五，讓玩家選擇成為先手或後手，並根據玩家的選擇進入四種情境函式其一，函式分別為先手為勝方(winA)、後手為勝方(winB)先手為敗方(loseA)及後手為敗方(loseB)。其判斷式如下：

圖六：四種情境函式程式碼

```
if (z == "B") and (s % (m + 1) == 0):  
    winB(s, m)  
elif (z == "A") and (s % (m + 1) == 0):  
    loseA(s, m)  
elif (z == "B") and (s % (m + 1) != 0):  
    loseB(s, m)  
else:  
    winA(s, m)
```

圖六資料來源：研究者自行製作

如圖六所示，當 $s \%(m + 1) = 0$ 時，玩家恰好選擇成為後手，則進入後手勝(winB)的函式；若此時玩家成為先手，則進入先手敗(loseA)的函式。反之，當 $s \%(m + 1) \neq 0$ 時，玩家選擇成為後手，則進入後手敗(loseB)的函式；玩家選擇先手，則進入先手勝(winA)。

1、後手勝

圖七：後手勝(winB)函式程式碼

```
def winB(s, m):  
    while True:  
        y = int(input("對手拿的數量："))  
        s -= y  
        x = m + 1 - y  
        s -= x  
        print("請拿", x, "顆")  
        if s < m + 1:  
            print("you win")  
            sys.exit()
```

圖七資料來源：研究者自行製作

圖七為當 $s \%(m + 1) = 0$ 且玩家成為後手進入後手勝(winB)的函式。遊戲一開始便為安全局面，代表對手須主動破壞此平衡，而後手只需被動拿取 $(m + 1) - y$ 顆，使場面的總數 s 始終保持 $s = k(m + 1)$ 。而總數在雙方拿取的過程中不斷重新計算直至 s 小於 $(m + 1)$ ，則代表場上剩下的數玩家足以全部拿走，程式輸出勝利字樣。

2、先手勝

圖八：先手勝(winA)函式程式碼

```
def winA(s, m):  
    if s > m:  
        x = s % (m + 1)  
        print("請拿", x, "顆")  
        s -= x  
        winB(s, m)  
    else:  
        print("you win. 拿", s, "顆")  
        sys.exit()
```

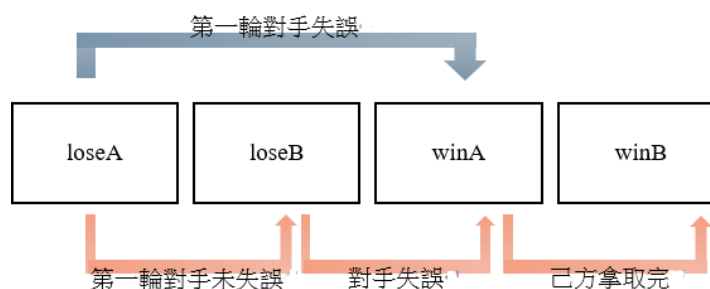
圖八資料來源：研究者自行製作

又如圖八所示，當 $s \%(m + 1) \neq 0$ 且玩家成為先手進入先手勝(winA)的函式。先

手勝的條件為遊戲初始數值不構成安全場面，因此先手玩家只需要先將 $s\%(m+1)$ 顆棋拿走，使場面變成安全局面，原先的先手也成為「相對後手」，進入 winB 的函式中，被動拿取 $(m+1)-y$ 顆，直至勝利。

3、函式關係圖

圖九：四種情境函式關係圖



圖九資料來源：研究者自行繪製

以圖九四種情境函式關係圖下方橘色箭頭的部分進行具體的範例：假設 $s=20$ ， $m=3$ ，即為後手勝，若玩家選擇成為先手，則進入 loseA 的函式中。在第一輪中，己方拿了 3 顆，而對手也沒有失誤拿了 $m+1-x$ 顆，也就是 1 顆，則在第二輪己方拿了 2 顆後進入 loseB 的函式中，輪到對方拿取，若此時對手發生失誤，原應 2 顆卻誤拿 3 顆，則此時場上剩下 11 顆棋子，不為安全局面，第三輪進入 winA 的函式，程式輸出應拿取 3 顆。此時場上剩下 8 顆棋子為安全局面，進入 winB ，即無論對手如何拿，皆能贏得比賽。

表一：程式模擬對戰表

輪次	s	x	y	說明
1	20	3	1	己方先手， $\text{loseA} \rightarrow \text{loseB}$
2	16	2	3	對方失誤，進入 winA
3	11	3	-	$s=8$ 安全局面，進入 winB
	s	y	x	己方成為相對後手
4	8	3	1	被動拿取 $m+1-y$
5	4	3	2，遊戲結束	

表一資料來源：研究者自行整理

如表一，對手在第二輪己方拿完後，發生失誤破壞自己的安全局面。因此己方玩家在第三輪拿取三顆將局面控制為己方的安全局面 winB 函式，進入，並在第四輪成為相對後手，被動拿取 $m+1-x$ 顆棋子直至拿完。

此程式可證明 winB 函式為安全局面，也進一步佐證安全局面在完全資訊賽局中的預測能力，當把對方逼進安全局面時便可掌握賽場節奏，用後手策略(winB)拿下勝利。

三、不完全資訊賽局應用

雖然巴什博弈為策略型的遊戲，但由於有安全局面的存在，因此只要掌握完全資訊，遊戲便大大降低趣味性與挑戰性。為了增加遊戲的競賽性，研究者將完全資訊中的拿取上限設為不完全資訊，使遊戲增加更多變數。此款變形的巴什博弈，除了策略的實施，還需運氣的成分，玩家才有機率贏下比賽。

（一）重新制定規則

新規則與傳統巴什博弈相似，唯一規則的改動在於將原本固定的拿取上限，改由抽卡決定上限。卡牌種類共有兩種，分別是「一次拿取 1~3 顆」或「一次拿取 2~4 顆」。每回合雙方玩家開局前都可各別抽取一張卡牌，並依照卡牌所示範圍內拿牌。由於有一次只能拿 2~4 顆的限制，因此若場上僅剩一顆棋子時玩家又恰巧抽到「拿取 2~4 顆」此卡牌，則該局視為平手，結束遊戲。將規則「固定上限」改為「抽卡決定」，引入了隨機性，使比賽除了依賴原本的策略，也須推測對方資訊、計算機率等挑戰，使遊戲變得豐富刺激。

（二）機率計算與最佳策略

本研究以樹狀圖的方式，分析在不完全資訊下的最佳拿取策略。計算剩餘數量下，分別抽到不同卡牌後，何種拿法的勝率最大。若有平手的機率出現，則以「勝率與和局率總和最大」作為最佳策略，而非單看勝率作為參考。

1、機率計算

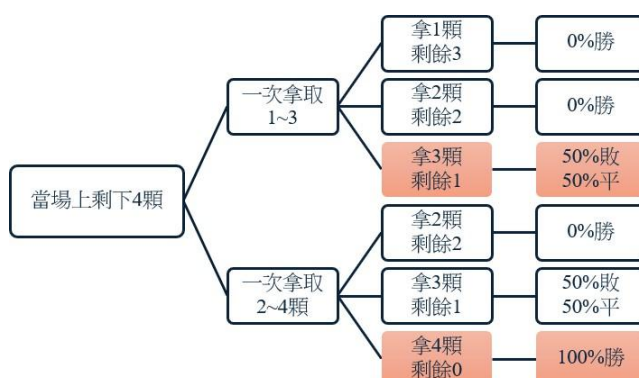
圖十：場上剩餘一顆時的最佳策略與機率計算樹狀圖



圖十資料來源：研究者自行繪製

如圖十所示，當場上剩下一顆時，若可拿 1~3 顆，則直接拿取 1 顆必勝。若可拿 2~4 顆，則遊戲平手。由此可之，當場上剩下一顆時，勝率及平手率皆為 50%。

圖十一：場上剩餘四顆時的最佳策略與機率計算樹狀圖



圖十一資料來源：研究者自行繪製

又如圖十一所示，當場上剩下四顆時，若可拿 1~3 顆，則拿取 3 顆時，根據圖十的數據可知，當場上剩下 1 顆，對方的勝率為 50%，亦代表己方的敗率為 50%、和局率 25%，依據「勝率與平手率總和最大」，因此拿三顆為最佳策略。若可拿 2~4 顆，則可將 4 顆拿完贏下遊戲。由此可之，當場上剩下 4 顆時，總勝率為 50%，總平率為 25%，總敗率為 25%。

2、最佳策略

(1) 同餘於模五最佳策略相同

表二：剩餘 10 顆內最佳策略及 25 顆內勝率機率（四捨五入至第一位）

剩餘總數(s)	1		2		3		4		5	
拿取上限	1~3	2~4	1~3	2~4	1~3	2~4	1~3	2~4	1~3	2~4
最適策略 (拿取 x 顆)	1	—	2	2	3	3	3	4	1	4
勝場率(%)	100	0	100	100	100	100	0	100	25	0
和局率(%)	0	100	0	0	0	0	50	0	25	50
敗場率(%)	0	0	0	0	0	0	50	0	50	50
總勝率(%)	50		100		100		50		12.5	
總和率(%)	50		0		0		25		37.5	
總敗率(%)	0		0		0		25		50	
剩餘總數(s)	6		7		8		9		10	
拿取上限	1~3	2~4	1~3	2~4	1~3	2~4	1~3	2~4	1~3	2~4
最適策略 (拿取 x 顆)	1	2	2	2	3	3	3	4	1	4
勝場率(%)	50	25	50	50	50	50	31.25	50	25	31.25
和局率(%)	37.5	25	37.5	37.5	37.5	37.5	31.25	37.5	34.375	31.25
敗場率(%)	12.5	50	12.5	12.5	12.5	12.5	37.5	12.5	40.625	37.5
總勝率(%)	37.5		50		50		40.6		28.2	
總平率(%)	31.3		37.5		37.5		34.4		39.1	
總敗率(%)	31.3		12.5		12.5		25		32.8	
剩餘總數(s)	11		12		13		14		15	
總勝率(%)	28.9		32.8		32.8		33.5		31.2	
總平率(%)	36.8		39.1		39.1		38		37.9	
總敗率(%)	34.3		28.1		28.1		28.5		30.9	
剩餘總數(s)	16		17		18		19		20	
總勝率(%)	29.5		31.5		32		32		31.2	
總平率(%)	36.8		38.2		38.5		38.5		38.3	
總敗率(%)	34.3		30.3		29.5		29.5		30.5	
剩餘總數(s)	21		22		23		24		25	
總勝率(%)	30.4		31		31.4		31.4		31.1	
總平率(%)	38.2		38.2		38.2		38.2		38.2	
總敗率(%)	31.4		30.8		30.4		30.4		30.7	

表二資料來源：研究者自行整理

根據表二，可以發現當 $s_1 \equiv s_2 \pmod{5}$ 相同時，其最佳策略皆分別相同。如：當剩餘總數 $4 \equiv 9 \pmod{5}$ ，當總數為 4 或 9 時，拿取上限為 1~3，則最佳策略為拿取 3 顆，若拿取上限為 2~4，則最佳策略皆為拿取 4 顆。

(2) 機率趨近穩定

數據顯示，當剩餘總數超過 10 顆後，其總勝率、和局率及敗場率逐漸趨近於穩定，分別是 31%、38%及 30%，比例並不會因總數增加而有大幅變動，因此在這種玩法下，當總數剩 10 顆時即是整個比賽的關鍵轉折點。

(3) 反向思考策略位置

部分剩餘總數雖然勝率低，卻可將其作為策略留給對手。例如當總數剩下 5 顆時，雖勝率僅有 12.5%，但若能將 5 顆留給對手，則己方的勝率將會提高。因此在當總數剩餘 6~9 顆時都可發現，最佳策略皆是將總數控制為 5 顆留給對手，進而佔據優勢。

本研究以傳統巴什博弈為基礎，透過改變「拿取上限」為隨機抽卡的方式，將完全資訊賽局轉化為不完全資訊賽局，為遊戲引入了機率與運氣的變數。

透過機率計算，統整並分析出最佳策略。可以發現同餘於模五最佳策略會相同，且當剩餘棋子數超過 10 顆時，勝率、和局率及敗率趨於穩定，顯示遊戲當剩餘十顆時進入策略關鍵轉折點。此外，部分局面雖勝率較低，卻能透過利用反向思考將其不利情況留給對手進而使己方玩家佔據優勢。

伍、研究結論與建議

一、 結論

巴什博弈是一個策略型的遊戲，並且有安全局面 $s = k(m + 1)$ 的存在，因此只要玩家掌握完全資訊，便能判斷勝負。由此可歸納出判斷勝負的公式：當 $s \% (m + 1) = 0$ ，後手勝。後手只需使局面維持在 $s = k(m + 1)$ ；當 $s \% (m + 1) \neq 0$ 時，先手勝。先手先操作使剩餘數量變成 $s = k(m + 1)$ 。

更可透過程式模擬後，證明安全局面的存在以及該遊戲的預測性，當對方玩家被逼進後手勝(winB)的函式中，則安全局面形成，只要雙方不失誤，則勝負已定。「後手勝」並不是真的後手才會勝，而是指讓對手陷入先手的角色，成為主動破壞安全局面的一方，而自己成為「相對後手」，就能確保拿下遊戲的勝利。

然而，正因巴什博弈為完全資訊賽局，對於理解其原理規則的玩家來說，趣味性也隨之降低。為了提升遊戲的變化與挑戰，筆者將巴什博弈中的拿取上限設計為不完全資訊，以抽卡方式增加遊戲變數，並算出最佳策略與機率。發現當總數同餘於模五時，最佳策略皆相同；其次當總數越大，勝率的數值也趨近於穩定，因此當總數剩下 10 顆時為整個比賽的關鍵點；最後，即使部份剩餘總數勝率低，也能作為戰術安排，刻意留給對方以提升己方的勝率。

二、建議

本研究雖成功以 Python 程式模擬巴什博弈對戰及改良規則並分析其最佳策略，但目前僅完成後端程式，尚未設計完整的圖形化介面，無法讓一般玩家直接體驗遊戲。建議未來可將本研究成果實體化，製作成具有視覺介面的互動遊戲或卡牌實體桌遊，讓玩家能更直觀地體驗策略與機率變化，提升研究的應用價值與趣味性。

此外，研究結果顯示，即使將規則改良成不完全資訊戰局後，增加了遊戲的不確定性，然而遊戲關鍵轉折主要集中在戰局僅剩十顆棋子時，代表在總數較多的情況下，前期回合幾乎不影響最終勝負，導致整體遊戲節奏單調、趣味性不足。建議未來可進一步調整抽卡規則或增加卡牌種類，如改變拿取範圍、增設特殊效果卡等，提升整體的刺激度與可玩性。

陸、參考文獻

Leetcode（無日期）。292. Nim Game。2025 年 8 月 20 日，取自

<https://leetcode.com/problems/nim-game>

Bouton, C.L. (1901). Nim, A Game with a Complete Mathematical Theory. *The Annals of Mathematics, 2nd Ser., Vol. 3, No. 1/4. (1901 - 1902), pp. 35-39.*

ACM Book（無日期）。巴什博弈。2025 年 10 月 6 日，取自 https://hrbust-acm-team.gitbooks.io/acm-book/content/game_theory/ba_shi_bo_yi.html

葉俊廷（2009）。不完全資訊賽局 - 蜜月橋牌之研究。國立臺灣師範大學資訊工程研究所：碩士論文。

李松沛（2020）。使用深度強化學習於拱豬撲克牌遊戲。逢甲大學資訊電機工程碩士在職專班：碩士論文。

ByteGather（2021 年 01 月 07 日）。博弈專題入門總結（Nim 巴什 SG 等證明+例題）[部落格文章]。 <https://fangkaipeng.com/?p=111>

AWS（無日期）。什麼是 Python？。2025 年 10 月 6 日，取自 <https://aws.amazon.com/tw/what-is/python/>