

# 中華民國第 56 屆中小學科學展覽會

## 作品說明書

---

國中組 生活與應用科學科

佳作

030802

2048 遊戲與人工智慧的分析

學校名稱：新竹市立培英國民中學

作者：  國二 樂峻佑  國二 陳彥廷  國二 劉彥承	指導老師：  王偉斌  江儒源
---	-----------------------------

關鍵詞：權重、人工智慧、比例關係

## 摘要

在 2048 人工智慧遊戲中，平滑度與單調性的權重為主要影響電腦判斷的變數，因此本研究主要針對這兩個變數的權重進行實驗並探討。並且在遊戲中步數愈多，愈容易達到更高的數字，在研究當中我們用遊戲移動步數的多寡來判斷當局的好壞。



在深度 1 的情況下「改進程式碼」，在研究結果中發現主要變數比值為 1 的同比例放大縮小效果都不錯，在 90 組數據中有  $\frac{1}{3}$  有到達 1024，甚至可以到達 2048；從步數來看，原程式平均步數為 332.8 步，而修改後的權重平均步數高達 561.8 步。

最後，我們利用相同的權重，並且將深度改回原設定。比較後發現，經過我們修改後的權重，達到 2048 與 4096 的獲勝率皆比原程式高。

## 壹、 研究動機

2048 是一種益智遊戲，不管在電腦或手機行動裝置都有這款遊戲的存在。我們玩了一陣子，埋下了日後想要研究的伏筆。後來，我們意外在“昌爸數學遊戲網站”中看到 2048 遊戲具有 2 種 AI 功能分為 Auto run 和 Get Hint 兩種，兩種功能類型如表 1-1，發現它可以給予玩家較正確的指示，我們使用這 2 種 AI 功能可以達到 2048，快要失敗時，也可以使用這 2 種功能使我們脫離困境。因此本篇研究以人工智慧為方向的實驗，期望能夠優化 AI 功能。

(表 1-1)

特點 2048-AI功能類型	相同點	相異點	
		完成時間	形式
1. 	皆為2048-AI功能，且成功機率相同，幾乎都可以達到2048。	較快	全自動 
2. 		較慢	半自動半手動 

## 貳、 研究目的

- 一、探討加入指令碼後，當局版面與四個方位四個變數的數據關係。
- 二、觀察權重對陣亡步數的影響。
- 三、利用數學軟體證明權重對陣亡步數的影響。
- 四、利用比例關係探究較佳權重對陣亡步數的影響。
- 五、在深度++的情況下，探究權重對獲勝率的影響。

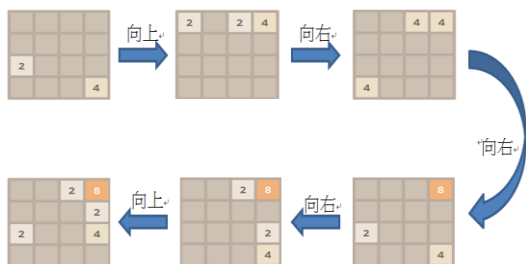
## 參、 研究設備及器材

- 一、2048-AI 程式檔：JavaScript 檔，內容為 2048-AI 的運作原理。
- 二、Excel：統計數據、製作成折線圖、蒐集數據。
- 三、Notepad ++：修改 2048AI 程式的軟體。
- 四、JavaScript 參考書、網路資源：幫助我們理解 2048-AI 程式檔。
- 五、GeoGebra：驗證實驗結果的數學軟體。

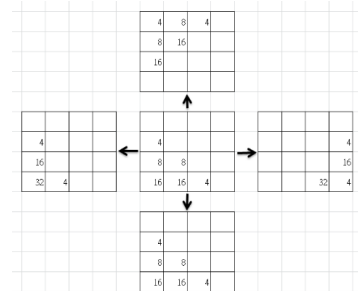
## 肆、 研究過程與方法

### 一、「2048」遊戲簡介

2048 是一個單人線上益智遊戲，遊戲任務是使用方向鍵使方塊上下左右的移動，而兩個相同數字的方塊則會合併成一個高一次方的方塊，每次移動會多出現一個 2 或 4 的方塊。而遊戲最終的目的是讓方塊合併成 2048，即成功。



(圖 4-1 遊戲中移動的過程)



(圖 4-2 遊戲中 4 個方向的示意圖)

## 二、人工智慧運作原理與 2048 的關係

「電腦遊戲程式設計」一書中表示遊戲中人工智慧與人類大腦的機制類似，一個典型的人工智慧系統一般有 4 部分：感知輸入、記憶儲存、分析推理、決策行為輸出(圖 4-3)。當然，一些簡單的人工智慧系統可能會將其中的一些設定元件，如記憶儲存和分析推理進行合併和簡化。不過，以下這個結構圖仍然適合大部分人工智慧的系統。



(圖 4-3 人工智慧系統結構圖)

在 2015 年 10 月，AlphaGo 成為第一個無需讓子即可在棋盤上擊敗圍棋職業棋士的電腦圍棋程式。而它其中一部分運用的即是蒙地卡羅樹狀搜尋法，AlphaGo 不斷地模擬對局(隨機的)，透過採樣，選擇最優的行動。而本研究中的人工智慧是利用  $\alpha$ - $\beta$  剪枝法，如表 4-1 所示。

(表 4-1 蒙地卡羅樹狀搜尋法與  $\alpha$ - $\beta$  剪枝法的運用及運作方法)

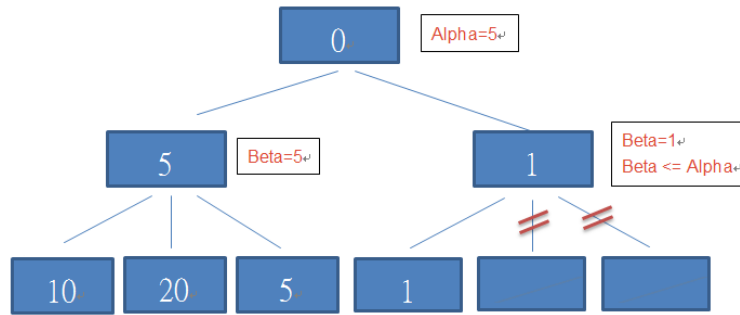
	AlphaGo	2048(本研究的遊戲)
運用方法	蒙地卡羅樹狀搜尋法	$\alpha$ - $\beta$ 剪枝法
運作原理	隨機取樣	考慮大於 $\alpha$ 或小於 $\beta$ 的值

以下將介紹兩種運作方法：

### 1. $\alpha$ - $\beta$ 剪枝法：

這個方法不需考慮所有情況，因此大幅減少需要考慮的數量。 $\alpha$ - $\beta$  剪枝加入了兩個  $\alpha$ 、 $\beta$  兩個紀錄值。如圖 4-4 為  $\alpha$ - $\beta$  剪枝示意圖。我方在取最大值的時候，若發現了一個大於等於  $\beta$  的值，就不用再對其它分枝進行搜尋，這就是所謂的  $\beta$  剪枝。

敵方在取最小值的時候，若發現了一個小於等於  $\alpha$  的值，也不用再對其它分枝進行搜尋，這就是所謂的  $\alpha$  剪枝。

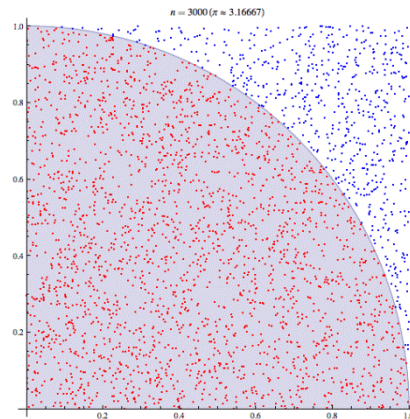


(圖 4-4  $\alpha$ - $\beta$  剪枝示意圖)

## 2. 蒙地卡羅樹狀搜尋法

蒙地卡羅樹狀搜尋法是搜尋最大勝率的一步，而不是最好的一步。這種方法是一種大數法則的方法，使用隨機取樣，而不需把所有的可能都估算考慮到。當取樣的次數越多，其所得出的平均值將趨近於理論值。

這個方式應用在圍棋時，會隨機下子直到盤末，由此依據各點模擬的結果，來對於盤面進行優劣的評估，從中選出最好的下子方式。



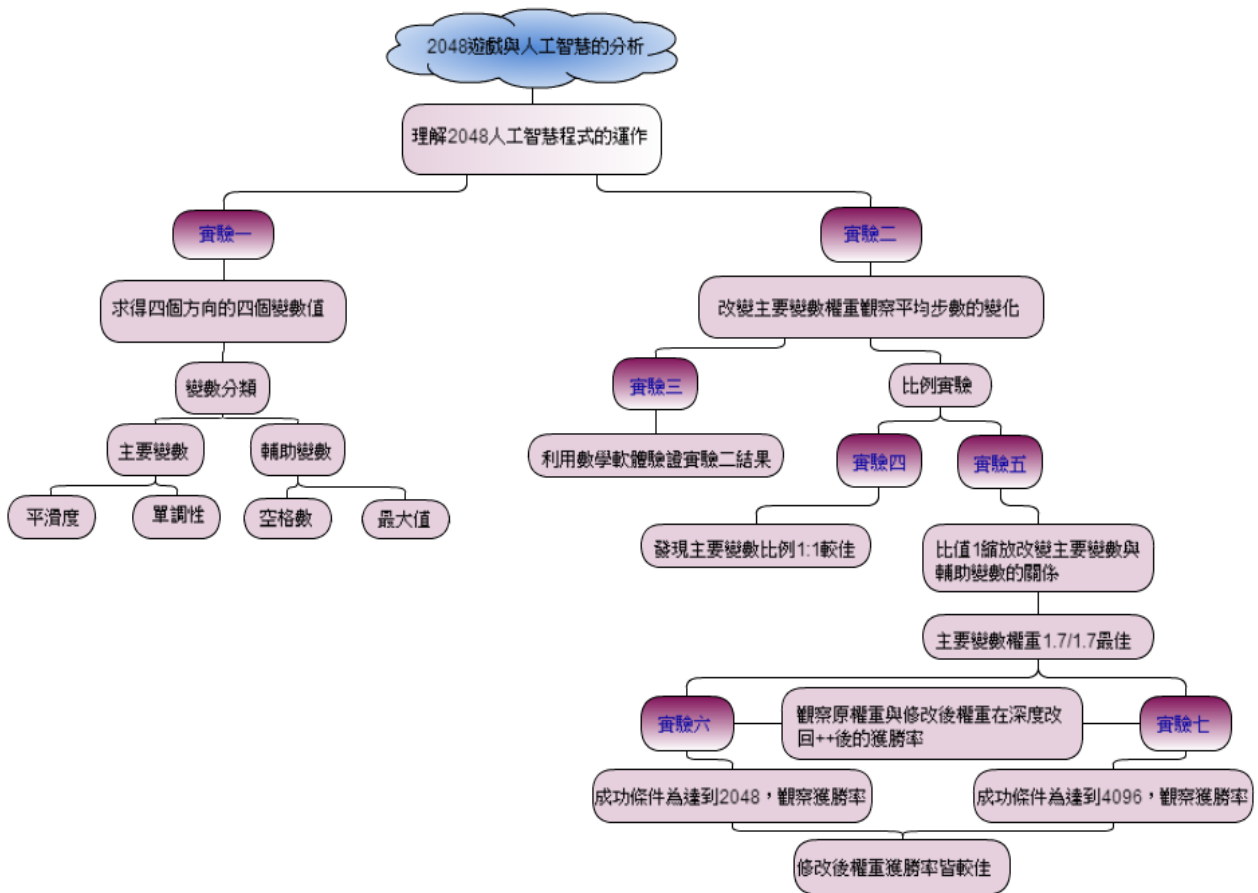
(圖 4-5 蒙地卡羅樹狀搜尋法估算 $\pi$ 值示意圖)

讓電腦每次隨機生成兩個 0 到 1 之間的兩個數，以這兩個實數為橫縱座標的點是否在單位圓內。生成一系列隨機點，統計單位圓內的點數與總點數，(圓面積：正方形面積之比= $\pi$ ：4)，當隨機點取得越多時，其結果越接近於圓周率。(節錄自維基百科／蒙地卡羅數狀搜尋法提供)

## 三、研究方法

首先我們想了解 2048-AI 程式檔的運作方式以及判斷下一步的方式。我們觀察許多次遊戲的進行，發現程式有時明明就有更好的方向可以選擇，卻選擇了一個對往後局面不利的方向，為了清楚了解這個選擇的原因，我們修改程式使其讓每走一步，就會顯示出四個變數的值，我們從數據中發現了一些蛛絲馬

跡，於是展開後續的研究。在接下來的實驗中，我們用遊戲移動到陣亡時所移動的步數(移動次數)來評斷那一局的好壞，以作為實驗觀察的數據。



(圖 4-6 研究架構)

## 了解程式檔

在我們解讀這兩個程式檔後，可知道總值由 4 個變數乘上權重加總而來，而這 4 個變數意義如下表所示，並且這些變數如果對當下局面有利，變數值就是正的，如果對局面不利，那值就是負的。

(表 4-2 變數功用表)

名稱	中文	意義
smoothness	平滑度	每個方塊與相鄰方塊數值的差
monotonicity2	單調性	方塊由左而右，從上到下，均遵從遞增或遞減
Emptycells	空格數	空格子越少越不利，所以空下格子的數量越多越好
maxValue	最大值	版面上最大方塊的值

如圖 4-7 為 AI 程式檔中的總值算式，總值是由四個變數值加總而來，以下將介紹四個變數值的算法。

```

AI.prototype.eval = function() {
    var emptyCells = this.grid.avAllableCells().length;

    var smoothWeight = 0.1,
        //monoWeight = 0.0,
        //islandWeight = 0.0,
        mono2Weight = 1.0,
        emptyWeight = 2.7,
        maxWeight = 1.0;

    return this.grid.smoothness() * smoothWeight
        // - this.grid.islands() * islandWeight
        // - this.grid.monotonicity() * monoWeight
        + this.grid.monotonicity2() * mono2Weight
        + Math.log(emptyCells) * emptyWeight
        + this.grid.maxValue() * maxWeight
};

```

(圖 4-7 2048-AI 程式檔中的總值算式)

(1)Smoothness 平滑度算法：

首先以左上角那格(1, 1)為基準點，算出基準點右邊格子與基準點下面格子以 2 為底的二進制對數的差(必須 $\geq 0$ ，因此都加上絕對值)。以此類推，每格皆完成這個動作後，會得到每個數字與相鄰數字的次方差，再將這些值相加。

以這個變數來說：此變數的功用就是讓相鄰次方差縮小。因此將每個數字與相鄰數字的次方差相加後是正數，就表示這個局面相鄰數字的次方差很大(如圖 4-8)，也就不容易合併，又為了配合以上所解釋的概念，我們必須將最後的平滑度變數值調整為負數來正確影響總值，也就是在加上負號前，值愈大愈不好。

		X座標			
		1	2	3	4
Y 座 標	1	2	2	4	8
	2			8	16
	3		2	2	64
	4				8

(圖 4-8 舉例版面)

依照平滑度計算方法，我們可以算出例圖 4-8 的平滑度值=-17。

$$\text{平滑度算法公式} = - [ |2, 1) - (1, 1)| + |(1, 2) - (1, 1)| + |(3, 1) - (2, 1)| + |(2, 2) - (2, 1)| + |(4, 1) - (3, 1)| + |(3, 2) - (3, 1)| + |(4, 2) - (4, 1)| + |(2, 2) - (1, 2)| + |(1, 3) - (1, 2)| + |(3, 2) - (2, 2)| + |(2, 3) - (2, 2)| + |(4, 2) - (3, 2)| + |(3, 3) - (3, 2)| + |(4, 3) - (4, 2)| + |(2, 3) - (1, 3)| + |(1, 4) - (1, 3)| + |(3, 3) - (2, 3)| + |(2, 4) - (2, 3)| + |(4, 3) - (3, 3)| + |(3, 4) - (3, 3)| + |(4, 4) - (4, 3)| ] = -17$$

(2) Monotonicity 2 單調性算法

首先，將單調性算法分成了兩大部分：上下方塊次方差以及左右方塊次方差。Total 表示「上下方塊次方差」以及「左右方塊次方差」的值。

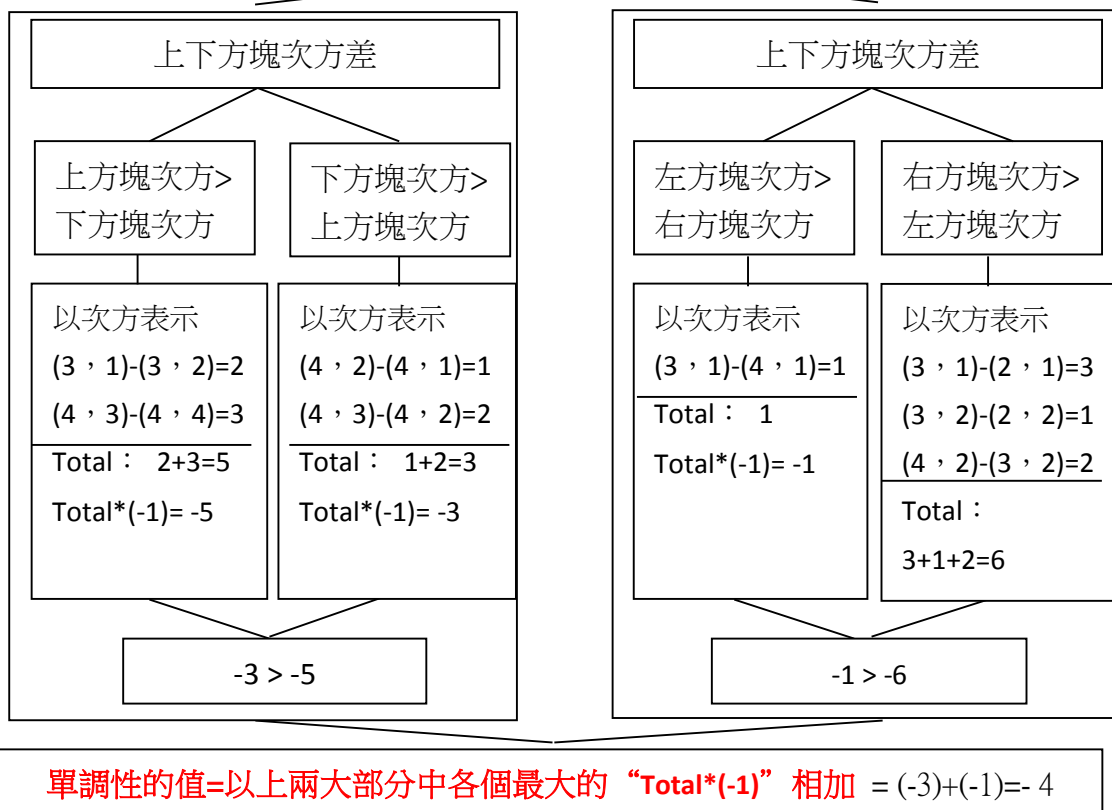
再來，「上下方塊次方差」以及「左右方塊次方差」之下，又可再分成「上方塊>下方塊」、「下方塊>上方塊」、「左方塊>右方塊」、「右方塊>左方塊」等 4 小部分。然後再進行方塊次方的相減，會有 4 個值。

最後，在「上下方塊次方差」以及「左右方塊次方差」中，各選出 1 個最大的值並相加，及當局那個方位之單調性數值。例如，我們用圖 4-9 的版面來計算單調性值(如圖 4-10)。

2	2	16	8
	2	4	16
			64
			8

單調性的值

(圖 4-9 舉例版面)



(圖 4-10 單調性值之算法)



(3)Emptycells 空格數算法

本程式中空格數算法須要用到數學概念「log」。數 x 的對數函數式子寫為  $y=\log_{\beta}X$ ，稱為以  $\beta$  為底的對數，舉例來說  $\log_3 81$ ，因為 3 的四次方為 81，所以  $\log_3 81=4$ 。而在  $y=\log_{\beta}X$  中， $\beta$  叫做底數，表 4-3 列出 2、e、10 為底數常用的領域。

(表 4-3 對數函數種類)

底數	名稱	ISO 表示法	適用領域
2	二進制對數	lbX	計算機科學、資訊理論、數學
e	自然對數	lnX	程式語言
10	十進制對數	lgX	工程學、生物學、天文學

空格數算法是將「空下的格子數」，以底數 e 為底所算出來的，也就是「空下的格子數」的自然對數。

設 X 為現在空下格子的數量，則空格數= $\ln(X)=\log_e(x)$

(附註：自然底數  $e=2.718281828459\dots\dots$ )

(表 4-4 各個空格子數的空格數值)

目前沒有數字的格子數量	以 e 為底的自然對數
1	$\log_e 1 = 0$
2	$\log_e 2 = 0.693147181$
3	$\log_e 3 = 1.098612289$
4	$\log_e 4 = 1.386294361$
5	$\log_e 5 = 1.609437912$
6	$\log_e 6 = 1.791759469$
7	$\log_e 7 = 1.945910149$
8	$\log_e 8 = 2.079441542$
9	$\log_e 9 = 2.197224577$
10	$\log_e 10 = 2.302585093$
11	$\log_e 11 = 2.397895273$
12	$\log_e 12 = 2.48490665$
13	$\log_e 13 = 2.564949357$
14	$\log_e 14 = 2.63905733$

由表 4-4 可以發現格子數量愈少，值之間的差距就愈大，因此我們推測作者可能認為版面愈少空格的時候，空格數這個變數較重要，但當版面很空的時候，空格數較不重要。例如：以圖 4-11 為例，版面上空下格子的數量為 2，由表 4-4 可知此版面的空格數值≈0.69。

8	32	64	512
4	8	16	256
2	4	8	32
		4	8

(圖 4-11 舉例版面)

(4)maxValue 最大值算法

最大值=現在版面上最大數字以二為底的二進制對數。例如：因為  $512=2^9$ ，所以圖 4-12 的最大值= $\log_2 512=9$ 。

8	32	64	512
4	8	16	256
2	4	8	32
		4	8

(圖 4-12 舉例版面)

(5)舉例總值算法

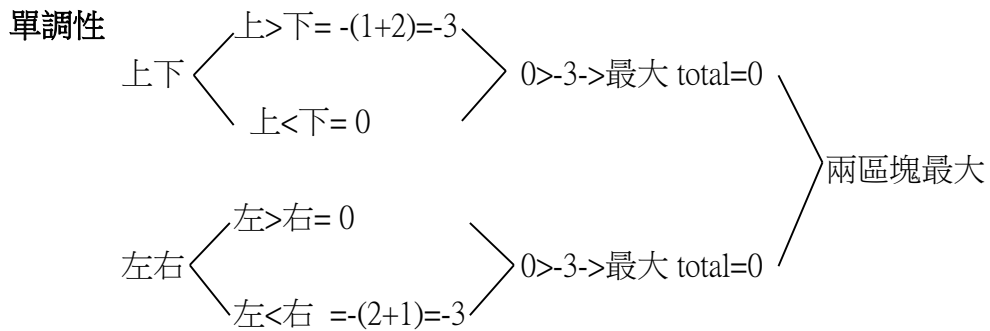
每次移動前，程式會推估出下一步(共 4 個方位)的版面，並且將這四個版面的 4 變數值得出總值。接下來 4 個方位中總值最大的方向就代表那個方向最有利。舉例來說，有一個版面如下：

		8	32
		4	16
		4	4

依照以上版面可推測出如果往上，下一步的版面會如下：

		8	32
		8	16
			4

平滑度 = -【|3-5|+|3-3|+|5-4|+|2-3|+|4-2|】 = -6



空格數：此版面將有 10 個空格， $\log_e 11 = 2.39$

最大值： $32=2^5$ ，所以此變數值為 5

方向判定：因此四個方向的總值如下列所示，

10.95(上)>10.6(下)>10.5(右)>10.3(左) 方向上的總值最大，故選擇往上。

- 求出往上的總值為  $(-5) \times 0.1 + 0 \times 1 + 2.39 \times 2.7 + 5 \times 1.0 = 10.95$
- 求出往下的總值為  $(-5) \times 0.1 + (-1) \times 1 + 2.3 \times 2.7 + 5 \times 1.0 = 10.6$
- 求出往左的總值為  $(-8) \times 0.1 + (-1) \times 1 + 2.3 \times 2.7 + 5 \times 1.0 = 10.3$
- 求出往右的總值為  $(-8) \times 0.1 + 0 \times 1 + 2.3 \times 2.7 + 5 \times 1.0 = 10.5$

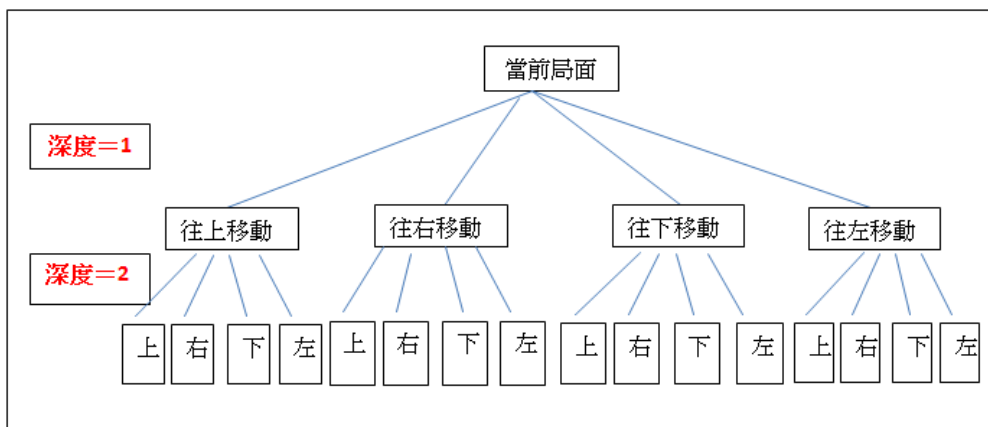
## 程式運作

(1)程式判斷下一步的方式

程式會把現在的版面分別往上、下、左、右移動一次並加上一個 2 或 4 在程式認為最差的地方，並以這個版面計算總值，程式會選擇總值最大的方向做為下一步的移動方向。

(2)程式碼的運作模式-深度(depth)

深度是這個程式運作的模式，深度也就是我們思考的深度。電腦會分別往 4 個方向移動一次，每一個方向分別往 4 個方向再移動一次，以此類推。因此深度越高，思考的範圍也愈多，遊戲的進行也會更順利，如圖 4-13 所示。



(圖 4-13 深度示意圖)

在原本的程式檔中，不是限制深度，而是限制程式的思考時間，在 0.1 秒內達到的最大深度中決定一個總值最高的方向。

我們決定把深度設成 1 有兩個原因：

(A) 由於 0.1 秒內能搜尋的深度會因為電腦的性能而不同，因此這個變因無法控制，所以我們決定把深度都設成深度 1。

(B) 由於程式本身已有辦法達到 2048，但我們猜測是因為時間 0.1 秒能搜尋的深度很高，因此我們把深度設成 1，發現平均只能合併到 256、512，與原本 0.1 秒能到達的 2048 還有很大的距離。因此我們想試試看有沒有辦法在深度=1 的情況下也有辦法到達 2048。

## 伍、 研究過程與探討

### 實驗一：探討四個變數並分類

一、實驗目的：求得四個方位四個變數的值並整理。

二、實驗器材：2048-AI 程式檔、EXCEL。

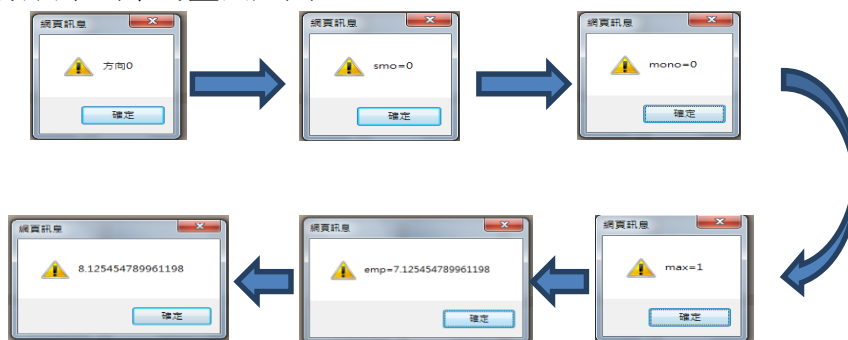
三、實驗步驟：

1. 我們從程式中增加部分指令檔(圖 5-1)，紀錄每一步 4 個方向、4 個變數以及總值，以便我們觀察：

```
alert("smo="+this.grid.smoothness() * smoothWeight);
alert("mono="+this.grid.monotonicity2() *
mono2Weight);
alert("emp="+Math.log(emptyCells) * emptyWeight);
alert("max="+this.grid.maxValue() * maxWeight);
alert(this.grid.smoothness() * smoothWeight
+ this.grid.monotonicity2() * mono2Weight
+ Math.log(emptyCells) * emptyWeight
+ this.grid.maxValue() * maxWeight);
alert("方向"+direction);
```

(圖 5-1)

2. 各變數表示出來的畫面如圖 5-2：



(圖 5-2)

3.不斷的重覆以上步驟，可獲得各方向的 4 個變數值，我們將數據蒐集起來並製成表格，再將移動前的局面繪製下來，如表 5-1：

(表 5-1)

	平滑度	單調	空格	MAX	總值				
上0	-1.5	-4	5.3	6	5.8	2	2	4	8
右1	-1.3	-4	5.6	6	6.3			8	16
下2	-1.5	-4	5.3	6	5.8		2	2	64
左3	-1.3	-15	5.6	6	-4.7				8
	平滑度	單調	空格	MAX	總值				
上0	-1.4	-4	5.6	6	6.2			8	8
右1	-1.4	-6	5.6	6	4.2	2		8	16
下2	-1.2	-5	5.6	6	5.4		2	4	64
左3	-1.6	-19	5.6	6	-9				8
	平滑度	單調	空格	MAX	總值				
上0	-1.4	-4	5.6	6	6.2	2	2	16	8
右1	-1.5	-4	5.6	6	6.1		2	4	16
下2	-1.2	-5	5.6	6	5.4				64
左3	-1.8	-11	5.6	6	-1.2				8

#### 四、實驗討論-討論步數：

由於存活的愈久，愈有機會到達 2048，因此在研究中我們使用步數來判斷該局遊戲的好壞。

透過討論步數，從我們已知的步數中推出該局最多可以合併到什麼數字，與實際合併到的數字差了多少。

由於一個 2048 是兩個 1024 所組成的，一個 1024 是兩個 512 所組成的……。

表 5-2 的縱軸是一個 2、4 方塊……，而橫軸是需要 2、4 方塊……的個數。最右邊是需要總步數。因為合併 4 需兩個 2，因此一次滑動就可讓 2 合併。由於一次滑動有可能造成兩個合併，以及並不是每次出現的數字都是 2，有可能會是 4，因此需要步數並不是固定的。我們做出表 5-2。

(表 5-2)

	2	4	8	16	32	64	128	256	512	1024	2048	所需步數
2	1											0
4	2	1										1
8	4	2	1									3
16	8	4	2	1								7
32	16	8	4	2	1							15
64	32	16	8	4	2	1						31
128	64	32	16	8	4	2	1					63
256	128	64	32	16	8	4	2	1				127
512	256	128	64	32	16	8	4	2	1			255
1024	512	256	128	64	32	16	8	4	2	1		511
2048	1024	512	256	128	64	32	16	8	4	2	1	1023

我們寫出公式以計算出版面的步數

設：當局步數為  $s$   
 $a_n$ =合併  $2^n$  所需的步數  
 $f$ =當局出現 4 的次數  
 $n_1, n_2, n_3 \dots$  為陣亡版面上的數字  
**步數公式：**  $s = (a_{n_1} + a_{n_2} + a_{n_3} + \dots) + 14 - f$

移動可分成三種情況：第一種是沒有造成任何合併，第二種是造成一個合併，第三種是造成兩個以上的合併。

遊戲一開始有 14 個空下的格子，而陣亡時沒有任何空下的格子，因此我們推測第一種合併與第三種合併相互抵銷後，總共還會有 14 步，是第一種情況的移動。括號內的為合併版面上的數字總共所需要的步數，這時就須用到上表格「合併方塊所需的步數」。

公式需要減掉  $p$  是因為合併一個 4 需要一步，而上表格「合併方塊所需的步數」都是假設一開始都出現 2，因此直接出現一個 4，步數就會少一步，故需要減掉遊戲出現 4 的次數。

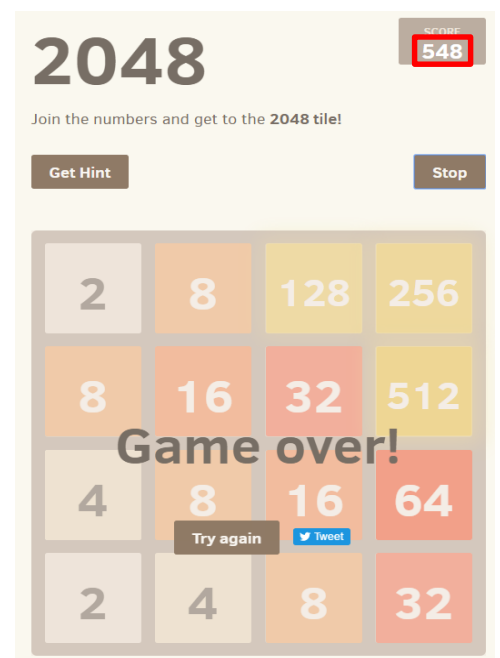
證明：

在此舉例的遊戲中已經讓遊戲每次移動不出現 4，只出現 2，因此  $f=0$ 。

帶入公式，得

$$(256 + 128 + 64 + 32 + 16 \times 2 + 8 \times 2 + 4 \times 4 + 2 \times 2) + 14 - 0 = 548$$

由此公式算出來的總步數與程式所顯示的步數相同(圖 5-3)，可證明我們所推出來的步數公式是正確的。



(圖 5-3 陣亡版面)

## 五、實驗討論-觀察變數優缺點：

雖然我們知道四個變數的算法，以及其各自的功用，但我們不知道它對遊戲有什麼優缺點，於是我們分別開啟各變數來進行遊戲，並觀察只開其中一個變數的「陣亡版面」，並找出變數的優點及缺點，並且把它分類。

### 1. 平滑度

以下是只開平滑度變數時的陣亡版面：

2	4	8	2
16	128	64	4
8	64	8	16
2	16	4	2

2	4	16	2
16	128	64	4
4	16	32	16
2	8	4	2

2	4	8	2
4	16	64	4
8	128	256	32
16	32	16	4

我們發現三圖都是以 1 個大數字為中心，依序往外遞減。

優點：照著順序遞減，只要有一個小數字合併，就能依序合併。

缺點：相鄰在大數字四邊的數字，有時會與大數字的次方相差太大，導致合併的機會降低。

### 2. 單調性

以下是只開單調性變數的陣亡版面

16	32	128	256
8	16	32	128
4	8	16	32
2	4	8	2

128	4	32	16
64	32	16	2
32	16	8	4
16	8	4	2

16	2	8	256
8	16	32	64
4	8	16	32
2	4	8	16

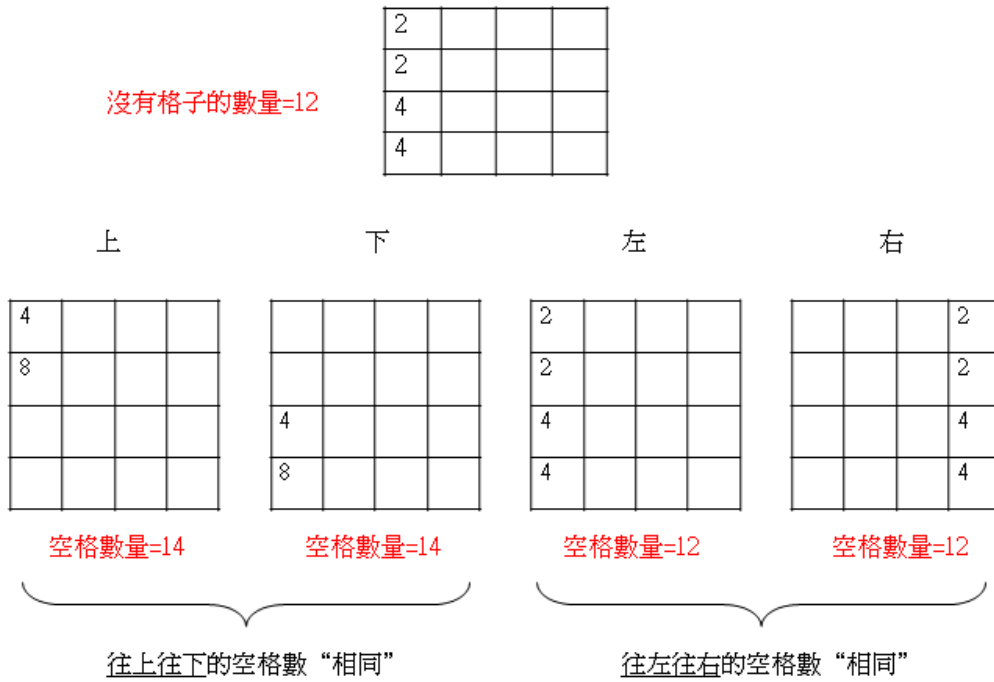
我們發現三圖都有一個大數字在角落，依序往另一方向遞減，不過有些例外，但還是以由大到小為主要趨勢。

優點：越大的數字越不容易合併，因此大數字在角落有助於遊戲。只要有 1 個小數字合併，就能依序合併很多大數字。

缺點：有時太過注重單調性，造成陣亡，例如上面的陣亡版面都是單調性很高的。

### 3. 空格數

由於同個版面往左往右，以及往上往下能剩餘的空格數量是一樣的，如圖 5-4。所以空格數往左右，或往上下的值會一樣，程式會因為寫法的關係依上、右、下、左的順序移動，所以當只開空格數的情況下，進行遊戲會變得無意義。



(圖 5-4 下一步的版面空格數量情形)



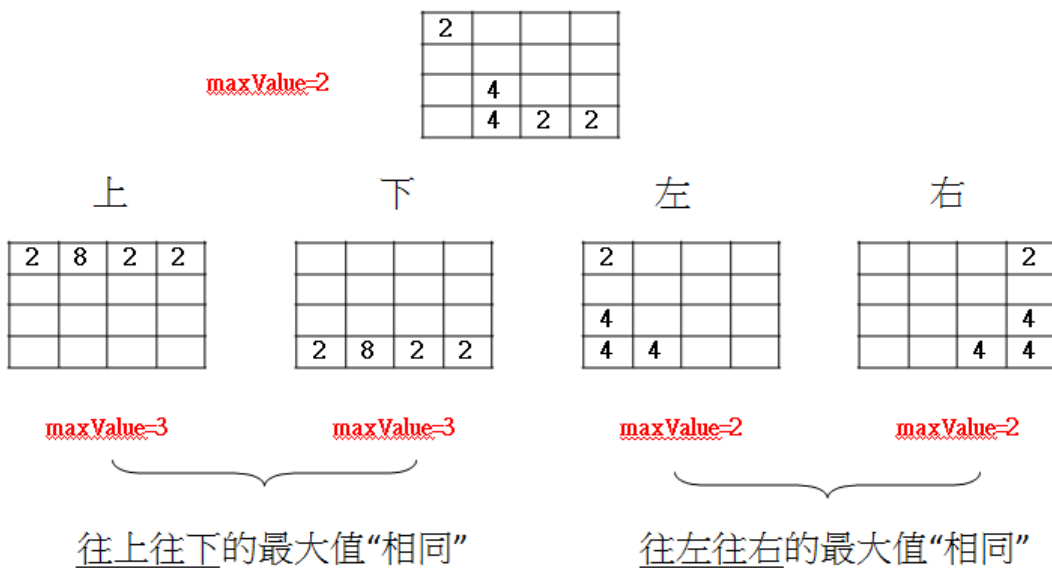
例如圖 5-5：由於「一個版面往左往右，以及往上往下能合併的數量是一樣的，所以空格數往左往右，或往上往下的值會一樣」，我們可以從以下 2 組數據觀察到這個現象。

	平滑度	單調	空格	MAX	總值	當局版面			
上0	-1.7	-4	5.3	6	5.6	2	2	4	8
右1	-1.4	-4	5.6	6	6.2			8	16
下2	-1.8	-6	5.3	6	3.5		2	2	64
左3	-1.3	-12	5.6	6	-1.7				8
	平滑度	單調	空格	MAX	總值	當局版面			
上0	-1.8	-4	4.8	6	5	8	8	16	8
右1	-2.4	-5	3.7	6	2.3		4	8	32
下2	1.5	-7	4.8	6	5.3			4	64
左3	-2.1	-16	3.7	6	-8.4	2			8

(圖 5-5 下一步的版面空格數量是一樣的)

#### 4. 最大值

由於最大值只有在合併最大方塊時才有用處，因此容易出現相同的值，而在總值都相等的情況下會因為程式寫法的關係而依著上、右、下、左的順序移動(與空格數的問題相同)，因此只開最大值也無意義：



(圖 5-6 下一步的最大值情形)

## 5. 變數分類

從數據中發現平滑度與單調性的值都有在變化，如圖紅框的部分；而空格數與最大值容易出現相同的值，如表 5-3 綠框的部分。

(表 5-3) 平滑度與單調性值的變化

方向	平滑度	單調性	空格數	最大值	總值				
上	-2.3	-4	4.3	6	4		16	16	8
右	-1.9	-7	5.3	6	2.4		4	8	32
下	-2.2	-9	4.3	6	-0.9	2	2	4	64
左	-1.9	-16	5.3	6	-6.6			2	8
方向	平滑度	單調性	空格數	最大值	總值				
上	-2.6	-4	3.7	6	3.1	2	16	16	8
右	-9.1	-4	3.7	6	-3.4	2	4	8	32
下	-2.2	-10	3.7	6	-2.5		2	4	64
左	-1.8	-16	3.7	6	-7			2	8

不容易出現相同值的變數在各種版面下都有作用，因此它是決定下一步方向的「主要變數」，而容易出現相同的值的變數只有在特定版面下才會有作用，因此它是決定下一步方向的「輔助變數」。

主要變數	輔助變數
平滑度	空格數
單調性	最大值

## 實驗二：改變主要變數權重，觀察步數的變化

### 一、實驗目的：

四個變數中單調性和平滑度兩個權重的改變，觀察平均步數的變化。

### 二、實驗說明：

此實驗是為了求得在各種權重情況下平滑度與單調性的平均步數。因為每個變數的重要性可能不同，其權重也就不同，因此，找出最佳的權重是我們一定要做的。

我們把滑動一次稱為一步，而愈多次的滑動愈有可能造成合併，更有機會達到更高次方，因此我們利用步數的多寡來判斷當局的好壞，將平滑度、單調性改過後各個權重的平均步數做成表格。

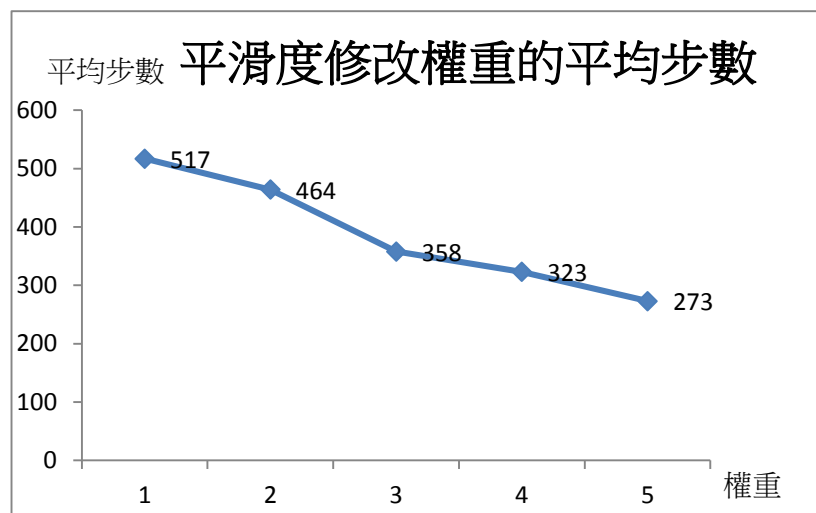
### 三、實驗器材：2048-AI 程式檔、EXCEL、加入的指令檔

#### 四、實驗步驟：

- (1)加入可顯示出移動步數的指令檔
- (2)執行
- (3)紀錄下每次的步數取平均並繪製成折線圖以便觀察

#### 五、實驗結果：

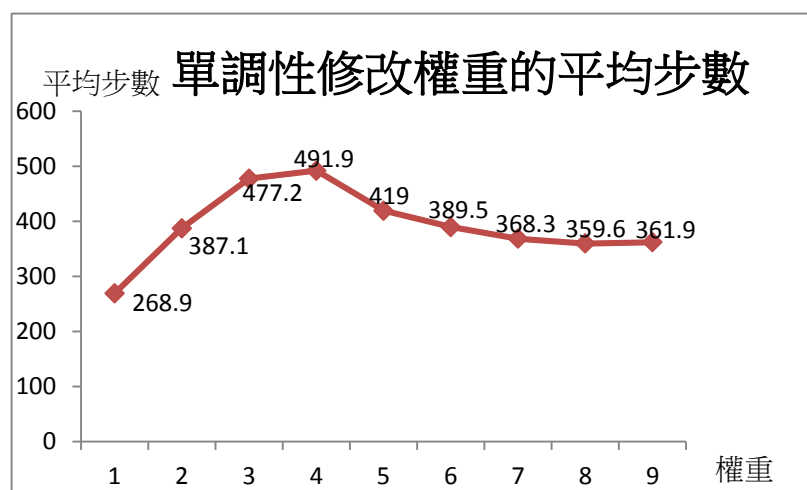
##### 1.改平滑度權重(如圖 5-5)：



(1)觀察圖 5-5，我們發現此圖成一個從低點到高点再到低點的圖，在  $x$  權重為 1 時，是  $y$  軸的最高點，平均步數達 517 步。

(2)發現步數愈來愈小，若繼續增加權重，步數也會越來越少。

##### 2.改單調性權重(如圖 5-6)：



(1)觀察圖 5-6)我們發現此圖成一個從低點到高点再到低點的圖，在 0.15 時，是  $y$  座標的最高點，平均步數達 491.9 步，比原本的權重好很多 (349.2 步)。

(2)由圖 5-6 也可以看到步數到後來相差不大。

3.將這兩個變數之權重改成單調性與平滑度最高的權重：我們在做完以上實驗後，我們將程式碼中單調性與平滑度的權重改為以上實驗最好的權重(1.0/0.1)，發現陣亡步數反而時減少了，因此將其各自的最佳權重合併是錯誤的。

## 實驗三：利用多項式回歸驗證實驗二結果

### 一、實驗目的：

改變單調性權重-利用數學軟體尋找平均步數區間最大值。

### 二、實驗說明：

由於我們在實驗二中只測試單調性 9 個權重，並沒有測試所有的權重，因此實驗三藉由多項式回歸的圖形找出區間的最大值(最佳權重)。我們發現可以用「多項式迴歸」來驗證我們在實驗二的結果，多項式迴歸目的在於建立那些變數之間的回歸方程式。**GeoGebra** 幫助我們找出區間內的最大值。

設  $x$  座標為單調性權重， $y$  座標為平均步數。因  $y$  座標是取平均步數是會有誤差，故我們利用 **GeoGebra** 幫助我們找出這些座標的回歸多項式，回歸多項式圖形可以驗證我們的假設，0.15 附近會有  $y$  座標的最高點。

### 三、實驗器材：一份 2048-AI 程式檔、GeoGebra

### 四、實驗步驟：(1)將實驗二所做出的數據以(x,y)呈現在直角坐標系上

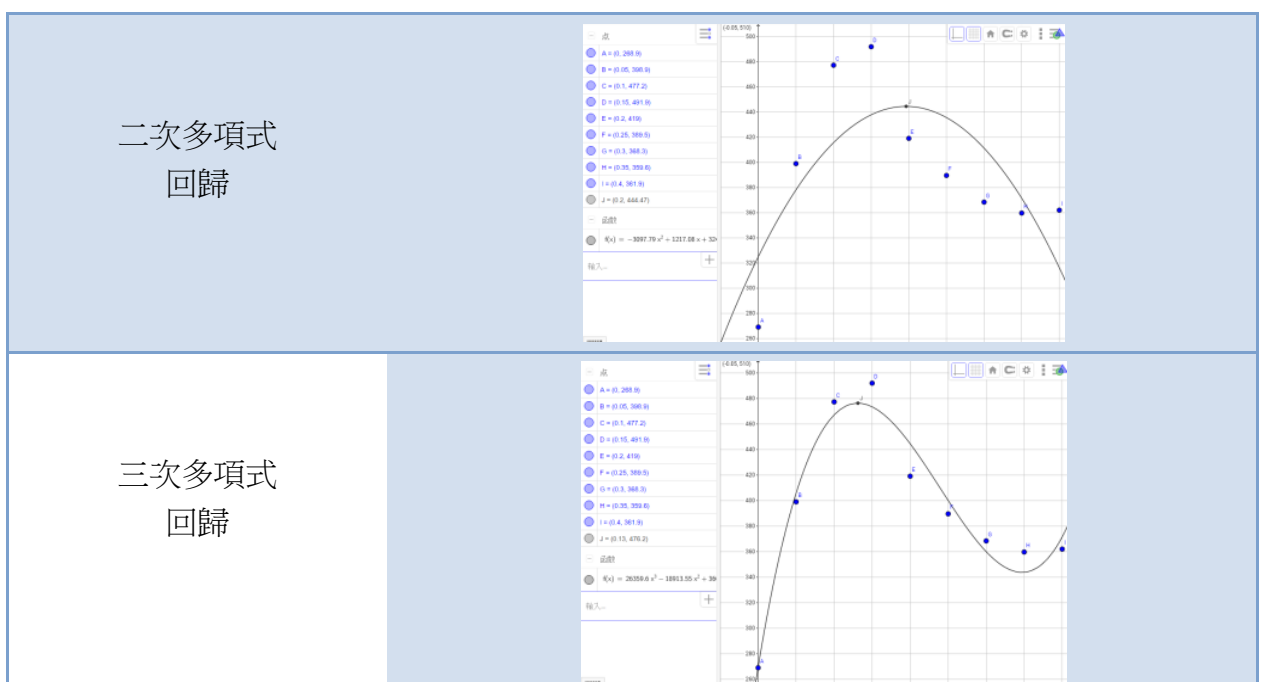
(2)做 2 到 8 次方的多項式回歸圖形

(3)尋找區間最大值

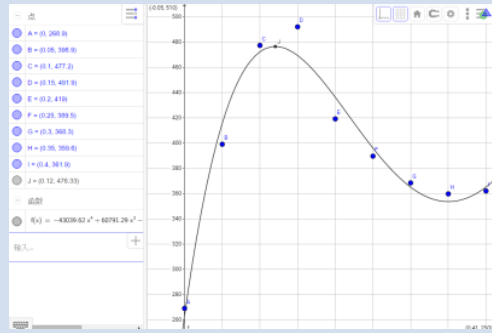
(4)紀錄

### 五、實驗結果：

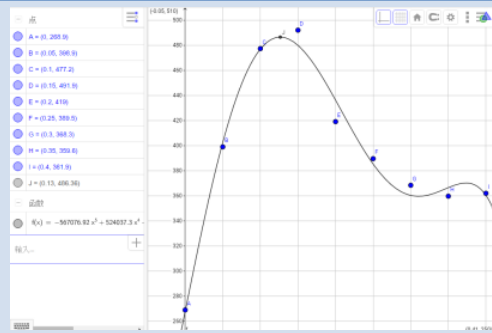
以下是 2 到 8 次方的多項式回歸圖形：



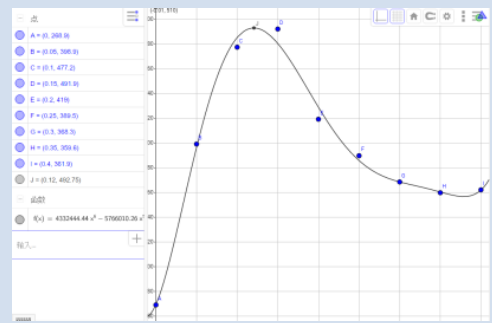
四次多項式  
回歸



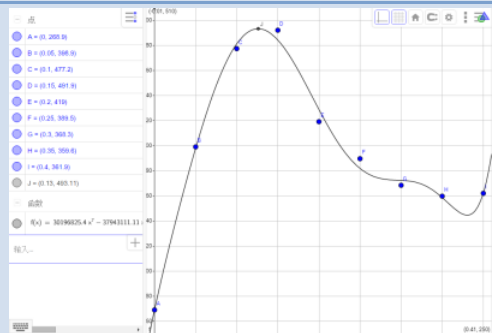
五次多項式  
回歸



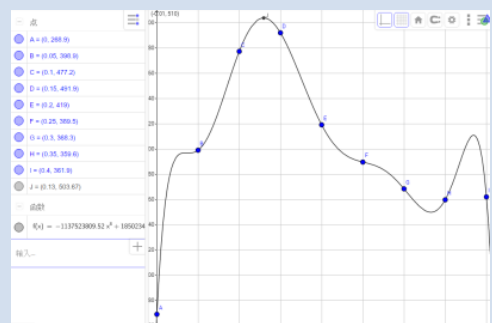
六次多項式  
回歸



七次多項式  
回歸



八次多項式  
回歸



我們將每次個圖形中最高陣亡步數以及其 xy 座標蒐集起來製成下表 5-4 並觀察：

(表 5-4 多項式回歸次數表)

多項式回歸次數	2 次方	3 次方	4 次方	5 次方	6 次方	7 次方	8 次方
X 座標	0.2	0.13	0.12	0.13	0.12	0.13	0.13
Y 座標	444	476	476	486	492	493	503

#### 六、實驗討論：

表 5-4 是 2~8 次多項式回歸在 0~0.4 這個區間內 y 座標最大時的 x 座標與 y 座標值。除了 2 次方以外，都與 0.15 相近(因為 2 次多項式回歸不夠接近所有的值)，跟我們預估單調性權重為 0.15 的結果相近，因此可以說明我把單調性改 0.15 是有助於此遊戲。

## 實驗四：利用比例觀念分析主要變數

#### 一、實驗目的：

觀察兩個主要變數的比例，找出較佳的比值。

#### 二、實驗說明：

在實驗二時，我們分別修改主要變數的權重，雖然研究結果發現單調性改 0.15 是有助於此遊戲，但是我們認為這樣的作法太過於土法煉鋼，因此在這裡我們用另一種方式來分析權重。主要變數的重要性是由兩個變數的權重所決定的，而權重的關係即為兩權重的比例，在實驗四我們以比值來表示兩權重比例。

#### 三、實驗器材：實驗二的數據

#### 四、實驗步驟：

(1)計算出實驗二數據的比值

(2)找出最高平均步數所對應到的比值

#### 五、實驗結果：

1.表 5-5 是固定單調性、改變平滑度兩主要變數比值以及平均步數：

平滑度權重	單調性權重	比值	平均步數
0.5	1	0.5	384
<b>1</b>	<b>1</b>	<b>1</b>	<b>517</b>
2	1	2	464
3	1	3	357.9
4	1	4	323
5	1	5	273

2.表 5-6 是固定平滑度、改變單調性兩主要變數比值以及平均步數：

平滑度權重	單調性權重	比值	平均步數
0.1	0	-	268.9
0.1	0.05	2	398.9
0.1	0.1	1	477.2
0.1	0.15	2/3	491.9
0.1	0.2	0.5	419
0.1	0.25	0.4	389.5
0.1	0.3	1/3	368.3
0.1	0.35	2/7	359.6
0.1	0.4	0.25	361.9

五、實驗討論：

1.數據中知道同比值但平均步數不同是因為主要變數與輔助變數的比例不同。

2.透過此次實驗，我們得知主要變數權重最佳比例為 1：1 的平均步數，下個實驗我們將會透過比值 1 改變主要變數的權重。

## 實驗五：利用比例觀念分析兩類變數

一、實驗目的：

以比例 1：1 改變主要變數的權重進行數據統計，尋找最佳權重。

二、實驗器材：一份 2048-AI 程式檔、EXCEL

三、實驗步驟：

- (1)以比值 1，討論主要變數的權重（0.1~2.0）
- (2)蒐集各組權重的平均步數
- (3)繪製折線圖
- (4)尋找折線圖中的趨勢
- (5)找出最佳權重

四、實驗結果：

權值	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2
1	499	263	437	500	618	673	267	202	203	399	363	319	463	515	265	957	497	610	338	348
2	297	278	516	569	478	469	529	340	216	410	400	509	268	318	384	990	260	489	256	270
3	354	415	866	497	573	638	658	365	181	301	330	542	806	316	385	482	406	537	351	341
4	388	602	854	444	425	530	593	468	364	552	363	648	532	790	288	407	613	232	439	414
5	396	547	532	1072	808	519	711	521	593	305	739	682	416	576	323	168	855	859	493	578
6	375	330	307	607	413	835	919	661	372	623	226	490	297	304	493	757	473	862	461	205
7	644	410	723	233	437	347	793	433	446	284	384	893	938	318	705	466	386	378	282	769
8	609	723	844	287	271	540	651	577	519	533	501	310	335	253	542	391	566	473	353	462
9	493	787	234	380	311	508	507	201	518	527	970	286	598	617	687	304	399	215	857	471
10	599	568	1023	407	357	580	329	317	424	704	402	466	1152	521	526	908	298	266	739	385
11	567	584	374	516	496	559	481	157	438	429	561	572	780	610	762	571	741	507	477	526
12	769	749	396	302	514	352	335	712	575	746	503	566	720	258	802	549	480	512	632	318
13	474	387	245	736	562	424	715	458	440	261	826	186	406	750	371	300	334	404	535	341
14	754	352	548	528	672	530	540	242	469	317	655	500	459	605	619	457	368	395	579	491
15	514	537	440	718	385	958	528	301	460	577	670	326	519	733	604	412	571	765	299	391
16	426	946	849	533	375	509	278	614	594	557	272	555	913	1031	490	588	535	262	1064	300
17	311	525	768	618	450	698	644	389	269	548	812	288	556	325	565	680	538	379	563	474
18	426	857	438	644	570	658	351	617	314	567	496	450	901	437	467	990	713	607	533	491
19	902	791	552	352	726	261	788	531	494	428	1016	528	538	657	381	392	682	607	578	256
20	548	500	602	512	927	544	474	789	673	447	527	101	490	796	298	653	771	242	166	295
21	326	538	696	555	665	493	349	780	544	593	819	623	426	534	535	264	554	865	444	496
22	363	470	600	350	527	506	750	321	805	619	760	327	889	462	614	367	539	709	329	509
23	535	629	566	897	317	862	825	196	762	626	483	360	495	493	497	495	499	941	221	657
24	473	565	708	505	85	343	326	416	550	267	831	408	1019	747	274	359	429	381	479	388
25	427	563	353	376	501	412	503	604	448	198	305	691	851	506	318	379	296	664	643	365
26	407	616	439	316	274	872	555	335	615	668	497	450	436	345	646	716	489	764	573	234
27	622	313	745	335	740	564	879	651	503	275	438	366	536	401	726	563	389	492	287	563
28	312	384	319	687	1068	812	488	386	464	629	577	233	185	518	419	779	298	356	709	302
29	472	572	295	699	769	281	898	567	495	617	196	493	293	349	307	689	773	824	391	779
30	590	509	292	403	463	467	594	663	583	681	603	712	260	494	417	363	621	476	354	828
31	343	499	582	500	522	371	1005	613	329	826	852	355	689	530	417	795	815	892	568	387
32	502	328	353	293	496	560	351	610	275	351	711	95	533	589	631	599	682	524	560	388
33	732	618	861	619	754	406	480	626	514	328	234	505	301	479	744	1003	566	1032	397	283
34	502	328	850	974	542	312	459	472	661	419	762	388	732	470	489	420	1043	462	1080	397
35	186	276	592	393	504	320	442	893	620	343	665	723	555	345	290	419	307	394	1181	272
36	347	709	466	342	603	689	630	530	970	238	357	385	516	662	371	599	528	539	378	335
37	295	486	422	392	725	233	987	365	639	521	416	783	482	661	534	378	556	720	505	541
38	527	463	553	523	649	826	354	327	309	307	875	530	489	692	535	570	821	305	543	380
39	353	561	497	535	288	772	696	326	301	535	419	561	312	361	700	463	618	542	491	563
40	472	544	255	558	541	255	490	274	394	640	661	532	667	499	411	504	524	550	575	772
41	670	326	335	485	408	748	529	684	189	411	464	366	204	556	408	406	1218	819	349	631
42	337	914	426	480	525	494	665	419	258	359	522	549	418	618	431	498	286	570	418	704
43	561	495	591	985	654	546	785	499	298	330	500	674	766	530	215	388	572	1050	350	844
44	239	433	487	297	357	691	332	393	582	421	527	682	921	246	489	509	611	478	316	487
45	647	453	1050	467	376	411	509	154	570	709	802	271	562	361	461	570	642	475	548	574
46	433	1011	395	511	347	581	484	500	279	589	586	952	337	315	220	439	753	516	1007	550
47	544	386	602	755	560	479	472	899	648	268	327	272	940	901	481	369	307	242	386	282
48	509	456	410	563	787	396	319	806	793	852	530	586	473	421	392	523	394	452	346	277
49	277	693	947	476	634	594	277	281	489	535	608	421	373	506	600	1014	512	544	605	533
50	687	735	527	836	577	217	796	452	753	468	739	717	272	421	561	631	553	580	902	441
51	695	602	765	772	930	634	848	907	569	442	265	372	343	432	563	933	343	987	270	440
52	178	385	553	560	295	392	441	711	537	403	505	376	356	540	277	351	944	369	828	761
53	193	509	452	386	557	805	487	497	283	851	448	995	818	292	348	455	577	603	367	437
54	264	462	271	556	1000	541	987	445	475	528	488	802	641	489	340	211	858	482	658	902
55	310	295	445	597	792	559	805	359	817	544	797	526	459	444	474	504	456	390	784	851
56	300	345	830	517	327	540	433	450	446	461	469	302	555	459	357	409	549	444	344	450
57	296	585	477	334	284	321	488	760	879	730	479	869	544	266	286	156	750	511	534	364
58	298	359	545	573	402	273	480	652	465	625	738	900	662	409	457	348	998	303	438	271
59	439	732	621	904	154	940	303	526	543	375	959	216	684	489	349	442	314	498	730	364
60	387	512	541	506	638	552	762	317	569	309	759	600	628	354	277	546	417	566	506	332
61	526	442	452	437	216	584	570	268	830	270	305	291	163	452	256	380	529	491	370	384
62	638	564	497	341	290	423	540	324	382	617	522	187	417	639	497	654	586	550	272	712
63	237	439	260	553	487	441	480	258	554	742	404	328	458	689	548	540	536	481	435	246
64	546	537	382	638	516	517	255	341	304	539	605	348	360	561	574	507	634	526	441	220
65	559	267	831	496	261	615	615	247	245	758	645	533	349	857	433	491	931	277	429	874
66	812	825	536	541	347	496	309	578	452	547	307	578	718	283	988	764	463	961	459	488
67	555	666	308	304	502	596	445	513	433	311	301	541	582	769	472	629	650	549	524	535
68	925	490	749	278	559	827	555	614	310	301	439	302	272	442	499	575	601	986	547	195
69	398	587	297	578	556	327	570	617	462	238	681	309	928	452	303	368	622	847	1064	561
70	213	844	280	259	384	297	260	461	513	537	405	622	282	441	920	349	649	567	886	326
71	1122	398	526	401	379	297	274	679	753	363	895</									



## 實驗六：驗證「利用比例觀念」的優化結果 1

### 一、實驗目的：

以權重 1.7 比對原程式的深度++證明公式優化成功。

### 二、實驗說明：

我們認為修改後的公式在改回深度++時，也能夠比原本的公式好，因此透由實驗五得出的最佳權重與原本權重執行，程式深度設為++蒐集數據，來觀察我們更改出的權重是否對遊戲較有利。

### 三、實驗器材：一份 2048-AI 程式檔、實驗五數據

### 四、實驗步驟：

(1)以平滑度單調性權重各為 1.7、1.7，以深度++做執行，蒐集 200 組數據

(2) 以平滑度單調性權重各為 0.1、1.0，以深度++執行，蒐集 200 組數據

### 五、實驗結果：

深度++	原程式權重(0.1/1.0)	修改後權重(1.7/1.7)
獲勝次數	163	170
失敗次數	37	30
200 次的獲勝率	81.5%	85%
兩者比較	較差	較好

### 六、實驗討論：

在修改 2048-AI 程式檔的權重之後，我們已成功使其在深度++的情況下使獲勝率大幅提升，修改程式後，我們依然限制時間為 0.1 秒，還是能夠比原程式碼效果好，以上可說明我們把權重改成 1.7/1.7 是對 2048-AI 公式有幫助的。

## 實驗七：驗證「利用比例觀念」的優化結果 2

### 一、實驗目的：

以主要變數權重 1.7 比對原程式權重，並將遊戲成功條件從 2048 改為 4096，觀察獲勝率。

### 二、實驗說明：

我們發現實驗六的結果兩者的獲勝率只差 3.5%，我們推測是成功條件不夠高，因此實驗七我們將成功條件改為 4096，並觀察獲勝率。

### 三、實驗步驟：

(1)將成功條件改為 4096

(2)以平滑度單調性權重各為 1.7、1.7，以深度++做執行

(3) 以平滑度單調性權重各為 0.1、1.0，以深度++執行

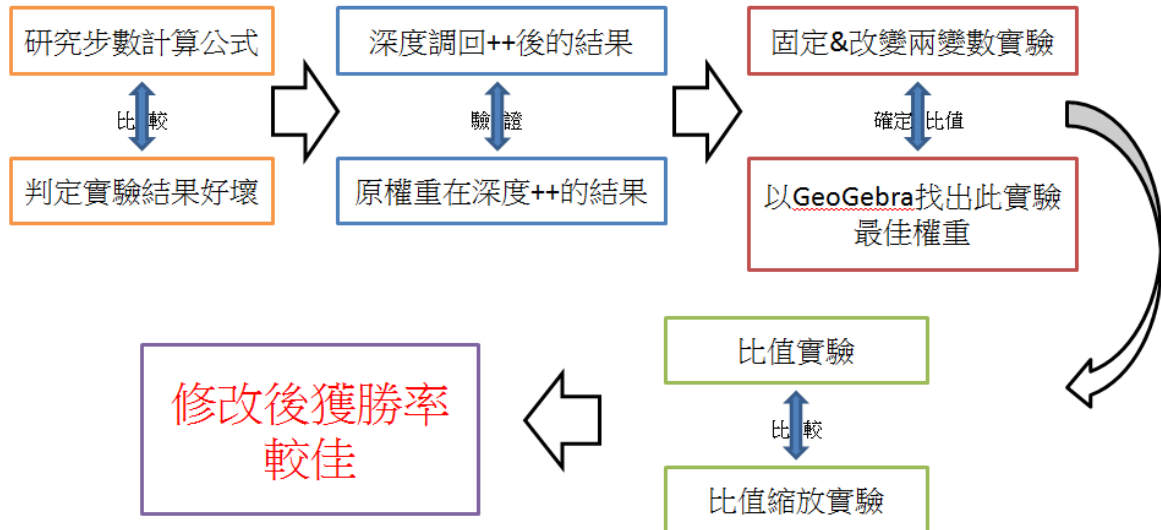
#### 四、實驗結果：

深度++	原權重(0.1/1.0)	修改後權重(1.7/1.7)
獲勝次數	22	35
失敗次數	78	65
獲勝率	22%	35%
兩者比較	較差	較佳

#### 五、實驗討論：

將成功條件改為 4096 後，我們發現成功率已經逐漸拉開了，從原本差 3.5%到改為 4096 後差 12.5%。

### 陸、研究結果



#### 一、步數公式

當局步數為  $m$ 、 $a_n$ =合併  $2n$  所需的步數、 $p$ =當局出現 4 的次數。

步數公式： $m=(a_1+ a_2+ a_3+.....+ a_{12})+14-p$

#### 二、各變數的功用分類

在實驗一當中我們發現平滑度與單調性在四個方向不容易出現相同的值，而空格數與最大值幾乎都相同，因此我們將平滑度與單調性稱為主要變數。

主要變數	輔助變數
平滑度	空格數
單調性	最大值

### 三、主要變數的最佳權重

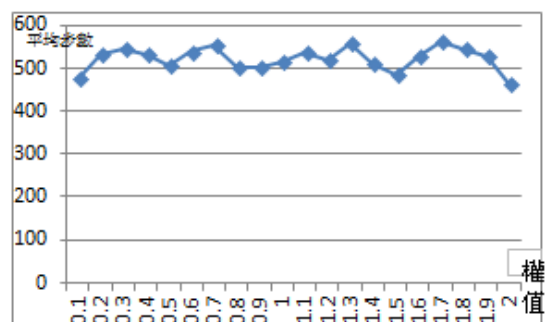
一次只改變一個權重，平滑度的最佳權重為 1，單調性的最佳權重為 0.15。但我們同時改變主要變數的權重(平滑度 1，單調性 0.15)反而使平均步數減少。

### 四、單調性(0.15)有助於 2048 的 AI 公式

我們測試 9 個單調性權重，而最佳權重不一定在這 9 個單調性權重中。因此我們以權重及步數當作 xy 座標，透過 GeoGebra 繪出 2-8 次方回歸方程式，發現 y 座標最大值(步數最大值)所對應的 x 座標值(單調性權重)都是在 0.15 附近，跟實驗二的最佳單調性權重(0.15)相近，可以說明我們把單調性改 0.15 是有助於 2048 的 AI 公式。

### 五、主要變數權重皆為 1.7 是平均步數的最佳情形

主要變數權重	平均步數
原始權重(平滑度 0.1,單調性 1)	332.8
主要變數權重皆為 0.1	477
主要變數權重皆為 1.7	562



由實驗四可知單調性和平滑度的比值 1 為平均步數較高的，我們以比值為 1 的去做等比例放大或縮小，發現平均步數都很高，尤其由實驗五可知兩主要變數權重皆為 1.7 是比值 1 中的最佳情形。透過比例也可以知道實驗二最後將兩個最佳權重合併是沒有助於此公式的。

### 六、在深度++的情況下，修改後的公式 2048 獲勝率較高

實驗六我們將深度調回++(限制時間 0.1 秒)，由 200 組數據知道修改後的公式獲勝率比較高(85%>81.5%)。

	原程式	修改權重後
深度++2048 獲勝率	81.5%	85%

### 七、在深度++的情況下，修改後的公式 4096 獲勝率較高

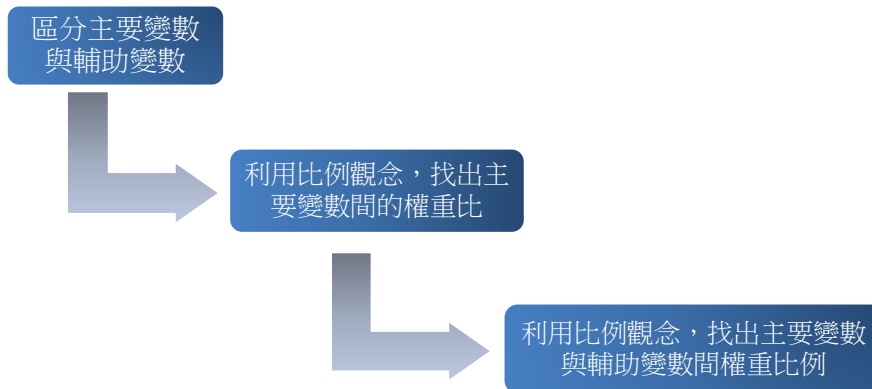
實驗七我們將成功條件改為 4096，並觀察獲勝率。得到以下實驗結果：

	原程式	修改權重後
深度++4096 獲勝率	22.5%	35%

## 柒、討論

- 一、在深度都為 1 的情況下，原程式平均步數為 332.8 步，而我們修改後的權重平均步數高達 561.8 步，雖然合併到 2048 的機率不高，但是以平均步數來看比原程式好很多，換句話說愈容易達到更高的數字。
- 二、我們研究的主要內容包括「研究程式碼」，以及在深度 1 的情況下「優化程式碼」，在研究結果中我們也發現主要變數比值為 1 的同比例放大縮小效果都不錯，在 90 組數據中有 $\frac{1}{3}$ 都有達到 500 多步，甚至有幾次達到 1000 多步，但是還是有數據的陣亡步數在 300 多步就陣亡了，換句話說以遊戲結束的結果來看，在同個權重下有 $\frac{2}{3}$ 的結果為 256 以及 512，而且有 $\frac{1}{3}$ 的機會達到 1024 及 2048。
- 三、在這份研究中，我們將原本程式檔的四個變數進行變數分類的特色分析，我們將其四個變數分成兩組變數：主要變數以及輔助變數。主要變數可以獨立進行遊戲判斷，而且扮演著執行策略的主要方向，例如：2048 遊戲獲勝策略之一是把大數字放在角落，在 AI 人工智慧程式中，就是指在四個變數中的單調性；而輔助變數是補充主要變數的不足，扮演輔助的功用，例如：玩 2048 遊戲時，如果空格數越多就表示有更多的空間及機會去達到 2048，在 AI 人工智慧程式中，就是指在四個變數中的空格數。
- 四、以比例的觀念，我們找出兩個主要變數權重的較佳比例為 1:1。而輔助變數會與主要變數的權重會互相影響，因此我們進行主要變數與輔助變數的比例縮放，這樣進行的原因是在於輔助變數會與主要變數的權重會互相影響，最後找出較佳權重(平滑度/單調性/空格數 /最大值)分別為 1.7、1.7、2.7、1.0。

五、由此研究發現 AI 人工智慧程式是可以藉由權重調整而得到優化，尤其利用區分變數類別(主要、次要)及比例觀念，可以讓優化的步驟更為簡單，往後遇上有變數有權重的程式時，有系統性的修改權重，可以得出明顯改善的較佳權重，而成功直接達到優化程式的目的，如圖所示。



(圖 7-1 本研究的優化步驟)

## 捌、未來展望

- 一、目前我們的研究藉由調整權重得出優化結果，進而有效率的達到優化，未來建議是否有其他有效率的策略進行改善 AI 人工智慧程式。
- 二、我們以比例的概念應用在 2048 遊戲上，對於有各種變數的 AI 人工智慧的相關程式，我們建議可以以利用區分變數類別(主要、次要)及比例觀念應用使其功效更佳。
- 三、本篇研究的 AI 人工智慧程式的運作方法為  $\alpha$ - $\beta$  剪枝法，未來相關研究方向可以討論如何優化蒙地卡羅樹狀搜尋法的 AI 人工智慧程式。

## 玖、參考資料及其他

### 一、網站資料

- 1.葉騏豪、梁朝欽、吳毅成 • TCGA 2014電腦對局研討會-2048人工智慧程式  
(<http://www.csie.ntnu.edu.tw/tcga2014/program-agenda/>)
- 2.2048-AI 程式原始檔  
(<http://bit.ly/1UIXeOF>)
- 3.JavaScript Tutorial - W3Schools  
(<http://www.w3school.com.cn/js/>)
- 4.魚蛋村 - Javascript 語法示範及教學  
(<http://old4.yudans.net/javascript/>)
- 5.維基百科 • Evaluation function  
([https://en.wikipedia.org/wiki/Evaluation\\_function](https://en.wikipedia.org/wiki/Evaluation_function))
- 6.維基百科 • Alpha-beta 剪枝  
(<https://zh.wikipedia.org/wiki/Alpha-beta%E5%89%AA%E6%9E%9D>)
- 7.維基百科 • 蒙地卡羅樹搜尋  
(<https://zh.wikipedia.org/wiki/%E8%92%99%E7%89%B9%E5%8D%A1%E6%B4%9B%E6%A0%91%E6%90%9C%E7%B4%A2>)
- 8.維基百科 • 對數  
(<https://zh.wikipedia.org/wiki/%E5%AF%B9%E6%95%B0#.E5.AF.B9.E6.95.B0.E5.87.BD.E6.95.B0>)
- 9.2048AI 討論  
(<http://tieba.baidu.com/p/3012462674>)

### 二、圖書資料

1. 耿衛東、陳為 • **電腦遊戲程式設計 = : Computer game programming** • 統一元氣資產管理出版。
2. 德洛拉 • **遊戲程式設計精華II** • 碁豐資訊出版。
3. 國中數學課本(康軒第二冊)
4. David Flanagan 著、黃銘偉 譯 • JavaScript 大全, 6/e (JavaScript: The Definitive Guide: Activate Your Web Pages, 6/e) • 碁峰資訊出版。
5. 陳思嘉、曹子涵 • 2048 的奧妙 • 第 55 屆第二分區科學展覽會得獎作品

## 【評語】 030802

1. 書面報告宜重排使其更具邏輯性。
2. 口頭報告流暢，宜增加自己玩 2048 經驗。
3. 宜先介紹遊戲規則，再進入變數定義。
4. 可以多考慮不同變數，如：相鄰同數個數、行與列等。
5. 若能加入機器自動學習方法來自動尋找多變數的加權。
6. 變數與加權可以先正規化(使值介於0~1之間)更能清楚比重。