遊戲產業和一些數學應用

一、遊戲產業(數學相關工作，遊戲公司在徵人時常用的大略敘述)：

1. 遊戲程式設計師

   需求:各種程式語言 常用 c、JAVA、Delphi

2. 遊戲企畫

   工作內容:

   A. 構思遊戲內容、玩法規則、系統設定與角色能力數值
   B. 針對測試結果與程式設計人員溝通並進行修改
   C. 評估製作過程所需之資源
   D. 製作內容規格書
   E. 建構遊戲數值體系、經濟體系與搭建數據模型。
   F. 遊戲分析、策劃和風險控制
   G. 基於業務分析角度，確保公司產品的低風險
   H. 遊戲用戶心理學方面的理解
   I. 透過數學機率方法，調配遊戲的回饋與張力
   J. 觀察市場生態

   需求:

   溝通能力、線性代數、機率、統計、離散幾何概念。

   保險、金融、風險評估知識

3. 遊戲數值專家(常常跟企畫混在一起徵人)

   工作內容:

   A. 建立數理模型
   B. 精通行業，能夠提出利於公司長期發展的產品提案
   C. 整體遊戲數值、獲利模式等策劃
   D. 構造數值發展曲線
   E. 遊戲數值規劃、平衡性調整
   F. 數值資料統計與分析

   例如:

   使用 Excel 製作遊戲的數學模型

   以 C 語言撰寫演算法，產生統計分析用的樣本資料

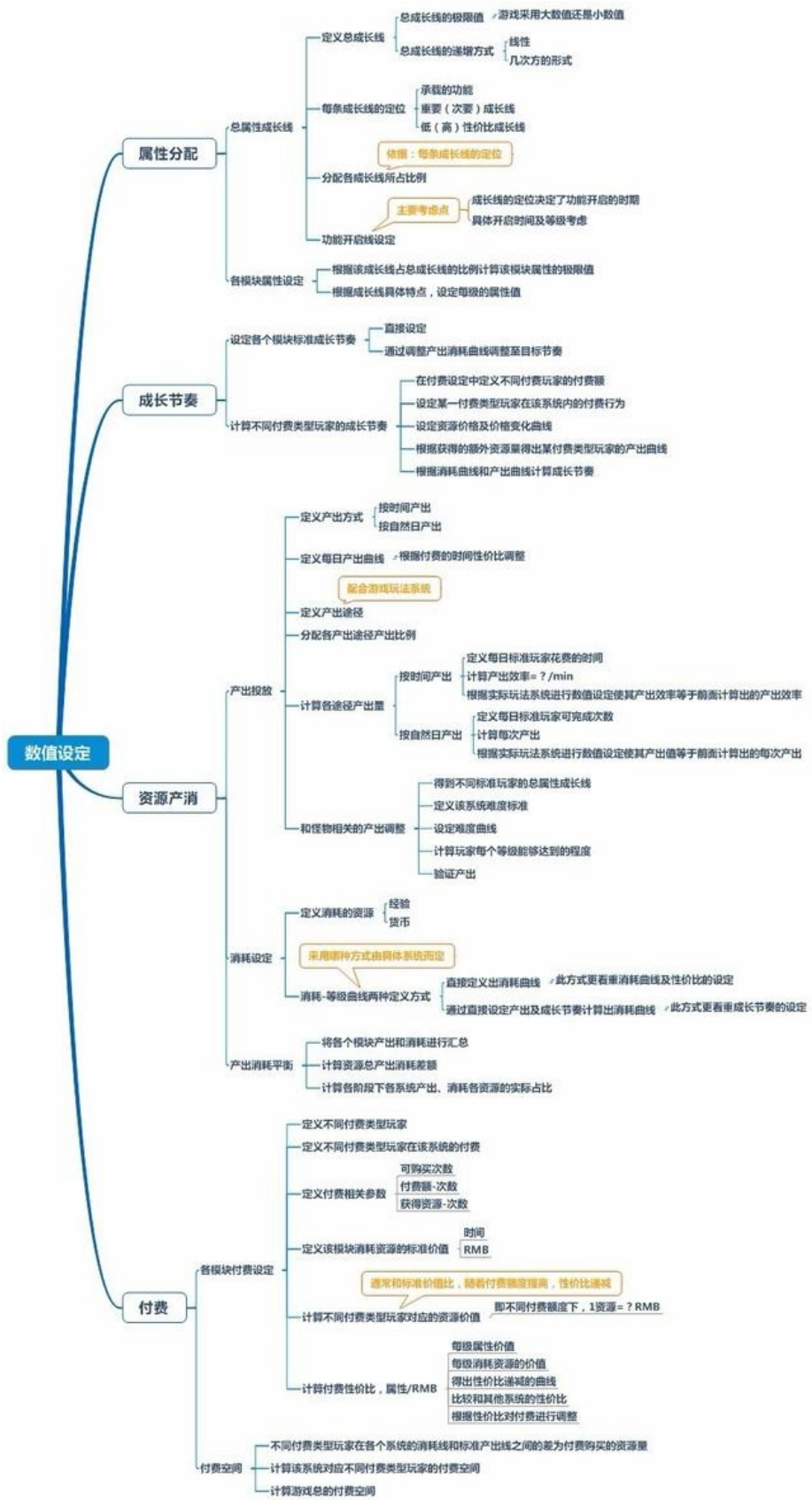
   檢驗演算法與數學模型的正確性

   需求: 數值分析、運算、模型建立能力

   編程語言、vba 等

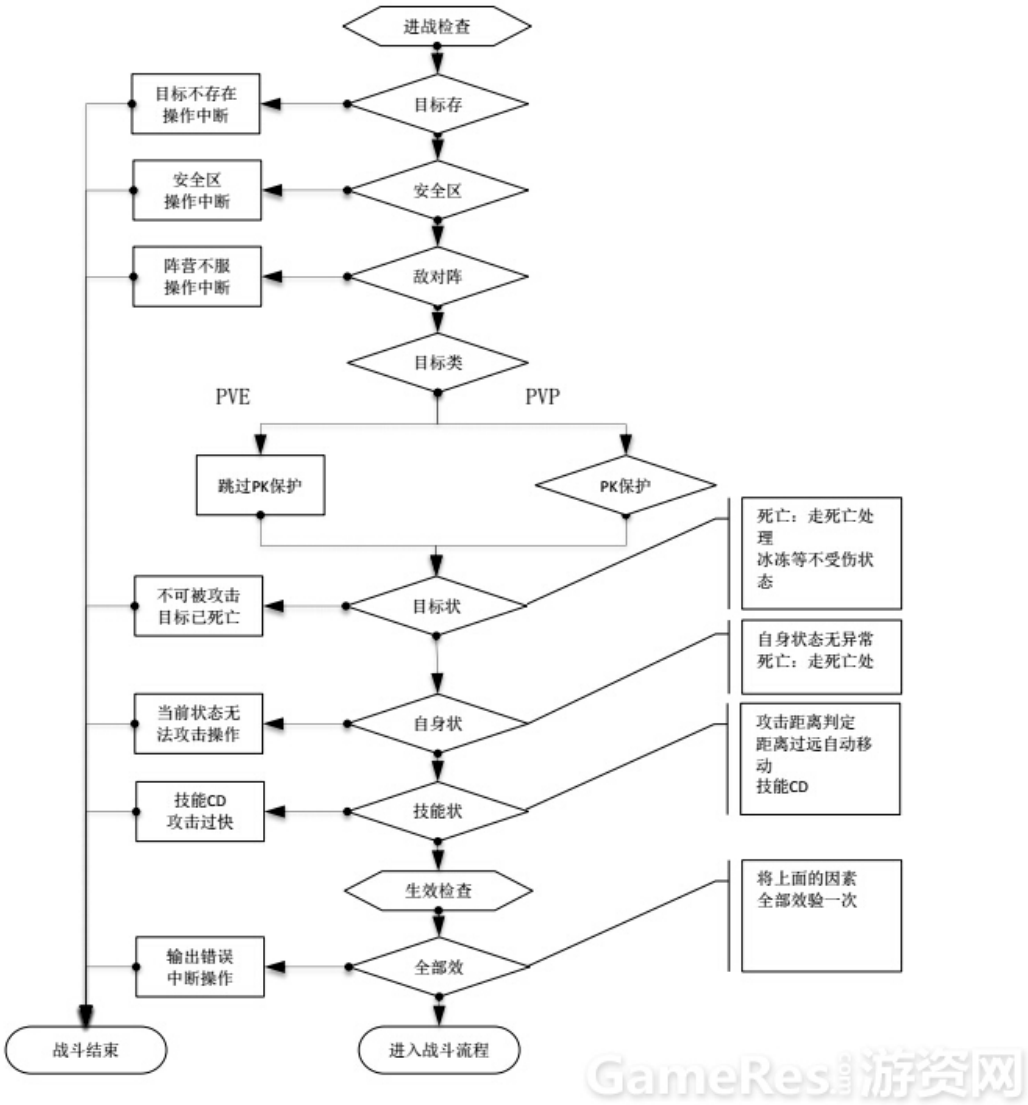
二、數值專家負責控制的數值：

http://bbs.gameres.com/thread_659627_1_1.html



三、一些事件的判定

關於遊戲數值的屬性策畫可參考 http://www.gameres.com/457630.html



四、一些將數學理論應用在遊戲中的簡單例子

例一：圓桌理論

來源：《魔獸世界》

核心概念

「一個圓桌的面積是固定的，如果幾件物品已經佔據了圓桌的所有面積時，其它的物品將無法再被擺上圓桌」

《魔獸世界》中，近戰攻擊結果的優先順序的遞減排列：

1. 先判定是否【未命中】
2. 如果命中是否【閃避】
3. 如果未閃避是否【招架】
4. 如果未招架是否【偏斜】
5. 如果未偏斜則是否【格擋】
6. 是否被怪物【碾壓】
7. 最後才是普通攻擊，決定是否【暴擊】

每次近戰攻擊都可能會出現未命中、閃避、招架、偏斜、格擋、碾壓，除此之外將是一次普通攻擊。

有些近戰攻擊結果的機率為 0% (古典機率)
例：玩家的自動攻擊造成碾壓的機率是0%，怪物的攻擊有0%的機率被偏斜，對沒有裝備盾牌的玩家的攻擊被格擋的機率是 0%

【圓桌理論】（遊戲特殊機制）
如果優先順序高的各部分機率總和超過 100%，會把優先順序低的各種結果出現機率降為 0%（擠出桌面）。

例：如果未命中、閃避、招架機率的和達到 100%或更高，攻擊的結果不僅不會出現普通攻擊，連偏斜、格擋、碾壓也不會出現。

設計：
挑戰遊戲 BOSS 的玩家團隊中，直接承受 BOSS 傷害的坦克職業（承受 Boss 攻擊的玩家），免於受到【碾壓】這種超強攻擊的傷害。並由此引申出針對不同職業的“免傷”方法。

與古典機率論的差異

例：假設有一個戰士對於他的目標作出了一次攻擊動作，只考慮閃避，招架，攻擊命中造成普通攻擊，攻擊命中造成暴擊四種情況。

目標的閃避機率……20%
目標的招架機率……5%
戰士的暴擊率……30%

古典機率論

在上述條件下，我們可以看到的擊中結果有如下幾種：

1、閃避，機率是 20%

2、招架，機率是（1-20%）*5%＝4%，要低於目標的原始 5%招架機率。

3、目標無法閃避或者招架此次攻擊，機率是 1-20%-（1-20%）*5%=76%

76%的命中機率，其中：

暴擊占 76%*30%=22.8%，小於原始 30%暴擊率

普攻佔 76%*70%=53.2%

最終運算結果是：

--> 閃避 20%

--> 招架 4%（小於 5%）

--> 暴擊 22.8%（小於 30%）

--> 普攻 53.2%

20%+4%+22.8%+53.2%=100%，合理。

不平衡點

上述例子中，目標每提高閃避 1%，就能確實提高 1%的機率不被攻擊到。然而目標提高 1%招架的時候，只能提高(1-閃避機率)*1%的機率不被攻擊到。

圓桌理論演算法

--> 閃避 20%

--> 招架 5%

--> 暴擊 30%

--> 普攻 （1-20%-5%-30%）=45%

20%+5%+30%+45%=100%，合理。

優點：

幾乎不需要進行運算。

不存在優先順序造成的衰減因數。

缺點：

普通攻擊的出現機率完全依附於其他的因素，相當於，將閃避，招架，暴擊處於同一優先順序，而普通攻擊作為最低級的存在。

極端情形

目標的閃避機率……20%+50%
目標的招架機率……5%
戰士的暴擊率……30%

結果：
--> 閃避 70%
--> 招架 5%
--> 暴擊 30%
--> 普攻 -5% ，不合理。

調整
--> 閃避 70%
--> 招架 5%
--> 暴擊 25%
--> 普攻 0%
合理。

例二：蘭徹斯特定律 Lanchester's laws

蘭徹斯特定律是英國工程師 F.W.蘭徹斯特提出了描述作戰雙方兵力變化關係的微分方程組，該方程組被稱為蘭徹斯特定律。

基本假設：
(1) 每一方兵力減損率與另一方兵力成正比；(2) 兩軍士兵都處於雙方火力範圍內；(3) 不考慮雙方支援部隊

交戰雙方 x,y
t 時刻雙方兵力: x(t) ,y(t)
初始兵力：x0,y0

動態模型為：

$$\frac{dx}{dt} = -yc_y$$

$$\frac{dy}{dt} = -xc_x$$

$$x(0) = x_0$$

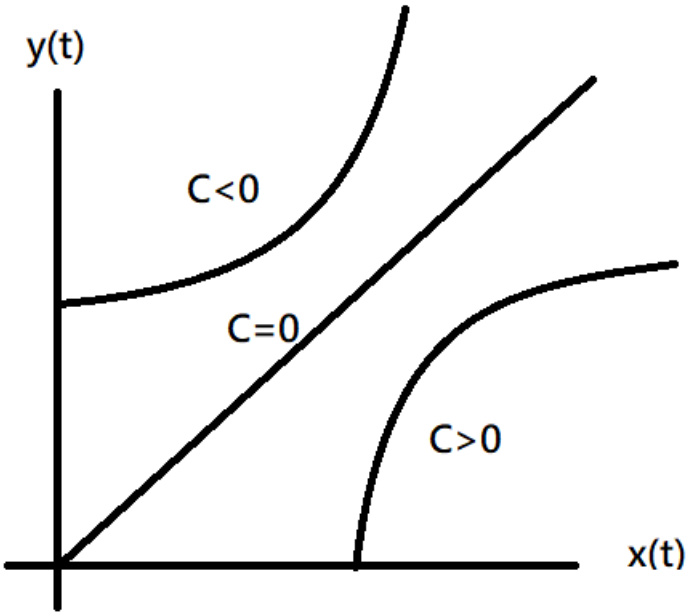
$$y(0) = y_0$$

$c_x > 0$ $c_y > 0$ 均為常數，分別代表 x 和 y 的有效攻擊係數。

整理後得到： $xc_x dx = yc_y dy$

積分得 $c_x x^2 - c_y y^2 = c = c_x x_0^2 - c_y y_0^2$

此為蘭徹斯特戰略公式。
該方程刻畫了隨著時間推移，雙方兵力的動態關係，為雙曲線。



公式顯示，這兩種基本方式對攻擊力及最終結果的影響力是不同的。兵力對攻擊力的影響是平方的正相關。這裡初始兵力對最終勝敗的重要性突顯出來了。這正是蘭徹斯特戰略公式的真正含義，說明兵力增加戰鬥力將大大增加。量變造成質變。

五、遊戲數值策劃工作中需要具備的技能：

節錄自 http://zhihu.timetw.com/55299.html

1. 數據庫結構

部分可參考 RPGMAKER，數據庫結構的大概念都跟其架構相似。

2. lua,C,VBA，fortran matlab 語言

初期要懂得 excel 至少要懂得查詢函數，偏移函數，if 判斷，可以寫出簡單的跨表查詢，利用自有控件和按鈕做出來一些簡單可調試的文件。懂得自定義函數，懂得定義單元格寫中文公式，懂得簡單的錄製宏(巨集)。

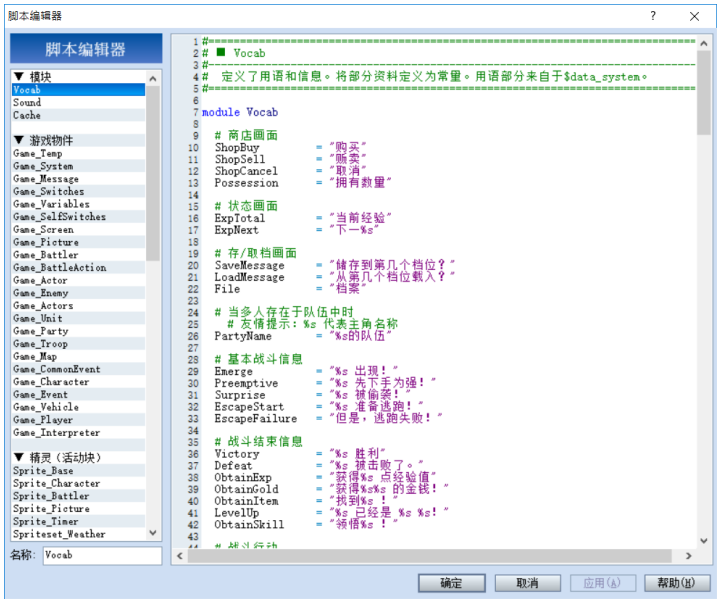
中期要懂得概率學，統計學的知識，如果不懂得高數，至少要去讀幾本書，看上幾十篇視頻，用 vba 完全脫離 excel 的框架，寫出來各種數值工具。

後期要懂得經濟學原理，去讀一些 fortran 語言，使用 matlab 工具，去研究物理碰撞(個人興趣)以及一些特殊的算法。

六、RPGMAKER

RPGMAKER 系列是 日本 Enterbrain 發行的 RPG 製作工具，隨時間不斷推出新版本，是許多獨立遊戲設計的模板。以 RMVX 為例：

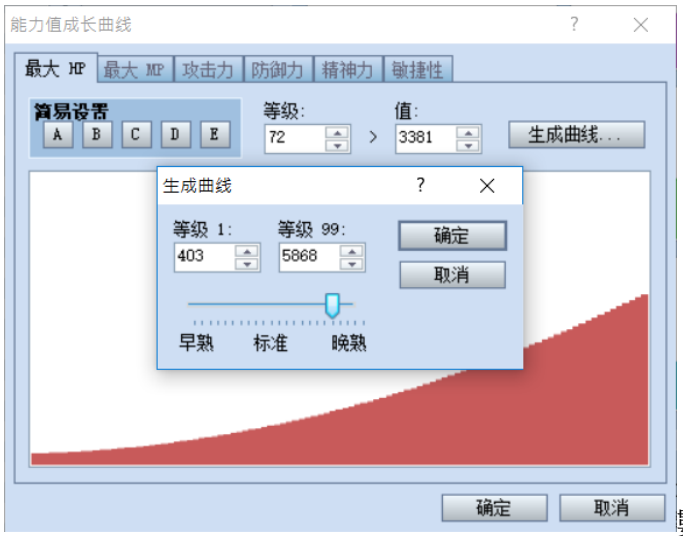
（隨便搜尋一下都有地方可以免費下載，想獨立製作遊戲或是想了解遊戲數據庫和運作的可以載來玩玩看）



實際上還是使用跟 C++差不多的語言模式

```ruby
def refresh
  @switch_max = ($data_system.switches.size - 1 + 9) / 10
  @variable_max = ($data_system.variables.size - 1 + 9) / 10
  @item_max = @switch_max + @variable_max
  create_contents
  for i in 0...@item_max
    draw_item(i)
  end
end
```

數據庫



數值策劃要做的內容

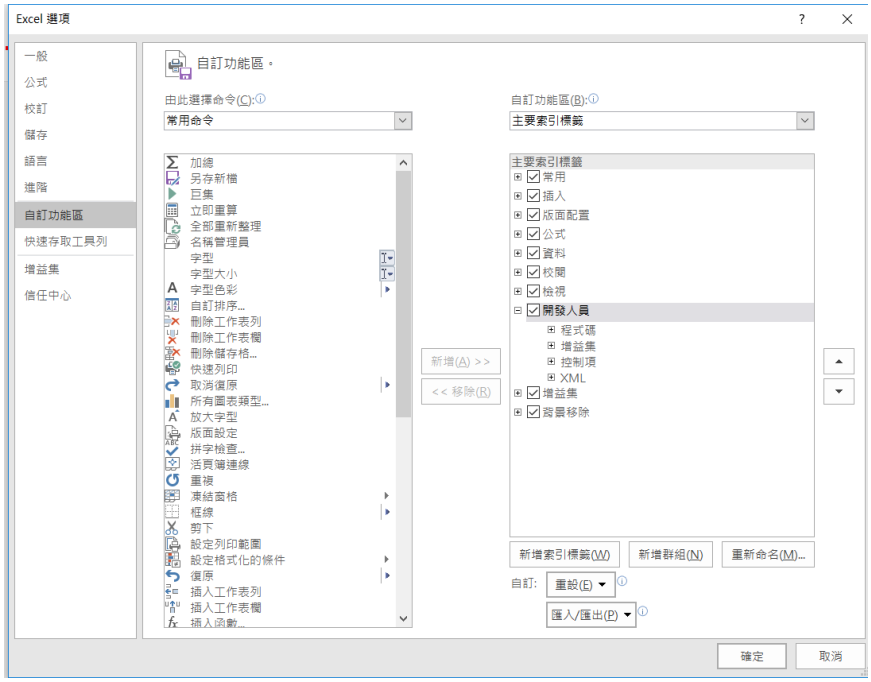
七、Excel 巨集和 VBA

http://www.excelhome.net/video/excel/vba/

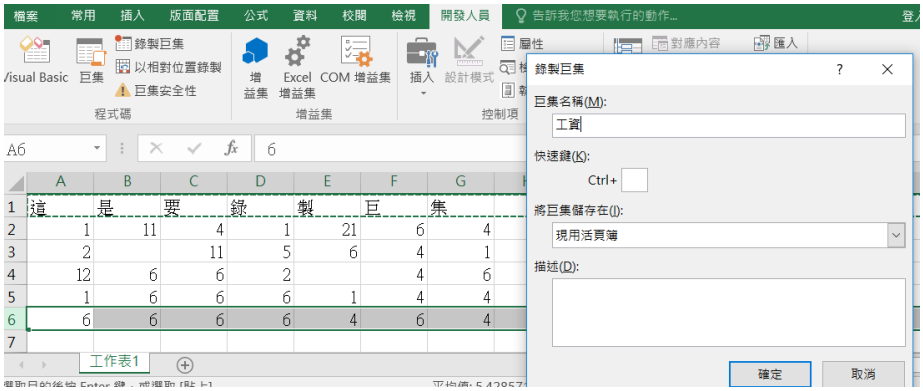
(影片裡有一堆廢話 很多可以跳過)

錄製巨集：

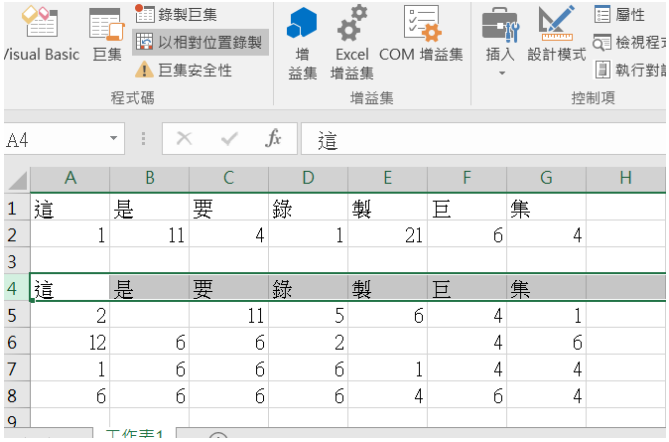
檔案-選項-自訂功能區 勾選開發人員



錄製巨集

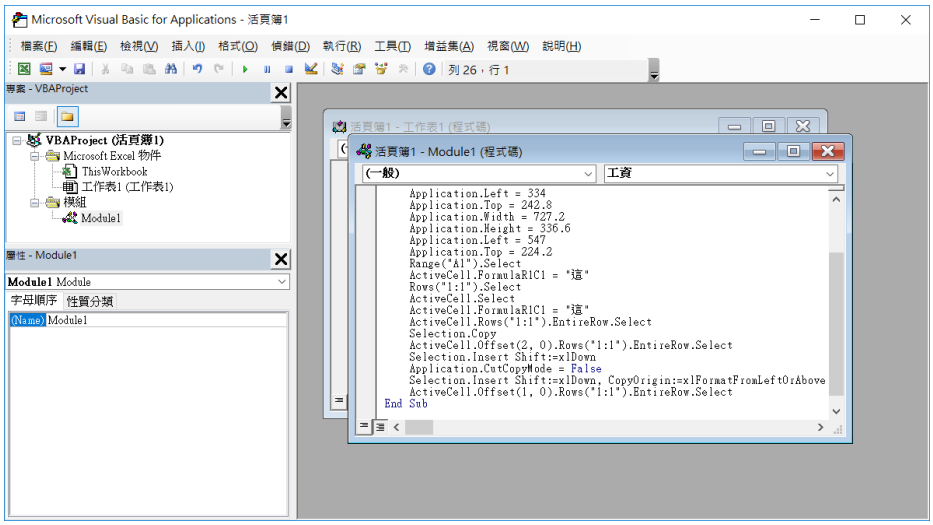


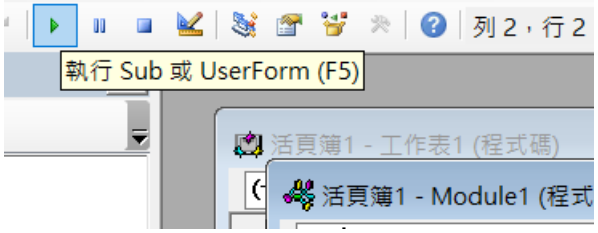
以相對位置錄製



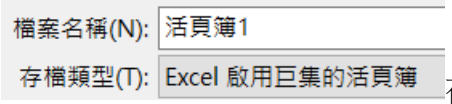
然後做一次複製動作
結束錄製

查看代碼快捷鍵 ALT+F11





按一次就會再複製一次(可是要調位置 有點麻煩)

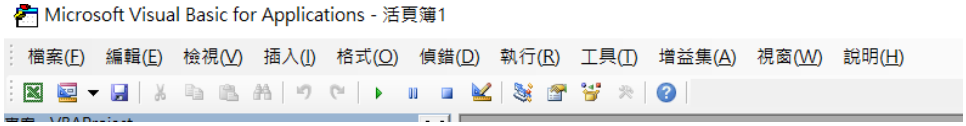


存檔方式。

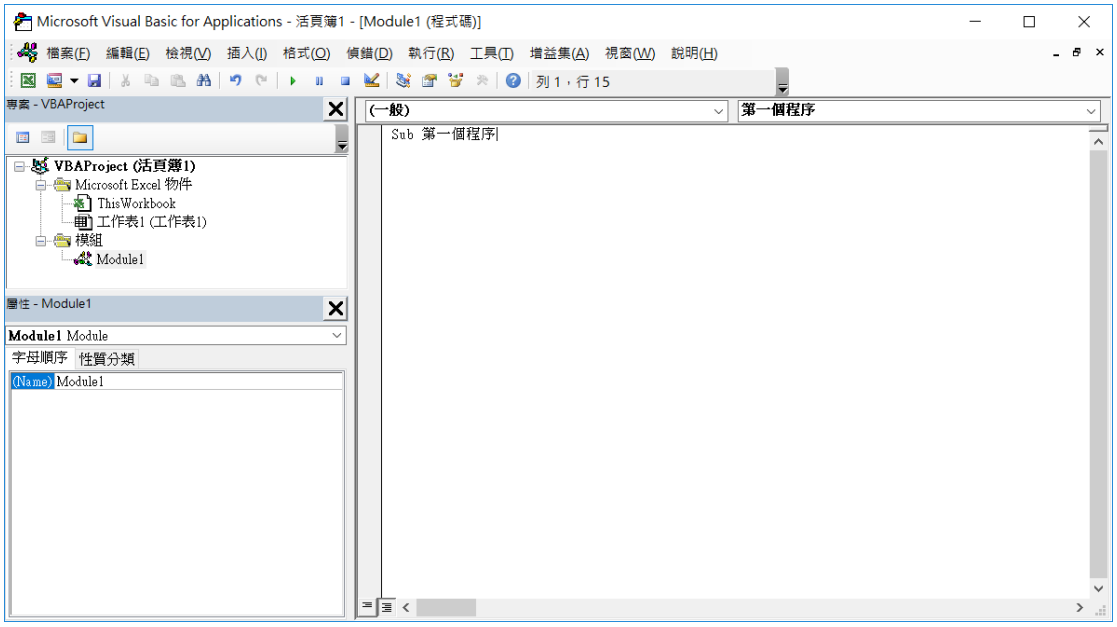
VBA:

進入 Excel 的 Visual Basic (ALT+F11)

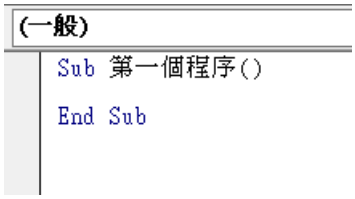
插入->模組



輸入 Sub 想要取的程序名字 (sub 過程)



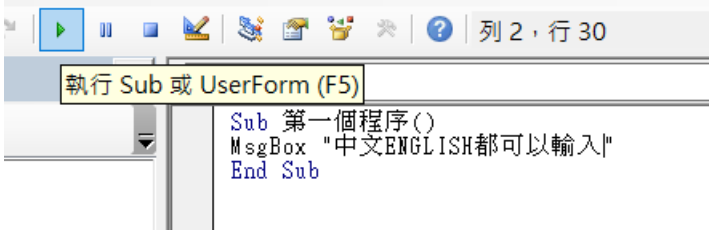
按 enter (可能要切回英文輸入) 會自動出現 end sub



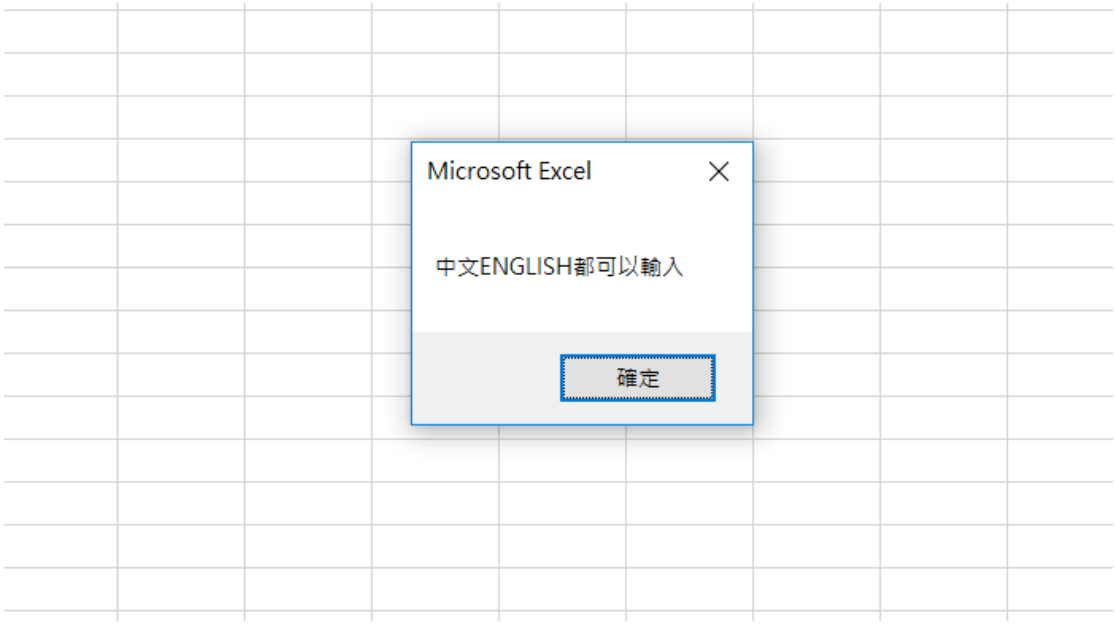
對話框命令 msgbox”” (messege)

```
(一般)                    第一個程序
Sub 第一個程序()
MsgBox "中文ENGLISH都可以輸入"
End Sub
```

按 F5 或是按那個綠三角



Excel 工作表就會自己跳出來了



可以有很多行

```
(一般)
Sub 第一個程序()
MsgBox "中文ENGLISH都可以輸入"
MsgBox "按確定之後就會再出現這一段話"
MsgBox "再按一次確定之後就會再出現這一段話"
End Sub
```

‘ 是注釋，跟程式本身沒關係

```
(一般)
Sub 第一個程序()
'這一段話對程式毫無影響
MsgBox "中文ENGLISH都可以輸入"
MsgBox "按確定之後就會再出現這一段話"
MsgBox "再按一次確定之後就會再出現這一段話"
End Sub
```

(沒錯，就跟在計算機與程式設計學到的一模一樣，只是換一個語法而已，真簡單…吧?)

八、附錄

未來科技與遊戲的結合
https://rocket.cafe/talks/17052

遊戲王 AR 影像
https://www.youtube.com/watch?v=uswCiok_UAk&list=PLggCI_883ruZ_HRJWzFMhX1kn_W07QnyK&index=1

未來科技遊戲
https://www.youtube.com/watch?v=w8Be9GjFTso

”上帝“的數學定義，人工智慧智商與等級劃分研究最新進展
http://blog.sciencenet.cn/blog-39263-1057881.html
https://link.springer.com/epdf/10.1007/s40745-017-0109-0?author_access_token=JK6WJ5nU6G5hXMaZ38De9_e4RwlQNchNByi7wbcMAY4lKpH-EueNdrn15pId9uQlE-JRJZ6r63rWnneLDiAAIueNxHm76TxKoE--cErrMUBHKPSrT63WQZqe7DoZEN2cQLAS-2vWAFY07JfkQ_Rxnw==
大意：

人工智慧科技發展迅速。許多人仍然認為這類科技如果不加以控制可能成為一個潛在威脅。於是大家就想用一種量化分析的方法來評估人工智慧的「智商」和「威脅程度」

“standard intelligence model”

$$Level1\colon M \xrightarrow{f} Q, \quad Q = f(M).$$

M 人工智慧　　Q 智商　　f 智商函數

M 應該具有四種能力：
知識獲取能力 I
知識輸出能力 O
知識掌握或存儲能力 S
知識創造能力 C

$$Q = f(M) = f(I, O, S, C) = a^* f(I) + b^* f(O) + c^* f(S) + d^* f(C)$$
$$a + b + c + d = 100\%.$$

$$Level2: Q \xrightarrow{\chi} K, K \in \{0, 1, 2, 3, 4, 5, 6\}$$
$$K = \chi(Q) = \chi(f(M))$$

智能等级	数学条件	范例
0	例 1，f(I）>0,f(O）=0; 例 2， f(I）>0,f(O）=0	理论上存在，现实中不存在
1	f(I）=0,f(O）=0	石头，钉子，木头
2.	f(I）>0,f(O）>0,f(S)>0,f(C)=0, f(S)=a ,a 为固定值，而且 M 掌握的知识与其他 M 不能共享使用	不能系统升级的洗衣机，电冰箱，扫地机器人
3	f(I）>0,f(O）>0,f(S)>0,f(C)=0, f(S)=a , a 随时间值增加	个人电脑，智能手机
4	f(I）>0,f(O）>0,f(S)>0,f(C)=0, f(S)=a , a 随时间值增加，而且 M 掌握的知识与其他 M 共享使用	云机器人，B/s系统，联网的阿尔法狗（AlphaGo）
5	f(I）>0,f(O）>0,f(S)>0,f(C)>0, f(S)=a , a 随时间值增加，而且 M 掌握的知识与其他 M 共享使用。	人类
6	f(I)>0，且趋近无穷大；f(O)>0，且趋近无穷大；f(S)>0，且趋近无穷大；f(C)>0，且趋近无穷大；	宗教中上帝或神的概念