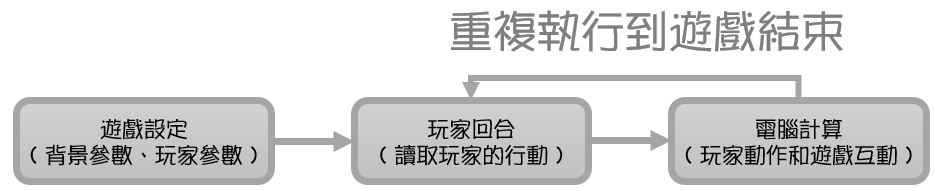
**第二章 終極密碼**

知道機率和迴圈是甚麼概念後，就可以製作非常多的遊戲，只要想得道通常都做得出來，基本遊戲流程如下：

* 1. 基本概念



簡單來說，一開始是參數設定，要設定的內容跟遊戲有關；則設定好後會到玩家回合，玩家輸入行動，由電腦(手機)計算之後再回去等待玩家

|  |  |
| --- | --- |
|  | 以左邊的抽獎程式為例：  第４～５行 🡪 參數設定  第７～９行 🡪 玩家回合  第10以後 🡪 電腦計算 |

接下來呢，將會基於這種架構，製作新的遊戲【終極密碼】

* 1. 終極密碼

第一步當然先進行參數設定，要玩終機密碼一開始需要3個數字，分別是：答案(num)、下限(low)、上限(high)，就可以組合出這遊戲的第一句話「從 ?? 到 ?? 中猜個數字」程式如下：

|  |
| --- |
|  |
| 第５行🡪 number=rand()%1000 代表從 0～999 中隨機取數值  第６行🡪 int low=0,high=1000 代表設定兩個數值  low 表示範圍下限，此時的下限是 0  high 表示範圍上限，此時的上限是 1000  第７&8行🡪 for( ; ; ){} ，在這迴圈裡面要放的就是 玩家回合 和 電腦計算 |

第一步的參數設定寫好之後進到第二步玩家回合，給出一個範圍讓玩家樹入數字開始猜，程式如下：

|  |
| --- |
|  |
| 在 for 迴圈增加了兩行程式：  第８行🡪 輸出的文字是：0 到 1000 中選一個數字，因為low=0 high=1000  第９行🡪 玩家輸入一個數字進去當成 x 的數值 |

GAME

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

完成參數設定和玩家回合後，就可以接著處理電腦計算的部分，這一步都要先思考遊戲運作的邏輯和流程，如下：

|  |
| --- |
|  |
| 1. 輸入一個數字進去程式，設數字為 X 2. 判斷 X 是否等於答案 3. 是的話代表猜到了，結束遊戲 4. 不是的話改變上限跟下限( low 跟 high ) 5. 如果 X 比答案還大，把上限改成 X。 【Ex. 0~100 🡪 40~100】 6. 如果 X 比答案還小，把下限改成 X。 【Ex. 0~100 🡪 0~60】 |

下一步就是把流程寫成程式碼，解說在下一頁

|  |
| --- |
|  |
| 第10行🡪 if(x==number) 表示 x 等於答案時，執行10~13中大括號的部分  第12行🡪 break 用來答對後會結束遊戲  第14行🡪 if(x>number) 表示 x 大於答案時，執行第15行  第15行🡪 hidh=x 代表「把上限改成X」  第16行🡪 if(x<number) 表示 x 小於答案時，執行第17行  第17行🡪 low=x 代表「把下限改成X」 |

|  |  |
| --- | --- |
| 那 x 跟答案的關係一定是大於、等於、小於的齊中一種，所以第10、14、16行都只會有一行是成立的，執行出來的成果如右： |  |
|  |  |

GAME

|  |  |
| --- | --- |
|  |  |

|  |  |
| --- | --- |
|  |  |

* 1. 遊戲改良

現在這樣的程式已經是一款遊戲了，但好像有些單調。終極密碼這款遊戲可以用２的次方數推估需要猜的次數。例如因為210=1024，所以今天從0～1000猜數字，理論上最多只需要10次

因次，接下來改良一下程式碼，讓每局都只有10次機會而已：

|  |  |
| --- | --- |
| 第7行🡪 times=10 代表猜10次  第9行🡪 輸出 剩下 times 次  第12行🡪 每次輸入一個數字， times 則會減一  第23行🡪 if(times==0) 判斷當次數等於0的時候就會結束遊戲 |  |

|  |  |
| --- | --- |
|  |  |

有比較需要動腦的玩法後，也要給玩家有重複挑戰或是直接放棄的選項



以下會示範那種「選項式的問答」怎麼實作：

|  |
| --- |
|  |
| 第7行🡪 char choise 代表，輸入後 choise 會記住輸入的東西  第9行🡪 輸出選項讓玩家看到  第10行🡪輸入選項存入 chiose  第11行🡪如果 choise 等於 Z 會執行12～23，執行遊戲  第25行🡪如果 choise 等於 X 會執行25～28，結束遊戲 |

GAME

|  |  |
| --- | --- |
|  |  |

