

終極密碼猜數字完整教案

台南市進學國小教師謝立益

lyshie@tn.edu.tw

一、電子書

1. 終極密碼猜數字—Scratch遊戲設計

<http://odata.tn.edu.tw/ebooktagapi/Intro/Index2?id=6782c2f27d2d4cbab1120b09c611b0b5>

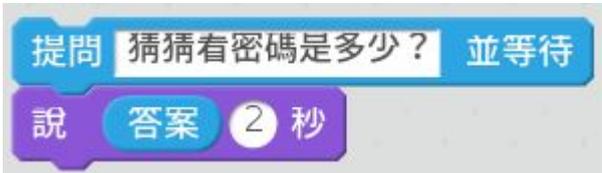
2. 進階終極密碼猜數字—Scratch遊戲設計

<http://odata.tn.edu.tw/ebooktagapi/Intro/Index2?id=8879fc4170684af09e03a323d87cfdfd>

3. 如果...否則...我的判斷正確嗎？—Scratch基礎概念

<http://odata.tn.edu.tw/ebooktagapi/Intro/Index2?id=2bd653b52e7d42afa81f4ef2da27e33b>

二、教學內容

活動內容	概念
不插電活動 <ol style="list-style-type: none">實際玩終極密碼2~3次(教師與學生皆可當關主)，準備小禮物2~3份。可以玩「幸運者」或「倒楣鬼」其中一種，猜中就中獎。數字選定0~99之間(zero-based，學生習慣使用1~100)二位數字，教師選定數字後寫在白紙上放進盒子(即變數空間，與執行過程中內容物不變的特性)。學生依序猜一個數字，猜錯則提示，提示「大一點」或「小一點」，並將猜錯之數字紀錄於白板上。重複上述步驟，直至猜對為止。最後強調「密碼」從頭到尾都未曾修改過，以符合遊戲的公平性。	遊戲規則 遊戲流程
Scratch創作 因學生對於遊戲進行已具有整體概念，遊戲設計不再進行Top-down功能分析，而是Bottom-up方式建構程式。	轉化遊戲流程為程式
<ol style="list-style-type: none">「提問」與「答案(回答)」成對出現。學習如何與使用者互動，並將使用者輸入的資料呈現出來。  名詞澄清：「答案」指的是玩家的回答，「密碼」則是老師一開始寫下的數字。	輸出、輸入互動
<ol style="list-style-type: none">加入「判斷」，檢查答案(回答)與密碼(71)是否正確。	條件判斷 (IF-THEN)、暫存空間

錯誤樣態：

3. 此處，引導猜對才有回應，那麼猜錯是否也要有回應。從 IF-THEN 進階成為 IF-THEN-ELSE，以便日後熟悉另一種更彈性的表列方式。請學生自行設計，答對與答錯的回應訊息。

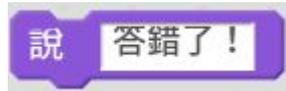
條件判斷 (IF-THEN-ELSE)，[反向表列](#)方式。

4. 教師可以提問，現在的遊戲只給一次機會而已，未免太難、太可惜了。是否，如同前面實際的遊戲中，給予重複的機會。此時，導入「重複」的概念。

重複



錯誤樣態：未等待，說話內容即消失。



5. 答錯後沒有提示，太困難。加入提示功能，增加遊戲的挑戰性與策略性。將提示的程式片段，加入在「答錯了！」之後。
- 如果猜錯的答案比密碼「大」，就提示「小一點」；
 - 如果猜錯的答案比密碼「小」，就提示「大一點」。



錯誤樣態：單用IF-THEN，雖然簡單好理解，但形成不必要

巢狀的條件判斷 (nested IF-THEN-ELSE)，此處難度增加，需適時導入其他活動來澄清概念

的判斷，及潛在的問題。

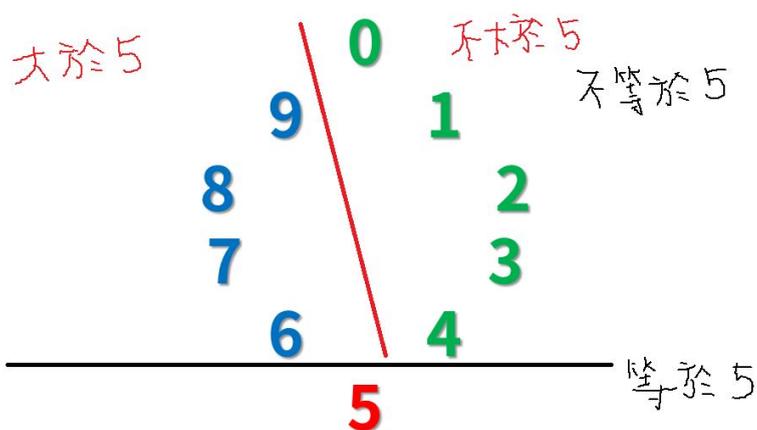


不插電活動

為避免學生使用多個IF-THEN來判斷，此處進行多次的二分法 (IF-THEN-ELSE)將數字分成多堆練習，如下圖。

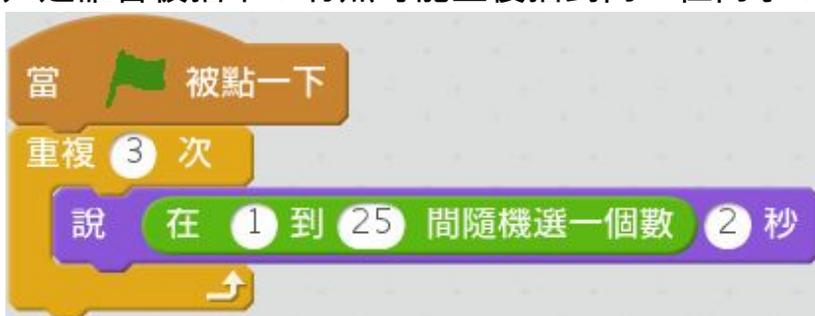
詳細活動內容：

<http://odata.tn.edu.tw/ebooktagapi/Intro/Index2?id=2bd653b52e7d42afa81f4ef2da27e33b>



插電活動

準備2~3個小禮物，將班上同學1~25號進行抽籤，使用隨機亂數抽取2~3次，抽到即贈送。提問學生，這2~3次的抽籤過程中是否知道誰會被抽中？有無可能重複抽到同一位同學？



隨機亂數

6. 密碼是寫在程式裡面，提示學生如果老師偷看到程式，就

亂數重複抽取，未

知道密碼，是否有方法不讓玩家知道。學生會運用前述的「隨機亂數」。發展出以下程式。
 錯誤樣態：意圖沒有錯，但未能考量這個隨機亂數的積木會變動。



能保持密碼的不變性

7. 提示學生，老師一開始將密碼寫在紙上放進盒子，如何模擬出類似的動作。引導變數儲存空間的概念。



為變數這個盒子(空間)命名，如「盒子」、「潘朵拉的盒子」、「口袋」與「密碼」等。須澄清「變數名稱」與「變數內容物」的差異。這個空間叫做「密碼盒」，裡面放的是「密碼」這兩個中文字。

資料儲存、變數不變性、變數命名

8. 使用變數空間來存放「一開始」選定的密碼。

變數的寫入

```

當 被點一下
設定變數 盒子 為 在 0 到 99 間隨機選一個數
重複執行
  提問 猜猜看密碼是多少？ 並等待
  如果 答案 = 盒子 就
    說 答對了！ 2 秒
  否則
    說 答錯了！ 2 秒
    如果 答案 > 盒子 就
      說 小一點！ 2 秒
    否則
      說 大一點！ 2 秒
  上一步

```

錯誤樣態：

```

答案 = 盒子
重複執行
  設定變數 盒子 為 在 0 到 99 間隨機選一個數
  提問 猜猜看密碼是多少？ 並等待
  如果 答案 = 盒子 就

```

9. 刻意不隱藏變數內容，讓學生自行發現密碼協留在畫面上。即呈現變數內容物的空間概念，也增加活動趣味性。最後，設法隱藏變數內容。

變數的內容物

後續可導入的進階概念：

1. 字串相加，顯示最終密碼。
2. 製作一個積木(功能模組化)，區分「猜數字」、「顯示得分」等功能模組。
3. 條件迴圈，限制遊戲次數。另將「無限迴圈」改用「條件迴圈」與「**不成立條件**」實作，以適應其他非平行的情況。
4. 使用「廣播訊息」與「接收訊息」，安排趣味圖示與音效於答對或答錯時呈現(廣播概念示範：「**請起立**」、「A同學請起立」與「B同學請起立」。此時，會發生有趣現象，**因未指名，則全班學生都會起立**。廣播是全部的人都聽得到，但每個人都會過濾訊息，專注與自己有關的訊息)。

```
當被點擊  
變數 number 設為 隨機取數 0 到 99  
變數 times 設為 1  
循環直到 times > 3  
  猜數字  
  變數 times 改變 1  
說道 三次機會到了 2 秒
```

控制遊戲次數

```
定義 猜數字  
提問 猜猜看，密碼是多少？ 並等待  
如果 答案 = number 那麼  
  說道 字串組合 答對了！ 和 字串組合 答案是 和 number 2 秒  
  停止 全部  
否則  
  如果 答案 > number 那麼  
    說道 數字小一點！ 2 秒  
  否則  
    說道 數字大一點！ 2 秒
```