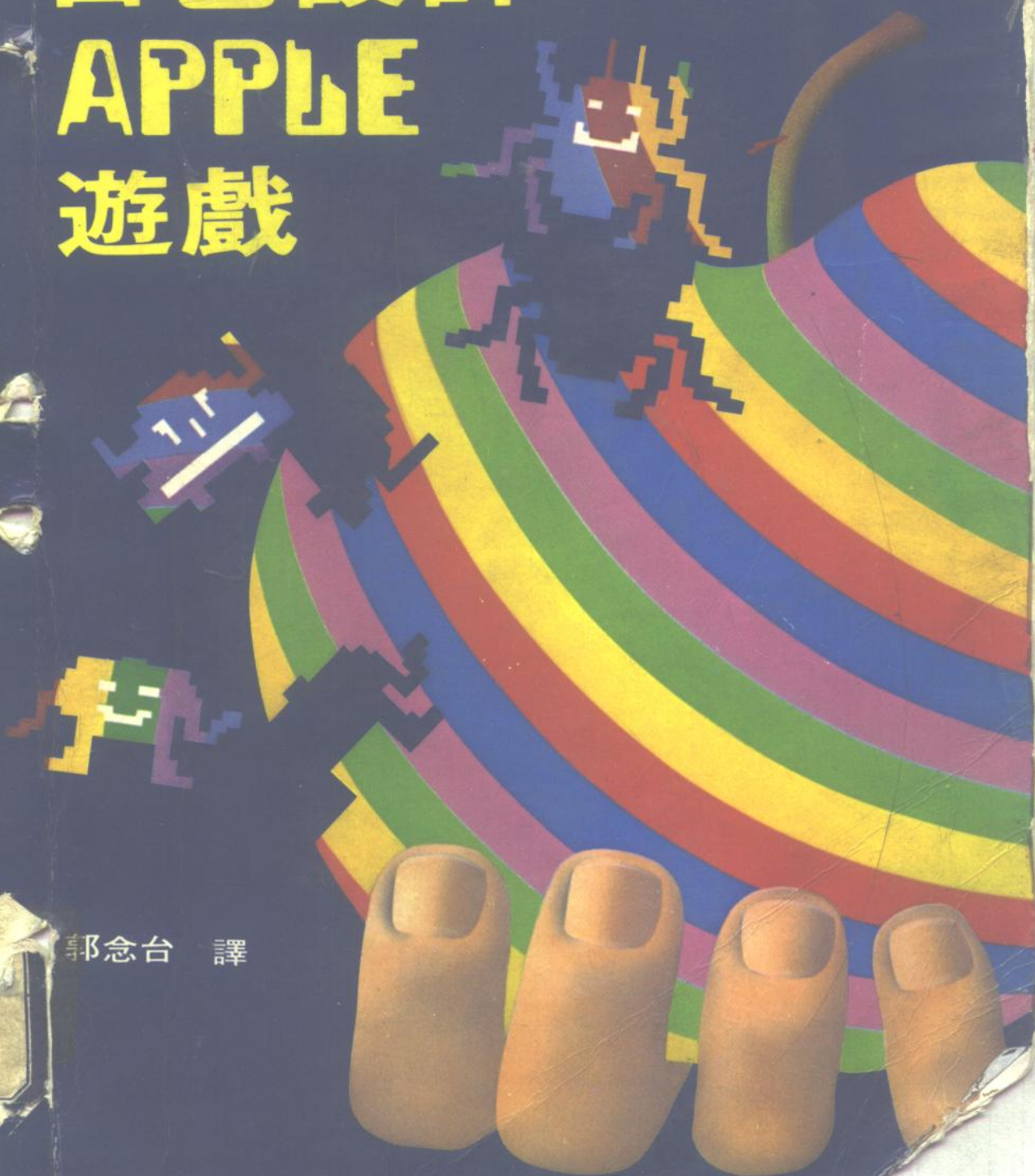


自己設計 APPLE 遊戲



邱念台 譯

TP36
121

自己設計 APPLE 遊戲

郭念台 譯

儒林圖書公司 印行

版權所有
翻印必究

自己設計 Apple 遊戲

原著發行日期：1983 年

原著書名：WRITE YOUR OWN APPLE GAMES

原著者：Stuart Anstis

譯者：郭念台

發行人：楊鏡秋

出版者：儒林圖書有限公司

地址：台北市重慶南路一段 111 號

電話：3812302 3110883 3140111

郵政劃撥：0106792-1 號

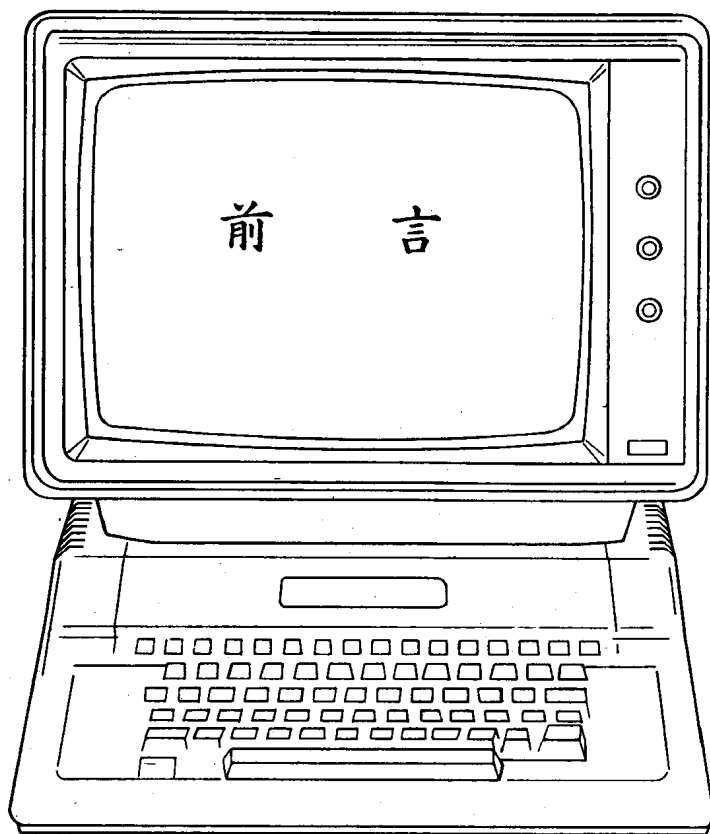
吉豐印刷廠有限公司承印

板橋市三民路二段正隆巷 46 弄 7 號

行政院新聞局局版台業字第 1492 號

中華民國七十三年十二月初版

定價新台幣 180 元正



首先，歡迎您加入 Apple 電算世界！Apple 是一部設計極佳的個人電腦，它有極佳的繪圖能力，並且可接遊戲控制器，因此特別適合做電腦遊戲。本書含有四十種以上 BASIC 遊戲和程式，它們都是特別為 Apple 設計的，可充分顯示 Apple 的優點。

這些遊戲是以 48 k 的 Apple II Plus 為準而寫的，而且也都在這種電腦測試過，但是因為它們都是用標準 Applesoft Basic 語言寫成的，因此應該可以在任何一部可使用該語言的 Apple 上執行。這些遊戲大多需要接上 Apple 遊

戲控制器，而且需要一部磁碟機來存取程式。注意，本書所示之遊戲圖片雖然都是以黑白印刷的，但是實際上這些遊戲程式都充分利用了 Apple 的彩色繪圖能力。

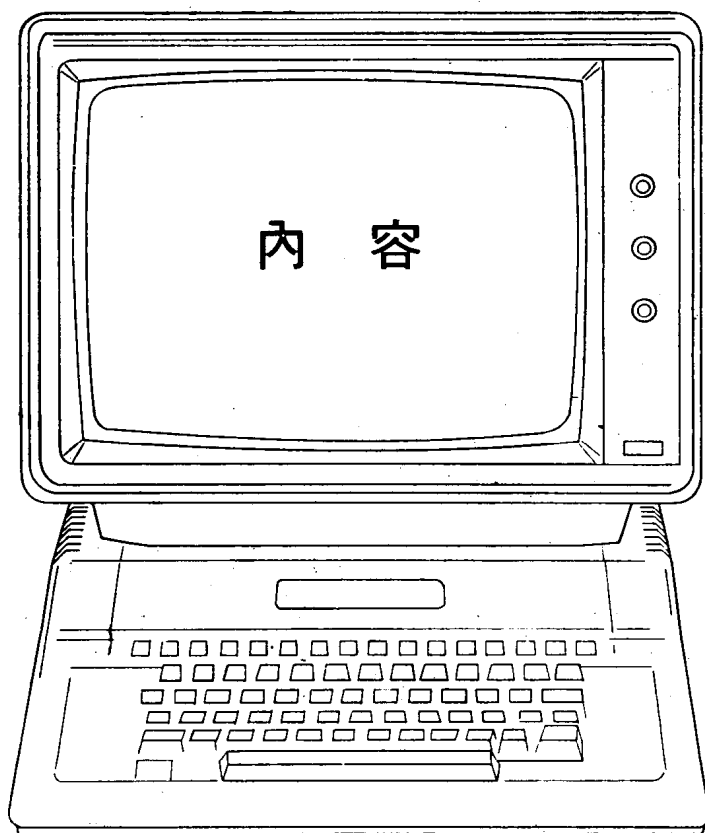
這些遊戲之中有些是老遊戲；例如“劊子手”（Hangman）和“井字遊戲”（Tic-Tac-Toe）等，有些是將市面上的程式改寫為 Apple 程式；例如“直昇機”（Helicopter）和“電位綫”（Electric Field）*等。不過，大多數都是全新的設計。

所有遊戲程式都列在本書中，您可以把它們打入電腦，SAVE（存放）在磁碟片上，再叫出來玩。如果只是為了玩這些遊戲，您根本無需去懂 BASIC 與電腦，但是為了獲得更大樂趣，您可以學習如何充分利用您的 Apple 與使用手冊自己設計 BASIC 遊戲。仔細研讀書中各遊戲的說明，並試着玩玩那些遊戲，然後再試着設計自己的遊戲程式。

第一章為您介紹一系列您可以利用來設計遊戲程式的技巧，副程式，和邏輯架構（algorithm）。筆者假定您已經很熟悉 BASIC，希望您將第一章內的所有副程式打入電腦，執行它們，並試着去修改它們。仔細研讀書中的遊戲程式，您將會發現這些副程式中有許多暗藏在那些遊戲程式內。

祝您成功！

* 譯者註：參閱“電位綫”那一節。



前 言	1
-----------	---

1. 寫作遊戲程式之技巧	1
--------------------	---

以模組為單位寫作	1
----------------	---

讓遊戲者輪流玩	3
---------------	---

指定顏色給遊戲者	5
----------------	---

索取遊戲者之姓名，並指定一個遊戲控制器給他	7
-----------------------------	---

多利用遊戲控制器.....	7
藉閃爍來吸引遊戲者.....	11
從鍵盤中斷程式執行之方法.....	13
字鍵代碼.....	15
利用鍵盤來控制動作.....	18
方格圖的畫法.....	19
在方格圖中選格子.....	20
在方格圖中移動.....	21
乒乓球拍.....	22
碼錶.....	23
記分.....	25
移動坦克.....	25
移用兩頁高解析度記憶頁的動態畫.....	27
高解析度圖形，形狀表，與商用繪圖軟體.....	28
圓，螺旋線與鐘面.....	29
多邊形，星形與旋形圖(Squiral)	34
平均值與插值法.....	37
爬行的字幕.....	39
亂數.....	40
整數 BASIC 中的亂數函數.....	40
Applesoft 的亂數函數.....	43
修截亂數.....	43

亂數可重複嗎？.....	44
零與壹.....	47
設定上下限.....	48
隨機時間延遲.....	48
東倒西歪 (Random walk).....	49
東倒西歪一加一或減一.....	50
不等步伐大小的東倒西歪.....	52
濾平 (Smoothing).....	53
不等頻率分佈的亂數.....	56
低解析度圖形的擲骰子遊戲.....	59
在遊戲程式中加註標題.....	61
徵求朋友的意見.....	63
一個好遊戲應該怎樣？.....	65

2. 由選擇表開始動手..... 67

“ Hello ” 程式.....	67
-------------------	----

3. 動作遊戲..... 76

坦克大決戰 (Tank).....	76
衝刺 (Sprint).....	82
轟炸機 (Bomber).....	86
伺機而發 (Wait for It).....	90

賽馬 (Horse Race)	93
打靶 (Target)	96
直昇機 (Helicopter)	99
緊迫盯人 (Tiddler)	104
乒乓 (Ping Pong)	106
太空突擊隊 (Space Zapper)	109
美式骰戲 1 (Craps)	113
美式骰戲 2 (Diddle)	116
踩尾巴 (Cross Porpoises)	119
舉重選手 (Weight Lifters)	122
 4. 策略遊戲	 125
浪子回頭 (Zig-Zag)	125
搭橋 (Bridge-It)	129
股票市場 (stock Market)	133
黃金大拍賣 (Gold Auction)	137
井字棋 (Tic-Tac-Toe)	142
翻面 (Turnover)	147
吞食蛋糕 (Chomp)	149
 5. 猜字遊戲	 153
創子手 (Hangman)	153

我的字是什麼？與 Wordlist 1	157
拼字遊戲 (Anagrams)	163
生日卡 (Birthday Card)	168
貴姓大名 (What's Your Name ?)	170
訊息看板 (Any Messages ?)	173
請勿亂動 (Do Not Adjust)	174

6. 繪圖副程式

蝴蝶 (Butterfly)	175
烟火 (Firework)	177
高解低度與低解析度等電位圖	179
李沙育圖 (Lissajou)	182
蒲公英 (Dandelions)	184
酒杯墊 (Doily)	185
六角星圖 (Stars)	186
多邊形與旋形圖 (Polygons and Squirals)	187
碼錶 (Stopwatch)	189

7. 單人遊戲

記憶人 (Memory Man)	193
風琴 (Organ)	197
音樂 (Music)	202

迷宮 (Elithorn)	205
爬坡 (Running on Empty)	208
數字猜謎 (Guess a Number)	210
尋寶 (Buried Treasure)	214
换位遊戲 (Jumpers)	216
拔釘子 (Peggy)	219
智慧盤 (Fifteen)	223
開鎖 (Safebreak)	227
耍把戲者 (Juggler)	231

8. 教育與科學程式

箭 (Arrow)	235
東倒西歪 (Random Walk)	239
擴散 (Scatter)	243
巴斯卡三角形 (Pascal's Triangle)	245
魔術方塊 (Magic Squares)	248
動畫描圖板 (Animated Sketchpad)	253



以模組為單位寫作

當您準備開始寫作一個遊戲程式時，最好先畫出它的流程圖，這樣可幫助您了解遊戲的邏輯架構，而且可將遊戲分解成若干較少的模組（**module**），然後您就可以以模組為單位來設計。舉例而言，假定您想寫作“井字棋”（**Tic-Tac-Toe**）的電腦程式，這遊戲可能包含以下諸步驟：

1. 設定各個變數的初值

2 自己設計 Apple 遊戲

2. 畫出方格圖（或棋盤）
3. 輸入遊戲者要下的位置
4. 檢查要下的位置是否已有棋子
5. 在選定的位置上畫出代表遊戲者一方棋子之符號
6. 電腦根據策略決定要下的位置
7. 檢查該位置是否已有棋子
8. 畫出代表電腦這一方棋子之符號
9. 檢查看看是否有三個棋子在一條綫上；亦即，是否有人贏了
10. 回到程序 3

上列步驟中 1 至 9 都是一個單獨的副程式，有些步驟看起來一樣，而且在盤遊戲中要出現好幾次，因此可以合成一個副程式。無論是輪到遊戲者或電腦，每下一步之後您都必須檢查那一步是否符合規則；亦即，那一步是否已經有棋子。步驟 4 與 7 都是做同樣的事，因此很明顯地可以用一個副程式來做，讓它在每一回合中執行兩次。如果那一步符合規則，則您需要根據輪到那一方而呼叫畫叉或畫圓圈的繪圖副程式。無論是輪到遊戲者或電腦，每下一步之後您都需要呼叫前述繪圖副程式（步驟 5 或步驟 8），再檢查看看是否有人贏了（步驟 9）。現在，將這些步驟畫成流程圖，將各個副程式放在一個方框內，在需要做選擇的地方把問題放在一個菱形框內，並在指向下一步的箭頭旁邊標上“是”或“否”。

您現在可以思考如何着手寫作各個副程式了。首先，看看那些步驟需要遊戲者輸入情報給電腦，或者需要電腦回饋情報給他，盡量試着讓這些情報以使用者為導向（user-oriented）；亦即，盡量使說明與規則簡潔易懂，使遊戲者儘快了解玩法。記住一個原則：如果能夠在螢光幕上用圖形說明就千萬不要用文字說

明。此外，如果需要遊戲者輸入資料時千萬不要期望他是位專業打字員，要儘量使遊戲者所需的動作減至最少。如果您需要的是“是／否”（YES / NO）的答案就不要要求他打“0”來代表“是”，或以“1”來代表“否”。對電腦而言，答案同樣是一連串的0或／與1，但是對人們而言，此種表示法稍嫌麻煩了一些，儘量讓他自然地回答“是”（YES）或“否”（NO）。不過，當您需要這種答案時，用“Y”來代表“YES”以及用“N”來代表“No”不是更簡單嗎！為什麼要讓他打三個字呢？

關於遊戲情報的顯示，利用圖形顯示井字是一件很自然的事，但是到底要使用高解析度圖形呢，還是低解析度圖形呢？對井字棋，這不是困難的問題——您只需要九個方格，每個格子能夠容納一個叉或一個圓圈即可。低解析度圖形就夠了，而且它會給遊戲者更堅實，更有色彩感。此外，低解析度圖形比高解析度圖形更適合步驟4的執行；步驟4的目的是檢查某個格子內是否已有棋子，利用低解析度的SCRN功能可輕易地完成這件事（SCRN的功用是檢查某已知方格內的顏色）。高解析度圖形法中沒有SCRN功能，這使得人們傾向於使用低解析度圖形來顯示這個遊戲。用來畫方格與棋子符號的繪圖副程式並不困難，我們稍後會說明。

步驟9是一個檢查是否有三個棋子（圓圈或叉）在一條綫上的副程式。寫作遊戲程式的基本觀念就是先把它分成若干較小的模組，然後分別寫作這些副程式，再把它們組成一個完整的程式。接下去，我們要討論寫作這些模組的各種技巧。

讓遊戲者輪流玩

在所有兩人遊戲中，遊戲者都是輪流參與的；在電腦上最好的方法是把主程

4 自己設計 Apple 遊戲

式交給一位遊戲者，然後再交給另一位遊戲者。要執行這件事，只要定義一個變數 P （代表 Player），讓它在 0 與 1 之間交互變化，並決定輪到誰即可；例如， $PDL(P)$ 就可根據輪到誰去讀取 0 號控制器或 1 號控制器的數值。下面這段程式即可完成上述手法：

```
100 P = 0
110 P = 1 - P
120 PRINT P, PDL(P); GOTO 110
```

若想令 P 在 1 與 2 之間交互變化，可使用

```
100 P = 1
110 P = 3 - P
120 PRINT P, PDL(P - 1); GOTO 110
```

下面是較通用的程式，它適用於包括負數在內的任何一對數目：

```
90 REM ALTERNATION
100 INPUT "TYPE IN TWO NUMBERS"; A; B
110 PRINT "THIS PROGRAM WILL ALTERNATE
    BETWEEN "; A; " AND "; B
120 P = A
130 P = (A + B) - P
140 PRINT P; " ";
150 GOTO 130
```

您可以把每個遊戲者的次數，積分等資料分別存放在陣列 (array) 中，您可以用下法宣告陣列：

```
10 DIM ASSETS (2), CAPITAL (2), TIME (2)
```

雖然您不一定要宣告它們，因為 Applesoft 會自動執行

```
DIM ASSETS (10)
```

只要您在程式中使用了它。如果一位遊戲者贏了 10 點，而他的對手輸了 8 點，您可以利用

```
145 ASSETS (P) = ASSETS (P) + 10
148 ASSETS (1 - P) = ASSETS (1 - P) - 8
```

來記錄他們的成績。(您不能用 SCORE 作為變數名字，因為 scORe 有保留字 OR 在裏面。)

指定顏色給遊戲者

在低解析度圖形遊戲中最好為每位遊戲者分派一個顏色。深藍色 (COLOR = 2) 與黃色 (COLOR = 13) 是很合適的顏色，因為它們在彩色電視螢光幕上看起來很醒目，而且即使在黑白電視螢光幕上仍然可以輕易地分辨出來，因為深藍色看起來比黃色要暗。

您可以直接寫

```
10 IF P = 0 THEN COLOR = 2
20 IF P = 1 THEN COLOR = 13
```

不過，較佳的作法是

```
10 COLOR = 2 + 11 * P
```

在兩種顏色 C1 與 C2 之間交互變化的通式如下：

```
10 COLOR = C1 + (C2 - C1) * P
```

假定您決定令 P 在 1 與 2 之間交互變化，不是 0 與 1，則答案應為：

```
10 COLOR = 2 + 11 * (P - 1)
```

，而其通式則為

```
10 COLOR = C1 + (C2 - C1) * (P - 1)
```

若要通知遊戲者“輪到您了”，您可以寫

```
50 PRINT "YOUR TURN, PLAYER ";P
```

或者，您可逕行稱呼遊戲者的顏色，最簡單的方法是

```
10 IF P = 0 THEN PRINT "BLUE"
20 IF P = 1 THEN PRINT "YELLOW"
```

6 自己設計 Apple 遊戲

不過，較佳的解法是：

```
10 C$ = "BLUE YELLOW"
20 FOR P = 0 TO 1
30 PRINT "YOUR TURN, "; MID$(C$, (1 + 6 * P), 6)
40 NEXT P
```

C\$ 包含兩個顏色名稱分字串 (substring) 所組成的一個字串。因為它們必須具有相等長度，因此在 BLUE 後面補上兩個空格，使它們都成為兩個 6 一字元的分字串。第一個分字串是 "BLUE"，由 C\$ 的第一個字母開始，第二個分字串是 "YELLOW"，由 C\$ 的第七個字母開始。這兩個分字串由 30 那一列 PRINT (印) 出來。記住

MID\$(C\$, A, B)

選擇的是由 C\$ 的第 A 個字元 (character) 開始而長度為 B 個字元的分字串。如果 $P = 0$ ，則 30 那一列中的 $(1 + 6 * P)$ 求出之值為 1，而如果 $P = 1$ ，則其值為 7。因此，程式執行的結果如下：

```
RUN
YOUR TURN, BLUE
YOUR TURN, YELLOW
```

C\$ 包含兩個 N 一字元分字串的通式為：

MID\$(C\$(1 + N * P), N)

例如，假定 0 號遊戲者被指派的顏色為藍色 (BLUE)，而 1 號遊戲者為紅色 (RED)，則 C\$ 包含的是兩個 4 一字元的分字串，因此 $N = 4$ ：

```
10 C$ = "BLUERED "
20 FOR P = 0 TO 1
30 PRINT "YOUR TURN, "; MID$(C$, (1 + 4 * P), 4)
40 NEXT P
```