

Apple Pascal 遊戲

許志堅 譯



協群科技出版社

APPLE PASCAL 遊戲

許志堅 譯

協群科技出版社

APPLE PASCAL 遊戲

編譯者：許志堅
出版：協群科技出版社
發行：協群科技出版社
印刷者：廣源印務局
香港中環卑利街684號二樓
青山道875號工廠大廈

定價：H.K.\$ 30.00

前 言

本書內的每一個 PASCAL 遊戲都包括兩部份。第一部份是“遊戲說明”。這一部份僅僅說明遊戲的規則和玩法，即使是完全不懂電腦的人也可以按圖索驥，享受電腦遊戲的樂趣。絕大部份的遊戲都有一個範例供您參考。

第二部份是“程式說明”。其中包含了程式的結構圖 (structure chart) 和程式列表 (program listing)。同時我們也詳細地說明了程式的工作原理。我們希望您一面閱讀程式工作原理，一面參閱程式列表，以收事半功倍之效。

圖 1 與圖 2 說明程式結構圖與程式的關係。圖 1 僅僅勾勒出程式的組織輪廓而沒有任何指令。其中包括下述的特點：

- 主程式 (main program) 放在程式的最後頭。它會召用 (call) 一些程序 (Procedure) 和功能 (Function) 以控制程式的步驟和動作。
- 定義在 CONST, TYPE, VAR 宣告區中的常數和變數可能是整體變數 (global) 或局部變數 (local)。整體變數可用在程式的任何地方；而局部變數只能用在它所定義的程式段 (程序或功能) 之中。
- 運用 USES 敘述以便使用一些特殊的單元。例常用的 APP LESTUFF 單元 (包括 RANDOM, RANDOMIZE, 和 KEYPRESS), 和 TRANSCEND 單元 (提供一些超越函數 transcendental Function)。
- 利用說明敘述的分界號 (* 和 *), 以各種不同的方式來註解程式。

而圖 2 是圖 1 程式的結構圖。它僅僅說明程式中各程序和功能的隸屬關係。例主程式召用程序 ONE 和程序 THREE，而程序 ONE 又召用程序 TWO。

```
PROGRAM EXAMPLE;
USES APPLESTUFF, TURTLEGRAPHICS, TRANSCEND;
CONST
    (* GLOBAL CONSTANT DECLARATIONS *);
TYPE
    (* GLOBAL TYPE DECLARATIONS *);
VAR
    (* GLOBAL VARIABLE DECLARATIONS *);
PROCEDURE ONE (      (* FORMAL PARAMETERS OF
                      * PROCEDURE ONE, INCLUDING
                      * VALUE PARAMETERS,
                      *)
                VAR   (* AND VARIABLE PARAMETERS *)
                );
VAR
    (* LOCAL VARIABLES FOR PROCEDURE ONE *);
PROCEDURE TWO;      (*
                    * PROCEDURE TWO IS LOCAL
                    * TO PROCEDURE ONE.
                    *)
BEGIN
    (*
     * MAIN BODY OF PROCEDURE TWO.
     *)
END;
BEGIN
    (*
     * MAIN BODY OF PROCEDURE ONE;
     * CONTAINS A CALL TO PROCEDURE TWO.
     *)
END;
```

圖 1 : Pascal 程式的輪廓

```

FUNCTION THREE (      (*
                        * FORMAL PARAMETERS OF THREE.
                        *)
                  ):    (* TYPE OF VALUE THAT THREE RETURNS *);

BEGIN
  (* BODY OF FUNCTION THREE *)
END;

BEGIN
  (*
  * MAIN PROGRAM.
  * CONTAINS CALLS TO
  * PROCEDURE ONE AND
  * FUNCTION THREE.
  *)
END.

```

Figure 1: Skeleton of a Pascal Program (cont.)

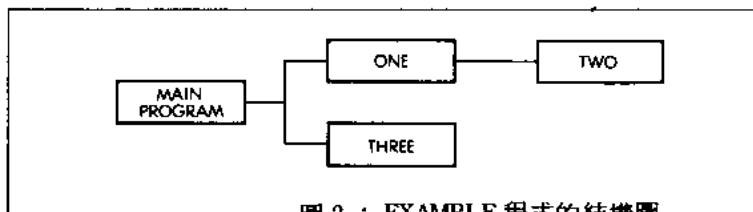


圖 2：EXAMPLE 程式的結構圖

爲了配合各種程度的讀者，我們將內容分成四大部份。愈後頭的部份愈具有挑戰性。

第一部份包含一些簡單的遊戲。您可能在其他的書籍中（例 BASIC 遊戲）見過類似的遊戲。但是您可以從這一部份的遊戲中學習到 PASCAL 語言的技巧，窺視出 PASCAL 語言的特性。

第二部份是一些較複雜的遊戲。您可以從這一部份的程式中學到 PASCAL 語言的特殊語法 (syntax) 和資料結構。例：

- 資料錄形態 (record type) — 賽馬遊戲。
- 集合形態 (set type) 和集合運算元 (set operator) — 屠龍遊戲。
- 指標形態 (Pointer type) 和鏈連串列 (linked list) — 巴斯喀三角形。
- 純量形態 (scalar type) — 蛇屋遊戲，黑箱遊戲。
- 檔案形態和檔案處理功能 — 吊人遊戲。
- 遞歸程序 (recursive procedure) 的召用 — 電話遊戲。
- 字串轉換和輸入測試 — 登路遊戲。
- CASE 敘述 — 天牌遊戲。

第三部份介紹 APPLE 的畫圖程式 TURTLEGRAPHICS。我們利用兩個遊戲說明了畫圖程式中大部份的程序和功能。

第四部份是一個大程式 — 紙牌遊戲。我們把這一個大程式分成三個檔案存在 APPLE 機器中。藉此，我們也說明了 PASCAL 編譯元 (compiler) 的“檔案包含” (include- File) 特性和 FORWAPD 敘述。同時您也可以學到 PASCAL 語言各種指令的用法。

書末的附錄對 APPLE 語言做一個簡要的摘要，您可以將它當做學習教材或參考資料。

當您在 APPLE II 計算機上執行過這些程式之後，您可以輕易地再將它們轉換到其他有 UCSD PASCAL 系統的計算機上。

Douglas Hergert
Berkeley, California
August 1981

目 錄

第壹部份——簡單的遊戲

| | | |
|-----|----------|----|
| 遊戲一 | 猜數字遊戲 | 1 |
| 遊戲二 | 運氣遊戲 | 7 |
| 遊戲三 | 骰子遊戲 | 16 |
| 遊戲四 | 數學遊戲 | 23 |
| 遊戲五 | 倍數還本遊戲 | 31 |
| 遊戲六 | 黑白數字猜謎遊戲 | 37 |
| 遊戲七 | 閃爍遊戲 | 44 |
| 遊戲八 | 方塊字遊戲 | 49 |
| 遊戲九 | 拿火柴遊戲 | 61 |

第貳部份——較複雜的遊戲

| | | |
|------|--------|-----|
| 遊戲十 | 賽馬遊戲 | 69 |
| 遊戲十一 | 天牌遊戲 | 79 |
| 遊戲十二 | 數字轉換遊戲 | 105 |
| 遊戲十三 | 砲擊遊戲 | 115 |
| 遊戲十四 | PASCAL | 122 |
| 遊戲十五 | 牌九遊戲 | 132 |
| 遊戲十六 | 戰略遊戲 | 145 |
| 遊戲十七 | 黑箱遊戲 | 169 |
| 遊戲十八 | 電話遊戲 | 181 |
| 遊戲十九 | 蛇屋遊戲 | 191 |

| | | |
|-------|--------|-----|
| 遊戲二十 | 吊人遊戲 | 203 |
| 遊戲二十一 | 生存遊戲 | 215 |
| 遊戲二十二 | 二十一點遊戲 | 229 |
| 遊戲二十三 | 屠龍遊戲 | 245 |
| 遊戲二十四 | 登陸遊戲 | 261 |

第叁部份——使用TURTLEGRAPHICS畫圖的遊戲

| | | |
|-------|------|-----|
| 遊戲二十五 | 獵雁遊戲 | 283 |
| 遊戲二十六 | 畫圖遊戲 | 293 |

第肆部份——紙牌遊戲

| | | |
|-------|------|-----|
| 遊戲二十七 | 紙牌遊戲 | 301 |
|-------|------|-----|

附 錄

| | |
|-----------------|-----|
| A — 保留字及建入功能 | 350 |
| B — PASCAL 語言摘要 | 353 |

遊戲一

猜數字遊戲



遊戲說明

這個以 Pascal 寫成的計算機猜數字遊戲玩法非常簡單：計算機從 1 與 100 之間選出一個數字，然後請你猜這一個數字，你每猜一次，計算機就會給你一些提示：

(太高)

或

(太低)

或 (猜中了) 。

CORRECT!!!!

你總共可以猜六次；如果六次都沒有猜中，計算機就會自動將謎底顯示出來。如果你想繼續猜另一個數字，你不須再重新啟動該程式，只須回答下一個問題 Y 就可以了：

WANT TO TRY AGAIN?

如果你想休息，則必須回答 N。

```
THIS IS MATCH IT

I WILL CHOOSE A NUMBER BETWEEN 1 AND 100
YOU WILL TRY TO GUESS THAT NUMBER
IF YOU GUESS WRONG, I WILL TELL YOU IF YOU GUESSED
TOO HIGH, OR TOO LOW
YOU HAVE 6 TRIES TO GET THE NUMBER

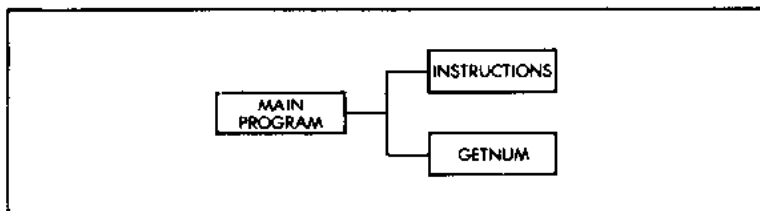
ENJOY

YOUR GUESS? 50
TOO HIGH
YOUR GUESS? 25
TOO HIGH
YOUR GUESS? 12
TOO LOW
YOUR GUESS? 18
CORRECT!!!!

WANT TO TRY AGAIN? Y
YOUR GUESS? 50
TOO HIGH
YOUR GUESS? 75
TOO LOW
YOUR GUESS? 37
TOO HIGH
YOUR GUESS? 32
TOO LOW
YOUR GUESS? 35
CORRECT!!!!

WANT TO TRY AGAIN? N
```

程式說明



我們利用一個建入程序 (built-in procedure)
 RANDOM 從 1 與 100 之間選出一個數字；所以主程式一開始
 就召用 (call) Pascal 的程序 RANDOMIZE。(它會產生
 一串亂數)。隨後主程式召用另一個程序 INSTRUCTIONS 以
 印出遊戲說明；之後，主程式進入一個 WHILE 迴環 (loop)
 直到猜謎者於遊戲結束時回答一個 " N "：

```
WHILE CH <> "N" DO ...
```

WHILE 迴環先將 GUESS 和 NUMTRY 的初值定為零。
 NUMTRY 是猜目前這一個數字已經用的次數，每召用
 GETNUM 程序之後，NUMTRY 就會加 1：

```
NUMTRY := NUMTRY + 1
```

謎底是由下一個指述 (statement) 得到的：

```
NUMBER := (RANDOM MOD MAXNUM) + MINNUM
```

程式每召用一次 RANDOM 就會得到一個介於 0 和 32767 之間
 的整數，而 mod 運算能執行兩個整數的除法而保存其餘數因為
 MAXNUM 的值是 100，所以 RANDOM MOD MAXNUM 會得
 到一個整數，其範圍由 0 到 99。再加 1 (MINNUM 的值)
 的目的是調整它的範圍 (由 1 到 100)。

亂數 (random number) 一經選定之後，計算機就可以接
 受猜謎者的答案。這一個 WHILE 指述包含了兩個條件
 (condition)：

```
(NUMTRY < MAXTRIES) AND (GUESS <> NUMBER)
```

GETNUM 不斷地被召用，一直到謎底被猜中或是猜謎者已
 經猜了六次為止。

程式中最複雜的部份是 GETNUM 程序，其中最困難的部份

就是驗證輸入值，我們用指述

```
GET(INPUT);
```

和內蘊定義的 (implicitly defined) 緩衝器變數 (buffer variable) INPUT 來驗證輸入值。輸入的第一個字元 (character) 被派定 (assign) 到 INPUT^1，然後再以下一個指述驗證它是否是 1 到 9 之間的整數：

```
IF NOT (INPUT^1 IN ['1'..'9']) THEN . . .
```

如果輸入值不合 (1 到 9 以外的任何字元)，計算機就會顯示一個錯誤的信號：

```
ILLEGAL NUMBER
```

然後利用 READLN 讀入不合的輸入，再遞歸 (recursive) 召用 GETNUM 以丟棄不合的輸入；因為重頭又執行 GETNUM 程序，因此又重新要求新的輸入。如果輸入是合理的值，就會被讀到變數 GUESS 之中：

```
READLN(GUESS);
```

另外還有一個驗證測試，那就是 GUESS 必須小於或等於 MAXNUM。

```
IF GUESS > MAXNUM . . .
```

如果不合於這個條件，計算機也會輸出一個錯誤信息，然後再遞歸召用 GETNUM。

最後，當 GUESS 驗證通過之後，程式就必須比較輸入值和謎底並輸入“太低”，“太高”，或“正確”的提示。在

GETNUM 程序末端，我們用一連串的巢式 (nested)

```
IF/THEN/ELSE 指述來完成。
```

隨後又回到主程式，NUMTRY 加 1，表示又猜了一次，如果猜了六次都沒有猜中，程式就會自動顯示謎底，同時詢問猜謎者是否要繼續猜下一個謎題：

```
WANT TO TRY AGAIN?
```

```

(*
 * GUESS IT
 * A COMPUTER GUESSING GAME
 *)
PROGRAM GUESSIT (INPUT,OUTPUT);
USES APPLSTUFF;
CONST
  MAXTRYS      = 6;          (* USER ALLOWED 6 TRIES TO GUESS NUMBER *)
  MAXNUM       = 100;       (* MAXIMUM NUMBER *)
  MINNUM       = 1;        (* MINIMUM NUMBER *)
VAR
  CH : CHAR;                (* USED TO ANSWER QUESTIONS *)
  GUESS : INTEGER;         (* HUMAN'S GUESS *)
  NUMBER : MINNUM..MAXNUM; (* THE NUMBER ITSELF *)
  NUNTRY : 0..MAXTRYS;     (* NUMBER OF TIMES HUMAN HAS GUESSED *)

(*
 * INSTRUCTIONS
 * PRINTS OUT THE INSTRUCTIONS
 *)
PROCEDURE INSTRUCTIONS;
BEGIN
  WRITELN('THIS IS MATCH IT');
  WRITELN;
  WRITELN('I WILL CHOOSE A NUMBER BETWEEN 1 AND 100');
  WRITELN('YOU WILL TRY TO GUESS THAT NUMBER');
  WRITELN('IF YOU GUESS WRONG, I WILL TELL YOU IF YOU GUESSED');
  WRITELN('TOO HIGH, OR TOO LOW');
  WRITELN('YOU HAVE ',MAXTRYS,' TRY TO GET THE NUMBER');
  WRITELN;
  WRITELN('ENJOY');
  WRITELN;
END; (* INSTRUCTIONS *)

(*
 * GETNUM
 * GET A GUESS FROM THE HUMAN,
 * WITH ERROR CHECKING
 *)
PROCEDURE GETNUM;
BEGIN
  WRITE('YOUR GUESS ');

  (*
   * WAIT FOR A CHARACTER
   *)

  GET(INPUT);

```

6 APPLE PASCAL GAMES

```
(*
 * CHECK TO SEE IF THE CHARACTER
 * COULD BE A NUMBER
 *)
IF NOT (INPUT^ IN ['1'..'9']) THEN
BEGIN
  READLN(INPUT);
  WRITELN('ILLEGAL NUMBER');
  GETNUM          (* RECURSE FOR LEGAL INPUT *)
END
ELSE
BEGIN
  READLN(GUESS);
  IF GUESS > MAXNUM THEN
  BEGIN
    WRITELN('ILLEGAL NUMBER');
    GETNUM
  END
  ELSE
  BEGIN
    IF GUESS > NUMBER THEN
      WRITELN('TOO HIGH')
    ELSE
      IF GUESS < NUMBER THEN
        WRITELN('TOO LOW')
      ELSE
        WRITELN('CORRECT!!!!')
    END
  END
END; (* GETNUM *)

(*
 * MAIN
 *)
BEGIN
  INSTRUCTIONS;
  RANDOMIZE;
  WHILE CH <> 'N' DO
  BEGIN
    GUESS := 0;
    NUMTRY := 0;
    NUMBER := RANDOM MOD MAXNUM + MINNUM;      (* GET NUMBER *)
    WHILE ( NUMTRY < MAXTRYS ) AND ( GUESS <> NUMBER ) DO
    BEGIN
      GETNUM;
      NUMTRY := NUMTRY +1
    END;
    WRITELN;
    IF GUESS <> NUMBER THEN
      WRITELN('THE NUMBER WAS ',NUMBER);
    WRITELN;
    WRITE('WANT TO TRY AGAIN? ');
    READLN(CH)
  END
END.
```

遊戲二

運氣遊戲



遊戲說明

歡迎你來到豪華遊樂廳。這裡的環境雖然有些俗雅，但是氣氛却非常舒適融洽。你只要花\$ 1.20元就能玩一盤運氣遊戲，而飲料是免費供應的。

遊戲的規則非常簡單。首先你必須在1與80之間的整數中選出8個幸運數字；在這同時，計算機也會隨機地在1與80之間的整數中選出20個數字。如果你選的8個幸運數字中有5個或以上的數字同時也出現在計算機所選的20個數字之中，你就是贏家，以下就是你可能贏的錢：

5個數字一樣：\$ 10

6個數字一樣：\$ 100

7個數字一樣：\$ 2,200

8個數字一樣：\$ 25,000

希望整個下午你都待在遊樂廳裡盡情享受，因為你待得愈久，你成爲一個大贏家的機會也愈大，無論你玩多久，計算機會自動地替你累加計算你所贏的錢數。

8 APPLE PASCAL GAMES

```
YOUR SPOT? 12
YOUR SPOT? 16
YOUR SPOT? 23
YOUR SPOT? 27
YOUR SPOT? 35
YOUR SPOT? 44
YOUR SPOT? 65
YOUR SPOT? 77
THE COMPUTER CHOOSES
  4  6  8 14 15 20 26 28 33 34
 39 43 46 57 69 70 74 76 77 80
SORRY, BUT YOU ONLY MATCHED 1
```

SO FAR YOU HAVE LOST \$ 1.20

```
WANT TO PLAY KENO AGAIN? Y
YOUR SPOT? 27
YOUR SPOT? 33
YOUR SPOT? 78
YOUR SPOT? 17
YOUR SPOT? 22
YOUR SPOT? 19
YOUR SPOT? 55
YOUR SPOT? 23
THE COMPUTER CHOOSES
  1 15 17 19 22 23 26 27 30 33
 37 38 48 51 52 60 62 63 68 80
YOU LUCKY PERSON, YOU HAVE 6 MATCHES
THAT MEANS YOU HAVE MADE 100
```

YOUR TOTAL MONEY IS \$ 97.60

```
WANT TO PLAY KENO AGAIN? Y
YOUR SPOT? 34
YOUR SPOT? 25
YOUR SPOT? 27
YOUR SPOT? 78
YOUR SPOT? 66
YOUR SPOT? 15
YOUR SPOT? 32
YOUR SPOT? 66
YOU HAVE ALREADY CHOSEN SPOT 66
YOUR SPOT? 65
THE COMPUTER CHOOSES
  4  8 10 17 18 22 23 26 30 36
 37 38 49 55 57 62 66 67 77 80
SORRY, BUT YOU ONLY MATCHED 1
```

YOUR TOTAL MONEY IS \$ 96.40

WANT TO PLAY KENO AGAIN? N

```
ALL RIGHT THEN LEAVE, SEE IF I CARE!!!
YOU MADE $ 96.40
```