

APPLE II

繪圖遊戲系列①

施威銘 校訂
林和賢 編著

APPLE II 圖形遊戲系統設計

GAME! GAME! GAME!

- J ■ 登陸星球
- 金 撲 克
- 迷 宮
- PIANO
- 螺 旋

大蜘蛛……



APPLE II

圖形遊戲系統設計

施 威 銘 編

偉達電腦圖書供應中心

〈版權聲明〉

登記證號碼：8316338-000-03

本社擁有其翻譯、編著版權，可以在香港、星加坡、印尼、馬來西亞、菲律賓、泰國、台灣，等東南亞地區內有權使用及授權其他人等使用所述之權利。

All Rights Reserved, certificate No8316338-000-03

No part of this book maybe reproduced in any form without permission in writing from the publisher ©victor BK 1983

Hong Kong, singapore, malaysia, Thailand, Philippine, Indonesia, Taiwan.

出版：偉達電腦圖書供應中心

香港北角英皇道 193-203號英皇中心大廈二樓L座

承印：高雅印刷有限公司

香港鰂魚涌華蘭路16-18號

萬邦工業大廈九樓C座

序

許多人是因為精彩有趣的 GAME 而進入微電腦海闊天空的領域。通常是由玩 GAME 開始，繼而對 GAME 程式的設計發生好奇，而終於造成不眠不休的研究狂熱。一般的情形是自己設計了一點簡單的程式，但終究覺得沒有市售成品的精彩好玩，速度上也相差甚多，好奇的人會拿一片小蜜蜂的磁片（磁帶）來觀摩其程式設計的技巧，往這方向走的，就開始一連串的解保護、造預移圖序、圖形壓縮等屬於機械語言的長期奮鬥了。

不過，相當多的人都因為陌生的機械語言而無法更上層樓，進一步追求研習的樂趣。但是，使用 BASIC 一樣是可以造出優秀的遊戲來的！畢竟語言只是一種工具，如果因為迷失於語言的困境中，而不能對程式的系統結構善加安排，說起來真是本末倒置可惜的事情呢！對於初學者而言，經由易於瞭解的 BASIC 語言來設計 GAME 程式，使得有足夠的精神來對程式結構策略加以研究，可說是最適當不過的事情了！

這本書便是以這樣的想法來編寫的，全書共分成兩大部份，前半部份是為學習者提供技術上的研習與參考，其中第一章是將一般書籍較少提到的 BASIC 繪圖技巧予以解說，在第二章當中，為了解決相當耗時的圖形表設計，我們介紹了自動造表程式來簡化程式設計的工作，所以讀者可以很方便的使用 DRAW、

XDRAW、ROT 等圖形而不必煩惱圖形表的設計。第 5、6 章則介紹了第 2 部份程式的架構，以及製造磁帶系統的詳細說明。在第二部份當中，我們共選輯了 10 個有趣的遊戲，這些程式均是經過了系統化的組合，你會發現除了故事內容的差異之外，整體上的構造及設計手法，都是有其一定的脈絡可尋的，希望經由對於這些程式的瞭解，能夠有助於你對於遊戲程式設計的學習。

總之，請開始閱讀這本書吧！相信你一定能在毫無困難的情況下，鍵入每一則程式而樂趣無窮的學習著！

目錄

第一部份 技術講話

第 0 章	初學者的下一步.....	1
	0 - 0 不知道問題在那裏！	1
	0 - 1 問題在那裏	2
	0 - 2 本書結構	3
第 1 章	活用 APPLE 的圖形表.....	5
	1 - 0 APPLE II 的 3 種繪圖方式	5
	1 - 1 APPLE 的圖形表	7
	1 - 2 圖形表怎麼用？	8
	1 - 3 有關圖形表的特殊位址	9
	1 - 4 頁次輪換的技巧	16
第 2 章	自動造表程式.....	19
	2 - 0 圖形表為什麼麻煩？	19
	2 - 1 磁碟機使用者的自動造表程式	23
	2 - 2 磁帶使用者的自動造表程式	36
第 3 章	戲劇效果的文數字形圖形表.....	47
	3 - 0 字形圖形表	47
	3 - 1 字形圖形表的使用法	51

第 4 章	音響的效果.....	55
4 - 0	一個發聲的例子.....	56
4 - 1	另一個聲音程式.....	58

第二部份 程式輯

第 5 章	如何製造您的程式(一).....	63
第 6 章	如何製造您的程式(二).....	67
第 7 章	大蜘蛛.....	75
第 8 章	外星人登陸(一).....	91
第 9 章	分秒不差.....	107
第10章	PIANO.....	129
第11章	迷宮.....	149
第12章	摘星星.....	165
第13章	螺旋.....	179
第14章	外星人登陸(二).....	197
第15章	精打細算.....	217
第16章	金撲克.....	233
第17章	節目單.....	261
第18章	創造自己的 GAME.....	269
參考書目		273

第 0 章

初學者的下一步

0-0 不知道問題在那裏！

相信電腦的愛好者皆會同意，無論您學了電腦一星期、一個月甚或一年，總會有那麼一個時候，當您面對著美麗而精緻的電腦畫面，不禁要問：這樣的程式是如何設計出來的呢？有哪些書本或資料可以教我有關的知識呢？而我是否也可以來設計這樣的程式？

於是，您便嘗試寫了一些程式，結果當然也有一番體會……

問題的癥結——我們的問題是因為不知道問題在哪裏

那麼，問題在哪裡？我們面對的困難是什麼呢？說起來，這也是電腦快速流行之後而普遍引起的一個現象：當學完 PRINT、FOR—NEXT、GOSUB、……之後的下一步，為什麼竟然是舉筆維艱的寫不出一個有用的程式呢！

包括作者在內的非科班出身的電腦人都曾經有這樣的經歷吧！那就是：我們所面對的問題是因為不知道問題在哪裡！

0-1 問題在那裏？

對一個沒有經過正規訓練（？）的電腦愛好者來說，設計程式就好像沒有經過寫作訓練的人寫文章一般的，只憑靈感而不考慮格式、結構、文體吧？誠然，靈感可能寫出動人的詩篇，而經世的論述卻必須真實的才學了！

所以，這便是問題的所在之一：我們必須學習如何規劃、安排、有系統的來設計程式，一旦程式設計有了一定的規範、有了可循的程序，那種不知如何着手或寫不出較大程式的困難，便迎刃而解了，這種結構化的程式設計方法（不只程式本身結構化，程式設計過程也結構化），尤以本身最不結構化的 BASIC* 及組合語言** 最為需要。

另外一個程式設計上的問題是，系統熟悉運用的程度，譬如，一直令初學者既好奇又困惑的 PEEK、POKE 指令便是一個例子，其實 PEEK、POKE 並不是什麼難以瞭解的指令，只不過是因為這些指令可以直接的改變系統狀態，而造出奇特的結果，如此一來，便使得對電腦系統構造不熟悉的初學者感到迷惑了。

明白了問題之後，我們要做的便是：熟悉 APPLE 繪圖方面的特殊功能，並加以結構化的運用。

* 結構化 BASIC 閔國田譯 波前電腦圖書公司

** APPLE II 徹底研究(含動畫技巧篇) 施威銘著 波前電腦圖書公司

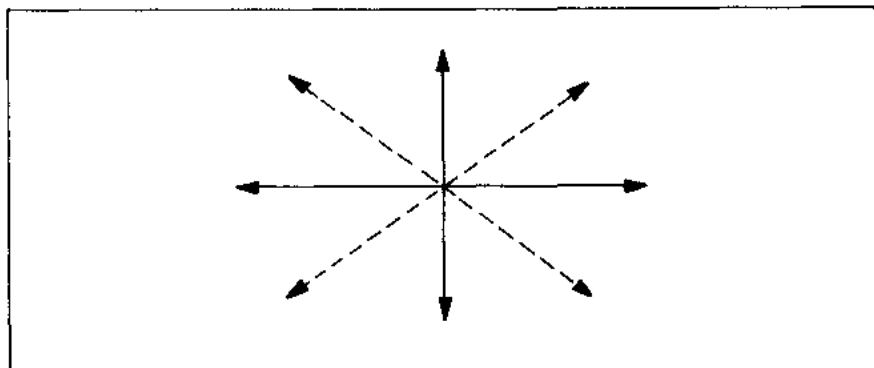
0-2 本書結構

現在，我們可以來介紹一下本書的內容了。本書共分為兩大部份，前半部份是為後半部份的程式設計而做的技術提要，其主要目的是在對於已熟悉 Applesoft 的讀者提供一些建議，譬如說，大家都知道 Applesoft 的圖形表繪圖指令，但使用起來却覺得很困難。到底要如何有效的使用圖形表繪圖呢？便是本書要討論的一個問題了。另外，對於磁碟機的使用者，可以在程式執行當中存取磁片資料，是一件很方便的事情，可是磁帶錄音機的使用者呢？本書也介紹您一種有趣的磁帶程式製造法，你也可以造出有保護的商品磁帶呢！

至於後半部份的程式，則是一系列有趣而系統化設計的遊戲，全部程式均是以前述的圖形表方法來造出吸引人的畫面動作，你可以經由這些程式而體會到向量繪圖法（圖形表法）的極限發揮。不過，除了遊戲的趣味性之外，我們更希望您能夠瞭解，一個程式的結構化與否，是如何的影響到其是否為一個優秀的作品呢！

第 1 章

活用 APPLE 的圖形表



1-0 APPLE II 的 3 種繪圖方式

基本上 APPLE 的繪圖方式有三種，各種方法均有其適用的場合及優缺點，現在分述如下：

1. 向量繪圖法 (XDRAW 與 DRAW)

這是大家熟悉的一種方式*，只要設計者繪出圖形，依規則造好圖形表，然後使用 Applesoft 內含的繪圖指令 XDRAW 及 DRAW 便可以將圖形畫在螢幕上。這種

* 參考資料 Ref. 1

方法的好處是不需要額外的繪圖副程式，只要將圖形表附加到程式內，以 BASIC 便可以畫出圖形來，並且使用 ROT 及 SCALE 更可以令圖形旋轉，放大。其中 DRAW 是單純的繪圖指令，而 XDRAW 則是具有 EOR（斥或邏輯）特性的繪圖指令，因此如果第一次 XDRAW 是用來畫出圖形的話，則第二次在原位置上使用 XDRAW，會將第一次所畫的圖形消去，但保留了背景。

向量繪圖法的缺點是：①只能畫單色圖形，也就是說，在一個圖形內部只有一種顏色。②不易畫實心圖形，這是向量繪圖的先天限制。③速度較慢，尤其是畫較大圖形時，更形明顯。④圖形表設計、換算，費時費力。

所以，通常向量繪圖法是使用在較小圖形、較少圖形、速度要求不嚴格的場合，不過，巧妙的運用，却也一樣能夠造出優秀而吸引人的效果來，本書的程式便是幾乎全部以這種方法來設計的。至於圖形表設計、換算的問題，我們會令 Apple 自動的來為我們處理。

2. HPlot 副程式繪圖法

這是直接利用 Apple 內部的 HPlot 及 HPlot TO 等指令所屬的 ROM 副程式來繪圖的一種方法。這種方法，由於使用了機械語言程式，因此速度上比較快（因為不必經過 DRAW、ROT、SCALE 等副程式處理），並且對於圖形上的多重彩色、實心圖形等問題，也可以獲得合理的解決。

3.位元映射繪圖法

雖然HPlot 方式的繪圖速度已獲改善，但是由於HPlot 是一個考慮到多方面適用性的副程式，因此必然會相互遷就而使得執行的效率（速度）有所折扣。位元映射繪圖法，是直接將圖形存入到畫面的記憶位址內，而立即表現到畫面上的繪圖方法。位元映射繪圖的方式也分成好幾種，在APPLE II 微電腦微底研究第5冊之中曾經介紹了一種利用畫面位址查表法及預移圖序法所構成的高效率繪圖法，有興趣讀者可參考之。這種位元映射繪圖法幾乎是大多數現代專業動畫程式所採用的方法了。

1-1 APPLE 的圖形表

前述的3種繪圖法，其中以向量繪圖的圖形表法最適合在BASIC 程式中使用，這是因為Applesoft 當中已含有ROT、SCALE、XDRAW、DRAW等指令，只要將圖形表造出來便很方便使用了。既然本書是以BASIC 為主要語言，所以我們便來探討一下如何活用Apple的圖形表。

許多初學Apple的人，一開始都會因為圖形表指令具有旋轉及放大的繪圖功能而甚感興趣，可是嘗試了幾次之後，却又多因圖形表本身繁複的換算手續而放棄了。其實，最初幾次動手來做換算的工作是必須的，由此，您才可以體會出其基本規則，但是真正的實用設計，却大可不必這樣一步一點的來加以計算，這種

工作，我們將在第二章介紹一個自動造表程式來為我們代勞。

所以，請參照Apple原始說明，依所學的例子畫個圖形看看，此後，您就不必再做這種工作了！因為，這只不過是Apple語言系統的一種規則，對電腦技術的學習，並無多大幫助！

1-2 圖形表怎麼用？

只要有了自動造表程式，可以自動為我們換算圖形表之後，我們就可以不要問“圖形表怎樣做”，而是問“圖形表怎麼用”了！

使用圖形表，首先要考慮的是圖形表存放的位址，如果是太長的表格（如第3章的字形表），則不適於存放在第3頁甚或是高解像畫面之前（小於\$2000）的位址，因為這樣可能影響BASIC程式的使用空間，另外一些陳列變數的存放區也要特別注意勿與之重疊。在本書當中，我們選擇高解像第2頁畫面之後的區域（大於\$5FFF）來存放圖形表，字形圖形表是由\$61A8（25000）開始存放，而圖案圖形表則由\$6978（27000）開始。

兩組以上的圖形表

通常，圖形表是只有一組的，但是像本書這樣系統化的一次設計出多個程式的場合，總是有如文數字形這種幾乎每則程式均可共同使用的圖形表。為了令這種“公用的”圖形表方便於附加到各個程式上，我們就必須將它自成一組的與程式個別的圖案圖

形分開來。這便是我們將字形圖形表置於\$ 61A8 而圖案圖形表則存放於\$ 6978 的原因了。至於為什麼採用這兩個奇怪的位址做圖形表起始點呢？因為它們的十進位值分別是 25 000 及 27 000，在BASIC 程式中的計算較方便。

但是有了兩組不同的圖形表又怎麼辦呢？

1-3 有關圖形表的特殊位址

要使用Apple的圖形表指令繪圖，除了將圖形表造好、存入記憶體之外，還要告訴Apple圖形表的起始位址才行。Apple的圖形表起始位址指位器是（\$E8，\$E9）=（232,233），所以，如果你要畫出英文字母，就必須以告訴Apple說我現在要

```
POKE 232,168
```

```
POKE 233,97
```

DRAW（或XDRAW）的圖形是在由\$ 61A8 開始的這組圖形表當中。其中232所存的是\$ 61A8 的低位元組（\$A8 = 168），233所存的是\$ 61A8 的高位元組（\$ 61 = 97）。

碰撞旗號

除了（\$E8，\$E9）這組位址之外，Apple還有一個與DRAW，XDRAW有關的位址：\$EA=234。這個緊接在（\$E8，\$E9）的位址，有一個很有趣的功能，那就是每次在執行DRAW指令時，如果發現所欲畫點（位元）的位置已經畫了點（該位置所對應的位元值=1）或所欲消點的點已被消去（該位置所對應

的位元值 = 0)，則 \$EA 的內含值便加一，因此 \$EA 內含值的增加值便是碰撞次數了。譬如一艘白色的太空船在黑色的背景上移動，事實上便是將黑色的背景畫上白色的圖點（將 bit = 0 變成 bit = 1）。這時如果發現所欲畫點的位置其位元值為 1，則必然是碰上黑色背景以外的物體了（因為若是黑色背景，位元值為 0 而不是 1）。

爲了說明碰撞旗號的功能，此處特別舉了一個自動塗色的程式，您可以將之與 APPLE II 徹底研究第五冊所提到的另一個自動塗色程式比較一下其優缺點。

自動塗色程式

```
DLIST
```

```
10 REM *****
20 REM * COLOR-FILLER *
30 REM *****
40 REM
50 REM
100 REM =====
110 REM INITIALIZE
120 REM =====
130 HGR : SCALE= 1: ROT= 0: HCOLOR= 3
140 MOVEREG = - 2613
150 HY = 226
160 CLR = 62450
170 TABLE = 768: REM THIS VALUE RELOCATES EVERYTHING
```