

APPLE II

彻底研究(之五)



APPLE

微電腦徹底研究

(五)

北方电脑公司信息资料部

唐凌 序

假如有一天，你用你自己的腦筋，寫了一本書，送到出版社去，或則只是費力地寫好一篇文章，投到某專業性雜誌去發表，你可能遭到一項極大的困難——出版商或雜誌的編輯會向你索取「原文」。

出版商或雜誌編輯之希望文有出處的心理，是不難理解的，一是沾光，一是如有謬誤可一推了之。

事實上，這樣的心態，普通地存在於現居台灣的每一個人心裡，而不只是出版界。

做事有個依據是對的，然而任何依據本身不可能是真理，即或是真理，在被引用的過程也可能造成扭曲。

下面就是一段隨處可見而扭曲得可笑的例子：

「在半計數 (Counter halves) 之間，外來的顫動 (Jumper) 必須供給。

「若經由 Clock 1 進入，而且 Q 8 跳躍至 Clock 1，計數比重為 1 - 2 - 4 - 5，Q 1 是位元比重最大的輸出而且在輸出端有一個對稱的方波。……………」

這是一段對 7490 動作的描述，如果你對 7490 毫無概念，那麼看了這一段，真的會令你「顫動」不已。

不幸的事情是，對這樣荒唐的謬誤，出版商給予容忍，讀者也同樣給予容忍。於是成千上萬的學子們，只好一天天地「顫動」下去。

另外一種情況也是很可悲的，也就是不知從什麼時候開始，大學生都喜歡抱厚厚的洋裝書，使得一些類如三用表用法、OTL、OCL 設計或 CMOS 玩具製作之類的小書，有興趣的人只好偷偷地看，總覺得這些小書，好像很不入流一樣。

我與施威銘先生只是初識，當我看完「APPLE II 微電腦徹底研究」第一冊之後，覺得與我主編的「音響技術月刊」頗有其臭相類之處，那就是作者躍然紙上，隨時與你共處。欲這樣，顯然是需要一些勇氣的，一些面對可能的疏失的勇氣，一些面對抱洋裝書的人的勇氣。我未嘗聽說有人照著食譜做菜，把菜燒焦了、燒爛了、糊了，會找作者理論一番的，但如果你寫技術的書、電腦的書，隨時你都得迎接焦了、爛了、糊了的挑戰。

總之，這不是一篇「序」，而是一點感想，我由衷地希望，每一位讀者都能珍惜作者施威銘先生在這本書中「掏」給你的東西。

序

有關電腦繪圖的書很多，電腦遊戲的書更多，但是你永遠會覺得，那些書談的與您所見的，都是完全不同的兩回事！繪圖的問題難嗎？遊戲設計的技巧艱深嗎？爲什麼，我們永遠探究不到問題的核心呢？

精彩有趣的繪圖功能，是個人用電腦的特徵之一，如果我們以傳統電腦的方式來使用，則非但不能發揮其長處，反而是壓抑了它的功能，這也難怪“××電腦繪圖”之類的教本永遠是畫不出小精靈與大嘴巴了。

畫不出小精靈當然不是頂嚴重的事情，可是如果您是生物教師而聽說Apple生動有趣的生物教學，如果您是化學教師而想圖示活化複雜的反應過程或是質量作用定律（mass reaction law）的……

在這一本書當中，我們將對Apple的電腦繪圖的關鍵技巧予以說明並與大家討論，其內容包含了靜態與動態的彩色繪圖法，而這些方法不獨是使用在遊戲程式裡頭，舉凡可以經電腦畫面顯示的各種應用，這些技法絕對是必須而有效的。早期如小蜜蜂之類的電腦遊戲稱爲Arcade Game（我們俗稱電動玩具），J. Stanton曾出版了“Apple Graphic & Arcade Game Design”一書，有關此類Game的基本技巧，本書也將其擇要歸納於第3章。但是像Adventure Game之類的最新動畫遊戲，其意義又遠超於Arcade Game了，所以本書在第6章當中將這種Game的技法做了相當詳盡的說明。並且，我們也發展出一套很巧妙的基本具具，使得動畫設計不再是一項煩雜無頭緒的工作。

在第0章，我們說明了動畫設計的原則，並且以畫面位址查表法來驗證所提出的理論，然後在第1章介紹您有趣的彩色技法。第2章是繪圖基本動作的詳細解說，尤其強調的是預移圖序的使用，此外還有各種繪圖一

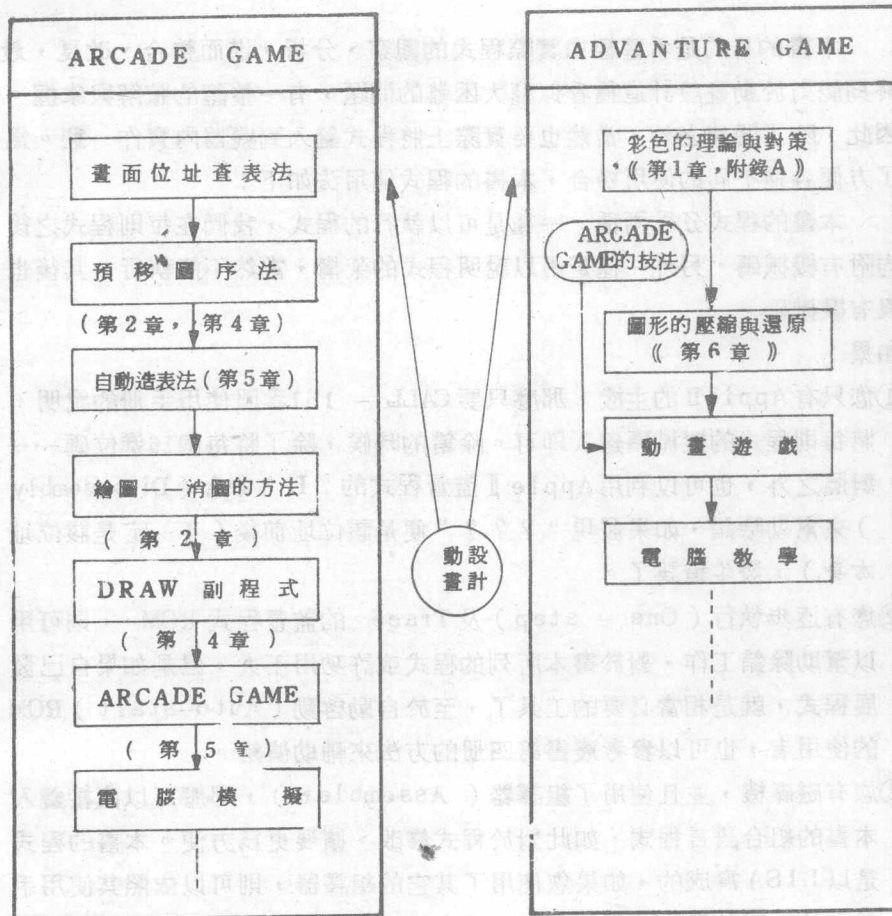
-消圖方式，以及碰撞檢知，第3章則擇要地說明了Arcade Game 的構造，並且驗證了各種繪圖——消圖方法，在第4章我們介紹一個很有用的繪圖副程式DRAW，並且熟悉了它的使用法。爲了令程式設計更爲電腦化，在第5章您將學到如何令Apple 自動的來繪圖、算圖序表、存圖序表……。當然，在第6章，您會學到Adventure Game 有趣的技巧了！

另外，要特別說明的是，爲免佔篇幅，除了少數需要摘要說明的程式外，一切程式說明均載於程式列表的附註上，而書中的主文，則多爲觀念及策略的探討。相信這樣的安排，將能提供讀者更大的視野、更明晰的概念！

總之，這本書，除了獻給所有Apple 的愛好者之外，更希望能對想用電腦繪圖來做其專業工作的教師、工程師、程式設計師……提供一點參考。對寫書的人來說，時間是永遠不夠使用的，如果有著述、校稿、編排上疏失的話，則是因爲我熱切想告訴你的心情吧！

作者 施成銘

本書結構



目 錄

如何使用本書

第 0 章 速度的意義

0-0	電腦遊戲——？	3
0-1	速度的意義	5
0-2	不要使用ROM的副程式	7
0-3	畫面位址查表法速度的極限	11
0-4	立竿見影的效果	16
0-5	所以，怎樣來設計動畫程式呢？	20

第 1 章 彩色繽紛的蘋果

1-0	APPLE II 只有六種顏色？——彩色的徹底研究	23
1-1	讓 APPLE 來替我們塗色	26
1-2	將背景移動	43
1-3	垂直與水平的背景移動	50

第 2 章 故事的主角——預移圖序繪圖法

2-0	畫故事的主角——預移圖序繪圖法	59
2-1	預移圖序法	62
2-2	彩色圖形的水平移動——彩色的預移圖序	65
2-3	畫了圖形之後——各種繪圖法的檢討	69
2-4	碰撞的問題	77

第3章 ARCADE GAME的構造

3-0	使用畫面位址表：GETADR 副程式	81
3-1	繪圖程式：XDRAW 副程式	83
3-2	讀鍵程式：READKEY	86
3-3	搖桿的問題：READPDL	88
3-4	嘗試寫一段動畫程式	98
3-5	火箭與背景的問題	104
3-6	彩色圖形的水平移動	110

第4章 走吧！大嘴巴！

4-0	預移圖序法的徹底研究	113
4-1	預移圖序的繪圖程序	118
4-2	程式4-0與3-8的比較	125
4-3	碰撞及兩種以上圖形的處理法	
	走吧！大嘴巴！`	127
4-4	更徹底的來解決問題——完全的查表法	144

第5章 動畫設計的策略與實現(一) (ARCADE GAME)

5-0	自動的造表程式：SHAPEDIT	151
5-1	造表程式及其使用法	157
5-2	動畫程式設計的策略	171

5 - 3	電玩的程式設計	176
-------	---------------	-----

第6章 動畫設計的策略與實現(二) (ADVANTURE GAME)

6 - 0	什麼是 Advanture Game ?	213
6 - 1	LUBAR 的構想——高解像畫面佔 8 K ? ...	214
6 - 2	HOLLE 的方式	217
6 - 3	畫面壓縮 ——還原使用例	223

附錄A APPLE II 高解像畫面結構

A - 0	顏色的規則	227
A - 1	高解像記憶位址與座標位址的關係	229
A - 2	Apple II 的畫面解析度是 560 點？(神秘的 b ₇ 位元)	231

附錄B 偵錯程式

B - 0	程式設計的原則	235
B - 1	偵錯程式	236
B - 2	LISA 的 TRACE 程式	237
B - 3	改良的偵錯程式	238

附錄C 電玩的聲音副程式

C - 0	視覺與聲音共存的秘密	241
-------	------------------	-----

C - 1	改進音響的效果	243
附錄D	LUBAR的PACK-UNPACK程式	245
附錄E	有趣的英文字母	251
附錄F	計分程式	259

LISA	Apple Tool Kit
ADR	DW
END	(不必用)
EQU (非零頁定址法)	EQU
EPZ (使用零頁定址法)	EQU
HEX	DFB
ICL	CHN
LST	LSTON
NLS	LSTOFF
PAG	PAGE

LISA 與 Apple Tool Kit 指令互換表 (僅列出對應不同者)

LISA	SWEET-16
LDR	LD
STO	ST
PPD	POPD
BRA	BR
BIC	BC
BIP	BP
BIM	BM
BIZ	BZ
BNM	BNM1
BKS	BK
RSB	RS
BSB	BS

LISA 與 WOZNIAK 的 SWEET-16 指令互換表 (僅列出不同者)

好奇心使您對這本書感到興趣，隨著本書的每一章，您會想過而不得其解的或是未曾想過新鮮的事，我們的第 0 章……

召您：

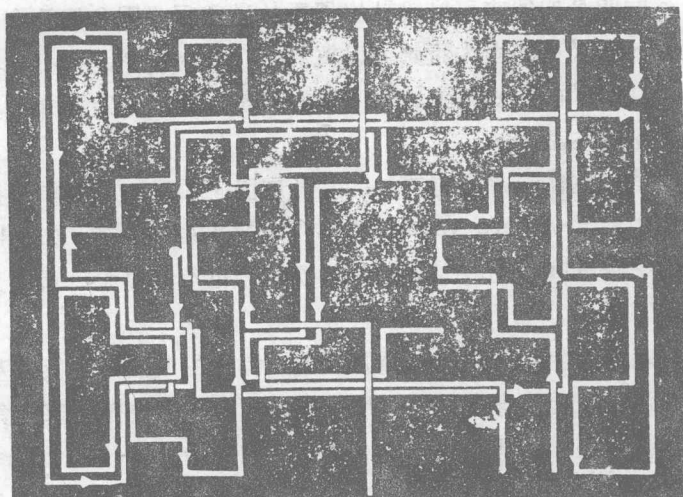
包括 Game) 設計的基本形態

的抉擇

特殊考慮因素

0

速度的意義



0-0 電腦遊戲 ?

如果您在舊書攤翻到一本十年前的電腦書籍，或許可以看到其中有一頁是“蒙那麗莎的微笑”這樣的畫，這種畫不是用圖點（Pixel）畫出來的，通常是用“** ”號或其它字母組合而成，如“*” 這發現到書中有所謂電腦遊戲的話，則不外乎心算、猜字或猜顏色之類的遊戲。

今天，如果有人要你鍵入一個用 5 H 代表紅桃 5，8 S 代黑桃 8 的 Game 在 Apple 上玩玩，你一定認為此君未免不經世面。在一個可以很生動的畫出各樣圖形的 Apple II 上使用 5 H，8 S，就像是當您學會彩色畫圖之後，再也不會用“RED 代表紅色，BLUE 說是藍色……”一樣的道理了。問題是如何才可以設計出優秀的遊戲程式呢？儘管今天，小精靈、小蜜蜂、金撲克的程式隨處可以買到，我們却仍舊迷惑不知其解（或者是已知其“解”而不知其設計的原理），可是為什麼一定要知道其原理呢？如果只是遊戲的話……

任何事情，往往不會只有單一的一個層面，如果一局圍棋不把它想成圍棋的話，如果拿 Apple Game 的技巧來做 CAD 的話，如果將遊戲程式設計者請來做電腦輔助教學的話，我們將再也不會無視於這種大雅之堂以外的……了。

在本書當中，我們將介紹各種設計遊戲程式的最新技巧及其基本原理，一方面是這些技巧本身就是值得學習的好題材，另外是希望能夠經由這樣的學習之後，可以藉此而引發創造力，大家來設計自己的 Game、自己的電腦程式、自己的電腦。至少，我們可以不必再玩 5 H 是紅桃 5 之類的遊戲了。

0-1 速度的意義

時間與空間——新的設計哲學

最初的一些微電腦程式，通常是以如何節省記憶空間為重要的考慮因素，這是因為在 1975 ~ 1979 年間，記憶體的价格並不是像今日 16 K RAM 值 50 元般的廉宜。因此，要在 KIM-1 或 4 K 的 Apple（系統已佔用了不少）來設計一個程式，“空間”絕對是一個考慮的重點。可是在幾乎每部微電腦都有數十 K RAM 的今天，要設計一個像“動畫”這樣以速度表現見長的程式，在“執行的時間”與“程式佔用的空間”當中，卻必須有一個抉擇，而往往是在必要的情况下，以空間來換取時間，這是今天 Game 設計上常見的一種策略。

速度為什麼重要

動畫程式的速度，與我們在螢幕上看到的圖形運動速度，並沒有直接的關係。譬如，一顆快速穿越螢幕的子彈，用 Applesoft 一樣也能達成。反之，一面微風吹動而緩緩飄揚的國旗，却必須要以絕對快速的機械語言才足以表現其悠揚有緻的自然韻味。因此，速度在現代動畫程式設計裡的意義，是在於如何發揚電腦快速的能力來完成畫面動作的每一細節，而不只是單純的圖形運動而已。

舉例來說，要讓畫面上的一個點做直線的移動，我們可以一次向旁邊移動一個圖點（pixel）距離，如此這個點是真正的經由路徑上的每一位置來移動的，另外，我們也可以間隔 3 個或 5 個圖點來移動，這樣的點是以跳躍的方式來行進的。對於相同的距離而言，前者勢必要比後者花更多的時間來完成全部的動作，下列的程式可以說明我們的看法：

```

LIST

```

```

10 HGR
20 J = 2: GOSUB 100
30 J = 15: GOSUB 100
40 GOTO 20
100 FOR I = 0 TO 277 STEP J
110 HCOLOR= 3
120 HPOINT I,100: HPOINT I + 2,100
125 HPOINT I,101: HPOINT I + 2,101
130 FOR R = 0 TO 7: NEXT R
140 HCOLOR= 0
150 HPOINT I,100: HPOINT I + 2,100
155 HPOINT I,101: HPOINT I + 2,101
160 NEXT I
200 RETURN

```

程式 0.0

這個程式在螢幕上畫出由 4 個白色點所組成的方塊，然後分別由第 20 行及第 30 行所設定的 J 值來控制此小方塊的水平移動。我們可以發現到，當畫點的間隔加大時（J = 15），則圖形的移動速度變快，但却有不連續的現象，而當畫點的間隔變小時，圖形的移動便很自然了。或許您會疑問到，雖然圖形移動自然了，但是速度不也變慢了嗎？是的，這也便是本書的主題所在：“如何以 速的畫圖速度（注意不是運動速度），來造出自然而生動的遊戲畫面”！