

HOPE

用户界面程序设计

■ 本书是成功的用户界面设计与实现的指南。详细讨论了交互式程序的输入、输出、屏幕布局、帮助与教学信息、以及错误处理，并包括各种用户风格的样例程序与屏幕，代码草图（包括窗口管理系统、菜单命令系统）；用户界面管理系统中的界面及其发展；彩色的原理及其在屏幕设计中的应用。



中国科学院希望高级电脑技术公司

用户界面程序设计

姒荷 冯庆文 唐晓飞 编译

中国科学院希望高级电脑技术公司

一九九一年四月

目 录

第一章 简介	(1)
1.1 用户界面系统一例	(1)
1.2 交互式系统设计	(3)
1.3 用户界面的基本原则	(4)
灵活性与一致性	(5)
1.4 一个用户界面工具箱	(6)
1.5 本书的组织	(7)
1.6 阅读材料	(7)
第二章 用户输入的原则	(9)
2.1 用户输入的类型	(9)
2.2 输入设备	(9)
模拟设备	(12)
2.3 控制输入	(14)
2.4 数据输入处理	(16)
2.5 向用户提供反馈信息	(16)
2.6 实例: 带三种用户输入方式的程序	(18)
2.7 阅读材料	(19)
第三章 菜单编程	(20)
3.1 菜单控制系统实例	(20)
AppleWorks	(20)
MacWrite	(21)
dBASE IV	(22)
Lotus 1-2-3	(22)
Auto CAD	(22)
3.2 正文菜单的编程设计	(23)
正文菜单原则	(24)
替换形式	(25)
正文菜单的实现	(27)
一种更一般的菜单方式	(31)
3.3 图形菜单的编程设计	(33)
图形菜单的原则	(33)
图形菜单实现	(35)
第四章 基于命令的系统的编程	(38)
4.1 命令控制系统的例子	(38)
Unix	(38)
dBASE IV	(41)
Tell-a-Graf	(42)
4.2 命令项	(43)

4.3 命令控制系统的总体结构	(45)
命令动作设计的工具	(45)
Undo 命令	(46)
4.4 允许用户定义命令	(46)
4.5 命令输入的编程	(48)
将命令分成单词表	(48)
将单词与命令部分匹配	(50)
4.6 命令动作的编程	(53)
4.7 阅读材料	(55)
第五章 数据输入	(56)
5.1 数据输入的不同格式	(56)
5.2 从程序中控制数据输入	(56)
5.3 正文输入技术	(57)
5.4 提供默认值	(59)
5.5 数值输入技术	(59)
读入一个整型值	(60)
读入一个实型值	(61)
从表达式中获取一个数值	(64)
5.6 “填空”输入管理	(70)
5.7 阅读材料	(74)
第六章 信息反馈	(75)
6.1 一般原则	(75)
计划	(75)
一致性	(75)
可读性	(75)
满足用户需求	(76)
给出信息	(77)
6.2 组织输出的方法	(79)
屏幕的主要成分	(80)
标题行	(80)
命令或状态行	(80)
错误或警告信息	(81)
屏幕体	(81)
不同类型屏幕的考虑	(81)
菜单	(82)
数据输入屏幕	(83)
问答屏幕	(83)
查询屏幕	(84)
信息屏幕	(84)
控制屏幕	(85)
强调关键部分	(86)
给出表格总结	(87)
给出图形总结	(87)
6.3 输出设备	(88)

第七章 屏幕技术	(90)
7.1 交互式屏幕	(90)
数据查询	(90)
动态介质屏幕	(92)
7.2 正确地显示信息	(95)
7.3 使用窗口技术	(100)
窗口的一般用途	(100)
活跃程序操作的窗口	(100)
窗口用于瞬时信息或控制	(101)
窗口布置	(102)
窗口控制	(102)
7.4 阅读材料	(103)
第八章 在输出中使用颜色	(104)
8.1 利用颜色来增加意义	(104)
在应用程序中使用颜色	(105)
让用户自己选择颜色	(105)
8.2 作为一门“精确科学”的颜色	(107)
颜色感知的物理学	(107)
8.3 颜色的相互作用——不再是“精确科学”	(107)
8.4 颜色说明	(108)
混合颜色	(108)
颜色模型	(109)
光谱模型	(109)
灰度级	(109)
Munsell 模型	(109)
RGB 说明	(110)
CMY 说明	(110)
HSV 说明	(110)
HLS 说明	(110)
HVC 说明	(111)
CNS 说明	(111)
实验与研究	(111)
复制颜色	(112)
正文使用多种颜色	(112)
8.5 有效使用颜色的原则	(112)
颜色组合的要素	(112)
疲劳	(114)
8.6 颜色查找表	(114)
使屏幕产生颜色	(116)
与颜色进行交互	(116)
建立查找表	(116)
选择要被设置的颜色	(116)
设置颜色	(117)
ICARE: 交互式计算机辅助 RGB 编辑器	(118)
8.7 颜色之意义	(119)
8.8 与单显的兼容性	(120)

第九章 直接操纵系统和特殊环境	(121)
9.1 直接操纵的概念	(121)
9.2 直接操纵的程序设计	(124)
9.3 数据输入的直接操纵技术	(126)
9.4 高级实验环境	(127)
ARK—Alternate Reality Kit (微型世界包)	(128)
Interactive Image (交互式图形)	(129)
Voice Recognition (语音识别)	(130)
NOOBIE:动物设计游戏站	(133)
VIDEOPLACE	(134)
Three_Dimentional Virtual Environment (三维虚环境)	(134)
9.5 阅读材料	(136)
第十章 窗口和用户界面管理系统	(138)
10.1 窗口系统的概念	(138)
10.2 窗口和窗口系统	(142)
字符—映象窗口 (Character-Mapped Windows)	(142)
Microsoft Windows	(147)
Macintosh ToolBox	(152)
SunNeWs	(153)
X 窗口系统	(161)
10.3 用户界面管理系统 (UIMS)	(167)
MacApp	(168)
Open Look	(176)
TIGER	(178)
iCpak 201	(179)
Actor	(180)
Open Dialogue	(184)
NewWave	(187)
10.4 阅读材料	(189)
第十一章 帮助用户与程序沟通	(191)
11.1 提供学习帮助	(191)
11.2 提供用户引导帮助	(193)
11.3 提示帮助的原则	(194)
11.4 减少和控制程序中的错误	(195)
11.5 阅读材料	(196)
第十二章 提示系统编程	(197)
12.1 提示系统例子	(197)
Appleworks	(197)
Lotus 1-2-3	(197)
dBASE IV	(199)
Prime 的 PRIMOS	(201)
Unix man	(202)
Zenographics Mirage	(202)

12.2	使提示系统正常运转	(203)
12.3	提示系统选择项	(203)
12.4	易犯的错误	(204)
12.5	例子	(204)
12.6	阅读材料	(206)
第十三章 错误的处理和避免		(207)
13.1	输入错误	(207)
13.2	算术错误	(208)
13.3	文件和设备错误	(211)
13.4	错误消息	(214)
13.5	失败的恢复	(216)
13.6	阅读资料	(216)
第十四章 特殊考虑:使程序便于残疾人使用		(217)
14.1	为手有残疾的用户考虑	(217)
	特殊输入设备	(218)
	菜单考虑	(220)
	管理命令	(221)
	获得对话输入的其它方法	(221)
14.2	为视力有障碍的用户考虑	(221)
14.3	色盲用户和老年用户	(224)
14.4	为有听力缺陷和阅读障碍的用户考虑	(224)
14.5	阅读资料	(225)

第一章 简介

如今，很多人都在使用计算机。每个从事写作、印刷、记录等信息工作的人都可能在办公桌上放台计算机，并经常使用它。拥有一台计算机便意味着一个技术天才的日子已一去不返，每个人都是一个计算机用户，或潜在的用户。一般人对自己想从计算机那儿得到些什么已有了相当成熟的看法。

在有个人计算机之前，程序主要是为商业化而编制的，必须经过特别训练并拥有一定技巧以后才能使用，训练比使软件适应于用户更容易些。随着个人计算机的推广，用户界面的质量已成为一大问题。从 70 年代中期开始，个人计算机的软件市场年销售额已上升到数十亿美元，其中包括用于开发界面的花费。

现在的计算机用户对于用户界面的质量要求很高，在评价软件时也对之寄以期望。欲取得一种认可，一个程序必须能帮助其用户完成任务。一个程序想要在市场上成功，必须能使用户最大限度地发挥其技能并将成就归功于用户。良好的用户界面对一个程序的被认可与成功起着很大的作用。

在按用户的思维编制程序时，需要记住的最重要的事情是没有人愿意为使用程序而使用程序。人们或许希望用计算机来完成某个任务，但该任务决非计算机与程序，编程者的工作是使得计算机与程序尽可能地无形，以便于用户工作。

本书致力于帮助程序员实现良好的用户界面。虽然我们谈到了如何进行良好的设计，但并不讨论实际界面的设计，而主要讲如何利用用户界面技术来完善一个良好的设计。

1.1 用户界面系统一例

在以后的章节中关于用户界面的讨论包括一些软件产品的例子。现在我们来看一个系统，了解它的各种操作模式，并讨论它们各自的含义。

该系统是 Lotus 发展公司的 Lotus 1-2-3 (以下称为 1-2-3)，它可以在 IBM PC 和其它 MS-DOS 计算机上运行。讨论仅限于一些关于不同的功能的例子，并说明这些功能如何向用户提供并交互地反应。

Lotus 1-2-3 是一种金融模型和预算程序，是一个著名的电子表格软件。它提供二维数组的表元，其中每一个表元中可以存放正文(标签)、数字(值)、或表达式，包括数学函数与运算符、常量、以及其它表元值(公式)。当一个公式写入一个表元时，该公式的值立刻被计算并在表元中显示出来。无论何时改变了一个表元的值，其它用到该值的表元也相应改变。图 1.1 显示了在 1-2-3 屏幕上填充一些表元时进行的一些简单计算。

注意该屏幕是如何分成几个区的：命令菜单区、工作表元，当前表元地址和表元输入区。在第五、六、七章将讨论特定屏幕区域的功能并说明如何实现这些功能。

在 1-2-3 中的基本操作围绕着电子表格和表元的输入值。这需使光标控制键能在表元上下左右移动。在显示屏上，用反视或彩色来突出显示被选中的表元。这使得用户对电子表格可以直接地控制存取。当一个值、标签或一个公式被输入时，在表元的输入区中它

被重现。输入结束后表元的数值、正文或公式值立即在选中的表元中写出，其他基于该表元值的表元重新计算它们的值。这几乎是同时发生的，用户的操作获得极快响应，并且在任何时刻控制着电子表格，本书第五章描述如何控制这种输入和显示。

B7:

	Worksheet	Range	Copy	Move	File	Print	Graph	Data	System	Quit	MENU
	Global,	Insert,	Delete,	Column,	Erase,	Titles,	Window,	Status,	Page		
	A	B	C	D	E	F	G	H			
1											
2	First		125								
3	Second		8								
4											
5	Sum		133								
6											
7	Product		1000								
8											
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

图 1.1 一个普通的 1-2-3 屏幕

A1:

Worksheet **Range** Copy Move File Print Graph Data System Quit MENU

Global, Insert, Delete, Column, Erase, Titles, Window, Status, Page

A1:

Format Label Erase Name Justify Protect Unprotect Input Value Transpose MENU

Format a cell or range of cells

A1:

Fixed Scientific **Currency** General +/- Percent Date Text Hidden Reset MENU

Currency format (\$x,xxx.xx)

H1:

Enter range to format: A1..H1 MENU

图 1.2 在 1-2-3 中格式化一行

除向表元输入之外，在电子表格上还可以作许多其它操作。可以增删行列，可以使电子表格的某一部分的内容格式化（标签可以居中、左移或右移，数值可以按指定的十进制

方式或带\$符号写出等等)。在电子表格中值或公式可以被拷贝,电子表格可以被保留或装入。这些全在屏顶菜单区的管理之下。实际上,1-2-3有一个很好的多层菜单,它允许初学者运用光标键选择操作(使用识别存储器)或训练用户 type-ahead 高速打印命令字母(用回忆存储器)。在第三章描述如何写这一菜单系统。

欲看菜单系统是如何工作的,图 1.2 部分显示了在 1-2-3 菜单中如何使用图中的一行格式化以带有一个\$符号和二个十进制位的三个步骤。打印/REC 命令,然后指定一行可以得到同样的结果。

Lotus 1-2-3 中有一个扩展的帮助系统。在程序中的任何一点上,用户可以按 F1 请求帮助,同时原工作表元中的内容被描述当前菜单选择的结果的内容替代。尽管语言常常是专门电子表格化的,帮助信息仍可作为程序的一个小小的辅导。关于帮助职能的详细描述与实例详见第十二章。

在 1-2-3 中对出错处理得很好。正文错误由操作系统处理,倘若是非扩展性正文出错程序不会停止执行;仅向用户传递系统错误信息。计算错误被很好地分开了。若进入了一个不合法的功能或一个不能完成的计算,例如,使用了一个已包括符号的工作表元或者除数为 0,该表元中包含的计算被赋以一个错误值,并且在该表元中显示“ERR”,其它使用该值的所有表元均得到这一错误值并且显示“ERR”。程序不会中断并且错误释放表元不受影响。当该表元被更正时(所有表元可被编辑),其值被显示,其它相应表元被重新计算,在第十三章中讨论出错处理。

1.2 交互式系统设计

本书并非关于交互式系统设计的,而是讨论如何实施交互式程序的许多标准特征。本书是为程序员写的,目的在于提高其在交互技术方面的造诣。界面设计还是让给用户专家来考虑吧,因为同人打交道,理解他们的想法并帮助他们达到目的需要专门的技术。程序员不具备用户界面设计的良好思路,因为他们所受的教育是关于计算机而非人类的。

或许理想的软件设计是由一班人马实现的。应用专家与有认识力的用户专家做理解用户的职责,以及用户如何考虑完成其职责的工作。然后将这些内容转化成交互作用与屏幕设计等特征:用户该做什么?机器如何响应?要求程序员实现这一工作。偶尔也会有人具备上述知识或凭直觉知道用户如何工作并且有程序设计技巧来实现它,但这种情况很少见。

一个软件的全面的设计包含许多组成部分。其任务必须被定义并落实到输入输出级上,这通常与用户界面分开实现。用户界面的设计必须提供从用户到任务的输入并将输出传给用户;这可能也是单独做的。自然,任务与界面均需完善并且相容,良好的用户界面掩饰不了任务质量的低劣,差的用户界面却可能使一个好的产品不被接受。

软件的界面设计通常经过一个标准的途径。第一步是了解用户如何考虑任务的,他们希望工作怎样做。同应用专家协作后,这些可能同用户的观点或者一些信息的收集或者实际技术有关。这一阶段的目的是建立一个关于用户如何在任务上工作的实体模式并且对如何在计算机上实现任务有一些影响。该模式用以支持软件执行任务的方式,从本质上决定如何建立用户界面。

第二阶段将建立用户界面的原型(prototype), 并让用户在这些原型上工作以求找到用户反映最好的一个原型。在这一阶段最好请一个图形设计者以保证用户与软件之间实行了正确的图形通讯。每一个原型均支持用户/任务模式, 但它们各不相同, 这些原型的测试是在不同环境由不同用户进行的, 从中选择最成功的一个加以完善, 生成最终界面。

在改进用户界面的各个部分以及用户通讯的支持时, 化费了大量时间讨论如何使用户界面有用并实现这一技术。若想更深了解设计方面的知识, 请阅读用户界面设计的文献; 在本章末列出了这些文献的名字。我们的目的在于描述一个良好的用户界面并指出用户界面设计的原则是使程序员(或许是您)与设计者(要求你让计算机做不可能的工作的人)更易于共同工作。苹果计算机公司(Apple Computer)的 Bruce Tognazzini 说的话曾被引证: 一个界面设计者化了 87%的时间以应付程序员。希望能降低这一百分比。

如果一个程序员没有界面设计者的支持, 请记住自己对用户界面的直感不一定最优, 阅读本书中的原理并尽量与自己的想法融合。多写几个草稿来试验自己的想法。在软件应用领域交一些益友—那些能够提出一些建设性的批评而又不必担心会耿耿于怀的人—给你提出一些中肯的意见与新的方法。最后从未来用户和有类似经验的人身上(他们可提供与他们自己当前应用的软件的比较)获取教益, 尽量从同行的看法中获益。这些努力的结果是值得的。

1.3 用户界面的基本原则

一些原则似乎与所有良好的用户界面是一致的, 在此将它们作为一般原则阐述。当涉及输入、输出以及用户界面的其它方面时将个别说明。

用户界面涉及所有输入输出。它必须向用户传达关于程序员使用的所有指令, 让用户尽可能自然地控制程序, 并向用户提供程序的结果以便于使用。若界面提供了不明确的指令, 或者用户不能够通过输入控制任务, 好象计算机不曾参与一样, 或者程序的输出含糊不清, 那么这种通讯、这一界面、最终这个软件是不成功的。

原则: 用户界面的主要功能是通讯

程序必须执行给出的指令的功能, 允许控制、显示结果而不干扰任务进程。这需要动点脑筋进行计划, 它尤其意味着所有的输入应该是自由格式的或自然选择的, 输出必须清楚, 没有错误的或含糊不清的显示、或语法错误。运用一切合适的工具使显示屏幕快速可读。这些事情并非微不足道。它使得程序令用户满意, 并防止计算机对任务的干扰。

原则: 界面必须防止计算机梗于用户与工作之间。

界面必须支持问题的单一模式及其解决方法, 允许用户在计算机上仅以一种单一的方法考虑任务, 而无须疑惑: “计算机希望我以哪种方式干这事?”这使得任务成为习惯性的, 并且增强计算机的不可见性。

原则：界面至始至终必须与程序相容

程序的用户可能从基于信号的到基于单词的变化不一。有的人喜欢用数字，有的人爱用字母，有的人希望不用记任何东西就可以使用程序而有的人愿意成为程序的专家，有的人乐意用菜单而有人坚持用命令。欲获得全面的成功，必须针对不同的人提供不同的最适合其要求的工作方式。

原则：界面必须是灵活的以适应广泛的用户

或许一个紧急电话通知了一个急事，用户必须放下手边的任务几小时后才能继续工作。一个好的助手将会帮助他们记忆发生了什么事；程序也能保持一些当前发生的信息做到这一点。

原则：界面必须使用户知道任务的当前进行情况

认为只有专家才会使用程序是没有道理的，即使专家也不可能随时记住每一件事。用户必须能获得一些援助，这些援助是针对其当前活动的也是最优的。这是一个具有挑战性的通讯任务，但通过努力思考与计划也可以完成。

原则：界面必须包括获最帮助

所有用户都讨厌在程序崩溃时丢失或中断自己的工作。这或许不是界面的问题而是系统的问题，但由于它造成崩溃，通常使用户工作损失，故在此提及。1-2-3 可以防止许多用户引起的崩溃，这是一个值得努力的重要目标。

原则：程序不可崩溃

程序员的主旨很简捷。

原则：程序必须努力工作，而用户无须如此

上述每一原则均需要程序员可观的努力以避免程序不恰当的行为。自由格式化的输入、仔细地组织过的显示输出、可读的屏幕、状态及帮助信息、以及证实中止等花费了大量的工作。而这些工作在用户响应、接受、以及方便地进行工作等方面得到了报偿。

灵活性与一致性

灵活性与一致性的原则显然是相互对立的，需要仔细推敲。一致性包括两个方面一应用用户的思维模式的一致，和用户控制应用的方式的一致。

如本章前述，用户思维方式的一致是应用专家的工作。只要用户掌握了应用及其控

制，就有安排控制的灵活性。所以只要控制的性质与控制本原的结果是一样的，程序可以由命令或菜单控制，思维模式是受保护的。甚至允许用户选择其喜爱的界面风格。

有时，应用模式需要更广泛意义的一致性：与同一应用的其它程序的一致。除非一个新的空白领域的程序，否则其设计者必须密切注意在类似应用中最成功的程序里使用的技术，必须尊重用户在使用这些程序时的习惯。否则只有用非常有力的理由去说明用户在使用一个新程序时改变其习惯。商业市场经验表明：每一电子表格程序的基本操作类似于 VisiCalc，大多数绘图程序与 Macpaint 基本类似，这些相似并非偶然；用户与顾客决定了在选择新软件时易于学习是最重要的。

原则：使自己的程序工作时类似于其相似者

在程序的交互式控制中一致性同样重要，在整个程序中这些控制必须使用一致的机制。例如，在一个全屏幕编辑系统中，如果用户删除了当前光标所在字符，他并不想删去光标前的另一字符。若程序使用了命令，该命令需使用一致的句法及结构。如果程序使用了菜单，每一个菜单的选择必须相同并对用户的选择作出同样的反应，这点可以如第三章菜单例中通过在所有时刻的菜单选择里允许几个选择项以做到，或者使程序从几种可能的技巧选择中仅选择一种。程序能生成一种结构文件并允许用户为程序设置所需的选择。无论如何处理，使用户用程序时方便而且有效，这种内在的一致性很重要。

原则：使程序的所有部分相似地工作

控制一致性的想法由一个简单程序实现。在许多不同的售货机和应用中，键对程序操作控制的一致性使 Macintosh 成功的因素之一。第十章中讨论的一些系统如 Microsoft Windows 和许多用户界面管理系统均对应用提供一种普通的相似性。

1.4 一个用户界面工具箱

如上所述，用户界面程序设计工作量很大，在最终确定界面设计以前，许多程序需要大量的界面经验。这使得将每一个界面分别编码是不实际或不可能的。界面程序设计者需要一个界面工具箱。

一个工具箱中包括许多普通的工具，完成界面的大多数任务，包括：

- 菜单显示
- 分析(parsing)命令
- 自由格式数值输入的读入
- 用编辑与卷滚处理正文输入
- 在格式化屏幕上显示信息
- 显示警告与对话框
- 使用窗口
- 帮助显示

• 出错处理

这都是要求一个工具箱去做的, 省略部分意味着编码重复或对设计技巧的省略。

本书展现了工具箱的一部分, 关于专门界面设计技巧的每一章均包括设计实例及其实现代码。这些实例很通用, 尽管读者可以在自己的工具箱中采用, 实例重点在于清楚易懂而非效率。一些公司, 通常是一些小公司, 售卖用户界面工具箱。这些常常是菜单或窗口控制系统。一些商人, 出售完整的用户界面管理系统(UIMS)。第十章中对此讨论较多。

1.5 本书的组织

总之, 本书的目的在于向程序员提供一些实现广泛使用的用户界面技术的工具。它分成三个部分: 输入、输出和帮助/错误处理。此外, 有一个单独的章节针对一些低能人处理特殊的用户界面问题, 每一部分均较详细地讨论了其主题的一些方面。

输入部分包括菜单、命令和数据输入, 菜单(第三章)以实例描述了正文图形菜单, 并描述了在用户界面中直接控制(direct manipulation)的地位与功能。命令(第四章)包含用于程序控制的正文输入, 并介绍了带参数的命令和用户自定义命令。数据输入(第五章)包含可编辑、单词可卷滚的正文输入、自由格式的数值输入和填空技巧。

后面两章讨论包括输入和输出的界面程序设计。特殊直接控制系统与环境(第九章)探讨直接控制和面向对象的程序设计, 并提供了使用直接控制技术激活软件的一些例子。用户界面管理系统与窗口(第十章)讨论窗口显示的多种系统与技术, 以及一些可用的窗口系统与用户界面管理系统, 这两章指出了用户界面的未来方向。

第十一到十三章讨论用户界面中时常忽略的一些问题。帮助/错误处理部分包含求助和出错处理的技术。帮助(第十二章)描述了如何实现帮助系统并使之工作, 而出错处理(第十三章)描述了控制文件错误和计算错误的技术; 对设备管理提出了一些建议; 并对程序或整个计算机崩溃的恢复技术提供了一些建议。

最后, 针对残疾用户的一章(第十四章)重点在于为这些用户设置的特殊界面, 它帮助界面实现者注意到一些问题并且使更多的软件适用这些用户。

1.6 阅读材料

用户界面是设计与通讯的一种特殊情况, 在此特别推荐关于设计的两本书和一本通讯方面的书, Dreyfuss 的《Design for people》是一本设计过程方面的优秀著作, 作者是二十世纪一位杰出的工业设计家, 他在书中提出了实验过程和设计方面的一些有用的观点。Norman 的《The Psychology of Everyday Things》提出了应用设计面向用户讨论的观点。Tufte 的《The Visual Display of Quantitative Data》, 是有关视觉(visual)通讯的, 并将帮助读者对什么是好的通讯产生一个看法。

在关于软件的用户界面设计的众多书籍中, Heckel 的《Elements of Friendly Software Design》较为突出。该书著成较早但预言了 Macintosh, 该书浅显易懂, 恰当地指出了一些设计问题。Heckel 为程序设计者指出了为什么程序设计的思维过程妨碍了用户界面的设计。建议读者仔细推敲此书。

用户界面设计的另一本通用著作是 Rubenstein 和 Hersh 合著的《The Human Factor》。此书再次提供了一些有帮助性的准则，但具体措施较少，是一本好的参考书。Shneiderman 的《Designing the User Interface》更近似于研究性著作，作者是该领域的带头人之一。此书对界面设计的一些问题提出了详细的看法，常被用作界面设计方面的教科书。Brown 的《Human Computer Interface Design Guidelines》为用户界面设计提供了一些扩充的准则，这些基于他在静电印刷方面的经验，并且该书的包装很好。用户界面设计方面其它的一般性著作有 Baeck 与 Buxton、Beech、Card、Moran，以及 Newell、Carroll、Couts 和 Vlaeminke、Dumas、Monk、Nickerson、Norman 与 Daper、Sime 与 Coombs、Thomas 与 Schnierder 等人写的，以及 Shneiderman 的《Software Psychology》最后，用户界面设计问题与研究的最新信息是在 Human Factors Society and the Special Interest Group on Computer-Human Interaction of the Association for Computing Machinery (ACM-SIGCHI) 年会上提出的。在此提出了关于用户交互式反应的研究论文与有深度的趋势，在 ACG-SIGCHI 的姐妹学会上一些相似的方向被提出，尤其是与窗口系统以及界面实现有关的趋势。

第二章 用户输入的原则

在用户开始使用程序时，他们得输入一些数据或控制信息。控制用户输入的方式形成一整个程序的喉舌与触觉。不恰当的输入设计可能比其他错误或不正确的功能更惹人注意，这是软件评价中最常提到的一个问题。

有时需要一些特殊的输入技巧或设备使程序适用于一些残疾人，这将在第十四章中讨论。任何为一般用户写软件的读者均应意识到这一问题，请阅读第十四章，十四章中还提出了一些建议。

在第九章中讨论了其他的特殊环境，这是指一些特殊的软、硬件设施，如博物馆的、为孩子们、为教育的、为着人工智能的、以及为残疾用户的。然而，一旦为特定目的产生某种技术，它被看成适用于更广泛的范围。

2.1 用户输入的类型

程序要求两种输入。控制型输入指用户控制程序活动的信息。用户选择在何时以何种参数执行哪一项程序功能。在某特定功能中，数据输入提供给程序以其工作所用的数字或正文：机械地看来，命令参数与数据似乎相象，它们的输入甚至由同一工具控制，如以后的章节中所述，但这些功能在用户看来很不一样。

在这两种输入模式中，数据输入较简单易设计，当概念熟悉之后：仅仅是用普遍打入技术输入值（尽管其他的数据输入也是可能的，偶尔也是自然的。），计算机用户很快就习惯了输入正文或数字的标准技术。然而，控制型输入与用户思考问题、设计程序解决问题的方式密切相关。或许需要化费大量设计与实验法发现用户如何思考处理问题的，这是用户界面设计的一个重要方面。本章末的一些文章讨论了这些设计以及实验的方式。

2.2 输入设备

交互式程序传统的输入设备是键盘。这有许多不同的格式，从简单到复杂。图 2.1 显示了几种不同的键盘设计。即使是键盘设计这种基本问题也是可变化的；Dvorak 键盘常被看作普通 QWZRTY 键盘的替换品，如图 2.2，它的键盘安排对打字者而言更自然，在熟悉以后有助于提高打字速度。当要求输入命令时，必须考虑用户使用的键盘范围（数据输入比较标准）。功能键与光标键很有吸引力，但并非在所有键盘上都具备或相同。

其他一些设备常见于图形或混合的图形 / 正文环境。根据计算机图形采用的方式，将这些设备按功能分类。

取(pick)设备的功能是在显示屏上选择一个目标。该目标通常是图形菜单的一部分或一个将被修改的目标。当选取设备的屏幕响应在目标区域上时，一个选取操作在图形屏幕上选择一个目标。这通常伴随着亮显一个目标。这种选择通常由诸如按一个钮或键的事件操作来确证。

定位器(locator)设备用以识别屏幕上的一点。当设备被操纵时,选中点被更新并在屏幕上显示出来,通常是一个小十字架或其他称为定位器的光标符号。当程序需要一点时,要请求一个定位器的定位仪和屏幕分辨率一样精确,因此一个返回点只能看作一个近似的近似。

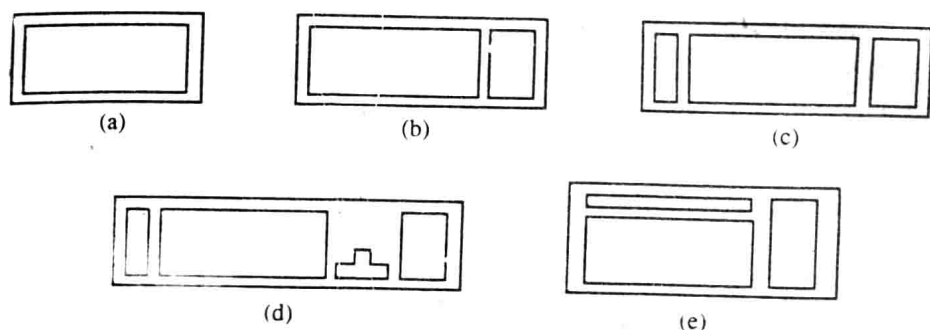
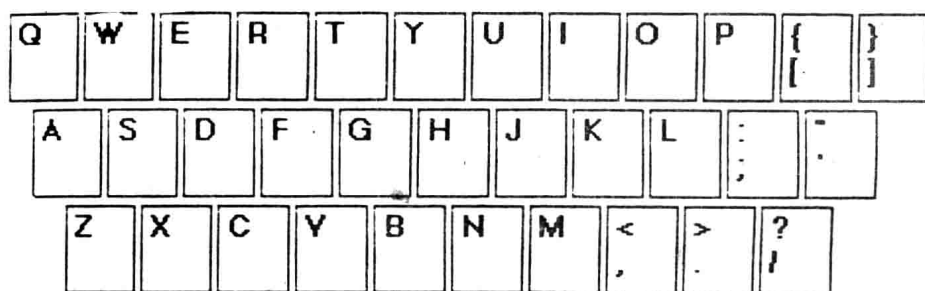
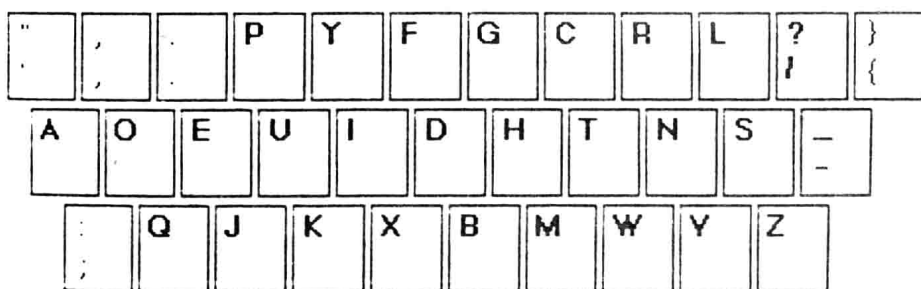


图 2.1 键盘安排的范围

- (a) 既非功能键亦非数字键盘 (Macintosh plus)
- (b) 无功能键 (Macintosh SE 标准键盘)
- (c) 功能键与数字键盘 (IBM PC)
- (d) 左边是功能键, 光标键盘
- (e) 顶部是功能键



QWERTY keyboard



Dvorak keyboard

图 2.2 QWZRTY 与 Dvorak 键盘安排

录入设备用以向程序中输入一系列坐标对。这种部件通常用于图形应用如数字化一个图像, 而非界面设计。