

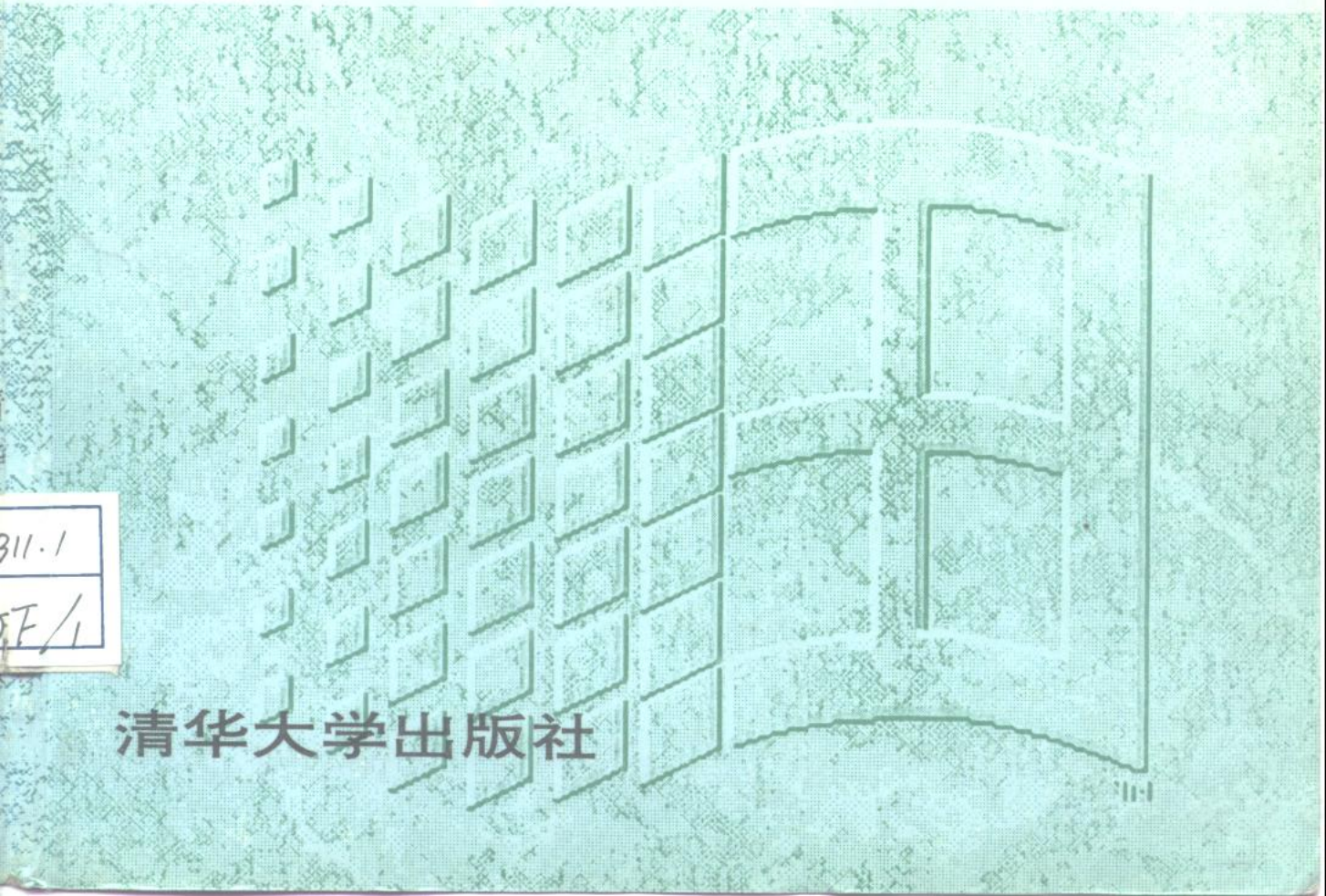
Microsoft Microsoft Microsoft

WINDOWS 界面 应用程序设计指南

沈金发 李莉 译 王旭 校

311.1
SF/1

清华大学出版社



TP311.1
SJF/1

WINDOWS 界面

应用程序设计指南

沈金发 李 莉 译
王 旭 校



1024810

清华大学出版社

THE WINDOWS INTERFACE

An Application Design Guide

本书英文版由 Microsoft 公司属下的 Microsoft 出版社(Microsoft Press)出版。版权为 Microsoft 公司所有。(©Microsoft Corporation 1992)。

本书中文版版权由 Microsoft Press 授予清华大学出版社独家出版、发行,1993。未经出版者书面允许,不得用任何手段复制本书部分或全部内容。

Adobe 和 PostScript 是 Adobe 系统的注册商标。

Apple 和 True Type 是 Apple 计算机的注册商标。

Helvetica、Linotype、Times 和 Times Roman 是 Linotype AG 或它下属的注册商标。

Microsoft、MS 和 MS-DOS 是 Microsoft 公司的注册商标。

Windows 是 Microsoft 公司的商标。

Arial 和 Times New Roman 是 The Monotype 公司的注册商标。

JS40/11

(京)新登字 158 号

WINDOWS 界面

应用程序设计指南

沈金发 李莉 译

王旭 校

☆

清华大学出版社出版

北京 清华园

清华大学印刷厂印刷

新华书店总店科技发行所发行

☆

开本:787×1092 1/16 印张:8.75 字数:213 千字

1993 年 2 月第 1 版 1993 年 2 月第 1 次印刷

印数:0001—5000

ISBN 7-302-01217-2/TP·458

定价:29.00 元

目 录

前言	(1)
介绍	(1)
目的	(1)
目标	(1)
实现	(1)
输入方法支持	(2)
推荐级别	(2)
键及键组合表示	(2)
第 1 章 原则与方法	(3)
1.1 用户界面设计的原则	(3)
1.1.1 用户控制	(3)
1.1.2 直接性	(3)
1.1.3 一致性	(3)
1.1.4 可辨性	(4)
1.1.5 美观性	(4)
1.1.6 反馈性	(4)
1.1.7 宽容性	(4)
1.1.8 认识人类的能力与局限	(4)
1.2 设计方法	(4)
1.3 部分参考文献	(5)
第 2 章 主要输入构件	(6)
2.1 鼠标输入	(6)
2.1.1 基本操作	(6)
2.1.2 鼠标操作使用指南	(7)
2.2 键盘输入	(7)
2.2.1 字符键	(7)
2.2.2 编辑键	(7)
2.2.3 方式键	(8)
2.2.4 导向键	(9)
2.2.5 加速键	(9)
2.2.6 退出键	(11)
第 3 章 一般技术	(12)
3.1 选择	(12)
3.1.1 选择的概念	(12)
3.1.2 鼠标选择	(14)
3.1.3 键盘选择	(16)
3.2 输入焦点	(17)
3.3 导向	(18)
3.3.1 鼠标导向	(18)
3.3.2 键盘导向	(18)
3.3.3 键盘调用控制杠	(19)
3.4 传递界面	(20)
3.5 直接操作	(20)
3.5.1 区别选择和直接操作	(22)
3.6 反馈	(22)

3.6.1	视觉反馈	(22)
3.6.2	声音反馈	(26)
3.7	编辑文本	(26)
3.8	移动对象	(27)
3.9	文本框	(27)
第4章	窗口	(28)
4.1	屏幕窗口类型	(28)
4.1.1	应用程序窗口	(28)
4.1.2	文本窗口(MDI)	(29)
4.1.3	用MDI应用程序打开文件	(30)
4.1.4	漂浮窗口	(31)
4.2	屏幕窗口组成	(32)
4.2.1	窗口边框	(32)
4.2.2	标题栏	(33)
4.2.3	菜单栏	(33)
4.2.4	滚动杠	(34)
4.2.5	分裂框和分裂杠	(34)
4.2.6	消息栏	(35)
4.2.7	状态杠	(36)
4.2.8	控制栏:带、尺、工具箱和调色板	(37)
4.3	窗口操作	(38)
4.3.1	移动窗口	(38)
4.3.2	重置窗口大小	(39)
4.3.3	关闭窗口	(39)
4.3.4	分裂窗口成窗口区	(39)
4.3.5	切换窗口和窗口区	(39)
4.3.6	窗口、窗口区中滚动数据	(41)
第5章	菜单	(43)
5.1	菜单类型	(43)
5.1.1	坠落式菜单	(43)
5.1.2	弹出式菜单	(43)
5.1.3	级联菜单	(45)
5.2	菜单组成	(46)
5.2.1	菜单标题	(46)
5.2.2	菜单项	(47)
5.3	菜单操作	(50)
5.3.1	鼠标方式	(50)
5.3.2	键盘方式	(51)
5.4	标准菜单	(52)
5.4.1	控制(Control)菜单	(52)
5.4.2	文件菜单	(54)
5.4.3	编辑菜单	(56)
5.4.4	帮助菜单	(57)
5.5	通用选择菜单	(59)
5.5.1	视口(View)菜单	(59)
5.5.2	窗口(Window)菜单	(59)
第6章	对话框控制	(61)
6.1	按钮	(61)
6.1.1	命令按钮	(61)
6.1.2	可选按钮	(62)
6.2	检取框	(63)

6.3	列表框	(64)
6.3.1	单项列表框	(64)
6.3.2	扩展和多项列表框	(67)
6.4	文本框	(68)
6.4.1	组合框	(69)
6.4.2	自旋框	(70)
6.5	只读弹出式文本域	(71)
6.6	滑动块	(72)
6.7	静态文本域	(72)
6.8	成组框	(72)
6.9	控制标签	(72)
6.10	输入确认	(74)
第7章	使用对话框	(75)
7.1	对话框类型	(75)
7.1.1	移动的与固定的对话框	(75)
7.1.2	对话框的放缩	(75)
7.1.3	模式的与无模式的对话框	(76)
7.1.4	消息对话框	(77)
7.2	对话框的位置	(79)
7.3	对话框中使用命令按钮	(79)
7.3.1	推荐的按钮	(79)
7.3.2	默认按钮	(79)
7.3.3	动态按钮标签	(80)
7.3.4	导向相关对话框	(80)
7.3.5	按钮的排放	(80)
7.3.6	消息对话框中的命令按钮	(84)
7.4	对话框中字体	(84)
7.5	对话框中样本	(85)
第8章	常用对话框	(86)
8.1	文件操作	(86)
8.1.1	文件打开对话框(File Open)	(86)
8.1.2	文件保存对话框(File Save As)	(90)
8.1.3	文件创建对话框(File New)	(91)
8.2	打印	(91)
8.2.1	打印对话框(Print)	(91)
8.2.2	打印设置对话框(Print Setup)	(94)
8.3	文字查找和替换	(95)
8.3.1	命令名和菜单定位	(96)
8.3.2	对话类型和命令动作	(96)
8.3.3	标签	(97)
8.3.4	其它控制	(97)
8.4	字符属性	(97)
8.4.1	字符对话框控制	(99)
8.5	页设置对话框	(101)
8.6	About(应用程序-名称)对话框	(101)
8.6.1	推荐的成分	(101)
8.6.2	可选信息	(102)
第9章	对象连接和嵌入	(103)
9.1	复合文本	(103)
9.2	OLE 概念	(104)
9.3	OLE 接口	(105)

9.3.1	客户与服务器	(105)
9.3.2	插入对象	(107)
9.3.3	观察对象	(110)
9.3.4	激活对象	(111)
9.3.5	编辑对象	(114)
9.4	连接和连接对话框	(116)
9.4.1	Update 选择按钮	(116)
9.4.2	连接命令按钮	(117)
9.4.3	对话框控制按钮	(118)
9.4.4	连接状态项	(118)
9.4.5	连接更新的其它对话框	(118)
9.5	状态行推荐信息	(119)
第 10 章	光笔接口	(121)
10.1	光笔输入	(121)
10.1.1	点	(121)
10.1.2	写	(121)
10.1.3	拖动	(121)
10.1.4	手势	(121)
10.2	设计光笔接口	(124)
10.2.1	简单性和直接性	(124)
10.2.2	识别问题	(125)
10.2.3	硬件限制	(128)
第 11 章	其它问题	(129)
11.1	装入和初始化	(129)
11.1.1	内存检查	(129)
11.1.2	显示应用程序窗口	(129)
11.1.3	显示启动信息	(129)
11.2	用户级别和自制环境	(130)
11.2.1	扩展对话框	(130)
11.2.2	自制对话框	(130)
11.2.3	为残疾用户的考虑	(130)
11.3	帮助	(131)
11.3.1	访问帮助	(131)
11.4	国际考虑	(131)
11.4.1	接口文字	(132)
11.4.2	硬件	(132)
11.4.3	格式	(132)

前 言

介绍

应用程序设计指南为在 Microsoft Windows 图形环境下运行的应用程序提供用户界面设计指导,它描述了 Windows 用户界面的各个组成部分,为 Windows 应用程序的开发、设计人员解释了其设计原则。

目的

本指南的目的是提高 Windows 应用程序之间及应用程序内部在视觉上、功能上的一致性。这种一致性有许多优点:应用程序之间的界面是一致的,用户可以很快从一个应用程序转向另一应用程序。当一新的应用程序推向市场时,一致性可以加快学习过程,减少所需训练,结果也就提高了生产率;一致性还可以减轻由于应用程序的界面不同而引起的混乱,节省提高效率和训练的所需费用;它给用户一种稳定感,增加了用户对使用一致界面的应用程序的信任。

目标

本书介绍的绝大部分设置和方法是和 Windows 操作系统 V3.1 相兼容的,其中有些可能在较早版本的 Windows 环境中得不到支持。要了解其支持的详细信息,请参考 Microsoft Windows 软件开发程序包(SDK)文档资料。

由于应用程序的不断发展,本指南不可能为所有可能的界面提供建议。若应用程序需要的构件或技术本书未讨论,设计人员或开发人员可以依据第一章中所列的原则扩充现有指导,但应该尽量保持与其它 Windows 应用程序的一致性,并以此评估自己的应用程序。

本书重点在为基于 Windows 应用程序的开发提供建议,这些指导和 IBM 公司的公共用户访问(CUA: Common User Access,《高级用户界面设计指南》,Boca Raton, FL: IBM, 1989)版本 2.0 基本兼容,但本书并不向用户介绍 CUA 的有关建议。

本指南包括为 Windows V3.0 应用程序提供的建议,也包括有关 Windows V3.1 新特性的建议。

本指南是为英语国家的应用程序开发而准备的,但是许多指导对非英语国家的应用程序开发也同样适用。

实现

Microsoft Windows 的软件开发程序包(SDK)中含有实现本指南中介绍的例子,要

进一步了解,请参考 SDK 文献。

输入方法支持

本指南介绍有关鼠标和键盘的技术。也包括对光笔的讨论,作为一种输入设备,它正越来越得到大家认可。尽管许多情况下,鼠标用作为交互设备的首选工具,但是也应为无鼠标或偏爱键盘的系统提供键盘界面支持。对那些主要依靠键盘输入数据的应用程序(例如,数据库,字处理或表处理),许多用户宁愿使用键盘作为交互手段,他们不愿把手离开键盘。相反,除了帮助(HELP)之外,这对图形应用程序(如绘画程序)是很繁琐的。因此,使用键盘困难的地方应尽量避免使用它。

推荐级别

如前文所述,本指南的目的是提高应用程序与应用程序之间及应用程序内部的视觉、功能上的一致性。这里的内容作为一个工具提供给愿意使用它的人们,开发者在其用户界面设计中可以选择它们,这里没有统一的要求。

下面的几条定义为用户说明了这些指导的重要程度:

- 带有“推荐”(Recommended)字样的指导,表示为保证最大程度的一致性而提供的实现特定特性、功能、操作的一般方法。尽管一个应用程序不必包括所有推荐项,但是若存在某项,本指南描述了首选的实现方法。例如,F6 是推荐用于切换窗口区的键,但并不是所有的应用程序都有多个窗口区。有些指导可以由系统定义(例如,ALT+ESC),有些可以由应用程序定义(例如 F1)。
- 带有“可选”(Optional)字样的指导,表示一个应用程序可以采用的一般扩展功能。如果这些特性或操作要实现,本指导是首选的实现方法。文件(File)和视图(View)菜单说明了推荐指导和可选指导之间的区别:文件菜单被推荐到所有通过文件存取数据的应用程序中,而视图菜单是可选的,但它的内容要遵循实现性原则。
- 带有“建议”(Suggested)字样的指导,在和应用程序中其它规则不冲突的情况下,是应用程序应该遵循的特定指导。

键及键组合表示

- 键名使用小号大写字母,如 CTRL 或 SHIFT。
- 同时按组合键用加号相连,例如 CTRL+B 或 CTRL+SHIFT+B,表示用户在按住 CTRL 键时再按 B 键,或按住 CTRL 和 SHIFT 键的同时按 B 键。
- 顺序按组合键用逗号相连,例如,ALT,F 表示用户应该按下并释放 ALT 键,然后再按 F 键。

第 1 章 原则与方法

1.1 用户界面设计的原则

由于应用程序的不断发展,不可能为每一界面提供特定建议。即使一个应用程序所需的构件和技术没有在本指南中讨论,也应该遵循本节的一般原则。

1.1.1 用户控制

用户界面设计的最重要的原则之一是:用户应始终控制着应用程序,而不是相反。这项原则具有好几层含义:

首先,应用程序应尽量是交互式的,用户不应花很长时间等待某一处理完成。一般来讲,应用程序应该避免限制用户随时取得交互控制,如果必须这样做,应明确表示出来(例如改变光标形状),并且要易于学习和退出。

用户控制的第二层含义是用户自定义。由于用户的能力和爱好不同,用户应能够自定义界面的外观(包括美观,如颜色和功能),例如菜单的结构和内容。不过设计者应提供好的默认值而不应依靠用户来做这些设置(自定义的方法在第 11 章 11.2 节中讨论)。

最后,界面应该是有益于用户的任务,而不是引起对它本身的兴趣,最好的用户界面往往不引起人的注意。用户是要完成自己的任务而不是单纯使用计算机。他们要书写信件、计算利率、工程管理和准备讲稿,而不是拉一下滚动块,打开一下下拉式列表,拉一下下拉菜单,或在对话框中移来移去。

1.1.2 直接性

界面应该给用户直接的、直观的方法来完成他们的任务,对象-操作范式支持这条原则。用户首先选择对象(如图符、窗口或一段文字),然后选择对该对象的操作(如移动、关闭或加下划线)。

尽管不是所有的情况都合适,但是直接操作对象常常要比敲入复杂的命令来得容易。例如,用鼠标拖动一个窗口要比先估测一下新的位置坐标,然后输入对话框中去移动该窗口要容易得多。

1.1.3 一致性

一致性的两类广义含义是很重要的:与现实世界的一致性,应用程序与应用程序之间及应用程序内部的一致性。首先,应用程序应该是通过具体原则和自然映象关系建立在用户的现实世界经验上,使用熟悉的概念和原则可以减少用户必须学习的新材料,因而使应用程序易于使用。其次,一个应用程序应该在其内部及与其它应用程序之间在概念上、语

言上、视觉上和功能上保持一致性。这种一致性给用户带来好处,也给设计者、开发者带来好处,他们可以通过复用标准界面构件更快地生产出设计优良的应用程序。

有时,平台间的一致性和平台内部的一致性产生冲突。例如,两个平台可能提供不同的界面来完成同一功能。这种情况下,平台内部的一致性应优先考虑,因为大部分用户仅在一个平台内工作。

1.1.4 可辨性

一个应用程序界面应该在视觉上、概念上和语言上易于识别。视觉构件应该一看就懂,理想情况下,它们就是真实世界的模拟。其功能也应便于理解;概念比喻应该简单、接近真实世界;界面文字应该清楚、无二义性,并且不含俚语。

1.1.5 美观性

通过空间组合、对比和三维立体表示可以极大加强外表美观和视觉清晰。最好的界面总是功能强、易于使用和合意外观的结合。

1.1.6 反馈性

用户应得到对于操作的立即、可见的反馈。例如,当一个人拣起一支笔,触觉和视觉器官立即反馈:这支笔被选中了。同样,若用户用鼠标选择了一数据对象,应用程序应提供可见反馈,表示此对象被选中。图形反馈最为有效,但文字、声音也很有用(参看第3章3.6节)。

1.1.7 宽容性

用户喜欢通过试验、出错来掌握应用程序。这种自我激发的学习方式极为有效,但用户可能不是总清楚潜在的危险。即使一个精心设计的界面,用户也会遇到错误——笔误(无意中指向了错误的命令或数据)和逻辑错误(选择命令和数据上作出了错误的决定),界面应毫无痛苦、不加惩罚地容忍用户的探索和错误,应该把出错的机会减至最小,并从容地对待错误。错误信息不应暗示用户做错了什么,相反,应该客观地叙述问题,提供可能的解决办法(参看第3章3.6.1.2小节)。

1.1.8 认识人类的能力与局限

即使是五岁的孩子,也能处理许多语言、视觉任务,这些即使是最先进的计算机系统也是无能为力的。但是我们人类在感觉、记忆、推理上也有不可避免的局限。应用程序应该尊重这些局限,而不是强迫用户去克服。例如,不应要求用户计算那些计算机可以提供的信息(如某一年的某一天是星期几)。同样,不应要求用户回忆复杂的选择项或命令,相反,可选项应显式地列出来,辨认某项要比回忆它容易得多。

1.2 设计方法

要取得最高的效率,必须把设计原则和设计方法学联系起来,这个方法学就是把用户

放在设计过程的核心,把应用程序放在更广泛的范围内去考虑,为界面再设计和反复测试留有空间。

即使最富创造力、最有经验的设计者也不可能一次就设计出正确的界面。事实上,有时界面设计方面的经验反而妨碍找到正确的答案。设计者对一种界面太熟悉,就排除了新的或特殊用户的观点。应该在实际用户——而不仅是在同伴中检验新设计。日程表中应留有依测试使用结果而重新设计界面的时间,花在可用性测试上的时间是值得的,在设计早期发现界面问题比产品已经销售后发现要好得多。

在设计和测试界面时,记住应用程序将要在更广泛的环境中运行是很重要的。例如,将来是在一台独立的计算机上运行,还是作为网络的一部分?是独立运行还是和其它应用程序一起运行?在90年代不断发展的网络、多任务计算机环境中,能够方便地和其它用户或应用程序交换数据的应用程序比传统的独立应用程序有很大的优势,用户界面应该通过提供一种从一应用程序到另一应用程序一致性的数据交换技术,反映出这种新的集成技术,并为这种技术提供方便。

1.3 部分参考文献

Chew, Jane Carrasco, and John Whiteside, eds. *Empowering People: CHI'90 Conference Proceedings*. New York, NY: ACM Press, 1990.

Helander, Martin, ed. *Handbook of Human-Computer Interaction*. Amsterdam: North-Holland, 1988.

IBM Corporation. *Common User Access: Advanced Interface Design Guide*. Boca Raton, FL: IBM, 1989.

Laurel, Brenda, ed. *The Art of Human-Computer Interface Design*. Reading, MA: Addison-Wesley, 1990.

Nielsen, Jakob, ed. *Coordinating User Interfaces for Consistency*. Boston: Academic Press, 1989.

Norman, Donald A. *The Design of Everyday Things*. New York: Basic Books, 1988.

Robertson, S. P., G. M. Olson, and J. S. Olson, eds. *Reaching Through Technology: CHI'91 Conference Proceedings*. Reading, MA: Addison-Wesley, 1991.

Shneiderman, Ben. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Reading, MA: Addison-Wesley, 1987.

Tufte, Edward R. *Envisioning Information*. Cheshire, CT: Graphics Press, 1990. 另外,ACM/SIGCHI出版的关于人机交互的会议录是主要的参考基础。

第 2 章 主要输入构件

用户界面建立在规模相对较小的输入构件集上。这些构件定义了用户与计算机的交互手段,它们是鼠标和键盘。

2.1 鼠标输入

鼠标的主要操作是定位、点取和拖动。这些操作还可以和修饰键(SHIFT, CTRL 和 ALT)组合使用。

2.1.1 基本操作

- 按钮 1 是选择按钮,大部分鼠标动作靠按钮 1 完成^①。
- 按钮 2 这个按钮用于特定环境中的动作和选择。无原则的规定按钮 2 的操作会使用户产生混乱。

附加按钮也可支持,但一般情况没有附加按钮。因此,第三按钮(或附加按钮)应赋予界面中已有的功能或操作。例如,第三按钮(中间按钮)可用于区分选择和直接操纵(拖动)^②。

表 2.1 列出了基本鼠标元素,并给出了它们使用的例子。有关修饰键和鼠标组合使用的情况参看第 3 章 3.1.2 节。

表 2.1 鼠标输入

操 作	定 义	一般使用(按钮 1)
定位	移动光标(热点*)到屏幕上所需位置。	导向;为选择或控制操作做准备。
按	按下并保持按钮按下状态。	标识要被选中的对象。
点取	鼠标不动,按下并释放按钮。	选择插入点或项;执行控制;激活非活动窗口或控制。
双点取**	鼠标不动,在一特定时间内,两次按下并释放按钮。	操作的简捷启动方式,例如,激活图符、打开文件、选择词。
拖动	按下按钮,在鼠标移动时保持按钮成按下状态。	定义对象范围;移动或重置项的大小。
双拖动	按钮两次,并在鼠标移动时,保持按钮成按下状态。	用较大单位选择(例如以词为单位)。

* 热点是鼠标光标中的一点,它表示鼠标动作将起作用的屏幕上的准确位置。

** 这里没有建议三点或多点操作。应用程序可以自己定义它们,但要记住对有些用户来讲这类操作是很困难的。

① 默认情况下,按钮 1 是左按钮;但是,Microsoft Windows 控制面板允许用户切换左右按钮。

② 两按钮鼠标设备可以仿真第三个鼠标按钮,方法是同时使用两按钮,或者使用 ALT 键+按钮 2 组合。

当鼠标按钮按下时,表示将有鼠标动作,而当按钮释放时鼠标动作才得以证实。例如,光标在一个菜单项上按下按钮时,这项菜单应加亮显示,而当按钮释放后再恢复原样。类似地,在拖动操作中,按下按钮时,用户会得到可见的反馈(例如加亮显示),但按钮释放之前操作并未执行。这只是一个一般原则,也可能有例外,例如,用户在滚动箭头上按下按钮,滚动操作即已执行,只要按钮按住,此操作自动重复。

2.1.2 鼠标操作使用指南

鼠标操作不应要求用户有高度准确的目测能力,例如

- 若一个对象很小或很细,要选择它需要用户精确的定位,这种情况下应在对象的四周建立一热区扩大对象的选择范围。
- 对有些用户来讲,双点或双拖动动作中要求的快速按钮动作是困难的。因此绝不要把这类技术作为实现操作的唯一方法,例如打开文件。
- 不要要求用户去定位一个移动对象,当然在游戏中除外。

2.2 键盘输入



键盘输入涉及到打各种键:字符键、编辑键、方式键、导向键和加速(功能)键。

2.2.1 字符键

字符键可以分为数字、字母(a—z,0—9)、标点符号、制表符和回车、空格键。在有字符输入的应用程序中,按下一字符键,相应的字符就会出现在屏幕上(除特殊情况外,制表符、回车键是看不见的)。在非字符方式下,字符键可以作为其它操作的简捷表示。例如,从工具箱中选取一工具或通过匹配一列表中的首字符而选择某项。另外,制表符还用于导向(参看 2.2.4 节),回车键用于对话框中按默认按钮,而空格杠则用作默认选择键,作为隐式键盘选择(参看第 3 章 3.1.3 节)。

PC 键盘有两个键标有回车(ENTER):普通回车键和小键盘回车键。它们有相同的标识,因此功能应该相同。

2.2.2 编辑键

表 2.2 列出了编辑键和它们的功能。除非有别的说明,这些功能都仅适合字符编辑。

表 2.2 编辑键

键	推荐的功能
DEL	<ul style="list-style-type: none">• 若存在一选择,删除整个选择。• 若没有选择而有插入点:删除插入点右边的字符。
BACKSPACE*	<ul style="list-style-type: none">• 若存在一选择:删除整个选择。• 若没有选择而有插入点:删除插入点左边的字符。
INS	在插入方式(新字符将旧字符向右推)和覆盖方式(新字符代替旧字符)之间切换。

* 用 DEL 和 BACKSPACE 删除的字符不被放入裁剪板,详细信息参看第 3 章 3.7 节。

2.2.3 方式键

方式键改变其它键的动作,这类键有两种:切换键和修饰键。

2.2.3.1 切换键

每次按下并释放切换键,它都打开或关闭一特定方式。例如,INS 键在插入方式(新字符把旧字符向右推)和覆盖方式(新字符代替旧字符)之间切换。

表 2.3 列出了主要的切换键,功能键也可以用作切换键。

表 2.3 切 换 键^①

键	功 能
INS	在插入方式(新字符把旧字符向右推)和覆盖方式(新字符代替旧字符)之间切换
CAPS LOCK	大小写字母切换。
NUM LOCK	数字键盘键数字与方向切换。
SCROLL LOCK	导向键滚动数据,不移动光标;保留已有选择。
F8	切换扩展方式。在扩展方式下,选择操作等同于 SHIFT 键按下时的方向键与鼠标操作。
SHIFT+F8	切换增加方式。在此方式下可以通过键盘做离散选择;导向键移动光标而不影响已有选择,SPACEBAR 改变选择项的选择状态。

* 尽管这里推荐了这些功能,应用程序没有必要支持所有这些键。

2.2.3.2 修饰键

修饰键有 SHIFT, CTRL 和 ALT。与切换键相似,修饰键也改变其它键的动作。但和切换键不同的是,修饰键建立的方式只有在此键按下时起作用。换句话说,这个方式是临时方式,用户必须按着键。这种临时方式比自保持方式好,它可以避免用户忘了某种方式在起作用。因此,如果要求的操作能够迅速完成,那么修饰键比切换键好用。

表 2.4 列出了最常用的修饰键功能,更详细的情况参看第 3 章中涉及表中提到功能的有关部分。通常情况下,如果修饰键的释放取消或改变了操作,那么在操作的开始和执行过程中修饰键必须按下。

表 2.4 修 饰 键

键	典 型 功 能
SHIFT	<ul style="list-style-type: none">• 作用于数字、字母键,得到大写字母,或者是标在键上半部的字符*。• 作用于鼠标或导向键,扩展或缩小连续选择的范围。• 作用于功能键,改变动作的含义(例如,F1 打开帮助应用程序窗口,按 SHIFT+F1 进入帮助方式)。
CTRL	<ul style="list-style-type: none">• 作用于鼠标点取操作,选择或取消选择一项而不影响前面的选择。• 作用于字母键,得到加速键。• 作用于导向键,导向键移动单位比无修饰键的导向键移动单位大。
ALT	作用于字母键,移向以该字符作为助记符的菜单或控制。

* 当 CAPS LOCK 按下时,得到小写字符。

^① 尽管有推荐设置,但是应用程序不必支持所有这些键。

2.2.4 导向键

导向键是 HOME、END、PAGE DOWN、PAGE UP 四个箭头键 (LEFT、RIGHT、UP 和 DOWN) 和 TAB 键。表 2.5 列出了这些键的独立功能和与修饰键的组合功能。CTRL+键组合产生的移动增量比不用修饰键的移动大。导向操作也可以赋予其它没列在这儿的键(如功能键)。

表 2.5 导 向 键

键	无修饰键移光标到…	CTRL+键移光标到…
HOME	行开始(当前行的最左边)。	数据的开始(当前域或文本的左上角)*。
END	行结束(当前行中数据的最右端)。	数据的结束(当前域或文本中数据的右下角)**。
PAGE UP	向上滚一屏(前一屏,水平位置不变)。	屏幕的左边/开始(窗口的顶端,或屏幕的左边)。
PAGE DOWN	向下滚一屏(下一屏,水平位置不变)。	屏幕的右边或底端(窗口底端,或向右移一屏)。
LEFT ARROW	左移一单位 §。	左移一单位(比前面的单位大) § §。
RIGHT ARROW	右移一单位 §。	右移一单位(比前面的单位大) § §。
UP ARROW	上移一个单位或一行。	上移一单位(比前面的单位大) ##。
DOWN ARROW	下移一个单位或一行 #。	下移一单位(比前面的单位大) ##。
TAB	对话框:下一域可以从左到右或从上到下,依设计者的要求不同。到最后一域后,折回第一域(SHIFT+TAB 以相反的顺序移动)。	未定义。

* 如果无左边界,本键也可用于移到顶端。

** 如果无右边界,也可用于移到底端。

§ 对文本而言,以字符为单位。

§ § 对文本而言,一般以词为单位(到下一或上一词的开始),其它用法也是可以的,只要遵循 CTRL+键表示比不加修饰键移动单位大的一般原则就行。

一般保持同一位置。

对文本而言,一般表示以段为单位,其它用法也可以,但要遵循上条规则中的一般原则。

2.2.5 加速键

加速键或组合键可为经常使用的操作提供更快的调用方法。功能键和 CTRL+字母组合常常用于产生加速键。除了 2.2.5.3 中提到的特殊情况外,也可以用 ALT+功能键组合加速键。注意,不推荐使用 ALT+字母组合表示加速键,因为它们是键盘访问菜单和控制的标准方法。

2.2.5.1 功能加速键

表 2.6 列出了推荐的 PC 功能键。这里没有分配作用的功能键供应用程序使用。

表 2.6 推荐的 PC 功能键

	不修饰	SHIFT	CTRL	ALT
F1	帮助(第 11 章, 11.3.1 节)	进入帮助方式(第 11 章, 11.3.1 节)	(未推荐)	(未推荐)
F2, F3	(未推荐)	(未推荐)	(未推荐)	(未推荐)
F4	(未推荐)	(未推荐)	关闭文本窗口(第 5 章, 5.4.1 节)	关闭应用程序窗口(第 5 章, 5.4.1 节)
F5	(未推荐)	(未推荐)	(未推荐)	(未推荐)
F6*	顺时针在活动窗口中移向下一窗口区(第 4 章, 4.3.5.2.2)	逆时针在活动窗口中移向下一窗口区(第 4 章, 4.3.5.2.2)	移到下一文本窗口; 最上面的窗口进入堆栈底(第 4 章, 4.3.5.2.3) (加 SHIFT 键; 与上述操作方向相反)	移到应用程序的打开的下一非文本窗口(第 4 章, 4.3.5.2.3)。(加 SHIFT 键; 与上述操作方向相反)
F7	(未推荐)	(未推荐)	(未推荐)	(未推荐)
F8	若支持的话, 切换扩展方式(第 3 章, 3.1.1.5)	若支持的话, 切换增加方式(第 3 章, 3.1.3.2)	(未推荐)	(未推荐)
F9	(未推荐)	(未推荐)	(未推荐)	(未推荐)
F10	切换菜单栏活动状态(与 CUA2.0 相兼容)	(未推荐)	(未推荐)	(未推荐)
F11, F12	(未推荐)	(未推荐)	(未推荐)	(未推荐)

* 除了这里讨论的键以外, 应用程序可以定义自己的导向键。

若应用程序的用户不喜欢有功能键的键盘, 应用程序应避免使用功能键作为完成主要操作的唯一方法。

2.2.5.2 控制加速键

CTRL+字母也可用于表示键盘加速键。表 2.7 列出了推荐的等价于 Edit 菜单命令的 CTRL+字母加速键。若一个应用程序不支持某推荐组合键, 那么应避免赋予此组合键任何其它功能。

表 2.7 推荐的 CTRL+字母加速键^①

键	功 能
CTRL+Z	逆操作 (Undo)
CTRL+X	剪 (Cut)
CTRL+C	拷贝 (Copy)
CTRL+V	贴 (Paste)

^① Windows V3.1 中, Undo, Cut, Copy 和 Paste 的加速键是新的。为了向上兼容, 我们推荐设计在 Windows V3.0 下运行的应用程序也支持旧的加速键: Undo=ALT+BACKSPACE, Cut=SHIFT+DEL, Copy=CTRL+INS 和 Paste=SHIFT+INS。但是旧的加速键不应写入用户手册, 也不在 Edit 菜单中列出。