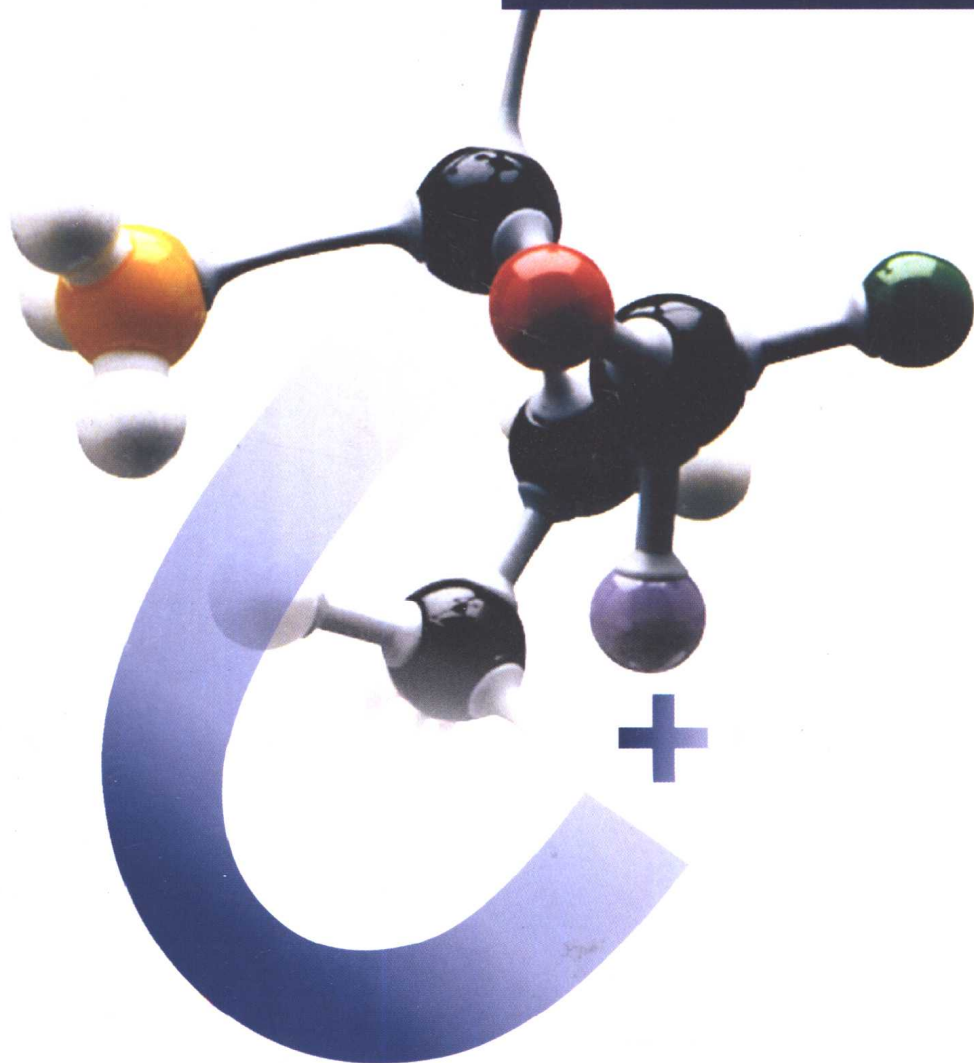




DirectInput原理

与API参考

武永康 编著



清华大学出版社
<http://www.tup.tsinghua.edu.cn>



DirectInput 原理与 API 参考

武永康 编著

清华大学出版社

(京)新登字 158 号

内 容 简 介

DirectInput 是 DirectX 应用程序设计接口的主要组件之一。DirectInput 为 Windows 环境的游戏和实时多媒体应用程序处理输入设备信息,尤其对游戏控制器和力反馈设备提供了完备的支持。本书完整地介绍了 DirectInput,包括 DirectInput 的结构和功能,以及 DirectInput 应用程序设计原理,给出了实用的 DirectInput 应用程序设计接口参考信息。本书以 Visual C++ 和 Visual Basic 两种语言环境为背景,分别用实例阐明了 DirectInput 应用程序设计的细节。

本书是 DirectInput 应用程序设计参考书,为 Windows 环境的 2D、3D 游戏和实时多媒体应用程序的开发,提供了实用的信息。本书适合实时多媒体和游戏开发人员,大专院校相关专业的师生和程序设计爱好者阅读。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无防伪标签者不得销售。

书 名: DirectInput 原理与 API 参考

作 者: 武永康

出 版 者: 清华大学出版社(北京清华大学学研大厦,邮编:100084)

<http://www.tup.tsinghua.edu.cn>

责任编辑: 许振伍

印 刷 者: 北京密云胶印厂

发 行 者: 新华书店总店北京发行所

开 本: 787×1092 1/16 **印张:** 21.75 **字数:** 509 千字

版 次: 2001 年 7 月第 1 版 2001 年 7 月第 1 次印刷

书 号: ISBN 7-302-04484-8/TP·2649

印 数: 0001 ~ 5000

定 价: 32.00 元

目 录

第 1 部分 DirectInput 原理

第 1 章 DirectInput 基础	3
1.1 DirectInput 的层次结构	3
1.1.1 DirectInput 层次结构概述	3
1.1.2 DirectInput 与 Windows 集成	5
1.1.3 人机接口设备	6
1.2 DirectInput 的创建	6
1.2.1 创建 DirectInput	6
1.2.2 DirectInput 设备枚举	6
第 2 章 DirectInput 设备	9
2.1 设备的设置	9
2.2 创建 DirectInput 对象	9
2.3 设备性能	11
2.4 协作级别	12
2.4.1 前台和后台	13
2.4.2 独占级与非独占级	14
2.5 设备对象枚举	14
2.6 设备数据格式	16
2.7 设备特性	17
2.8 获得设备	20
2.9 识别设备的变化	21
第 3 章 DirectInput 设备数据基本概念	23
3.1 缓冲数据和立即数据	23
3.2 时间印章和序列号	25
3.3 轮询和事件	26
3.4 相对轴坐标和绝对轴坐标	27
3.5 输出数据	27

第 4 章 鼠标数据	30
4.1 立即鼠标数据	30
4.2 缓冲鼠标数据	31
4.3 解释鼠标轴数据	33
4.4 检查丢失的鼠标输入	34
第 5 章 键盘数据	35
5.1 立即键盘数据	35
5.2 缓冲键盘数据	36
5.3 解释键盘数据	37
5.4 检查丢失的键盘输入	38
第 6 章 游戏杆数据	39
6.1 立即游戏杆的数据	39
6.2 缓冲游戏杆的数据	40
6.3 游戏杆的轴数据	42
6.4 检查游戏杆状态	44
第 7 章 力反馈	45
7.1 力反馈的基本概念	45
7.2 效果枚举	47
7.3 装载一个文件中的效果	49
7.4 有关被支持效果的信息	50
7.5 创建一种效果	51
7.6 效果的方向	53
7.7 包络和偏移量	56
7.8 效果重现	57
7.9 下载和卸载效果	58
7.10 修改效果	59
7.11 增益	60
7.12 力反馈状态	61
7.13 枚举效果对象	62
7.14 效果类型	62
7.14.1 常力	63
7.14.2 斜坡力	64
7.14.3 周期效果	65
7.14.4 状况	67
7.14.5 定制力	69
7.14.6 设备特定的效果	70

第 8 章 DirectInput For C/C++ 应用举例	72
8.1 使用键盘	72
8.1.1 步骤 1: 创建 DirectInput 对象	73
8.1.2 步骤 2: 创建 DirectInput 键盘设备	73
8.1.3 步骤 3: 设置键盘数据格式	74
8.1.4 步骤 4: 设置键盘特性	74
8.1.5 步骤 5: 获得使用键盘的权利	74
8.1.6 步骤 6: 从键盘检索数据	75
8.1.7 步骤 7: 关闭 DirectInput 系统	76
8.1.8 示例函数 1: DI_Init	76
8.1.9 示例函数 2: DI_Term	77
8.2 使用鼠标	78
8.2.1 步骤 1: 创建鼠标设备	78
8.2.2 步骤 2: 设置鼠标数据格式	79
8.2.3 步骤 3: 设置鼠标特性	79
8.2.4 步骤 4: 为鼠标的缓冲输入做准备	79
8.2.5 步骤 5: 管理对鼠标的访问	81
8.2.6 步骤 6: 从鼠标检索被缓冲的数据	82
8.3 使用游戏杆	92
8.3.1 步骤 1: 枚举游戏杆	92
8.3.2 步骤 2: 创建 DirectInput 游戏杆设备	93
8.3.3 步骤 3: 设置游戏杆数据格式	94
8.3.4 步骤 4: 设置游戏杆特性	94
8.3.5 步骤 5: 获得使用游戏杆的权利	96
8.3.6 步骤 6: 从游戏杆检索数据	97
8.4 使用力反馈	98
8.4.1 步骤 1: 枚举力反馈设备	99
8.4.2 步骤 2: 创建 DirectInput 力反馈设备	99
8.4.3 步骤 3: 枚举被支持的效果	101
8.4.4 步骤 4: 创建一种效果	102
8.4.5 步骤 5: 操纵一种效果	103
8.4.6 步骤 6: 改变效果	104
第 9 章 DirectInput For Visual Basic 应用举例	105
9.1 使用键盘	105
9.1.1 步骤 1: 创建 DirectInput 和键盘设备	105
9.1.2 步骤 2: 设置键盘参数	106
9.1.3 步骤 3: 获得使用键盘的权利	106

9.1.4	步骤 4: 从键盘检索立即数据	107
9.2	使用鼠标	112
9.2.1	步骤 1: 设置鼠标	112
9.2.2	步骤 2: 设置通知	113
9.2.3	步骤 3: 管理对鼠标的独占管理	113
9.2.4	步骤 4: 检索缓冲数据	114
9.3	使用游戏杆	118
9.3.1	步骤 1: 枚举和创建游戏杆	118
9.3.2	步骤 2: 获得游戏杆性能	119
9.3.3	步骤 3: 设置游戏杆属性	121
9.3.4	步骤 4: 从游戏杆检索立即数据	122
9.4	使用力反馈设备	127
9.4.1	步骤 1: 初始化力反馈设备	128
9.4.2	步骤 2: 设置设备属性	129
9.4.3	步骤 3: 创建一种效果	129
9.4.4	步骤 4: 修改效果	130

第 2 部分 DirectInput 应用程序设计接口

第 10 章	DirectInput For C/C++ 接口	147
10.1	COM 接口	147
10.1.1	IDirectInput7 接口	147
10.1.2	IDirectInputDevice7 接口	154
10.1.3	IDirectInputEffect 接口	184
10.2	函数	193
10.2.1	DirectInputCreate 函数	194
10.2.2	DirectInputCreateEx 函数	195
10.3	回调函数	196
10.3.1	DIEnumCreatedEffectObjectsCallback	197
10.3.2	DIEnumDeviceObjectsCallback	197
10.3.3	DIEnumDevicesCallback	198
10.3.4	DIEnumEffectsCallback	198
10.3.5	DIEnumEffectsInFileCallback	198
10.4	宏	199
10.4.1	DIDFT_GETINSTANCE	199
10.4.2	DIDFT_GETTYPE	200
10.4.3	DIDFT_MAKEINSTANCE	200
10.4.4	DIEFT_GETTYPE	200

10.4.5	DIMAKEUSAGEDWORD	201
10.4.6	DISEQUENCE_COMPARE	201
10.4.7	GET_DIDEVICE_SUBTYPE	202
10.4.8	GET_DIDEVICE_TYPE	202
10.5	数据结构	202
10.5.1	DICONDITION 结构	203
10.5.2	DICONSTANTFORCE 结构	205
10.5.3	DICUSTOMFORCE 结构	205
10.5.4	DIDATAFORMAT 结构	206
10.5.5	DIDEVCAPS 结构	208
10.5.6	DIDEVICEINSTANCE 结构	210
10.5.7	DIDEVICEOBJECTDATA 结构	213
10.5.8	DIDEVICEOBJECTINSTANCE 结构	214
10.5.9	DIEFFECT 结构	217
10.5.10	DIEFFECTINFO 结构	220
10.5.11	DIEFFESCAPE 结构	222
10.5.12	DIENVELOPE 结构	223
10.5.13	DIFILEEFFECT 结构	224
10.5.14	DIJOYSTATE 结构	224
10.5.15	DIJOYSTATE2 结构	225
10.5.16	DIMOUSESTATE 结构	228
10.5.17	DIMOUSESTATE2 结构	229
10.5.18	DIOBJECTDATAFORMAT 结构	230
10.5.19	DIPERIODIC 结构	232
10.5.20	DIPROPDWORD 结构	233
10.5.21	DIPROPGUIDANDPATH 结构	234
10.5.22	DIPROPHEADER 结构	234
10.5.23	DIPROP RANGE 结构	235
10.5.24	DIPROPSTRING 结构	236
10.5.25	DIRAMPFORCE 结构	237
10.6	设备常量	237
10.6.1	键盘设备常量	238
10.6.2	鼠标设备常量	243
10.6.3	游戏杆设备常量	244
10.7	返回值	245
第 11 章	DirectInput For Visual Basic 参考	249
11.1	类	249

11.1.1	DirectInput 类	249
11.1.2	DirectInputDevice 类	252
11.1.3	DirectInputDeviceInstance 类	273
11.1.4	DirectInputDeviceObjectInstance 类	279
11.1.5	DirectInputEffect 类	284
11.1.6	DirectInputEnumDeviceObjects 类	290
11.1.7	DirectInputEnumDevices 类	291
11.1.8	DirectInputEnumEffects 类	292
11.2	数据类型	295
11.2.1	DICONDITION 类型	296
11.2.2	DICONSTANTFORCE 类型	297
11.2.3	DIDATAFORMAT 类型	297
11.2.4	DIDEVCAPS 类型	298
11.2.5	DIDEVICEOBJECTDATA 类型	299
11.2.6	DIEFFECT 类型	300
11.2.7	DIENVELOPE 类型	302
11.2.8	DIJOYSTATE 类型	302
11.2.9	DIJOYSTATE2 类型	304
11.2.10	DIKEYBOARDSTATE 类型	306
11.2.11	DIMOUSESTATE 类型	306
11.2.12	DIOBJECTDATAFORMAT 类型	307
11.2.13	DIPERIODICFORCE 类型	308
11.2.14	DIPROPLONG 类型	309
11.2.15	DIPROPRANGE 类型	310
11.2.16	DIRAMPFORCE 类型	311
11.3	枚举类型	311
11.3.1	CONST_DICOMMONDATAFORMATS 枚举类型	312
11.3.2	CONST_DICONDITIONFLAGS 枚举类型	312
11.3.3	CONST_DIDATAFORMATFLAGS 枚举类型	313
11.3.4	CONST_DIDEVCAPSFLAGS 枚举类型	313
11.3.5	CONST_DIDEVICEOBJINSTANCEFLAGS 枚举类型	315
11.3.6	CONST_DIDEVICETYPE 枚举类型	316
11.3.7	CONST_DIDFTFLAGS 枚举类型	318
11.3.8	CONST_DIDGDDFLAGS 枚举类型	320
11.3.9	CONST_DIEFFFLAGS 枚举类型	320
11.3.10	CONST_DIEFTFLAGS 枚举类型	320
11.3.11	CONST_DIEGESFLAGS 枚举类型	322
11.3.12	CONST_DIENUMDEVICESFLAGS 枚举类型	322

11.3.13	CONST_DIEPFLAGS 枚举类型	322
11.3.14	CONST_DIESFLAGS 枚举类型	324
11.3.15	CONST_DIGFFSFLAGS 枚举类型	324
11.3.16	CONST_DIJOYSTICKOFS 枚举类型	325
11.3.17	CONST_DIKEYFLAGS 枚举类型	327
11.3.18	CONST_DIMOUSEOFS 枚举类型	327
11.3.19	CONST_DINPUT 枚举类型	327
11.3.20	CONST_DINPUTERR 枚举类型	328
11.3.21	CONST_DIPHFLAGS 枚举类型	328
11.3.22	CONST_DISCLFLAGS 枚举类型	329
11.3.23	CONST_DISDDFLAGS 枚举类型	329
11.3.24	CONST_DISFFCFLAGS 枚举类型	330
11.4	键盘设备常量	331
11.5	错误码	334

第 1 部分

DirectInput 原理

在游戏和多媒体应用程序中,人机交互处理是重要的组成部分,迅速、高效地处理用户的输入信息是这类应用程序的目标之一。DirectX 7.0 的 DirectInput 就是为此而设计的。DirectInput 除了为 Microsoft Win32 API 不支持的设备提供新服务以外,还可以不依赖于 Windows 的消息直接用硬件驱动更快地实现数据的输入。即使处于后台的应用程序,DirectInput 也能够从输入设备中检索数据。同时 DirectInput 对任何输入设备都能提供全面支持,尤其是对力反馈设备的支持,而在许多高级 3D 游戏和实时多媒体应用程序中,力反馈设备是主要输入设备。

DirectInput 扩展的服务和改善的功能使之成为运行于 Windows 平台的游戏、仿真及其他实时交互应用程序的有力工具。

第 1 部分的内容是对 DirectInput 基本概念和组件的介绍,并为实际应用中 DirectInput 系统的实现提供足够的信息,包括以下内容:

- ✓ DirectInput 基础
- ✓ DirectInput 设备
- ✓ DirectInput 设备数据基本概念
- ✓ 鼠标数据
- ✓ 键盘数据
- ✓ 游戏杆数据
- ✓ 力反馈
- ✓ DirectInput For C/C++ 应用示例
- ✓ DirectInput For Visual Basic 应用示例

第 1 章 DirectInput 基础

DirectInput 是输入设备的应用程序设计接口(API),能识别如下类型基本输入设备:

- ✓ 鼠标,包括所有类似于鼠标的设备,如触摸板、跟踪球等。
- ✓ 键盘,标准系统键盘。
- ✓ 游戏杆和其他游戏操控器,也包括力反馈(输入/输出)设备等。

实际上用户的输入数据来自上述 3 种类型输入设备上的零件,如键、按钮、轴、视点帽等。DirectInput 如何表示这些物理设备以及这些设备上的零件,如何处理来自这些设备的信息,是本章讨论的中心。下面先介绍 DirectInput 的层次结构,然后讨论如何创建 DirectInput。

1.1 DirectInput 的层次结构

本节介绍 DirectInput 的基本结构以及 DirectInput 如何作用于 Windows 操作系统和输入硬件,包括如下内容。

- ✓ DirectInput 层次结构概述
- ✓ 与 Windows 的集成
- ✓ 人机接口设备

1.1.1 DirectInput 层次结构概述

【C++】

DirectInput 实现的基本层次结构由支持 IDirectInput7 COM 接口的 DirectInput 对象每个提供数据的输入设备的对象组成。这些输入设备都包含设备对象的实例、单个控制或开关、键、按钮、轴等更简单的设备对象。

DirectX 7.0 中的 DirectInput 组件包括如下 COM 接口:

- IDirectInput7

IDirectInput7 接口的方法枚举、创建并取回 DirectInput 设备的状态,同时初始化 DirectInput 对象并调用 Windows 控制面板的一个实例。

- IDirectInputDevice7

IDirectInputDevice7 接口的方法取得并释放对 DirectInput 设备的访问,管理设备属性和信息,设置操作,执行初始化,创建和实现力反馈效果并调用一个设备控制面板。

- IDirectInputEffect

IDirectInputEffect 接口的方法管理力反馈设备的作用效果。

注释:对象既指描述代码对象又指一个单独的输入设备的控制。在本文中,设备对象是指一个输入设备而不是指实例化的 IDirectInputDevice7 接口。代码对象代表着全体设备,即 DirectInputDevice 对象。

每一个 DirectInput 设备对象表示一个输入设备,如鼠标、键盘或游戏杆等。实际上一个硬件是由几个不同类型的输入设备构成的,如带有触模板的键盘,可以由两个以上的 DirectInputDevice 对象来表示,力反馈设备可以由处理输入和输出的两个游戏杆对象表示。DirectInputDevice 对象实例化 IDirectInputDevice7 接口。

通过使用 IDirectInputDevice7::EnumObjects 方法,应用程序可以确定可用的设备对象类型和数量。个别设备对象未被封装成代码对象,但在 DIDEVICEOBJECTINSTANCE 结构进行了描述。

IDirectInputEffect 接口用来表示力反馈效果,这个接口的方法可用来创建、修饰、打开和停止效果。

所有 DirectInput 接口都有 ANSI 和 Unicode 两种版本可用。如果在编译期间定义了 UNICODE,则要使用 Unicode 版本。

【Visual Basic】

DirectX for Visual Basic 类库包括如下 DirectX 类:

- DirectX7

DirectX7 类是 DirectX for Visual Basic 应用程序的主类。这个类包含创建 DirectInput 应用程序主对象的方法也包含 DirectInput 使用的全局方法。这个类的一个实例通常是 DirectX for Visual Basic 应用程序中创建的第 1 个对象,可以通过把这个对象变量声明为 New 来获取该实例,如下所示:

```
Public objDX7 As New DirectX7
```

- DirectInput

DirectInput 类表示 DirectInput 系统。每个应用程序应该有一个这个类的单独对象,该类用于枚举可用的设备,创建并获取设备的状态以及调用 Windows 控制面板的一个实例。使用 DirectX7.DirectInputCreate 方法可以获得 DirectInput 对象。

- DirectInputDevice

DirectInputDevice 类的方法用于取得并释放对 DirectInput 设备的访问,管理设备属性和信息,设置性能,执行初始化并调用设备的属性表。DirectInputDevice 对象是通过使用 DirectInput.CreateDevice 方法获取的。

- DirectInputDeviceInstance

DirectInputDeviceInstance 类用于获取关于 DirectInput 设备实例的信息。这个类的对象由 Direct-InputDevice.GetDeviceInfo 和 DirectInput-Device.GetDeviceInfo 方法返回。

- DirectInputDeviceObjectInstance

DirectInputDeviceObjectInstance 类用于表示输入设备上的诸如按钮或轴等对象。DirectIn-

putDeviceObjectInstance 类的对象由 DirectInputDevice.GetObjectInfo 和 DirectInputEnumDeviceObjects.GetItem 方法返回。

- DirectInputEffect

DirectInputEffect 类的对象表示由应用程序创建的一个力反馈效果。该对象由 DirectInputDevice.CreateEffect 方法创建。

- DirectInputEnumDeviceObjects

DirectInputEnumDeviceObjects 类用于枚举一个系统上安装的 DirectInputDevice 对象。调用 DirectInputDevice.GetDeviceObjectsEnum 方法能够创建这个对象并填充数据。

- DirectInputEnumDevices

DirectInputEnumDevices 类用于枚举安装在系统上的 DirectInput 设备。调用 DirectInput.GetDIEnumDevices 方法能够创建这个对象并填充数据。

- DirectInputEnumEffects

DirectInputEnumEffects 类的对象用于表示设备上所支持的力反馈效果的集合,并用来获取关于效果的信息。调用 DirectInputDevice.GetEffectsEnum 方法可以创建这个对象。

对于每一个应用程序使用的输入设备,DirectInput 工具的基本结构是由一个 DirectInput 对象和一个 DirectInputDevice 对象组成的。DirectInputDevice 对象用来获取输入数据。

对于所有的可用的设备,无论是否被 DirectInput 使用,都可以用 DirectInputDeviceInstance 对象来表示,该对象可以用于获取设备的多种信息。

每一个 DirectInputDevice 对象代表一个输入设备,例如鼠标、键盘或者游戏杆。本书中,游戏杆包括除鼠标和键盘以外的所有游戏控制器。由多个不同输入设备构成的硬件,例如带有触摸板的键盘,可以用有两个或两个以上 DirectInputDevice 对象表示。力反馈设备可用由单个的游戏杆对象表示,该对象用于处理用户输入和输出。

所有的设备有相应的设备对象,即各自的控制或者开关,例如键、按钮或者轴。

所有的设备对象都用一个 DirectInputDeviceObjectInstance 类的实例表示,这个类的方法可以用于获取关于该设备对象的信息,但输入数据总是由 DirectInputDevice 类似总数来获取。

注释:对象既表示代码对象又表示输入设备上单独的控制。本书中,设备对象既表示一个 DirectInputDevice 对象,也表示着输入设备控制。

可以被枚举设备和设备对象,得到的集合由 DirectInputEnumDevices 和 DirectInputEnumDeviceObjects 对象表示。

1.1.2 DirectInput 与 Windows 集成

因为 DirectInput 直接作用于设备的驱动程序,所以可抑制或忽略鼠标和游戏杆的信息。在独占模式使用鼠标时,DirectInput 屏蔽了鼠标信息,因此 Windows 不显示系统光标。DirectInput 也忽略控制面板中对鼠标和键盘的设置。

对于键盘,DirectInput 不使用字符重复的设置。使用缓冲数据时,DirectInput 将每次的

键盘敲击都解释为单独事件,不循环。使用立即数据时,DirectInput 只关心键盘当时的物理状态,而不是由 Windows 解释的键盘事件。

对于鼠标,DirectInput 忽略控制板设置的诸如加速和按钮转换。同时,DirectInput 直接作用于鼠标驱动程序,绕过解释窗口应用程序鼠标数据的 Windows 子系统。

注释: DirectInput 只识别驱动程序本身的设置,例如三按钮鼠标,驱动程序将鼠标的中按钮设为双击的快捷按钮,DirectInput 则将中按钮的单击报告为左按钮的双击。

对于游戏杆或其他游戏控制器,DirectInput 可使用控制面板中设置的刻度。

1.1.3 人机接口设备

人机接口设备(HID)是使用串行总线(USB)标准的一个类。DirectInput 完全支持符合 HID 的设备。HID 设备获取数据的方式与传统设备完全相同。另外,HID 设备可以用于输出,例如打开或关掉键盘的 LED 灯。

通过查询 HID 设备,DirectInput 可以确定它的使用页(*usage page*)和用法(*usage*),这些都是预先定义的含有设备类型和子类型信息的代码。

HID 控制分组在集合中,集合可以嵌套。

1.2 DirectInput 的创建

本节将讨论如下内容:

- ✓ 创建 DirectInput
- ✓ 枚举 DirectInput 设备

1.2.1 创建 DirectInput

【C++】

任何 DirectInput 应用程序完成的第一步是获取 IDirectInput7 接口,完成这个任务的最容易的办法是调用 DirectInputCreateEx 函数。应该创建一个单独的 DirectInput 对象,并且不要在应用程序关闭之前释放它。

【Visual Basic】

DirectInput 应用程序完成的第一步是创建 DirectInput 对象,可以调用 DirectX7.DirectInputCreate 方法完成这个任务。

应该创建一个单独的 DirectInput 对象,并且不要在应用程序完成之前释放它。

1.2.2 DirectInput 设备枚举

DirectInput 能够查询所有可能的输入设备,确定这些设备是否被正确地连接,并返回

其状态信息,这个过程称为枚举。

【C++】

如果应用程序仅使用标准的键盘或鼠标,或两者都使用,则不必枚举可用的输入设备。就像在创建 DirectInput 设备一节中说明的那样,调用 IDirectInput7::CreateDeviceEx 方法时使用预定义的全局变量即可。

对于其他所有输入设备,或具有多个键盘和鼠标设备的系统,就要调用 IDirectInput7::EnumDevices 获取至少一个全局惟一标识符 GUIDs,用以创建 DirectInputDevice 对象,可能还要枚举用户选择的可用的输入设备。

如下所示的程序代码是用 IDirectInput7::EnumDevices 方法实现的枚举的例程:

```
/* lpdi is a valid IDirectInput7 interface pointer. */
GUID KeyboardGUID = GUID_SysKeyboard;
lpdi->EnumDevices(DIDEVTYPE_KEYBOARD,
                 DIEnumDevicesCallback,
                 &KeyboardGUID,
                 DIEDFL_ATTACHEDONLY);
```

第一个参数指定将枚举的设备类型。如果无论何种类型都要枚举所有的设备,则这个参数为 NULL,否则是 DIDEVICEINSTANCE 结构中的一个 DIDEVTYPE * 值。第二个参数是枚举某个设备时对应的回调函数的指针,这个函数可以任意命名。本文使用占位符的名字 DIEnumDevicesCallback。第三个参数是要传递给回调函数的任意的 32 位值。本例中是一个 GUID 类型的变量指针,由回调函数指定键盘实例 GUID 时传入的。第四个参数是一个标志,用于确定是枚举所有的设备,还是只枚举已连接的设备 DIEDFL_ALLDEVICES 和 DIEDFL_ATTACHEDONLY 标志之一。

如果应用程序当前使用多个输入设备,则回调函数就是初始化被枚举的设备的好地方。可以从回调函数获取设备的实例 GUID,也可以在回调函数中完成其他处理,例如寻找特定的设备子类型,把设备名称加进列表框中。

代码用于检查是否存在一个增强键盘,只要其存在就停止枚举过程。该实例把找到键盘的实例 GUID 分配给 KeyboardGUID 变量,该变量在前面 EnumDevices 调用的例子中,作为 pvRef 参数传入,可以在以后对 IDirectInput7::CreateDeviceEx 成员函数的调用中使用该变量:

```
BOOL hasEnhanced;

BOOL CALLBACK DIEnumKbdCallback(LPCDIDEVICEINSTANCE lpddi,
                                LPVOID pvRef)
{
    *(GUID*) pvRef = lpddi->guidInstance;
    if (GET_DIDEVICE_SUBTYPE(lpddi->dwDevType) ==
        DIDEVTYPEKEYBOARD_PCENH)
    {
        hasEnhanced = TRUE;
        return DIENUM_STOP;
    }
}
```