

Microsoft(DirectX(3

ffffffŠ”fff

ffffff,<□,,,,,□●,□—
—,,,<%o,,, ““ <Š“,,,,,•Ž •—,,,• “,,,,,,
Microsoft , ,ffffff,< ,,,, ffff fff,“,,, ’,“< • ’ Ć ,,’“ —
Ć,•Ž,,, ,ffffff, Microsoft , —,,,<%o,,, “< • ’ Ć ,,’“ —Ć,<%o,—
,,,,,,

(1996 Microsoft Corporation. All rights reserved.

Microsoft ActiveMovie Direct3D DirectDraw DirectInput DirectPlay DirectS
ound DirectX MS-DOS Win32 Windows ,,Windows NT, • Microsoft
Corporation,• ,,,,‘ , ,,, •,,,

‘6	
DirectInput	
μDirectX 3,Š,, ,, • §	
DirectInputŠ— §	
DirectInput,,,ffffff,ff f §	
DirectInput,,,f f f,,,fff,ff f §	
DirectInputffffff §	
fff §	
Š §	
f ffffŠ §	
IClassFactory.....	§
IDirectInput	§
IDirectInputDevice.....	§
“.....	§
ff ff•	§
,	§
f f f ffff’	§
fff ffff’	§
DirectInputffff fff< f f.....	§
DirectInputf f ff fff fff.....	§
•’.....	§

DirectX 3, §, , , • XE "DirectX 3, §, , , •" §
 (E ,Microsoft® DirectX™, ffffffff,,,,,fff,,f f f"—
 ffff,ff f,,, fff,,f f f f,‘,,API, COM,ffffff,,fff ff f,—,,
 DirectX, fff,,f f f f f f,‘,,• Win32 API,,, ‘,ffff,,
 Windows 95,‘, ,,,DirectInput
 1.0,,, ffff ffffffff ffff,,ffff,%o”,,, ,joyGetPosEx,,,,,,API f
 ffff fff—
 fffff(joy.cpl) ,,,ffff ffffffff ffff,ff f,%o”,,VJOYD.VXD,Š—
 ,,,ffff fff,, ,,,,
 f□fff 1.0,ff□f^—□DirectX SDK,Š,□,%o—
 ,—,‘,,,“,,,, fff,,f□f□f□f□f,ffff,,,,,Windows,,,,,□%o,,,,,, ,,—
 f□fff,DirectInput,—•,,,,,□DirectInput
 3.0,,,□fff,,f□f□f□f□f,‘,,,, ‘,ffff,%o”,,, DirectInput 1.0
 ,^,,□DirectInput 3.0 の API COM

DirectInput の COM
 ff□ffff,ff□f,,fffffffff□ffff,‘,,, DirectInput ,,,ff□f,•%o,,□”—
 ffff(“,DirectInputfff□fff f,,,,,ffff),ff□f,‘,‘,•%o,,
 DirectInputŠ— XE "DirectInputŠ—" §

Windows DirectX SDK

,,,□□“,ffffff□ff□f,□□,,,,,,,□DirectInput,□“—
 ffff□f□f,,□‘,•—,ffff,%o”,,,DirectX
 SDKfff□fff,,□

—□,—,□ff COM API COM DirectInput 1.0

DirectInput の XE "DirectInput の" §

XE " : " § XE " :
 ff " §DirectInput
 API,□ffff,,fff□fffffffff,‘, ‘, ,,,fff,‘<, ,API Microsoft
 Win32® (SDK),ffffff API の ffff□f の
 , DirectInputfff□ffff,□• ,OEM<<ffffffff,□,^ ‘ ,,,fff の
 •,ffff,Š”,,
 ffffffff,ff□f,,DirectInputŠ□,ff
 Win32fffffffffAPI,,,•□“,—, DirectInput API,□ffffffff□f,ffff

の DirectInput の

- の

- の
- の

の XE "の" §

DirectInput DirectInput

XE " :の " §DirectInput

のの

4 4 2

2 4 16 4 1

6 32

XE " : " §

ののの

XE " : " §

16 ののjoyGetNumDevs の 0

joyGetPosEx の JOYERR_NOERROR

JOYERR_UNPLUGGED

joyGetDevCaps

joyGetNumDevs の‘,,,ffffffff ,,,, ffff,ff f,,ffffffff ,Ž,

ffffffff,ffffff fff,,fff XE "ffffffff,ffffff fff,,fff" §

“^,fff,, fffffffff,ffffff fff XE "ffffffff:ffffff fff"

§,,,fff,,,,, Microsoft Windows®

95,ffff ffff,,ffffffff ffff fff,, f f, ,,ffff fff,,^%

,ffffffff,“,,,

- ^“,ffffffff XE "ffffffff:“^“,ffffffff"•EM XE "
- :OEM " §
- XE " :
- " §

の 6 32 ののののの

joyConfigChanged のの

の XE "の" §

joyGetPosEx の x y z Win32 API の 4 joyGetPosEx の

• 4 5 6 の r u v の

• XE " : " §

• XE " : " §

•••

• の の XE " :
" §

DirectInput の XE "DirectInput の" §

の DirectInput の

• DirectInput

• DirectInputDevice

• の

•

•

•

DirectInput XE "DirectInput " §

DirectInput DirectInput
DirectInputCreate API DirectInput
DirectInputCreate API IDirectInput

IDirectInput の DirectInput
IDirectInput::EnumDevices

DirectInputDevice XE "DirectInputDevice " §

—**ffff**, **DirectInputDevice** の

□ •,

□ **ffff** □ **fff**, **IDirectInput::CreateDeviceffff**, **œ**

, □ ,, **DirectInputDeviceffff**, □ □ ,, □

IDirectInput::CreateDeviceffff, □ **IDirectInput**
Deviceffff □ **ff** □ □ , •, □

IDirectInput::CreateDevice, ^, **fff** □ **f**, **fffff** **G**,, GUID
GUID_SysMouse GUID_SysKeyboard の " §

XE " : " § **c_dfDIMouse**
IDirectInputDevice::SetDataFormat

```

    XE " : " §
    IDirectInputDevice::GetDeviceState
    IDirectInputDevice::GetDeviceData 0(0 IDirectInputDevice::Unacquire
    ) ( )

```

XE "—ffff §fff,,ffff f f,Ž“,CEOBJECTDATA の
 の IDirectInputDevice::GetDeviceData

```
DIMOFS_BUTTON3 DIMOFS_X DIMOFS_Y ,,DIMOFS_Z,, ,,,',
, DIMOUSESTATE “,,',ffff,Ž ,,,,’,—
,, DIDEVICEOBJECTDATA “,f f,fff ,,,fffff,Ž ,,,, Š,’,,,,,
,,fffff,‘,,Žf f, “,dwDataff ff,‘ , , fff ffffff,,,,dwData,%o^f
ff,,—^,,, ,fff, ^fff, ,fff,%o,,,fff,, —,,,fff,,
```

Scrawlffff ffff ffff, ffff f f,Ž ,IDirectInputDevice::GetDeviceData,,
Ž“, •, —,•,,,,,Ž,,,,,i
f f,IDirectInputDevice::GetDeviceData,,Ž“,,,,, , Ž •,,fff f f,—
‰”,,,
ffff fff, ffff,”Ž,,f f,Ž“,,’ fff ffff,‘fff XE"‘fff:fff ffff,"
§,fff,,,,,,,,,i,,,, •—,‘fff,Ž,fff,fff,,Iows 95Š‰%, fff ffff,‘,,‰%

XE " の : " §
 irectInputDevice::GetDeviceData
 DirectInput
IDirectInputDevice::GetDeviceData
 の DIDEVICEOBJECTDATA の
 の
 ÒA” の ÒR”
 IDirectInputDevice::GetDeviceData DIDEVICEOBJECTDATA の 3
 の“A” ÒA”
 ÒR” IDirectInputDevice::GetDeviceData のの
 DIK_* DIDEVICEOBJECTDATA の dwOfs の
 () DIK_*
 DIDEVICEOBJECTDATA のののの dwData のの dwData の
 IDirectInputDevice::GetDeviceData のの
 XE " : " § ; の
IDirectInputDevice::SetCooperativeLevel Windows 95Š‰%
 ,, f□f□f□ffff,‘,,‰%,<’fff,ff□f,,,,,□ DISCL_BACKGROUND |
 DISCL_NONEXCLUSIVE DISCL_FOREGROUND |
 DISCL_NONEXCLUSIVE DirectInput 3.0
DISCL_BACKGROUND | DISCL_EXCLUSIVE DISCL_FOREGROUND
 | **DISCL_EXCLUSIVE** f,f□f□f□ffff,‘,,ff□f,,,,,
 f□f□f□f,, DirectInput Windowsf ,^
 Š<’fff –,Š,,□のの
 DirectX,□—f□fff,,□Windows NT DirectInput,ff□f,,,,,,□,, Windows
 NT f□f□ffff,‘,,DISCL_FOREGROUND | DISCL_NONEXCLUSIVE
 Windows NT
 のの <’ の
Windows 95 の DirectInput DISCL_FOREGROUND |
DISCL_NONEXCLUSIVE の
 XE "の: f□f□ "
 §IDirectInputDevice::GetDeviceData,,IDirectInputDevice::GetDeviceState
IDirectInputDevice::Acquire
 IDirectInputDevice::Unacquire
IDirectInputDevice::Acquire
 DISCL_FOREGROUND
 IDirectInputDevice::GetDeviceData IDirectInputDevice::GetDeviceState
 の DIERR_INPUTLOST
 DirectInput
 DIERR_INPUTLOST
 DirectInput

XE " " §
 XE " : " §
 の §,“,,, ',,,,,ffff□fff,□□, P US PC
 の PC XE " :Windows " §Windows
 (DIK_LWIN DIK_RWIN DIK_APPSの
 XE " : " §DIK_PAUSE
 PC DIK_PAUSE のDIK_LCONTROL
 DIK_NUMLOCK の òPause”
 XE " :の" § のの XE " :
 の" §

の XE " :PC XT " §PC XT PC
 ATK_F11 DIK_F12 の(DIK_* 0x80)の PC XT
 DIK_SYSRQ
 NEC PC-98 US

XE " US
 NEC PC-98 XE " :NEC PC-98 " §
 DIK_CAPSLOCK DIK_KANA ,,

DOS/V 106 ,
 NEC PC-90NVERT
 DIK_KANA
 DIK_KANJI
 DIK_NOCONVERT
 DIK_YEN
 DIK_APOSTROPHE
 DIK_EQUALS
 DIK_GRAVE

NEC PC-98

NEC PC-98
DIK_AT
DIK_CIRCUMFLEX
DIK_COLON
DIK_F13, F14, F15
DIK_KANA
DIK_KANJI
DIK_NOCONVERT
DIK_NUMPADCOMMA
DIK_NUMPADEQUALS
DIK_STOP
DIK_UNDERLS
DIK_GRAVE
DIK_NUMLOCK
DIK_NUMPADENTAX
DIK_AX
DIK_CONVERT
DIK_KANJI
DIK_NOCONVERT
DIK_YEN
DIK_RCONTROL
DIK_RMENU

J-3100
DIK_KANA
DIK_KANJI
DIK_NOLABEL
DIK_YEN
DIK_RCONTROL
DIK_RMENU

DirectInput XE "DirectInput" §

DIDFT_GETINSTANCE XE "DIDFT_GETINSTANCE" §
BYTE DIDFT_GETINSTANCE(
 DWORD dwType)

DirectInputf□f□ff□fff□ §
BYTE DIDFT_GETTYPE(
 DWORD dwType)

DirectInput

dwType
DirectInput

DISEQUENCE_COMPARE XE "DISEQUENCE_COMPARE" §
BOOL DISEQUENCE_COMPARE(
 DWORD dwSequence1, cmp,
 DWORD dwSequence2)

の DirectInput =" "<" ">" "<=" ">="

dwSequence2

の

の0

の dwSequence1 dwSequence2

if (DISEQUENCE_COMPARE(dwSequence1, <, dwSequence2)) {

```
...
}

の dwSequence1  dwSe
```

<pre>quence2 if (DISEQUENCE_COMPARE(dw Sequence1, >=, dwSequence2)) { ... } GET_DIDEVICE_SUBTYPE XE "GET_DIDEVICE_SUBTYPE" § BYTE GET_DIDEVICE_SUBTYPE(DWORD dwDevType) ののの DirectInput dwDevType DirectInput GET_DIDEVICE_TYPE XE "GET_DIDEVICE_TYPE" § BYTE GET_DIDEVICE_TYPE(DWORD dwDevType) DirectInput dwDevType</pre>		
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

<div>DirectInput</div> <div>MAKEDIPROP XE "MAKEDIPROP" §</div> <div>Microsoft</div> <div>GUID</div> <div>DirectInputCreate XE "DirectInputCreate" § HRESULT DirectInputCreate(HINSTANCE hinst, DWORD dwVersion, LPDIRECTINPUT * lplpDirectInput, LPUNKNOWN punkOuter);</div> <div>IDirectInput COM DirectInput</div> <div>*lplpDirectInput</div> <div>⓪ punkOuter = NULL</div> <div>CoCreateInstance(&CLSID_DirectInput, punkOuter, CLSCTX_INPROC_SERVER, &IID_IDirectInput, lplpDirectInput) Initialize</div> <div>⓪ punkOuter != NULL</div>		
----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--	--

CoCreateInstance(&CLSID_DirectInput, punkOuter, CLSCTX_INPROC_SERVER, &IID_IUnknown, lpDirectInput)		
の AN		

Sif□fff,,UNICODEf□fff,•'□,,ANSIf□fff,IDirectInputWfff□fff,ff□f,,ffffff,□□,,
 □reate,"□,•Žfff,ffffff,,□
 hinst
 DirectInputffffff,□□,,,ffff□fff,,DLL,ffffff ffff□
 dwVersion

dinput.h の DIRECTINPUT_VERSION

DirectInput の の DirectInput
 DLL の

lpDirectInput
 IDirectInputfff ffff,,ffff,,•,,Ž,Ž,
 punkOuter
 OLE□□%<"-, Ē,,ffff□fff□fff,□□%
 ,,irectInput,•,,ffffff,IUnknown,,ffff,,□
 OLE□□',—<,,IDirectInput,,ffff,,,,

COM の

DI_OK = S_OK:

DIERR_INVALIDPARAM = E_INVALIDARG: lpDirectInput

DIERR_OUTOFMEMORY = E_OUTOFMEMORY: fff,•',,,

DIERR_DIERR_OLDDIRECTINPUTVERSION:

DirectInput,f□ffctInputf□f□fff',,ffff□fff,□,,,□

joyConfigChanged XE "joyConfigChanged" §

MMRESULT joyConfigChanged(
 DWORD dwFlags);

ffff,□,,ffffffff,□'

OEM,ffffffffff□ffffff□ffff□ffff□fff,,,□ffff□fff,fffffffff,fff□f
 ff,ffffff,.,.,
 JOYSTICK USER',•
 HKEY_LOCAL_MACHINE
 dwFlags
 0

JOYERR_NOERROR の
JOYERR_NOCANDO

ffff,ž,č",fff
o

```

the JOYSTICKID1 □□□ JOYSTICKID2,,,□
pjc
ffffffffff<“,Š,JOYCAPS□“,ffff□
cbjc
JOYCAPS□“,fff.fff□,Ž,□
’
□€,,, JOYERR_NOERROR □Ž”,,,Ž,ff□,•,□

```


MMSYSERR_INVALIDPARAM

MMSYSERR_NODRIVER

```
JOYERR_REGISTvs"$
joyGetNumDevs XE "joyGetNumDevs" $
UINT joyGetNumDevs(VOID);
```

```
ffffffff□ffff,ff□f,,,ffffff
```

```
joyGetPosEx
```

0

```
joyGetDevCaps
```

```
xe "joyGetPosEx"$
```

```
joyGetPosEx XE "joyGetPosEx" $
```

```
MMRESULT joyGetPosEx(UINT uJoyID, LPJOYINFOEX pji);
```

```
^',,,,fff ' ,ffffffff,Ž“,,
```

```
„Š □ □ Ą,□,‘,□JOYINFOEX□“ ,dwFlags—‘ ,,,,^fff□Ž“fff□‘□,fff,Ž,ffff,„,Š'ffff,„□‘6Ž,—
```

```
„□•Ą„ffff,%“„,□
```

```
uJoyID
```

```
□%„,fffffffff,Ž•Ž□
```

```
pji
```

```
の JOYINFOEX の
```

```
JOYERR_NOERROR の
```

```
JOYERR_UNPLUGGED
```

```
MMSYSERR_BADDEVICEID
```

MMSYSERR_INVALIDPARAM

MMSYSERR_NODRIVER

Ž JOYINFOEX

f fffŠ

DIEnumDeviceObjectsProc XE ACK DIEnumDeviceObjectsProc(
LPCDIDeviceObjectInstance lpddoi,
LPVOID pvRef);

IDirectInputDevice::EnumObjects

lpddoi

□□□□□□□□□□□□□□□□

DIDeviceObjectInstance□“,fff□

pvRef

IDirectInput“,□
•’

DIENUM_CONTINUE —<Æ’,
DIENUM_STOP —<Ž,,

DIEnumDevicesProc XE "DIEnumDevicesProc" S
BOOL CALLBACK DIEnumDevicesProc(
LPDIDeviceInstance lpddi,
LPVOID pvRef
);

IDirectInput::EnumDevicesfff,—,,,f□ffffŠ□□
lpddi

—<,,,,,fff,<’,,DIDeviceInstance の

pvRef

IDirectInput::EnumDevices

∞

DIENUM_CONTINUE

DIENUM_STOP

IClassFactory XE "IClassFactory" §

IClassFactory	OLE	
OLE	DirectInput	の
OLE		

IClassFactory::AddRef XE "IClassFactory\:\.AddRef" §

HRESULT AddRef(
 LPCLASSFACTORY *lpClassFactory*);

IUnknown::AddRef

OLE の

IClassFactory::CreateInstance XE "IClassFactory\:\.CreateInstance" §

HRESULT CreateInstance(
 LPCLASSFACTORY *lpClassFactory*,
 LPUNKNOWN *punkOuter*,
 REFIID *riid*,
 LPVOID * **ppvOut**);

DirectInput	OLE の
IClassFactory::CreateInstance	

punkOuter

OLE	の
000	

riid

ppvOut

の

COM の

S_OK:

E_INVALIDARG:ppvOut

CLASS_E_NOAGGREGATION:

E_OUTOFMEMORY:

E_NOINTERFACE:

ClassFactory::LockServer XE "ClassFactory\:\LockServer" §

HRESULT LockServer(

LPCLASSFACTORY lpClassFactory,

BOOL fLock);

DLL

0

OLE

DLL

の

ClassFactory::LockServer

fLock

TRUE fff ffff,ffffff,,

FALSE,,, fff ffff,ffffff,,

•,

COMff f f,•, ^%,ff f f,•,,,f f,,,,,E,,,

S_OK:

E_OUTOFMEMORY: fff,•,,,,

ClassFactory::QueryInterface XE "ClassFactory\:\QueryInterface" §

HRESULT QueryInterface(

LPCLASSFACTORY lpClassFactory,

REFIID riid,

LPVOID *ppvObj);

ffffff ,',fff ffff,ffffff ffff,—

,, OLEfffff,IUnknown::QueryInterface,Ž ,,,

riid

•—, IID

ppvObj

.,ffff,Ž,Ž,

•,

COMff f f,•,

IClassFactory::Release XE "IClassFactory\:\:Release" §
HRESULT Release(*LPCLASSFACTORY* lpClassFactory);

fff ffff,.,Ž ffff,ffffff,, fffffff,Ž ffff,0,,, ,ffffff,fff,,
 %o•,,, OLEfffff,IUnknown::Release,Ž ,,,,
 •,
 fffffffŽ ffff,•,

IDirectInput XE "IDirectInput" §
DirectInputffffff,DirectInput**ffffff,•, ’“—**
ffff,•,DirectInputDeviceffffff, ,,

IDirectInput::AddRef XE "IDirectInput\:\:AddRef" §

DWORD AddRef(

LPDIRECTINPUT lpDirectInput);

IDirectInput **IUnknown** のの COM

AddRef

1
Release
 0

の
 lpDirectInput
 の DirectInput

DWORD

IDirectInput::CreateDevice XE "IDirectInput\:\:CreateDevice" §
HRESULT CreateDevice(

LPDIRECTINPUT

lpDirectInput,

REFGUID rguid,
LPDIRECTINPUTDEVICE

***lplpDirectInputDevice,**

LPUNKNOWN * pUnkOuter);

GUID

の **punkOuter** = NULL

CoCreateInstance(&CLSID_DirectInputDevice, NULL,
 CLSCTX_INPROC_SERVER, riid, lplpDirectInputDevice),%
 ,,ffffff,□□,Init,,□ CoCreateInstance(&CLSID_DirectInputDevice,
 punkOuter, CLSCTX_INPROC_SERVER, &IID_IDirectInput, lplpDirectInput)

lpDirectInput

の DirectInput

rguid

GUID の GUID EnumDevices
GUID の

lpDirectInputDevice

IDirectInputDevice

punkOuter

OLE の
0 の

0

COM の

DI_OK = S_OK:

DIERR_INVALIDPARAM = E_INVALIDARG: ppvOut

DIERR_OUTOFMEMORY = E_OUTOFMEMORY:

DIERR_NOINTERFACE = E_NOINTERFACE: の

DIERR_DEVICENOTREG: DirectInput

IDirectInput::EnumDevices XE "IDirectInput\:\EnumDevices" §

HRESULT EnumDevices(

LPDIRECTINPUT lpDirectInput,

DWORD dwDevType,

LPDIENUMCALLBACK

lpCallback,

LPVOID pvRef,
DWORD dwFlags);

‘,

”

—<TACHEDONLY,fff,,,,,□ffff,,,,□

```

ffff fff,'<,,f ffffŠ , lpCallbackfff f,IDirectInput::EnumDevices,“,
,„Š ,
—<,,,ffff,,,‘,,E, ,, f ffff,, ffff fff,,,ffff f ,, •GUID
∞∞∞ DirectInput
,,ff f,DirectInput,, fff,,f f f ffff,,—<,,,
lpDirectInput
,,fff,E,□,,,DirectInputffffff,
dwDevType

```

```
ffff□fff□ffff□0,,□□,□,,,,fff□fff,—⟨,,□,,,,□□, DIDEVTY
PE *□□□□□□□□
,,fff□fff,Ž,□,,ff□f, DirectInput,,fff□ffff,,f□f□f,,—⟨,,□
lpCallback
ŠDirectInputffff,⟨□,,,Œ,□,,,□ffff□fff'⟨f□ffffŠ□,Ž,Ž,□
```

**BO
OL CALLBACK
DIEnumDevicesProc(
LPDIDeviceInstance lpddi,
LPVOID pvRef)**

```

lpddi
„ffff□ffffff,□„,□“Ž,Ž,□
pvRef
EnumDevices,—,,,□ffff□f

```

 ff

pvRef

[illegible]

DIEDFL_ATTACHEDONLff□f□f,•,f□f,,,,,€,,,□
DI_OK = S_OK: ‘ , —,, f□fffff,“

DIERR INVALIDPARAM=E INVALIDARG: fl

IDirectInput::GetDeviceStatus XE "IDirectInput\.\:GetDeviceStatus" §

HRESULT GetDeviceStatus(

```
LPDIRECTINPUT lpDirectInput,  
REFGUID rguidInstance);
```

⑦ DirectInput,□□□□□□□□□□□□□□□□□□□□□□□□□□□□ OK □□
 ,,,,,DirectInputf f f f f,ž f f,ž *,□
 •'


```

IDirectInput,CE ,IUnknownfff ffff,^• ffff fff,Ž—
,,,,,,fff ffff,,,ffffff,ff f,,,,,,—, ,,,,—, —
, ,,“ ,COMfff ffff,ffffff,ff f,,, ffff fff,,,fff ffff,’,Ž
—,Ž, ppvObj の
Release ,fff ffff,%o•,,,,,, ,fff,,, Š‘,,, —
,“, “ Š ,,,,DirectInputffffff,Š’,,, ,,,, OLEffff,IUnknown:
:QueryInterface,Ž ,,,,

```

lpDirectInput

```

,fff,CE, ,,,DirectInputffffff,Ž,Ž,
riid
—,fff ffff,Ž•,fff ffffŽ•Ž(IID),Ž,Ž,
ppvObj
-,□,,,□□,Ž□,,,□□•,,,fff□ff

```

DI_OK

DIERR_INVALIDPARAM

DIERR_NOINTERFACE

IDirectInput::Release XE "IDirectInput\: \:Release" §

```

DtInput,CE□,,IUnknownfff□ffff,^□Š~,,,COMffffff,Ž□ffff,CE,,,,,—
,,,□ffffff,□□,□□,,,□Ž□ffff,1,fff,,,□AddRef,CE,□,,,,,Ž□ffff,ffff
fff,,,□Release,CE,□,,,,,Ž□ffff,ff
の OLE の

```

lpDirectInput

```

,fff,CE, ,,,DirectInputffffff,Ž,Ž,
•’
,,Ž ffff,Š,DWORD ,,•,fff—“,Ž—,,,,,^

```

IDirectInput::RunControlPanel XE "IDirectInput\:\:RunControlPanel" §
 HRESULT RunControlPanel(

LPDIRECTINPUT

lpDirectInput,

HWND hwndOwner,
DWORD dwFlags);

Windows の DirectInputfff□ffff,Ž□,,,,,—,,,□f□,□ **DirectInput**

hwndOwner

の UI
NULL

dwFlags

の 0

COM のの

DI_OK = S_OK:

IDirectInputDevice XE "IDirectInputDevice" §

IDirectInputDevice::Acquire XE "IDirectInputDevice\:\:Acquire" §
HRESULT Acquire(
LPDIRECTINPUTDEVICE lpDirectInputDevice);

のGetDeviceState GetDeviceData

ffff,
lpDirectInputDevice

の *fff,œ,□,,,DirectInputffff□ffffff,ž,ž,□*
•'

COM*ff□f□f,•,□%o,ff□f□f,,,,^,,,□*
DIERR_INPUTLOST: *ffff,,ffff,—,,,,,□*
DIERR_INVALIDPARAM = E_INVALIDARG: *ffff,“,,f□f□ff□fffœž,,,□*

IDirectInputDevice::AddRef XE "IDirectInputDevice\:\:AddRef" §
DWORD AddRef(LPDIRECTINPUTDEVICE lpDirectInputDevice);

IDirectInputDevice IUnknown COM

I AddRef
Release
lpDirectInputDevice
,,fff,œ, ,,,DirectInputffff fffffff,ž,ž,
•'
„ž ffff,š,DWORD

IDirectInputDevice::GetCapabilities XE "IDirectInputDevice\:
\:GetCapabilities" §
HRESULT GetCapabilities(

```
LPDIRECTINPUTDEVICE lpDirectInputDevice,  
    LPDIDEVCAPS lpDIDevCaps);
```

```

“—ffff, •,Ž“,
lpDirectInputDevice
,fff,Ⓔ,□,,DirectInputffff□ffffff,Ž,Ž,
lpDIDevCaps
Ⓞ DIDEVCAPS ⓄⓄ
dwSize
DIDEVCAPS

```

COM 99

DIERR_INVALIDPARAM = E_INVALIDARG: lpDIDevCaps

```

IDirectInputDevice::GetDeviceData XE "IDirectInputDevice\
\GetDeviceData" $
HRESULT GetDeviceData(
    LPDIRECTINPUTDEVICE
    lpDirectInputDevice,
    DWORD cbObjectData,

```

```
LPDIDeviceObjectData rgdod,  
LPDWord pdwInOut,
```

```

DWORD <'fff,□SetDataFormat,f□f□fff□fff□fff□fff□Acquire,ffff,"Ž,,,,,,,,,□
lpDirectInputDevice
,,fff,€,□,,,DirectInputffff□fffff,Ž,Ž,□
cbObjectData

```

DIDeviceObjectData の

rgdod

⑦ DIDEVICEOBJECTDATA の*pdwInOut

 $\mathcal{O}(\text{NULL})$ \mathcal{O}

pdwInOut

[illegible]

```
DI_BUFFEROVERFLOW = S_FALSE:
f□f,□□,Ž“,,,□ffff,ffff□fff,□•,,,,,f□f,,,,Ž,,,□,,□□□ffff□f□f,Ž“%□□‘,,,a,□□,Œ,□,,,,•,,,
□,,f□f,□□f□f□f□f□f□f,,,,,’^□
```

```
DIERR_NOTACQUIRED: ffff,Ž,,,,□
```

```
DIERR_INPUTLOST: ffff,,ffff,“,,,□ffff□fff,ffff,□”Ž,,,,,□
```

```
DIERR_INVALIDPARAM = E_INVALIDARG: -Œ,ff
```

の10

```
DIDeviceObjectData rgdod[10];
```

```
DWORD dwItems = 10;
```

```
hres = IDirectInputDevice_GetDeviceData(
    pdid,
```

```
    sizeof(DIDeviceObjectData),
```

```
    rgdod,
```

```
    &dwItems,
```

```
    0);
```

```
if (SUCCEEDED(hres)) {
```

```
    // ffff,□□,fffff,
```

```
    // dwItems =
```

```
    if (hres == DI_BUFFEROVERFLOW) {
```

```
        // ffff,f□-□,•,,□
```

```
        dwItems = INFINITE;
```

```
        hres = IDirectInputDevice_GetDeviceData(
```

```
            pdid,
```

```
            sizeof(DIDeviceObjectData),
```

```
            NULL,
```

```
            &dwItems,
```

```
            0);
```

```
if (SUCCEEDED(hres)) {
```

```
    //
```

```
    // dwItems =
```

```
    if (hres == DI_BUFFEROVERFLOW) {
```

```
        //
```

```

    }
}

```

```

rgdod NULL
ff,,f

```

```

dwItems = INFIata(
    pdid,
    sizeof(DIDEVICEOBJECTDATA),
    NULL,
    &dwItems,
    DIGDD_PEEK);
if (SUCCEEDED(hres)) {
    // dwItems =
    if (hres == DI_BUFFEROVERFLOW) {
        // ffff,f ff,
        // ,,f,ff,,-,""
        dwItems = 0;
        hres = IDirectInputDevice_GetDeviceData(
            pdid,
            sizeof(DIDEVICEOBJECTDATA),
            NULL,
            &dwItems,
            0);
    }
    if (hres == DI_BUFFEROVERFLOW) {
        // 
    }
}

```

```

IDirectInputDevice::GetDeviceInfo XE "IDirectInputDevice\:\:GetDeviceInfo" §
HRESULT GetDeviceInfo(
    LPDIRECTINPUTDEVICE lpDirectInputDevice,

```

①

④ DirectInput

①①

•

DI OK = S OK:

,,,^□,fff□f,-œ,,,,□

DirectInputffff,Žf□f,Ž“,□

```
fffff□f□fZ',,□SetCooperativeLevel,<'fff,□SetDataFormat,f□f□ff□ffff,,,,fff,□Acquir
```

e,ffff,"Ž,,,,,,,,,□

IpDirectInputDevice

„fff,Œ,□,,,DirectInputffff□fffff,Ž,Ž,□

lpvData $\mathcal{O}\mathcal{O}$

lpvData

①①

SetDataFormat

 \mathcal{O} COM のの

DI_OK = S_OK:

E PENDING: $\mathcal{O}(\text{USB})$

 \mathcal{O}

00

```
GetDeviceState    E_PENDING
```

DIERR_NOTACQUIRED:

DIERR_INPUTLOST: *の*

DIERR_INVALIDPARAM = E_INVALIDARG: lpvData の SetDataFormat
cbData

IDirectInputDevice::GetObjectInfo XE "IDirectInputDevice\:\:GetObjectInfo" §
HRESULT GetObjectInfo(

LPDIRECTINPUTDEVICE lpDirectInputDevice,
LPDIDEVICEOBJECTINSTANCE pdidoi,
DWORD dwObj,
DWORD dwHow)

lpDirectInputDevice
,,fff,Ⓔ, ,,DirectInputffff fffffff,Ž,Ž,
pdidoi
ffffff,Š,, •,Ž,Ž, ,
Ⓔ, ‘,DIDEVICEOBJECTINSTANCE “,dwSizeff ff, Š‰,,,,,,,,,
dwObj
ffffff,ffff,,ffffff,Ž•,,
dwHow
dwObj,‰Ž•–,Ž•,,

Ž DIPROPHEADER
•’

COMff□f□f,•,□^‰,ff□f□f,—□“,,,^□
DI_OK = S_OK: , —,,
DIERR_INVALIDPARAM = E_INVALIDARG: –Ⓔ,fff f,‘ ,,
DIERR_OBJECTNOTFOUND: Ž’,,ffffff,‘ ,,,

IDirectInputDevice::GetProperty XE "IDirectInputDevice\:\:GetProperty" §
HRESULT GetProperty(

```

LPDIRECTINPUTDEVICE lpDirectInputDevice,
                                REFGUID rguidProp,
                                LPDIPROPHEADER pdiph);

IDirectInputDevice::SetProperty
IDirectInputDevice::SetProperty
lpDirectInputDevice
    の DirectInput
    rguidProp
                                DIPROP_*の GUID の
(C++ )
    の
DIPROP_AXISMODE
DIPROP_BUFFERSIZE
DIPROP_GRANULARITY
DIPROP_RANGE
ののののの
pdiph
                                DIPROPHEADER

COM のの
DI_OK = S_OK:
DIERR_INVALIDPARAM = E_INVALIDARG: pdiphfff□f,—□E,ffff,,,,□dwHowfff□ff,—
□E,□,,,,,fffff;□,,,□
DIERR_UNSUPPORTED = E_NOTIMPL:ffff,,,fffff,fffff,ff□f,,,,□
—
^%,C□E□f□f,^,□DIPROP_BUFFERSIZEfffff,'Ž“—,Ž,,,,□

DIPROPDWORD dipdw;
HRESULT hres;
dipdw.diph.dwSize = sizeof(DIPROPDWORD);
dipdw.diph.dwHeaderSize = sizeof(DIPROPHEADER);
dipdw.diph.dwObj = 0; // の
dipdw.diph.dwHow = DIPH_DEVICE;

hres = IDirectInputDevice_GetProperty(pdid, DIPROP_BUFFERSIZE,
&dipdw.diph);

if (SUCCEEDED(hres)) {
    // dipdw.dwData
}

IDirectInputDevice::EnumObjects XE "IDirectInputDevice\\.\EnumObjects" §
HRESULT EnumObjects(
    LPDIRECTINPUTDEVICE lpDirectInputDevice,
    LPDIENUMDEVICEOBJECTSCALLBACK lpCallback,

```



```
LPVOID pvRef,
DWORD fl);
```

```
( )
```

```
lpDirectInputDevice
```

```
の DirectInput
```

```
lpCallback
```

```
DirectInputDevice
```

```
BOOL CALLBACK DIEnumDeviceObjectsProc(
LPCDIDEVICEOBJECTINSTANCE lpddoi,
LPVOID pvRef);
```

```
pvRef
```

```
( )
```

```
fl
```

```
DirectInput
```

```
COM の
```

```
DI_OK = S_OK:
```

```
DIERR_INVALIDPARAM = E_INVALIDARG: fl
```

```
IDirectInputDevice::Initialize XE "IDirectInputDevice\:\:\Initialize" §
```

```
HRESULT Initialize(
```

```
LPDIRECTINPUTDEVICE lpDirectInputDevice,
```

```
HINSTANCE hinst,
```

```
DWORD dwVersion,
```

```
REFGUID rguid);
```

DirectInputDevice

\mathcal{O}
 ,,, “ Š%o,•—,,,,,^,,,,, fffffff, ‘,•’,,,,Œ,,,, ‘,,,,ffffff,Ž—
 IDirectInput::CreateDeviceffff,ffff, ,,,Ž““,,, Š%o
 ,,, ’ ,ffff fff,,,Š ,Œ, ,•—,,,

lpDirectInputDevice

,,fff,Œ, ,,,DirectInputffff fffffff,Ž,Ž,
hinst
DirectInputDeviceffffff, ,,,,ffff fff,,,DLL,ffffff ffff DirectIn
put, ,,’—,ffff fff,,,DLL,” ,,,,,,,,’,
dwVersion
Ž—,,dinput.h $\mathcal{O}\mathcal{O}$
,DIRECTINPUT_VERSION,,,,,,,
DirectInput,,,’—
,, ,f fff,DirectInput,‘,ffff fff,,,DLL, Œ,,,,,Š’,,
rguid

fff□ffff,Š~,,,ffff,ffffff,Ž•,,□IDirectInput::Enum
Devicesffff,—
,,□ffff,,,ffffffGUID,ff□f,,,,,’,,,,,□
•,
COMff□f□f,•,□^%o,ff□f□f,•,,,f□f,,,,,Œ,,,□
DI_OK = S_OK: fff, ‘,,,,,,

DIERR_DIERR_OLDDIRECTINPUTVERSION: DirectInput \mathcal{O}

DIERR_DIERR_BETADIRECTINPUTVERSION:

ff□fŠ,,,DirectInputf□f□f□fff,‘,ffff□f
S_FALSE: ffff,rguid,“,,,ffffffGUID,,, Š%o,,,,,
DIERR_ACQUIRED: ffff,”Ž,,,,, Š%o,,,,,

IDirectInputDevice::QueryInterface XE "IDirectInputDevice\
\:QueryInterface" §

HRESULT QueryInterface(

 LPDIRECTINPUTDEVICE lpDirectInputDevice,

 REFIID riid,

 LPVOID FAR* ppvObj);

IDirectInputDevice

IUnknown

COM

ppvObj

Release

IpDirectInputDevice

```

,,,fff,Æ,bj
-,□,,,□□,Ž□,,,□□•,,,fff□ffff□ffff,“,,,~,Ž,Ž,□
•'

DI_OK
DIERR_INVALIDPARAM
DIERR_NOINTERFACE

IDirectInputDevice::Release XE "IDirectInputDevice:\.Release" §
DWORD Release(LPDIRECTINPUTDEVICE lpDirectInputDevice);

```

IDirectInputDevice IUnknown の COM

1

```

AddRef
,, Release,E, ,,,,Ž ffff,ffffff,,, Ž ffff,0,,,, ,,ffffff,%o*,,,
,,,, OLE   OIUnknown::QueryInterface,Ž ,,,,
lpDirectInputDevice
,,fff,E,□,,,DirectInputffff□ff

```

DWORDO

```

IDirectInputDevice::RunControlPanel XE "IDirectInputDevice\:\
\RunControlPanel" $
HRESULT RunControlPanel(
LPDIRECTINPUTDEVICE
IpDirectInputDevice,
HWND hwndOwner,
DWORD dwFlags);

```

 \mathcal{O}

```

IpDirectInputDevice
  の
  DirectInput
  の hwndOwner
  の UI □□□□□fffff, —, , fffff□ffff,ž•, □NULL, □fffff, '□, , , , ff□f□f, •, , f□f, , , , €, , □
  DI_OK = S_OK: '□, □□, □—, □

```

```

IDirectInputDevice::SetCooperativeLevel XE "IDirectInputDevice\:\SetCooperativeLevel" §
HRESULT SetCooperativeLevel(
LPDIRECTINPUTDEVICE lpDirectInputDevice,

HWND hwnd,

DWORD dwFlags);

```

のののののの

```

Acquire                                の,,,,,
hwnd
ffff,Š~,fffff
DISCL_FOREGROUNDfff,“,,,,,      ,,fff f,NULL^Š,,,,,,,,, ,,fffff,
ffffff,,,,,,,,, DirectInput      ,ffffff,,Š,ffffff,•,,ff ,,
dwFlags
ffff,Š~,<’fff,< ,,fff
DISCL_*fff,,,,,•“• %o,,,,,
•’
COMfff f f,•, ^%o,ff f f,•,,f f,,,,,Œ,,,
DI_OK = S_OK: ‘ , , —,,
DIERR_INVALIDPARAM = E_INVALIDARG: hwndfff f,—
Œ,ffffff ffff,,,,—Œ,fff,,fff,‘, ,,“,,,

```

```

IDirectInputDevice::SetDataFormat XE "IDirectInputDevice\:\SetDataFormat" §
HRESULT SetDataFormat(
LPDIRECTINPUTDEVICE
lpDirectInputDevice,
LPCDIDATAFORMAT lpdf);

```


Š,,, DirectInput,ffff’ffff,“,,,ResetEvent,Œ, ,,
,ffff fff,,ffff,Ž“ffff ffff,,, ,,

hEvent,NULL, ’’,•%”,,,
•’

COMff f f,• Ɔ ,,f f,,,,,Œ,,,
DI_OK = S_OK: ‘ , , —,,

DIERR_ACQUIRED:
IDirectInputDevicefffff,Ž,,,,□IDirectInputDevice::Unacquire
Ɔ

DIERR_HANDLEEXISTS:
IDirectInputDevicefffff,□ffff’ffff,,,‘□,,□
DirectInp

E_INVALIDARG: ffff□ffff,,,□
—
ffff,ŒŽ“,fff,,,,,□fffff,,,ffff,,□

dwResult = WaitForSingleObject(hEvent, 0);
if (dwResult == WAIT_OBJECT_0) {
 //
 //
 //
}

ƆƆ(Windows)

dwResult = WaitForSingleObject(hEvent, INFINITE);
if (dwResult == WAIT_OBJECT_0) {
 //
 //
 //
}

```
HANDLE ah[2] = { hEvent1, hEvent2 };
```

```
while (TRUE) {
```

```
    dwResult = MsgWaitForMultipleObjects(2, ah, FALSE,  
                                         INFINITE, QS_ALLINPUT);
```

```
    switch (dwResult) {
```

```
        case WAIT_OBJECT_0:
```

```
            // fff1
```

```
            //
```

```
            //
```

```
            ProcessInputEvent1();
```

```
            break;
```

```
        case WAIT_OBJECT_0 + 1:
```

```
            // ffff2,fff,,
```

```
            // ffff,Ž“ffffT_0 + 2:
```

```
            // Windowsfff□f,Ž,Ž,,
```

```
            // fff□f,,,,,,□—,□,
```

```
            while(PeekMessage(&msg, NULL, 0, 0, PM_REMOVE)){
```

```
                if (msg.message == WM_QUIT) {
```

```
                    goto exitapp;
```

```
                }
```

```
                TranslateMessage(&msg);
```

```
                DispatchMessage(&msg);
```

```
            }
```

```
            break;
```

```
        default:
```

```
            //
```

```
            Panic();
```

```
            break;
```

```
    }
```

```
}
```

```
HANDLE ah[2] = { hEvent1, hEvent2 };
```

```
DWORD dwWait = 0;
```

```
while (TRUE) {
```

```
    dwResult = MsgWaitForMultipleObjects(2, ah, FALSE,  
dwWait, QS_ALLINPUT);
```

```
    dwWait = 0;
```

```
    switch (dwResult) {
```

```
    case WAIT_OBJECT_0:
```

```
        // ffff1,fff,,
```

```
        // ffff,Ž“ffff,, ,,,
```

```
        // ,,,,ffff,,
```

```
        ProcessInputEvent1();
```

```
        break;
```

```
    case WAIT_OBJECT_0 + 1:
```

```
        // ffff2,fff,,
```

```
        // ffff,Ž“ffff,, ,,,
```

```
        // ,,,,ffff,,
```

```
        ProcessInputEvent2();
```

```
        break;
```

```
    case WAIT_OBJECT_0 + 2:
```

```
        // Windows
```

```
        //
```

```
    while(PeekMessage(&msg, NULL, 0,  
0, PM_REMOVE)){
```

```
        if (msg.message == WM_g);
```

```
        DispatchMessage(&msg);
```

```
    }
```



```

        break;

default:
    //
    // の

    // のの
    // “—,fff□f,−□œ,‘‘,,,žž,,
    if (!DoGame()) {
        "IDirectInputDevice\:\.SetProperty" §
        HRESULT SetProperty(
            LPDIRECTINPUTDEVICE lpDirectInputDevice,
            REFGUID rguid,
            LPCDIPROPHEADER pdiph);

        の
        のの

IDirectInputDevice::GetProperty

lpDirectInputDevice
    の DirectInput
    rguidProp
    の DIPROP_*の GUID の(C++の)
        の ‘,,,,,’<,,,,,
        DIPROP_AXISMODE
        DIPROP_BUFFERSIZE
        ,,,,ffff,,,,,□□,,,,,□□“,,,’ , ,œ□,f
        pdiph
        DIPROPHEADER

COM
    の
    の

DI_OK = S_OK:
DI_PROPNÖEFFECT = S_FALSE:
    の
DIERR_INVALIDPARAM = E_INVALIDARG: pdiph
        dwHow dwHow
DIPH_DEVICE dwObj 0
DIERR_OBJECTNOTFOUND:

```



```

„Ž□,Žf□fffff,fff,„„„□,fff,DIDF_ABSAXIS,tProperty,%
„Ž□,Žf□fffff,fff,„„„□,fff,DIDF_RELAXIS,‘□,„„„„□
dwDataSize
fff,•,fff□f□f,fff□,‘4,“□,„„„□ffffffffff,Ž‘„„,fffff,„„„dwOfs‘,‰„„„„□
dwNumObjs
rgodf’—,

rgodf

DIOBJECTDATAFORMAT ののの
ffff□f□f,„„„„„,’,„„„„,‘,„„„,^,□,’,„„„„,„„„,“^,□,2f□,’,„„„„, r
godf

の2 ‘•,’„„,„„„„,fff,—,„ffff fff,—
„„,f f ff fff,fff,„„

// ffff fff,‰, “,—,ffff f f,“, „‰’,

typedef struct MYDATA {
    LONG IX;                // X

                                LONG IY;                // Y
                                BYTE bButtonA;           //
                                BYTE bButtonB;           // の
                                BYTE bPadding[2];         //

    dword の
} MYDATA;

// の

DIOBJECTDATAFORMAT rgodf[] = {
    { &GUID_XAxis, FIELD_OFFSET(MYDATA,
IX), 0, DIDFT_AXIS |
DIDFT_ANYINSTANCE, },
    { &GUID_YAxis, FIELD_OFFSET(SET(MYDATA, bButtonA), 0, DIDFT_BUTTON |
DIDFT_ANYINSTANCE, },
    { &GUID_Button, FIELD_OFFSET(MYDATA, bButtonB), 0, DIDFT_BUTTON |
DIDFT_ANYINSTANCE, },
    };
#define numObjects (sizeof(rgodf) / sizeof(rgodf[0]))

DIDATAFORMAT df = {
    sizeof(DIDATAFORMAT), // „□“
    sizeof(DIOBJECTDATAFORMAT), //

    DIDF_ABSAXIS, //
    sizeof(MYDATA), //

```

```

    numObjects,          //
    rgodf,                //
};

```

DIDEVCAPS XE "DIDEVCAPS" §

```

typedef struct {
    DWORD dwSize;
    DWORD dwDevType;
    DWORD dwFlags;
    DWORD dwAxes;
    DWORD dwButtons;
    DWORD dwPOVs;
} DIDEVCAPS;

```

IDirectInputDevice::GetCapabilities 0
•,

dwSize

□“,fff,fff,Ž’,, IDirectInputDevice::GetCapabilities,☉,□,‘, Š%0

```

dwDevType
ffff fffŽ’Ž’,,ff ff,☉ ,,,, DirectInputffff fff☉ f f,Ž’ ,,,
dwFlags
ffff,Š’,,fff ^%0,fff,‘,,,,,
DIDC_ATTACHED: ffff,☉—“, ‘, ,,,,
DIDC_POLLEDDEVICE:
ffff,Š’,☉☉,,,f fff☉,,, f f,“,,,,,ffff fff,—
Ž’,GetDeviceState,☉, ,,,,,, fff
dwAxes
ffff ,—%0”,Ž’, ,Ž’,,
dwButtons
ffff ,—%0”,fff, ,Ž’,,
dwPOVs
ffff ,—%0”,Ž’ f, ,Ž’,, DirectInputf fff3.0,,—,,,

```

DIDeviceInstance XE "DIDeviceInstance" §

```

typedef struct {
    DWORD dwSize;

```

```

GUID guidInstance;
GUID guidProduct;
DWORD dwDevType;
TCHAR tszInstanceName[MAX_PATH];
TCHAR tszProductName[MAX_PATH];
} DIDEVICEINSTANCE;

```

```

IDirectInput::EnumDevices          IDirectInputDevice::GetDeviceInfo

```

```

dwSize
  の
guidInstance

```

```

          GUID                      GUID
          ,,          GUID の
GUID
guidProduct

```

```

dwDevType
  DirectInput
tszProductName[MAX_PATH]
  の          "Frobozz Industries SuperStick 5X"
tszInstanceName[MAX_PATH]
  の Joystick 1"

```

```

DIDEVICEOBJECTDATA XE "DIDEVICEOBJECTDATA" §
typedef struct {
  DWORD dwOfs;
  DWORD dwData;
  DWORD dwTimeStamp;
  DWORD dwSequence;
} DIDEVICEOBJECTDATA;

```

```

IDirectInputDevice::GetDeviceData の

```

```

dwOfs

```

```

IDirectInputDevice::GetDeviceState          dwData

```

```

          ,,          dwOfs の
          DIMOFS_* の
          DIK_*',,,,
ffff f ff fff,fff,,,,, fffff f ff fff,','fffff
dwData
ffff,"mf□f□,f□f,fff□fff,fff,fff,^,,,□,□□,□f□f,,,,,ÆŽ,",,,□
DIDFT_AXIS: fff,Žf□f,,,□□□Ž%o,",,,□ffff,□Žf□f,,,□□□□Ž□•,",,,□
DIDFT_BUTTON: dwData,%o^fff,,,^-,□ffff,%o,,,%o^fff,□^fff,fff,,□fff,—,,,fff,,,□
dwTimeStamp
ff•^,"ffff□ffff,□,,,ffff,□□,,,□□,fff,fff□ffff,□
GetTickCountffffŠ□,Æ□,,"□,',,,50",^□,,,^□
dwSequence

```



```

IX
fff,xŽ,Š,,□•□ffff,“Žf□f,,,□□□fff,xŽ•Ė,•^□ffff,□'Žf□f,,,□□□fff,xŽ•Ė,□'^□
IY
fff,yŽ,Š,,□•□ffff,“Žf□f,,,□□□fff,yŽ•Ė,•^□ffff,□'Žf□f,,,□□□fff,y•Ė,□'^□
IZ
fff,zŽ,Š,,□•□ffff,“Žf□f,,,□□□fff,zŽ•Ė,•^□ffff,□'Žf□f,,,□□□fff,zŽ•Ė,□'^□
fff,zŽ,,,,□,,',0,,,□
rgbButtons[4]
fff,□,'—□“ „fff,%o,,,,□□□□^fff,fff,,,□

```

DIOBJECTDATAFORMAT XE "DIOBJECTDATAFORMAT" §

```

typedef struct {
    const GUID * pguid;

```

DWORD dwOfs;

DWORD dwType;

DWORD dwFlags;

```

}
DIOBJECTDATAFORMAT□fff,•□, DIOBJECTDATAFORMAT□“,, ,
Šffffff(Ž□fff,,),‘,,,,, ““,,,, ,,, “”—
, IDirectInputDevice::SetDataFormat,“,,,DIDATAFORMAT□“Ž,□’ ,fff
□fff,DIOBJECTDATAFORMAT□“,”—, ,•—,,, , “‘,
c_dfDIMouse c_dfDIKeyboard の

```

pguid

```

Ž f f ff fff,—, , ,ff ff,NULL,fff,, ,,,,fff,ffffff,,,,,^—
,,
dwOfs
“—f f,‘,
f ffff“„ffffff, Ž,,,DWORDfff,f f,,,,4,” ,,,,,, ,,, fff,‘,,*,,,,
fff fffffff,’

```

dwType

```

ffffff,‘ ,,,,ffff□fffŽ’Ž□,,,□fffffffff(Ž□fff,,)
,‘ ,,,DIDFT_*fff,‘, ,,, ^fff,,ff DIDFT_ANYINSTANCE

```

dwFlags

の 0

の

DirectInput

4 DWORD

DIOBJECTDATAFORMAT dfAnyAxis = {

0, // fffff□f

4, // 0,,,,,,

};

```

Ž,ffffff□f□f□ff□fffŽ',,□ffff,XŽ,ffff□f□f,,,ffffff12,DWORD,Š",,□ffff,•□,XŽ,'□,,
□□□□□,——%"',,□
DIOBJECTDATAFORMAT dfAnyXAxis = {
    &GUID_XAxis,    // XŽ,,,,,,,,
    12,              //
    DIDFT_AXIS | DIDFT_ANYINSTANCE, // の X
    0,               // 0
};

```

の DirectInput
 の の BYTE の 16

```

DIOBJECTDATAFORMAT dfAnyButton = {
    0,              //
    16,             // ff
    DIDFT_BUTTON | DIDFT_ANYINSTANCE, // ,,fff,,,,,
    0,             // 0
};

```

```

Ž,ffffff f f ff fffŽ',, DirectInput, ,——
%"",ÒFire”fff,“, ,',BYTE, ^f
fff17,’,,
ffff,ÒFire”fff,‘ ,,, ,f f ff fff,fff•%"",,,
DIOBJECTDATAFORMAT dfAnyButton = {
    &GUID_FireButton,    // ffffffff
    17,                  //
    DIDFT_BUTTON |
    DIDFT_ANYINSTANCE, // の
    0,                   // 0
};

```

の ', の 0
 18 BYTE の
 0

```

DIOBJECTDATAFORMAT dfButton0 = {
    0,              //
    18,             //
    DIDFT_BUTTON |
    DIDFT_MAKEINSTANCE(0), // 0
    0,             // 0
};
DIPROPDWORD XE "DIPROPDWORD" §

```



```
typedef struct {
    DIPROPHEADER
    DWORD
} DIPROPDWORD;
diph;
dwData;
```

DWORD の

diph
 の
dwSize = (DIPROPDWORD)の
dwHeaderSize = (DIPROPHEADER)の
dwObj =
dwHow = **dwObj** の Ž•
dwData
SetProperty のの
GetProperty の の

```
DIPROPHEADER XE "DIPROPHEADER" §
typedef struct {
    DWORD dwSize;
    DWORD dwHeaderSize;
    DWORD dwObj;
    DWORD dwHow;
} DIPROPHEADER;
```

,,,ffff□“,□“,’,,,”—□“□

dwSize

Š,,,□“,fff□

dwHeaderSize

```
DIP□ff,DIPH_BYOFFSET,,,□□fffff.fffff,.....fffff,☒□,f□f□f□fff,fffff□
dwHowff□ff,DIPH_BYID,,,□□□%o
,IInputDevice::EnumObjects,☒,□,,,•,,,DIDeviceObjectInstance,dwTypeff□ff,•
,,,fffff□fff,fffffŽ•Ž□
dwHow
dwObj,%Ž•–,Ž’,,□
```

```
DIPROPRange XE "DIPROPRange" §
typedef struct {
    DIPROPHEADER diph;
    LONG lMin;
    LONG lMax;
} DIPROPRange;
```

DIPROP_RANGEfffff, Ź,,,ffffff,”^,fff,Ž“,—,, ffff,”^,—
Œ,,, ”,,,”^,lMin = DIPROP_RANGE_NOMIN,,,lMax =
DIPROP_RANGE_NOMAX,,, ”^,— Œ,,,ffff,,%o, ,, ,,,,,,”^

diph
^%o,,, Š%o,,,,,,,,,,,,,
dwSize = (DIPROP_RANGE),fff
dwHeaderSize = (DIPROPHEADER),fff
dwObj = fffffffŽ•Ž
dwHow = dwObj のŽ•—
lMin
”^,%oŒ
lMax
”^, Œ

JOYCAPS XE "JOYCAPS" §
typedef struct {
WORD wMid;
WORD wPid;

CHAR szPname[MAXPNAMELEN];

UINT wXmin;
UINT wXmax;
UINT wYmin;
UINT wYmax;
UINT wZmin;
UINT wZmax;
UINT wNumButtons;
UINT **wPeriodMin**;
UINT wPeriodMax;
\\ ののの **Windows**
UINT wRmin;
UINT wRmax;
UINT wUmin;
UINT **wUmax**;
UINT wVmin;
UINT wVmax;
UINT wCaps;
UINT wMaxAxes;

UINT wNumAxes;

UINT wMaxButtons;

CHAR szRegKey[MAXPNAMELE•Ž□
wPid

```

    □・Ž・Ž□
    szPname
    ffffffff,□・-,Š,,,□NULL,□,,・Ž—□
    wXmin□wXmax
    x□・',□□',,,,□"□
    wYmin□wYmax
    y□・',□□',,,,□"□
    wZmin□wZmax
    z□・',□□',,,,□"□
    wNumButtons
    ffffffff.fff□□
    wPeriodMin□wPeriodMax
    ffff□fff,ffffffff

```

の

wRmin wRmax

の

4 の

wUmin wUmax**u** (5 の)の**wVmin wVmax****v** (6 の)の**wCaps**

のの

JOYCAPS_HASPOV**JOYCAPS_HASR**

(4 の)

JOYCAPS_HASU

$u \quad (5 \mathcal{O})$

JOYCAPS_HASV

ffffffff,v •(6”–,Ž) •,Ž,

JOYCAPS_HASZ

ffffffff,z • •,Ž,

JOYCAPS_POV4DIR

,ff f,, ’%o ‘ Ć ,,,%o

JOYCAPS_POVCTS

ffffffff,Ž“,~“,•Š,ff f,,

wMaxAxes

ffffffff,ff f,,Ž, ‘

wNumAxes

ffffffff,,Ĭ Ž—,,,,Ž

wMaxButtons

ffffffff,ff f,,fff, ‘

szRegKey

ffffffff,‘,,fffff f ,Š, NULL, ,,•Ž—

szOEMVxD

ffffffffffffOEM,Ž•,, NULL, ,,•Ž—

Ž joyGetDevCaps

JOYINFOEX XE "JOYINFOEX" §

typedef struct joyinfoex_tag {

DWORD dwSize;

DWORD dwFlags;

```

        DWORD dwXpos;
        DWORD dwYpos;
        DWORD dwZpos;
    DWORD dwRpos;
        DWORD dwUpos;
        DWORD dwVpos;
    DWORD dwButtons;
        DWORD dwButtonNumber;
        DWORD dwPOV;
        DWORD dwReserved1;
        DWORD dwReserved2;
} JOYINFOEX;

```

の の

```

dwSize
  の の
dwFlags
  の
0 i の
JOY_RETURNALL

```

JOY_RETURNRAWDATA JOY_RETURN
i

JOY_RETURNBUTTONS

dwButtons

JOY_RETURNCENTERED

ののの

JOY_RETURNPOV

dwPOV

JOY_RETURNPOVCTS

The dwPOV 100 1

JOY_RETURNR

The dwRpos—‘,, —Œ,ff fff •,Š,,, ,,,4”–,Ž,‘ ,,,,Ž,i

JOY_RETURNRAWDATA

fffff□fff,,,,,ffffffffff,ŽŽ',,,□“,Š”,,,,,Ž,i

JOY_RETURNU

dwUpos—‘,,□ffffff,5”–,Ž,Ž–‰”,,,□,,Ž,Š,,—Œ,f□f,Š,,,□□,0,•,,,□

JOY_RETURNV

dwVpos—‘,,□ffffff,6”–,Ž,Ž–‰”,,,□,,Ž,Š,,—Œ,f□f,Š,,,□□,0,•,,,□

JOY_RETURNX

dwXpos—‘,,□ffffff,x□•,Š,,—Œ,f□f,Š,,,□

JOY_RETURNY

dwYpos —‘,,□ffffff,y□•,Š,,—Œ,f□f,Š,,,□

JOY_RETURNZ

dwZpos —‘,,□ffffff,z□•,Š,,—Œ,f□f,Š,,,□

JOY_USEDEADZONE

ffffff,ff□fff^,^,Š,□,,^,ffff□f,Œ,□ffffff,ffff□f“ ,,,,^:,,’□’,•,□

^‰,fff,□ffffff,ffff□fff ,,,,f□f,‘<,□ffff□ffff□ffff□ffff□ffff,‘□,,,,,□

JOY_CAL_READ3

x□y□z□•,“,□□,„,’,dwXpos□dwYpos□dwZpos—‘,,,,,Š”,,□

JOY_CAL_READ4

$ff \square \square \bullet, , , x \square y \square z \square \bullet, \text{“}, \square, \square, , , , , \text{’}, dwRpos \square dwXpos$
 $\square dwYpos \square dwZpos \text{—‘}, , , , , \text{Š”}, , \square$

JOY_CAL_READ5

$ff \square \square \bullet, , , x \square y \square z \square u \square \bullet, \text{“}, \square, \square, , , , , \text{’}, d, \text{’}, \square, \square, \square v \check{Z} f \square f, , , , , \text{“}, \square, \square, , , , , 0, \bullet, \square$

JOY_CAL_READALWAYS

$ffff.ffff, \text{Æ’}, , , , , \square \square, , , , , \square fffffffffff. f \square f, \text{“}, \square, \square$

JOY_CAL_READRONLY

$ffffffffffffffffff, \square, f \square f, \text{’}, \text{’}, \text{’}, \square, \square, f f \square \square \bullet, \text{“}, \square, \square, , , , , \text{’}, dwRpos \text{—‘}, \text{Š”}, , \square, , , , , , 0, \bullet, \square$

JOY_CAL_READUONLY

$ffffffffffffffffff, \text{’}, f \square f, \text{’}, \text{’}, \text{’}, , , , , u \square \bullet, \text{“}, , , , , , \text{’}, dwUpos$
 0

JOY_CAL_READVONLY

0, \bullet,

$v \text{ の } dwVpos$

JOY_CAL_READXONLY

$x \bullet, \text{“}, , , , , , \text{’}, dwXpos \text{—‘}, \text{Š”}, ,$

JOY_CAL_READXYONLY

$x \ y \text{ の } dwXpos \ dwYpos \text{—‘}, , , , , \text{Š”}, ,$

JOY_CAL_READYONLY

y •,“, , ,,,,’,dwYpos—‘,Š”,,

JOY_CAL_READZONLY

z□•,“,□,□,,,,,’,dwZpos—‘,Š”,,□

dwXpos□dwYpos□dwZpos

,,,,□Œ□,x□•□y□•□z□•□

dwRpos

ffffffff,4”–,Ž,,,ff□,Œ□,^”□

dwUpos□dwVpos

,,,,□5”–□6”–,Ž,^”□

dwButtons

32,ffffffffffff,Œ□,□‘□,,—‘,’JOY_BUTTONNflags 00

,,fff,,□,,,n,1,,32,,,’,,,,i,,,,,’%0,,,,,fff,“,,

dwButtonNumber

Œ %0,,,,,fff—‘

dwPOV

Œ ,Ž“ffff f,^” ,—‘,’’,0,,35,900,,,”^“,,, ,,,,’, ŠŽŠ,“^•,100”,,,,,,

dwReserved1 dwReserved2

—,,,,,,Ž—,,

dwSize—‘,’ joyGetPosExŠ□,“の

の5 の JOY_RETURNPOV

のの JOY_POV i

JOY_POVBACKWARD

18,000 180.00 のi

JOY_POVCENTERED

-1

JOY_POVFORWARD

0 0.00 の

JOY_POVLEFT

27,000 270.00 の

JOY_POVRIGHT

□□□□□□□□,,□'9,000,90.00“,•œ,•,□

Windows 95,ffffff,fffffffffffff,„œŽ“,,,,i',Š“,ffff□fff,—,,,,,□□□ —
 %“,,,,~“,f□f,JOY_RETURNPOVCTSfff,—
 „Ž“,,,,,i,,JOY_RETURNPOVCTSfff,□JOY_RETURNPOVfff,—,,,,JOY
 POV'□,ff□f,,,□
 Ž□□joyGetPosEx

ff□ff•□
 c_dfDIKeyboard XE "c_dfDIKeyboard" §
 ,,,,,'<,,,DIDATAFORMAT の
 DINPUT.LIB

の IDirectInputDevice::SetDataFormat

c_dfDIMouse XE "c_dfDIMouse" §
 DIDATAFORMAT
 の DINPUT.LIB

の IDirectInputDevice::SetDataFormat

GUID_SysKeyboard XE "GUID_SysKeyboard" §
 DirectInput の GUID

,', fffff f f,',,fff ffff, ,,,,IDirectInput::CreateDevice,“,,,,,,
 ,

GUID_SysMouse XE "GUID_SysMouse" §
 ,,,,,'<,,,DirectInput,ffffffGUID, fffff,ffffff,„Ž , , ,', ffffff
 f,',,fff ffff, ,,,,IDirectInput::CreateDevice,“,,,,,,
 ,

DIPROP_AXISMODE XE "DIPROP_AXISMODE" §
 ,,,,,'<,,,ffffff, Žf ff f,fff, の
 ,ffffff,',,,,,ffff“,—,,,, dwHowff ff,DIPH_DEVICE,,,,,,
 ,,ffffff,DIPROPDWORD “,—,
 DIPROPDWORD “,pdiphff ff,DIPROPHEADER “,ffff,,,,,, d
 wDataff ff,Žf f,Š,, ,,,Ž,Ž,,,,,
 DIPROPHEADER “,dwObjff ff,0,,,,,, ,,,,ffffff ’,—,ffffff,,,ff
 ff“,—,,,,,Ž, dwSize
 DIPROPDWORD “,fff,fff,,,,,,
 DIPROPDWORD “,dwDataff ff,^%o',,,,
 DIPROPAXISMODE_ABS: ‘ •,Ž~,’,,, Ž~“,ŽŠ,,,,—œ,,,


```
DISCL_FOREGROUND    DISCL_BACKGROUND 0
IDirectInputDevice::SetCooperativeLevel
```

DirectInput

DISCL_EXCLUSIVE XE "DISCL_EXCLUSIVE" \$
SetCooperativeLevel 0 f f, ""ffff,—,,,,Ž, ""ffff,—
,,,, ffff,'ffffff, "Ž,,,,ffff,"""ffff,",,, , 0
fff,"""ffff,",, ,,,,, ffff,"""",ffff,,,',,,,'^,,("" ",,€,€,€
,, DirectDraw,,,,—Ž,Š,"Š,,,,Ž, ,,,,) WM_ENTERSIZEMOVE,, WM_ENTERMENULOOP fff□f,Ž,Ž,□,□ ff
f,,,f□f□f□ffff,0
DISCL_EXCLUSIVE,DISCL_NONEXCLUSIVE,,,,,,,,SetCooperativeLevel,"f,—,□
DISCL_FOREGROUND XE "DISCL_FOREGROUND" \$
SetCooperativeLevel,,,fff□f,□fffffffff□ffff,—,,,,Ž,□fffffffff□fff
0

DISCL_FOREGROUND **DISCL_BACKGROUND** の
IDirectInputDevice::SetCooperativeLevel

DISCL NONEXCLUSIVE XE "DISCL NONEXCLUSIVE" §

SetCooperativeLevel ののの
の

DISCL_EXCLUSIVE DISCL_NONEXCLUSIVE の
IDirectInputDevice::SetCooperativeLevel

XE "

"§

```
□□□□□□□□□□□□dinput.h □□□□□□□□□□□□□□□□□□
□□□□□(256 の)のの
Ž',,,f□f,S''''''—','^,Ž•,,□,,fffff,^,,,f□f,□f
□f□f,f□,%o
,,,,□^'',□,,,,',DIDeviceObjectData□
DIObjectDataFormat□□□□□
DIDeviceObjectInstance □ dwOfs □□□□
□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□
,□,"Ž,,,,—,,,□
^^"" ,f□f□f□ffff'□,%o,Ž,□□□□□□
DIK_ESCAPE Escf□
    DIK_1      1□
```

DIK_9 9 f□
 DIK_0 0 f□
 DIK_MINUS ffff□f□f, - f□
 DIK_EQUALS = f□
 DIK_BACK Backspacef□
 DIK_TAB Tabf□
 DIK_Q Q f□
 DIK_W W f□
 DIK_E E f□
 DIK_R R f□
 DIK_T T f□
 DIK_Y Y f□
 DIK_U U f□
 DIK_I I
 DIK_O O
 DIK_P P
 DIK_LBRACKET [
 DIK_RBRACKET]
 DIK_RETURN Enter
 DIK_LCONTROL Ctrl
 DIK_A A
 DIK_S S
 DIK_D D
 DIK_F F
 DIK_G G
 DIK_H H
 DIK_J J f
 DIK_K K f
 DIK_L L f
 DIK_SEMICOLON ; f
 DIK_APOSTROPHE fffffff□ (')
 DIK_GRAVE accent grave (ˆ) f
 DIK_LSHIFT Shiftf
 DIK_BACKSLASH \ f
 DIK_Z Z f
 DIK_X X f
 DIK_C C f
 DIK_V V f
 DIK_B B f□
 DIK_N N
 DIK_M M f

DIK_COMMA , *f*
 DIK_PERIOD *ffff f f, .f*

DIK_SLASH

DIK_RSHIFT
 DIK_MULTIPLY
 DIK_LMENU
 DIK_SPACE
 DIK_CAPITAL
DIK_F1
DIK_F2

ffff□f□f, / f□

Shift

の *

Alt

Caps Lock

F1

F2

DIK_F3 F3 *f*□**DIK_F4 F4 *f*□****DIK_F5 F5 *f*□****DIK_F6 F6 *f*□****DIK_F7 F7 *f*□**DIK_F7 *fff□,7f□*DIK_NUMPAD8 *fff□,8 f□*DIK_NUMPAD9 *fff□,9 f□*DIK_SUBTRACT *fff□, - f□*DIK_NUMPAD4 *fff□,4 f□*DIK_NUMPAD5 *fff□,5 f□*DIK_NUMPAD6 *fff□,6 f□*DIK_ADD *fff□,+ f□*DIK_NUMPAD1 *fff□,1 f□*

DIK_NUMPAD2 の 2

DIK_NUMPAD3 の 3

DIK_NUMPAD0 の 0

DIK_DECIMAL の .

DIK_F11 F11

DIK_F12 F12

DIK_NUMPADENTER の Enter

DIK_RCONTROL Ctrl

DIK_DIVIDE の /

DIK_SYSRQ SysRq

DIK_RMENU Alt

DIK_HOME Home

DIK_UP
 DIK_PRIOR PgUp
 DIK_LEFT
DIK_RIGHT
 DIK_END End
 DIK_DOWN
 DIK_NEXT PgDn

DIK_INSERT Insertf
 DIK_DELETE Deletf
 DIK_LWIN Windowsf
DIK_RWIN %oWindowsf
 DIK_APPS AppMenuf

fff ffff' XE "fff ffff' " §

DIMOFS_BUTTON0

DIMOUSESTATE “, “, ‘,,fffff0, ‘,fffff ,,’, DIDEVICEOBJECT
 DATA “,dwOfsff ff,,•,, f f,fffff0,“—,,,,,Ž,

DIMOFS_BUTTON1

DIMOUSESTATE “, “, ‘,,ff 1 のの
 ,’, DIDEVICEOBJECTDATA “,dwOfsff ff,,•,, f f,fffff1,“—
 ,,,,,Ž,

DIMOFS_BUTTON2

DIMOUSESTATE□“, “, ‘,,fff 2 ののDIDEVICEOBJECTDATA
 の dwOfs 2

DIMOFS_BUTTON3

DIMOUSESTATE の 3 ののDIDEVICEOBJECTDATA の dwOfs
 3

DIMOFS_X

DIMOUSESTATE □ □ x ののの

DIDEVICEOBJECTDATA □ dwOfs の x の

,,,,,Ž,□

DIMOFS_Y

DIMOUSESTATE□“,□“,’,,fff,y•CE,^,fffff□,,’,□DID□“,’,,fff,z•CE,^,fffff□,,’,□DIDEVICE
 OBJECTDATA□“,dwOfsff□ff□,,•,,□f□f,fff,z•CE,^,“—,,,,,Ž,□

DirectInputffff□fff□f□f XE "DirectInputffff□fff□f□f" §
 DirectInputffff□fff□ f□f,DIDEVICEINSTANCE

DIDEVTYPE_MOUSE: ()

DIDEVTYPE_KEYBOARD:

2 のの

:

DIDEVTYPEMOUSE_UNKNOWN:

DIDEVTYPEMOUSE_TRADITIONAL:

DIDEVTYPEMOUSE_FINGERSTICK:

DIDEVTYPEMOUSE_TOUCHPAD:

DIDEVTYPEMOUSE_TRACKBALL:

:

DIDEVTYPEKEYBOARD_PCXT: IBM PC/XT 83

DIDEVTYPEKEYBOARD_OLIVETTI: 102

DIDEVTYPEKEYBOARD_PCAT: IBM PC/AT 84

DIDEVTYPEKEYBOARD_PCENH: IBM PC 101/102
Microsoft f f f

DIDEVTYPEKEYBOARD_NOKIA1050: Nokia 1050 f f f

DIDEVTYPEKEYBOARD_NOKIA9140: Nokia 9140 f f f

DIDEVTYPEKEYBOARD_NEC98: “–ŒNEC PC98 f f f

DIDEVTYPEKEYBOARD_NEC98LAPTOP: “–ŒNEC PC98

DIDEVTYPEKEYBOARD_NEC98106: NEC PC98 106

DIDEVTYPEKEYBOARD_JAPAN106: 106

DIDEVTYPEKEYBOARD_JAPANAX: AX

DIDEVTYPEKEYBOARD_J3100: J3100

DirectInput XE "DirectInput

" §

DirectInput

DIDFT_ALL

① **IDirectInputDevice::EnumObjects** □ □ □ ,,, —
œ ,,, □ ,,,, f f f f f f , f f f , Š ,,, — < ,,, □ ,,, f f f , ‘ f f f , ‘ , □ ,,,, ,,
,,, □

DIDFT_RELAXIS

f f f f f f , 0 , " □ , " , , , , , , □ , , , Ž , □ □ ^ , 20 , , , , , Ž , □ Ž , DIPROP_GRANULARITY f f f f f , , , , Ž , □ □ ^ , " , □

“Žffff,,□‘•‘,,, ’’,, ‘•’, ffff,”Ž,,,,Š,,,ffff,,”,,“•, Œ,,
,□Œ%,,,□“Žffffff,,“,“•

DIDFT_ABSAXIS

の

の

の 20 の の 20 の
DIPROP_GRANULARITY

DIDFT_AXIS

の IDirectInputDevice::EnumObjects の

DIDFT_PSHBUTTON

DIDFT_TGLBUTTON

DIDFT_BUTTON

ののの
DirectInputŠ ,,,•,,,ff ,—<,,, ŠŠ ,•,,,ff f f,fff,,,, Œ ,Š ,’Ž,Ž
,,,
JOYERR_NOCANDO

ffffff,ffffff,,ffff •, ,,,

JOYERR_NOERROR

—<, —,,

JOYERR_REGISTRYNOTVALID

fffff,,^ ,fffffffffffff,—Œ,f f,Š,,,

JOYERR_UNPLUGGED

Ž',,,,ffffffffff,ffff□f,□',,,,,,□

MMSYSERR_BADDEVICEID

Ž',,,,ffffffffff,Ž•Ž,-œ,,,i

MMSYSERR_INVALIDPARAM

-œ,fff□f,“,“,□

MMSYSERR_NODRIVER

ffffffffff,‘□,,,□