



'5  
 Direct3D,Š—  
 µDirect3D,Š— §  
 Direct3D,,, §  
 Direct3Df fffff §  
 3Dffffff,Š' §  
 fff fff, "%o §  
 •Žf f,Š— §  
 •Žf f,,, §  
 ffffffff: Direct3D,•Žf f fffff §

**Direct3D,•Žf□f,ff□ffff §**

'□f□f,Š—.....	§
'□f f,,,.....	§
fffffff: Direct3D,' f f fffff.....	§
Direct3D,' f f,ff ffff.....	§



Direct3D の Microsoft の 3D API  
の、ffffffffff

Windows

の 3D の の の

の

の の 3D の の

Direct3D OLE  
DirectDraw

COM  
Microsoft の の Windows Direct3D  
: Direct3D の

Direct3D API の DirectX API  
HAL

HAL の Direct3D HAL  
HEL

Direct3D

HEL の  
の API

Direct3D HAL DirectDraw HAL GDI  
HAL Microsoft の API の 3D の Direct3D DirectDraw GDI  
OpenGL の 3D の 3D の Direct3D HAL  
の の

DirectDraw

DirectDraw

DirectDraw 2D DirectDraw の 3D の の Windows

の  
の の

DirectDraw の

Windows の

DirectDraw

DirectDraw

の の Direct3D

DirectDraw  
Microsoft の MS-DOS®

DirectDraw

DirectDraw の

Windows

DirectDraw COM の API

Microsoft の の Windows

DirectDraw

DirectDraw

OpenGL

OpenGL

WindowsNT

CAD/CAM

Windows95

3D の 3D の OpenGL

Windows95 OpenGL

Win32

Win32 OpenGL

の

OpenGL

OpenGL

OpenGL の

の Direct3D、f ffff fff fff、 Direct3D  
 API、%3Df ffff OpenGL、—、—、E,ff f,,,、—  
 ,,,,,,—、,

Direct3Df fffff

,, , Direct3Dfff fff,,,'DirectXfff fff,fff ffff ffff ,,ffff f f  
 fff,,SE,,,、 “、•,< ,,,, ,,,,^%,ffff,,,、—、

• Direct3D,fff

- 
- 
- 
- 
- 
- 
- 
- 
- 
- 

• Direct3D

Direct3D の

Direct3D Windows  
 3D  
 3D の

Direct3D

3D

の

Direct3D の

Direct3D の □□□□□□□□ DirectDraw  
 3D の

DirectDraw

の

2D の

の

3D の

Direct3D

3D

Direct3D

□□□□□□□□ の

の

∞∞∞

の Direct3Dfff□fff,□

Direct3D DirectDraw

DirectDraw の COM

Direct3D の COM

Direct3D DirectDraw の

DirectDraw

の 3D の 3D の

Direct3D の

Direct3D の 3D

- の 4 4 の
- の
- の

2D 3D の

Direct3D

§

の

DirectDraw Direct3D  
DirectDraw

§

”,,,□,□Direct3D,-,□,,,□ffff□fff□f□f,“□,,,fff□fff,Š  
Direct3D

Direct3D

ISV の

の

Direct3D API の

Direct3D の D3DOPCODE の D3DINSTRUCTION の

§

の D3DOP\_TRIANGLE

の

の

IDirect3DDevice::GetCaps  
dwMaxBufferSize  
,,,□□□^”“,□□“,Ž□fff 64K の

D3DDEVICEDESC の  
Ž—

Direct3D の の の

の  
の

4 の

3D の の

3D 2D

の の

の の

の D3DTLVERTEX

D3DTLVERTEX

D3DLVERTEX の の  
D3DLVERTEX

D3DVERTEX の の の の の の

D3DTRANSFORMDATA

の の の の の API  
の の 3D

*fff, ..., ffff, 0, 0,*

D3DLIGHTINGELEMENT

の  
の

RGB の D3DLIGHTDATA の

D3DLIGHTINGELEMENT の

の ” ”  
の の  
の

RGB の の

の 0 の

1 の の の の

§



の  
の D3DLIGHTDATA

の

IDirect3DDevice::Execute の

の(0, 0)の ( -1, -1)の 0 1  
のののののののの

Direct3D の

のDirect3D の

- 
- 

IDirect3DRM::CreateDeviceFromSurface  
Direct3D

DirectDraw

DirectDraw の  
の SDK のDirect3D DirectDraw

WM\_ACTIVATE

Direct3D 8 の DirectDraw の  
IDirectDrawPalette::GetEntries

の

D3DPAL\_FREE

D3DPAL\_READONLY

D3DPAL\_RESERVED

の Win32 の PALETTEENTRY の peFlags の D3DRMPALETTEENTRY の  
D3DRMPALETTEFLAGS RGB の

RGB の

D3DTLVERTEX の specular の  
RGBF "F" "fog" の F

の  
の  
の  
の  
の

§

•Ž□fff□f□f□f,,□Ž□fff□f

§

の e 2.71828,,, fff,“-,•,Š ,,,  
fff,’ ,,,, fffffff,☉,,,,,  
ffff fff,Ž fff f f,Ž—, fff,-“0.5,,,,, Ž,—,Ž,,, fff,,0.8,←  
,,^,fff,’0.6703,,,

§

ff f,ff f f f fff

Direct3D, ☉ , ‘ , ’,,,,, Ž ffff’,f f,%o ,, ffff fff, ffff,f f,ff  
ffff,,,,,ŽŽ,,‘ , ‘,ffffff,, D3DSTATE “, , ,f fffff,Ž ,,Ž,,,—☉,  
Š,,,, D3DTRANSFORMSTATETYPE,•Šfff f, ‘ , ’,, , D3DLIGHT  
STATETYPE, -fff f, ‘ , D3DRENDERSTATETYPE,fff —  
fff f, ‘ , ’,,  
,,,, ‘ , “,Ž, —,Ž,BOOL’,fff,Ž,,,, ,fff,TRUE, ,^ , • , ,,,  
ffff fff, D3DSTATE\_OVERRIDEfff,Ž—, “,Ž, — ‘,-  
☉,,,,,,, ,<”,,,,ffff fff, Ž ffff, —  
, ffff, ‘,•,,,,,“ ,• ,,,,,, Direct3D,•Žf f,, ff f f f fff,—  
—,Ž ,,,fff,,, ff f f f fff,—,,, Ž ffff,Š‘ , ’,•—  
, ,,,, ,,,, ffff,ffffff,ff f,ffffff,’Š,,,, •Žf fAPI,ff f f f fff  
,—,,

ffff□fff,□  
ff□□ff□ffff□f□f,fff,,,ffffff,,,,,D3DSTATE\_OVERRIDEfff,Ž—  
,,,,,,□,,○



の  
の z の

の

Direct3D の

- Direct3D の
- U V

Direct3D の

Direct3D の z の

§

の

Direct3D

- v0 v1 v2 v0 v2 v1 Direct3D
- Z -1 の D3DMATRIX の\_13
- \_33 \_43 の

U V

Direct3D の u v  
 v の z u  
 y u v [0,0,0]  
 IDirect3DRenderWrap

3D

3D の

- の の
- の
- 

の 4 の (x, y, z) (x', y', z')

§

の (x', y', z') (x, y, z)

§

の

§

のの

D3DMATRIX scale = {

D3DVAL(s), 0, 0, 0,

0, D3DVAL(s), D3DVAL(t), 0,

0, 0, D3DVAL(s), D3DVAL(v),

0, 0, 0, D3DVAL(1)

};

のDirect3D

3D

- 
- 
- 

ののののの API

のDirect3D

の(x, y, z) (x', y', z')

§

の

□, " ,,,,,, □,fff

の(x, y, z) x

(x', y', z')

§

のy

§

のz

§

のののθ

の(x, y, z) x y zの(x', y', z')

§

Direct3D の 3 のののの

の Direct3D

- 
- 
- 
- の

§

のの

の

§

の の  
Direct3D のの Direct3D

§

Direct3D ののの

§

ののの  
の  
ののの  
の  
の

**Note**

∞

§

ののの

*ffffff*, '□,,,,,,

の

§

§

の

∞

□□□□□  
のDirect3D

の

の

∞

•

•

•

•





D3DRMVectorDotProduct

D3DRMVectorModulus

D3DRMVectorNormalize

D3DRMVectorRandom

D3DRMVectorReflect

D3DRMVectorRotate

D3DRMVectorScale

D3DRMVectorSubtract

Direct3D の DirectX  
の 53 の

53 の

の



のDirect3D

HAL

HAL

の

のののの

•

CPU の

• の

• の 256×256 の4 の 128×128 の256×256 の 1 ののの256×256 の,,,,□,,,,□‘

D3DTRIANGLE の wFlags のの

のの ffff fff,\*,Ž ,,,,,,,

D3DTRIFLAG\_STARTFLAT(len)

CE ,ŽŠCE,Ž, ,,,, ffff, “Š , CE,, CE,‘,ŽŠCE,len,Ž, ,,Ž, ,,,,,,

D3DTRIFLAG\_ODD,D3DTRIFLAG\_EVEN

ffff,□ŽŠCE,,,,,□,’,“,,, f□f,,,,□‘,,,,“,fffff

のD3DTRIFLAG\_STARTFLAT

D3DTRIFLAG\_ODD

D3DTRIFLAG\_EVEN の

D3DTRIFLAG\_STARTFLAT のの

の

の

D3DTRIFLAG\_ODD D3DTRIFLAG\_EVEN

D3DTRIFLAG\_START

D3DTRIFLAG\_START の

D3DTRIFLAG\_START の

D3DTRIFLAG\_ODD

D3DTRIFLAG\_EVEN

の SDK D3DTRIFLAG\_ODD D3DTRIFLAG\_EVEN

の

の

のの

- の
- の64
- 

ffff,fff fff,16^%o,,

- 

,,,fffff fffff ffff, , ,“^,  
 , fffff,ff fff,Ž’,,,,,,, ,,,  
 ,,,fffff,” ,,, ”Ž—  
 ;“;” ‘,ffff fff,, ,,,f f,,  
 ^ ,fffff,•—  
 ,, ,,,œ,”Ž,,,,fffff,, ,,,,”

Ž,

- 
- の256×256 の1 の
- 

のの □□□□□□ のの

のののの □□□□□□□□ のののの  
 f□f□ff,Ž

8の

のののの

Z

Z

Z

Z

の

Z

の Z 2

の Z Z  
の

### API Z の の の Z

の の SDK の D3dtest.exe の 4 の

の の の の

の SDK の Direct3D の

の の の の D3dtest.exe の の の の の

Direct3D の の-systemmemory

の の fail

DirectDraw

WIN16

WIN16fff, □GDI, USER,, ffff, ffff%, □IDirectDrawSurface2::Lock

IDirectDrawSurface2::Unlock の の Windows

IDirectDrawSurface2::GetDC IDirectDrawSurface2::ReleaseDC の の

### D3DTEXTUREBLEND の

D3DTBLEND\_COPY

8 の

16

16

の 8 の 2

8 の の

D3dtest.exe

の

の Direct3D の 3D  
3D

Microsoft

Direct3D の API

Direct3D DirectDraw

DirectDraw

DirectDraw Direct3D の の IDirectDraw::QueryInterface





の COM                      GetElement    GetSize ののの IDirect3DRM

IObjectName::QueryInterface ののの IDirect3DRMDevice::QueryInterface  
IDirect3DRMWinDevice  
IDirect3DRMVisual

Direct3DRMAnimation  
IDirect3DRMAnimation

Direct3DRMAnimationSet  
IDirect3DRMAnimationSet

Direct3DRMDevice  
IDirect3DRMDevice, IDirect3DRMWinDevice

Direct3DRMFace  
IDirect3DRMFace

Direct3DRMFrame  
IDirect3DRMFrame, IDirect3DRMVisual

Direct3DRMLight  
IDirect3DRMLight

Direct3DRMMaterial  
IDirect3DRMMaterial

Direct3DRMMesh

IDirect3DRMMesh, IDirect3DRMVisual

Direct3DRMMeshBuilder

IDirect3DRMMeshBuilder, IDirect3DRMVisual

Direct3DRMShadow

IDirect3DRMShadow, IDirect3DRMVisual

## Direct3DRMTexture

IDirect3DRMTexture, IDirect3DRMVisual

Direct3DRMUserVisual

IDirect3DRMUserVisual, IDirect3DRMVisual

Direct3DRMViewport

IDirect3DRMViewport

Direct3DRMWrap

IDirect3DRMWrap

の Direct3DRMDevice  
IDirect3DRM::CreateObject Direct3DRMDevice  
の  
IDirect3DRMDevice::InitFromClipper  
*IDirect3DRMDevice::QueryInterface* の Direct3DRMDevice の WM\_PAINT  
WM\_ACTIVATE の IDirect3DRMWinDevice

```
d3drmapi->CreateObject(CLSID_CDirect3DRMDevice, NULL,  
IID_IDirect3DRMDevice,(LPVOID FAR*)&dev1);  
dev1->InitFromClipper(lpDDClipper, IID_IDirect3DRMDevice,  
r.right, r.bottom);
```

```
dev1->QueryInterface(IID_IDirect3DRMWinDevice, (LPVOID*) &dev2);
```

```
QueryInterface  
①  
Direct3D      IDirect3DRMObject      IUnknown  
                IDirect3DRMObject  
                •—  
                ,,,, fffž•ž CLSID ,ž,,,, IDirect3DRM::CreateObjectffff,€, ,,, ”  
                —ffffff, ,,,,,, ,,‘,, Šfff ffff,,,%o, ffff,ž—,,  
                ”—fff ffff  
                ffff
```

```
IDirect3DRMDeviceArray
```

```
IDirect3DRM::GetDevices
```

```
IDirect3DRMFaceArray
```

```
IDirect3DRMMeshBuilder::GetFaces
```

```
IDirect3DRMFrameArray
```

```
IDirect3DRMPickedArray::GetPick
```

```
IDirect3DRMFrame::GetChildren
```

```
IDirect3DRMLightArray
```

```
IDirect3DRMFrame::GetLights
```

# IDirect3DRMPickedArray

IDirect3DRMViewport::Pick

IDirect3DRMViewportArray

IDirect3DRM::CreateFrame

IDirect3DRMVisualArray

IDirect3DRMFrame::GetVisuals

fffff, Ź ffff,0,,,,, ,,,,,,  
ffff fff, f f,f ffffffff,,,,, Ź ffff,•Ź,,•—  
,, ‘,ffffff,Ź ffff, Ź““,ffff,Š—,, ffff fff, —  
,,‘f f,ff f f ffff,%o\*,,,,,, ffff fff,ff f f,%o\*,,, ffff,Ź““,ff  
f,Ź ffff, —,, —““, ,,ff f f,ffff,%o  
,, ,,, ffff fff,ffff,%o\*,,,ff f f,%o\*,,,,,, ,, ffff,%o\*,,,,, “  
—,ff f f,%o\*,,,,,,,

Źffffff,ffffff□ffffff,Ź□ffff,□ff□f,%o  
,,,,,‘,,,□IDirect3DRMFrame::AddChildffff,Ź—  
,,□,,□ffffff,,•,□,Źffffff,^“,,,□ffff,Ź““,Ź□ffff,‘□,,□  
ffff□fff,ffff

ののののの

IDirect3DRM

IDirect3DRM

の

Direct3DRM のののIdirect3DRM

IDirect3DRM

のの

- *fff□ff*
- 
- *face*
- *ff f*
- *Š%o,,,,,~”ffffff*
- *Ⓔ*
- *fffff*
- *ffff,ffff fff*
- *%o%o*
- *fffff*
- *f f fffff*
- 
- 

IDirect3DRMAnimation IDirect3DRMAnimationSet

Direct3DRMAnimation

Direct3DRMFrame

Direct3DRMAnimation

Direct3DRMVisual

Direct3DRMLight

Direct3DRMViewport

IDirect3DRMAnimation::AddPositionKey  
 IDirect3DRMAnimation::AddRotateKey  
 IDirect3DRMAnimation::AddScaleKey のの 99 の 49 のの0  
 のの

IDirect3DRMAnimation::SetTime ののの  
 IDirect3DRMAnimation::SetTime の

Direct3DRMAnimationSet □□□□□□□□Direct3DRMAnimationSet □□□□□□□□□□□□□□□□  
 のの *ffff□f,<-*  
*,□•ž,ffff□fff,□□,Š-%*  
*,,,□ffff□fff,□IDirect3DRMAnimationSet::AddAnimationffff,-*  
*,,ffff□fff,ffff□fff□fff,'%*  
*,,,,,,,□ffff□fff,□□,,,□IDirect3DRMAnimationSet::DeleteAnimationffff,*  
*ž-,□ffff□fff□fff,□IDirect3DRMAnimationSet::SetTimeffff,Ⓔ,□,,,,,Ⓔ,,,,□*

Š~□•,,,,,□□IDirect3DRMAnimation□,,,□IDirect3DRMAnimationSet□fff□ffff,  
 Ž□,,,,□  
 IDirect3DRMDevice,IDirect3DRMDeviceArrayfff□ffff  
 ffffff,,,,,,□—ŒŽ,□□—ffff,Š~,,,,,,□ffff□ffff□ffff□—  
 □,%Žfffff,•,,,,□  
 ffffff,“□,□Ž’,,,□

- ””
- Žf f, fff f,fffff ,,ffff fff fff,’ ffffff,,ffff,ff f,,  
 Š~ •,,,,, IDirect3DRMDevice ,Ž ,,,  
 ,,fffff,, Direct3Dff f, —ffff,•Ž,, ,—Œ,fffff,,,, -,  
 • fffff  
 • ff fff  
 •

IDirect3DRMDevice::SetQuality  
 IDirect3DRMMeshBuilder::SetQuality のIDirect3DRMDevice::GetQuality  
 IDirect3DRMMeshBuilder::GetQuality

RGB

IDirect3DRMDevice::GetColorModel

RGB ,•□,ŒŒ  
 の 8 16 24  
 32 の  
 24 の

の

の □f,,,,,のののの RGB のの

の

の

RGB 8 16 24 32 の  
 8 の

RGB

Direct3D の IDirect3D::EnumDevices  
のの

IDirect3D::FindDevice

WM\_PAINT WM\_ACTIVATE

IDirect3DRMWinDevice::HandlePaint

IDirect3DRMWinDevice::HandleActivate

WM\_MOVE

Direct3D

IDirect3DRMWinDevice

IDirect3DRMFace IDirect3DRMFaceArray

のの

IDirect3DRMFace::SetColor

IDirect3DRMFace::SetColorRGB IDirect3DRMFace::SetTexture

IDirect3DRMFace::SetMaterial の

IDirect3DRMFace::AddVertex

IDirect3DRMFace::AddVertexAndNormalIndexed

の

IDirect3DRMFace::GetVertices IDirect3DRMFace::GetVertex *ffff, Z*—

IDirect3DRMFace

IDirect3DRMFrame IDirect3DRMFrameArray

ののののの

ののの

の□□□の□□□□□ の

のの 1 NULL IDirect3DRM::CreateFrame

*Direct3D の3D*

IDirect3DRMFrame

のの

- 
- 
- 
-



のDirect3D  
の3D

0 □□□□□□

の

□f,‘,

IDirect3DRMFrame::AddMoveCallback

Direct3DRMFrame::DeleteMoveCallback

のの

ののの

の

IDirect3DRMLight IDirect3DRMLightArray

,, ,,fffff, ,, fffff, , ff  
ffff’,\$CE%,%o<, —,

CE’ の

ののののの

のの

の

Direct3D

の

5 の

**IDirect3DRMLight**

**IDirect3DRMLight**

**Direct3D**

- 
- 
- 
-

•

のの  
ののの

の

の

の

の

ののののの

の  
IDirect3DRMLight::GetPenumbra IDirect3DRMLight::GetUmbra  
IDirect3DRMLight::SetPenumbra IDirect3DRMLight::SetUmbra

§

IDirect3DRMMaterial

ののemissive specular□,□,□,□—‘,,, ” ののののの  
5 のの

の

IDirect3DRMMaterial::GetEmissive IDirect3DRMMaterial::SetEmissive  
IDirect3DRMMaterial::GetSpecular  
IDirect3DRMMaterial::SetSpecular  
IDirect3DRMMaterial::GetPower IDirect3DRMMaterial::SetPower

IDirect3DRMMaterial ののIDirect3DRMMaterial

IDirect3DRMMesh IDirect3**DRMMeshBuilder**

のの

ののの

のの

ののの

の2D

IDirect3DRMMesh IDirect3DRMMeshBuilder のの COM  
IDirect3DRMMesh

IDirect3DRMMeshBuilder IDirect3DRMMesh の  
IDirect3DRMMeshBuilder

**Direct3DRMMeshBuilder**

**Direct3DRMMesh**

の

のの

の

IDirect3DRMMesh

のの

IDirect3DRMMesh::AddGroup

の

のの

IDirect3DRMMeshBuilder IDirect3DRMMesh の  
64 のののののDirect3DRMMesh

**Direct3DRMMeshBuilder API Direct3D API**

IDirect3DRMMeshBuilder::AddVertex

IDirect3DRMMeshBuilder::AddFace

IDirect3DRMMeshBuilder::AddFaces

IDirect3DRMMesh::SetGroupColor

IDirect3DRMMesh::SetGroupColorRGB

**IDirect3DRMMesh::SetGroupTexture** IDirect3DRMMesh::SetGroupMaterial

ののののの 1

**IDirect3DRMFrame::AddVisual**

IDirect3DRMMesh::SetGroupQuality ののの  
D3DRMRENDERQUALITY の

のIDirect3DRMMeshBuilder::GenerateNormals

Direct3DRMObject

Direct3DRMObject ののDirect3DRMObject の







**IDirect3DRM::CreateUserVisual**  
**IDirect3DRMUserVisual::Init**  
の

**IDirect3DRMViewport and IDirect3DRMViewportArray**

3D の 2D

の

ののの

IDirect3DRMViewport

ののの

•

• **Z** □ ‘

•

•

**Direct3DRMFrame**

z

y

の

の

**IDirect3DRMViewport::SetCamera**  
**IDirect3DRMViewport::GetCamera**

のののの

,3Dfff□f,,,□Z□‘,’,,,,ffffff,%oZ□‘,,,□“Z“%o,,□ffff□ffffff□ののの

§

のの z

D の

F の IDirect3DRMViewport::SetFront

**IDirect3DRMViewport::SetBack** **IDirect3DRMViewport::GetFront**

IDirect3DRMViewport::GetBack の 2h の

h の

IDirect3DRMViewport::SetField

IDirect3DRMViewport::GetField

の A の

の h の

§

**D3DRMPROJECTIONTYPE**  
**IDirect3DRMViewport::GetProjection**  
**IDirect3DRMViewport::SetProjection**

**3D 2D**

4 の [x y z w]  
 3 の [x/w y/w z/w] [x/w y/w]  
 z/w 0 1 の 0  
 1 の の の

の の h F の z D の z

§

Direct3D 4,4 の

の の の s の o の

§

の  
 W の

の P

§

sx sy ox oy [-h -h D] [h h D] の の

f •,ffff,Š,,, IDirect3DRMViewport::Transform,IDirect3DRMViewport:  
 :InverseTransformffff,Ž—,, Ž,—,,, ffff fff, ,,,,ffff,—  
 ,,ffff,ff f,,,,,,

/\*

\* Drag a frame by [delta\_x delta\_y] pixels in the view.

\*/

void DragFrame(LPDIRECT3DRMVIEWPORT view,

LPDIRECT3DRMFRAME frame,

LPDIRECT3DRMFRAME scene,

int delta\_x, int delta\_y)

{

D3DVECTOR p1;

D3DRMVECTOR4D p2;

frame->GetPosition(scene, &p1);

view->Transform(&p2, &p1);

p2.x += delta\_x \* p2.w;



v z u y  
[0, 0, 0]

IDirect3DRMWrap  
IDirect3DRMWrap

4

- 
- 
- 
- 
- 

D3DRMMAPPING D3DRMMAP\_WRAPU  
D3DRMMAP\_WRAPV の

fffff □ Š □ ←  
u v の  
∞  
v ∞∞∞ u

- fffff fff,fff,,,,,•fffff f f,, u,,v •Ž',,,•-, fffff,-  
E,fffff,,, ,, —E,u,,v •',1.0^ ,,, “(0.1, 0.1),(0.9, 0.9)  
,E, ', ' , “(0.5, 0.5),’%o,,

D3DRENDERSTATE\_WRAPU,,D3DRENDERSTATE\_WRAPV,,,,,fff,  
,, fffff,%oŽ1.0, ,,,,%o“E,, 1.0^ ,fffff •, fff,,,,,,”“,—  
E,, fffff •Š, '←  
,fffff fff,,,^,, D3DRENDERSTATE\_WRAPU,fff,,,,,,, “(0.1, 0.1)  
,,(0.9, 0.9),,, ', ' , “ 0, 0.5 ,’%o,,

- D3DRENDERSTATE\_WRAPU,D3DRENDERSTATE\_WRAPVfff,—  
•,fff,,,,, fffff,%oŠ: f ff ,,, ffff,• ,,,, 1.0^ ,fffff •,—  
E,, “(0.1, 0.1),(0.9, 0.9),,, ', ' , “(0, 0),’%o,,

—E—^,,,fffff •,—E—^Ž,,,,, ,,,,“ ,• ,,,,  
~,fffff fff, fffff,E -,•,^,,,,, %o' ,fff,,,,,fffff fff,fff,, ,,,,—  
,fffff,”•^ Ž—,,, ,, fffff fff, ',,

—

•—fff,, fffff, ffffff ,fffff,,,ff,,,fffff,—^’,,

Ž,“Ž,, ffff[x y z],[u v] •,’,,

u=sux—ou

$$v = sv_y - ov$$

,,,EŽ,,, s,ffff,ff fffE o,ffff,E“,Ž,,, ffff fff, 1’,ff fff  
 E , u,,,v,—E,’Ž,,,x,y,0,,1,”^,fff,,,,ffff,E’,,,,,,,,

%’

%’fff,, fffff,%’,%o,,•Ž,,,^,, , ,’,%o’,,,,,,, fffff,%oŽ,’%o  
 ,”,, fffff,f fff, ,,ffff,•E,,,

%o , %’ffff fff,,,Šffff,E%o,Ž,,,,,,

§

•Effff,%o’,Ž,Ž,□□ffff,%o“,Š‘,u = 0,,“,Ž,□ffff,□ffff[x y z],’,ffff,[u  
 v]□•O

§

u v z 0 1 O v  
 O

[x y 0] x O u v [x y z]  
 z O O z

§

O

§

u v O 0 1  
 O

,”Ž,,E,,,,ffff •Š“,,, fff  
 fff,Ž ff f,^,Ž“,

”Ž O O u O v O O

Direct3D O O

Windows O Direct3D O

Windows □,,,,,

O O O O O

O

§

のの

- Helworld.c

•

•

Windows の

- の
- 3D
- 
- の

•

## Helworld.c

の 3D Direct3D のの の .c DirectX  
 のの SDK の Hello.ppm の Sphere3.x の ffff,-^, 3D  
 の の SDK の Direct3D の  
 の Direct3D のの

•

•

- DirectDraw の

の

の Helworld.c  
 SDK の Shpere3.x

Helworld.c Helworld.c DirectX  
 Hello.ppm の

のDirectX SDK のGlobe の —%o  
 ,,,,□SDK のDirect3D のGlobe Rmmain.cpp Helworld.c  
 Rmmain.cpp のC++ C

のののののの”Hello, world!” 3D の

□□□ **fff**□**ffffff**

Direct3D,Žf□f□ffff□fff,□ffffff□ffff Winmn.lib D3drm.lib

DirectDraw の

のDirect3D DirectDraw  
 のDirectDraw のの

## のHelworld.c

## Helwold.c ののの

```
INITGUID のdefine DirectX

////////////////////////////////////
//
// Copyright (C) 1996 Microsoft Corporation. All Rights Reserved.
//
// File: Helworld.c
//
// "Globe" SDKffff,Š,, Š',Direct3DŽf f,ffff
//
////////////////////////////////////

#define INITGUID // 'fff'<ffff f,'
                // '<,,,,,,,'
#include <windows.h>
#include <malloc.h> // memset,CE, ,,•—
#include <d3drmwin.h>

#define MAX_DRIVERS 5 // D3Dffff, ‘
```

```

// ff ff•

LPDIRECT3DRM lpD3DRM; // Direct3DRMffffff
LPDIRECTDRAWCLIPPER lpDDClipper;// DirectDrawClipperffffff

struct _myglobs {
    LPDIRECT3DRMDEVICE dev; // Direct3DRMffff
    LPDIRECT3DRMVIEWPORT view; // f f,Ž,,Direct3DRMff f f

    LPDIRECT3DRMFRAME scene; //
‘,ffffff,’”,,,fff□□ff□f
    LPDIRECT3DRMFRAME camera; // の POV

    GUID DriverGUID[MAX_DRIVERS]; // D3D の GUID
    char DriverName[MAX_DRIVERS][50]; // D3D の
    int NumDrivers; // D3D の
    int CurrDriver; // D3D の

    BOOL bQuit; //
    BOOL bInitialized; // の D3DRM
    BOOL bMinimized; //

    int BPP; // の

} myglobs;

// の

static BOOL InitApp(HINSTANCE, int);
long FAR PASCAL WindowProc(HWND, UINT, WPARAM, LPARAM);
static BOOL EnumDrivers(HWND win);
static HRESULT WINAPI enumDeviceFunc(LPGUID lpGuid,
    LPSTR lpDeviceDescription, LPSTR lpDeviceName,

```

```

    LPD3DDEVICEDESC lpHWDesc, LPD3DDEVICEDESC lpHELDesc,
    LPVOID lpContext);
static DWORD BPPToDDBD(int bpp);
static BOOL CreateDevAndView(LPDIRECTDRAWCLIPPER lpDDClipper,
    int driver, int width, int height);
static BOOL SetRenderState(void);
static BOOL RenderLoop(void);
static BOOL MyScene(LPDIRECT3DRMDEVICE dev,
    LPDIRECT3DRMVIEWPORT view,
    LPDIRECT3DRMFRAME scene, LPDIRECT3DRMFRAME camera);
void MakeMyFrames(LPDIRECT3DRMFRAME lpScene,
    LPDIRECT3DRMFRAME lpCamera,
    LPDIRECT3DRMFRAME * lpLightFrame1,
    LPDIRECT3DRMFRAME * lpWorld_frame);
void MakeMyLights(LPDIRECT3DRMFRAME lpScene,
    LPDIRECT3DRMFRAME lpCamera,
    LPDIRECT3DRMFRAME lpLightFrame1,
    LPDIRECT3DRMLIGHT * lpLight1, LPDIRECT3DRMLIGHT *
    lpLight2);
void SetMyPositions(LPDIRECT3DRMFRAME lpScene,
    LPDIRECT3DRMFRAME lpCamera, LPDIRECT3DRMFRAME
    lpLightFrame1,
    LPDIRECT3DRMFRAME lpWorld_frame);
void MakeMyMesh(LPDIRECT3DRMMESHBUILDER * lpSphere3_builder);
void MakeMyWrap(LPDIRECT3DRMMESHBUILDER sphere3_builder,
    LPDIRECT3DRMWRAP * lpWrap);
void AddMyTexture(LPDIRECT3DRMMESHBUILDER lpSphere3_builder,
    LPDIRECT3DRMTEXTURE * lpTex);
static void CleanUp(void);

```

Windows の

のHelworld.c

Windows の

- WinMain



```

        return 1;

while (!myglobs.bQuit) {

    //

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

        if (!TranslateAccelerator(msg.hwnd,
accel, &msg)) {
            TranslateMessage(&msg);
            DispatchMessage(&msg);
        }
    }

    //

    // D3DRM

    if (!myglobs.bMinimized && !myglobs.bQuit &&
        myglobs.bInitialized) {

        // 2

        //

        if (!RenderLoop())
            ++failcount;
        if (failcount > 2) {
            CleanUp();
            break;
        }
    }
    return msg.wParam;
}

InitApp☐

```



```

// fffff,00,,0

win =
    CreateWindow
    ( "D3DRM Example",          // fffff0fff
      "Hello World (Direct3DRM)", // fffff0
      WS_VISIBLE | WS_OVERLAPPED | WS_CAPTION | WS_SYSMENU |
        WS_MINIMIZEBOX | WS_MAXIMIZEBOX,
      CW_USEDEFAULT,           // 0x0•
      CW_USEDEFAULT,           // 0y0•
      300,                      // 0$,•
      300,                      // 0$,0,
      NULL,                     // 0fffff
      NULL,                     // fff0fffff
      this_inst,                // fffff,ffffff0fffff
      NULL                       // 00fff0f
    );
if (!win)
    return FALSE;

// 0,ffffff,fff",,,fff,<%,,,0

hdc = GetDC(win);
myglobs.BPP = GetDeviceCaps(hdc, BITSPIXEL);
ReleaseDC(win, hdc);

// D3Dffff,-<,0,,,,`\,,0

if (!EnumDrivers(win))
    return FALSE;

// D3DRMffffff,D3DRMffffff,00,,0

lpD3DRM = NULL;
Direct3DRMCreate(&lpD3DRM);

// fff00f0f,ff0f,fff0ff0f,00,,0

lpD3DRM->lpVtbl->CreateFrame(lpD3DRM, NULL, &myglobs.scene);
lpD3DRM->lpVtbl->CreateFrame(lpD3DRM, myglobs.scene,
    &myglobs.camera);
myglobs.camera->lpVtbl->SetPosition(myglobs.camera,
myglobs.scene,
    D3DVAL(0.0), D3DVAL(0.0), D3DVAL(0.0));

// DirectDrawClipperffffff,00,0fffff,š~•,,0

DirectDrawCreateClipper(0, &lpDDClipper, NULL);
lpDDClipper->lpVtbl->SetHWND(lpDDClipper, 0, win);

// `,,,,D3Dffff,-,,D3DRMffff,00,,0

GetClientRect(win, &rc);

```

```

if (!CreateDevAndView(lpDDClipper, myglobs.CurrDriver, rc.right,
    rc.bottom)) {
    return FALSE;
}

// ffffff,,,f□f,□□,,□

if (!MyScene(myglobs.dev, myglobs.view, myglobs.scene,
    myglobs.camera))
    return FALSE;

myglobs.bInitialized = TRUE; // □Š%Š-

// fffff,•Ž,,□

ShowWindow(win, cmdshow);
UpdateWindow(win);

return TRUE;
}

ffff□fffff□ffff□ff
Helworld.cffff,fff□fffff□ffff□ff,"□,'□,,,□ž□□,,,ffff,f□f,“-
, \,ž,•,,,ffff□ffff,,,□
fffff□ffff□ff,□WM_DESTROYfff□f,ž,ž,,CleanUpŠ□,☉,□□
,,WM_ACTIVATEfff□f,ž,ž,,,,,□fffff□ffff□ff, IDirect3DRMWinDevice,ž“,□ff
fff,fffff□ffff□□□□,•ž,,,,, IDirect3DRMWinDevice::HandleActivatefff
,☉,□□,□□“-
,□WM_PAINTfff□f,%“,□fffff□ffff□ff, IDirect3DRMWinDevice::HandlePaintff
ff,☉,□□
////////////////////////////////////
//
// WindowProc
// fff□fffff,fff□f□ffff
//
////////////////////////////////////

LONG FAR PASCAL WindowProc(HWND win, UINT msg,
    WPARAM wparam, LPARAM lparam)
{
    RECT r;
    PAINTSTRUCT ps;
    LPDIRECT3DRMWINDEVICE lpD3DRMWinDev;

    switch (msg)    {

    case WM_DESTROY:
        CleanUp();
        break;

    case WM_ACTIVATE:
        {

            // ,,ffff□f,□-,□fffff☉-,D3DRMfffff□fffff,

```

```

// □□,,□

LPDIRECT3DRMWINDEVICE lpD3DRMWinDev;
if (!myglobs.dev)
    break;
myglobs.dev->lpVtbl->QueryInterface(myglobs.dev,
    &IID_IDirect3DRMWinDevice, (void **) &lpD3DRMWinDev);
lpD3DRMWinDev->lpVtbl->HandleActivate(lpD3DRMWinDev,
    (WORD) wparam);
lpD3DRMWinDev->lpVtbl->Release(lpD3DRMWinDev);
}
break;

case WM_PAINT:
    if (!myglobs.bInitialized || !myglobs.dev)
        return DefWindowProc(win, msg, wparam, lparam);

// ,,ffff□f,□-,,,□ffffffE-,D3DRMffffff□ffff,
// □□,,□

if (GetUpdateRect(win, &r, FALSE)) {
    BeginPaint(win, &ps);
    myglobs.dev->lpVtbl->QueryInterface(myglobs.dev,
        &IID_IDirect3DRMWinDevice, (void **) &lpD3DRMWinDev);
    if (FAILED(lpD3DRMWinDev->lpVtbl-
>HandlePaint(lpD3DRMWinDev,
        ps.hdc)))
        lpD3DRMWinDev->lpVtbl->Release(lpD3DRMWinDev);
    EndPaint(win, &ps);
}
break;
default:
    return DefWindowProc(win, msg, wparam, lparam);
}
return 0L;
}

```

の

Direct3D の

- EnumDrivers





```

if (!(lpDesc->dwDeviceRenderBitDepth & BPPToDDBD(myglobs.BPP)))
    return D3DENUMRET_OK;

// のの GUID

memcpy(&myglobs.DriverGUID[myglobs.NumDrivers], lpGuid,
    sizeof(GUID));
lstrcpy(&myglobs.DriverName[myglobs.NumDrivers][0], lpDeviceName);

//                                     RGB

if (*lpStartDriver == -1) {

    // の

    *lpStartDriver = myglobs.NumDrivers;
    hardware = lpDesc == lpHWDesc ? TRUE : FALSE;
    mono = lpDesc->dcmColorModel & D3DCOLOR_MONO ? TRUE :
FALSE;
    } else if (lpDesc == lpHWDesc && !hardware) {

    // の

    *lpStartDriver = myglobs.NumDrivers;
    hardware = lpDesc == lpHWDesc ? TRUE : FALSE;
    mono = lpDesc->dcmColorModel & D3DCOLOR_MONO ? TRUE :
FALSE;
    } else if ((lpDesc == lpHWDesc && hardware) ||
        (lpDesc == lpHELDesc && !hardware)) {
    if (lpDesc->dcmColorModel == D3DCOLOR_MONO && !mono) {

```

```

// の RGB
//

*lpStartDriver = myglobs.NumDrivers;
hardware = lpDesc == lpHWDesc ? TRUE : FALSE;
mono = lpDesc->dcmColorModel & D3DCOLOR_MONO ? TRUE :
FALSE;
}
}
myglobs.NumDrivers++;
if (myglobs.NumDrivers == MAX_DRIVERS)
return (D3DENUMRET_CANCEL);
return (D3DENUMRET_OK);
}

```

BPPToDDBD

```

enumDeviceFunc          BPPToDDBD
の                      enumDeviceFunc
                      enumDeviceFunc

////////////////////////////////////
//
// BPPToDDBD
// の DirectDraw の
//
////////////////////////////////////

static DWORD
BPPToDDBD(int bpp)
{
switch (bpp) {
case 1:
return DDBD_1;
case 2:
return DDBD_2;
case 4:
return DDBD_4;
}
}

```











```

{
    LPDIRECT3DRMFRAME lpLightframe1 = NULL;
    LPDIRECT3DRMFRAME lpWorld_frame = NULL;
    LPDIRECT3DRMLIGHT lpLight1 = NULL;
    LPDIRECT3DRMLIGHT lpLight2 = NULL;
    LPDIRECT3DRMTEXTURE lpTex = NULL;
    LPDIRECT3DRMWRAP lpWrap = NULL;
    LPDIRECT3DRMMESHBUILDER lpSphere3_builder = NULL;

    MakeMyFrames(lpScene, lpCamera, &lpLightframe1, &lpWorld_frame);
    MakeMyLights(lpScene, lpCamera, lpLightframe1, &lpLight1,
        &lpLight2);
    SetMyPositions(lpScene, lpCamera, lpLightframe1, lpWorld_frame);
    MakeMyMesh(&lpSphere3_builder);
    MakeMyWrap(lpSphere3_builder, &lpWrap);
    AddMyTexture(lpSphere3_builder, &lpTex);

    // fffff,00,,*-,,,000,,,,0E,<,-,00,,0000
    // ,,CreateMaterial,SetMaterial,E,0,0

    // ,,fffff0fffff,00,,,,,0f0ff0ff0f,'%,,0

    lpWorld_frame->lpVtbl->AddVisual(lpWorld_frame,
        (LPDIRECT3DRMVISUAL) lpSphere3_builder);

    lpLightframe1->lpVtbl->Release(lpLightframe1);
    lpWorld_frame->lpVtbl->Release(lpWorld_frame);
    lpSphere3_builder->lpVtbl->Release(lpSphere3_builder);
    lpLight1->lpVtbl->Release(lpLight1);
    lpLight2->lpVtbl->Release(lpLight2);
    lpTex->lpVtbl->Release(lpTex);
    lpWrap->lpVtbl->Release(lpWrap);

    return TRUE;
}

MakeMyFramesŠ
MySceneŠ,MakeMyFramesŠ,E,0,0Helworld.c,-
,, ,ffffffffff0Efff,f0ff0ff0f,00,,0MakeMyFramesŠ,0IDirect3DRM::Create
Frameffff,E,0,,0,0-ž0,,0
////////////////////////////////////
//
// MakeMyFrames
// f0f,ž-,,fff,00,,0
//
////////////////////////////////////

void MakeMyFrames(LPDIRECT3DRMFRAME lpScene, LPDIRECT3DRMFRAME
lpCamera,

```



```

,CE, ,,,,Ž ,, IDirect3DRMFrame::SetRotationfff, <','%o
,,,fff f,%o“ , ’,,
////////////////////////////////////
//
// SetMyPositions
// ∅
//
//
////////////////////////////////////

void SetMyPositions(LPDIRECT3DRMFRAME lpScene,
    LPDIRECT3DRMFRAME lpCamera, LPDIRECT3DRMFRAME
lpLightFrame1,
    LPDIRECT3DRMFRAME lpWorld_frame)
{

lpLightFrame1->lpVtbl->SetPosition(lpLightFrame1, lpScene,
    D3DVAL(2), D3DVAL(0.0), D3DVAL(22));

lpCamera->lpVtbl->SetPosition(lpCamera, lpScene,
    D3DVAL(0.0), D3DVAL(0.0), D3DVAL(0.0));
lpCamera->lpVtbl->SetOrientation(lpCamera, lpScene,
    D3DVAL(0.0), D3DVAL(0.0), D3DVAL(1),
    D3DVAL(0.0), D3DVAL(1), D3DVAL(0.0));

lpWorld_frame->lpVtbl->SetPosition(lpWorld_frame, lpScene,
    D3DVAL(0.0), D3DVAL(0.0), D3DVAL(15));
lpWorld_frame->lpVtbl->SetOrientation(lpWorld_frame, lpScene,
    D3DVAL(0.0), D3DVAL(0.0), D3DVAL(1),
    D3DVAL(0.0), D3DVAL(1), D3DVAL(0.0));

lpWorld_frame->lpVtbl->SetRotation(lpWorld_frame, lpScene,
    D3DVAL(0.0), D3DVAL(0.1), D3DVAL(0.0), D3DVAL(0.05));
}

```







IDirect3D

COM

IDirect3DDevice

IDirect3DTexture

DirectDraw

IDirect3DMaterial

IDirect3DLight

IDirect3DViewport

*IDirect3DDevice*

4 4 の

□□ ***ffff***

**IDirect3DExecuteBuffer***ffff*□*ffff*

”***f***□***f,fff,ffffff***•–,ŽŽ

*fff*

の ∞ COM

- *COM*
- COM の の の COM の
- の

,, Direct3D,Šff  
ffff,‘ ,,ŒŽ,Ž,  
”

COMfff ffff fffff

“

ffff

ffff

fffff

ŒŒ

ff f f

Textureffffff

Materialffffff

—

Lightffffff

Ž ffff

Viewportffffff

Direct3DffffffŒ

,,ffff,, Direct3

D,ffffffŒ,fff

ffff,,, -

,, ffff fff, Ž

,‘,,, , Direct3

Dffffff,fff ff

ff, ,,,,,,

Direct3Dfff fff

f fffff

Devicefffff

Execute-bufferfffff

IDirect3Dfff ffff

IDirect3Dfff ffff, DirectDrawfffff  
,,fff ffff,, IDirect3Dfff ffff,  
,,, Ź,,,IDirectDraw2::QueryInterface  
fff,Œ, ,, ,

lpDirectDraw->QueryInterface(  
IID\_IDirect3D, //  
IDirect3Dfff ffff,ID  
lpD3D); //  
Direct3Dfffff,ffff

IDirect3Dfff ffff,,,Ź ,,fffff,  
ff f f ŒŒ ffff ,,ffff,fff  
,Š,,, IDirect3D,ffff, ‘fffff,  
,, Direct3Dffff,Œ,,,,,Ź—  
,,,,,,

IDirect3DDevicefff ffff

IDirect3DDevicefff ffff, DirectDr  
aw

Surface のIDirect3DDevice の  
IDirectDrawSurface2::QueryInterface のIDirectDraw::CreateSurface  
IDirectDrawSurface::GetAttachedSurface  
IDirectDraw2 IDirectDrawSurface2 のQueryInterface

```

lpDirectDraw->CreateSurface(
    lpDDSurfDesc, // DDSURFACEDESC の
    lpFrontBuffer, // DIRECTDRAWSURFACE の
    pUnkOuter); // NULL
lpFrontBuffer->GetAttachedSurface(
    &ddscaps, // DDSCAPS “,ffff
    &lpBackBuffer); // DIRECTDRAWSURFACE “,ffff
lpBackBuffer->QueryInterface(
    GUIDforID3DDevice, // IDirect3DDevicefff ffff,ID
    lpD3DDevice); // DIRECT3DDEVICEffffff,,ffff

```

```

ffffff,,IDirectDrawSurface::QueryInterfacefff,☉,□,Ž’,, ,fff□f,□I
Direct3DDevicefff□ffff,Ž,□ff□ff□fff,—
^,Ž•Ž□GUID□,,□,,GUID,□IDirect3D::EnumDevicesffff,☉,□,,,,,Ž“,,,,,,,
fff IDirect3D::EnumDevices のの D3DENUMDEVICESCALLBACK

```

```

ffŠ ,☉, ,, ,,
GUID,◀,,
Direct3D,ffff f
ffff, fff fff
f fff;‘ ,,,, ‘
, Ž ,Ž ffff
,ff f f,fff,Ž,,
, ,, fffff,ffff
f,fff,Š,,, ,,,,ff
f’,Ž,ffff,ffff,
,ffff, ‘ffff,,ff
ff,—
•,Ž,,, ,,Š‘ ‘,,,
,,, fffff, ‘
,

```

```

Ž□,,,,□
IDirect3DDevicefff□ffff,ff ののの

```

```

Direct3D の IDirect3DDevice::CreateMatrix
IDirect3DDevice::SetMatrix
ffff,Ž ffff,—,,,

```

```

 IDirect3DTexture ffff ffff
 fffff, *,ffff,<E,*,,,,,, <E,*,,, •E,,,,
 ,,, ffff, ,E
 —““ ,,, •E, ,,, fffff,ffffff ffff
 f,-
 ,,Ž,,,,,, , , •E,,,,,, RGBff ff
 f,—,ffff fff,, 8 24 32fff,ffffff,Ž—
 ,,,,,, ffff fff ff fff,, 8fff,ffff
 f,,Ž
 
```

```

 ,” □
 IDirect3DTexture fff □ ffff, □ DirectDrawSurface
 IDirect3DTexture
 IDirectDrawSurface2::QueryInterface IID_IDirect3DTexture
 Direct3D の DirectDraw の Direct3D
 
```

```

 IDirect3DTexture □□□□□□□□□□□□—
 , □ IDirect3DTexture::GetHandle,,, IDirect3DTexture::Loadffff,—
 ,, fffff, f□f,, •-, Ž,,,, □
 lpDDS->QueryInterface(IID_IDirect3DTexture,
 lpD3DTexture); // DIRECT3DTEXTUREffffff,, ffff
 lpD3DTexture->GetHandle(
 lpD3DDevice, // DIRECT3DDEVICE
 lpTexture); // D3DTEXTUREHANDLE の
 lpD3DTexture->Load(
 lpD3DTexture); // DIRECT3DTEXTURE の
 
```

```

 ¥ののののの
 ,ffff,Ž□ffff,—,, □ff□f□f,,Zffff,,Ž,, □ffff,f□ のの
 IDirect3DTexture
 
```

のDirect3D のの

- 
- 
- 
-



D3DTEXTUREFILTER の  
の  
**D3DPRIMCAPS** の **dwTextureFilterCaps**

DirectDraw の

D3DRENDERSTATE\_BLENDEENABLE  
D3DRENDERSTATETYPE の

の **DirectDraw**

の

DirectDraw の

## **IDirect3DMaterial** の **ff**

IDirect3DMaterial **ff** の **ff**,  
IDirect3DMaterial の IDirect3DMaterial::SetMaterial  
IDirect3DMaterial::GetHandle の

lpDirect3D->CreateMaterial(  
lpDirect3DMaterial, //  
pUnkOuter); // NULL

lpDirect3DMaterial->SetMaterial(  
lpD3DMat); // D3DMATERIAL の

lpDirect3DMaterial->GetHandle(  
lpD3DDevice, // DIRECT3DDEVICE の  
lpD3DMat); // D3DMATERIAL の

の

の

の

IDirect3DMaterial の

IDirect3DLight

IDirect3Dlight の IDirect3D::CreateLight  
の IDirect3Dlight の IDirect3DLight::SetLight



pUnkOuter); // NULL

Ž□ffff,ffff□fff,Š,,,,,□Ž□,ffff—  
^,Š•,,,,□IDirect3DDevice::CreateExecuteBuffer の

IDirect3DExecuteBuffer::Lock  
IDirect3DExecuteBuffer::Unlock IDirect3DExecuteBuffer::SetExecuteData  
の

lpD3DExBuf->Lock(  
lpDesc);. // DIRECT3DEXECUTEBUFFERDESC の

// .  
// . Store contents through the supplied address

// .

lpD3DExBuf->Unlock();

// .

lpD3DExBuf->SetExecuteData(  
lpData); // D3DEXECUTEDATA の

のの IDirect3DExecuteBuffer::SetExecuteData ののDirect3D  
IDirect3DExecuteBuffer::Lock  
IDirect3DExecuteBuffer  
の

のの IDirect3DExecuteBuffer::SetExecuteData ののDirect3D  
IDirect3DExecuteBuffer::Lock

IDirect3DExecuteBuffer

の

ff f f, —  
,,,, ,,,,,ff f f,,,,,ŠŒŒ,•Ž,,,,,Ž,,,,,  
§  
ffffff,ffffff,ffffff, ,,^ ,ffff,Š~•,,,,,, Ž, ,Ž,,,Šf f, fff, “,ffff,•  
Ž,,, ,, ,’,ffff,•Ž,,,,, ,,ffffff,f f,,, ,,ffffff,f f,,, ,,ffffff,  
“,Š~,,ffff,ffff,Ž“,,,,,  
§  
fff ffff,ffff fffff,ffffff, ,,QueryInterfaceffff,Œ, , fffff Œ  
Œ ff f f,ffffff, IDirect3Dfff ffff,ffff,Œ, ,, ,,,,,, Ž fff  
f, —  
, IDirect3DDevicefff ffff,,, ,, fff ffff fffffff Direct3D  
object ,, ,,,,,,ffff,ff f f ŒŒ fffff,ffffff,fff,•Ž,,, Ž ff  
ff,ffffff,ffffff,Š,,,  
f f,Š—  
,,,’ f f ffff fff, fffffff,Ž ,,’,IDirect3DDevice::BeginSceneffff,Œ,  
,,,,,, ,, fffffff,Š—,,,,, IDirect3DDevice::EndSceneffff,Œ, ,•—  
,,, ffff fff, Œ,3Df ffff “ ,,,,,, ,,ffff, ,,Ž—,,,,,,

```
f f,"•%ffffff ŽŠĀ ' " , ' , , " , "%o,fff fff,,,,ff f,%o
,,,,Ž—
,,, IDirect3DDevice::BeginSceneffff,f f,Ž,,, ,,,,ff f,ŠŽ, Ž, IDirect3DDevice::EndSceneffff,f f,ff f, —,' , , , , , ff f,•%o
,Ž,,,Ž ffff—
, 1“,IDirect3DDevice::BeginScene,IDirect3DDevice::EndScene,Ā, ,Š,,,
,,,,,,
,,fffff,^%o,ffff,,,, -,,,,
• %o- <
• 2D,3D,‘Ā —
• f fŠ—fff ff
```

```
%o- <
3Dfffff□f,” fff,,□□—,Zfff
IDirect3DDevice::BeginScene
IDirect3DDevice::EndScene
```

```
ののの
の
ののののの
1 の IDirect3DDevice::BeginScene IDirect3DDevice::EndScene の
,,,,,,□,,,ff□f,□□',□•□%o,IDirect3DDevice::BeginScene
IDirect3DDevice::EndScene の
```

```
のの
IDirect3DDevice::BeginScene IDirect3DDevice::EndScene
```

**IDirect3DDevice::BeginScene IDirect3DDevice::EndScene**

**2D 3D の**

```
IDirect3DDevice::BeginScene
IDirect3DDevice::EndScene 3D のの 2D ののDirectDraw の
DDCAPS2_NO2DDURING3DSCENE のDirectDraw のGetDC の
IDirect3DDevice::EndScene の
```

のの 3D ののの

```
1 のの1 IDirect3DDevice::BeginScene
Direct3DDevice::EndScene
```

**2 IDirect3DDevice::BeginScene の IDirect3DDevice::EndScene の DirectDraw の GetDC の 2D の IDirect3DDevice::EndScene**

```

3 IDirect3DDevice::Executeffff, Ɛ,-
,,, ,,Ž ffff,Š,,,ffffff fffffff,fff f f fff,ffffff,Ɛ,,,,,
f f fffff f f, f f,Š—
,,, ,, IDirect3DDevice::EndSceneffff,Ɛ, ,, fffff,%Š,

```

Direct3D, f f,ff ffff  
 Windows f f,Direct3D, f f ffff fff, Ɛ, ,, ,DirectDraw Direct3D  
 の

SDK の

の SDK の D3dmain.cpp D3dmain.cpp Windows の の の の の  
 Direct3D の の の

- Step 1:
- Step 2: DirectDraw Direct3D の
- Step 3: の
- Step 4: の
- Step 5: の の
- Step 6:
- Step 7: の
- Step 8:

の  
 SDK,,' f f,ffff,, ,,< ,,f f,,,Š,,, “,,SDK,, D3DappŠ ,,  
 Ɛ, ,,fffŠ ,ffffff,Š,,, “Ž,’ f f,ffff fff,< ,, ,—  
 ,, ,,ffffff,,fffŠ ,•”Ž ,, ,, ,,ff ,, ,, fffŠ , f f ff  
 ff D3dapp.c Ddcalls.c D3dcalls.c Texture.c ,, Misc.c  
 ,ffffff, ,, f f ffff Stats.cpp  
 , ff f f f,fff f f f, •,fff f,‘

D3main.cpp ,—,,,ffff, ,, “ ,ffffff,,,,,, ~%Š ,ffffff,,,,,,

- InitScene
- InitView
- RenderScene
- ReleaseView

- ReleaseScene
- OverrideDefaults

,,, ffff,SetMouseCallbackŠ ,SetKeyboardCallbackŠ ,E, , fff,f f f,,,  
 “—,Ž“,,

Step 1: Š%o,ŠŽ

D3dmain.cpp,WinMainŠ□,□,□,fff,□f□ff,‘<,,,AppInitŠ□,E,□,□ffff□ff  
 f□ffff,□□,□fffff,ŠŽ,•—,,,,fffff,□Š%o  
 ,,□,,□WinMainŠ□,D3dmain.cpp,fff□f□fff,fffff,□f□ff’,RenderLoopŠ  
 □,CleanUpAndPostQuit AppInit ののののののの

Windows ののAppInit InitScene 3Dmain.cpp  
 InitScene Oct1.c のInitScene  
 TRUE Tunnel.c

InitScene

AppInit D3dmain.cpp CreateD3DApp  
 CreateD3DApp

Step 2: DirectDraw Direct3D の

D3dmain.cpp CreateD3DApp DirectDraw  
 Direct3D CreateD3DApp  
 のの D3DAppCreateFromHWND  
 D3DAppGetRenderState OverrideDefaults D3DAppSetRenderState  
 ReleaseView InitView の D3DApp の  
 D3DApp のの ,,

WinMainŠ “,,,ffff fff fffff“,,,, CreateD3DAppŠ ,,,,, —  
 E,fffff, -systemmemory,-emulation,,, -systemmemoryfffff, ffff  
 —,—,,, -  
 emulationfffff,Ž’,,, ffff fff,DirectDraw,Direct3D,f ffff fffff f,  
 Ž—,,,

CreateD3DAppŠ□,□ffff,□□,,,,,D3DAppAddTextureŠ□,E,□,□,,□D3DA  
 ppAddTextureŠ□,□f□f□fffff,f□fff,fffff,ffff□fff“, ,, f□ffff,ff  
 f□fff,Ž,,,□□,□□Š□’,<,fffff□f□fff,fff□fff“, ,, f□f□ffff,□“  
 ,fffff□f□fff,f□f,,E,”Š,,,□,2’Š,ffff,,,□ffff,□fff□fff“,,,,,,ffff  
 f□fff,^□,,□ff□fff,,,,,,□ffff□f□f,,□IDirect3DTexturefff□ffff,Ž,,,  
 ,,IDirectDrawSurface::QueryInterfacefff,E,□,□IDirect3DTexture::Load  
 のIDirect3DTexture::GetHandle

のCreateD3DApp の DirectDraw  
 Direct3Dfffff,□□,,□fff□f□f,,□D3DAppCreateFromHWND  
 D3DAppCreateFromHWND D3dapp.c  
 D3dcalls.c Texture.c Ddcalls.c

D3DAppCreateFromHWND DirectDrawEnumerate  
 DirectDrawCreate DirectDraw のの  
 IDirectDraw::EnumDisplayModes

IDirectDraw の IDirectDraw2 の IDirectDraw IDirectDraw  
IDirectDraw::EnumDisplayModes の  
IDirectDraw2::EnumDisplayModes

D3DAppCreateFromHWND Direct3D Direct3D  
Direct3D の IID\_IDirect3D の

IDirectDraw::QueryInterface  
IDirect3D::EnumDevices

IDirect3D::EnumDevices の Direct3D IDirect3D::FindDevice  
の

GUID

の

の

GUID

D3DAppCreateFromHWND の の の の IDirectDraw::CreateClipper

DirectDrawClipper ffffff, □□, □ IDirectDrawClipper::SetHWND ffffff, ffffff  
ffffffffff, S~, □ IDirectDrawSurface::SetClipper

D3DAppCreateFromHWND

の

IDirectDraw::CreatePalette

IDirectDrawSurface::SetPalette

の

IDirectDraw::CreateSurface の Z

IDirectDrawSurface::AddAttachedSurface Z

Z

IDirectDrawSurface::GetSurfaceDesc

IDirect3DDevice

IDirectDrawSurface::QueryInterface

IDirect3DDevice::EnumTextureFormats

の

Created3DApp

の

の

Step 3: の

の

D3DAppCreateFromHWND の Step 5: の の

Direct3D

の

D3DAppCreateFromHWND TRUE

D3DAppCreateFromHWND の の Direct3D

DirectDraw

FALSE

Step 3: の

D3DAppCreateFromHWND の 3 AfterDeviceCreated  
 D3dmain.cpp  
 AfterDeviceCreated Direct3D  
 D3DAppCreateFromHWND

**IDirect3D::CreateViewport**  
**IDirect3DDevice::AddViewport** Direct3D  
**D3DVIEWPORT** の ののの  
 IDirect3DViewport::SetViewport の

**AfterDeviceCreated** **InitView** **InitView**  
 D3dmain.cpp の InitScene D3dmain.cpp  
 InitView ののStep 4: の

InitView ののののCleanUpAndPostQuit AfterDeviceCreated  
 CleanUpAndPostQuit Step 8:

Step 4: の

D3dmain.cpp の

Oct1.c InitView の  
 InitView の

InitView **IDirect3D::CreateMaterial**  
**IDirect3DMaterial::SetMaterial**  
 IDirect3DMaterial::GetHandle IDirect3DViewport::SetBackground

**InitView** ののの  
 InitView MAKE\_MATRIX  
 MAKE\_MATRIX D3dmacs.h の

```
#define MAKE_MATRIX(lpDev, handle, data) \
    if (lpDev->lpVtbl->CreateMatrix(lpDev, &handle) != D3D_OK) \
        return FALSE; \
    if (lpDev->lpVtbl->SetMatrix(lpDev, handle, &data) != D3D_OK) \
        return FALSE
```

MAKE\_MATRIX IDirect3DDevice::CreateMatrix  
 IDirect3DDevice::SetMatrix

**InitView**  
 D3DEXECUTEBUFFERDESC の**IDirect3DDevice::CreateExecuteBuffer**  
 IDirect3DExecuteBuffer::Lock

InitView D3dmacs.h OP\_STATE\_TRANSFORM  
STATE\_DATA Step 5: ののののの

### InitView

**IDirect3DExecuteBuffer::Unlock**  
IDirect3DExecuteBuffer::SetExecuteData  
**IDirect3DDevice::BeginScene** IDirect3DDevice::Execute  
**IDirect3DDevice::EndScene**  
IDirect3DExecuteBuffer::Release

InitView IDirect3D::CreateMaterial **D3DMATERIAL** の  
IDirect3DMaterial::SetMaterial  
IDirect3DMaterial::GetHandle のD3DLIGHTSTATETYPE の  
D3DLIGHTSTATE\_MATERIAL

InitView のD3DVERTEX の D3DVALUE  
D3DVALP x  
D3DRMVectorNormalize

のInitView のののの

InitView Oct1.c のD3DLIGHT  
IDirect3D::CreateLight IDirect3DLight::SetLight  
IDirect3DViewport::AddLight

Step 5: のの

D3dcalls.c D3DAppISetRenderState の  
のの  
D3DAppCreateFromHWND D3dapp.c **D3DAppISetRenderState**

D3DAppISetRenderState のの  
D3DAppISetRenderState

**D3DAppISetRenderState** **D3DEXECUTEBUFFERDESC**  
D3DEXECUTEDATA の

**IDirect3DDevice::CreateExecuteBuffer**  
IDirect3DExecuteBuffer::Lock

BOOL D3DAppISetRenderState()

```
{  
D3DEXECUTEBUFFERDESC debDesc;  
D3DEXECUTEDATA d3dExData;  
LPDIRECT3DEXECUTEBUFFER lpD3DExCmdBuf = NULL;  
LPVOID lpBuffer, lpInsStart;  
size_t size;
```

```

//

size = 0;
size += sizeof(D3DINSTRUCTION) * 3;
size += sizeof(D3DSTATE) * 17;
memset(&debDesc, 0, sizeof(D3DEXECUTEBUFFERDESC));
debDesc.dwSize = sizeof(D3DEXECUTEBUFFERDESC);
debDesc.dwFlags = D3DDEB_BUFSIZE;
debDesc.dwBufferSize = size;

LastError = d3dappi.lpD3DDevice->lpVtbl->CreateExecuteBuffer(
    d3dappi.lpD3DDevice, &debDesc, &lpD3DExCmdBuf, NULL);

LastError = lpD3DExCmdBuf->lpVtbl->Lock(lpD3DExCmdBuf, &debDesc);
memset(debDesc.lpData, 0, size);

lpInsStart = debDesc.lpData;
lpBuffer = lpInsStart;

```

**IDirect3DDevice::CreateExecuteBuffer** の `d3dappi.lpD3DDevice`  
*Direct3DDevice* の `debDesc`      `D3DEXECUTEBUFFERDESC` の  
`lpData`

`D3DAppISetRenderState` の  
**D3DAppISetRenderState**      **OP\_STATE\_DATA**      の  
`PUTD3DINSTRUCTION` の SDK `D3dmacs.h` の

```

#define PUTD3DINSTRUCTION(op, sz, cnt, ptr) \
    ((LPD3DINSTRUCTION) ptr)->bOpcode = op; \
    ((LPD3DINSTRUCTION) ptr)->bSize = sz; \
    ((LPD3DINSTRUCTION) ptr)->wCount = cnt; \
    ptr = (void *)(((LPD3DINSTRUCTION) ptr) + 1)
#define OP_STATE_RENDER(cnt, ptr) \

```

```
PUTD3DINSTRUCTION(D3DOP_STATERENDER, sizeof(D3DSTATE),
cnt, ptr)
```

```
PUTD3DINSTRUCTION の D3DINSTRUCTION の
OP_STATE_RENDER の PUTD3DINSTRUCTION の 1
D3DOP_STATERENDER D3DOPCODE の 2 の
D3DRENDERSTATETYPE の D3DSTATE の
```

```
D3dmacs.h STATE_DATA
の D3DSTATE の D3DRENDERSTATETYPE
の
```

```
#define STATE_DATA(type, arg, ptr) \
((LPD3DSTATE) ptr)->drstRenderStateType =
(D3DRENDERSTATETYPE)type; \
((LPD3DSTATE) ptr)->dwArg[0] = arg; \
ptr = (void *)((LPD3DSTATE) ptr) + 1
```

```
D3DAppISetRenderState の OP_STATE_RENDER
STATE_DATA 14
d3dapprs D3dapp.h D3DAppRenderState
```

```
OP_STATE_RENDER(14, lpBuffer);
STATE_DATA(D3DRENDERSTATE_SHADEMODE,
d3dapprs.ShadeMode, lpBuffer);
STATE_DATA(D3DRENDERSTATE_TEXTUREPERSPECTIVE,
d3dapprs.bPerspCorrect, lpBuffer);
STATE_DATA(D3DRENDERSTATE_ZENABLE, d3dapprs.bZBufferOn
&&
d3dappi.ThisDriver.bDoesZBuffer, lpBuffer);
STATE_DATA(D3DRENDERSTATE_ZWRITEENABLE,
d3dapprs.bZBufferOn,
lpBuffer);
STATE_DATA(D3DRENDERSTATE_ZFUNC, D3DCMP_LESSEQUAL,
lpBuffer);
STATE_DATA(D3DRENDERSTATE_TEXTUREMAG,
d3dapprs.TextureFilter,
lpBuffer);
```

```

STATE_DATA(D3DRENDERSTATE_TEXTUREMIN,
d3dapprs.TextureFilter,
    lpBuffer);

STATE_DATA(D3DRENDERSTATE_TEXTUREMAPBLEND,
d3dapprs.TextureBlend,
    lpBuffer);

STATE_DATA(D3DRENDERSTATE_FILLMODE, d3dapprs.FillMode,
lpBuffer);

STATE_DATA(D3DRENDERSTATE_DITHERENABLE,
d3dapprs.bDithering,
    lpBuffer);

STATE_DATA(D3DRENDERSTATE_SPECULARENABLE,
d3dapprs.bSpecular,
    lpBuffer);

STATE_DATA(D3DRENDERSTATE_ANTIALIAS,
d3dapprs.bAntialiasing,
    lpBuffer);

STATE_DATA(D3DRENDERSTATE_FOGENABLE,
d3dapprs.bFogEnabled,
    lpBuffer);

STATE_DATA(D3DRENDERSTATE_FOGCOLOR, d3dapprs.FogColor,
lpBuffer);

OP_STATE_RENDER STATE_DATA OP_EXIT D3DOPCODE
D3DOP_EXIT PUTD3DINSTRUCTION

OP_STATE_LIGHT(3, lpBuffer);

STATE_DATA(D3DLIGHTSTATE_FOGMODE, d3dapprs.bFogEnabled ?
    d3dapprs.FogMode : D3DFOG_NONE, lpBuffer);

STATE_DATA(D3DLIGHTSTATE_FOGSTART,
    *(unsigned long*)&d3dapprs.FogStart, lpBuffer);

STATE_DATA(D3DLIGHTSTATE_FOGEND, *(unsigned
long*)&d3dapprs.FogEnd,
    lpBuffer);

```

```
OP_EXIT(lpBuffer);
```

```
D3DAppISetRenderState
```

```
IDirect3DExecuteBuffer::Unlock  
    IDirect3DExecuteBuffer::SetExecuteData  
    IDirect3DDevice::BeginScene IDirect3DDevice::Execute  
IDirect3DDevice::EndScene
```

```
LastError = lpD3DExCmdBuf->lpVtbl->Unlock(lpD3DExCmdBuf);
```

```
memset(&d3dExData, 0, sizeof(D3DEXECUTEDATA));
```

```
d3dExData.dwSize = sizeof(D3DEXECUTEDATA);
```

```
d3dExData.dwInstructionOffset = (ULONG) 0;
```

```
d3dExData.dwInstructionLength = (ULONG) ((char*)lpBuffer -  
    (char*)lpInsStart);
```

```
lpD3DExCmdBuf->lpVtbl->SetExecuteData(lpD3DExCmdBuf, &d3dExData);
```

```
LastError =
```

```
    d3dappi.lpD3DDevice->lpVtbl->BeginScene(d3dappi.lpD3DDevice);
```

```
LastError =
```

```
    d3dappi.lpD3DDevice->lpVtbl->Execute(d3dappi.lpD3DDevice,  
    lpD3DExCmdBuf, d3dappi.lpD3DViewport);
```

```
LastError = d3dappi.lpD3DDevice->lpVtbl->EndScene(d3dappi.lpD3DDevice);
```

```
D3DAppISetRenderState
```

```
IDirect3DExecuteBuffer::Release
```

```
lpD3DExCmdBuf->lpVtbl->Release(lpD3DExCmdBuf);
```

```
return TRUE;
```

```
}
```