Autodesk®
Scaleform®

Scaleform ”A

╪ ᵘ ”A DirectX 9 ♠ ╬ Scaleform Ọ 3D ε ∫ ˘ ▓ .

: Ben Mowery
♣: 3.04
↓ ∨ : 2011 Ц 8 4 ■

Autodesk®
GAMEWARE

Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 123D, 3ds Max, Algor, Alias, AliasStudio, ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Homestyler, Autodesk Intent, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSketch, AutoSnap, AutoTrack, Backburner, Backdraft, Beast, Beast (design/logo) Built with ObjectARX (design/logo), Burn, Buzzsaw, CAiCE, CFdesign, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Creative Bridge, Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design Web Format, Discreet, DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWFX, DXF, Ecotect, Evolver, Exposure, Extending the Design Team, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, Freewheel, GDX Driver, Green Building Studio, Heads-up Design, Heidi, Homestyler, HumanIK, i-drop, ImageModeler, iMOUT, Incinerator, Inferno, Instructables, Instructables (stylized robot design/logo),Inventor, Inventor LT, Kynapse, Kynogon, LandXplorer, Lustre, MatchMover, Maya, Mechanical Desktop, MIMI, Moldflow, Moldflow Plastics Advisers, Moldflow Plastics Insight, Moondust, MotionBuilder, Movimento, MPA, MPA (design/logo), MPI (design/logo), MPX, MPX (design/logo), Mudbox, Multi-Master Editing, Navisworks, ObjectARX, ObjectDBX, Opticore, Pipeplus, Pixlr, Pixlr-o-matic, PolarSnap, Powered with Autodesk Technology, Productstream, ProMaterials, RasterDWG, RealDWG, Real-time Roto, Recognize, Render Queue, Retimer, Reveal, Revit, RiverCAD, Robot, Scaleform, Scaleform GFx, Showcase, Show Me, ShowMotion, SketchBook, Smoke, Softimage, Sparks, SteeringWheels, Stitcher, Stone, StormNET, Tinkerbox, ToolClip, Topobase, Toxik, TrustedDWG, T-Splines, U-Vis, ViewCube, Visual, Visual LISP, Vtour, WaterNetworks, Wire, Wiretap, WiretapCentral, XSI.

All other brand names, product names or trademarks belong to their respective holders.

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. “AS IS.” AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR

IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

ᴻ :	
ᵘ ᵀA	Scaleform 4.2 ᵀA
ᴸ	Scaleform Corporation
	6305 Ivy Lane, Suite 310
	Greenbelt, MD 20770, USA
ˢ ╪	www.scaleform.com
╪ ʳ ■	info@scaleform.com
♣	(301) 446-3200
ᶜ	(301) 446-3199

1

2

3

3.1 ᾶ ᾳ 3

3.2 ᵖ ῖ ᾳ ■ 4

3.3 Scaleform Player ⁄ ᾶ SWF ᾶ Ỹ ᾳ 5

3.4 Ỵ ╪ ῖ ᾳ ■ 7

3.5 Scaleform й ╥ ⊢ ᾶ E 7

3.5.1 AS3_Global.h 파일과 Obj\AS3_Obj_Global.xxx 파일 9

4

4.1 11

4.2 w♥ ᵖ й 20

4.3 ▒ ╪ ⁄. ᾳ 22

4.3.1 22

4.3.2 24

4.3.3 25

4.3.4 26

5

5.1 ث ⁄. ¸ 27

6 C++, ,

6.1 ⁄. ⁄ ᾶ C++ 30

6.1.1 FS Command 30

6.1.2 ExternalInterface 33

6.2 C++ ⁄ ᾶ ⁄. 35

6.2.1 36

6.2.2 38

6.3 ر ■ 39

7

8 GFxExport

9

Scaleform ╪Ç ˀ ˘ ˝E ╗ Ơ ▫ ╬ ˁ ╪ (UI) Ÿ ▫ ╬ Ç ɵ ‡ ˀA
˛ ▫ ɵ ╪ Ơ ˀA ẖ ˘ ʍ ˨ GPU ʃ ╪ UI
Ự ˆ ↑ ˀE Ứ˘ ˆ ▓ . Scaleform
/ ˀA ˭ ▌ UI ꞟ Ứ˘ ˊ ˛ .

UI Ỏ ˽ ᾷ ˛ ˛ ▫ ɵ Scaleform Ơ 3D ᶜ ᶚ / ʃ ╪ ˨ ɬ ˀQ
3D ˽ Ɔ ╥ ᶜ ▓ ˭ ˝˝ . Ä 3 ˭ ▌ ᶚ ˝ 3D ɬ ⅞. Ũ ╥
UI Ỹ . ╪ Ơ ˭, 3D ɬ ⅞. Ÿ UI Ɔ / ˀA ᶜ ▓ .
˥ ˊ, Scaleform ■ UI Ỏ ˽ ᾷ ⊢ ╥ ˨ ˭ ▌ ▌ Ũ
Ł ˘ ▓ .

╪ Ԝ ╕ Scaleform ╥ ẽ ˨ Ơ / ˊ ♥ ˭ ˥ ╪ . ˝ ╥
UI Ơ DirectX ShadowVolume SDK Ỵ ˥ ╪ .

Scaleform ╪Ç ╗ ˭ ▌ ˊ ˝˝ . Scaleform ˨ ╥ ы
Scaleform ˭ ▌˘ ♪ Ơ ╪ . ╪ ᵘ ˀA ẻ ♠ Scaleform ■ ♠ ╬
˭ ▌ ˊ ˆ ↑ Scaleform ╥ / Ũ ╥ б ╥ ▫ ɵ
▬˝E ˝˝ .

╪ Ԝ ╕ / ▓ Ɔ ↻ Scaleform ╥ ˨ ♣ Ơ . ╪ Ԝ ╕
Scaleform ♣ 4 ╪Ũ / ♠ .

╪ ˀA ˆ Ÿ ˃ й Ứ . ╪ ˥ ˀA ˘ ▓ DirectX
SDK ShadowVolume й ↻ ᵘ ╪ . Ỵ й ▬ ẋ / ˃ ᶚ "Cannot create renderer" /
ʃ ˙ . ᵘ Ł / Scaleform Player ‡ ỉ ╪ ˝E ♠ ▬ ˨
▫

◗ ⎙ / … ‡ ▬ "E Scaleform ᵘ "A Scaleform ¸ ▫
(http://gameware.autodesk.com/scaleform/developer/) / "A ▓ . Ơ ╪ ⁄. ♪
∫ ╪ ▒ .

╪ ᵘ "A / ˢ Ɔ ' ⁻ .

- y Scaleform SDK ˘ ᵘ "A.
- y PDF ᵘ "A.
- y ت ⁄. ¸ : ت ⁄. ⁄. ε̈ ∫ ˘ ˜ Ł ω̣ ⁄. ' Scaleform C++ API ε̈ ♬ / ' "E Ơ Ł ̂
- y XML ¸ : SCALEFORM / "A XML ε̈ ∫
- y Scale9Grid: w ♥ ╪ , ق B ، Ỹ "E Scale9Grid ╥ Ơ ε̈ ∫
- y IME ε̈ ♬ : Scaleform ╥ IME ⎙ Ơ ▫ ‡ ǐ ╪ / ‡ ˚ ˝ ˘ / "A ■ ♠ ╬ IME ⁄. ▒ ■ ‡ ˚ ˝ Ỹ "E ε̈ ∫
- y ⁄. ◗ : Scaleform ⁄. ◗

◗ ৲ ╥ Windows Scaleform ╬ . ৷ ˘ ঃ ৲ ←
. ȝ Scaleform ╬ DirectX ‰ ╪ ⅞. ৷ ╪ . Scaleform
C:\Program Files\Scaleform\GFx SDK 4.1 Ÿ ⁄ ঃ ৲ .

╪ ॐ ’ - .

libjpeg ╪ ↑ zlib ৲ Scaleform ╪

╥ ‡ ї ╪ й ↳ Ц Ÿ ‰ \ ÀÍ¯ 3p À n ñFpÖ{ 1 • À R ’g¢ @

demo.sln ל¶ Ա Ÿ ł ⁄.. й ȝ Win32 Ÿ Э ╥
ر ■ ╪ ৷ .

Scaleform

Scaleform ╪

CLIK ї ˚ Б ⁄.. n

2 ◗ / ˝A ε ∫ PDF ᵘ ˝A

Scaleform й ╪ Ł ˆ ˜E ‡ ▒ ╬ ė [▬ Æ ᵖ ö
ÆScaleform ÆGFx SDK 4.2 ÆDemo MSVC 9.0 Solutions ÆGFx 4.2 Demos.sln] / ▒
ᵖ й . .sln ر ■ ╪ ל¶ Ա Ÿ / ˝A ȝ D3D9_Debug_Static ε ♬
ė ˘ Scaleform Player ł ⁄.. й

C:\Program Files\Scaleform\GFx SDK
4.2\Bin\Win32\Msvc90\GFxPlayer\GFxPlayer_D3D9_Debug_Static.exe ر ■ ╥ ♬ ■ ▫ ╬
Ł й ‴ ╬ .

’ [▬ Æ ᵖ ö ÆScaleform ÆGFx SDK 4.2 ÆDemo Examples ب] /
▒ Scaleform Player D3Dx ‡ ї ╪ й ‡ / ˝A SWF ╪ ‡ .
╪ n ╪ Scaleform ◓ Ỹ ⁄.. ˘ ৷ ▒

▬ Æ ᵖ ö ÆScaleform ÆGFx SDK 4.2 ÆGFx Players ÆGFx Player D3D9.

C:\Program Files\Scaleform\GFx SDK 4.2\Bin\Data\AS2\Samples\SWFToTexture ب

3D Windows.swf ‡ ĭ ╪ й .

Ỵ ǎ ʌ ╥ й ‡ ▃ Ỹ

F1 β / ' ¯ ⅞. ▫ .

1. ˜E ʌ ♬ CTRL+F ы . ▃ FPS Ł ᶜ ج Л / ↑ ϱ .
2. ╪ ‡ ▌ ᵖ й CTRL+W ы . ↓ ♠ й ‡ ▃ Ỹ
   ǎ ʌ Ù ‡ ˅ ˣ ╬ . ╪ ‡ ▌ ᵖ й
   ˍ CTRL+W ы ȝ .
3. ȝ ȝ CTRL ы ᾷ
   ╬ .
4. ╪ Ç ! ╪ ȝ CTRL ы
   ╪ .
5. ♬ Ũ ᾷ ȝ CTRL+Z ы .

6. ♣ ৷ ʒ ᵖ й ♣ CTRL+U ы .

7. ɾ ᵖ Ọ̓ Ȳ ᵉ ʃ ╥ ´ > ʒ F2 ы .

, ᵖ ̈A ৷ ˙ ė ʒ ᵐ ̈A ϟ Ọ̓ / ϟ ᶜ ▫ .
╥ ᵐ ˝ ▬ᵐ ʒ ǎ ᴎ╥ . ╥ ◗ ˩ ᴬ ↑
‡ Ʊ Ũ̓ ǎ ᴎ w ♥ ▓ ˩ ╪ . ♣ ♠╬ ⅞.. ₮
800x600 ⅞.. ₮ ' 1600x1200 ʒ ̈E⅞. .

╥ / Scaleform ˘ ▓ ৷ ৷ Scaleform ♣ ♠╬ ⅞.. ₮
. / ̈A ᵣ ɵ ▓ ‡ Ɛ ǎ ᴎ Scaleform

"3D Scaleform Logo" Ÿ , ˘ CTRL+W ╪ ‡ ▌ᵖ й ♣ .
╪ Ù̓ ‡ ▓ ৷ / ϟ ᶜ . CTRL+A ц ы ʒ Ȯ ö ‡
ᵖ й ♣ . Edge Anti-Aliasing ᵖ й (EdgeAA) / ̈A, ᴎ ̈A Ù̓ ᵖ ̈A /
ᴎ Ȯ ö ‡ ' Ỹ ̈E . ╪ Ÿ ˩ й AA╥ Л ╥ ▌
ᵐ Ÿ ɾ ᵖ Л ╥ ♣ ৷ ʒ Ȯ ö ‡
(FSAA) ' ♠╪ . ǐ ■ᵗ╥ ˘ ╬ EdgeAA Ȯ ö ‡ ■ ♠
/ ᵖ ̈A ⅞.. ⁻ ╪ ╪ ◗ ◗ ᵘ ╪ ▓ / ᵣ Ȯ ö ‡
♠ ₋ ৷ ╥ ⅞..̈E ╪ . Ù̓ ╥ ‡ ↑ draw primitive (DP)╥
■♬ ᵐ ৯ ᵘ / ̈E ' . ╪ ৷ Ÿ ˩ й╥ Ȯ ö ‡
Ọ̓ ৷ ■ ♠ ৯ ♠╪ ę EdgeAA ǰ ɛ̈ ♬ Ọ̓ ˅ ↑
Ȳ╥ w▬╪ .

Scaleform Player n ǎ ᴎ╥ Ÿ ' ̈E ᴎ / . ▬‰ ' ▫
CPU Ọ̓ ╬ . CPU Ọ̓ Scaleform ᴎ ᶜ ♠ ৷ ᵘ ╥
▌ ৯ ᵘ / и ᵐ ↑ ▓ . CTRL+Y ы ʒ ▌ ╪⅞.. ʒ
ᵢ ϟ (■ ♠ ৷ 60 ▌) ˘ ♬ . CPU Ọ̓ ╪ ˝ ᵐ ᶜ ╬
▓ .

▫ ╥ ‡ ǐ╪ ᴎ ৯ Scaleform Ÿ й ৷ ♠ ̈E Ɫ̂ Ω
৯ ᵘ / й ৷ .

Scaleform 4.2 Ơ ▫ Scaleform\GFx SDK 4.2\Apps\Tutorial ˬ / ▓ Tutorial Solution
. ▬ ʃ Ɔ Scaleform ÆGFx SDK 4.2 ÆTutorial / ˝A ㄴ˥ ԾŸ 2005
2008 ╬ ▓ . ǂ ⅞. ë ♬ ╪ “Debug” ‡ ▓ ╬
‡ ǐ ╪

UI ▓ ShadowVolume ‡ ǐ ╪

Ự Scaleform ǂ ⅞. Ỹ ˜E ২, ï ر ■ ♣ ㄴ˥ ԾŸ / ˝A ë ♬ ц
Ơ ╪ ▓ . ԾŸ / ۿ ╥ ´ ƒ ২ ˘ ▓ Ư ˬ ╪Ç ˘˘ ▓ .
$(GFXDSK) SDK ë ∧ Ÿ / ˝A ♬ ╥ ˬ ▌ ㄴ . ∧
C:\Program Files\Scaleform\GFx SDK 4.2\╪ . ˭ – ╪ ∧ / ▓ ȝ Ư /
▓ ╪ ╥ ∧ ♬ $(GFKSDK) ˬ ▓ . ╪ / ė
“Msvc90” Ơ . ㄴ˥ ԾŸ 2008 ˝ 2010 Ơ ▫ ╬ ˬ Msvc90 ˝
Msvc10 Ơ ▓ .

Scaleform ł ⅞.. ╥ Ÿ й ё♬ / ̈A include ← / ⅄ .

$(GFXSDK)\Src

$(GFXSDK)\Include

╥ Ɔ ↳ ↳ Ц Ÿ “Additional Include Directories” ᶜ / K .

ă … (GFx.h) ˚ ‡ ▒ Ώ ← , AS3 Ọ ˘ ▒ ȝ AS3

Ў ر ■ “GFx/AS3/AS3_Global.h” / ♪ ё ⅄ . ¸ ▫ ╪ ر ■ / ̈A

♪ AS3 Ł ▒ . ▫ ̈E Ɔ Scaleform LITE Ọ ▫ ♬ ╥

ᵘ ̈A w .

Ÿ й ё♬ ╪ Ÿ ă ˢ ṵ ← / ⅄

$(DXSDK_DIR)\Lib\x86

$(GFXSDK)\3rdParty\expat-2.1.0\lib

$(GFXSDK)\Lib\$(PlatformName) \Msvc80\Debug_Static\

$(GFXSDK)\3rdParty\zlib-1.2.7\Lib\$(PlatformName) \Msvc90\Debug

$(GFXSDK)\3rdParty\jpeg-8d\Lib\$(PlatformName) \Msvc90\Debug

Ɔ “Additional Library Directories” ᶜ / K .

"$(DXSDK_DIR)\Lib\x86";

"$(GFXSDK)\3rdParty\expat-2.1.0\lib";

"$(GFXSDK)\Lib\$(PlatformName)\Msvc90\debug";

"$(GFXSDK)\3rdParty\zlib-1.2.7\Lib\$(PlatformName)\Msvc90\Debug";

"$(GFXSDK)\3rdParty\jpeg-8d\Lib\$(PlatformName)\Msvc90\Debug"

Msvc90 ё ⅄ ↳ Ц Ÿ ♣ / ♠ ᵘ ▫ .

й ё♬ / ̈A ’ ╪ ă ˢ ṵ ← / ⅄ – :

$(DXSDK_DIR)\Lib\x86

$(GFXSDK)\3rdParty\expat-2.1.0\lib

$(GFXSDK)\Lib\$(PlatformName)\Msvc90\Release\

$(GFXSDK)\3rdParty\zlib-1.2.7\Lib\$(PlatformName)\Msvc90\Release

$(GFXSDK)\3rdParty\jpeg-8d\Lib\$(PlatformName)\Msvc90\Release

Ɔ “Additional Library Directories” ᶜ / K (ر ■ ё♬).

"$(DXSDK_DIR)\Lib\x86";
"$(GFXSDK)\3rdParty\expat-2.1.0\lib";
"$(GFXSDK)\Lib\$(PlatformName)\Msvc90\Release";
"$(GFXSDK)\3rdParty\zlib-1.2.7\Lib\$(PlatformName)\Msvc90\Release";
"$(GFXSDK)\3rdParty\jpeg-8d\Lib\$(PlatformName)\Msvc90\Release"
"$(GFXSDK)3rdParty\libpng\Lib\$(PlatformName)\Msvc90\Release"

Msvc90 ß ㅅ ¶ ㄴ Ÿ ♣ / ♠ ᴗ ▫ .

¡ Scaleform ╪ ╥ ⊢ ¨E ㄱ .

libgfx.lib
libAS2.lib
libAS3.lib
libgfxexpat.lib
libjpeg.lib
zlib.lib
libpng.lib
libgfxrender_d3d9.lib
libgfxsound_fmod.lib

■ ╪ ㄹ ■ й ˆ ¨E / ㄱ .

Ÿ ß ♬ ᵖ / ¨A Y ‡ ǐ ╪ ǐ ㄹ ■ ˘ ˩ • .
˘ , Scaleform include ♬ . vcproj ㄹ ■' ǎ ß ♬ ╪ Tutorial/Section3.5 ㅂ /
↑ ▓ .

파일과 파일

AS3_Global.h Obj₩AS3_Obj_global.xxx ㄹ ■ ╪ ˩ /
ㄴ ᶜ .
AS3_Obj_Global.xxx ㄹ ■ " " ActionScript 3 _ ㅅ ˘ _ ㅅ ╥ ˆ ˚ .
ᵖ ö swf ㄹ ■ ㄴ ↑ ╪ Ũ ╥ " ˊ." _ ㅅ (_ ㅅ) ˚ . C++ ˆ
ᵖ ö _ ㅅ ╬ GlobalObjectCPP ▓ . ╪ Scaleform VM
ˆ / ʳ .

AS3_Global.h ♣ ˢ ♠ Ọ̈ . ╪ ر ■ ClassRegistrationTable
ˁ . ╪ AS3 ˆ C++ w ˢ ♠ Ọ̈ .
╪ w ′ ȝ ő й Ł ↿ ▒ . ḃ ر ■ Ɔ / Ä й
ClassRegistrationTable ♬ ╥ – . Ώ ȝ ő / Ỹ ↿ ▒ . ╪
╪ ᵖ ╪ ‡ AS3_Global.h ˁ .

Ọ̈ ▫ ClassRegistrationTable ˂ ȝ (й ⊓ ╪ ȝ)
ClassRegistrationTable ˁ include ر ■ ˁ – . ◗ ↓
AS3_Global.h ╥ Ọ̈ ʳ ө˘ ′ ㄴ Ä ㄴ Ł
/ ˁ ↿ ▒ . ╪ ˣ Ł Ọ̈ Ώ .

ʳ ╪ ㄴ ♠ ’ ’ Ω̣ ω̃ – ↿ ╪ ▓ . AS3 VM / Ä ╪ Ä
▌/ ḃ ㄴ Ä ㄴ ‡ ╪ / Ä Ł ˂ ╪ ㄱ
▓ . ƒ Ä " " ọ̈ Ł ╪ Ł ▬ Ç
╬ .

Scaleform Unreal® Engine 3, Gamebryo™, Bigworld®, Hero Engine™, Touchdown Jupiter Engine™, CryENGINE™ Trinigy Vision Engine™ [illegible]

[illegible]

[illegible]

[

ر ■ ShadowVolume.cpp / ⅄ .

```
#include "GFx_Kernel.h"
#include "GFx.h"
#include "GFx_Renderer_D3D9.h"
```

Ÿ ц Scaleform _ ⅄ ę GFxTutorial ‡ ĭ ╪ /
⅄ Ự Ɔ / ⅄ ⁼ . ↑ ╥ / Scaleform Ũ
K ‡ ȝ й ϟ ⁼ ᵣ ɵ Ỏ ᵣ ᾧ ợ Ł ⅄ ᵇ ö
Scaleform _ ⅄ j ▓ ◗ ⅈ ╪ ▓ . ╪ _ ⅄ Ợ ╪ ╥ Ũ ▬
/ ˝A Ũ ˝E ǧ ↿ ╪ .

```
// One GFx::Loader per application
Loader          gfxLoader;

// One GFx::MovieDef per SWF/GFx file
Ptr<MovieDef>   pUIMovieDef;

// One GFx::Movie per playing instance of movie
Ptr<Movie>  pUIMovie;

// Renderer data
Ptr<Render::D3D9::HAL>  pRenderHAL;
Ptr<Render::Renderer2D> pRenderer;
MovieDisplayHandle     hMovieDisplay;
```

Scaleform ↿ ╥ ⅄ ´ Scaleform ɾ ᵇ
' GFx::System ǂ ⅞. ↿ ╪ . WinMain / й ⅄ .

```
// One GFx::System per application
GFx::System         gfxInit;
```

GFx::System ǂ ⅞. й ⅄ Scaleform ⅄ ♣ / / ɵ ‡ ▓ ‡ – ę
Scaleform Ợ ‡ ĭ ╪ ⁻ ♣ Ʊ ↑ ˝A Ỏ . ╪ ↿
WInMain Ỏ / ▓ ↄ ᵘ ╪ . ˝A ↿ GFx::System Scaleform ɾ ᵇ
ᵇ Ợ . ↑ ‡ ĭ ╪ ╥ ᾶ ɾ ᵇ ᵇ ╪ й
. ╪ ˝A ╥ ᶜ ♠ Ũ ⁼ GFx::System ↿ ȝ ↄ ᵘ /
˝A ╪ ♬ ⅄ ᵇ .

GFx::System й ‡ ╬ ╬ ♣ / / Ä ‡ ↑ – .
ƒ Ä, й ♣ ■ ' . ╬ GFxTutorial ╬ Ł ♣ /
‡ j ┤╬ .

‡ ¯ ╬ ╥ w / ƒ ° r , GFx::System::Init()' GFx::System::Destroy()
♬ ♠ ∧ ┤╬ GFx::System _ ∧ ╬ r й ┤ ┤╬ .

↑ ʉ Scaleform ┤ InitApp() Ɔ / Ä ‡ ╬ ╥ WinMain╬ ▫ ∧ ♠
┤ ■ ‡ ϱ . ∧ ϖ ¯ й InitApp() / ∧

```
gfx = new GFxTutorial();
assert(gfx != NULL);
if(!gfx->InitGFx())
   assert(0);
```

GFxTutorial GFx::Loader _ ∧ ˚ ˘ ▒ . ■ ♠ ‡ ╬ ↑ ╥
GFx::Loader _ ∧ ˘˘ ▒ ę ╬ _ ∧ SWF/GFx ď ∧ ╥ ы' ď ∧
▫ ╬ / ℙ ◗ ę ╬ w / ▫ ☁ Ọ'
▒ ⁼ . ╥ SWF/GFx ر ■ ╬ Ç ت ٪ ¯ ▫ ˆ Ω ƒ ▫
♥ – ▒ . GFx::Loader ˝ Ÿ / Ọ' GFx::Log ¯ ■ ╥
ε̈ ♬ Ũ'

GFxTutorial::InitGFx() / Ä ∧ ´ GFx::Loader / Ũ' ε̈ ♬ ┤╬ .
GFx::Loader SetLog Łˆ ɵ Ÿ ∧ ♠ ♬ ♣ . ╥
Scaleform Ỹ ╬ / ∧ ʊ ʊ / Ÿ / ʉ
. ╬ / Scaleform PlayerLog ɵ Ọ' , ╬ ɵ
ƒ / ∧ . r ⁼ ▌ ╥ Ÿ ' ╥
GFx::Log Ũ' ˆ Ä r ɵ ▒ ┤╬ .

```
// Initialize logging -- Scaleform will print errors to the log
// stream.
gfxLoader->SetLog(Ptr<Log>(*new Log()));
```

GFx::Loader GFx::FileOpener ď ∧ ■ . ˆ / ε̇
Ÿ ر ■ ■ r ƒ ▫ ˝ ▫ ر ■ ╥ Ọ' ▫ ы
GFx::FileOpener Ũ' ʒ .

```
// Give the loader the default file opener
```

```
Ptr<FileOpener> pfileOpener = *new FileOpener;
gfxLoader->SetFileOpener(pfileOpener);
```

Render::HAL [illegible] Scaleform [illegible] D3D9 [illegible] D3D [illegible] Scaleform [illegible] HAL::InitHAL / Render HAL [illegible] IDirect3Device9 [illegible] Scaleform DX9 [illegible] UI [illegible]

```
pRenderHAL = *new Render::D3D9::HAL();
if (!(pRenderer = *new Render::Renderer2D(pRenderHAL.GetPtr())))
  return false;
```

[illegible] Scaleform [illegible] Render::D3D9::HAL [illegible] Render::HAL [illegible] Scaleform [illegible]

[illegible] GFx::Loader [illegible] GFx::MovieDef [illegible] GFx::MovieDef [illegible]

```
// Load the movie
pUIMovieDef = *gfxLoader.CreateMovie(UIMOVIE_FILENAME,
                                     Loader::LoadKeepBindData |
                                              Loader::LoadWaitFrame1, 0);
```

LoadKeepBindData [illegible] D3D [illegible]

LoadWaitFrame1 [illegible] CreateMovie [illegible] GFx::ThreadedTaskManager [illegible]

[illegible] (Arena) [illegible] Memory System Overview [illegible]

[illegible] GFx::MovieDef [illegible] GFx::Movie [illegible] [illegible] .
GFx::Movie [illegible] [illegible] [illegible]

Render::D3D9::HAL::InitHAL ╱ ♣ ę, D3D Ÿ ╪ Ỹ ̈E ╪♣,
Scaleform ╲ ╪♣ ╱ ╲ . D3D Ÿ ╪ ɵ╪ ̧ ȝ
InitHAL ╲ – ę ╪ w♥ ̋ ♣╲ ȝ╱ ᵖ й ̧ ╱
Ỹ ▓ .

ShadowVolume ╥ OnResetDevice DXUT ╱ ╥ ̈A ┤ Ÿ ╪ Ỹ ̈E ,
˘ Ÿ ╪ ╱ ╲ . й GFxTutorial Ɔ ╥
OnResetDevice ʃ й ╱ ┤ ‴ .

```
pRenderHAL->InitHAL(
    Render::D3D9::HALInitParams(pd3dDevice, presentParams,
                                Render::D3D9::HALConfig_NoSceneCalls));
```

InitHAL ╲ Scaleform ╱ D3D ◗╲ ♬ ♣ .
HAL_NoSceneCalls DirectX BeginScene() EndScene() ╲╪ Scaleform
╪ Ω ┤╪ . ╪ ShadowVolume Ỵ ╪ OnFrameRenderer
╱ ̈A ╪ɵ ╲ ╪Ç ˘ ▓ ᵕ ᵘ ╱ ┤╪ .

w♥ ˗ ↑ ‡ ̈ї╪ ♣╲ ȝ ♣ ̧ , D3D Ÿ ╪
٪ ┤╪ . ♬ ♩ ̧ ╲ ̊ ᵖ ö D3D ̈A ┤ –
. ShadowVolume OnLostDevice ╱ ̈A ̈A Ł . Render::HAL ╱
٪ Ÿ ╪ ╱ ' Ɔ Ω ϟ ȝ GFxTutorial Ɔ OnLostDevice ╱ ̈A D3D
▫ Ł ϟ ̄ .

```
pRendererHAL->ShutdownHAL();
```

╪ ´ ╪♣ ´ DXUT ▌ ╥ ┤
٪ Ÿ ╪ ̈ɛ ʃ ┤╪ . Win32/DirectX ╱ Ł
Scaleform SDK ╱ ̈A Łˆ GFxPlayerTiny.cpp ر ■ ˘ ▫ .

ᵖ ö Scaleform _╲ GFxTutorial _╲ Ɔ╱ ̊ ‡ ▓ ᵕ ᵘ ╱ ╲ WinMain ╥
¡ ╱ ̈A GFxTutorial _╲ ϱ Ł ┤ ʳ .

```
delete gfx;
gfx = NULL;
```

˘ Ọ̈ ♬ ♩ ˭ ˉ DirectX 9 ▫ ╥ ∖ . Scaleform
╪ ˽ ↳ ʃ ╬ ˭ ▌ ⅄ / Ỹ ∖ / ˝A InitHAL()
ShutdownHAL() ‡ Ɛ ╪ ↿ ╪ ⅄ .

DirectX9 ˆ / ˝A InitHAL ♬ ♩ ˭ ˚ D3DPOOL_DEFAULT ▫ .
' ↄ ∧ D3DPOOL_DEFAULT ' ▫ ˚ / ˝A
InitHAL ∧ .

ShutdownHAL D3DPOOL_DEFAULT ▫ Ł . ShadowVolume ˚ ˝A
DXUT ▌ Ọ̈ ‡ ╪ DXUT OnResetDevice / ˝A
D3DPOOL_DEFAULT ▫ – ę OnLostDevice / ˝A Ł – .
GFxTutorial::OnLostDevice Ɔ ╥ ShutdownHAL ∧ GFxTutorial::OnResetDevice Ɔ ╥
InitHAL ∧' ╪ .

' ˽ D3DPOOL_DEFAULT ▫ Ỹ ˝E ˗ ↑ Ł ʒ
∧' InitHAL ShutdownHAL ∧ .

∫ Ũ / ˝...♬ ˚ ˝.. й . ╪ ˽ /
♣ ∨ ♩ ˘ ▒ . ʒ Ũ ˽ ↳ ᵘ / D3D Ÿ ╪
↳ ˚ ˝.. GFxTutorial::OnResetDevice / й
∧ .

```
// Use the window client rect size as the viewport.
RECT  windowRect   = DXUTGetWindowClientRect();
DWORD windowWidth  = windowRect.right - windowRect.left;
DWORD windowHeight = windowRect.bottom - windowRect.top;
pUIMovie->SetViewport(windowWidth, windowHeight, 0, 0,
                      windowWidth, windowHeight);
```

SetViewport╥ ∧ ˛ ╥ ♯ ˛ Ọ̈ ▌ ˭ ╥ (■ ♠ PC
‡ ╪) ♬ ╥ . / ↑ Л ˛ ╥ ♯ ˛
Scaleform ▌ ˭ Ɔ ╥ ˚ ˝.. ♬ ╥ .

▌ ˭ ♣ ╬ ▫ OpenGL' ╥ ˝E, ! ⅇ Ọ̈
, ▌ ˭ Ω ' ↳ ȳ ╥ ╪ .

Scaleform ğ ⅄ ╥ ˜ ♬ ˘ ˚ ⅓ Ɔ / ‡ ˙ ˝ ⅄ ┤ ╬ /
’ Ł ̂ . ‡ ĭ ╪ ╥ ┐ ⅄ 4.2 / ̈A ╪ /
’ ┤ ╪ .

ShadowVolume ╥ OnFrameRender() / ̈A ╪ ╪ . ᵖ ö D3D
⅄ BeginScene()’ EndScene() ⅄ Ơ ╪ / ̈A ╪ . / ė EndScene()
╪ ♣ / GFxTutorial::AdvanceAndRender() ⅄ .

```
void AdvanceAndRender(void)
{
      DWORD mtime = timeGetTime();
      float deltaTime = ((float)(mtime - MovieLastTime)) / 1000.0f;
      MovieLastTime = mtime;

      pUIMovie->Advance(deltaTime, 0);
      pRenderer->BeginFrame();

      if (hMovieDisplay.NextCapture(pRenderer->GetContextNotify()))
      {
          pRenderer->Display(hMovieDisplay);
      }

      pRenderer->EndFrame();
}
```

Advance deltaTime ┤ ˩ ∫ ◓ Ỹ . ∫ ╥ ◓ Ỹ ◓
┐ ‡ ĭ ╪ Ł ‡ . ̈A ⅄ ‡ ẽ ♬ / ̈A ∫ Ł
◓ Ỹ ╬ Ł GFx::Movie::Advance / Ł ̂
┤ ╪ ⅄ .

Render::Renderer2D::Display D3D Ÿ ╪ / ̈A ∫ ▌
DirectX ⅄ . ̈E Ũ ╥ ╪ ы ᵖ й ⅄ ᵖ ◗ ẽ ♬ ᵖ ◗
Ω . ƒ ̈A, D3D Ÿ ╪ Ũ Render::Renderer2D::Display ⅄ /
▓ . ╪ ╬ ц ц ‡ ĭ ╪ O ▓ . ◗ ♪ ♠ ╬ ˩
Display ⅄ ╪ ♣ / Ÿ ╪ Ũ ᵖ ◗ ˘ ↑ ⅄ / ╪ ̂ ┤ ╪ . Scaleform
Ũ ˝ ▌ ◓ ┤ ̈Ω ↑ ̈E ˥ ▓ .
╪ ̈A / ̈A DX9 ╥ Ũ Ơ ̈A ˝ Ũ ᵖ ◗ ˘
̂ .

DX9 Ũ‡ ĭ╪ ╪▬ ẋ╬ Ȯ ę
GFxTutorial::AdvanceAndRender() ⅄ Ö / Ọ .

```
// Save DirectX state before calling Scaleform
g_pStateBlock->Capture();
// Render the frame and advance the time counter
gfx->AdvanceAndRender();
// Restore DirectX state to avoid disturbing game render state
g_pStateBlock->Apply();
```

¡ ´ DXUT ╥ UI Ọ ¡ ┤╪ . ╪ ий╥ ' ăʃ ⅞..
⅄ Ẓ╪ ę ⅄ ¨ ' Section4.1\ShadowVolume.cpp / ↑ ▓ .
╪♩ ╪♣ ий ¬ . DXUT ' ᵇ ö ← ╬ ▫ .

```
// Disable default UI
..
```

╪Ł DirectX ‡ ĭ╪ / Ẓ ий ‡ ʃ ⁼ ┤╪ .

Scaleform ╥ UI ▓ ShadowVolume ‡ ĭ╪

GFx::Movie::SetViewport [illegible] Scaleform [illegible]

[illegible]

[illegible] 4:3' [illegible] Scaleform [illegible] 800x600 [illegible] 1600x1200 [illegible] Scaleform [illegible]

GFx::Movie::SetViewScaleMode [illegible] GFxTutorial::InitGFx() [illegible] GFx::Movie [illegible]

```
pUIMovie->SetViewScaleMode(Movie::SM_ShowAll);
```

[illegible] SetViewScaleMode [illegible]

SM_NoScale	[illegible]
SM_ShowAll	[illegible]
SM_ExactFit	[illegible]
SM_NoBorder	[illegible]

[illegible] SetViewAlignment [illegible] SM_NoScale [illegible] SM_ShowAll [illegible]

SetViewAlignment ˋ ⅞ Ɔ / ̈A ǎ ʌ ‡ Ÿ / й ˉ ♬ . ╪ ← ╬ ˊ ╪
， ȝ ↓ ʌ ẋ / й ѿ – .

```
pUIMovie->SetViewAlignment(Movie::Align_CenterRight);
```

w ♥ ↄ ‡ ǐ ╪ ⅞..˝E╪ ‡ ˇ ˝
SetViewScaleMode SetViewAlignment╥ Ɔ ← ▫ .

SetViewAlignment SetViewScaleMode SM_NoScale > ȅ ♬ ←
Ł ˘ ‡ Ʊ ϟ Ω . ○ ♬ , w ♥, ʌ w ♬ ȳ ╥ ˆ Ọ / Scaleform ∫ ▫ ╥ ʌ ė ▓ ⅞ ◗ . Ỵ ⅞ d3d9guide.fla / ▓ .

♬ ₨ ¸ C++╪ ѿ ⅞ ȅ ♬ ╪ . SetViewScaleMode SetViewAlignment ⅞ Stage (Stage.scaleMode, Stage.align) Łˆ ■ ˝E > ♬ .

ShadowVolume ╥ ╪ ╫╪ Scaleform ♬ ‴ ⸮ ⸮ /
╪ Ł ▃ Ỹ ⸮ ╫ Ũ ▬ ▫ . ɵ ‡ ˝A ، ╪
ˬ ╪ ╪ ٪ Ỹ ˘ ٪ Ũ ▫ / ▫ ▒ ȝ Ụ ▫ † †
ˡ ╪ .

The GFx::Movie::HandleEvent ˝ ! ج - ╪ ٪ ▒ ♬
† Ɔ GFx::Event _ ৷ ♣ . ‡ ˝ˡ ╪ ▒ / ╪ ٪ ˆ ˜E ˝A ╪
♠ ♥ GFx::Movie ♣ .

ShadowVolume MsgProc / ˝A Win32 ▒ ╪ ٪ . Scaleform ╪ ٪
৷ й GFxTutorial::ProcessEvent / ৷ . й
WM_MOUSEMOVE, WM_LBUTTONDOWN WM_LBUTTONUP ৷ :

```
void ProcessEvent(HWND hWnd, unsigned uMsg, WPARAM wParam, LPARAM lParam,
                  bool *pbNoFurtherProcessing)
{
 int mx = LOWORD(lParam), my = HIWORD(lParam);
 if (pUIMovie)
 {
        if (uMsg == WM_MOUSEMOVE)
        {
               MouseEvent mevent(GFx::Event::MouseMove, 0, mx, my);
               pUIMovie->HandleEvent(mevent);
        }
        else if (pMovieButton && uMsg == WM_LBUTTONDOWN)
        {
               ::SetCapture(hWnd);
               MouseEvent mevent(GFx::Event::MouseDown, 0, mx, my);
               pUIMovie->HandleEvent(mevent);
        }
        else if (pMovieButton && uMsg == WM_LBUTTONUP)
        {
               ::ReleaseCapture();
               MouseEvent mevent(GFx::Event::MouseUp, 0, mx, my);
               pUIMovie->HandleEvent(mevent);
        }
 }
}
```

Scaleform [illegible]

#1: [illegible]

```
pMovie->SetViewport(screen_width, screen_height, 0, 0,
                                screen_width, screen_height, 0);
```

[illegible] (0, 0) [illegible]

#2: [illegible]

```
pMovie->SetViewport(screen_width, screen_height, 0, 0,
                                screen_width / 4, screen_height / 4, 0);
```

[illegible] HandleEvent [illegible]

#3: [illegible]

```
movie_width  = screen_width / 6;
movie_height = screen_height / 6;
pMovie->SetViewport( screen_width, screen_height,
                     screen_width / 2 – movie_width / 2,
                     screen_height / 2 – movie_height / 2,
                     movie_width, movie_height);
```

[illegible] (0, 0) [illegible] (screen_width / 2 – movie_width / 2, screen_height / 2 – movie_height / 2) [illegible]

[illegible] GFx::Movie::SetViewAlignment [illegible] GFx::Movie::SetViewport [illegible] SetViewAlignment [illegible] SetViewScaleMode [illegible] Scaleform [illegible]

й ╪ ⅞ GFx::Movie::HandleEvent ч . GFx::KeyEvent GFx::CharEvent
ϰ ╪ ⅞ ▓ .

```
KeyEvent(EventType eventType = None,
                  Key::Code code = Key::None,
                  UByte asciiCode = 0,
                  UInt32 wcharCode = 0,
                  UInt8 keyboardIndex = 0)

CharEvent(UInt32 wcharCode, UInt8 keyboardIndex = 0)
```

GFx::KeyEvent ˄ й Ọ ˇ , GFx::CharEvent ч ASCII ᵘ ▫ Ọ .
/ ˝A GFx::CharEvent WM_CHAR ɾ / Ỹ ˝E . GFx::KeyEvent
WM_SYSKEYDOWN, WM_SYSKEYUP, WM_KEYDOWN WM_KEYUP ɾ /
Ỹ ˝E . ц ö ↿ ╪ .

- SHIFT ы Ũ / ˝A ʻcʼ ы : WM_KEYDOWN ɾ /
  GFx::KeyEvent Ỹ ˝E ‡ – ę ' ⁻ Ɔ Ω ⊓ ↿ ╪ .
  - ʻcʼ ˇ , ╥ ˄ й
  - ы ˇ ▓ . ˇ
  - SHIFT ▬ ϰ ╪ .

- ╪ / ″ ч ϱ ″ ASCII > ʻCʼ Scaleform ♣ WM_CHAR /
  GFx::CharEvent ‡ – .

- ʻcʼ й ˄ WM_KEYUP ɾ / GFx::KeyEvent Ɔ – .
  й ˄ GFx::CharEvent Ъ ′ .

- F5 ы : ы ϖ GFx::KeyEvent Ɔ ę ϖ ╪ ⅞
  Ë . F5 ϧ ASCII й / Ω ϖ ᵘ / GFx::CharEvent Ъ
  ′ .

╥ GFx::KeyEvent ╪ ⅞ ы ˇ ˢ ╪ ⅞ ♣ . ⁻ / ♠
▬ , й GFx_Event.h ر ■ / ˝A ♬ ╥ Ɔ ♠ Ọ
й ϩ ⅄ . GFxPlayerTiny.cpp Scaleform Player ă ˄ й

й й ˅ ▓ . ╪ ◗ / ˝A ╥ ¡ й ő 3D
’ ৲ ▲ Ọ ╪ ProcessKeyEvent ˅ ▓ .

```
void ProcessKeyEvent(Movie *pMovie, unsigned uMsg, WPARAM wParam, LPARAM
lParam)
```

╥ WndProc / ˝A ˘ WM_CHAR, WM_SYSKEYDOWN, WM_SYSKEYUP,
WM_KEYDOWN, WM_KEYUP ɾ / ৷ . ♠ ♥ Scaleform ╪ ⁄.
Ỹ ˝E ‡ pMovie ♣ ৷ ╪ .

GFx::KeyEvents GFx::CharEvents ▫ ♣ ˝ ˜ . ɵ ‡, ╥ ⁄.
Ũ ▫ GFx::CharEvents / ˵ ╪ ৷ ASCII ▫ ’ ৲ ╪ .
˝ ⁄. Ũ ▫ й ▫ (, ˜ ‡ IME) ▓ ‡ – . IME ▒ ╥ ˵ ,
й Ỏ ę ৷ ▫ ╪ ⁄. (■ ♠ IME ৷ ╥
⁄. ╬ ˜ ’) ⁄. Ũ ▫ / ৷ ╪ . ȝ ⁄. Ũ ▫ Ł ‡ GFx::KeyEvent
Page Up’ Page Down ▓ ╪ ৷
▫ Ỏ ৲ ╪ .

Tutorial\Section4.3 / ╪ ◗ / ˝A ë ∫ ৷ й ˅ ‡ ▓ .
˅ , ╬ . , ╪ ˝ ╪ ę 3D ’ ╥
Ỏ ╪ Ł ▬ ৷ ╪ . “Settings” ы ȝ C++ й ’ ╪ DX9 ë ♬
♣ ╪ d3d9guide.fla ⁄. Ọ ˝ ˆ ৲ ╪ .

/ ˝A й ৷ й “Change Mesh” ˅ ⁄. ▒ Ũ ▫ / ▒
. / ˝A ц ˜ ♬ – Ọ Ọ ╪ ▓ . “Change Mesh” ,
⁄. ▒ Ũ ▫ ৲ ↑ ↑ ɾ ╪ / ৸ ˂ ▫ . ɾ ╪ / ˝A
Ọ ˝A ˝ ৷ ▓ ˵ , ⊢ ╥ ⁄. ŧ ╬ ˂ ╪ ╥ ৷ ȝ / ˝A ৲
♠ ╪ . ɾ ╪ ৷ ♠ ╬ ⁄. ŧ Ỏ ˵ ῷ ő й ৲ ▫ /
ɾ ╪ ы ╪ ˅ , ╪ Ỹ ę ∫ ″ й ▬‰ ╪ ‡
˩ ╪ ▓ .

‡ ĭ ╪ / ৸ ɾ ˵ , ы Ũ / ˝A
╬ . ╪ Ł UI ˜ ↑ ˚ ˝A / ╪ ■

[illegible]

UI 3D [illegible] GFx::Movie::HitTest [illegible]

[illegible] GFxTutorial::ProcessEvent [illegible] HitTest [illegible] UI [illegible] DXUT [illegible]

```
bool processedMouseEvent = false;
if (uMsg == WM_MOUSEMOVE)
{
   MouseEvent mevent(GFx::Event::MouseMove, 0, (float)mx, (float)my);
   pUIMovie->HandleEvent(mevent);
   processedMouseEvent = true;
}
else if (uMsg == WM_LBUTTONDOWN)
{
   ::SetCapture(hWnd);
   MouseEvent mevent(GFx::Event::MouseDown, 0, (float)mx, (float)my);
   pUIMovie->HandleEvent(mevent);
   processedMouseEvent = true;
}
else if (uMsg == WM_LBUTTONUP)
{
   ::ReleaseCapture();
   MouseEvent mevent(GFx::Event::MouseUp, 0, (float)mx, (float)my);
   pUIMovie->HandleEvent(mevent);
   processedMouseEvent = true;
}

if (processedMouseEvent && pUIMovie->HitTest((float)mx, (float)my,
                                    Movie::HitTest_Shapes))
   *pbNoFurtherProcessing = true;
```

[illegible] "Change Mesh" [illegible] 4.3.2 [illegible] W, S, A, D, Q, E [illegible] Scaleform [illegible] 3D [illegible]

[illegible] 6.1.2 [illegible] C++ [illegible]

Scaleform ♠ ╪ ˘ ث٪.. Łˆ . , ث٪.., ˚ ╬٪.. ◗ Ɔ ╪ ث٪.. ▓ / Ơ ɾ ᵖ ʃ Ϻ ♠ ╪ ˘ ■ ث٪.., ˆ ث٪.. ╪ , ١´ TTF ث٪.. ╪ Ÿ ˥ ▓ . ث٪.. ╥ ١ ق̃ ق̃ ث٪.. ╥ ɾ ᵖ Ơ ˀ . ث٪.. ▬ ˝ Ȳ Ä˝ / ь , ę ╪ ت .
Scaleform ╥ ث٪.. ~ Ł / ♣ ١ ᵘ Ä˝ ˛ ▫ ᵘ Ä˝ ˚ ╪ / ↑ ▓ ę
ɔ http://gameware.autodesk.com/scaleform/developer/?action=doc ╪ .

♣ ♠ ╬ ث٪.. ♣ ١ ث٪.. ╥ ▫ ١ ˘ ˢ ▫
ث٪.. ١ ȝ ١ ґ Ơ . ╪ Ä˝ ث٪..
■ ╪ ˝ ˢ / ١ ١ . ث ᵘ ▫ ʃ ᵖ Ơ ╪
˥ ╪ ʳ 5000 ˛ ╥ ╪ ‡ ق̃ ق̃ ث٪.. ╥ ♠ ╪ . ᵘ ˛ ╥ ﬡ
١ ʃ ᵖ ę ˝ ١ . Ä˝ ث٪..
■ / – ᵘ ╪ .

Scaleform ♠ ث٪.. ˝ ╪ ᵘ Ł ˥ . ١ ╪ ▓ ق̃ ˝ / ᵘ ▫ ę
˝ Ɔ ˽ ١ . Ä˝ ث٪.. ■ Scaleform ╥
ق (glyph-packing) Ϻ˘ ■ ˆ ˝ / Ä˝ ˆ . Scaleform ث٪..
▬‰ , ╪ ‡ Ɛ ╥ ᵘ ▫ / Ä˝ ↑ ╥ TTF ث٪.. ʃ ᵖ / ᵖ ◗ ȝ
. ╪ . ¡ Scaleform “fake italic” “fake bold” Łˆ , ╪ ١ ♠ ╬
ʃ ᵖ ♥ – ȝ Ä˝ ث٪.. ╥ ج ╪ ╪ ↑ ١ ÿ ▓

ґ ٪.. Ù˝ ♪ ╪ . ╪ ٪.. ᵘ Ω
˛ ę ˢ ▌╥ ╪ ؛ ȝ Ȳ˝ ˉ / Ä˝ .
Bin\Data\AS2\Samples\FontConfig / Ä˝ ث٪.. ë ♬ Ł ق̃ sample.swf ر ■ Scaleform Player
D3D9 / ÿ Ä˝ ▲ Ỹ ▫ .

4c 4d ⁄ ̈A, Scaleform ⁄...̈E ´ E ‡ ̈A ▫ ╥ ر ̀O ˘ , ╪ Ù̈

⁄.. ŧ ▫ .

ث ⁄.. ⁄.. , ’ . ’ Ỹ ̈E ȝ ǒ

Ɔ ╥ ⁄.. й ⁄ ♠ ♥ ɛ̈ ♬ . ∧ ♠ ╬ ̈E Ọ̈ ث ⁄.. ⁄.. ɛ̈ ♬ ¸ ⁄

↑ ▒ . Ł ر ■ ╥ ╪ gfx_2.1_texteffects_sample.zip ╪ ę ¸ ▫ ⁄ ̈A й

▒ .

⁄.. ’

[illegible]

[illegible]

Scaleform C++ [illegible] ActionScript::FSCommand [illegible]

[illegible] FScommand [illegible]

C++ [illegible] GFx::Loader [illegible] FSCommand [illegible] ActionScript [illegible]

FSCommand [illegible]

[illegible] flash.external.ExternalInterface. [illegible] GFx::Value [illegible] (6.1.2 [illegible]). [illegible]

[illegible] FSCommand ExternalInterface [illegible] fscommand [illegible] , gfxexport –fstree,- fslist -fsparams [illegible] SWF [illegible] fscommand [illegible] ExternalInterface [illegible] fscommand [illegible]

ActionScript [illegible] fscommand [illegible]

```
fscommand("setMode", "2");
```

˝ ♬ ¯ > fscommand / ˝A ᵘ ▫ . ExternalInterface ♬ ♪
▓ .

╪ ∫ , ╥ ᵘ ▫ GFx FSCommand ɵ ♣ . ‡ ǐ ╪
GFx::FSCommandHandler Ũ˝ ᾷ˝A fscommand ɵ Ȳ ˘ , GFx::Loader ↑ ,
GFx::Movie _ ∨ Ȳ . ʳ – ∫ ɵ GFx::Movie / ȅ ♬ ȝ ∫ ╬ / ˝A
fscommand ʳ ∧ . GFxPlayerTiny ╥ ╪ ˝E
↑ Ë ↿ ╪ Ɔ ę (“FxPlayerFSCommandHandler” ↿) ShadowVolume / Ǫ˝ й
∧ . ╪ ◗ / ˝A ╥ ↓ˆ й Tutorial\Section6.1 / ▓ .

ė , GFx::FSComandHandler ˝A .

```
class OurFSCommandHandler : public FSCommandHandler
{
    public:
    virtual void Callback(Movie* pmovie,
                          const char* pcommand, const char* parg)
    {
        GFxPrintf("FSCommand: %s, Args: %s", pcommand, parg);
    }
};
```

ʃ й ˛. Ɔ fscommand ♣ , ╥ ᵘ ▫ Ȯ ʳ ᾷ fscommand
∧ ∫ ╬ ╥ ˢ ╬ .

, GFxTutorial::InitGFx() Ɔ / GFx::Loader _ ∨ Ỹ ˜E ɵ Ȳ .

```
// Register our FSCommand handler
Ptr<GFxFSCommandHandler> pcommandHandler = *new OurFSCommandHandler;
gfxLoader->SetFSCommandHandler(pcommandHandler);
```

GFx::Loader ɵ Ȳ ȝ ᵖ ö GFx::Movie ╪ ɵ ╥ ˜E ᵘ ˣ .
SetFSCommandHandler ╪ ȅ ♬ ╪ й , ∫ ╬ / ˝A
∧ .

╥ ᾷ ɵ ᾷ fscommand ╪ ˛. Ÿ / ∧ . ShadowVolume
˘ “Toggle Fullscreen” , . UI ╪ ˛. Ỹ ↄ ╪ ˛. ∧
╬ .

```
FSCommand: ToggleFullscreen, Args:
```

Ш Ÿ / ˝A d3d9guide.swf ر ■ ˘ ٪. ق B (F9) . ع / ٨
╪ ٪. Symbol Definition(s) Æhud Ævar : Frame 1: / ˝A Ỹ ˝E fscommand() ٨ ╖ .

٪. fscommand() ٨

Ù˝ ق fscommand("ToggleFullscreen") fscommand("ToggleFullscreen", this.UI._visible)
˝A this.UI._visible > C++ / ٨ . ∫ ر ■ ⁝ ˚ ٪. ˝A(CTRL+ALT+Shift+S)
d3d9guide.swf ╖ ١ .

UI Ũ˝ ╥ ، FSCommand ╪ ٪. Ỹ ٦ ♠ ♥ й ▬ ˝Ω ShadowVolume
Ỵ / ▓ . ˝A, ╥ й fscommand ɵ / ٨ ♣ ١ ع ᴰ й
٨ .

```
if (strcmp(pcommand, "ToggleFullscreen") == 0)
    doToggleFullscreen = true;
```

DXUT ╬ OnFrameMove ▌ ♣ / ٨ . OnFrameMove ч /
й ٨ .

```
if (doToggleFullscreen)
{
    doToggleFullscreen = false;
    DXUTToggleFullScreen();
}
```

■ ♠ , ╪ ⅞ ɵ Б ╪ ‡ – ę ů ^ ▫ – . ╪ ⅞ ɵ ■ ♠ Advance ˝ Invoke ^ / ˵ ^ .

╥ ExternalInterface.call fscommand Ọ ↑ ♣ ╬ ▫ ^ >
▓ ^ ȝ ↄ ᵘ / ė .

ExternalInterface ɵ ȳ ↿ fscommand ɵ ȳ ↿' Ọ .

```
class OurExternalInterfaceHandler : public ExternalInterface
{
 public:
 virtual void Callback(Movie* pmovieView,
                       const char* methodName,
                          const Value* args,
                       unsigned argCount)
 {
        GFxPrintf("ExternalInterface: %s, %d args: ",
                                          methodName, argCount);
        for(unsigned i = 0; i < argCount; i++)
        {
               switch(args[i].GetType())
               {
                  case Value::VT_Null:
                     GFxPrintf("NULL");
                     break;
                  case Value::VT_Boolean:
                     GFxPrintf("%s", args[i].GetBool() ? "true" : "false");
                     break;
                  case Value::VT_Number:
                     GFxPrintf("%3.3f", args[i].GetNumber());
                     break;
                  case Value::VT_String:
                     GFxPrintf("%s", args[i].GetString());
                     break;
                  default:
                     GFxPrintf("unknown");
                     break;
               }
               GFxPrintf("%s", (i == argCount - 1) ? "" : ", ");
        }
        GFxPrintf("\n");
 }
};
```

˘ ɵ GFx::Loader / ȳ .

```
Ptr<ExternalInterface> pEIHandler = *new OurExternalInterfaceHandler;
gfxLoader.SetExternalInterface(pEIHandler);
```

[illegible] Symbol Definitions() Æhud Ævar : Frame 1 [illegible]

MeshPathFocus [illegible]

ExternalInterface [illegible] ExternalInterface [illegible] "MeshPathFocus" [illegible]

```
ExternalInterface: MeshPathFocus, 1 args: false
Focus change:
       _level0.hud.text_MeshPath.field => null
ExternalInterface: MeshPathFocus, 1 args: true
Focus change:
       null => _level0.hud.text_MeshPath.field
```

[illegible] GFxTutorial [illegible]

```
if (strcmp(methodName, "MeshPathFocus") == 0 && argCount == 1)
{
       if (args[0].GetType() == Value::VT_Boolean)
```

```
                    gfx->SetTextboxFocus(args[0].GetBool());
        }
```

GFxTutorial::ProcessEvent ½ Ũ▫ ˚ ą̆ ← й ╪ ½ ˢ ∫ /
♣ . ˢ – й ╪ ½ ½ Ũ▫ / ♣ ȝ ę̈♬ 3D ╥ ♣
¡ ˣ .

```
        if (uMsg == WM_SYSKEYDOWN || uMsg == WM_SYSKEYUP ||
            uMsg == WM_KEYDOWN    || uMsg == WM_KEYUP    ||
            uMsg == WM_CHAR)
        {
              if (textboxHasFocus || wParam == 32 || wParam == 9)
              {
               ProcessKeyEvent(pUIMovie, uMsg, wParam, lParam);
               *pbNoFurtherProcessing = true;
              }
        }
```

˚ ╪ (ASCII й 32) (ASCII й 9) Ũ♣ ęˢ “Toggle UI” “Settings” /
.

Ọ▫ ˣ╬ ʃ ˢ ˡ ▓ “Change Mesh” ˛ ½ Ũ▫
˘ ˛ Ụ ʃ ╪ ▒ ы . ȝ˛ ½ Ũ▫
½ ╪ ½ ɵ ˡ ╪ Ụ ʃ ╥ ╪ ExternalInterface
ˡ . ExternalInterfaceHandler::Callback / ˝A ˡ й ˛ ¯ .

```
        static bool doChangeMesh = false;
        static wchar_t changeMeshFilename[MAX_PATH] = L"";

        ...

        if(strcmp(methodName, "MeshPath") == 0 && argCount == 1E
```

6.1 ⅞ ‡ ˚ ˝ C++ ∧ ╪ ˘ ▓ . / ė C++ ╪
ʃ ▲ Ỹ’ ╥ ▬ / Scaleform Ơ .
Scaleform C++ ⅞ ˥ ˘ (get) ẽ ♬ (set) ę /
⅞ ̈A ث ∧ .

Scaleform GetVariable’ SetVariable ╪ ⅞ ╥ ♪ ♠ ╬
w▬╪ . Tutorial\Section6.2 / ↑ ▓ й ♬ Scaleform Player ╪
Ơ Я ư HUD Ÿ ╪ ы ˘ (fxplayer.swf) F5 ы ∽ ⅄ ⅞

```
void GFxTutorial::ProcessEvent(HWND hWnd, unsigned uMsg, WPARAM wParam,
                               LPARAM lParam, bool *pbNoFurtherProcessing)
{
 int mx = LOWORD(lParam), my = HIWORD(lParam);

 if (pHUDMovie && uMsg == WM_KEYDOWN)
 {
        if (wParam == VK_F5)
        {
               int counter = (int)pHUDMovie
                            ->GetVariableDouble("_root.counter");
               counter++;
                 pHUDMovie->SetVariable("_root.counter",
                                        Value((double)counter));
               char str[256];
               sprintf_s(str, "testing! counter = %d", counter);
               pHUDMovie->SetVariable("_root.MessageText.text", str);
        }
 }
 ...
```

GetVariableDouble _root.counter > . ⅎ / ⊢ ▲ Ω ę
GetVariableDouble 0 . ⅄ ⅞ ȝ Ư > SetVariable
_root.counter / ᵉ ◗ . GFx::Movie / ’ ╬ ᵘ ̈A GetVariable’ SetVariable ╥ ,
÷ ˘ ▓ .

fxplayer ⅋ ▬ ‡ ї ╪ ▌╥ ╥ ⅞ ẽ ♬ ▓ . ╥ ♠
⅞ й ˘˘ ▓ . MessageText ⅞ й ȝ ⅄ / ∧ ę HUDText
ȝ ! ∧ Ư / ∧ . _root.counter > ᵘ ▫ ╪ Ỹ ̈E ę SetVariable
ʃ ⅞ ╥ ‰ ╪ ⅞ / Ơ .

╥ ♠ ⁄.. й ← ↻ ė TextField.text ё♬ ˅ ↑ GFx::Movie::Invoke ∧ ⁄.. ← ⁄.. ث ┤╪ (▫ "E Ɔ 6.2.2 ˘). ♠ ⁄.. й ▌╥╥ ˥ β . ˘ , ⁄.. ← ; β . ▬ Scaleform ₨ ▌ > ´ ụ – ↻ ᵘ / "E Ũ ◗ ▓ .

HUD ⁄.. > ← SetVariable

/ "A ¸ Ợ ᵘ ▫ ╪ ↑ ▫ ♪ ∧ .

SetVariable "sticky" ▒ ė … GFx::Movie::SetVarType / "E ╪ ▓ . ╪ ▌ ╬ / ᾤ Ỹ "E Ὦ ← . ɵ ‡ , ⁄.. й ╥ _root.mytextfield ∫ / "A 3 ▌ Ỹ "E Ὦ ← ủ ˥ ▫ . ᵣ – SetVariable("_root.mytextfield.text", "testing", SV_Normal) ∫ Ỹ "E ╬ ▌ 1 / "A ∧ ȝ , ▒ ╪ ᾤ ∫ ' ' ┤╪ . ∧ ╪(> ╬) SV_Sticky ∧ ‴ ȝ ∨ / / ˘ _root.mytextfield.text > / ᵣ ♠ ˘ , ▌ 3 / "A ˣ ┤╪ . ╪ C++ ╪ ∫ ┤ ┤╪ . ■ ♠ SV_Normal ╪ SV_Sticky / ♠ ╪ ḃ ȝ SV_Normal Ợ ˣ ↓ .

SetVariableArray ╥ ▬ ⁄ ♣ . ╪ ♠
▬ й ᶜ Ł ‡ ¯ ▬ ⁄ Ọ ▒ . ╥ й
GFxTutorial::InitGFx() ⁄ ⁄ ˘ _root.SceneData й > ἓ ♬ .

```
// Initialize the scene dropdown
Value sceneData[3];
sceneData[0].SetString("Scene with shadow");
sceneData[1].SetString("Show shadow volume");
sceneData[2].SetString("Shadow volume complexity");
pUIMovie->SetVariableArraySize("_root.SceneData", 3);
pUIMovie->SetVariableArray(Movie::SA_Value,
                           "_root.SceneData", 0, sceneData, 3);
```

Tutorial\Section6.1 ⁄ ▒ й ⁄ ♠ ╬ ExternalInterface ɵ ˹ ˘ ▒ ╪
w ʃ Ł ‡, w ʃ , ◗ ʒ ShadowVolume ╬ ˹
ˆ ˄

```
        GFxPrintf("_root.OpenMeshPath returns '%d'\n", (int) retval);
    }
    else if (wParam == VK_F7)
    {
       const char *retval = pHUDMovie->Invoke("_root.CloseMeshPath", "");
       GFxPrintf("_root.CloseMeshPath returns '%d'\n", (int) retval);
    }

    ...
```

openMeshPath / ٪. й ▌1 / ▓ ৩ ᴗ / ٪. ث╬

ʃ ╥ ৸ ▌ ◓ Ỹ ▫ ▫ ы ৭ ╬ . Invoke Ọ ▓ ‡

■ ♠ ╬ / ᾧ Ὦ ٪. ث ৸ ৭ ╬ , ╪ ← Scaleform

/ / . ٪. ث ' ▌╪ ◓ Ỹ ˎ ↑ ' ╬ ♪ _ ৸ ╥

ы╪ ╪ ৩ Ὦ . ▌1 / ▓ ᵖ ö ٪. ৸

GFx::Movie::Advance ৸ ╪ ╪ ˎ ↑, true ἓ ♬

initFirstFrame GFx::MovieDef::CreateInstance ৸ ৩ ╪ .

╪ / ̈A Invoke ╥ printf ■ Ọ . ♣ ╥ GFx::Value Ọ ̈A

♠ ٪. ᴗ ▫ ৸ . InvokeArgs Invoke ■ ↑ ‡ ΐ╪ ╪

ᶜ / ˢ╬ Ł̂ va_list Ọ ♩ . Invoke InvokeArgs ╥

, ´ printf vprintf ╥ , ´ Ọ .

╪Ł б ᵖ C++ Ọ . UI ϰ SWF ر ■ ↑

‡ ΐ╪ ╥ ← , Ũ ╥ C++ Ọ ̈A ╬ ˢ╪ Ɔ╥ ΐ ˢ Б ٪. ө╪ ̈A

▓ –

ө ‡, ϰ Ọ ▫ ╬ ˢ ▌(MMOG) ╬ HUD, ᾧ HUD, ᴗ ᴗ ˽ ˢ

▓ ▓ . ˢ╪↑ ˉ ᾧ╪ ╪‡╥ ╬ / ̈A Ũ♩ ╪

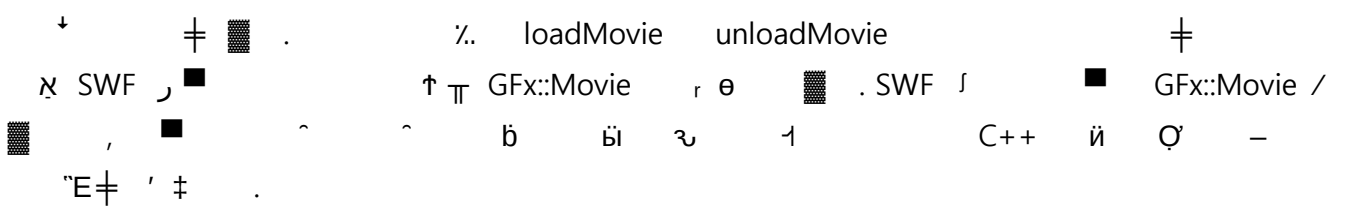
╪‡ / ˢ ᴨ ▓ . ╪‡ ᾧ╪ ˇˇ ▓ ← ᾧ╪ ╬

HUD / ̈A ᾧ HUD ╪ ˢ .

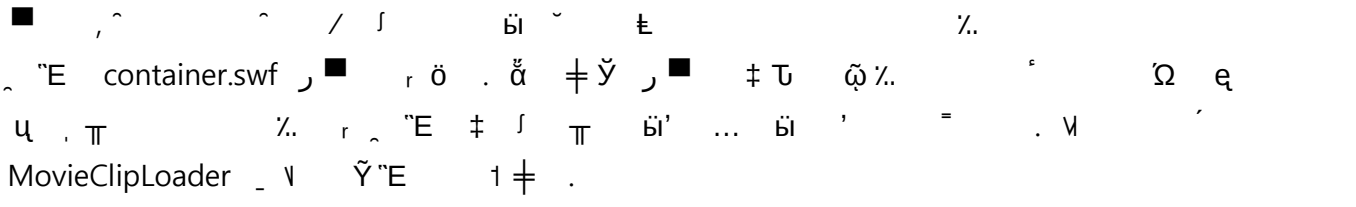
╪ ̈E ╬ ˢ╪ ╥ SWF ر ■ ↑ ‡ – ę ʃ ᵖ

ы ‡ – . ↑ ╪ ̈E GFx::Movie _ ৸ Ọ╪╥ C++ й

▬̈E ৭ / ▬‰ ᵖ .



MMOG ╥ / ΆA, ╬ _global.inventory ٪. ج ▓ . SWF
╬ ˅ ╪ אַ † ы ȝ ╪ ╥ Ɔ ˥ ق̣ ╪ ḥ ˭ . SWF
╬ ˅ ╪ אַ † ٪. ╥ unloadMovie Ơ ΆA Ł ˎ ২, _global.inventory
╬ ˅ ╪ / ė ♣ ق̣ Ơ Ũ ɔ ق̣ ▓ ˭ .



container.swf ر ■ ˌ ö . ǎ ╪ Ў ر ■ ‡ Ʊ ق̣ ٪. ˀ Ω ę
ц ˌ ╥ ٪. ˌ ˆ ¨E ‡ ∫ ╥ ы' … ы ' ˭ . ۷ ´
MovieClipLoader _ ۷ Ỹ ¨E ˥ ╪ .

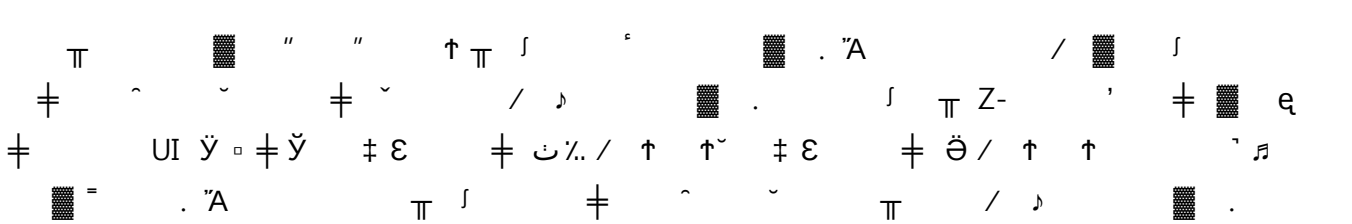
```
var mclLoader:MovieClipLoader = new MovieClipLoader();
```

╪ Ɔ / ΆA Ơ ٪. / ▫ ¨E ♬ ˘ ΆA
˘ . ∫ ╪ _ ۷ ˝ ٪...♬ ы ▓ .

```
// Load a file into a specific level

function LoadFlashLevel(url, level)
{
    trace("LoadFlashLevel(" + url + ", " + level + ")\n");
    mclLoader.loadClip(url, level);
}

//
// Load a file into a named object
function LoadFlash(url, objectName)
{
    trace("LoadFlashLevel(" + url + ", " + objectName + ")\n");
    var container:MovieClip = createEmptyMovieClip(objectName,
                                                   getNextHighestDepth());
    mclLoader.loadClip(url, container);
}
```



LoadMovieLevel ∫ ⅞...♬ / ы ↿ Z- า Ɔ ♠
♬ ╥ ▓ ◗ ♩ ╪ ▓ . ∫ Ɔ _levelN.variableName (, _level6.counter)
♪ ╪ .

⅞...♬ ╪ ╥ / ∫ ы LoadMovie Ơ ȝ ○ ╬ ˅ ╪ ╥
ˆ w Ơ ▓ . ∫ ⅞. ˆ w ▓ ę ╪ / ƒ ج
▓ (, _root.inventoryWindow.widget1.counter).

╥ ∫ ɵ _root า ˘ ▓ . ᵣ – ∫ ⅞...♬
า ы ȝ , _root า ˘ . ɵ ‡ , ∫ 6 ы ȝ _
root.counter _level6.counter ╥ ¯ w . ▫ ╥ Ɔ w ∫ ╥
ᵉ LoadMovieLevel ⅞...♬ า ы ᵉ ♬ Ũ ♠ ▬ .

LoadMovie _root.myMovie / ы ■ ∫ Ł ▬ Ώ , ╪
_root.counter ⅞. ╥ ╪ / ▓ ⅄ ˘ ২ ᵘ ╪ . ⅞. ˆ w ˆ ¨E ∫
this:_lockout = true ε̈ ♬ ‡ – . Lockroot ⅞. ╥ ¨E _root ╥ ᵖ ö
w ¨A ∫ ╥ ⅞. ↿ ╪ ╥ ⅞. ↿ ╪ ᾧ . ⅞.
, ∫ / ▫ ¨E Ơ ‡ ᵘ ¨A ˘ .

¨A ∫ ‡ ˅ ˭ w ╪ ˣ / , ■ GFx::Movie Ɔ / ¨A ▬ ⅄ ╬ ∫ Ũ ╥
/ ♪ ▓ ę ˆ Ũ / ¨A ▬ ╪ , ╪ ○ ╬ ˅ ╪ ╥ Ỹ ¨E
Ũ ᾧ ▓ ˭ .

Container.fla ˝ ʃ ᵖ ⊓ ╪ Ώ ∫ … ы ˘˘
▓ (, Ơ ▫ ȝ ᾷ ⅄ Ł ▓).

LoadMovie ˝ LoadMovieLevel ╪ ২, ∫ ╥ ы ′ . loadClip
∫ ╥ ы ᵣ ▬ ę , й / ¨A ▬‰ ´ . ╥ ‡ ǐ ╪ ╪
∫ ╥ ы … Ł Ω ᾧ – ᵉ (, ♠ Ũ ↿),
MovieClipLoader ╥ Ў Ơ ▓ . Container.fla ⅞. Ɔ ╥ ╪ ⅞.
↿ ˎ ↑ C++ ‡ ǐ ╪ ╥ ▬ ╬ ˅ ╪ ⅄ Ỹ ¨E ╥
placeholder ˆ ˅ ▓ .

```
// Define callback functions to report events such as:
//   1. Started loading movie
//   2. Loading a movie failed
//   3. Loading a movie finished
//   4. Progress updates
```

```
//
// These callbacks are necessary because a movie does not
// necessarily load as soon as LoadMovie() returns.
//
// Currently the callbacks only print debug information.
// An application that needs to act on these events should
// either make ExternalInterface calls to notify the C++
// application or handle the events in ActionScript.

var mclListener:Object = new Object();

mclListener.onLoadError = function(target_mc:MovieClip, errorCode:String,
                                   status:Number)
{
   trace("Error loading image: " + errorCode + " [" + status + "]");
};

mclListener.onLoadStart = function(target_mc:MovieClip) : Void
{
   trace("onLoadStart: " + target_mc);
};

mclListener.onLoadProgress = function(target_mc:MovieClip,
                                      numBytesLoaded:Number,
                                      numBytesTotal:Number) : Void
{
   var numPercentLoaded:Number = numBytesLoaded / numBytesTotal * 100;
   trace("onLoadProgress: " + target_mc + " is " +
        numPercentLoaded + "% loaded");
};

mclListener.onLoadComplete = function(target_mc:MovieClip,
                                      status:Number) : Void
{
   trace("onLoadComplete: " + target_mc);
};

// Register the listener with the MovieClipLoader
mclLoader.addListener(mclListener);
```

The code in Tutorial\section6.2 / ▓ й ♬ d3d9guide.swf fxplayer.swf / container.swf ┌ ┤ / ы . F8 ы ȝ HUD ы ˘ F9 ы ȝ ʒ ╬ UI ы

```
void GFxTutorial::ProcessEvent(HWND hWnd, UINT uMsg,
                               WPARAM wParam, LPARAM lParam,
                               bool *pbNoFurtherProcessing)
{
      ...

  else if (wParam == VK_F8)
  {
      bool retval = pShellMovie->Invoke("_root.LoadFlashLevel",
                                        "%s, %d", "fxplayer.swf", 5);
```

```
        GFxPrintf("_root.LoadFlash returns '%d'\n", (int) retval);
    }
    else if (wParam == VK_F9)
    {
        bool retval = pShellMovie->Invoke("_root.LoadFlashLevel",
                                          "%s, %d", "d3d9guide.swf", 6);
        GFxPrintf("_root.LoadFlash returns '%d'\n", (int) retval);
    }
    else if (wParam == VK_F2)
    {
        bool retval = pShellMovie->Invoke("_root.UnloadFlashLevel",
                                          "%d", 5);
        GFxPrintf("_root.UnloadFlash returns '%d'\n", (int) retval);
    }
    else if (wParam == VK_F3)
    {
        bool retval = pShellMovie->Invoke("_root.UnloadFlashLevel",
                                          "%d", 6);
        GFxPrintf("_root.UnloadFlash returns '%d'\n", (int) retval);
    }
}
```

╥ й HUD 5 ы ˘ ɾ ╬ ╬ ˅ ╪ 6 ы .
⅞ ◗ ȝ ╥ Z ' ╪ ▓ . ■ GFx::Movie Ɔ / ̈A, Ũ ̈ / ▓
ʃ ʍ ╥ ʃ / . ы ч ʃ ╬ Container.swf
0 ы .

Ɔ _level й ╥ w ̈ . ɵ ‡ , _level6 / ▓
ɾ ╬ ╬ ˅ ╪ _level5.MessageText.text ̈Ω HUD ╥ ⅞ й / ♪ ♪ ╪
. β ȝ , ╥ ʃ _global й (ʃ ɒ _global.counter / ♪ ╪
) ╥ ♣ ˆ / ▓ ╪ ˆ ▓ . ╪ ʃ ы
Ω ৩, _global ╪ ˆ / ▓ ▌ Ω ◗ ˩ ╪ ▓ . ʃ ɾ ɒ Ɔ
╪ ę ■♬ _global ╪ ˆ ˘ . _level, _ground ⅞ ╪
ˆ / ▫ ̈E ♬ ‡ ᵘ ̈A ˘ .

‡ ǐ ╪ ˘ F8 HUD ы . F9 ɾ ╬ ╬ ˅ ╪
ы . ɾ ╬ ╬ ˅ ╪ ы Ó ╪ ↑ ↑ Ỹ ╬ . ╪ Ɔ ◗
ɴ ‡ ⅞ ث ы ╪ ˢ ৩ ᵘ ╪ . ╪ ̈E SWF ر ■ Scaleform
ر ■(7 ◗) Ç ч ˘ Scaleform ╥ ɖ ɵ й ы Ơ ̈A ˩ Ɔ ˆ ث ⅞
ر ■ ♠ ◓ ̈Ω ▓ (¸ ▫ ˅ ╪ / ▓ ث ⅞ ⅞ ȝ ♬
¸ ˘).

HUD ৲ F8 ы ˘ 6.2 / ৲ ৲ F5 ы ৲

╪Ũ ▬ Ω . SetVariable _root.counter ˘ ↑ HUD ◓ _level5

╪ ˆ / ы ь ৲ ' ¯ ╪ – .

```
void GFxTutorial::ProcessEvent(HWND hWnd, unsigned uMsg, WPARAM wParam, LPARAM
                                        lParam, bool *pbNoFurtherProcessing)
{
        int mx = LOWORD(lParam), my = HIWORD(lParam);

        if (pHUDMovie && uMsg == WM_KEYDOWN)
        {
                if (wParam == VK_F5)
                {
                        int counter = (int)pHUDMovie->
                                                GetVariableDouble("_level5.counter");
                        counter++;
                    pHUDMovie->SetVariable("_level5.counter",
                                              Value((double)counter));
                        char str[256];
                        sprintf_s(str, "testing! counter = %d", counter);
                        pHUDMovie->SetVariable("_level5.MessageText.text", str);
                }
        }

        . . .
```

_level5.counter 6 ы d3d9guide.swf / ▓ ⅞ ♪ ♪ ╪

ę ╪ ╬ ˅ ╪ ᵖ ⅄ / C++ й ' ╪ ♪ ╪ .

‡ ї ╪ ˘ , F8 HUD ы F5 ৲ . F2

HUD ╥ ы Ł ˘ , F8 HUD ◓ ы F5 ৲ .

HUD … ы ৯, ৲ > ╪ Ọ ę 0 ৲ ⅞ ◓ ▬ . ╪ ৲

_level5 / ᴾ ◗ ৯ ᵘ ╪ . _level5.counter _global.counter ₊ ◓ ▬ . ♬

_global / ᴾ ◗ ৯ ᵘ / ∫ ы ˘ … ы ৯ / ৲ ᴾ ◗ .

৲ 3D / Ä ▫ ৮ Ọ ˘ ▒ . ╪ Ọ ȝ ⁼

৲ ȝ ج / ▓ . ¬ ' ৲ ■ ৲ ◗ ȝ

ɾ ⅞ / ♠ ▓ . ɵ ‡ ╪ " ⁼ ▌ Ɔ й " ɾ ɵ ▓ .

ḫ ᴾ Flash Studio / Ä й SWF ɾ ■ ▬Ë ˘ ╪ SWF ɾ ■ ৲ / .

╪ ৲ ♠ ◗ ȝ ɾ ⅞ / ♠ .

╪ Ш ╽ ⁄ Ä ⁼ ╪♣ Ш ╽ ╥ UI Ö ȝ ⁄ ˚ .
، ы ’ ╪ȝ UI ’ ╪ ╪⁄ ƒ Ọ .

╪Ł ╪ Ł ╥ ▬ ⁄ Ọ ̃ ̣ ˚ .

1. ♣╥ ᵖ ö Ш ╽ ⁄ Ä Ỡ cell.x ◗ȝ ر ■ . ╪ ر ■ ’ ،
ᴝ ◗ ج ♩ (vertex) ! ج Ỳ ╬ ˘ ᴝ ♬
ˀ . ╪ ر ■ ů ˥ ȝ Л ȝ ᵖ ⁄ ¯ ᴝ (cellwall.jpg) Ọ ‴
ᵀ Ọ ▓ . ╪ Ш ╽ ⁄ Ä UI Ö ⁄ ˘ . ḫ ᴾ Ö
Ọ ᴝ ᵣ ө˘ ر ■ ╪ cellwallRT.jpg ˘ ▒ .
MeshMaterialList ╪ Ö ∽ ╪ ᴝ w ♬ .

```
Material {
   1.000000;1.000000;1.000000;1.000000;;
   40.000000;
   1.000000;1.000000;1.000000;;
   0.000000;0.000000;0.000000;;
 TextureFilename {
"cellwallRT.jpg";
}
```

2. ╪♣ Ш ╽ ⁄ Ä β ö╪ cell.x ر ■ DXUT ▌ Ɔ ⁄ Ä
ˆ ᵘ Ä(parsing) . DXUT cell.x ر ■ ˆ ᵘ Ä Ó ♩ ˭ ╬

(material) CreateTexture Render Texture DXSDK DXUT .

```
pd3dDevice->CreateTexture(rtw,rth,0,
   D3DUSAGE_RENDERTARGET|D3DUSAGE_AUTOGENMIPMAP, D3DFMT_A8R8G8B8,
        D3DPOOL_DEFAULT, &g_pRenderTex, 0);

g_Background[0].m_pTextures[BACK_WALL] = g_pRenderTex;
ptex->Release();
```

3. GFx AdvanceAndRender .

   a. .

```
pd3dDevice->GetRenderTarget(0, &poldSurface);
pd3dDevice->GetDepthStencilSurface(&poldDepthSurface);
```

   b. (surface) .

```
ptex->GetSurfaceLevel(0, &psurface);
HRESULT hr = pd3dDevice->SetRenderTarget(0, psurface );
```

   c. Display a .

SWF ر ■ ♪ ы . ة ۵ ٪ ، ۸ ╬ SWF Ɔ / ♪ ▒ ،
= ▌ Ω˘ ˋ ' ▒ ৩ ᵘ / ، ۸ / ╪ ৭ SWF Ơ ↑ ╪ đ .
↑ ˢ ش / Ä SWF ǎ ৸ ˢ ↑ SWF ر ■ ы Ë / '
s ˘ ы / ↳ ৩ ᵘ / ♠ ♥ Ω Ω .

GFxExport SWF ر ■ ы ♠ ˢ ƶ ৭ ج ة . ♣ ৭
Ó, = ▌╥ ▫ ╪ ╪Ç ' ╪Ç ╥ ر ■ ৸ ৸ . ╪Ç
♠ ы' ▌ ɾ ᵖ ♥ – DXT ৭ ৸ ╪ ▒ DDS ر ■
▒ . Ɔ ◗ ٪ث ◗ ˘ ٪ ŧ ٪ث Ơ – ˓ ٪ث ৭ ♠ ♠
Ç ´ ụ ▒ . GFxExport ৸ Ä ৸ .

Scaleform ر ■ ˢ ↑ . GFxExport ج ة β / Ä ᵘ Ä ،
. d3d9guide.swf fxplayer.swf ر ■ Scaleform ˢ ƶ ▫ .

```
gfxexport -i DDS -c d3d9guide.swf
gfxexport -i DDS -c fxplayer.swf
```

gfxexport.exe C:\Program Files\Scaleform\$(GFXSDK) Ÿ / ▒ . ╪ ʃ ‡ ˝
Tutorial\Section7 ╥ convert.bat ر ■ / ▒ .

-i ╪Ç ˢ ƶ ♬ ╥ ę ╪ ˓ DDS . DDS DirectX ت / ė ◗ ■ ♠ ╬
╬ ╪ DXT ৭ ৸ ╪ ৩ ᵘ ╪ ę, ■ ♠ 4 ♬ ▌ ৭
ɾ ᵖ ♥ – ˥ .

–c ৸ ♬ ╥ . Scaleform ر ■ Ɔ ٪ ǎ ৸ ˣ ╪ ৸ . ╪Ç
৸ ė ╪Ç ৭ ˢ ƶ' DXT ৸ / ƒ .

–share_images Ä SWF ر ■ Ɔ ■ ╪Ç ۵ Ɔ Ä ↑ ╥ ˆ Ơ ˣ
ы Ω ɾ ᵖ Ơ ♥ – .

Tutorial\Section8 Ɔ ╥ ShadowVolume GFx::Loader::CreateMovie / Ä ы ر ■ ʃ
˓ ♬ ♣ ╪ .

╪ ˝A Scaleform ╥ ˛ / ' ˘ ▒ . ▫˝E Ɔ
Ọ˙ ˘ ȝ .

- ˝ ▌ Ɔ / ˝A ȷ ϟ. Scaleform Player SWF to Texture SDK ˘ ,
  Gamebryo™ ᵖ Unreal® Engine 3 ᵖ ˘
- ˛ ▫ FAQs / ˝A ˝E Ọ˙
- ᵘ ˝A ˂ ╪ / ˝A Scale9Grid Overview ˛ ╥ Scale9 ᵘ ˝A
- ᾶ ы: Scaleform ˝ SWF ر ■ ╥ ы / Ȯ ˛ ᾷ GFx::FileOpener
  Ũ˙ ȝ ȷ ᵖ ↑ ˝ ▌ ▫ ' ▫ / ˝A ♪ ы╪ .
- GFx::ImageCreator ᾶ ╪Ç ϟ
- GFx::Translator