

Autodesk® Scaleform®

CLIK 入門

本文介紹了如何使用和有效運用 Scaleform®通用精簡介面工具包（Common Lightweight Interface Kit，CLIK™）

作者: Matthew Doyle
版本: 3.0
最新修訂: 2010 年 8 月 18 日

Copyright Notice

Autodesk® Scaleform® 4.2

© 2012 Autodesk, Inc. All rights reserved. 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.

Disclaimer

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.

Autodesk Scaleform 聯繫方式：

文檔 | CLIK 入門

地址 | Autodesk Scaleform Corporation
6305 Ivy Lane, Suite 310

Greenbelt, MD 20770, USA

網站 www.scaleform.com

郵箱 info@scaleform.com

電話 (301) 446-3200

傳真 (301) 446-3199

目 錄

1. 概要	1
1.1 最低軟體配置	1
2. Scaleform CLIK 和 Scaleform 面板安裝和配置.....	2
2.1 增加 Scaleform CLIK 類路徑到 Flash CS3	2
2.2 添加 Scaleform CLIK 類路徑到 Flash CS4	3
2.3 增加 Scaleform 启动面板	5
3. 用元件創建一個簡單的完整介面功能表.....	6
3.1 創建功能表 Shell	8
3.2 添加主功能表導航控制功能.....	11
3.3 監聽 Options Screen 按鈕	14
3.4 增加可選步驟難度	15
3.5 增加視頻設置核取方塊和選擇框.....	17
3.6 增加音量調節條.....	20
3.7 增加 OK 和 Cancel 按鈕	22
4. 用 ActionScript 增加功能.....	23
4.1 增加難度動作選項的難度等級	23
4.2 设计 aaGroup 按钮群	24
4.3 设置选项的初始状态.....	25
4.4 退出選擇介面	26
5. 功能表介面.....	27
5.1 繪製選擇按鈕背景	28
5.2 繪製捲軸背景	32
5.3 繪製選擇按鈕和核取方塊背景	35
5.4 描繪難度選擇框皮膚背景.....	36
6. 結論	38

1. 概要

Autodesk® Scaleform® CLIK™ 使開發者快速有效得原型化遊戲的介面功能表，然後實現和應用這些原型。本文包括了利用 **Scaleform CLIK** 和 **ActionScript™** 創建一個自定義遊戲介面的基本步驟，以及幾個最典型的元件和介面創建實例。文中還描述了創建過程的每一個步驟。

說明： 在 **Flash** 中完成相同的工作會有不同的方式。開發者可以找到並使用那些比本教程中所採用的方法更適合自己的替代工作流。

1.1 最低軟體配置

- Scaleform 3 及以上版本
- Adobe® Flash® CS3 及以上版本
- Adobe Photoshop® CS3 及以上版本

2. Scaleform CLIK 和 Scaleform 面板安裝和配置

Scaleform CLIK 類似於 Adobe Flash 的設計面板以提高工作流程。使用 Scaleform CLIK 元件和工作流增強面板的第一步為確認 Scaleform CLIK 類的路徑已正確添加到 Flash 應用環境中。然後使用 Adobe 擴展管理器來安裝擴展面板。

安裝 Creative Suite® 3 (CS3)和 CS4 略有不同。我們為 CS3 和 CS4 兩者提供了安裝指令。

2.1 增加 Scaleform CLIK 類路徑到 Flash CS3

Flash 類必須為 Scaleform CLIK 類設置路徑，以對授權環境開放。

1. 啟動 Flash CS3。
2. 點擊 *Flash* 功能表頂部，在下拉功能表中選擇參數。
3. 在目錄下選擇 *ActionScript*。
4. 在 *Language* 選項中點擊 *ActionScript 2.0* 設置。
5. 在 *ActionScript 2.0* 設置對話方塊中，通過[+]按鈕增加一個新路徑。

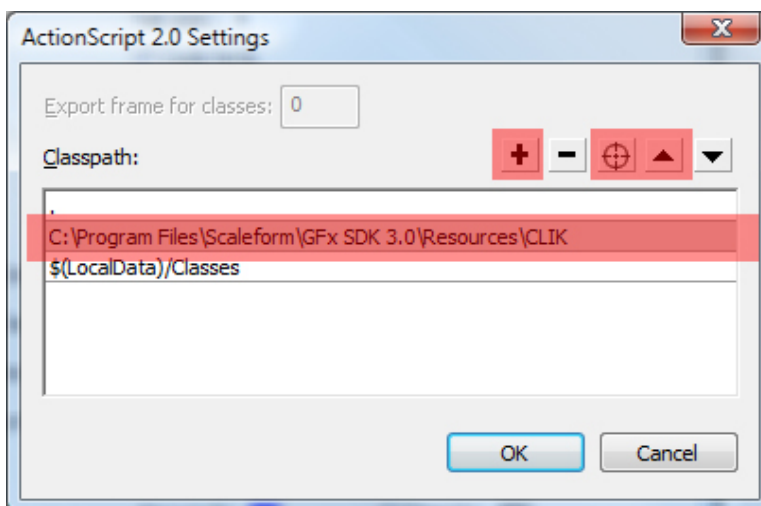


圖 1：CS3 中的 *ActionScript 2.0* 類路徑(Windows)。

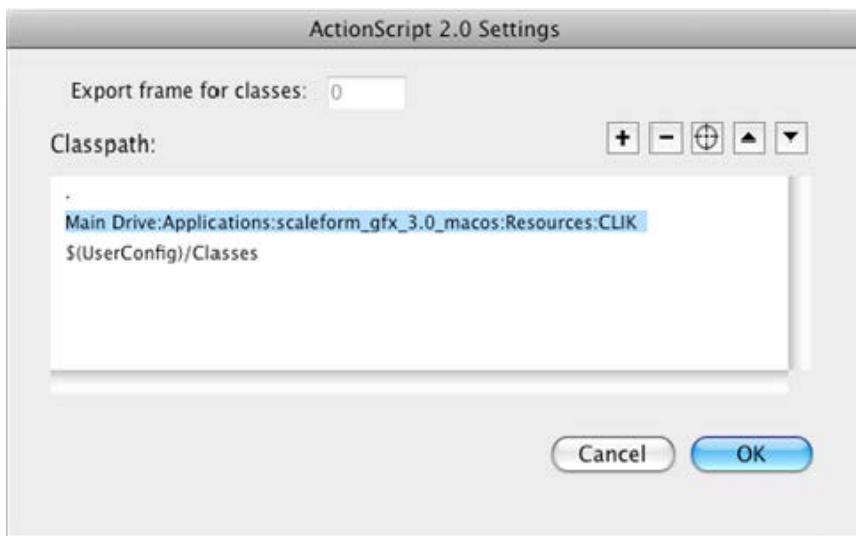


圖 2：CS3 中的 **ActionScript 2.0** 類路徑(Mac)。

6. 點擊十字交叉圖示流覽 CS3 中路徑,選擇 Scaleform CLIK 安裝目錄。其 Windows 預設安裝路徑為:C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/,其蘋果系統預設安裝路徑為 scaleform_gfx_4.2_macos/Resources/AS2/CLIK 在 Scaleform 4 SDK 目錄。
7. 返回 ActionScript 2.0 設置對話方塊,在 Classpath 下選擇增加路徑。使用 Arrow 按鈕移動到\$(LocalData)/Classes 行之上。如果是評估系統的話,不用使用 Arrow 按鈕,可以直接把\$(UserConfig)/Classes 行拖過來。來確保 Scaleform CLIK 類的索引位於內嵌類之前。如果不進行這步操作,則內嵌 Adobe 元件將替代 Scaleform CLIK 元件。
8. 點擊 OK 關閉 ActionScript 2.0 設置窗口,然後點擊 OK 關閉 Preferences 窗口。

2.2 添加 **Scaleform CLIK** 類路徑到 **Flash CS4**

Flash 類必須為 Scaleform CLIK 類設置路徑,以對授權環境開放。

1. 啟動 Flash CS4。
2. 點擊 *Flash* 功能表頂部,在下拉功能表中選擇參數。

3. 在目錄下選擇 *ActionScript* 。
4. 在 *Language* 選項中點擊 *ActionScript 2.0* 設置 。
5. 在 *ActionScript 2.0* 設置對話方塊中，通過[+]按鈕增加一個新路徑 。

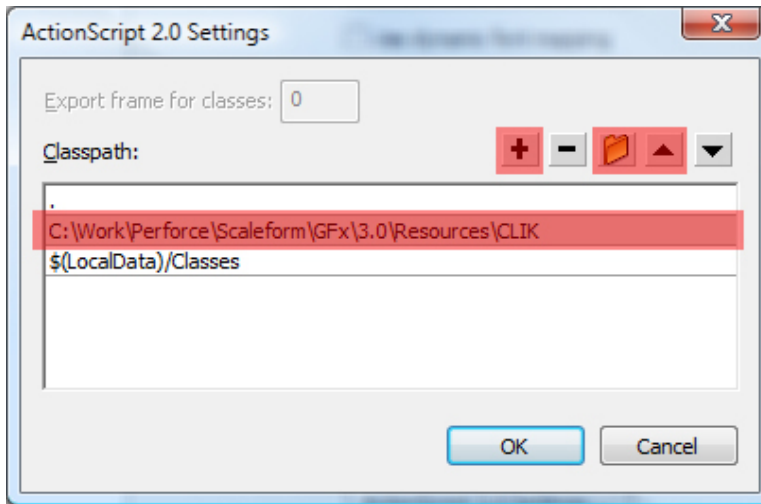


圖 3: CS4 中 *ActionScript 2.0* 類路徑視窗(Windows)

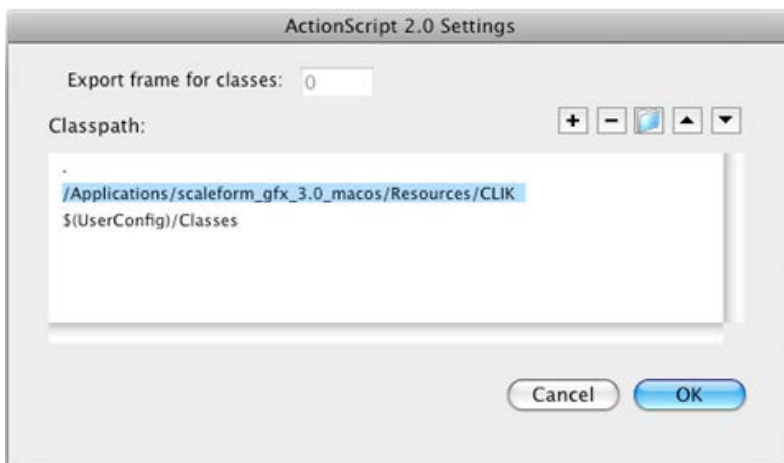


圖 4: CS4 中 *ActionScript 2.0* 類路徑視窗 (Mac)

6. 點擊十字交叉圖示流覽 CS4 中路徑,選擇 Scaleform CLIK 安裝目錄。其預設安裝路徑為:C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/,然後在蘋果系統,元件在 scaleform_gfx_4.2_macos/Resources/AS2/CLIK/ 。

7. 返回 **ActionScript 2.0** 設置對話方塊,在 **Classpath** 下選擇增加路徑。使用 **Arrow** 按鈕移動到`$(LocalData)/Classes` 行之上。如果是評估系統的話,不用使用 **Arrow** 按鈕,可以直接把`$(UserConfig)/Classes` 行拖過來。來確保 **Scaleform CLIK** 類的索引位於內嵌類之前。如果不進行這步操作,則內嵌 **Adobe** 元件將替代 **Scaleform CLIK** 元件。
8. 點擊 **OK** 關閉 **ActionScript 2.0** 設置窗口。
9. 點擊 **OK** 關閉 **Preferences** 窗口。

2.3 增加 **Scaleform** 啟動面板

Scaleform 啟動面板為從 **Flash** 創作環境直接向 **Scaleform** 播放機發佈 **SWF** 檔提供了一個方便快捷的方法。如果您還未安裝該面板,請參考 **Scaleform 4** 入門檔中第 2.4 到 2.6 節中的安裝與使用說明。

3. 用元件創建一個簡單的完整介面功能表

本節將詳細介紹創建一個遊戲功能表的必要步驟。功能表包括兩個介面：主功能表和選擇功能表。主功能表擁有一個按鈕使用戶轉移到選擇介面。功能表工作面板在選擇介面當中，其中包括一系列複雜的設置選項，視覺效果的視頻選項和使用內嵌核取方塊和選擇框的防混淆按鈕，以及一個音量控制器。

本指南的資源位於在Scaleform 4.2 SDK 目錄的 CLIK 安裝目錄下:

Resources/AS2/CLIK/tutorial

Resources/AS2/CLIK/tutorial/art_assets

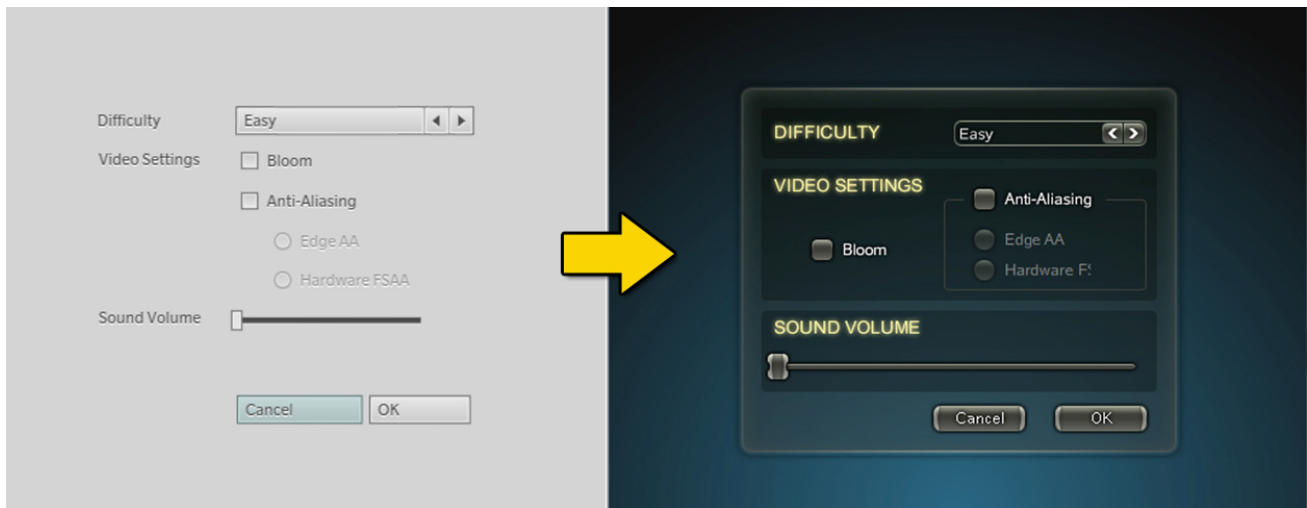


圖 5: 無皮膚和有皮膚背景最終選擇介面

本指南中的用戶介面創建有如下過程：

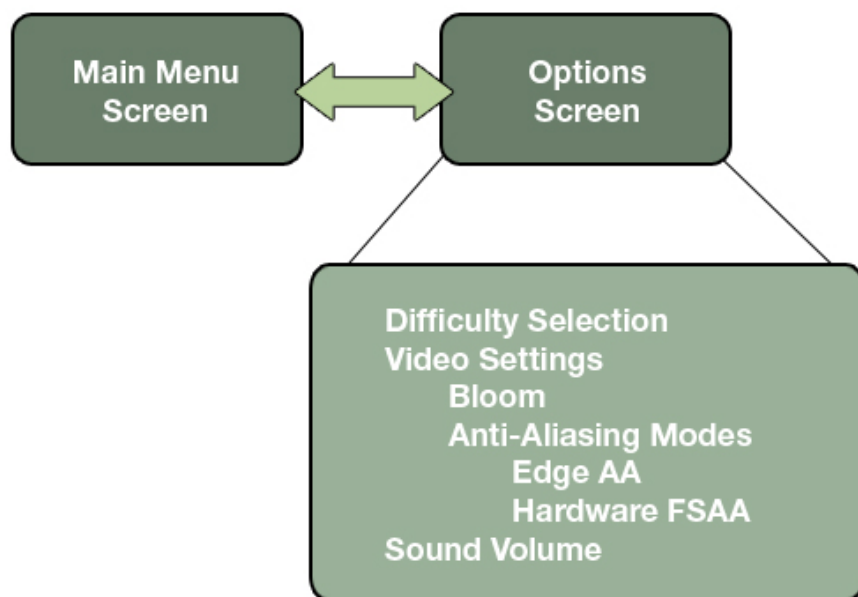


圖 6: 完整介面功能表創建流程

3.1 創建功能表 *Shell*

第一步為創建為功能表創建多種類型的幀和關鍵幀。這裏介紹了創建的大致步驟。

1. 創建一個新的 Flash（ActionScript 2.0）文檔。
2. 確保文檔的發佈設置為 Player 8，從頂部功能表選擇 *File*，在底部下拉清單中選擇 ‘Publish Settings’ 然後點擊 *Flash* 標籤。
3. 設置 Version 下來框為 ‘Flash Player8’。
4. 設置 *ActionScript version* dropdown 為 ‘ActionScript 2.0’。
5. 點擊 *OK* 來關閉 *Publish Settings* 窗口。

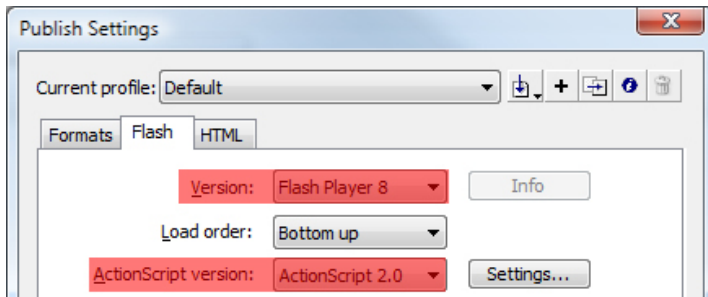


圖 7：發佈設置窗口

6. 在主文檔時間軸上選擇當前圖層，當前標籤為 *Layer 1*，雙擊 *Layer 1* 將其改名為 ‘scene’。如果名稱並不重要，這種方法命名最為合適。
7. 點擊 *Insert Layer* 按鈕創建一個新的圖層，命名此幀圖層為 ‘actions’。同樣圖層的名稱並不重要，只要容易記憶即可。

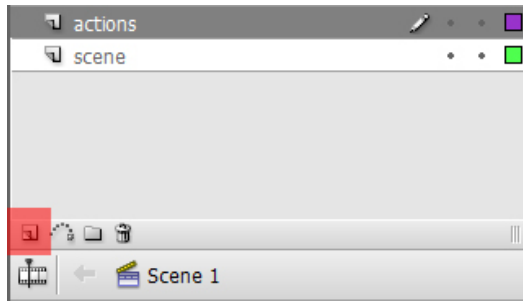


圖 8: 插入圖層按鈕

8. 在幀 10 上為 *action* 和 *scene* 兩個層添加第二個關鍵幀。選擇 *scene* 層，然後點擊幀 10 上的時間軸。從彈出功能表選擇 'Insert Keyframe'。該關鍵幀將被用來創建功能表的 Option 介面部分。在 *action* 層上重複該步驟。

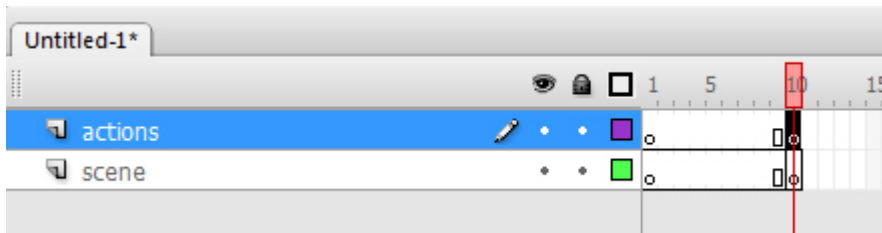


圖 9: 在第 10 幀上添加關鍵幀的主功能表時間軸

9. 在 *scene* 圖層 'mainMenu' 的幀 1 上標記為關鍵幀。可以通過選擇關鍵幀並在 *Properties* 標籤中輸入名字來實現。該標籤可以被代碼用來告知 Scaleform "goto" mainMenu 幀上的時間軸。第 1 幀將被用來創建功能表的 Main 介面。

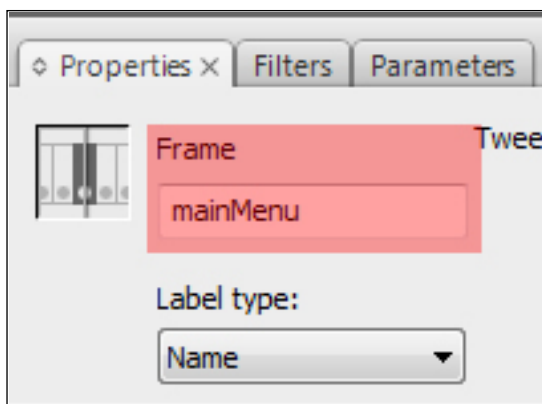


圖 10: 關鍵幀 1 命名為 'mainMenu'

10. 在 *scene* 層 ‘optionsScreen’ 的幀 10 上標記為關鍵幀。該標籤可以被代碼用來告知 Scaleform “goto” 時間軸上的 *optionsScreen* 幀。
11. 在 *actions* 圖層選擇第 1 幀。打開 *Actions window* (F9)並在 *actions* 圖層的幀 1 和幀 2 輸入的第一行輸入“stop();”（無雙引號）。該代碼告知播放器在此幀位置停止播放。如果該位置沒有代碼，播放器將繼續播放直到出現 **stop** 命令位置，或者直到時間軸結束處，在本例中為第 10 幀。從 *actions* 圖層選擇第 10 幀，添加如上代碼。
12. 此時，正好可以保存 Flash 文件。

3.2 添加主功能表導航控制功能

本節描述了如何在 *Main Menu* 增加一個 *Options* 按鈕，用來使用戶進入 *Options Screen*。

1. 打開位於 *Scaleform 4 SDK* 目錄中的 *Resources/AS2/CLIK/components* 目錄下的 *CLIK_Components.fla* 檔。
2. 首先添加功能表標題。進行該操作須只需點擊 *CLIK_Components.fla* 文件的中 *Library* 面板中的 *Label* 元件並從彈出功能表選擇 *Copy*。

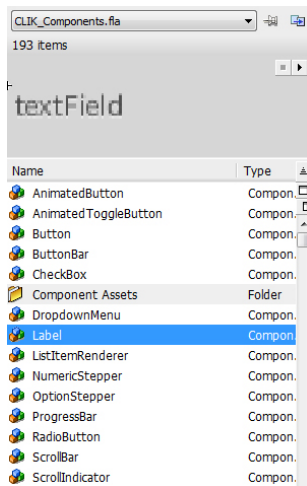


圖 11:library 面板和 label 元件

3. 不要關閉 *CLIK_Components.fla* 文件，後面還需要用到。返回到在創建指導功能表的原始 *Flash* 文檔。
4. 選擇 *scene* 圖層的第 1 幀。
5. 在 *Library* 面板中，只要點擊空白區域並選擇 *Paste* 以拷貝 *Label* 元件到文檔庫。
6. 點擊並拖動 *Library* 面板中的 *Label* 元件至場景中，適合位於中間，臨近頂部。
7. 為新創建的標籤打開參數符，在 *text* 域中輸入 '*Main Menu*'。這個區域中的內容在播放器運行時將被顯示出來。**注意：**在 *CS3* 中，符號位於 *Properties* 和 *Filters* 標籤旁邊。在 *CS4* 中位於 *Component Inspector*，默認情況下為關閉。按[Shift + F7] 組合鍵則顯示，或者從彈出功能表選擇 *Window – Component Inspector*。

注意：這些參數只有 SWF 文件被發佈並播放時才會被顯示出來。

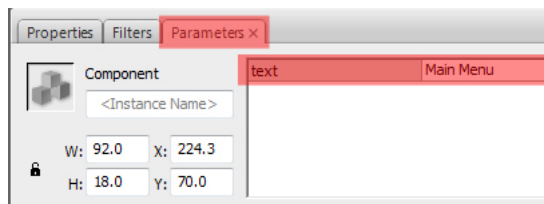


Figure 12: CS3 中參數標籤位於 properties 標籤旁邊

8. 選擇第 10 幀並重復步驟 6 和 7 創建一個 Options 介面的標題 - 在當前文本域輸入 'Options' 。
9. 返回 *CLIK_Components.fla* 文件，只要點擊文件 *Library* 面板中的 *Button* 元件並再次選擇 *Copy* 。

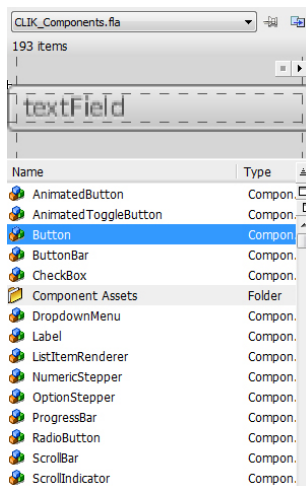


圖 13：library 面板中的按鈕元件

10. 選擇原始的 Flash 文檔。
11. 在 *Library* 面板中，只要點擊空白區域並選擇 *Paste* 以將 *button* 元件拷貝到文檔庫。
12. 選擇 *scene* 層的第一個關鍵幀。
13. 從庫面板中點擊並拖動按鈕元件到文檔場景中來，在 *Main Menu* 標籤下面。

14. 選擇場景中的新實例，改變實例名-將 *Properties* 改為‘optionsBtn’。該實例名稱可以被定義的按鈕行為索引。

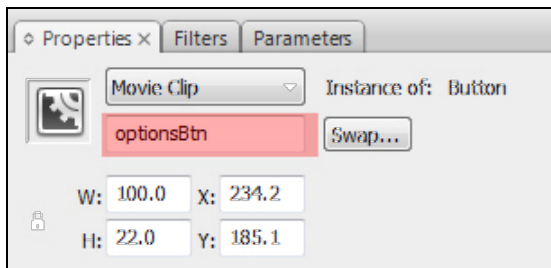


圖 14: 命名 option 按鈕實例為‘optionsBtn’

15. 點擊 *Parameter* 標籤，並在 *label* 區域輸入 ‘Options’。此域中包含了運行時按鈕上所要顯示的內容。

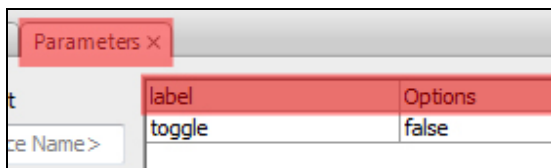


圖 15: 標記按鈕‘Options’

16. 祝賀你！用 *Scaleform* 啟動面板測試第一個場景，按下 *Scaleform Launcher* panel 中的 *Test with[Executable Name]* 按鈕在 *Scaleform* 播放器中運行動畫。

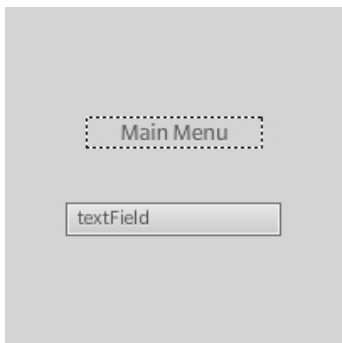


圖 16: *Scaleform* 播放器中 option 介面按鈕的主功能表

3.3 監聽 *Options Screen* 按鈕

需要實現交互功能受限需要在主功能表額選擇按鈕中添加功能，使用戶點擊按鈕可以跳轉到 *Options Screen*。本例中 **Scaleform CLIK** 事件模型用來實現此項功能。相對於其他可以使用的方法，在使用 **Scaleform CLIK** 元件時 **Scaleform CLIK** 事件模型具有更強的易用性和高效性。在此模型中，出發一個特殊事件將使 *Options* 按鈕調用一個回調函數。調用函數使時間軸前進到 *Options Screen* 關鍵幀，觸發事件為一個滑鼠點擊動作，當用戶按下滑鼠左鍵時即被觸發。

1. 選擇 *actions* 圖層的第一個關鍵幀。
2. 將下面代碼插入到“`stop();`”命令之前。代碼功能為為用戶提供選項功能表，位於第 10 幀：

```
function OpenOptionsScreen()  
{  
    gotoAndPlay( "optionsScreen" );  
}
```

3. 現在插入使 *optionsBtn* 實例在“點擊”事件觸發後調用上面所描述函數—*openOptionsScreen*—的代碼

```
optionsBtn.addEventListener( "click", this, "OpenOptionsScreen" );
```

4. 最後，用最後一行代碼設置選項功能表的初始化焦點。

```
optionsBtn.focused = true;
```

5. 測試動畫。*Options* 按鈕默認情況下應該為選中狀態。點擊 *Options* 按鈕應使用戶跳轉到 *Options screen*，當前該介面為空。

3.4 增加可選步驟難度

利用 *Options Stepper* 增加一個遊戲複雜控制元件到 *Options Menu* 當中。該元件為一個動態資料驅動元素用來顯示一個擁有多個選項的文本域，選項顯示為：Easy、Medium、Hard 和 Insane。使用兩個按鈕元件可以遍曆整個選項列表。

1. 返回到 CLIK_Components.fla 文件。
2. 只要點擊 *Library* 面板中的 *OptionsStepper* 元件並選擇 *Copy*。
3. 返回到原始 Flash 文檔。
4. 在 *scene* 圖層（*Options Menu* 關鍵幀）選擇第 10 幀上的關鍵幀。
5. 粘貼 *OptionsStepper* 到 *Library* 面板。

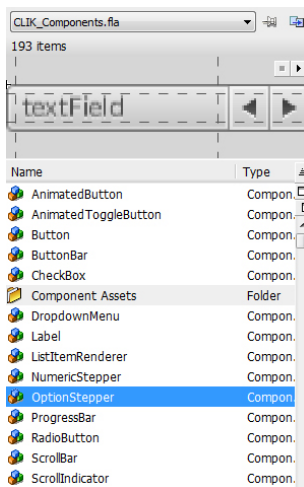


圖 17: library 面板中的 **OptionStepper** 元件

6. 如果 *OptionsStepper* 已經被粘貼到場景中，則跳過該步驟。否則，從 *Library* 面板點擊和拖動 *OptionsStepper* 元件到場景。
7. 選擇 *OptionsStepper*，改變實例名為 'difficultyOption'。

8. 從庫中點擊並拖動一個新的 *Label* 元件，定位到 *Difficulty* 控制元件的左邊。這表示難度設置控制文本描述

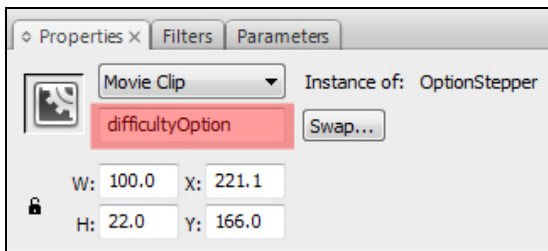


圖 18: 命名複雜控制實例為‘difficultyOption’

9. 在 *Label* 標籤為‘Difficulty’的元件改變 *text* 域。

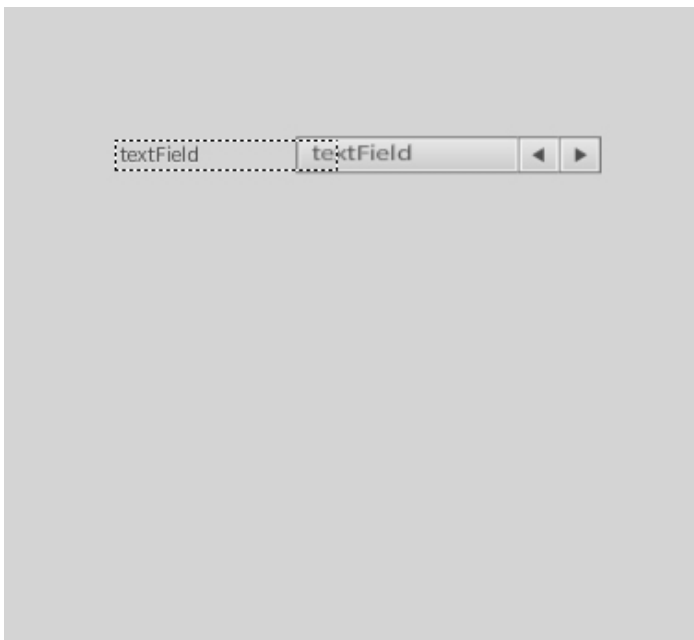


圖 19: 最終無皮膚 difficulty option stepper

3.5 增加視頻設置核取方塊和選擇框

視頻設置控制可以利用核取方塊將兩個選項的其中之一 **on** 或者 **off**：泛光和反混淆類似於一組兩個選擇按鈕來改變反混淆如何在遊戲中工作。核取方塊的選中和取消用來使能或禁止某項功能；選擇按鈕在一組選項中只有其中一個是有效的。

1. 在 *Difficulty* 控制下創建一個新的 *Label* 元件實例，設置文本參數為‘Video Settings’。
2. 選擇 CLIK_Components.fla 文件。
3. 只要點擊庫裏的 *CheckBox* 元件並選擇 *Copy*。
4. 返回到工作 *Flash* 文件粘貼核取方塊到 *Library* 面板。
5. 從 *Library* 面板點擊和拖動核取方塊元件到場景。

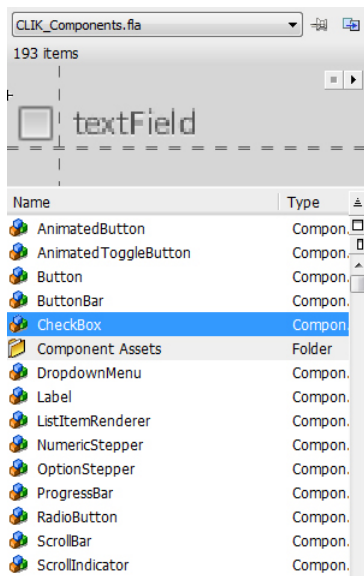


圖 20: library 面板中的核取方塊元件

6. 選擇新的實例，改變實例名為‘bloomBtn’。

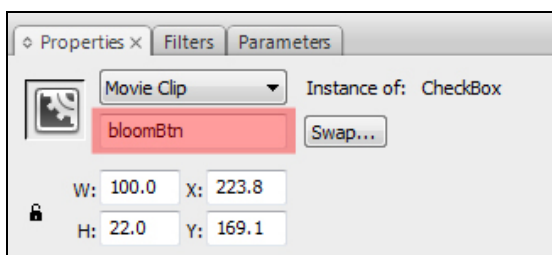


圖 21: bloom 多選按鈕為實例命名為‘bloomBtn’

7. 改變 *label* 域參數為‘Bloom’。
8. 複製實例，放置 *copy* 到 *bloom* 核取方塊，改變新實例的名字為‘aaBtn’。
9. 改變 *label* 域參數為‘Anti-Aliasing’。
10. 打開文件 *RadioButton.fla*。
11. 複製和粘貼選擇框元件到工作庫。
12. 拖動一個新的選擇按鈕實例到反混淆核取方塊的場景中。
13. 改變選擇按鈕的實例名為‘edgeaaBtn’。

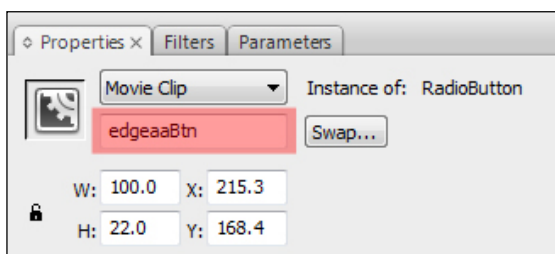
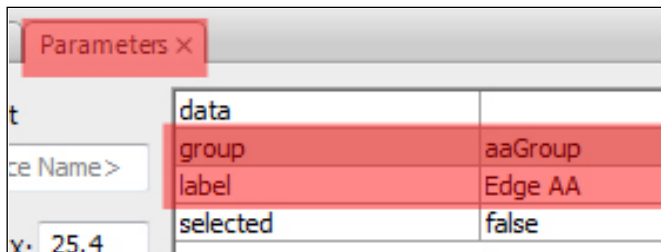


圖 22: Edge AA 選擇按鈕實例名為‘edgeaaBtn’.

14. 改變 *label* 區域的參數名為‘Edge AA’。
15. 改變分組區域為‘aaGroup’。

16. 複製實例，放置複製直接到 Edge AA 選擇按鈕，改變實例名字為‘hwaaBtn’。
17. 改變 *label* 域參數名為‘Hardware FSAA’。
18. 改變組區域為‘aaGroup’。



Parameters x	
data	
group	aaGroup
label	Edge AA
selected	false

圖 23: 設置選擇按鈕組為‘aaGroup’。

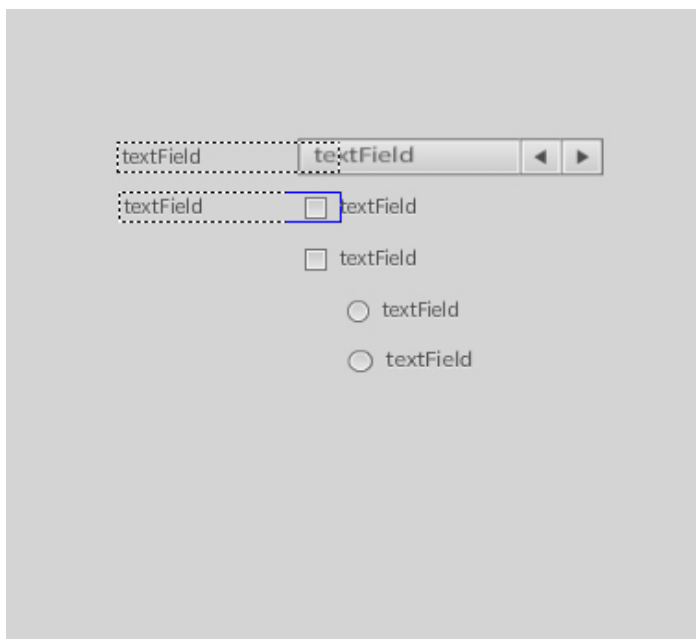


圖 24: 增加的無皮膚背景 Video Settings 設置介面

3.6 增加音量調節條

音量調節可以添加一個滑動條元件來。滑動條利用了跟蹤和移動控制。這兩者都是簡單的按鈕元件。移動條決定了滑動的長度，因此用戶不能移動出左右邊界。同樣垂直方向上也如此。

1. 複製或者創建一個新的 **Label** 元件實例在場景中的視頻設置控制裏，設置其文本參數為‘Sound Volume’。
2. 選擇 CLIK_Components.fla 文件並拷貝和粘貼滑動條元件到工作 FLA 文檔。

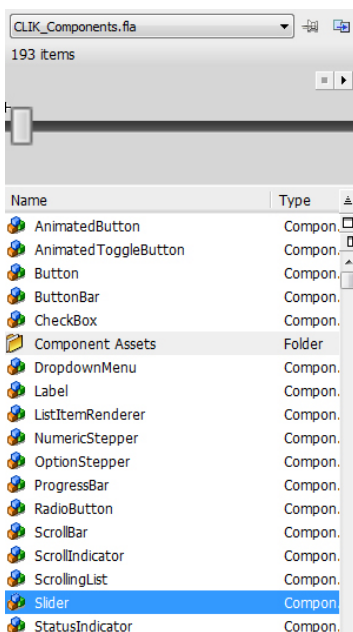


圖 25: library 面板中的滑動條元件

3. 拖動一個滑動條實例到音量調節標籤邊上的場景並改變實例名為‘soundSlider’。

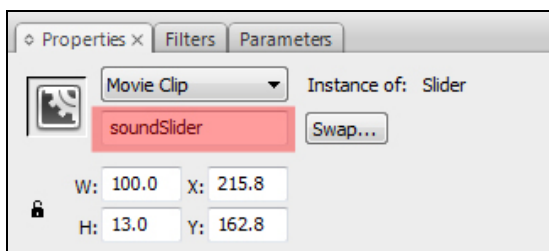


圖 26: 命名音量調節實例為‘soundSlider’。

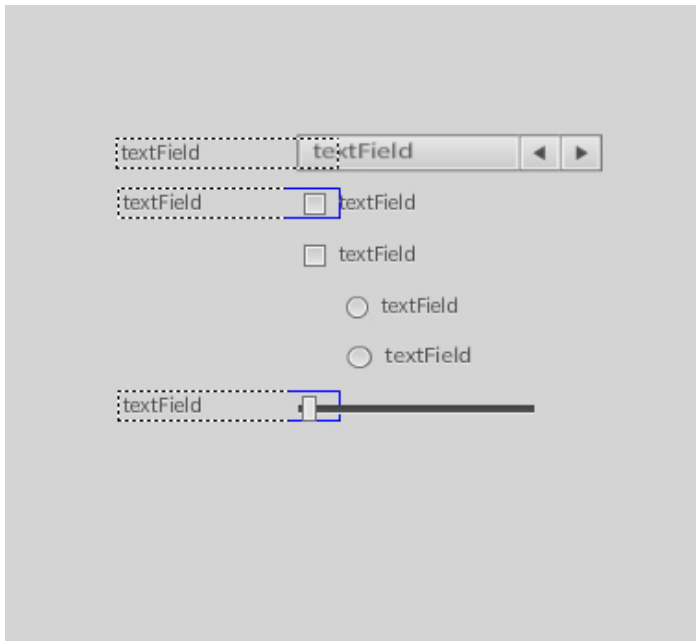


圖 27: 添加的最終的無皮膚背景音量調節滑動條

3.7 增加OK 和 Cancel 按鈕

OK 和 Cancel 按鈕使用基本的按鈕元件。

1. 從庫拖動一個按鈕元件到場景。
2. 選擇按鈕。
3. 點擊按鈕的參數標籤並這是標籤為‘OK’。
4. 改變實例名為‘okBtn’。
5. 拷貝按鈕到場景並改變新按鈕的實例名為‘cancelBtn’。
6. 點擊新按鈕拷貝的參數標籤，並設置標籤域為‘Cancel’。

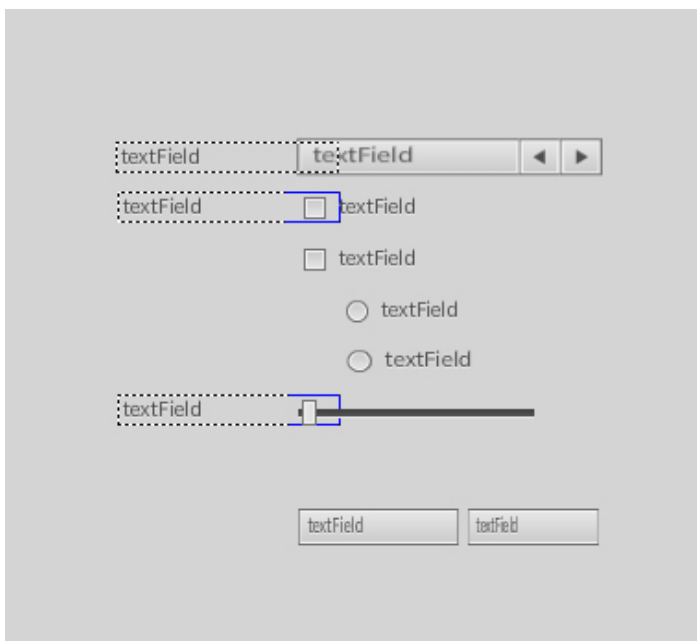


圖 28: 無皮膚背景 Options 介面

4. 用 **ActionScript** 增加功能

在這點上，所有的單獨默認皮膚背景最元件已被創建，但是功能表無實際功能。需要增加 **ActionScript** 腳本。如果 **Actions** 面板尚為打開，按下快捷鍵(F9)調出 **Actions** 面板。

如果指導手冊中的代碼不起作用，打開指導手冊文件瀏覽代碼查看哪里輸入錯誤。

4.1 增加難度動作選項的難度等級

爲了使難度動作能夠顯示難度等級並允許改變設置，參數資料必須與之相匹配。這可以使用難度選項元件的‘**dataProvider**’屬性來實現，傳遞預期的資料。這種情況下，傳遞的資料爲一個難度等級列表：**Easy**、**Medium**、**Hard** 和 **Insane**。

1. 選擇 **actions** 圖層的第 10 幀上的關鍵幀。
2. 在 **Actions** 面板內用滑鼠點擊打開代碼輸入框。
3. 在"**stop();**"命令行上插入如下代碼：

```
difficultyOption. dataProvider = [ "Easy", "Medium", "Hard", "Insane" ];
```

4. 測試動畫，現在可以用到遊戲難度設置選項控制功能 - 顯示四種難度等級點擊向前或向後按鈕進行選擇。

4.2 設計 aaGroup 按鈕群

抗锯齿的按钮需要在源代码中结合在一起。设置视频设置、防混淆复合框，关键反混淆选择按钮应该被禁止（无需改动）除非用户选中选择按钮的选项。

1. 选择 **actions** 图层的第 10 帧上的关键帧。
2. 先，在 **Actions** 面板上面加这行源代码为使用 CLIK 的 **ButtonGroup** 类：

```
import gfx.controls.ButtonGroup;
```

3. 下一步，在 `difficultyOption.dataProvider` 那行源代码，创建一个新的 **ButtonGroup** 叫‘aaGroup’。分配到 `edgeaaBtn` 和 `hwaaBtn`：

```
var aaGroup:ButtonGroup = new ButtonGroup();  
edgeaaBtn.group = hwaaBtn.group = aaGroup;
```

4. 下一步,創建一個函數實現使能/禁止防按鈕:

```
function EnableAARadioButtons()  
{  
    edgeaaBtn.disabled = hwaaBtn.disabled = !aaBtn.selected;  
}
```

5. 最后，添加一个事件侦听器到防混淆复合框调用以下函数：

```
aaBtn.addEventListener("click", this, "EnableAARadioButtons");
```

6. 测试动画，默认情况下，视频设置的反锯齿选择按钮当前应该被禁止，但只需点击检验框就能使其有效

4.3 設置選項的初始狀態

当用户按 OK 键返回主菜单时，这些选项应当保持其状态，如果按取消键则应忽略任何更改。

1. 在時間線的頂部創建一個新層(位於行動層之上),並將其重命名為“vars”。
2. 選擇變數層中的幀 1.
3. 新建一個名為“選項”的物件,它能保持各種選項的狀態。將這個新物件放到一個條件陳述式中,以測試該物件是否已經存在:

```
if (!options) {  
    var options:Object = {};  
}
```

4. 選擇行動層中的幀 10
5. 設置 difficultyOption 的初始狀態。如果還未對 options.selectedDifficulty 的屬性進行定義,則將 difficultyOption 的初始值設為指數 0(與“容易”相對應),否則,對 options.selectedDifficulty 進行如下設置:

```
difficultyOption.selectedIndex = (options.selectedDifficulty) ?  
    options.selectedDifficulty : 0;
```

7. 設置 bloomBtn 與 aaBtn 的初始狀態:

```
bloomBtn.selected = options.bloom;  
aaBtn.selected = options.aa;
```

8. 根據 aaBtn 核取方塊所選擇的狀態,將 edgeaaBtn 和 hwaaBtn 設置為“可用”或“不可用”。

```
edgeaaBtn.disabled = hwaaBtn.disabled = !aaBtn.selected;
```

9. 設置 edgeaaBtn 的初始狀態。如果 options.edgeaa 屬性未被定義,則將 edgeaaBtn 的所選狀態設置為“真”,否則對 edgeaaBtn 進行如下定義:

```
edgeaaBtn.selected = (options.edgeaa) ? options.edgeaa : true;
```

10. 對 hwaaBtn 設置同樣的初始狀態,但是當其未被定義時,需將其設置為“假”。

```
hwaaBtn.selected = (options.hwaa) ? options.hwaa : false;
```

11. 設置 soundSlider 的初始狀態。

```
soundSlider.value = options.soundVolume;
```

4.4 退出選擇介面

代码的最后一个位能够使用户退出选择界面，当按下 *Cancel* 或者 *OK* 按钮时返回到主菜单。

1. 选择 *actions* 图层的第 10 帧的关键帧。
2. 插入能够点选 *OK* 按钮时引导用户返回到帧 1（主菜单）的函数到第 4.3 节中的代码之后，并且需要在 `stop();` 命令之前。这个函数设计（和存）*options* 对象的特性，所以用户回来 *Options* 屏幕状态还是一样。

```
function ReturnToMainMenu()  
{  
    options.selectedDifficulty = difficultyOption.selectedIndex;  
    options.bloom = bloomBtn.selected;  
    options.aa = aaBtn.selected;  
    options.edgeaa = edgaaaBtn.selected;  
    options.hwaa = hwaaBtn.selected;  
    options.soundVolume = soundSlider.value;  
    gotoAndPlay("mainMenu");  
}
```

3. 插入点选 *Cancel* 按钮时引导用户返回到帧 1（主菜单）。这个函数不存任何组件状态的改变。

```
function CancelToMainMenu()  
{  
    gotoAndPlay("mainMenu");  
}
```

4. 下一步监听用户点击 *OK* 或者 *Cancel* 按钮动作：

```
okBtn.addEventListener("click", this, "ReturnToMainMenu");  
cancelBtn.addEventListener("click", this, "CancelToMainMenu");
```

5. 最後,給 *Cancel* 按钮預設獲得焦點。

```
cancelBtn.focused = true;
```

6. 测试动画，所以的 *Options Screen* 控制当前应已运行。

5. 功能表介面

最後一步為將用戶自定義、美觀的介面應用到功能表和每個 UI 元件。與其描述如何裝飾每個部件，避免複製，文本只描述如何描繪一個應用元件實例。一旦該過程已掌握，描繪剩餘的元件也可以按照相同的工作流程。

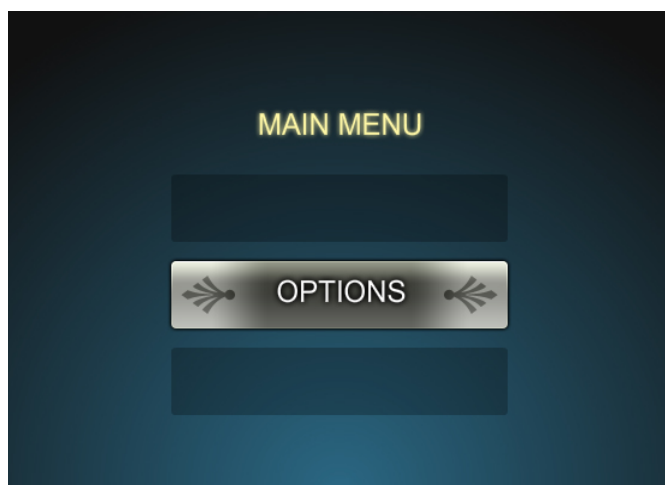


圖 29: 最終帶皮膚背景的功能表畫面

5.1 繪製選擇按鈕背景

這裏描述的工作流程針對 Photoshop®、Illustrator® 和 Flash CS3，且使用任何一個工具都能實現目的效果。

UI 介面中的每個物件都可以有多個狀態。在按鈕實例中，狀態為 **Up**（默認情況下）、**Over**（選中或滑鼠位於上方）和 **Down**（當點擊）。每種不同的狀態應該用不同的形式來表示以向用戶產生視覺化反饋資訊。這些狀態可以在 Photoshop 或 Illustrator 中進行編輯。實際應用中，一個元件的每種不同的狀態圖示最好包含在單個 Photoshop 或 Illustrator 文件中。每個不同狀態在美術資源檔案中應該為一個獨立的圖層。這樣使 Flash 在時間軸上的關鍵幀上可以合併圖層。在導入美術資源前，將每個狀態的相關圖層合併到一個圖層。最合適的做法為，保持工作文件和導入資源檔案的獨立，工作文件中包含了原始的圖層資訊，這些資訊在以後的編輯和調整中需要用到，導入文件中那個所有圖層應該經過合併以導入到 Flash。在一個簡單按鈕的例子中，最終導入資源檔案應該只有 3 個圖層- **Up**、**Over** 和 **Down**。

1. 構造同個 Photoshop 文件（PSD）中的 *Options* 按鈕的每一個狀態圖示。
 - a. 查看每個部件的文件，可以發現元件需要一系列的狀態圖示，或索引時間軸上未經渲染的元件所在的 *Labels* 圖層。
 - b. **重要提示：**避免創建擁有相同的狀態特徵的圖層副本。這種情況下，所需要的狀態為 **up**、**over** 和 **down**。
2. 一旦完成，保存 PSD 文件。如果用多個圖層創建一個狀態，則合併所有相同狀態下的圖層為同一個圖層。合併過程首先按下(CTRL)鍵，然後點擊每個需要合併的圖層到 Photoshop 的 *Layers* 面板，然後只需點擊所選擇圖層並從彈出功能表中選擇 *Merge Layers*。
 - a. 例如，合併還處於狀態 **over** 的每個圖層為狀態為 **over** 的圖層。
3. 確保圖層預先排列好，Flash 元件時間軸上的第一個狀態 - 通常為 **up** - 位於 Photoshop 的 *Layers* 面板底部，而時間軸上最後一個狀態為最底部圖層。這個次序在 Flash 導入每個圖層中必須嚴格遵循。

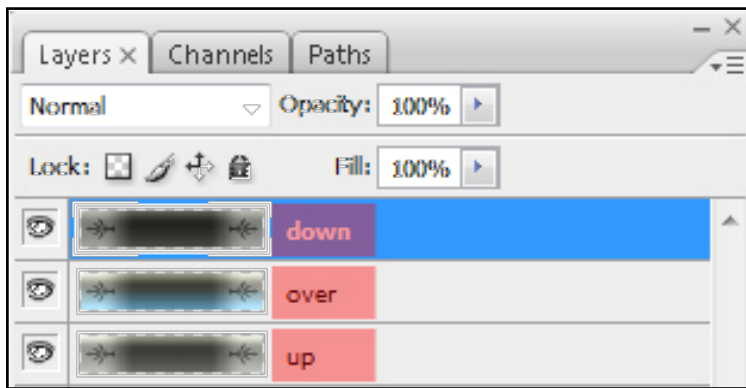


圖 30: 選項按鈕皮膚的 Photoshop 圖層資訊

4. (Optional)重命名每個層則可以匹配 Flash 中的狀態名稱：*up*、*over*、*down* 等。這樣取名完全是便於組織。
5. 保存文件為一個新的 PSD 文件，將 ‘mergedLayers’ 追加到檔案名末尾。不要覆蓋原始的步驟 2 中保存的 PSD 文件，除非合併前圖層資訊不再需要。
6. 在 Flash 中，打開無背景圖的 *Options* 按鈕，在場景中雙擊該按鈕可以看到其時間軸，
7. 選擇時間軸上的 *button* 圖層並刪除所有的幀。只點擊圖層名稱選擇所有的圖層，時間軸上點擊幀，從彈出功能表中選擇 *Remove Frames* 。
8. 添加一個新的空白關鍵幀到圖層並選中。該步驟在步驟 9 之前是必須的。
9. 點擊頂部功能表文件，選擇‘Import’，然後從彈出功能表中選擇‘Import to Stage’或者按鍵 (CTRL+R)。
 - a. 如果‘Import to Stage’無法獲得，請確認導入的圖層沒有被鎖。如果在圖層名稱右邊有一個鎖住圖示，只要點擊圖示對圖層解鎖。
10. 瀏覽並選擇 PSD 或者具有背景渲染的 UI 元素的 Adobe Illustrator (AI)文件 - 本例中用了 PSD 文件。
11. 導入對話方塊打開。確保每個導入圖層已被選中。
 - a. **重要提示:** 導入對話方塊當在導入 Flash 8 中的 PSD 文件並不顯現，而在 CS3 和 CS4 中顯現。如果使用 Flash 8 並導入 PSD 文件，則需要更改工作流程。

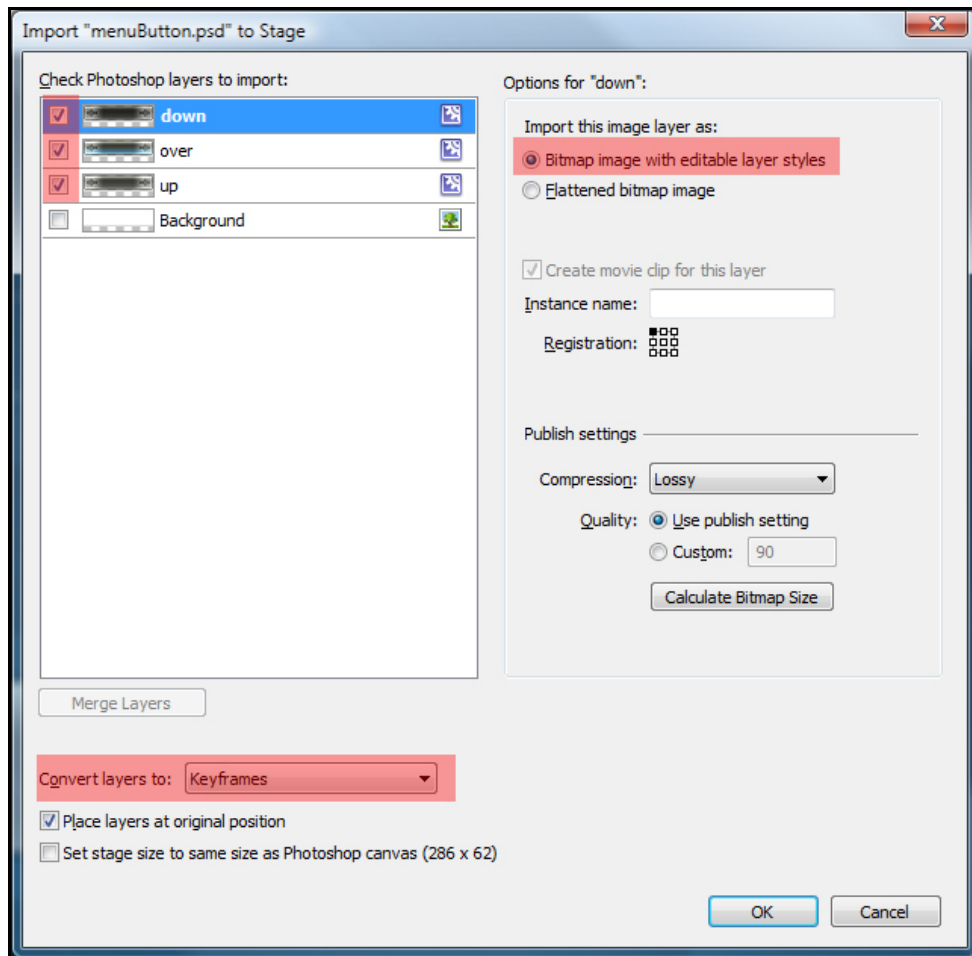


圖 31: 導入到場景對話方塊

12. 確保下拉 *Convert layers* 為：設置為“關鍵幀”。該下拉層應該在對話方塊左下方。
13. 點擊列表中每個圖層並選擇 *Import this image layer as* 標題下面的 *Bitmap image with editable layer styles* 。
 - a. **重要提示:** 如果該步驟未在每個層上執行，任何圖層類型將不能在場景中展現，-儘管添加到了庫中-每個圖層在第一個圖層之外將為一個平面圖像只包括所有的其他圖層。
14. 圖層將在時間軸上作為獨立的關鍵幀導入。按照最初狀態到最後一個狀態排列。
 - a. **重要提示：**當導入 PSD 文件，關鍵幀將添加到時間軸頂部的新圖層中，並刪除原來的 *button* 圖層，重命名新創建的 *button* 圖層，點擊並拖動新的 *button* 圖層到圖層列表底部。

- b. 當導入一個 AI 文件，關鍵幀應該添加到 *button* 圖層，只要在導入前選擇即可。
15. 點擊並拖動每個關鍵幀，排列成合適的狀態標籤；這些都將在時間軸的 *Labels* 圖層展現出來。移動第一個關鍵幀到第一個幀（*up*）。移動到第二個關鍵幀到 *over* 關鍵幀之下。移動第三和最後一個關鍵幀到 *down* 關鍵幀。本指南未對禁止功能進行描述。

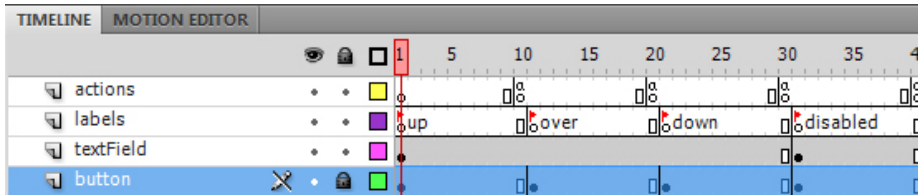


圖 32: 添加關鍵幀到 *button* 圖層

16. 編輯 *Text* 圖層的每個關鍵幀中 *textField* 實例的字體設置-*up*、*over*、*down*-若按鈕標籤在用戶按鈕介面表現為不同的形狀。否則，保持所有狀態中的 *textField* 原狀。
17. 測試動畫。*Options* 按鈕添加表面皮膚。

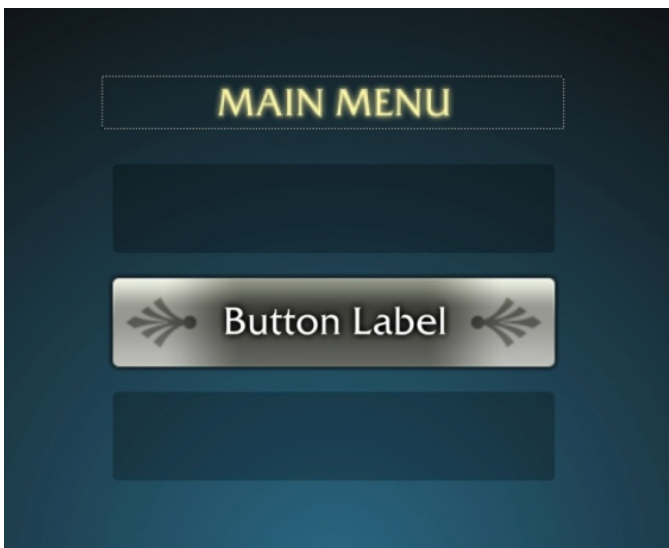


圖 33: 有皮膚背景和主功能表的選擇按鈕

5.2 繪製捲軸背景

捲軸由兩個必不可少的按鈕組成。一個按鈕為捲軸（從左滑動到右邊）另外一個按鈕為捲軸滑動軌迹。在這種情況下，創建每個按鈕-捲軸和軌迹-如同創建一個如 5.1 節所那個描述的常規按鈕。

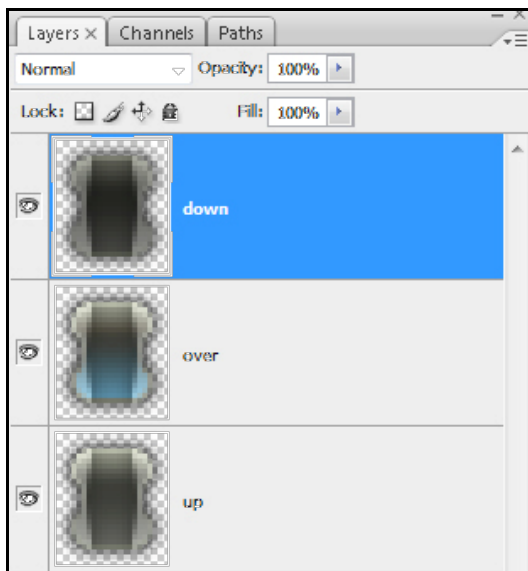


圖 34: 音量捲軸滾動狀態皮膚 Photoshop 圖層

1. 雙擊場景中的捲軸輸入時間軸。
2. 雙擊捲軸的軌迹元件。
3. 導入為軌迹創建的 PSD 或者 AI 文件，對齊關鍵幀。軌迹的註冊點應該在圖像的左邊，應該與滑動條的 registration 點相匹配。
4. 返回到滑動條時間軸（在上層）。
5. 雙擊滾動元件輸入時間軸。
6. 導入捲軸 PSD 或 AI 文件。
 - a. **重要提示:** 當導入捲軸皮膚，確認改變了每個層的 registration 位置在中間。

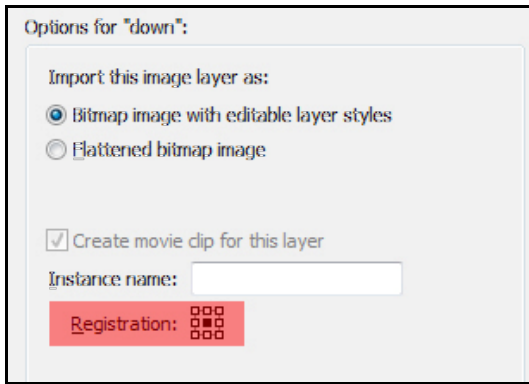


圖 35: 設置 registration 點到中間位置

7. 選擇每個關鍵幀上的捲軸圖像，設置 X 座標為‘0.0’。
8. 返回到運動條時間軸（向上一層）並選擇捲軸元件進行移動以保持左上角邊框與捲軸 registration 點匹配。

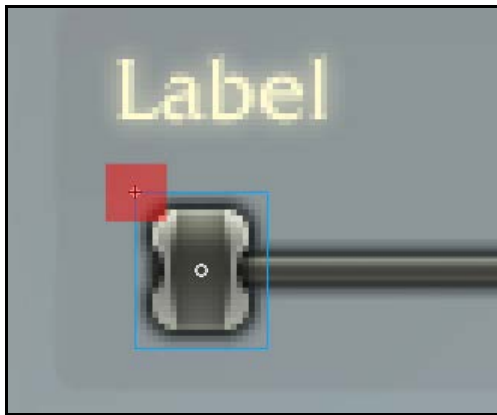


圖 36: 將滾動按鈕左上角與滾條基點對齊

9. 編輯 *textField* 標籤的字體屬性。
10. 測試動畫。



圖 37: 具有皮膚背景的音量調節條

5.3 繪製選擇按鈕和核取方塊背景

選擇按鈕和核取方塊在以下狀態下為必不可少的按鈕元件：*selected_up*、*selected_over*、*selected_down*、*selected_out* 和 *selected_disabled*。這裏，無需介紹如何繪製背景，只要遵照 5.1 節中的描述步驟即可；但是，必須包含新的狀態作為背景繪製過程的一部分。通常，選擇按鈕包括在按鈕內部用一個小點表示被選中，核取方塊則包括一個確認標記來表示同樣的含義。但是，設計不是都必須按照這個範例進行。

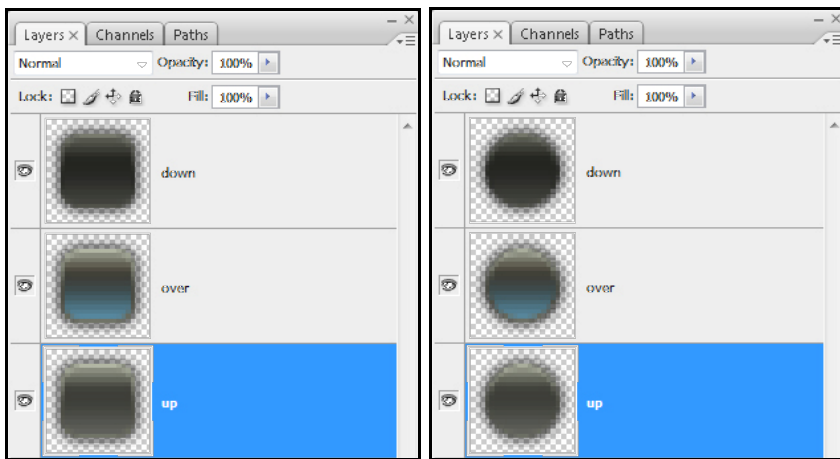


圖 38: Photoshop 圖層中的選擇按鈕和核取方塊介面

最後，注意編輯 *textField* 標籤實例中的字體屬性。



圖 39: 具有皮膚背景的 edge AA 選擇按鈕

5.4 描繪難度選擇框皮膚背景

難度選擇框，主要由按鈕部件組成。需要兩個按鈕：一個左按鈕，一個右按鈕。第三個圖形元素用來渲染元件背景/邊框，當不是必不可少的。

1. 用 5.1 節中提到的方法創建一個左和右方向按鈕。
 - a. 或者，可以使用相同的圖像來表示兩個按鈕，只需要在 **Flash** 中對圖像鏡像處理即可。
2. 在場景上，雙擊難度選擇框輸入時間軸。下一步，如果與設計相匹配，則替換底部背景/邊界圖像，下面可以看到部件周圍的灰度邊界。選擇左按鈕並在按鈕上雙擊輸入時間軸。現在簡單得導入一個新的擁有皮膚背景的 **PSD** 文件到狀態圖層，如 5.1 節中所述，對齊關鍵幀。
3. 接下來，如果適合設計，則替換底部北京/邊框圖形，見下圖中逐漸為灰色邊框。
4. 雙擊進入時間軸選擇 *left* 按鈕。
5. 現在只需導入新皮膚 **PSD** 文件到 5.1 小結中描述的 **States** 圖層並對其關鍵幀。
6. 對右按鈕重複步驟 4-5。
7. 返回到上級時間軸上的兩個按鈕 — **Difficulty option stepper** 時間軸 — 並編輯標籤名為 *textField* 的屬性字體
8. 測試動畫。



圖 40: 具有皮膚背景的難度選擇框

6. 結論



圖 41:具有皮膚背景的選擇功能表

祝賀你！本文介紹了一種使用 **Scaleform CLIK** 的方法。儘管還有很多種替換方案，在大多數工程中都有其獨具挑戰的方法。但是，熟悉文中所介紹的 **Scaleform CLIK** 的基本工作流程還只局限於基本的適當應用和 **Scaleform CLIK** 框架的正式使用。需要深入學習，推薦技術方面的文檔，實驗 **Scaleform CLIK** 自帶的各種演示文件，瀏覽每個元件的源代碼並閱讀其中的注釋。提供了完善的操作步驟來確保最終用戶成功應用 **Scaleform CLIK**。

演示檔和原始程式碼所在 **Scaleform 4.2 SDK** 目錄:

Windows 系統:

- 演示文件 – *C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/demos*
- 源代码 – *C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/gfx*

蘋果系統:

- 演示 – *scaleform_gfx_4.2_macos/Resources/AS2/CLIK/demos*
- 源代码 – *scaleform_gfx_4.2_macos/Resources/AS2/CLIK/gfx*

附加文檔：

- [Getting Started with CLIK 按鈕入門](#) – CLIK 中按鈕逐漸應用入門。