

Autodesk® Scaleform®

CLIK 入门

本文介绍了如何使用和有效运用 Scaleform®通用精简接口工具包（Common Lightweight Interface Kit, CLIK™）

作者: Matthew Doyle
版本: 3.0
最新修订: 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 联系方式:

文档	CLIK 入门
地址	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
网站	www.scaleform.com

邮箱	info@scaleform.com
电话	(301) 446-3200
传真	(301) 446-3199

目 录

1. 概要	1
1.1 最低软件配置	1
2. Scaleform CLIK 和 Scaleform 面板安装和配置.....	2
2.1 增加 Scaleform CLIK 类路径到 Flash CS3.....	2
2.2 添加 Scaleform CLIK 类路径到 Flash CS4.....	3
2.3 增加 Scaleform 启动面板.....	5
3. 用组件创建一个简单的完整界面菜单.....	6
3.1 创建菜单 Shell.....	8
3.2 添加主菜单导航控制功能.....	10
3.3 监听 Options Screen 按钮	13
3.4 增加可选步骤难度	14
3.5 增加视频设置复选框和选择框.....	16
3.6 增加音量调节条.....	19
3.7 增加 OK 和 Cancel 按钮	21
4. 用 ActionScript 增加功能.....	22
4.1 增加难度动作选项的难度等级.....	22
4.2 设计 aaGroup 按钮群	23
4.3 设置选项的初始状态.....	24
4.4 退出选择界面	25
5. 菜单界面.....	26
5.1 绘制选择按钮背景	27
5.2 绘制滚动条背景.....	31
5.3 绘制选择按钮和复选框背景	34
5.4 描绘难度选择框皮肤背景.....	35
6. 结论	36

1. 概要

Autodesk® Scaleform® CLIK™ 使开发者快速有效得原型化游戏的界面菜单，然后实现和应用这些原型。本文包括了利用 Scaleform CLIK 和 ActionScript™创建一个自定义游戏界面的基本步骤，以及几个最典型的组件和界面创建实例。文中还描述了创建过程的每一个步骤。

说明： 在 Flash 中完成相同的工作会有不同的方式。开发者可以找到并使用那些比本教程中所采用的方法更适合自己的替代工作流。

1.1 最低软件配置

- Scaleform 3.0 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 面板安装和配置

Scaleform CLIK 类似于 Adobe Flash 的设计面板以提高工作流程。使用 Scaleform CLIK 组件和工作流增强面板的第一步为确认 Scaleform CLIK 类的路径已正确添加到 Flash 应用环境中。然后使用 Adobe 扩展管理器来安装扩展面板。

安装 Creative Suite® 3 (CS3)和 CS4/CS5 略有不同。我们为 CS3 和 CS4/CS5 两者提供了安装指令。

2.1 增加 Scaleform CLIK 路径到 Flash CS3

Flash 类必须为 Scaleform CLIK 类设置路径，以对授权环境开放。

1. 启动 Flash CS3。
2. 点击 *Flash* 菜单顶部，在下拉菜单中选择参数。
3. 在目录下选择 *ActionScript*。

ActionScript 2.0

4. 在 *Language* 选项中点击 *ActionScript 2.0* 设置。
5. 在 *ActionScript 2.0* 设置对话框中，通过[+]按钮增加一个新路径。

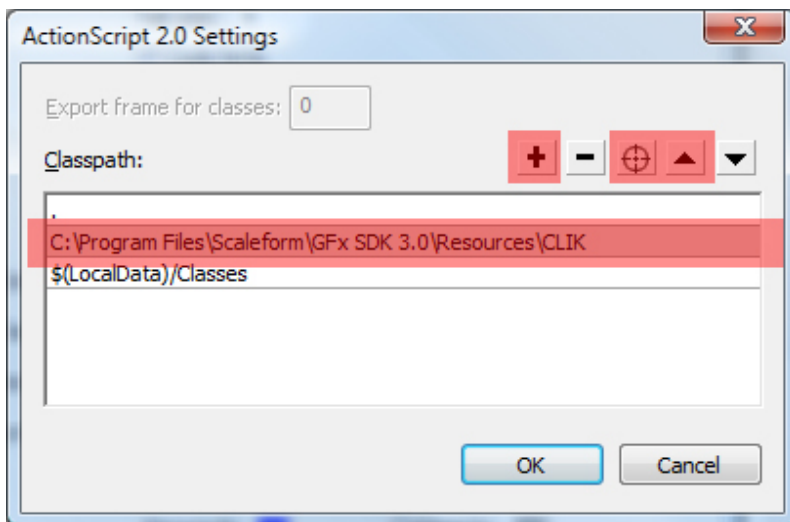


图 1: CS3 中的 *ActionScript 2.0* 类路径。

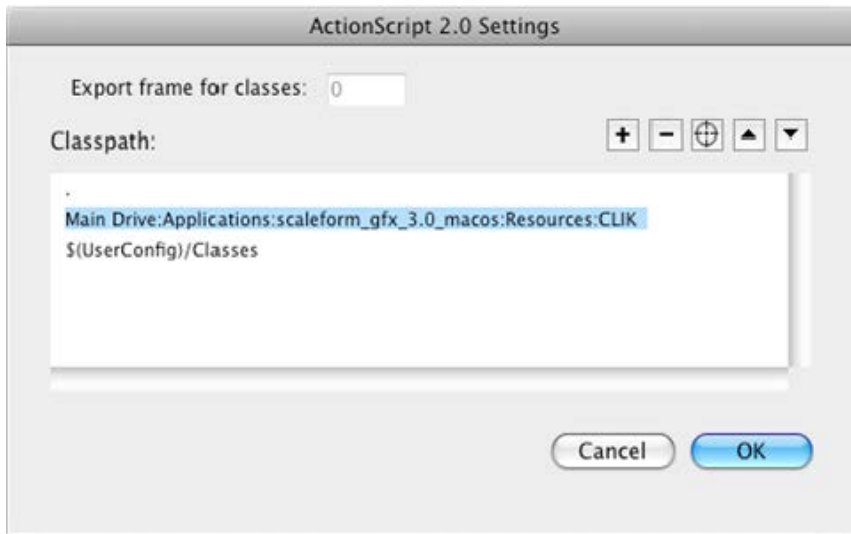


图 2: CS3 中的 **ActionScript 2.0** 类路径。

6. 点击十字交叉图标浏览 CS3 中路径，选择 Scaleform CLIK 安装目录。其 Windows 默认安装路径为: *C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/*，其苹果系统默认安装路径为 *scaleform_gfx_4.2_macos/Resources/AS2/CLIK* 在 Scaleform 4 SDK 目录。
7. 返回 *ActionScript 2.0* 设置对话框，在 *Classpath* 下选择增加路径。使用 *Arrow* 按钮移动到 *\$(LocalData)/Classes* 行之上。如果是评估系统的话，不用使用 *Arrow* 按钮，可以直接把 *\$(UserConfig)/Classes* 行拖过来。来确保 Scaleform CLIK 类的索引位于内嵌类之前。如果不进行这步操作，则内嵌 Adobe 组件将替代 Scaleform CLIK 组件。
8. 点击 OK 关闭 ActionScript 2.0 设置窗口，然后点击 OK 关闭 Preferences 窗口。

2.2 添加 Scaleform CLIK □ 路径到 Flash CS4, CS5

Flash 类必须为 Scaleform CLIK 类设置路径，以对授权环境开放。

1. 启动 Flash CS4 or CS5。
2. 点击 *Flash* 菜单顶部，在下拉菜单中选择参数。
3. 在目录下选择 *ActionScript*。
4. 在 *Language* 选项中点击 *ActionScript 2.0* 设置。

5. 在 *ActionScript 2.0* 设置对话框中，通过[+]按钮增加一个新路径。

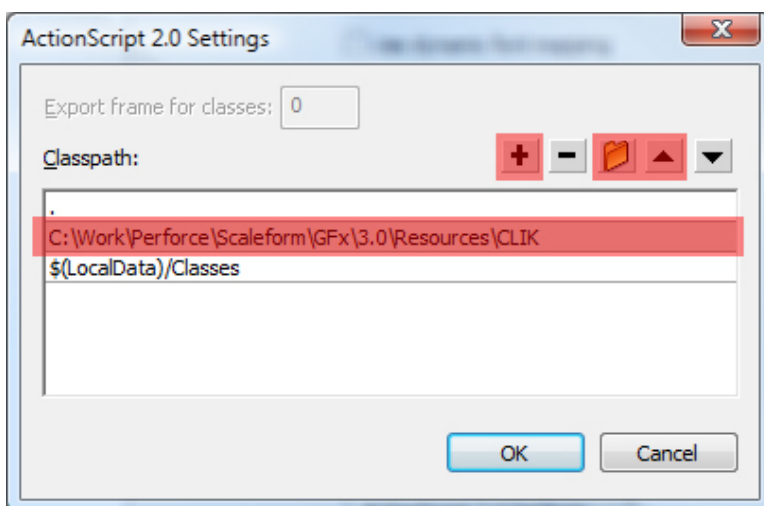


图 3: CS4 中 *ActionScript 2.0* 类路径窗口

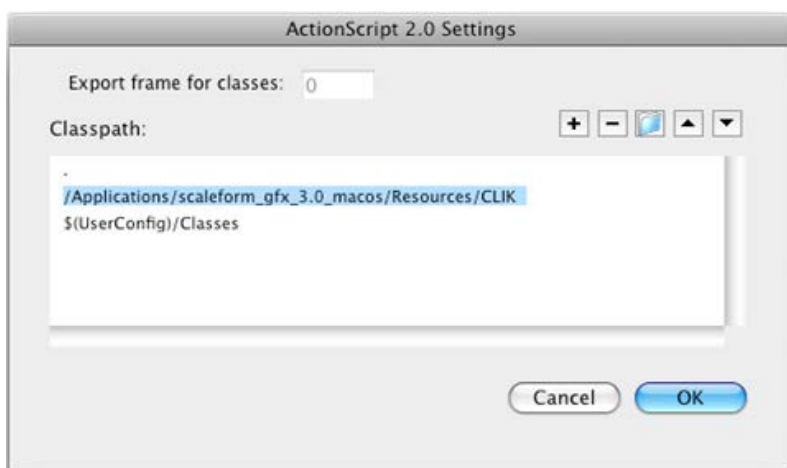


图 4: CS4 中 *ActionScript 2.0* 类路径窗口

6. 点击十字交叉图标浏览 CS4 中路径，选择 Scaleform CLIK 安装目录。其默认安装路径为：C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/，然后在苹果系统，组件在 scaleform_gfx_4.2_macos/Resources/AS2/CLIK/。
7. 返回 *ActionScript 2.0* 设置对话框，在 *Classpath* 下选择增加路径。使用 *Arrow* 按钮移动到 *\$(LocalData)/Classes* 行之上。如果是评估系统的话，不使用 *Arrow* 按钮，可以直接把 *\$(UserConfig)/Classes* 行拖过来。来确保 Scaleform CLIK 类的索引位于内嵌类之前。如果不进行这步操作，则内嵌 Adobe 组件将替代 Scaleform CLIK 组件。
8. 点击 *OK* 关闭 *ActionScript 2.0* 设置窗口。

9. 点击 *OK* 关闭 *Preferences* 窗口。

2.3 增加 Scaleform 启动面板

Scaleform 启动面板为从 Flash 创作环境直接向 Scaleform 播放器发布 SWF 文件提供了一个方便快捷的方法。如果您还未安装该面板，请参考 Scaleform 4 入门文件中第 2.4 到 2.6 节中的安装与使用说明。

3. 用组件创建一个简单的完整界面菜单

本节将详细介绍创建一个游戏菜单的必要步骤。菜单包括两个界面：主菜单和选择菜单。主菜单拥有一个按钮使用户转移到选择界面。菜单工作面板在选择界面当中，其中包括一系列复杂的设置选项，视觉效果的视频选项和使用内嵌复选框和选择框的防混淆按钮，以及一个音量控制器。

本指南的资源位于在 **Scaleform 4.2 SDK** 目录的 **CLIK** 安装目录下：

Resources/AS2/CLIK/tutorial

Resources/AS2/CLIK/tutorial/art_assets

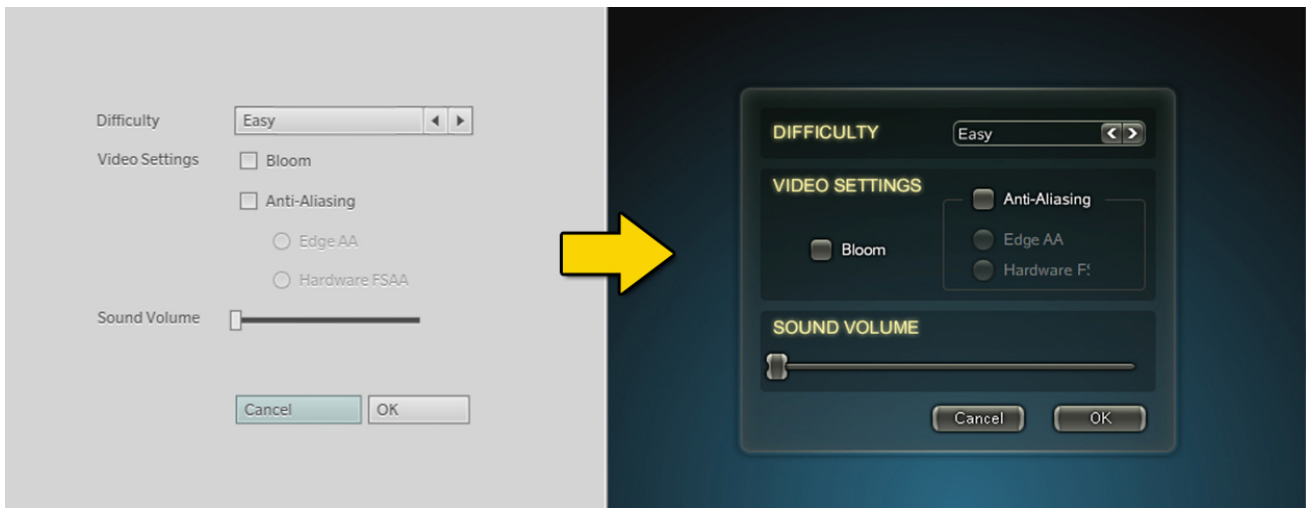


图 5: 无皮肤和有皮肤背景最终选择界面

本指南中的用户接口创建有如下过程：

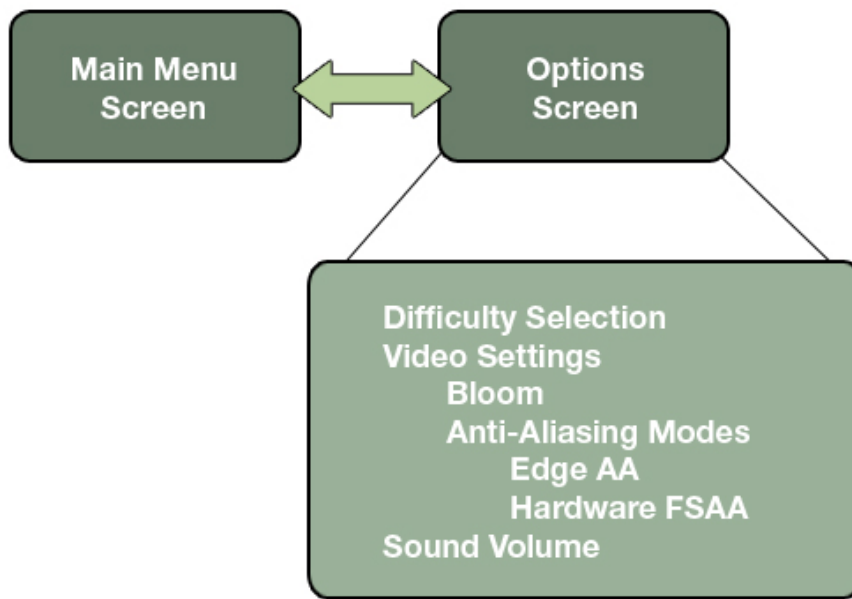


图 6: 完整界面菜单创建流程

3.1 创建菜单位置 Shell

第一步为创建为菜单创建多种类型的帧和关键帧。这里介绍了创建的大致步骤。

1. 创建一个新的 Flash（ActionScript 2.0）文档。
2. 确保文档的发布设置为 Player 8，从顶部菜单选择 *File*，在底部下拉框中选择‘Publish Settings’然后点击 *Flash* 标签。
3. 设置 Version 下来框为 ‘Flash Player 8’。
4. 设置 *ActionScript version* dropdown 为 ‘ActionScript 2.0’。
5. 点击 *OK* 来关闭 *Publish Settings* 窗口。

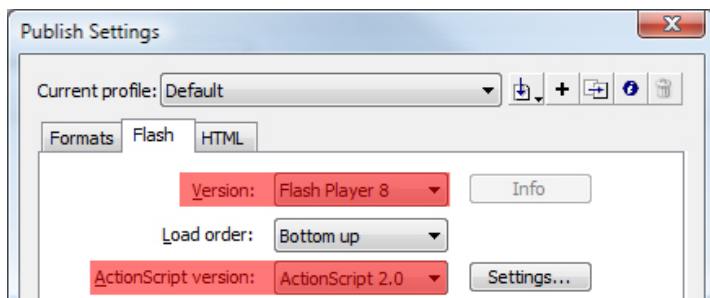


图 7：发布设置窗口

6. 在主文档时间轴上选择当前图层，当前标签为 *Layer 1*，双击 *Layer 1* 将其改名为 ‘scene’。如果名称并不重要，这种方法命名最为合适。
7. 点击 *Insert Layer* 按钮创建一个新的图层，命名此帧图层为‘actions’。同样图层的名称并不重要，只要容易记忆即可。

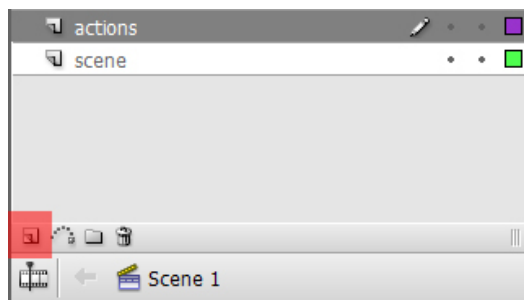


图 8：插入图层按钮

- 在帧 10 上为 *action* 和 *scene* 两个层添加第二个关键帧。选择 *scene* 层，然后单击帧 10 上的时间轴。从弹出菜单选择 ‘Insert Keyframe’。该关键帧将被用来创建菜单的 Option 界面部分。在 *action* 层上重复该步骤。

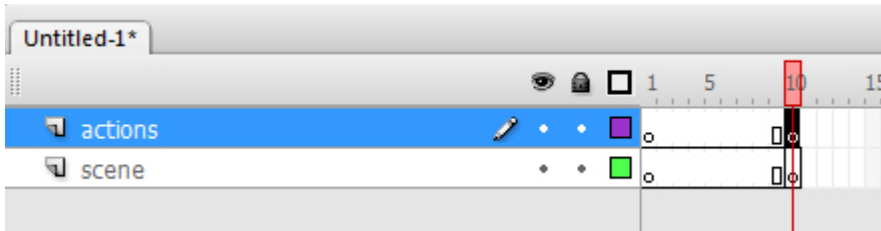


图 9: 在第 10 帧上添加关键帧的主菜单时间轴

- 在 *scene* 图层 ‘mainMenu’ 的帧 1 上标记为关键帧。可以通过选择关键帧并在 *Properties* 标签中输入名字来实现。该标签可以被代码用来告知 Scaleform “goto” mainMenu 帧上的时间轴。第 1 帧将被用来创建菜单的 Main 界面。

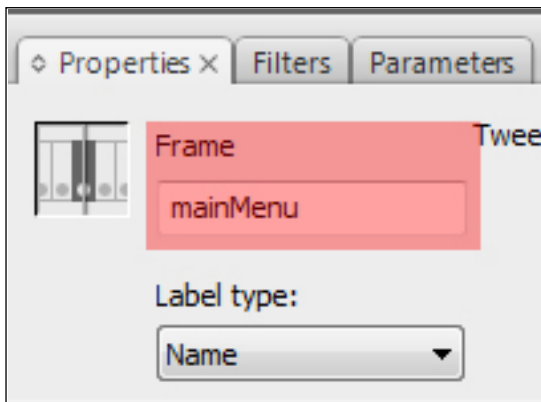


图 10: 关键帧 1 命名为 ‘mainMenu’

- 在 *scene* 层 ‘optionsScreen’ 的帧 10 上标记为关键帧。该标签可以被代码用来告知 Scaleform “goto” 时间轴上的 *optionsScreen* 帧。
- 在 *actions* 图层选择第 1 帧。打开 *Actions window* (F9)并在 *actions* 图层的帧 1 和帧 2 输入的第一行输入“stop();”（无双引号）。该代码告知播放器在此帧位置停止播放。如果该位置没有代码，播放器将继续播放直到出现 stop 命令位置，或者直到时间轴结束处，在本例中为第 10 帧。从 *actions* 图层选择第 10 帧，添加如上代码。
- 此时，正好可以保存 Flash 文件。

3.2 添加主菜单□□航控制功能

本节描述了如何在 *Main Menu* 增加一个 *Options* 按钮，用来使用户进入 *Options Screen*。

1. 打开位于 Scaleform 4 SDK 目录中的 *Resources/AS2/CLIK/components* 目录下的 *CLIK_Components.fla* 文件。
2. 首先添加菜单标题。进行该操作须只需点击 *CLIK_Components.fla* 文件的中 *Library* 面板中的 *Label* 组件并从弹出菜单选择 *Copy*。

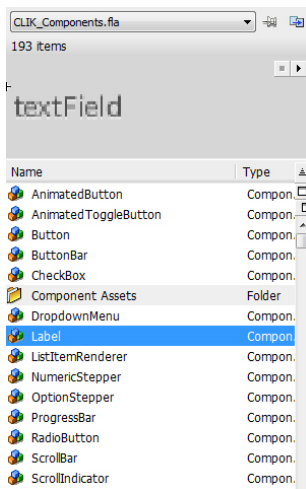


图 11:library 面板和 label 组件

3. 不要关闭 *CLIK_Components.fla* 文件，后面还需要用到。返回到在创建指导菜单的原始 Flash 文档。
4. 选择 *scene* 图层的第 1 帧。
5. 在 *Library* 面板中，只要点击空白区域并选择 *Paste* 以拷贝 *Label* 组件到文档库。
6. 点击并拖动 *Library* 面板中的 *Label* 组件至场景中，适合位于中间，临近顶部。
7. 为新创建的标签打开参数符，在 *text* 域中输入 ‘Main Menu’。这个区域中的内容在播放器运行时将被显示出来。**注意：**在 CS3 中，符号位于 *Properties* 和 *Filters* 标签旁边。在 CS4 中位于 *Component Inspector*，默认情况下为关闭。按[Shift + F7]组合键则显示，或者从弹出菜单选择 Window – Component Inspector。

注意：这些参数只有 SWF 文件被发布并播放时才会被显示出来。

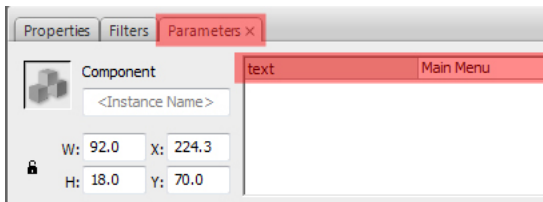


Figure 12: CS3 中参数标签位于 properties 标签旁边

8. 选择第 10 帧并重复步骤 6 和 7 创建一个 Options 界面的标题 - 在当前文本域输入 ‘Options’ 。
9. 返回 *CLIK_Components.fla* 文件，只要点击文件 *Library* 面板中的 *Button* 组件并再次选择 *Copy* 。

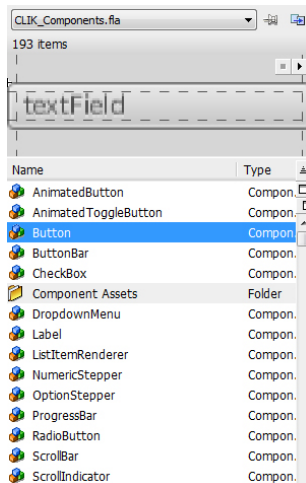


图 13 : library 面板中的按钮组件

10. 选择原始的 Flash 文档。
11. 在 *Library* 面板中，只要点击空白区域并选择 *Paste* 以将 *button* 组件拷贝到文档库。
12. 选择 *scene* 层的第一个关键帧。
13. 从库面板中点击并拖动按钮组件到文档场景中来，在 *Main Menu* 标签下面。
14. 选择场景中的新实例，改变实例名-将 *Properties* 改为‘optionsBtn’。该实例名称可以被定义的按钮行为索引。

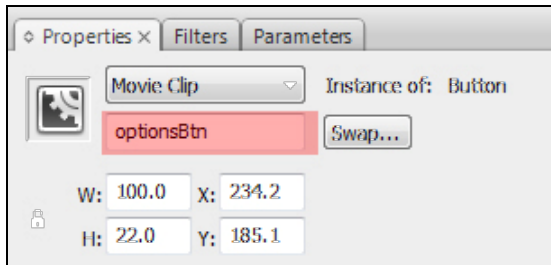


图 14: 命名 option 按钮实例为‘optionsBtn’

15. 点击 *Parameter* 标签，并在 *label* 区域输入 ‘Options’。此域中包含了运行时按钮上所要显示的内容。

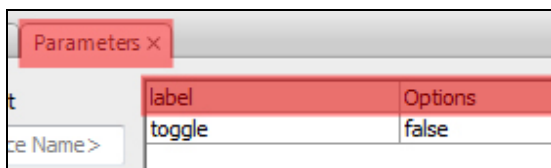


图 15: 标记按钮‘Options’

16. 祝贺你！用 *Scaleform* 启动面板测试第一个场景，按下 *Scaleform Launcher* panel 中的 *Test with[Executable Name]* 按钮在 *Scaleform* 播放器中运行动画。



图 16: *Scaleform* 播放器中 option 界面按钮的主菜单

3.3 监听 *Options Screen* 按钮

需要实现交互功能受限需要在主菜单额选择按钮中添加功能，使用户点击按钮可以跳转到 *Options Screen*。本例中 *Scaleform CLIK* 事件模型用来实现此项功能。相对于其他可以使用的方法，在使用 *Scaleform CLIK* 组件时 *Scaleform CLIK* 事件模型具有更强的易用性和高效性。在此模型中，出发一个特殊事件将使 *Options* 按钮调用一个回调函数。调用函数使时间轴前进到 *Options Screen* 关键帧，触发事件为一个鼠标点击动作，当用户按下鼠标左键时即被触发。

1. 选择 *actions* 图层的第一个关键帧。
2. 将下面代码插入到 `stop();` 命令之前。代码功能为为用户提供选项菜单，位于第 10 帧：

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

3. 现在插入使 *optionsBtn* 实例在“点击”事件触发后调用上面所描述函数—*openOptionsScreen*—的代码

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

4. 最后，用最后一行代码设置选项菜单的初始化焦点。

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

5. 测试动画。*Options* 按钮默认情况下应该为选中状态。点击 *Options* 按钮应使用户跳转到 *Options screen*，当前该界面为空。

3.4 增加可选步骤难度

利用 *Options Stepper* 增加一个游戏复杂控制组件到 *Options Menu* 当中。该组件为一个动态数据驱动元素用来显示一个拥有多个选项的文本域，选项显示为：**Easy**、**Medium**、**Hard** 和 **Insane**。使用两个按钮组件可以遍历整个选项列表。

1. 返回到 CLIK_Components.fla 文件。
2. 只要点击 *Library* 面板中的 *OptionsStepper* 组件并选择 *Copy*。
3. 返回到原始 Flash 文档。
4. 在 *scene* 图层（*Options Menu* 关键帧）选择第 10 帧上的关键帧。
5. 粘贴 *OptionsStepper* 到 *Library* 面板。

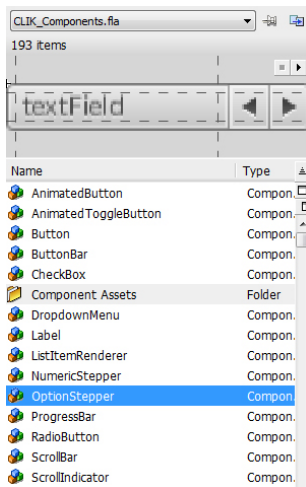


图 17: library 面板中的 OptionStepper 组件

6. 如果 *OptionsStepper* 已经被粘贴到场景中，则跳过该步骤。否则，从 *Library* 面板点击和拖动 *OptionsStepper* 组件到场景。
7. 选择 *OptionsStepper*，改变实例名为 'difficultyOption'。
8. 从库中点击并拖动一个新的 *Label* 组件，定位到 *Difficulty* 控制组件的左边。这表示难度设置控制文本描述

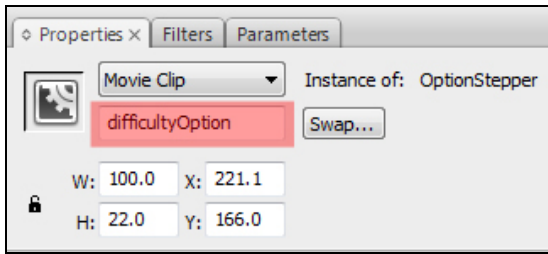


图 18: 命名复杂控制实例为‘difficultyOption’

9. 在 *Label* 标签为‘Difficulty’的组件改变 *text* 域。

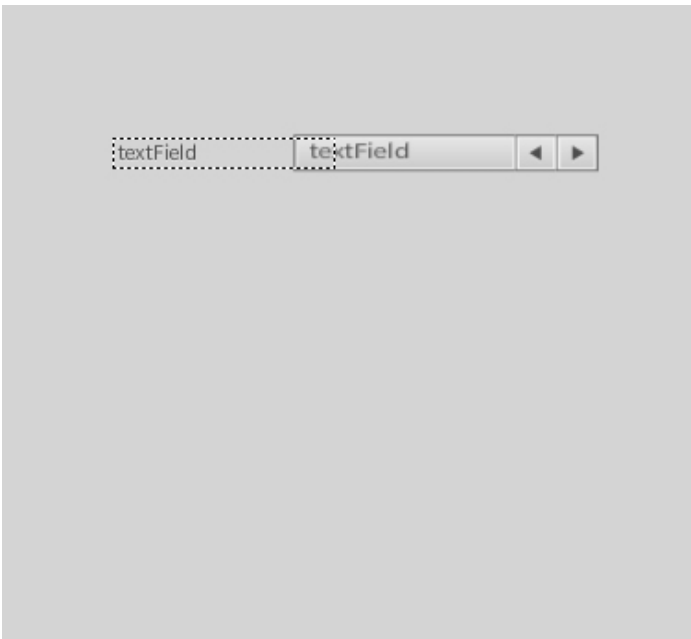


图 19: 最终无皮肤 difficulty option stepper

3.5 增加视频设置复选框和选择框

视频设置控制可以利用复选框将两个选项的其中之一 **on** 或者 **off**：泛光和反混淆类似于一组两个选择按钮来改变反混淆如何在游戏中工作。复选框的选中和取消用来使能或禁止某项功能；选择按钮在一组选项中只有一个是有效的。

1. 在 *Difficulty* 控制下创建一个新的 *Label* 组件实例，设置文本参数为‘Video Settings’。
2. 选择 CLIK_Components.fla 文件。
3. 只要点击库里的 *CheckBox* 组件并选择 *Copy*。
4. 返回到工作 *Flash* 文件粘贴复选框到 *Library* 面板。
5. 从 *Library* 面板点击和拖动复选框组件到场景。

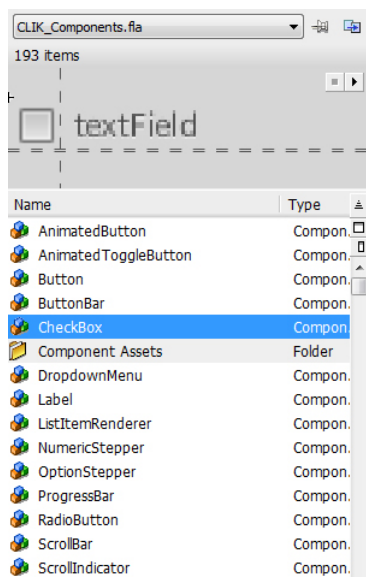


图 20: library 面板中的复选框组件

6. 选择新的实例，改变实例名为‘bloomBtn’。

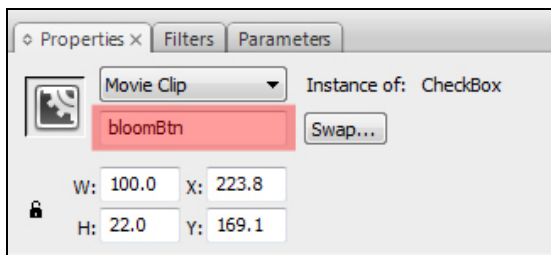


图 21: bloom 多选按钮为实例命名为‘bloomBtn’

7. 改变 *label* 域参数为‘Bloom’。
8. 复制实例，放置 *copy* 到 *bloom* 复选框，改变新实例的名字为‘aaBtn’。
9. 改变 *label* 域参数为‘Anti-Aliasing’。
10. 打开文件 *RadioButton.fla*。
11. 复制和粘贴选择框组件到工作库。
12. 拖动一个新的选择按钮实例到反混淆复选框的场景中。
13. 改变选择按钮的实例名为‘edgeaaBtn’。

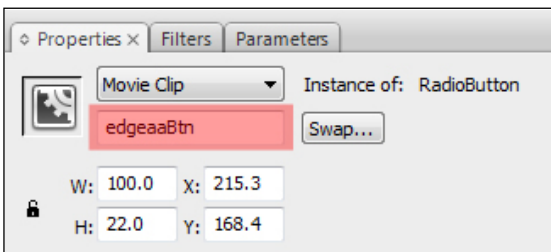
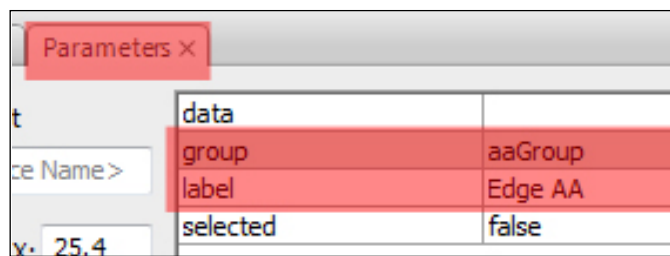


图 22: Edge AA 选择按钮实例名为‘edgeaaBtn’。

14. 改变 *label* 区域的参数名为‘Edge AA’。
15. 改变分组区域为‘aaGroup’。
16. 复制实例，放置复制直接到 Edge AA 选择按钮，改变实例名字为‘hwaaBtn’。

- 17. 改变 *label* 域参数名为‘Hardware FSAA’。
- 18. 改变组区域为‘aaGroup’。



Parameters x	
data	
group	aaGroup
label	Edge AA
selected	false

图 23: 设置选择按钮组为‘aaGroup’。

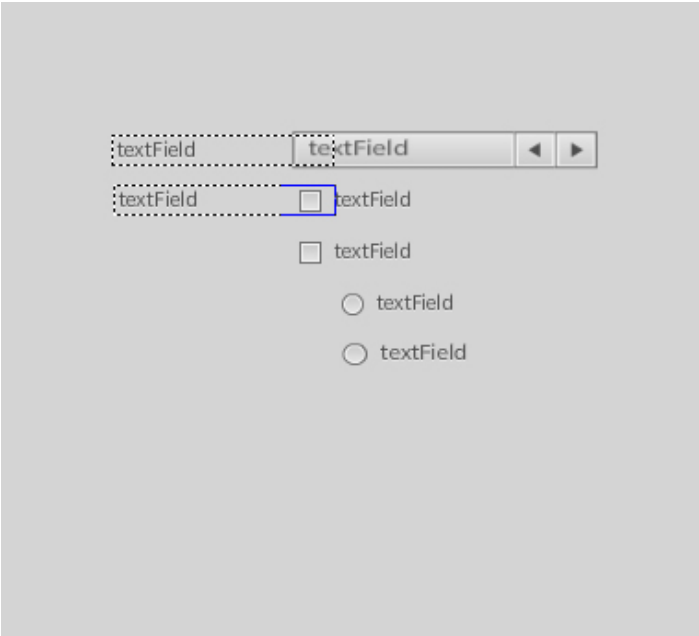


图 24: 增加的无皮肤背景 Video Settings 设置界面

3.6 增加音量□□条

音量调节可以添加一个滑动条组件来。滑动条利用了跟踪和移动控制。这两者都是简单的按钮组件。移动条决定了滑动的长度，因此用户不能移动出左右边界。同样垂直方向上也如此。

1. 复制或者创建一个新的 *Label* 组件实例在场景中的视频设置控制里，设置其文本参数为 ‘Sound Volume’。
2. 选择 CLIK_Components.fla 文件并拷贝和粘贴滑动条组件到工作 FLA 文档。

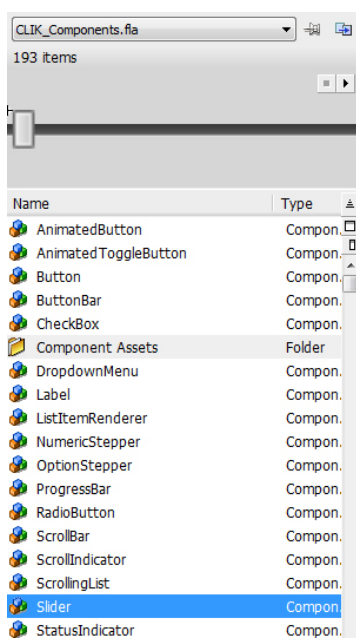


图 25: library 面板中的滑动条组件

3. 拖动一个滑动条实例到音量调节标签边上的场景并改变实例名为 ‘soundSlider’。

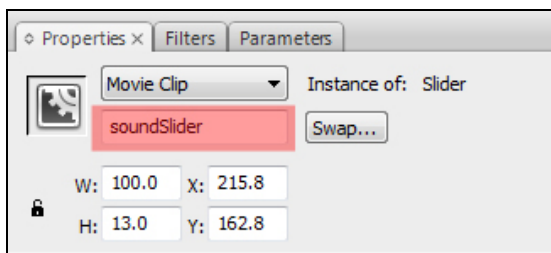


图 26: 命名音量调节实例为‘soundSlider’。

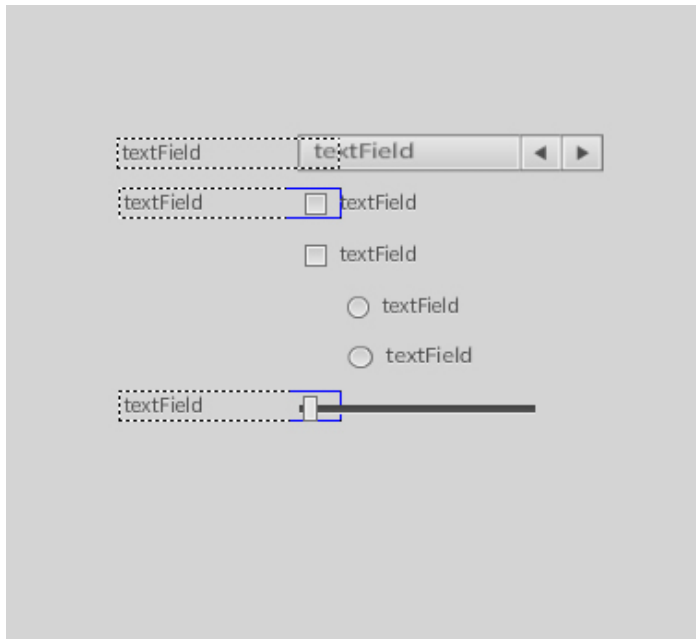


图 27: 添加的最终的非皮肤背景音量调节滑动条

3.7 增加OK 和Cancel 按钮

OK 和 Cancel 按钮使用基本的按钮组件。

1. 从库拖动一个按钮组件到场景。
2. 选择按钮。
3. 点击按钮的参数标签并这是标签为‘OK’。
4. 改变实例名为‘okBtn’。
5. 拷贝按钮到场景并改变新按钮的实例名为‘cancelBtn’。
6. 点击新按钮拷贝的参数标号，并设置标签域为 ‘Cancel’。

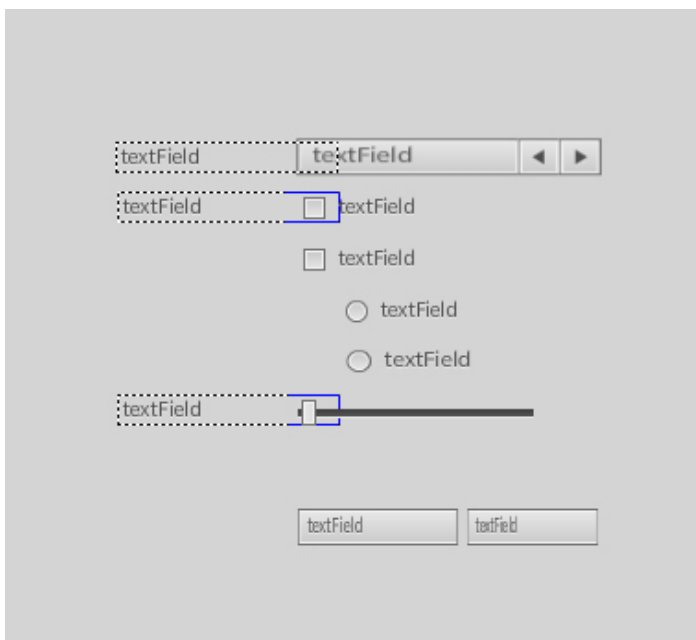


图 28: 无皮肤背景 Options 界面

4. 用 **ActionScript** 增加功能

在这点上，所有的单独默认皮肤背景最组件已被创建，但是菜单无实际功能。需要增加 **ActionScript** 脚本。如果 **Actions** 面板尚未打开，按下快捷键(F9)调出 **Actions** 面板。

如果指导手册中的代码不起作用，打开指导手册文件浏览代码查看哪里输入错误。

4.1 增加难度动作的难度等级

为了使难度动作能够显示难度等级并允许改变设置，参数数据必须与之相匹配。这可以使用难度选项组件的‘dataProvider’属性来实现，传递预期的数据。这种情况下，传递的数据为一个难度等级列表：
Easy、Medium、Hard 和 Insane。

1. 选择 **actions** 图层的第 10 帧上的关键帧。
2. 在 **Actions** 面板内用鼠标点击打开代码输入框。
3. 在"stop();"命令行上插入如下代码：

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

4. 测试动画，现在可以用到游戏难度设置选项控制功能 - 显示四种难度等级点击向前或向后按钮进行选择。

4.2 设计 aaGroup 按钮群

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，创建一个函数实现使能/禁止防按钮：

```
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 设置选项的初始状态

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. 在时间线的顶部创建一个新层（位于行动层之上），并将其重命名为“vars”。
2. 选择变量层中的帧 1。
3. 新建一个名为“选项”的对象，它能保持各种选项的状态。将这个新对象放到一个条件语句中，以测试该对象是否已经存在：

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

4. 选择行动层中的帧 10
5. 设置 difficultyOption 的初始状态。如果还未对 options.selectedDifficulty 的属性进行定义，则将 difficultyOption 的初始值设为指数 0（与“容易”相对应），否则，对 options.selectedDifficulty 进行如下设置：

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

6. 设置 bloomBtn 与 aaBtn 的初始状态：

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

7. 根据 aaBtn 复选框所选择的状态，将 edgeaaBtn 和 hwaaBtn 设置为“可用”或“不可用”。

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

8. 设置 edgeaaBtn 的初始状态。如果 options.edgeaa 属性未被定义，则将 edgeaaBtn 的所选状态设置为“真”，否则对 edgeaaBtn 进行如下定义：

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

9. 对 hwaaBtn 设置同样的初始状态，但是当其未被定义时，需将其设置为“假”。

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

10. 设置 soundSlider 的初始状态。

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

4.4 退出选择界面

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* 屏幕状态还是一样的。

```
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. 最后，给 *Cancel* 按钮默认获得焦点。

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

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

5. 菜单界面

最后一步为将用户自定义、美观的界面应用到菜单和每个 UI 组件。与其描述如何装饰每个部件，避免复制，文本只描述如何描绘一个应用组件实例。一旦该过程已掌握，描绘剩余的组件也可以按照相同的工作流程。

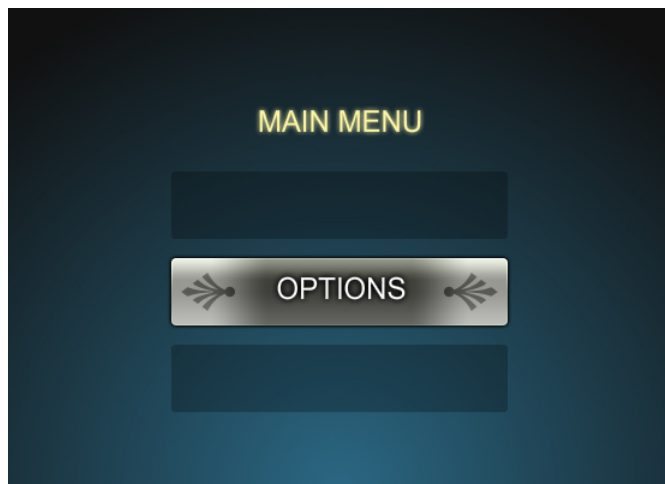


图 29: 最终带皮肤背景的菜单画面

5.1 绘制选择按钮背景

这里描述的工作流程针对 Photoshop®、Illustrator® 和 Flash CS3，且使用任何一个工具都能实现目的效果。

UI 界面中的每个对象都可以有多个状态。在按钮实例中，状态为 **Up**（默认情况下）、**Over**（选中或鼠标位于上方）和 **Down**（当点击）。每种不同的状态应该用不同的形式来表示以向用户产生可视化反馈信息。这些状态可以在 **Photoshop** 或 **Illustrator** 中进行编辑。实际应用中，一个组件的每种不同的状态图标最好包含在单个 **Photoshop** 或 **Illustrator** 文件中。每个不同状态在美术资源文件中应该为一个独立的图层。这样使 **Flash** 在时间轴上的关键帧上可以合并图层。在导入美术资源前，将每个状态的相关图层合并到一个图层。最合适的做法为，保持工作文件和导入资源文件的独立，工作文件中包含了原始的图层信息，这些信息在以后的编辑和调整中需要用到，导入文件中那个所有图层应该经过合并以导入到 **Flash**。在一个简单按钮的例子中，最终导入资源文件应该只有 3 个图层- **Up**、**Over** 和 **Down**。

1. 构造同个 **Photoshop** 文件（**PSD**）中的 *Options* 按钮的每一个状态图标。
 - a. 查看每个部件的文件，可以发现组件需要一系列的状态图标，或索引时间轴上未经渲染的组件所在的 *Labels* 图层。
 - b. **重要提示：**避免创建拥有相同的状态特征的图层副本。这种情况下，所需要的状态为 **up**、**over** 和 **down**。
2. 一旦完成，保存 **PSD** 文件。如果用多个图层创建一个状态，则合并所有相同状态下的图层为同一个图层。合并过程首先按下(**CTRL**)键，然后点击每个需要合并的图层到 **Photoshop** 的 *Layers* 面板，然后只需点击所选择图层并从弹出菜单中选择 *Merge Layers*。
 - a. 例如，合并还处于状态 **over** 的每个图层为状态为 **over** 的图层。
3. 确保图层预先排列好，**Flash** 组件时间轴上的第一个状态 - 通常为 **up** - 位于 **Photoshop** 的 *Layers* 面板底部，而时间轴上最后一个状态为最底部图层。这个次序在 **Flash** 导入每个图层中必须严格遵循。

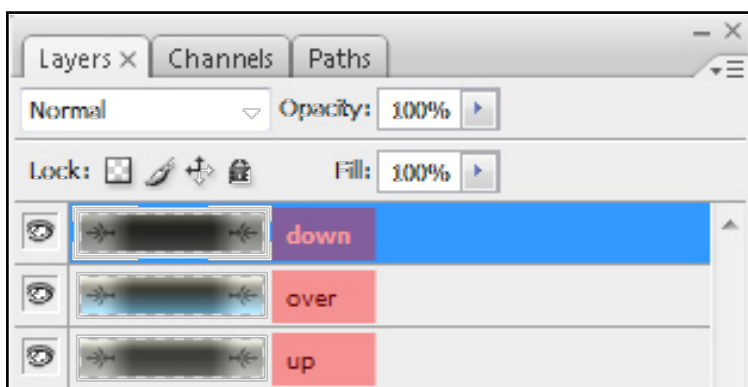


图 30: 选项按钮皮肤的 Photoshop 图层信息

4. (Optional)重命名每个层则可以匹配 Flash 中的状态名称: *up*、*over*、*down* 等。这样取名完全是便于组织。
5. 保存文件为一个新的 PSD 文件, 将 ‘mergedLayers’ 追加到文件名末尾。不要覆盖原始的步骤 2 中保存的 PSD 文件, 除非合并前图层信息不再需要。
6. 在 Flash 中, 打开无背景图的 *Options* 按钮, 在场景中双击该按钮可以看到其时间轴,
7. 选择时间轴上的 *button* 图层并删除所有的帧。只点击图层名称选择所有的图层, 时间轴上点击帧, 从弹出菜单中选择 *Remove Frames* 。
8. 添加一个新的空白关键帧到图层并选中。该步骤在步骤 9 之前是必须的。
9. 点击顶部菜单文件, 选择‘Import’, 然后从弹出菜单中选择‘Import to Stage’或者按键 (CTRL+R)。
 - a. 如果‘Import to Stage’无法获得, 请确认导入的图层没有被锁。如果在图层名称右边有一个锁住图标, 只要点击图标对图层解锁。
10. 浏览并选择 PSD 或者具有背景渲染的 UI 元素的 Adobe Illustrator (AI)文件 - 本例中用到了 PSD 文件。
11. 导入对话框打开。确保每个导入图层已被选中。
 - a. **重要提示:** 导入对话框当在导入 Flash 8 中的 PSD 文件并不显现, 而在 CS3 和 CS4 中显现。如果使用 Flash 8 并导入 PSD 文件, 则需要更改工作流程。

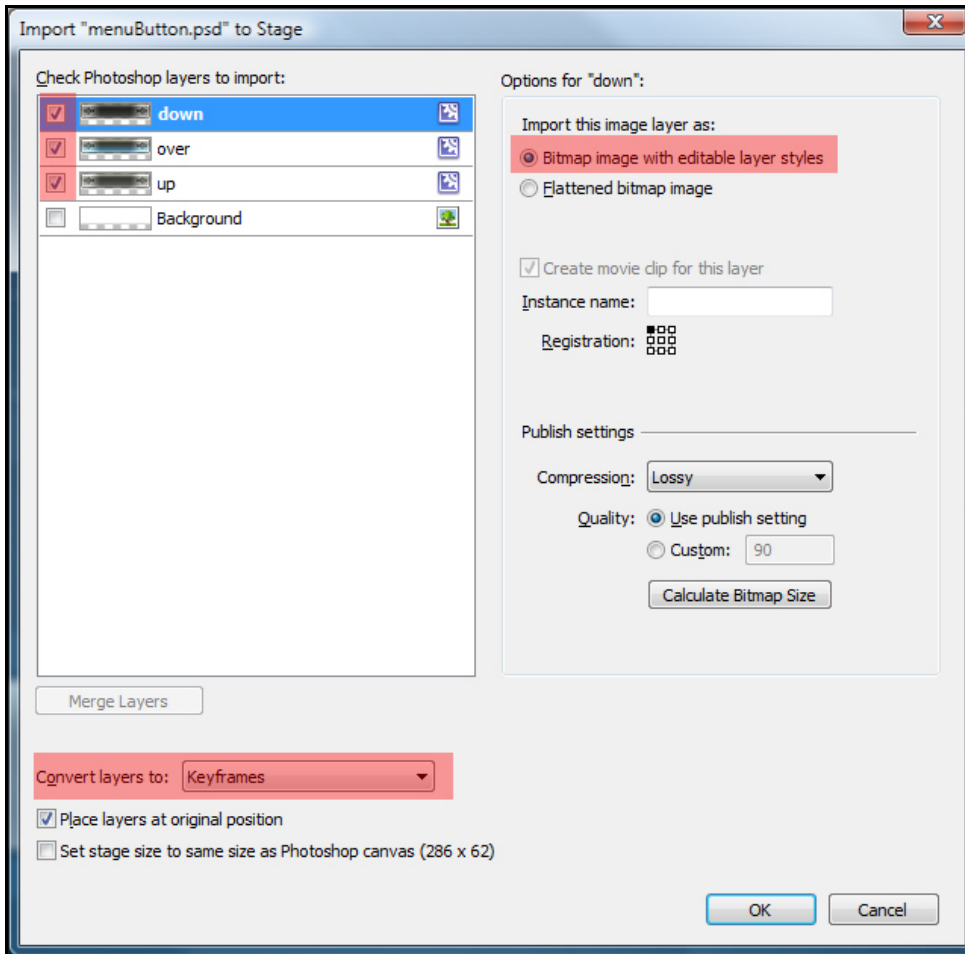


图 31: 导入到场景对话框

12. 确保下拉 *Convert layers* 为: 设置为“关键帧”。该下拉层应该在对话框左下方。
13. 点击列表中每个图层并选择 *Import this image layer as* 标题下面的 *Bitmap image with editable layer styles* 。
 - a. **重要提示:** 如果该步骤未在每个层上执行, 任何图层类型将不能在场景中展现, -尽管添加到了库中-每个图层在第一个图层之外将为一个平面图像只包括所有的其他图层。
14. 图层将在时间轴上作为独立的关键帧导入。按照最初状态到最后一个状态排列。
 - a. **重要提示:** 当导入 PSD 文件, 关键帧将添加到时间轴顶部的新图层中, 并删除原来的 *button* 图层, 重命名新创建的 *button* 图层, 点击并拖动新的 *button* 图层到图层列表底部。
 - b. 当导入一个 AI 文件, 关键帧应该添加到 *button* 图层, 只要在导入前选择即可。

15. 点击并拖动每个关键帧，排列成合适的状态标签；这些都将在时间轴的 *Labels* 图层展现出来。移动第一个关键帧到第一个帧（*up*）。移动到第二个关键帧到 *over* 关键帧之下。移动第三和最后一个关键帧到 *down* 关键帧。本指南未对禁止功能进行描述。

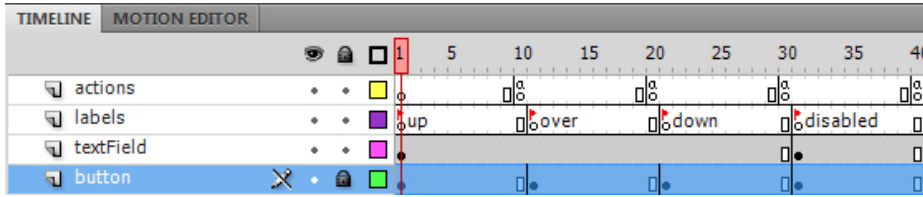


图 32: 添加关键帧到 **button** 图层

16. 编辑 *Text* 图层的每个关键帧中 *textField* 实例的字体设置-*up*、*over*、*down*-若按钮标签在用户按钮接口表现为不同的形状。否则，保持所有状态中的 *textField* 原状。
17. 测试动画。*Options* 按钮添加表面皮肤。

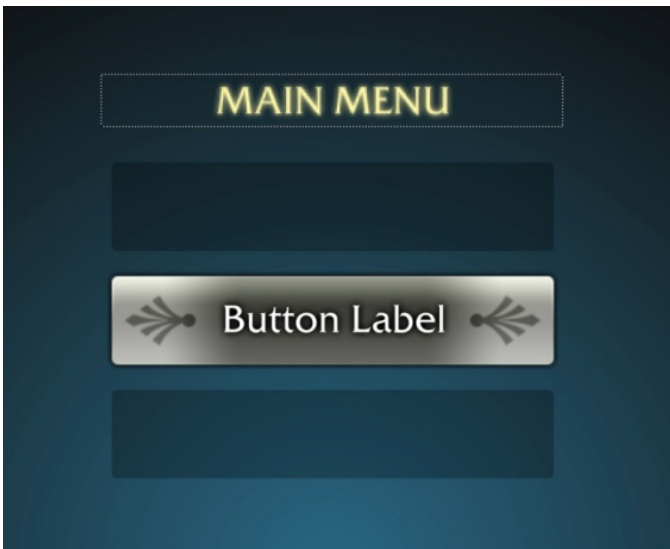


图 33: 有皮肤背景和主菜单的选择按钮

5.2 绘制滚动条背景

滚动条由两个必不可少的按钮组成。一个按钮为滚动条（从左滑动到右边）另外一个按钮为滚动条滑动轨迹。在这种情况下，创建每个按钮-滚动条和轨迹-如同创建一个如 5.1 节所那个描述的常规按钮。

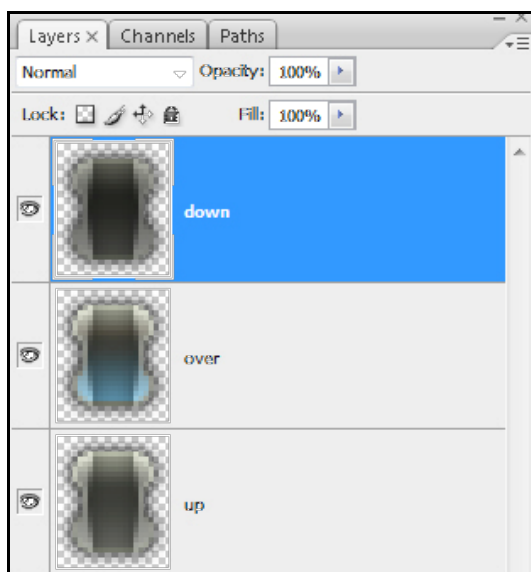


图 34: 音量滚动条滚动状态皮肤 Photoshop 图层

1. 双击场景中的滚动条输入时间轴。
2. 双击滚动条的轨迹组件。
3. 导入为轨迹创建的 PSD 或者 AI 文件，对齐关键帧。轨迹的注册点应该在图像的左边，应该与滑动条的 registration 点相匹配。
4. 返回到滑动条时间轴（在上层）。
5. 双击滚动组件输入时间轴。
6. 导入滚动条 PSD 或 AI 文件。
 - a. **重要提示:** 当导入滚动条皮肤，确认改变了每个层的 registration 位置在中间。

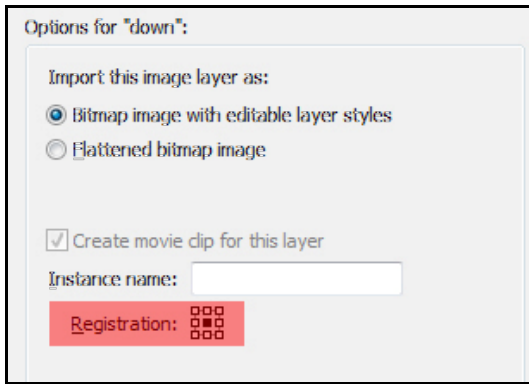


图 35: 设置 **registration** 点到中间位置

7. 选择每个关键帧上的滚动条图像，设置 X 坐标为‘0.0’。
8. 返回到运动条时间轴（向上一层）并选择滚动条组件进行移动以保持左上角边框与滚动条 **registration** 点匹配。



图 36: 将滚动按钮左上角与滚条基点对齐

9. 编辑 *textField* 标签的字体属性。
10. 测试动画。



图 37: 具有皮肤背景的音量调节条

5.3 绘制选择按钮和复选框背景

选择按钮和复选框在以下状态下为必不可少的按钮组件：*selected_up*、*selected_over*、*selected_down*、*selected_out* 和 *selected_disabled*。这里，无需介绍如何绘制背景，只要遵照 5.1 节中的描述步骤即可；但是，必须包含新的状态作为背景绘制过程的一部分。通常，选择按钮包括在按钮内部用一个小点表示被选中，复选框则包括一个确认标记来表示同样的含义。但是，设计不是都必须按照这个范例进行。

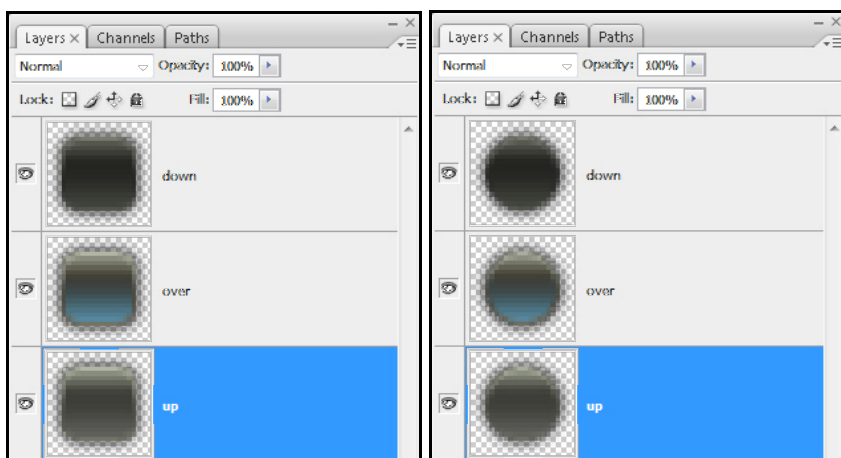


图 38: Photoshop 图层中的选择按钮和复选框界面

最后，注意编辑 *textField* 标签实例中的字体属性。



图 39: 具有皮肤背景的 edge AA 选择按钮

5.4 描绘难度选择框皮肤背景

难度选择框，主要由按钮部件组成。需要两个按钮：一个左按钮，一个右按钮。第三个图形元素用来渲染组件背景/边框，当不是必不可少的。

1. 用 5.1 节中提到的方法创建一个左和右方向按钮。
 - a. 或者，可以使用相同的图像来表示两个按钮，只需要在 Flash 中对图像镜像处理即可。
2. 在场景上，双击难度选择框输入时间轴。下一步，如果与设计相匹配，则替换底部背景/边界图像，下面可以看到部件周围的灰度边界。选择左按钮并在按钮上双击输入时间轴。现在简单得导入一个新的拥有皮肤背景的 PSD 文件到状态图层，如 5.1 节中所述，对齐关键帧。
3. 接下来，如果适合设计，则替换底部背景/边框图形，见下图中逐渐为灰色边框。
4. 双击进入时间轴选择 *left* 按钮。
5. 现在只需导入新皮肤 PSD 文件到 5.1 小结中描述的 *States* 图层并对其关键帧。
6. 对右按钮重复步骤 4-5。
7. 返回到上级时间轴上的两个按钮 — Difficulty option stepper 时间轴 — 并编辑标签名为 *textField* 的属性字体
8. 测试动画。



图 40: 具有皮肤背景的难度选择框

6. 结论



图 41:具有皮肤背景的选择菜单

祝贺你！本文介绍了一种使用 **Scaleform CLIK** 的方法。尽管还有很多种替换方案，在大多数工程中都有其独具挑战的方法。但是，熟悉文中所介绍的 **Scaleform CLIK** 的基本工作流程还只局限于基本的适当应用和 **Scaleform CLIK** 框架的正式使用。需要深入学习，推荐技术方面的文档，实验 **Scaleform CLIK** 自带的各种演示文件，浏览每个组件的源代码并阅读其中的注释。提供了完善的操作步骤来确保最终用户成功应用 **Scaleform CLIK**。

演示文件和源代码所在 **Scaleform 4 SDK** 目录：

Windows系统：

- 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*

苹果系统：

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

附加文档：

- [Getting Started with CLIK](#) 按钮入门 – CLIK 中按钮逐渐应用入门。