

Autodesk® Scaleform®

Getting Started with CLIK

本書では、Scaleform® Common Lightweight Interface Kit (CLIK™) を設定して使用方法について説明しています。

著者: Matthew Doyle
バージョン: 3.0
最終更新日: 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 の連絡先:

ドキュメント	Getting Started with CLIK (CLIK スタートガイド)
住所	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
ホームページ	www.scaleform.com
電子メール	info@scaleform.com
電話	(301) 446-3200
Fax	(301) 446-3199

目次

1. はじめに	4
1.1 ソフトウェアの最小要件	4
2. CLIK の構成と Scaleform Launcher.....	2
2.1 Scaleform CLIK クラス パスを Flash CS3 に追加する	2
2.2 Scaleform CLIK クラス パスを Flash CS4 に追加する	4
2.3 Scaleform Launcher パネルを追加する	5
3. CLIK コンポーネントを使ってメニューをつくってみる	6
3.1 メニュー シェルを作成する	8
3.2 Main Menu のナビゲーション コントロールを追加する.....	11
3.3 Options Screen ボタンをリスンする	15
3.4 難易度オプション設定を追加する	16
3.5 ビデオ設定のチェックボックスとラジオ ボタンを追加する	18
3.6 音量スライダを追加する	21
3.7 OK ボタンとキャンセル ボタンを追加する	23
4. ActionScript で機能を追加する	24
4.1 難易度レベルを難易度オプション設定に追加する	24
4.2 aaGroup にあるラジオ ボタン グループを設定する	25
4.3 Options の初期ステートを設定する	26
4.4 Options Screen を終了する.....	27
5. メニューのスキニング	29
5.1 Options ボタンをスキニングする	30
5.2 音量スライダをスキニングする.....	34
5.3 ラジオ ボタンとチェックボックスをスキニングする	37
5.4 難易度オプション設定をスキニングする	38
6. 最後に	39

1. はじめに

Autodesk® Scaleform® Common Lightweight Interface Kit (CLIK™) を使うと、開発者はゲームの効果的なフロントエンド メニュー インターフェイスをすばやくプロトタイプ化して、そのプロトタイプを最終形までもっていくことができます。本書では、Scaleform CLIK コンポーネントと ActionScript™を使ったゲームのカスタム フロント エンド作成の基礎と、さらにコンポーネントの作成とスキニングのベストプラクティスを提供しています。また、作成の段階ごとに順を追った詳細な指示も提供しています。

ご注意： このチュートリアルでは、私どもが現時点で最善と思われる方法をお薦めしております。しかしながら、Flash でのオーサリングでは同じ結果を得るためにいくつかの手法が存在するのも事実です。そのため、開発者がより良い方法を見出すことも充分あり得ることです。

1.1 ソフトウェアの最小要件

- Scaleform 3 以上
- Adobe® Flash® CS3 以上
- Adobe Photoshop® CS3 以上

2. CLIK の構成と Scaleform Launcher

Scaleform CLIK は、ワークフローの向上のため設計された新規の Adobe Flash パネルを搭載しています。Scaleform CLIK コンポーネントとこのワークフロー拡張パネルを使用するには、まず始めに Scaleform CLIK クラスパスが適切に Flash 環境に追加されていることを確認します。次に、Adobe Extension Manager を使ってパネル拡張機能をインストールします。

Creative Suite® 3 (CS3) と CS4 の場合のインストールはやや異なります。本書では両方のインストール手順を説明しています。

2.1 Scaleform CLIK クラスパスを Flash CS3 に追加する

Flash クラスパスを Scaleform CLIK クラスに対して設定し、このオーサリング環境に公開する必要があります。

1. Flash CS3 を起動します。
2. Flash メニューの [編集] をクリックして、ドロップダウン メニューから [環境設定] を選択します。
3. [カテゴリ] の [ActionScript] を選択します。
4. [言語] の [ActionScript 2.0 設定] をクリックします。
5. [ActionScript 2.0 設定] ダイアログで、[+] ボタンを押して新規のパスを追加します。



図 1: CS3 (Windows)の ActionScript 2.0 クラス パス ウィンドウ

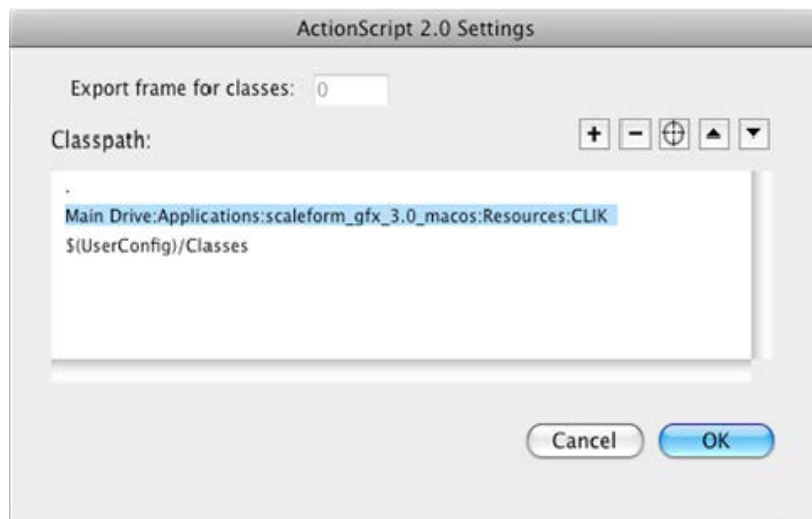


図 2 : CS3 (Mac) の ActionScript 2.0 クラスパス ウィンドウ

6. [パスの参照] アイコンをクリックして、Scaleform CLIK コンポーネントがインストールされているディレクトリを選択します。このアイコンは CS3 では十字形になっています。Windows 用バージョンでは、デフォルトでは以下の場所にインストールされています：
`C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/`
 Mac の場合には、Scaleform 4 SDK をインストールしたディレクトリの、
`scaleform_gfx_4.2_macos/Resources/AS2/CLIK`
7. [ActionScript 2.0 設定] ダイアログに戻り、[クラスパス] の追加されたパスを選択します。▲ / ▼ ボタンを使ってこのパスを、Windows では“\$(LocalData)/Classes” 行の上に移動します。Mac では、“\$(UserConfig)/Classes” 行の上です。これにより Scaleform CLIK クラスは内蔵クラスの前に参照されるようになります。このステップを省略すると、Scaleform CLIK コンポーネントではなく内蔵の Adobe コンポーネントが使用されます。
8. [OK] ボタンをクリックして [ActionScript 2.0 設定] ダイアログを閉じます。
9. [OK] ボタンをクリックして [環境設定] ウィンドウを閉じます。

2.2 Scaleform CLIK クラス パスを Flash CS4 に追加する

Flash クラス パスを Scaleform CLIK クラスに対して設定し、このオーサリング環境に公開する必要があります。

1. Flash CS4 を起動します。
2. Flash メニューの [編集] をクリックして、ドロップダウン メニューから [環境設定] を選択します。
3. [カテゴリ] の [ActionScript] を選択します。
4. [言語] の [ActionScript 2.0 設定] をクリックします。
5. [ActionScript 2.0 設定] ダイアログで、[+] ボタンを押して新規のパスを追加します。

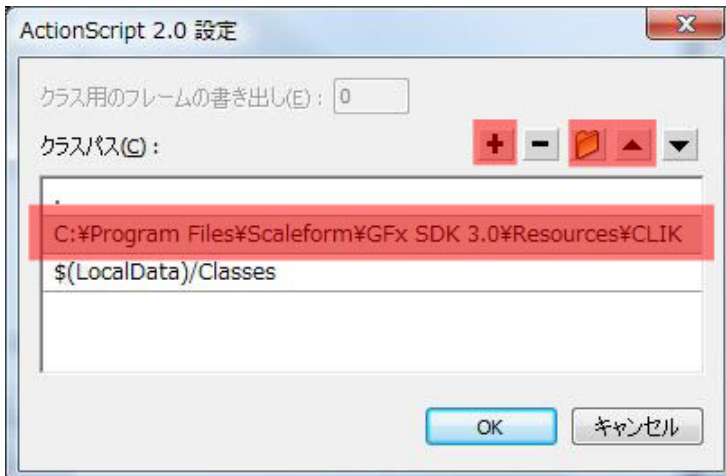


図 3: CS4 (Windows) の ActionScript 2.0 クラス パス ウィンドウ

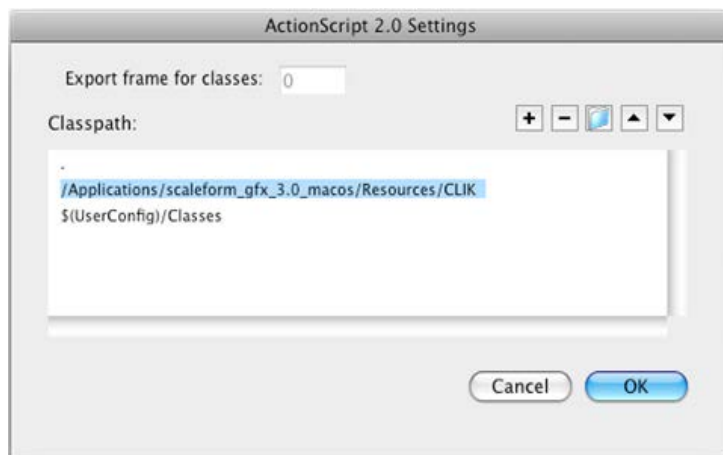


図 4: CS4 (Mac) の ActionScript 2.0 クラス パス ウィンドウ

6. [パスの参照] アイコンをクリックして、Scaleform CLIK コンポーネントがインストールされているディレクトリを選択します。このアイコンは CS4 ではフォルダ アイコンになっています。Windows ではデフォルトのインストール先は:
`C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/`
Mac では、`scaleform_gfx_4.2_macos/Resources/AS2/CLIK/`です。
7. [ActionScript 2.0 設定] ダイアログに戻り、[クラスパス] の追加されたパスを選択します。▲ / ▼ ボタンを使って、このパスを Windows では“\$(LocalData)/Classes” 行の上に移動します。Mac では、“\$(UserConfig)/Classes”行の上です。これにより Scaleform CLIK クラスは、内蔵されているクラスの前に参照されるようになります。このステップを省略すると、Scaleform CLIK コンポーネントではなく内蔵の Adobe コンポーネントが使用されます。
8. [OK] ボタンをクリックして [ActionScript 2.0 設定] ダイアログを閉じ、次に [OK] ボタンをクリックして [環境設定] ウィンドウを閉じます。

2.3 Scaleform Launcher パネルを追加する

Scaleform Launcher パネルを使うと、Flash のオーサリング環境の中から手早く便利に SWF ファイルをパブリッシュして直接 Scaleform プレイヤーで実行することができます。Scaleform Launcher のインストール方法については、[Getting Started with Scaleform 4.1 \(Scaleform 4.1 スタートガイド\)](#) の 2.4 章から 2.6 章で、詳しく説明しています。Scaleform Launcher パネルをインストールされていないユーザーは、参照してインストールしてください。

3. CLIK コンポーネントを使ってメニューをつくってみる

この章はフロントエンド ゲーム メニューの作成に必要な手順を詳細に説明しています。メニューはメイン メニュー画面とオプション画面の2つの画面で構成されます。メイン メニュー画面には、オプション画面に進むボタンが1つ含まれます。このメニューの作業の大部分はオプション画面上で行われます。この画面にはオプション ステッパーで実装される難易度の設定、チェックボックスやラジオ ボタンを使ったビデオ オプションのブルームやアンチエイリアス設定、さらに音量のスライダ コントロールが含まれます。

このチュートリアルのアセットは以下の Scaleform 4.2 SDK インストール ディレクトリ内の CLIK フォルダにあります:

Resources/AS2/CLIK/tutorial

Resources/AS2/CLIK/tutorial/art_assets

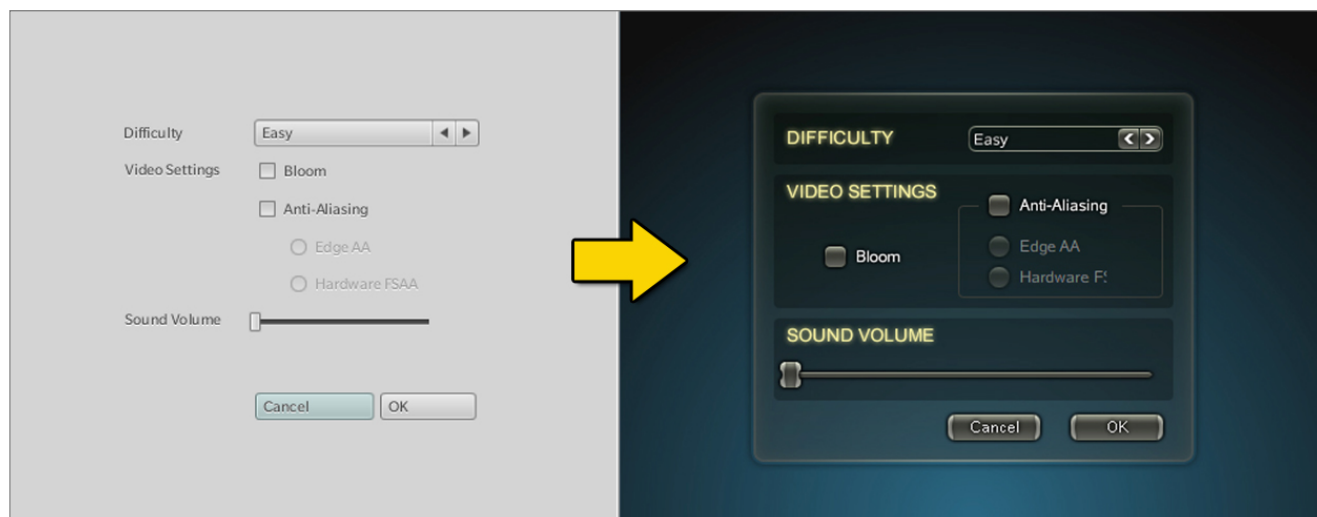


図 5: スキニング前とスキニング後の最終の Options Screen

このチュートリアルで作成するカスタムユーザー インターフェイスは、以下のワークフローを備えています:

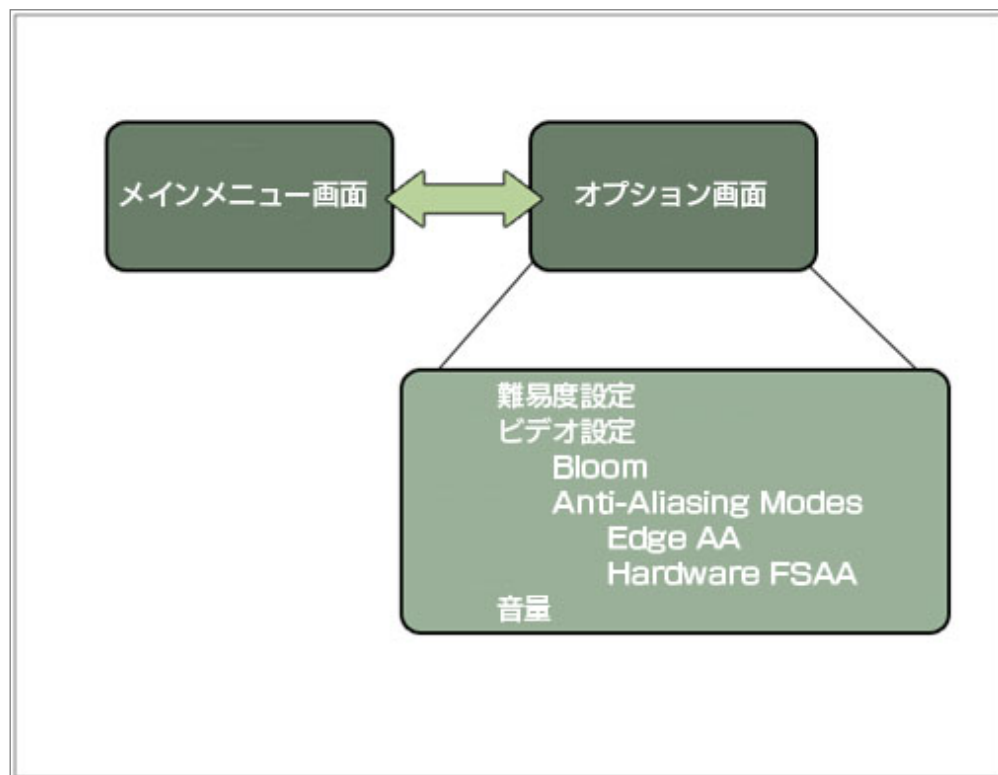


図 6: フロント エンド メニューのフロー

3.1 メニュー シェルを作成する

第一段階はこのメニューのさまざまなレイヤーとキーフレームをセットアップすることです。フロント エンドの大まかな骨格設定をここで作成します。

1. 新規の *Flash (ActionScript 2.0)* ドキュメントを作成します。
2. このドキュメントのパブリッシュ設定が *Flash Player 8* に設定されていることを確認します。トップ メニューの [ファイル] を選択し、ドロップダウンから [パブリッシュ設定] を選択して [Flash] タブでこの設定を確認できます。
3. [バージョン] ドロップダウンを [Flash Player 8] に設定します。
4. [ActionScript のバージョン] ドロップダウンを [ActionScript 2.0] に設定します。
5. [OK] ボタンをクリックして [パブリッシュ設定] ウィンドウを閉じます。



図 7: [パブリッシュ設定] ウィンドウ

6. メイン ドキュメントのタイムラインで現在「レイヤー 1」という名前のレイヤーを選択し、その「レイヤー 1」という単語をダブルクリックして、「scene」という名前に変更します。この名前が重要なわけではありませんが、レイヤーにこのような名前を付けるのが最も効率的だと思います。
7. [レイヤーの追加] ボタンをクリックして新規レイヤーを作成し、「actions」という名前に設定します。もう一度言いますが、このレイヤー名は重要ではありません。ただし、レイヤーの持つ目的が分かりやすくなります。

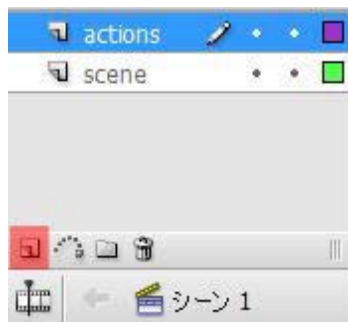
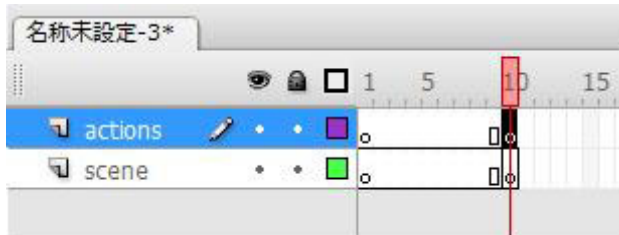


図 8: [レイヤーの追加] ボタン

8. *actions* と *scene* の両方のレイヤーのフレーム 10 に 2 つ目のキーフレームを追加します。*scene* レイヤーを選択して、タイムラインのフレーム 10 で右クリックします。ポップアップメニューから [キーフレームの挿入] を選択します。このキーフレームを使って、メニューの *Options Screen* の部分を作成します。*actions* レイヤーに対してこのステップを繰り返します。

図 9: フレーム 10 にキーフレームを追加したメイン メニューのタイムライン



9. *scene* レイヤーのフレーム 1 のキーフレームに '*mainMenu*' というラベルを付けます。この名前はそのキーフレームを選択し、[プロパティ] タブで入力します。コードはこのラベルを使って、*Scaleform* にタイムラインの *mainMenu* フレームに「進む」ように指示します。フレーム 1 を使って、メニューの *Main* 画面を作成します。



図 10: キーフレーム 1 に 'mainMenu' というラベルを付ける

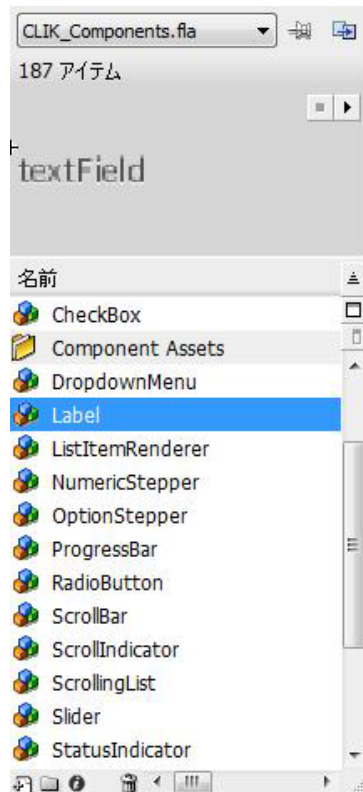
10. *scene* レイヤーのフレーム 10 のキーフレームに '*optionsScreen*' というラベルを付けます。コードはこのラベルを使って、*Scaleform* にタイムラインの *optionsScreen* フレームに「進む」ように指示します。
11. *actions* レイヤーのフレーム 1 を選択します。[アクション] ウィンドウ (F9) を開き、コード "*stop*();" (引用符なし) を一行目に入力します。このコードは *Player* にこのフレームで再生を止めるように指示します。このコードがそこになければ、*Player* は *stop* コマンドに達するまで、あるいはタイムラインの最後、この例ではフレーム 10 に到達するまで再生を続けます。*actions* レイヤーのフレーム 10 を選択して、同じコードを追加します。
12. この時点で、この *Flash* ファイルを保存することをお勧めします。

3.2 Main Menu のナビゲーション コントロールを追加する

ここでは *Main Menu* に *Option* ボタンを追加する方法を説明します。このボタンを使うとユーザーは *Options Screen* に進むことができます。

1. Scaleform SDK 4.0 インストール ディレクトリにある *Resources/AS2/CLIK/components* の *CLIK_Components.fla* を開きます。
2. 最初にメニューのタイトルを追加します。*CLIK_Components.fla* の [ライブラリ] パネルで *Label* コンポーネントを右クリックして、ポップアップ メニューから [コピー] を選択します。

図 11: [ライブラリ] パネルと *Label* コンポーネント



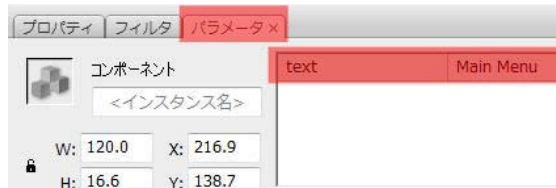
3. *CLIK_Components.fla* ファイルを閉じないでください。また必要になるからです。使用中の元の Flash ドキュメントに戻り、このチュートリアルメニューを作成します。scene レイヤーの最初のキーフレームを選択します。
4. [ライブラリ] パネルの空白のエリアで右クリックし、[ペースト] を選択して *Label* コンポーネントをそのドキュメントの [ライブラリ] にコピーします。
5. [ライブラリ] パネルから *Label* コンポーネントをクリックして、ステージのどこか (上部近くが望ましいです) にドラッグします。

- この新規に作成した Label の [パラメータ] タブを開き、*text* フィールドに 'Main Menu' と入力します。このフィールドが実行時に Player で表示されるものです。

注意：この [パラメータ] タブは、CS4 では [コンポーネントインスペクタ] にあり、デフォルトではオフになっています。Shift + F7 キーを押して表示するか、または [ウィンドウ] メニューの [コンポーネントインスペクタ] を選択します。CS3 では、このタブは [プロパティ] タブの近くにありますが。

注意：これらのパラメータは、SWF がパブリッシュされ再生される場合に限って表示されます。

図 12: CS3 では、[パラメータ] タブは [プロパティ] タブの近くにある



- キーフレーム 10 を選択し、ステップ 6 と 7 を繰り返して、Options Screen のタイトルを作成します。今回はテキスト フィールドに 'Options' と入力します。
- CLIK_Components.fla ファイルを開き、[ライブラリ] パネルの *Button* コンポーネントを右クリックして、再度 [コピー] を選択します。

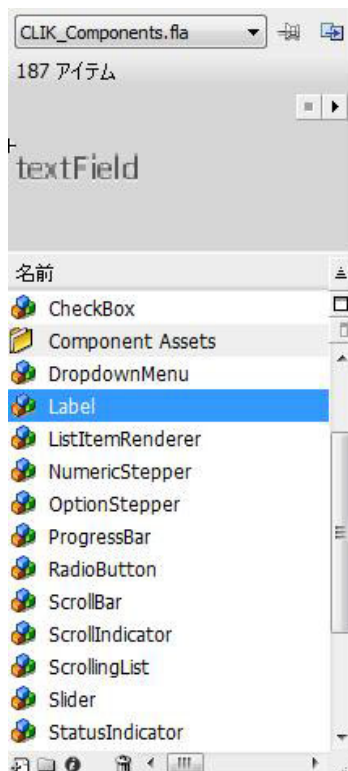


図 13: [ライブラリ] パネルの Button コンポーネント

- 元の Flash ドキュメントを選択します。

10. [ライブラリ] パネルの空白のエリアで右クリックし、[ペースト] を選択して *Button* コンポーネントをそのドキュメントの [ライブラリ] にコピーします。
11. *scene* レイヤーの最初のキーフレームを選択します。
12. [ライブラリ] パネルから *Button* コンポーネントをクリックして、ドキュメントのステージ上の *Main Menu* ラベルの下にドラッグします。
13. ステージ上のこの新規のボタンを選択して、[プロパティ] タブでインスタンス名を 'optionsBtn' に変更します。このインスタンス名は、このボタンの動作を定義するコードが参照します。



図 14: Options ボタン インスタンスの名前を 'optionsBtn' に設定する

14. [パラメータ] タブをクリックして、*label* フィールドに 'Options' と入力します。このフィールドが実行時にボタン上で表示されるものです。



図 15: このボタンに 'のボタンに。このというラベルを付ける

15. 設定が完了しました。 *Scaleform Launcher* パネルを使って最初の画面をテストします。 *Scaleform Launcher* パネルの [Test with (実行ファイル名)] ボタンを押して、 *Scaleform Scaleform Player* でこのムービーを実行します。

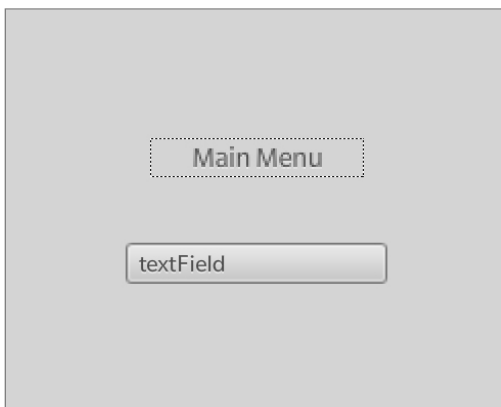


図 16: Scaleform Player に表示された Options Screen ボタンを含む Main Menu

3.3 Options Screen ボタンをリスンする

最初に必要な対話機能は、Main Menu の Options ボタンに機能性を加えることです。このボタンを押すとユーザーは *Options Screen* に進むことができます。このサンプルでは Scaleform CLIK イベント モデルを使ってこの機能を実行します。他の方法を使用することもできますが、Scaleform CLIK のコンポーネントを使用するときは、この Scaleform CLIK イベント モデルの方がずっと柔軟性があり効率的です。このモデルでは、特定のイベントが発生したときに、Options ボタンはコールバック関数を呼び出すように指示されます。このコールバックはメインのタイムラインに、Options Screen のキーフレームに進むように指示します。イベントはマウスのクリックで、ユーザーが左のマウス ボタンを押すと発生します。

1. *actions* レイヤーの最初のキーフレームを選択します。
2. 以下のコードを `stop()`; コマンドの上部に挿入します。これはユーザーをフレーム 10 に位置する Options Screen に進める関数です:

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

3. 次に、"click" イベントで上記の関数 OpenOptionsScreen を 呼び出すように *optionsBtn* インスタンスに指示するコードを挿入します。

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

4. 最後に、コードの最後の行で Options ボタンに初期フォーカスを設定します:

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

5. このムービーをテストします。この Options ボタンはデフォルトで選択されるか、フォーカスが適用される必要があります。この Options ボタンを押すとユーザーは Options Screen に進むはずですが、今のところこの画面は空白です。

3.4 難易度オプション設定を追加する

OptionsStepper を使ってゲームの難易度のコントロール設定を *Options Menu* に追加します。このコンポーネントは、ダイナミックに使用されるデータ ドリブンなエレメントで、選択肢のリストから選んだテキスト フィールドを表示します。この例では表示される選択肢は Easy、Medium、Hard、Insane です。このコンポーネントは 2 つのボタン コンポーネントを使って、リストの各オプションを切り替えます。

1. CLIK_Components.fla ファイルに戻ります。
2. [ライブラリ] パネルの *OptionsStepper* コンポーネントを右クリックして、[コピー] を選択します。
3. 元の Flash ドキュメントに戻ります。
4. scene レイヤーのフレーム 10 のキーフレーム (*Options Menu* キーフレーム) を選択します。
5. *OptionsStepper* を [ライブラリ] パネルにペーストします。

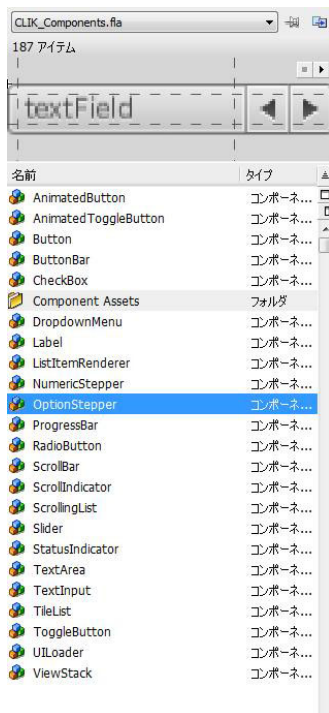


図 17: [ライブラリ] パネルの OptionStepper コンポーネント

6. この *OptionsStepper* がすでにステージ上にペーストされている場合は、このステップは省略します。そうでない場合、[ライブラリ] パネルから *OptionsStepper* コンポーネントをクリックして、ステージにドラッグします。
7. この *OptionsStepper* を選択して、そのインスタンス名を 'difficultyOption' に変更します。

8. [ライブラリ] パネルから新規の *Label* コンポーネントをクリックしてステージにドラッグし、難易度コントロールの左に置きます。このコンポーネントは、難易度設定コントロールを記述するテキストを表示します。



図 18: 難易度コントロール インスタンスに'インスタンス'にール置きます。このコンという名前を付ける

9. [パラメータ] タブでこの *Label* の *text* フィールドを'Difficulty'に変更します。



図 19: 最終的なスキニング前の難易度オプション設定

3.5 ビデオ設定のチェックボックスとラジオ ボタンを追加する

ビデオ設定のコントロールはチェックボックスを使用して、2 つの設定、[Bloom] と [Anti-Aliasing] のオン/オフを切り替えます。さらにゲームのアンチエイリアスの方法を変更する 2 つのラジオ ボタンも使用します。チェックボックスはボタンに望む機能が有効/無効を切り替えることである場合に使用し、ラジオ ボタンは一連のオプションをリストにグループ化して、グループ内のオプションを 1 つだけを有効にしたいときに使用します。

1. 難易度コントロールの下に新規の *Label* コンポーネント インスタンスを作成して、そのテキスト パラメータを 'Video Settings' に設定します。
2. CLIK_Components.fla ファイルを選択します。
3. [ライブラリ] パネルの *CheckBox* コンポーネントを右クリックして、[コピー] を選択します。
4. 実行中の Flash ドキュメントに戻り、この *CheckBox* を [ライブラリ] パネルにペーストします。
5. [ライブラリ] パネルから *CheckBox* コンポーネントをクリックして、ステージにドラッグします。

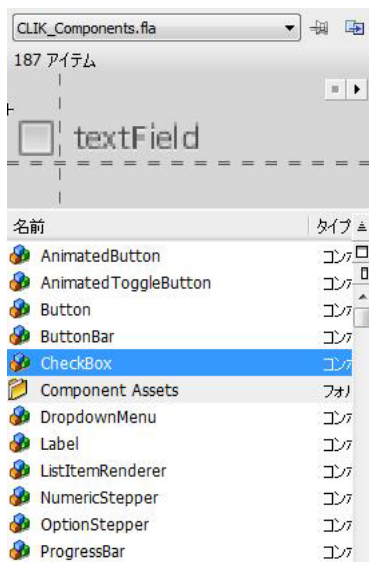


図 20: [ライブラリ] パネルの *CheckBox* コンポーネント

6. この新規インスタンスを選択して、そのインスタンス名を'bloomBtn'に変更します。



図 21: [Bloom] ラジオ ボタン インスタンス名を'インスタンス名'にする

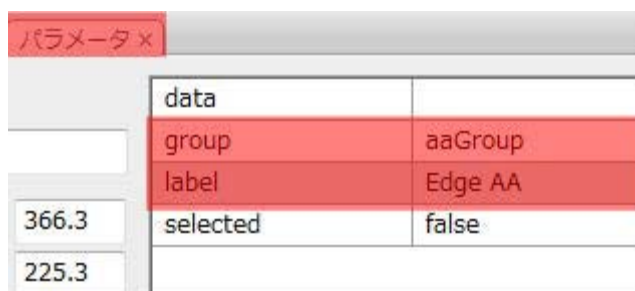
7. [パラメータ] タブで *label* フィールドを'Bloom'に変更します。
8. このインスタンスをコピーして Bloom チェックボックスの下に直接配置し、新規インスタンスの名前を 'aaBtn' に変更します。
9. [パラメータ] タブで *label* フィールドを'Anti-Aliasing'に変更します。
10. *CLIK_Components.fla* に戻り、これまで同様に[ライブラリ]パネルの、*RadioButton* コンポーネントを選択してコピーします。
11. 作業中のファイルに戻り、RadioButton コンポーネントを[ライブラリ]にペーストします。
12. ペーストした RadioButton コンポーネントをステージ上の Anti-Aliasing チェックボックスの下にドラッグして、新規インスタンスを作成します。
13. この *RadioButton* のインスタンス名を'edgeaaBtn'に変更します。



図 22: Edge AA radiobutton インスタンスの名前を 'edgeaaBtn' にする

14. [パラメータ] タブで *label* フィールドを'Edge AA'に変更します。
15. *group* フィールドを'aaGroup'に変更します。
16. このインスタンスをコピーして Edge AA ラジオ ボタンの下に直接配置し、インスタンス名を 'hwaaBtn'に変更します。
17. [パラメータ] タブで *label* フィールドを'Hardware FSAA'に変更します。

18. *group* フィールドを'aaGroup'に変更します。



パラメータ ×	
data	
group	aaGroup
label	Edge AA
selected	false

図 23: ラジオ ボタンの *group* を 'aaGroup' に設定する



図 24: 追加された最終のスキニング前のビデオ設定

3.6 音量スライダを追加する

音量はスライダ コンポーネントを使って追加します。スライダは Track と Thumb コントロールを使用します。両方とも簡単なボタン コンポーネントです。Thumb はスライダの長さを制御するので、ユーザーは Thumb をスライダの左右から外すことはできません。また、垂直方向にも制限されるので、上下に移動することはできません。

1. ステージ上でビデオ設定の下に Label コンポーネントの新規インスタンスをコピーするか、または作成して text パラメータを 'Sound Volume' という名前にします。

CLIK_Components.fla ファイルを選択して、Slider コンポーネントを実行中の FLA ドキュメントにコピー & ペーストします。

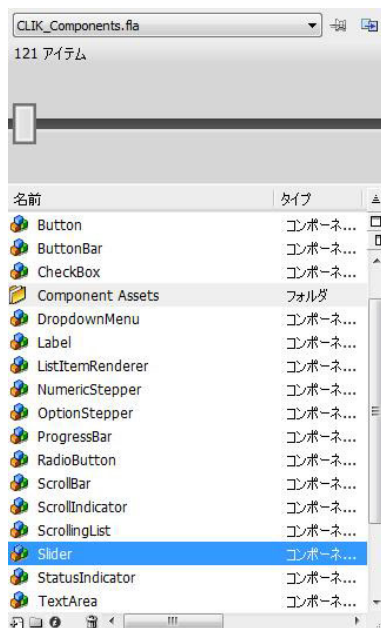


図 25: [ライブラリ] パネルの Slider コンポーネント

2. ステージ上の Slider インスタンスを Sound Volume ラベルの隣にドラッグして、インスタンス名を 'soundSlider' に変更します。

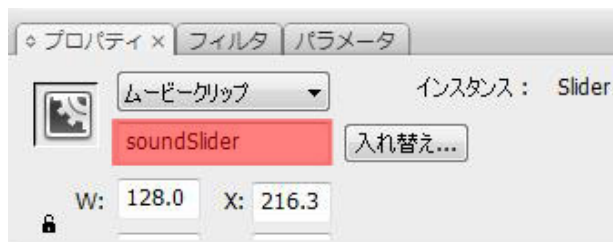


図 26: 音量スライダ インスタンスの名前を 'soundSlider' に変更する

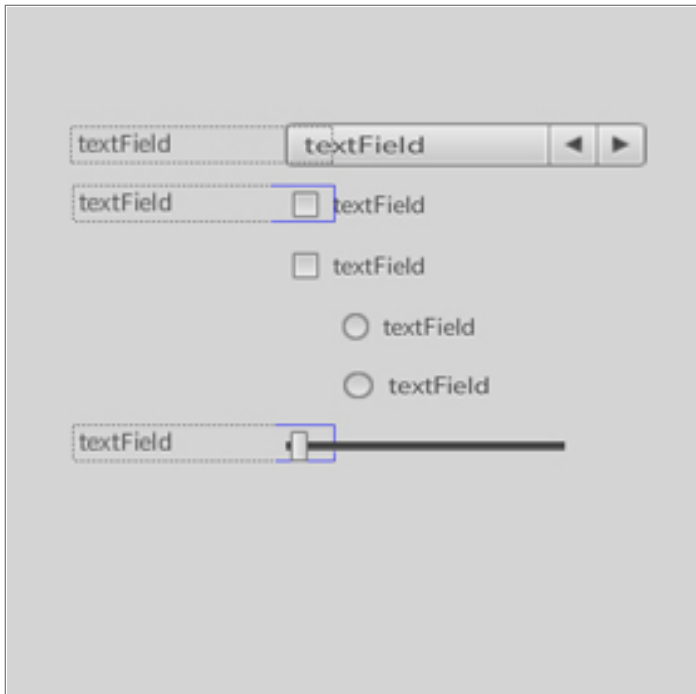


図 27: 追加された最終のスキニング前の音量スライダ

3.7 OK ボタンとキャンセル ボタンを追加する

OK ボタンとキャンセル ボタンは基本的なボタン コンポーネントを使用します。

1. [ライブラリ] パネルから Button コンポーネントをステージにドラッグします。
2. このボタンを選択します。
3. ボタンの [パラメータ] タブをクリックして *label* を 'OK' に設定します。
4. インスタンス名を 'okBtn' に変更します。
5. ステージ上のこのボタンをコピーして、インスタンス名を 'cancelBtn' に変更します。
6. コピーした新規ボタンの [パラメータ] タブをクリックして、*label* フィールドを 'Cancel' に設定します。

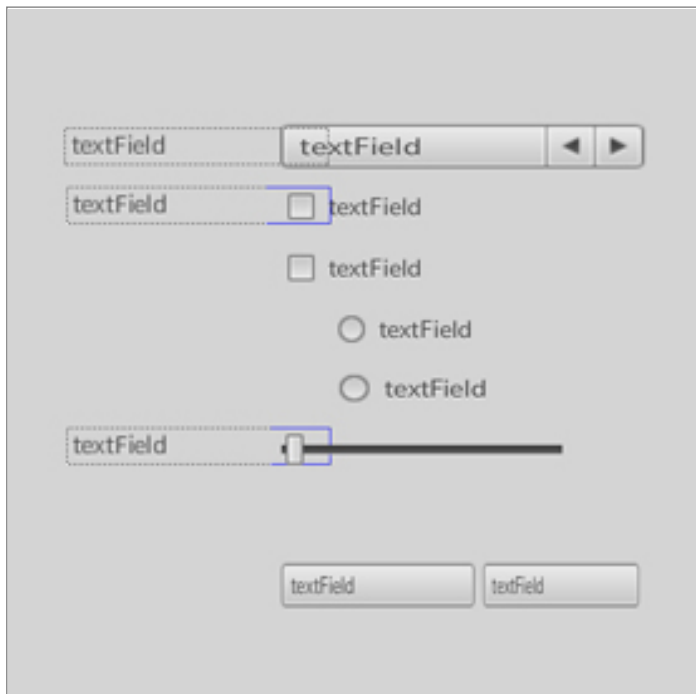


図 28: 最終的なスキニング前の Options Screen

4. ActionScript で機能を追加する

この時点でデフォルトのスキン状態で、それぞれのコンポーネントは作成されましたが、このメニューはまだ実際の機能を備えていません。これから ActionScript を追加します。[アクション] パネルが開いていない場合、F9 キーを押してこのパネルを表示します。

コードが動作しない場合、チュートリアル ファイル (C:\Program Files\Scaleform\GFx SDK 4.0\Resources\AS2\CLIK\tutorial\CLIK_tutorial_unskinned.fla) を開き、コードに目を通して比較しながら、入力ミスがないかどうかを確認してください。

4.1 難易度レベルを難易度オプション設定に追加する

難易度設定に設定されている難易度を表示させ、ユーザーにその設定を変更させるには、データをその設定に割り当てる必要があります。これは difficultyOption コンポーネントの 'dataProvider' プロパティを使って、必要なデータを渡すことで可能になります。この場合、渡すデータは Easy、Medium、Hard、Insane の難易度リストです。

1. *actions* レイヤーのフレーム 10 のキーフレームを選択します。
2. [アクション] パネル内の任意の場所をクリックして、コードの入力を始めます。
3. 以下のコードを `stop();` コマンドの上部に挿入します:

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

4. このムービーをテストします。このゲームの難易度設定オプション コントロールは、もう動作するようになっているはずです。左右のボタンを押すと 4 つの難易度を表示します。

4.2 aaGroup にあるラジオ ボタン グループを設定する

Anti-Aliasing ラジオ ボタンは、コードでグループ化しなければなりません。Anti-Aliasing のラジオ ボタンは、ユーザーが Anti-Aliasing チェックボックスをオンにしない限り無効 (変更不可) にする必要があります。

1. *actions* レイヤーのフレーム 10 のキーフレームを選択します。4.1 章のコードの直後で `"stop();"` コマンドの前に、以下のコードを入力します。
2. まず、Actions パネルの冒頭部分に、以下のコード行を挿入して、CLIK ButtonGroup クラスへのアクセスを付与します：

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

3. 次に、`difficultyOption.dataProvider` というコード行の下に移動して、新たに、'aaGroup' という名前のボタン グループを作成します。`edgeaaBtn` と `hwaaBtn` をそこにアサインします。：

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

4. ラジオ ボタンを有効/無効にする関数を作成します：

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

```

5. 最後に上記の関数を呼び出すイベント リスナーを Anti-Aliasing チェックボックスに追加します：

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

6. このムービーをテストします。このビデオ設定のアンチエイリアスのラジオ ボタンは、これでデフォルトでは無効になり、Anti-Aliasing チェックボックスをオンにすると有効にすることができるようです。

4.3 Options の初期ステートを設定する

オプション (options) は、ユーザーが OK ボタンを押して *Main Menu* に戻るまで、各ステートを保持していなければなりません。また、*Cancel* ボタンが押されたときには、そこでの変更は全て無視されるようであればなりません。

1. タイムラインの一番上 (*actions* レイヤーの上) に新規のレイヤーを作成します。これを、'vars' という名前に変更します。
2. *var* レイヤーのフレーム 1 を選択します。
3. 新規のオブジェクトを 'options' という名前で作成します。これは各種オプションのステートを保持する機能を持ちます。この新規オブジェクトがすでに存在しているかどうかテストするために、条件文に入れてみます：

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

4. *actions* レイヤーのフレーム 10 を選択します。
5. *difficultyOption* の初期ステートを設定します。options.selectedDifficulty プロパティが、まだ定義されていない場合には、difficultyOption の初期値をインデックス '0' (これは 'Easy' に相当します) にセットします。すでに定義されている場合には、options.selectedDifficulty があるがままで設定を続けます：

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

6. bloomBtn と aaBtn の初期ステートを設定します：

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

7. edgeaaBtn と hwaaBtn を無効または有効に設定します。これは、aaBtn チェックボックスの選択されたステートによって左右されます。

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

8. edgeaaBtn の初期ステートを設定します。options.edgeaa プロパティが、未だ定義されていない場合には、edgeaaBtn が選択されているステートを true としてください。すでに定義済みの場合には、options.edgeaa をあるがままで設定します。

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

9. 同様に、hwaaBtn の初期ステートを設定します。未定義の場合には、ここでは false とします。

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

10. soundSlider の初期ステートを設定します。

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

4.4 Options Screen を終了する

コードの最後の部分は、OK ボタン、または *Cancel* ボタンを押したときに、ユーザーが *Options Screen* を終了して *Main Menu* に戻れるようにします。

1. actions レイヤーのフレーム 10 のキーフレームを選択します。
2. OK ボタンが押されたときに、ユーザーをフレーム 1 (*Main Menu*) に戻す関数を 4.3 章のコードの後ろ、`"stop();" コマンドの前に、挿入します。`この関数は、options オブジェクトのプロパティを設定（そして保存）します。これによってユーザーは *Options Screen* に戻ったときには、options オブジェクトは同じ状態になります。:

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

3. *Cancel* ボタンが押されたときに、ユーザーをフレーム 1 (*Main Menu*) に戻す関数を挿入します。この関数は、いかなる変更も保存することはありません。

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

4. 次にユーザーの *OK/Cancel* ボタンのクリックをリスンします:

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

5. 最後に、デフォルトのフォーカスを *Cancel* ボタンに適用します。

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

6. このムービーをテストします。 *Options Screen* のコントロールはすべて動作するようになりました。

5. メニューのスキニング

このプロセスの最後の段階は、メニューと UI コンポーネントそれぞれに、デザイナーが作成したカスタムのスキンを適用することです。すべてのコンポーネントをスキニングする方法を説明するよりも、繰り返しを避けるため、本書では使用しているコンポーネントのサンプルのスキニング方法を説明します。このプロセスを一度理解すると、他のコンポーネントのスキニングも同じワークフローで行うことができます。

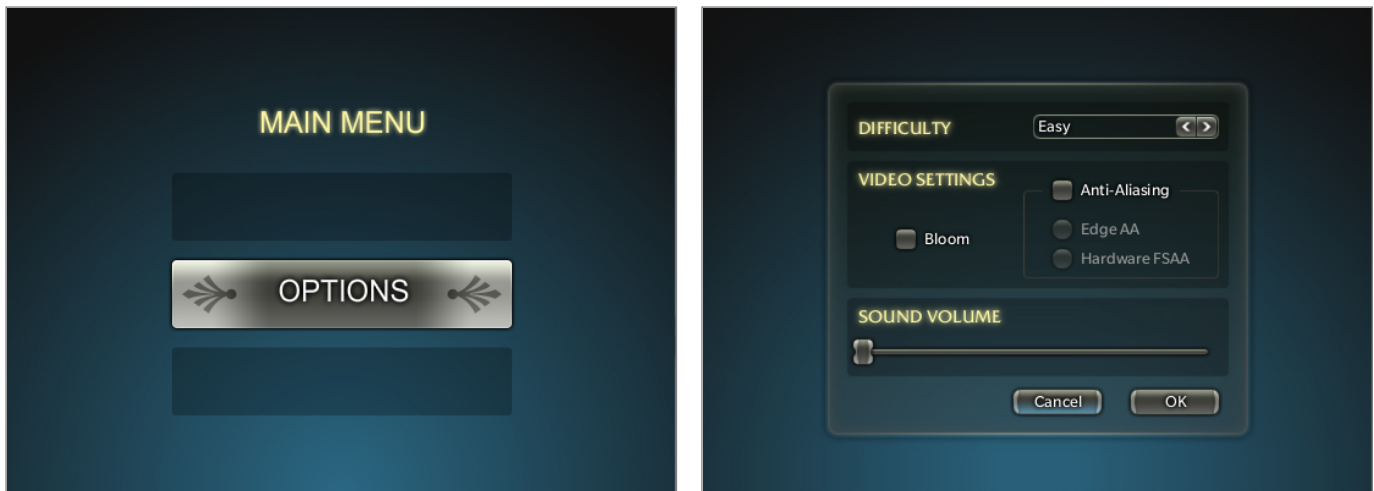


図 29: 最終のスキニング済みのメニュー

5.1 Options ボタンをスキニングする

Photoshop®、Illustrator®と Flash CS3 の組み合わせを使ったこのワークフローは、Photoshop、Illustrator と Flash CS4 の組み合わせでも同じように使えるはずです。

UI のすべてのオブジェクトは複数のステートを持つ場合があります。ボタンの場合、このようなステートは Up (デフォルト)、Over (選択された場合、またはマウス ポインタをその上に置いた場合)、そして Down (押した場合) です。各ステートは、ユーザーに視覚的なフィードバックを提供するそれぞれのアートを持っていないければなりません。このようなステートは Photoshop や Illustrator で作成することができます。最善の方法は、1 つのコンポーネントの各ステートのすべてのアートを、1 つの Photoshop または Illustrator ファイルに持たせることです。各ステートはそれぞれ、そのアート ファイルでは異なるレイヤーにしておく必要があります。これにより、Flash は各レイヤーをタイムライン上のキーフレームとして結合することができます。アートワークをインポートする前に、ステートごとに関連するすべてのレイヤーを 1 つのレイヤーに結合します。最善の方法は、作業アート ファイルとインポート アート ファイルを分けて保管しておくことです。今後、編集や調整が必要になる場合もあるので、作業ファイルにはオリジナルのレイヤーを残しておきます。インポート ファイルは、Flash へのインポートのために適切に結合された全レイヤーを備えている必要があります。単純なボタンであれば、最終的なインポート アート ファイルには Up、Over、Down の 3 つのレイヤーしかないはずです。

1. 同じ Photoshop Document (PSD) ファイルに *Options* ボタンの各グラフィック ステートを作成します。
 - a. 各コンポーネントの資料を参照して、コンポーネントが必要とするステートのリストを探るか、またはタイムライン上でスキニング前のコンポーネントの *Labels* レイヤーを参照します。
 - b. 重要：まったく同じ内容のステートに複数のレイヤーを作成することは避けてください。今回の場合、必要なのは up、over、down のレイヤーです。
2. 完成したらこの PSD を保存します。一つのステートが複数のレイヤーで構成されている場合には、次に同じステートのパーツであるすべてのレイヤーを 1 つのレイヤーに結合します。結合は、Photoshop の [レイヤー] パネルで Ctrl キーを押したまま結合する各レイヤーをクリックし、次に選択したレイヤーを右クリックしてポップアップ メニューから [レイヤーを結合] を選択して行います。
 - a. たとえば、over ステートのパーツであるすべてのレイヤーを over レイヤーに結合します。
3. Flash ではコンポーネントのタイムライン上の最初のステート (通常は *up*) が、Photoshop の [レイヤー] パネルの一番下に、タイムライン上の最後のステートが一番上に来るようにレイヤーを並べ替えます。この順序は Flash が各レイヤーをインポートする方法にとって重要です。



図 30: Options ボタンの Photoshop のレイヤー

4. (オプション) Flash のステート名と一致するように各レイヤーを、*up*、*over*、*down* などの名前に変更します。これは純粋に構成のためです。
5. ファイルを新規の PSD として保存し、ファイル名の末尾に '*_mergedLayers*' を追加します。結合していないレイヤーがもう必要ない場合を除いて、ステップ 2 で保存したオリジナルの PSD を上書きしないでください。
6. Flash でスキニング前の *Options* ボタンを開き、ステージ上でダブルクリックしてそのタイムラインを表示します。
7. タイムラインで *button* レイヤーを選択し、レイヤー名をクリックしてすべてのフレームを選択し、タイムラインでフレームを右クリックして、ポップアップメニューから [フレームの削除] を選択し、すべてのフレームを削除します。
8. このレイヤーに新規の空白のキーフレームを 1 つ追加して、それを選択します。これはステップ 9 のために必要です。
9. [ファイル] メニューをクリックして [読み込み] を選択し、ポップアップメニューから [ステージに読み込み] (Ctrl + R) を選択します。
 - a. [ステージに読み込み] が使用できない場合、インポートしようとしているレイヤーがロックされていないことを確認してください。レイヤー名の右にロックアイコンが付いている場合、そのロックをクリックして、レイヤーをロック解除に切り替えます。
10. スキニング UI エLEMENT の PSD または Adobe Illustrator (AI) ファイルを選択します。このチュートリアルの場合、PSD ファイルが使用されています。
11. [ライブラリに読み込む] ダイアログが開きます。インポートする各レイヤーがオンになっていることを確認します。
 - a. 重要：この読み込みダイアログは Flash 8 で PSD ファイルをインポートしているときは表示されませんが、CS3 と CS4 では表示されます。Flash 8 を使用して PSD をインポートする場合、この先のワークフローはやや異なります。

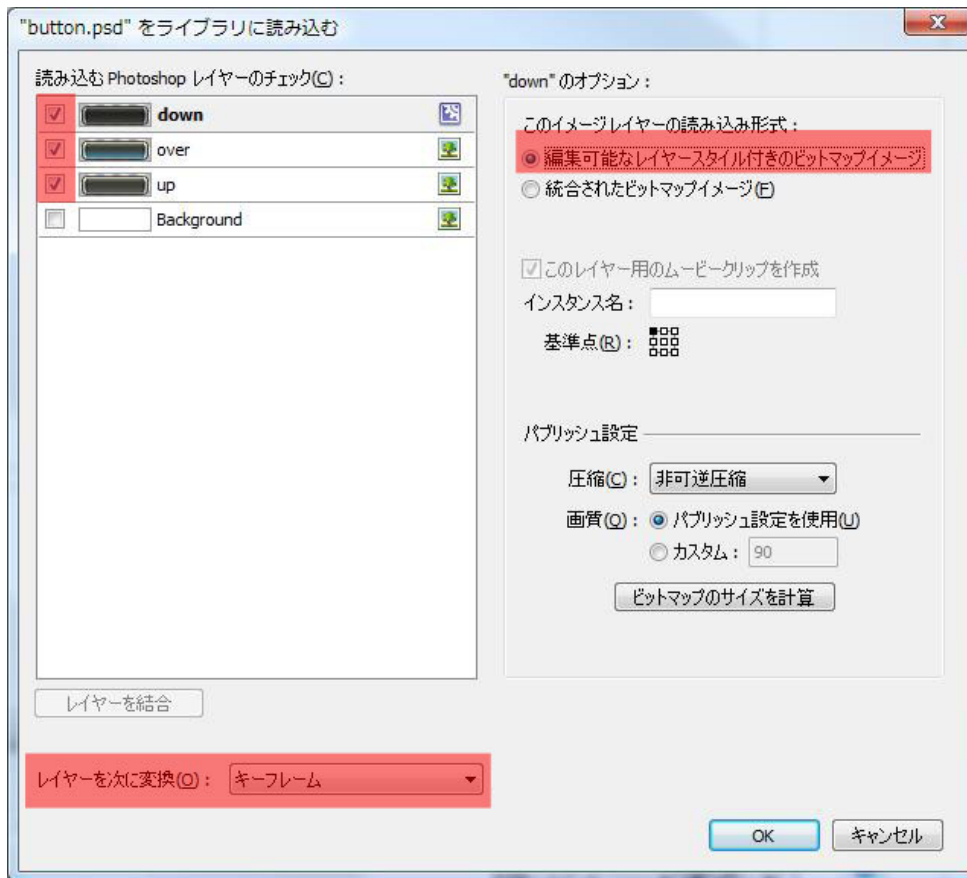


図 31: [ライブラリに読み込む] ダイアログ

12. [レイヤーを次に変換] ドロップダウン メニューを [キーフレーム] に設定します。このドロップダウン メニューはダイアログの左下にあります。
13. リストの各レイヤーをクリックして、[このイメージレイヤーの読み込み形式] で [編集可能なレイヤースタイル付きのビットマップイメージ] を選択します。
 - a. 重要：この手順を各レイヤーで行わなかった場合、たとえライブラリに追加したとしても、レイヤー スタイルを持つどのレイヤーもステージ上に表示されず、最初のレイヤー以外の各レイヤーは、下位の残りのレイヤーをすべて含んだフラット化したイメージになります。
14. レイヤーはタイムライン上に個別のキーフレームとしてインポートされます。最初のステートから最後のステートの順に並べられます。
 - a. 重要：PSD ファイルをインポートするとき、キーフレームはタイムラインの最上部の新規レイヤーに追加されます。単純に、古い *button* レイヤーを削除して、この新規に作成したレイヤーを *button* という名前に変更します。次にこの新規の *button* レイヤーをクリックし、レイヤー リストの一番下にドラッグします。
 - b. AI ファイルをインポートするとき、インポートする前に選択している限り、キーフレームは *button* レイヤーに追加されます。

15. 該当するラベルに揃うように、各キーフレームをクリックしてドラッグします。これらのキーフレームは *labels* レイヤーのタイムライン上に表示されます。最初のキーフレームを最初のフレーム (*up*) に移動します。2 つ目のキーフレームを *over* キーフレームの下に移動します。3 つ目の最後のキーフレームを *down* キーフレームの下に移動します。disabled はこのチュートリアルでは除外されています。

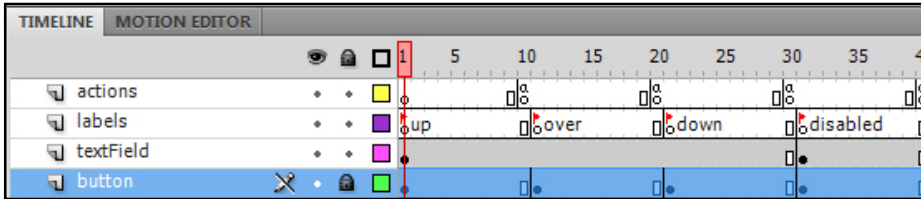


図 32: キーフレームを button レイヤーに追加する

16. ユーザーがこのボタンを使用したときの、ボタン ラベルの表示を変えたい場合、*Text* レイヤーの各キーフレーム、*up*、*over*、*down* の *textField* インスタンスのフォント設定を編集します。そうでない場合は、この *textField* をすべてのステートに対して同じにしておきます。
17. このムービーをテストします。これで *Options* ボタンは完全にスキニングされました。

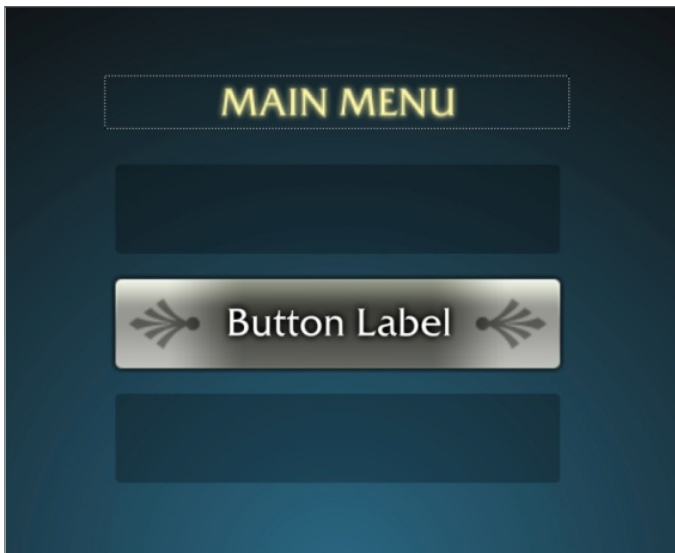


図 33: 最終のスキニング済み Options ボタンと Main Menu

5.2 音量スライダをスキニングする

音量スライダは基本的に 2 つのボタンで構成されています。1 つのボタンはスライダの Thumb (左から右へスライドする部分) を表します。もう 1 つのボタンは Track を表します。Thumb はこの上を移動するよう制限されています。今回は 5.1 章で説明した標準のボタンを作成する場合とほとんど同じ方法で、Thumb と Track の各ボタンを作成します。ただし、いくつか注意点があります。

図 34: 音量スライダの Thumb スキンの Photoshop レイヤー



1. ステージ上の音量スライダをダブルクリックして、そのタイムラインを表示します。
2. スライダ内の *Track* コンポーネントをダブルクリックして、そのタイムラインを表示します。
3. *Track* 用に作成された *PSD* または *AI* ファイルをインポートして、キーフレームを揃えます。
Track の基準点はこのグラフィックの左端にあり、スライダ コンポーネントの基準点と正確に一致する必要があります。
4. スライダのタイムライン (1 レベル上) に戻ります。
5. Thumb コンポーネントをダブルクリックしてそのタイムラインを表示します。
6. *Thumb* の *PSD* または *AI* ファイルをインポートします。
 - a. 重要 : Thumb スキンをインポートするときに、各レイヤーの基準点を必ず中央に変更します。

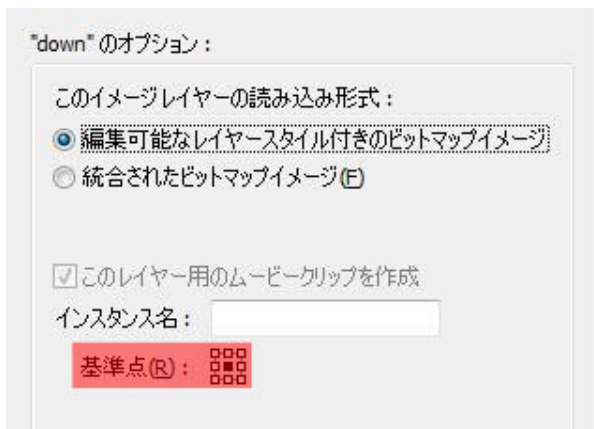


図 35: 基準点を中央に設定する

7. 各キーフレーム上で *Thumb* グラフィックを選択して、[プロパティ] タブの *[X]* を *[0.0]* に設定します。
8. スライダー タイムライン (1 レベル上) に戻り、*Thumb* コンポーネントを選択して、その境界ボックスの左上の角がスライダー コンポーネントの基準点と一致するように移動します。



図 36: Thumb の左上の角をスライダーの基準点に揃える

9. *textField* ラベルのフォントのプロパティを編集します。

10. このムービーをテストします。

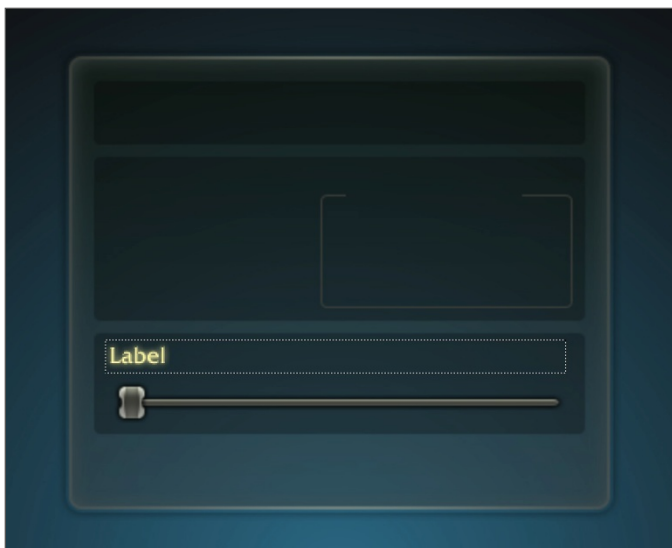


図 37: 最終のスキニング済みの音量スライダ

5.3 ラジオ ボタンとチェックボックスをスキニングする

ラジオ ボタンとチェックボックスは基本的に、*selected_up*、*selected_over*、*selected_down*、*selected_out*、*selected_disabled* という追加ステートを持つボタン コンポーネントです。したがって、このコンポーネントのスキニング方法を説明する必要はありません。5.1 章の標準のボタンの手順に従うだけですが、このスキニング プロセスの一部として、上記の新規のステートを必ず追加します。通常、ラジオ ボタンはボタンの内部に小さな点があり、そのボタンが選択されたことを表します。チェックボックスの場合は、同じことをチェック マークで表しています。ただし、最終的なデザインはこの範例に従う必要はありません。

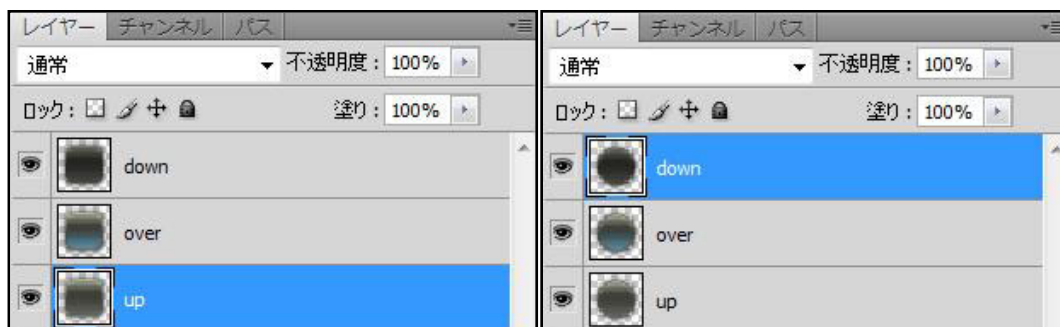


図 38: ラジオ ボタンとチェックボックス スキンの Photoshop レイヤー

最後に、*textField* ラベル インスタンスのフォントのプロパティを必ず編集します。



図 39: 最終のスキニング済みの Edge AA ラジオ ボタン

5.4 難易度オプション設定をスキニングする

難易度設定も同様にボタン コンポーネントで構成されています。このエレメントには左右の 2 つのボタンが必要です。3 つ目のグラフィックを使ってこのコンポーネントの背景/境界を表現することもできますが、必要ではありません。

1. 5.1 章で説明した方法で左右のボタンを作成します。
 - a. もう 1 つの方法として、両方のボタンに同じイメージを使用する場合、ボタンを 1 つだけ作成して Flash でミラーリングすることもできます。
2. ステージで難易度オプション設定をダブル クリックして、そのタイムラインを表示します。
3. 次に、デザインと一致していれば、最下部の背景/境界イメージを以下のようなコンポーネントの周囲のグレーの境界に置き換えます。
4. 左のボタンをダブルクリックして選択し、そのタイムラインを表示します。
5. ここで 5.1 章で説明したように新規のスキニング PSD ファイルを *States* レイヤーに読み込んで、キーフレームを揃えます。
6. 右ボタンに対してステップ 4-5 を繰り返します。
7. 2 つのボタンの親タイムライン、つまり難易度オプション設定のタイムラインに戻り、*textField* というラベルのフォントのプロパティを編集します。
8. このムービーをテストします。



図 40: 最終的なスキニング済みの難易度設定

6. 最後に



図 41: 最終のスキニング済みの Options メニュー

お疲れさまでした。本書は Scaleform CLIK の可能な使用法の 1 つを紹介しただけです。多くの方法があり、ほとんどのプロジェクトでは、それなりの困難にぶつかることでしょう。しかし、本書で紹介した方法で Scaleform CLIK の基本的なワークフローを理解していれば、Scaleform CLIK フレームワークの使用に必要な十分な準備と情報を、しっかりとお持ちのはずです。さらに理解を深めたい場合、もっと技術的な資料を読み、Scaleform CLIK に添付されているデモを試して、各コンポーネントのソースコードを確認し、その中のコメントを読むことをお勧めします。Scaleform CLIK を使いこなすために必要なものを、エンド ユーザーにすべて確実に提供するように手順が用意されています。

デモとソース コードは Scaleform 4.2 SDK ディレクトリにあります:

Windows では、

- デモ - `C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/demos`
- ソース コード - `C:/Program Files/Scaleform/GFx SDK 4.2/Resources/AS2/CLIK/gfx`

Mac では、

- デモ - `scaleform_gfx_4.2_macos/Resources/AS2/CLIK/demos`
- ソース コード - `scaleform_gfx_4.2_macos/Resources/AS2/CLIK/gfx`

その他の資料:

- [Getting Started with CLIK Buttons](#) - CLIK の Button コンポーネントのガイドです。