

Autodesk® Scaleform®

CLIK 시작하기

이 문서는 Scaleform® Common Lightweight Interface Kit (CLIK™)에 빠르게 적응하는 방법에 대하여 알려줄 것이다.

집필: Matthew Doyle

버전: 3.0

최종편집일: 2010 년 8 월 18 일

Copyright Notice

Autodesk® Scaleform® 4.3

© 2013 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, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk 123D, Autodesk Homestyler, Autodesk Intent, Autodesk Inventor, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSketch, AutoSnap, AutoTrack, Backburner, Backdraft, Beast, Beast (design/logo), BIM 360, Built with ObjectARX (design/logo), Burn, Buzzsaw, CADmep, CAiCE, CAMduct, 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, Design Server, DesignStudio, Design Web Format, Discreet, DWF, DWG, DWG (design/logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DWGX, DXF, Ecotect, ESTmep, Evolver, Exposure, Extending the Design Team, FABmep, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, ForceEffect, Freewheel, GDX Driver, Glue, 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, Map It, Build It, Use It, 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, Revit LT, RiverCAD, Robot,

Scaleform, Scaleform GFx, Showcase, Show Me, ShowMotion, SketchBook, Smoke, Softimage, Socialcam, 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.

연락처:

문서	CLIK 시작하기
주소	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
웹사이트	www.scaleform.com
이메일	info@scaleform.com
직통전화	(301) 446-3200
팩스	(301) 446-3199

목차

1. 개요	1
1.1 필요한 최소 소프트웨어	1
2. CLIK 설정하기와 Scaleform 런처 설치하기	2
2.1 Scaleform CLIK 클래스 패스를 Flash CS3 에 추가하기	2
2.2 Scaleform CLIK 클래스 패스를 Flash CS4 에 추가하기	4
2.3 Scaleform 런처 패널 추가하기	5
3. 컴포넌트로 간단한 전면 게임메뉴 만들기	6
3.1 메뉴셀 만들기	8
3.2 메인 메뉴 네비게이션 컨트롤 추가	11
3.3 옵션화면 버튼 대기	14
3.4 난이도 옵션 스텝퍼 추가	15
3.5 비디오 세팅 체크박스과 라디오 버튼 추가	17
3.6 사운드 볼륨 슬라이더 추가	20
3.7 OK 와 Cancel 버튼 추가하기	22
4. 액션스크립트로 기능 추가하기	23
4.1 난이도 옵션스텝퍼에 난이도 레벨 추가하기	23
4.2 aaRadio 버튼 그룹 설정하기	24
4.3 옵션의 초기상태 설정하기	25
4.4 옵션 화면 종료	26
5. 메뉴에 스킨 입히기	28
5.1 옵션버튼에 스킨입히기	29
5.2 볼륨 슬라이더에 스킨적용	34
5.3 라디오버튼과 체크박스 스킨적용	37
5.4 난이도 옵션 스텝퍼 스킨적용	38
6. 결론	39

1. 개요

Autodesk® Scaleform® Common Lightweight Interface Kit (CLIK™)을 사용하면 개발자들이 게임을 위한 전면 인터페이스를 빠르게 만들 수 있다. 이 문서에서는 CLIK 과 액션스크립트를 사용해서 커스텀 게임의 인터페이스를 만들기 위한 기초적인 내용을 다룰 것이다. 또한, 컴포넌트를 만들거나 스킨작업에 대해서도 다룰 예정이다. 무엇보다, 이 문서에서는 대단히 상세한 작업과정을 묘사할 예정이다.

주의 할 점은 플래시에서 같은 작업 물을 만드는데 여러 가지의 방법이 있다는 것인데, 개발자들은 이 튜토리얼의 접근 방식보다 좋은 대안이 될 수 있는 작업흐름을 찾아서 적용할 수도 있다.

1.1 필요한 최소 소프트웨어

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

2. CLIK 설정하기와 Scaleform 런처 설치하기

Scaleform CLIK 은 워크플로우를 개선하기 위해서 새로운 플래시 패널을 제공한다. CLIK 콤포넌트와 개선된 워크플로우 패널을 사용하려면 일단 CLIK 클래스 패스를 플래시 환경에 잘 설치해야 한다. 그리고 나서 어도비 확장 매니저를 사용해서 확장 패널을 설치한다.

Creative Suite® 3 (CS3) 와 CS4 의 설치과정이 살짝 다르기 때문에 여기서는 따로 설명하도록 하겠다.

2.1 Scaleform CLIK 클래스 패스를 Flash CS3 에 추가하기

저작환경에서 CLIK 클래스를 접근할 수 있도록 플래시 클래스 패스를 설정한다

1. 플래시 CS3 실행
2. 상단 메뉴의 *EDIT*→*Preferences* 선택
3. *Category* 의 *ActionScript* 선택
4. 언어에서 ActionScript 2.0 설정 클릭
5. ActionScript 2.0 설정 대화상자에서 [+]버튼을 눌러서 새로운 패스 추가

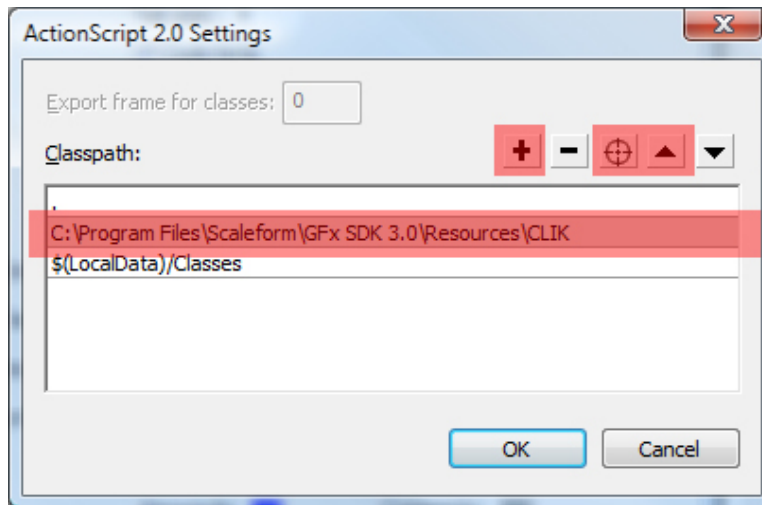


그림 1: CS3 의 ActionScript 2.0 클래스 패스

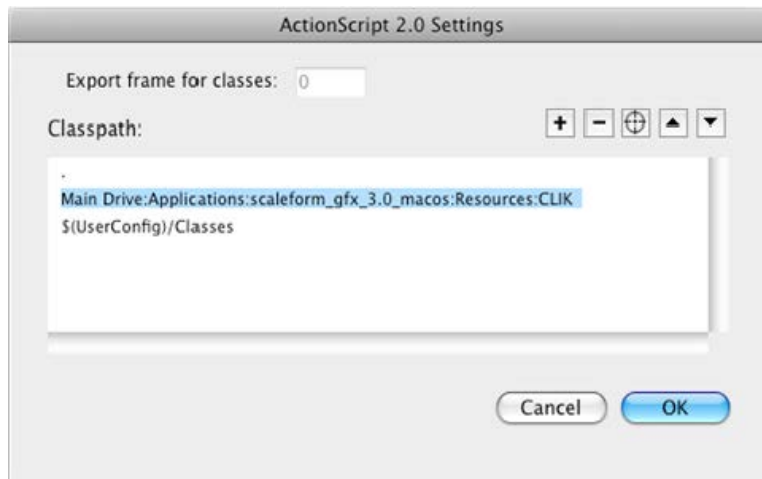


그림 2: Mac 상에서 CS3 의 ActionScript 2.0 클래스 패스

6. CS3 상에서 십자망선처럼 보이는 Browse to Path 아이콘을 클릭하여 Scaleform CLIK 컴포넌트가 설치된 폴더를 선택. 윈도우에서 기본 폴더에 설치되었다면 다음과 같음. *C:/Program Files/Scaleform/GFx SDK 4.3/Resources/AS2/CLIK/* 그리고 Mac 에서의 컴포넌트들은 Scaleform 4 SDK 설치 경로 안의 *scaleform_gfx_4.3_macos/Resources/AS2/CLIK/* 에 존재.
7. ActionScript 2.0 설정 대화상자로 돌아와 클래스 경로 아래에 경로삽입을 선택한다. 윈도우에서는 화살표 버튼을 이용해 *\$(LocalData)/Classes* 행 위로 Mac 에서는 *\$(UserConfig)/Classes* 행 위로 각각 올려준다. 이렇게 해서 기본 클래스보다 CLIK 의 클래스가 먼저 참조 가능해 짐. 이 과정을 빼먹으면 내부클래스가 먼저 참조될 수 있음.

8. *ActionScript 2.0* 설정 창의 OK 클릭.
9. *Preferences* 창의 OK 클릭.

2.2 Scaleform CLIK 클래스 패스를 Flash CS4 에 추가하기

저작환경에서 CLIK 클래스를 접근할 수 있도록 플래시 클래스 패스를 설정한다

1. 플래시 CS4 실행
2. 상단 메뉴의 EDIT→Preferences 선택
3. *Category* 의 *ActionScript* 선택
4. 언어에서 *ActionScript 2.0* 설정 클릭
5. *ActionScript 2.0* 설정 대화상자에서 [+]버튼을 눌러서 새로운 패스 추가

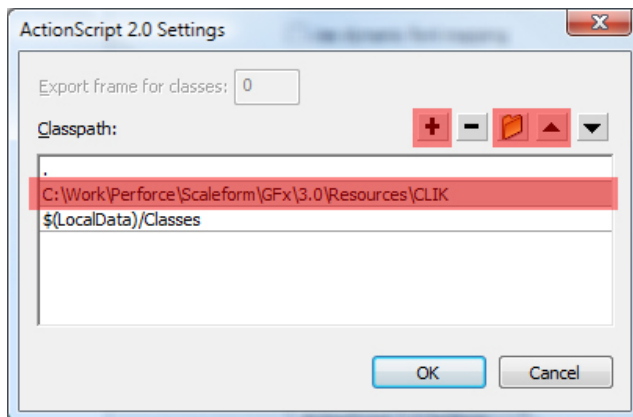


그림 3: CS4 의 ActionScript 2.0 클래스 패스

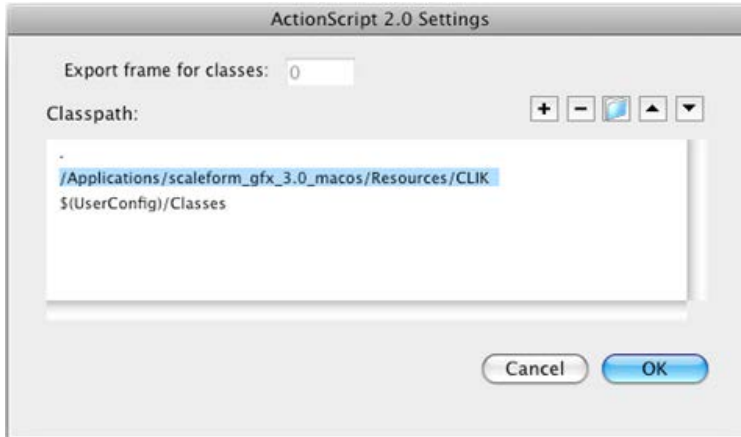


그림 4: Mac 상에서 CS4 의 ActionScript 2.0 클래스 패스

6. CS4 에서 폴더 아이콘처럼 생긴 *Browse to Path* 아이콘을 클릭하여 Scaleform CLIK 컴포넌트가 설치된 폴더를 선택. 기본 폴더에 설치되었다면 Windows 에서는 다음과 같은 경로에 위치. *C:/Program Files/Scaleform/GFx SDK 4.3/Resources/AS2/CLIK/* Mac 에서는 컴포넌트들이 *scaleform_gfx_4.3_macos/Resources/AS2/CLIK/* 상에 위치함.
7. ActionScript 2.0 설정 대화상자 종료후, Classpath: 밑의 추가된 패스를 선택. 화살표 버튼을 사용해서 Windows 인 경우 *\$(LocalData)/Classes* 위로, Mac 의 경우 *\$(UserConfig)/Classes* 라인 위로 이동. 이렇게 해서 기본 클래스보다 CLIK 의 클래스가 먼저 참조가능해 짐. 이 과정을 빼먹으면 내부클래스가 먼저 참조될 수있음.
8. ActionScript 2.0 설정 창의 OK 클릭후 *Preferences* 창의 OK 클릭

2.3 Scaleform 런처 패널 추가하기

Scaleform 런처 패널을 사용하면 빠르게 SWF 파일을 Scaleform player 로 보내는 것이 가능하다. 만약 패널을 아직 설치하지 않았다면 [Getting Started with Scaleform 4.3](#) 문서의 2.7 항부터 2.9 항까지 서술된 설명 내용을 참고하여 설치 방법과 사용 방법에 대하여 공부해보자.

3. 컴포넌트로 간단한 전면 게임메뉴 만들기

이번 장에서는 게임메뉴를 만들기 위해서 필요한 것들에 대하여 설명한다. 메뉴는 2 개의 화면으로 구성될 것이다. 하나는 메인 메뉴, 다른 하나는 옵션 메뉴다. 메인 메뉴는 사용자가 옵션 화면을 부를 수 있는 하나의 버튼만 제공될 것이다. 메뉴의 대부분의 역할은 옵션 스크린에 있을 것인데, 난이도 설정 옵션 스텝퍼, 블룸과 안티 앨리어싱을 선택할 수 있는 체크박스과 라디오 버튼으로 구성된 있는 비디오 옵션, 사운드 볼륨 슬라이더 컨트롤 등이다.

이 예제용 애셋(asset)들은 설치된 Scaleform SDK 4.3 경로안의 Scaleform CLIK 의 다음 폴더에 존재한다.

Resources/AS2/CLIK/tutorial

Resources/AS2/CLIK/tutorial/art_assets

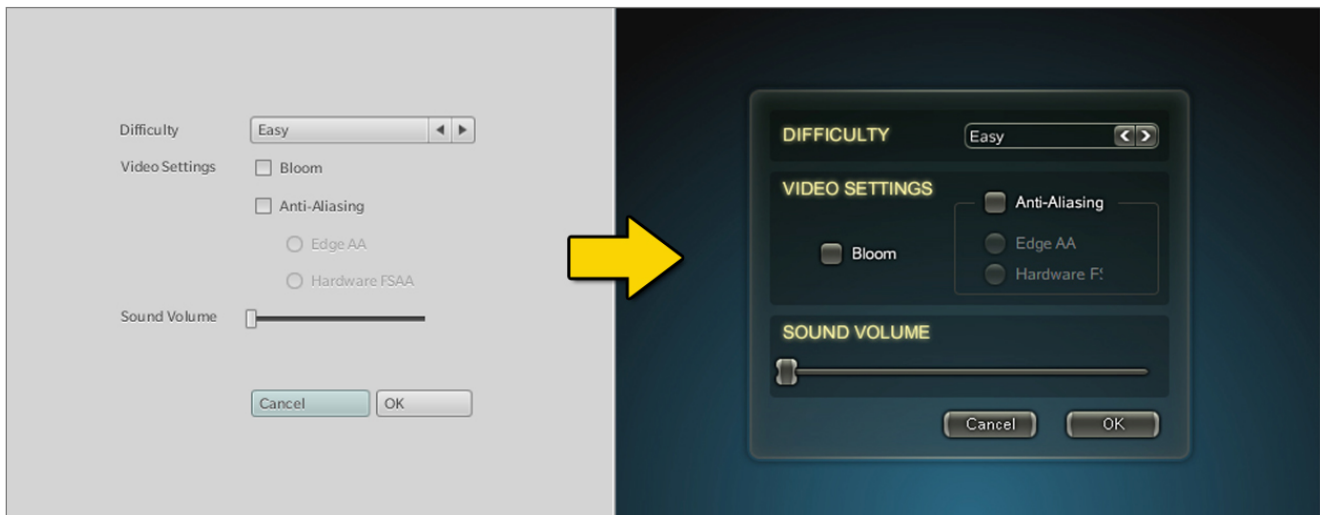


그림 5: 옵션 화면의 최종버전(스킨이 적용된 것과 적용되지 않은 것)

우리가 만드는 커스텀 사용자 인터페이스는 다음과 같은 워크플로우를 따른다.:

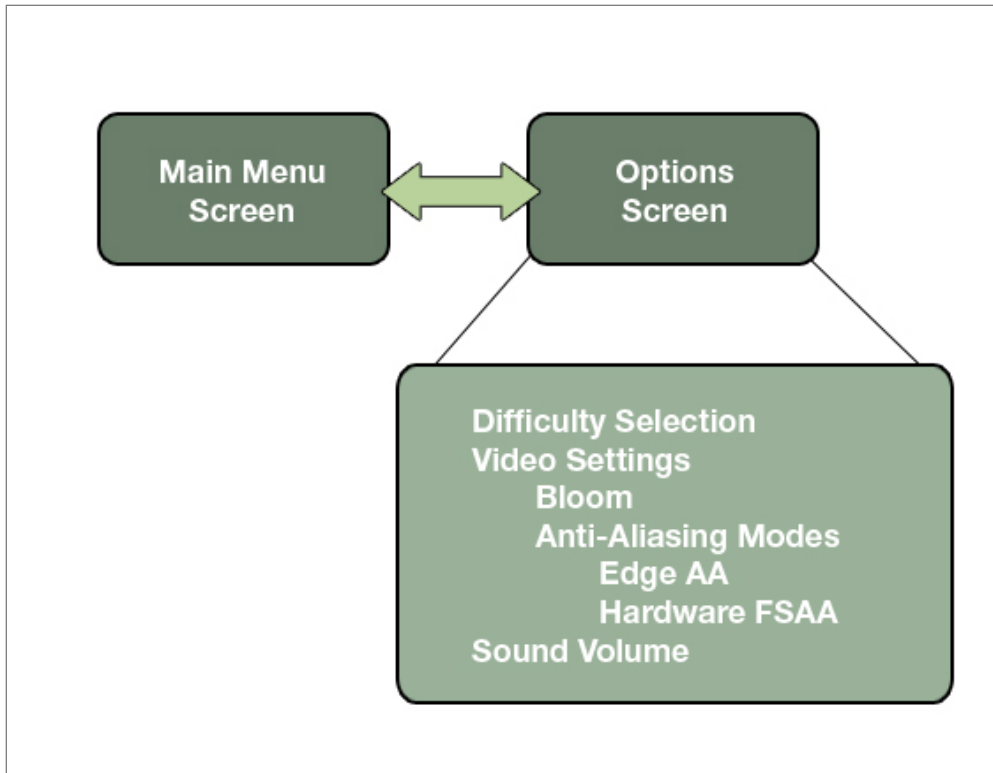


그림 6: 전면 메뉴 워크플로우

3.1 메뉴셀 만들기

먼저 할 일은 메뉴를 위한 다양한 레이어와 키프레임을 설정하는 것이다. 여기서는 대략적인 메뉴의 뼈대가 되는 것들을 설정해 보자.

1. 플래시 문서(액션스크립트 2.0)를 새로 생성한다.
2. 문서의 퍼블리시 세팅을 플레이어 9 로 한다. File 메뉴에서 'Publish Settings' 의 *Flash* 탭을 선택한다.
3. 버전 드롭다운 메뉴에서 'Flash Player 8'선택
4. 액션스크립트는 2.0 으로 선택
5. *Publish Settings* 창의 OK 클릭

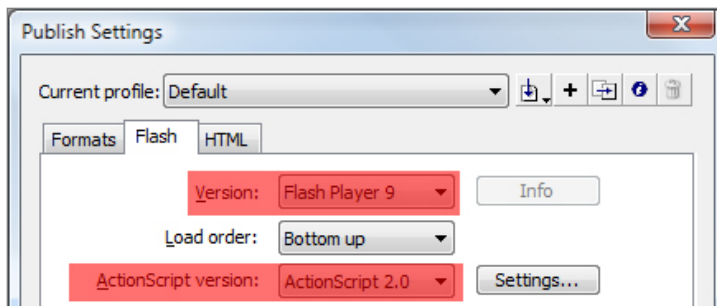


그림 7: Publish settings 창

6. 메인 문서 타임라인에서 현재 레이어 선택하고 Layer 1 으로 되어 있는 것을 더블 클릭해서 'scene'으로 변경. 이름이 중요한 것은 아니지만, 이런 식으로 레이어명을 짓는 것은 중요한 연습이 됨.
7. *Insert Layer* 버튼을 눌러서 새로운 레이어를 생성하고 이름을 'actions'으로 변경. 이번에도 이름이 중요한 것은 아니지만 각 레이어의 용도를 이름으로 기억할 수 있기 편리하다.

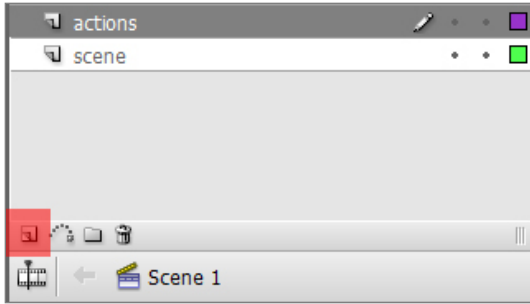


그림 8: layer 추가 버튼

8. *actions* 과 *scene* 레이어의 10 번째 프레임에 두번째 키프레임을 추가한다. *scene* 레이어를 선택하고 타임라인의 10 번째 프레임에서 오른쪽 마우스를 클릭한다. 팝업메뉴에서 'Insert Keyframe'을 선택한다. 이 키프레임이 메뉴에서 옵션메뉴를 생성하는 용도로 사용될 것이다. 이와 같은 과정을 *actions* 레이어에서 똑 같이 반복한다.

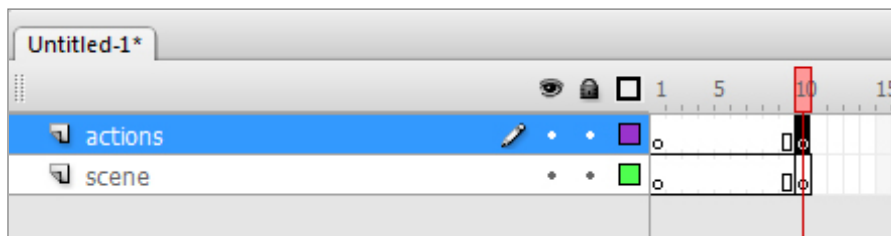


그림 9: 10 프레임에 키프레임이 추가된 메인메뉴

9. *scene* 레이어의 1 번 키프레임을 'mainMenu'로 레이블 명을 정한다. 키프레임을 선택하고 *Properties* 탭에서 이름을 넣으면 된다. 이 레이블은 Scaleform 코드에서 *mainMenu* 프레임으로 "goto"하는 용도로 사용될 것이다. 1 번 프레임은 메뉴의 메인 화면을 생성하는데 사용될 것이다.

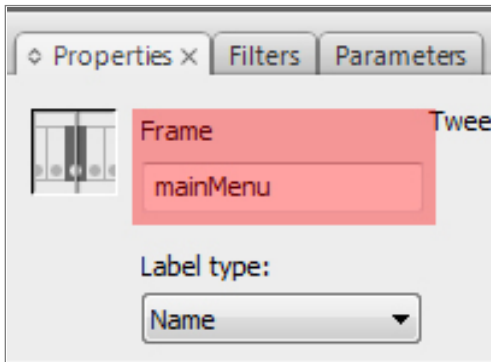


그림 10: 키프레임 1 은 'mainMenu'레이블 명을 갖는다.

10. scene 레이어의 1 번 키프레임을 'optionsScreen'으로 레이블 명을 정한다. 이 레이블은 Scaleform 코드에서 *optionsScreen* 프레임으로 "goto"하는 용도로 사용될 것이다.
11. *actions* 레이어의 1 번 프레임을 선택한다. Action 창을 열어서(F9) "stop();"코드를 따옴표 없이 첫줄에 입력한다 이 코드는 플레이어가 지정된 프레임에서 멈추게 해준다. 이 코드가 없으면 플레이어가 stop 명령을 만나거나 타임라인의 끝에 도달할 때까지 계속 플레이를 하게 된다. 이 경우에는 10 번 프레임이 될 것이다. *actions* 레이어의 10 번 프레임을 선택해서 동일한 코드를 추가한다.
12. 이쯤에서 플래시 파일을 저장해 두는 것이 좋겠다.

3.2 메인 메뉴 네비게이션 컨트롤 추가

여기서는 메인메뉴에 옵션버튼을 추가하는 방법에 대해서 설명할 것이다. 그래야 사용자가 옵션화면을 볼 수 있을 테니까 말이다.

1. 설치된 Scaleform 4 SDK 경로의 *Resources/AS2/CLIK/components* 안에 위치한 *CLIK_Components.fla* 파일을 연다.
2. 먼저 메뉴 제목을 추가한다. *CLIK_Components.fla* 의 라이브러리 패널의 label 컴포넌트에서 오른쪽 클릭을 하고 팝업메뉴의 'Copy'를 선택한다.

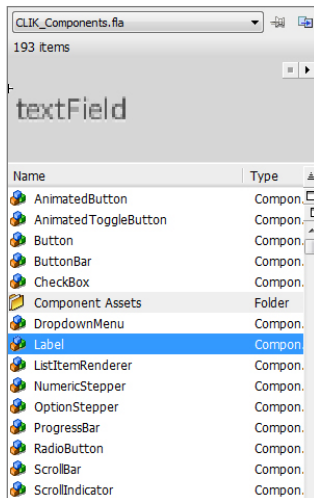


그림 11: 라이브러리 패널과 레이블 컴포넌트

3. 앞으로도 계속 필요할 것이므로 *CLIK_Components.fla* 을 아직 닫지 말고, 예제 메뉴를 생성하기 위해 원본 플래시 문서로 돌아오자.
4. *scene* 레이어의 첫 번째 키 프레임을 선택한다
5. 라이브러리 패널의 빈 영역에서 오른쪽 버튼을 클릭해서 *Paste* 를 선택하면 복사한 *Label* 컴포넌트를 붙여 넣을 수 있다.
6. 라이브러리 창에서 레이블 컴포넌트를 끌어다가 스테이지의 상단 가운데쯤에 가져다 놓는다.

7. 새로 생성한 label 의 *Parameters* 탭을 열어서 text 항목에 'Main Menu'를 입력한다.
이 항목은 플레이어에서 실행될 때 무엇이 출력될지를 나타내는 것이다. **주의:** CS3에서는 *Properties* 와 *Filters* 탭 옆에 있지만, CS4에서는 *Component Inspector* 안에 있다. 게다가 기본적으로 비활성화 되어 있다. [Shift+F7]이나 상단메뉴에서 Windows-Component 를 선택하면 나타난다.

주의: 이들 인자값들은 SWF 파일이 재생되면 출력될 것이다.

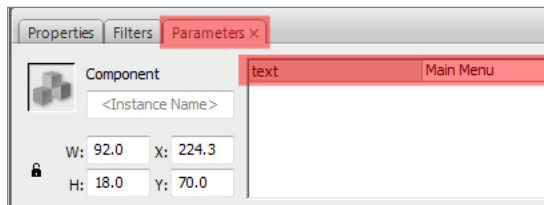


그림 12: CS3에서는 parameters 탭이 properties 탭 옆에 존재한다.

8. 10 번 키프레임을 선택하고 6~7 과정을 반복해서 옵션화면 제목을 생성한다. 당연히 text 항목에는 'Options'를 입력해야 한다.
9. *CLIK_Components.fla* 파일을 열어서 *Button* 컴포넌트에서 오른쪽 버튼을 클릭하고 Copy 를 선택한다.

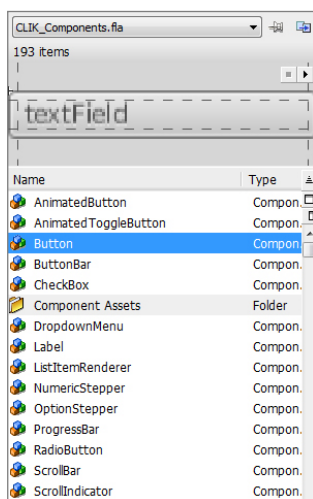


그림 13: 라이브러리 창의 Button 컴포넌트

10. 원본 플래시 문서를 선택한다.

11. 라이브러리 패널의 빈 영역에서 오른쪽 버튼을 클릭하여 *Paste* 를 선택하면 복사한 Button 컴포넌트를 붙여 넣을 수 있다.
12. *scene* 레이어의 첫 번째 키프레임을 선택한다
13. 라이브러리 창에서 *Button* 컴포넌트를 끌어다가 문서에 가져다 놓는다. 이때 위치는 *Main Menu* 레이블 밑에 놓도록 하자.
14. New button 을 선택해서 인스턴스 명을 'optionsBtn'으로 바꾼다(*Properties* 탭에서 바꿀 수 있다) 이 인스턴스명은 나중에 버튼이 눌렸을 때 어떤 행동을 할지 코드에서 사용될 것이다.

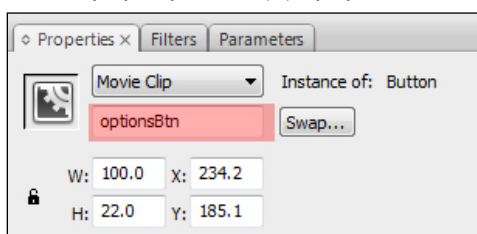


그림 14: 'optionsBtn'으로 명명된 버튼 인스턴스

15. *Parameter* 탭은 선택해서 label 항목에 'Options' 라 입력한다. 나중에 실행될 때 버튼이 어떤 이름으로 출력될지를 결정하는 것이다.

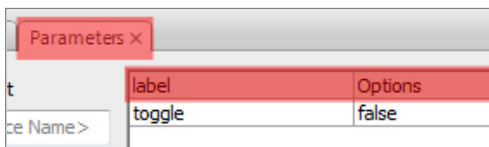


그림 15: 'Options'라 명명한다.

16. 축하한다! Scaleform 런처패널을 사용해서 테스트를 해보자. Scaleform 런처패널에 있는 *Test with [실행파일 명]* 버튼을 사용하면 Scaleform 플레이어를 사용해서 무비를 재생할 수 있다.

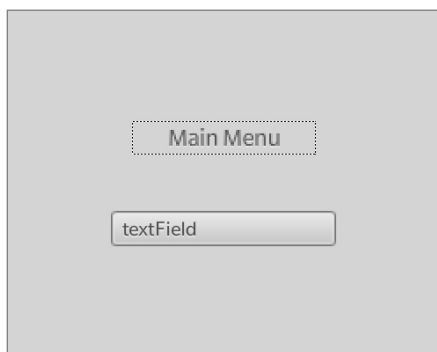


그림 16: Scaleform 플레이어로 돌려본 메인메뉴(옵션화면 버튼 포함)

3.3 옵션화면 버튼 대기

가장 먼저 필요한 상호작용은 메뉴메뉴의 옵션버튼에 기능을 부여해서 사용자가 버튼을 누르면 옵션화면으로 들어가게 하는 것이다. 이번 예제에서는 스케일폼 CLIK의 이벤트 모델을 사용해서 이 기능을 구현해 보도록 하자. 다른 방법들도 존재하겠지만 스케일폼 CLIK의 이벤트 모델을 사용하면 CLIK 컴포넌트 사용시에 엄청난 유연성과 효율성을 얻을 수 있다. 이 모델에서는 특정 이벤트가 발생하면 옵션버튼이 콜백 함수를 호출하도록 한다. 콜백함수는 메인 타임라인을 옵션화면 키프레임으로 이동시키는 것이 역할이다. 이때 이벤트는 사용자가 마우스의 왼쪽 버튼을 눌렀을 때 발생하는 마우스 클릭이다.

1. *actions* 레이어의 첫번째 키프레임 선택

2. "`stop();`" 명령어 위에 다음 코드 삽입. 이 함수는 10번째 프레임에 있는 옵션메뉴로 사용자를 이동시켜줌.

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

3. `optionsBtn` 인스턴스가 클릭 이벤트가 발생했을 때 위에 있는 `openOptionsScreen` 함수를 호출할 수 있도록 다음 코드 삽입

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

4. 마지막으로 최초 포커스를 옵션버튼에 두도록 다음 코드를 마지막에 삽입

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

5. 무비 테스트. Options 버튼이 기본적으로 선택되어 있어야 함. Options 버튼을 누르면 옵션화면으로 이동(물론 현재는 비어있음)

3.4 난이도 옵션 스텝퍼 추가

게임 난이도 설정 컨트롤은 옵션 스텝퍼를 사용해서 옵션메뉴에 추가될 것이다. 이 컴포넌트는 동적으로 채워진 데이터를 출력해 주는 컨트롤인데 여러 가지 리스트 중에서 선택하는 용도로 사용한다(여기서는 Easy, Medium, Hard, Insane 에서 선택). 옵션 스텝퍼는 리스트에 있는 옵션들을 선택할 때 2 개의 버튼 컴포넌트도 함께 사용한다.

1. CLIK_Components.fla 파일로 간다.
2. 라이브러리 창의 *OptionsStepper* 에서 우측버튼으로 *Copy* 선택
3. 원본 플래시 문서로 간다.
4. *scene* 레이어에서 10 번 키프레임 선택 (*Options Menu* 키프레임).
5. 라이브러리 창에 *OptionsStepper* 붙여 넣기

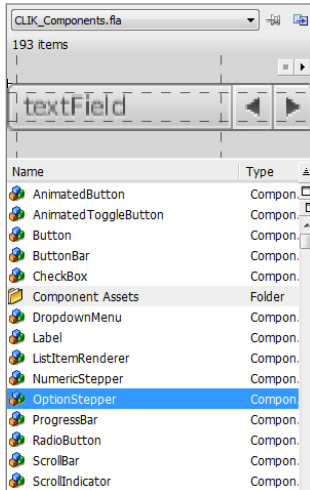


그림 17: 라이브러리 창의 OptionStepper 컴포넌트

6. 이미 OptionsStepper 가 존재할 경우에는 이 과정 생략. 라이브러리 창의 OptionsStepper 를 스테이지로 끌어오기
7. OptionsStepper 를 선택해서 인스턴스 명을 'difficultyOption'으로 바꿈

8. 라이브러리 창의 Label 컴포넌트를 문서로 끌어와서 *Difficulty* 컨트롤 옆에 위치시키기 난이도 설정 컨트롤에 대한 설명을 하는 레이블이 될 것임.

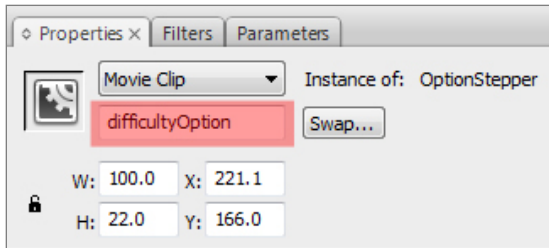


그림 18: 난이도 컨트롤 인스턴스를 'difficultyOption'으로 바꾸기

9. *Label*의 *Parameters* 탭에서 이름을 'Difficulty'로 바꿈.



그림 19: 스킨 없는 난이도 옵션 스텝퍼 최종화면

3.5 비디오 세팅 체크박스과 라디오 버튼 추가

비디오 세팅 컨트롤은 On 과 Off 라는 2 가지 상태를 가진 체크박스를 사용할 것이다(여기서는 Bloom 과 Anti-Aliasing). 또한 게임에서 안티앨리어싱이 어떻게 작동할지를 결정할 2 개의 라디오버튼 그룹도 사용할 것이다. 일반적으로 체크박스는 원하는 기능을 활성화 할 것인지, 비활성화 할 것인지를 결정할 때 사용하고, 라디오 버튼은 여러 개의 옵션리스트 중에서 하나를 선택할 때 사용한다.

1. *Difficulty* 컨트롤 아래에 *Label* 컴포넌트를 새로 만들고, 텍스트 파라미터를 'Video Settings'로 설정한다.
2. *CLIK_Components.fla* 파일 선택
3. 라이브러리 창의 *CheckBox* 에서 우측버튼으로 *Copy* 선택
4. 플래시 문서로 돌아가서 *CheckBox* 를 라이브러리 창에 붙여넣기
5. *CheckBox* 컴포넌트를 라이브러리 창에서 스테이지로 끌어놓기

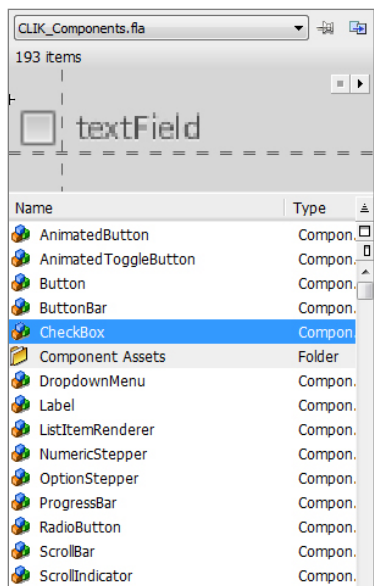


그림 20: 라이브러리 창의 체크박스 컴포넌트

6. new instance 를 선택하고 인스턴스 명을 'bloomBtn'으로 바꾸기

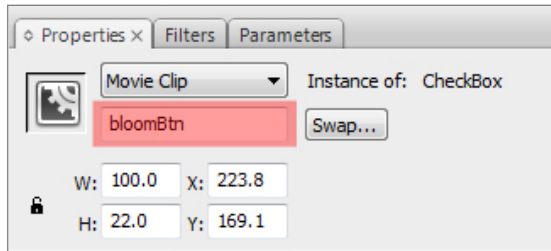


그림 21: bloom 라ियो 버튼 인스턴스를 'bloomBtn'으로 명명

7. *Parameters* 밑의 *label* 필드를 'Bloom'으로 변경
8. 인스턴스 복사. 복사본은 Bloom 체크박스 바로 밑에 놓기. 새 인스턴스 명은 'aaBtn'
9. *Parameters* 밑의 *label* 필드를 'Anti-Aliasing'으로 변경
10. RadioButton.fla 파일 열기
11. 작업 라이브러리에 RadioButton 컴포넌트 복사 & 붙여넣기
12. RadioButton 의 새 인스턴스를 스테이지에 있는 Anti-Aliasing checkbox 밑에 놓기
13. RadioButton 의 인스턴스 명을 'edgeaaBtn'으로 변경

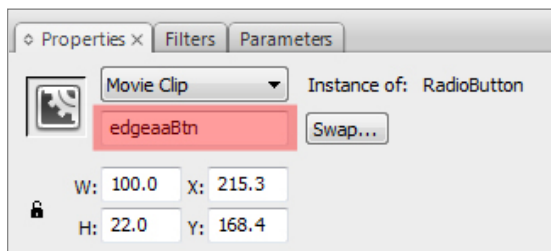
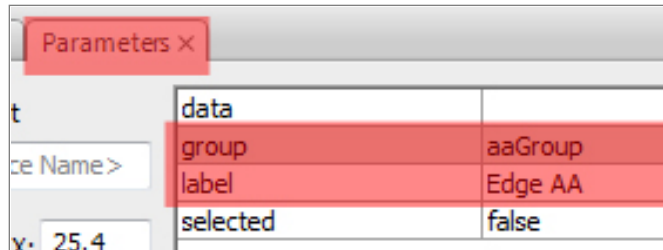


그림 22: Edge AA 라디오버튼 인스턴스를 'edgeaaBtn'으로 명명

14. *Parameters* 밑의 *label* 필드를 'Edge AA'로 변경
15. 그룹 필드를 'aaGroup'으로 변경
16. 인스턴스 복사. 복사본은 Edge AA 라디오 버튼 바로 밑에 놓기. 새 인스턴스 명은 'hwaaBtn'

17. *Parameters* 밑의 *label* 필드를 'Hardware FSAA'로 변경.

18. 그룹 필드를 'aaGroup'으로 변경



Parameters ×	
data	
group	aaGroup
label	Edge AA
selected	false

그림 23: 라디오버튼 그룹을 'aaGroup'으로 설정



그림 24: 스킨 없이 비디오 설정이 추가된 최종화면

3.6 사운드 볼륨 슬라이더 추가

사운드 볼륨은 슬라이더 컴포넌트를 사용해서 추가할 것이다. 슬라이더는 트랙(track)과 썸(thumb) 컨트롤을 사용한다. 이들 모두 간단한 버튼 컴포넌트다. 썸은 슬라이더의 길이를 제한하기 때문에 사용자는 슬라이더의 왼쪽이나 오른쪽 바깥으로 넘어갈 수 없다. 물론 수직으로 움직이는 것도 제한되기 때문에, 상하로 움직이는 것도 불가능하다.

1. 스테이지에 있는 *Video Settings Controls* 밑에 *Label* 컴포넌트를 생성하거나 복사한다. 그리고 text 파라미터를 'Sound Volume'으로 설정한다.
2. CLIK_Components.fla 파일을 선택하고, *Slider* 컴포넌트를 복사해서 작업중인 플래시 문서에 붙여넣는다.

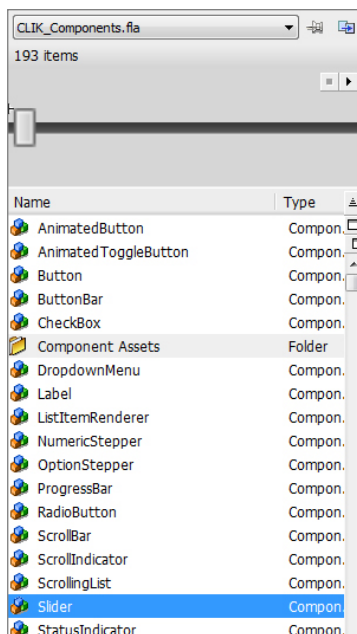


그림 25: 라이브러리 창의 슬라이더 컴포넌트

3. *Slider* 인스턴스를 끌어서 스테이지의 *Sound Volume* 레이블 바로 옆에 놓는다. 인스턴스명은 'soundSlider'로 한다.

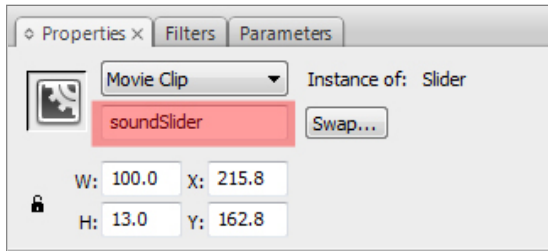


그림 26: 볼륨 슬라이더 인스턴스를 'soundSlider'로 명명

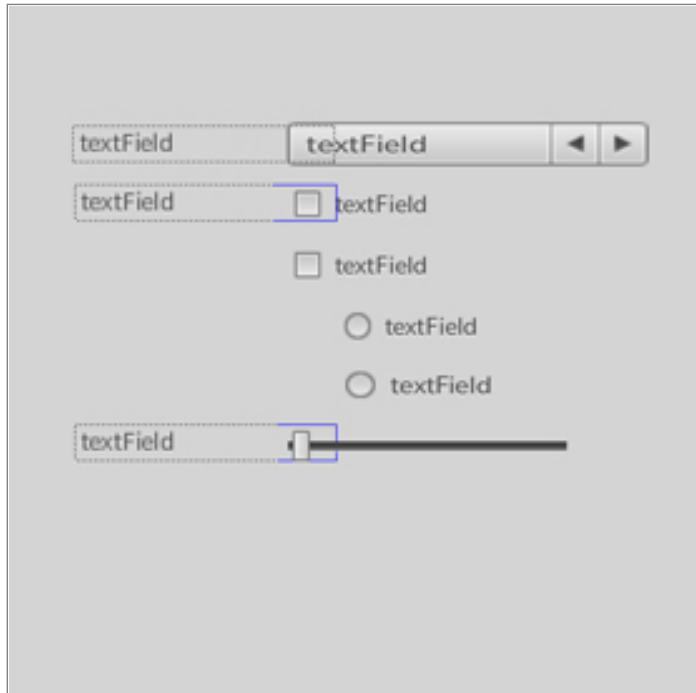


그림 27: 스킨 없이 볼륨 슬라이더가 추가된 최종화면

3.7 OK 와 Cancel 버튼 추가하기

OK 와 Cancel 버튼은 기반 버튼 컴포넌트를 사용한다.

1. 라이브러리에서 버튼 컴포넌트를 스테이지로 끌어온다
2. 버튼 선택
3. 버튼의 *Parameter* 탭을 선택해서 *label*을 'OK'로 설정
4. 인스턴스 명을 'okBtn'으로 변경
5. 스테이지에 있는 버튼을 복사하고 새 인스턴스의 이름은 'cancelBtn'으로 변경
6. 버튼의 *Parameters* 탭을 선택해서 *label*을 'Cancel'로 설정.

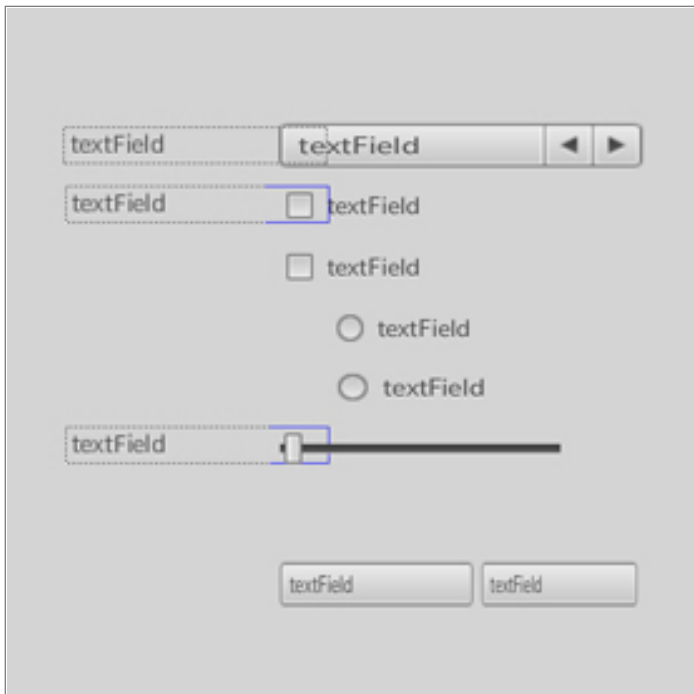


그림 28: 스킨 없는 최종 옵션화면

4. 액션스크립트로 기능 추가하기

이제 기본 스킨으로 무장된 모든 컴포넌트 들이 만들어졌다. 하지만 메뉴로서의 기능은 아직 전무하다. 드디어 액션스크립트를 추가할 시간이다. F9 버튼을 눌러서 액션 패널을 불러오자.

만약 예제코드가 작동하지 않을 경우에는 예제 파일을 열어서 코드를 살펴보도록 하자. 분명히 잘못 입력된 코드가 있을 것이다.

4.1 난이도 옵션스텝퍼에 난이도 레벨 추가하기

난이도 스텝퍼에 사용자가 어떤 값을 선택할 수 있는지를 설정하려면 데이터를 적용해야 한다. `difficultyOption` 컴포넌트의 `'dataProvider'` 속성에 값을 넘기면 된다. 여기서 우리는 Easy, Medium, Hard, Insane 으로 설정할 것이다.

1. `actions` 레이어의 10 번째 프레임 선택
2. 액션패널 안쪽의 아무 곳이나 클릭해서 코드삽입 모드로 전환
3. `"stop();" 명령어 위쪽에 다음 코드 삽입`

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

4. 무비 테스트. 화살표 버튼을 사용해서 게임 난이도 설정 옵션 스텝퍼 컨트롤에 값이 잘 들어가 있는지 확인.

4.2 aaRadio 버튼 그룹 설정하기

안티앨리어싱 라디오 버튼은 코드상에서 그룹화 되어야 한다. 안티앨리어싱 라디오버튼은 사용자가 안티앨리어싱 체크박스가 체크되기 전까지는 비활성화(상태변경 불가) 되어 있어야 한다.

1. 액션 레이어 10번째 프레임의 키프레임을 선택한다.
2. 첫째, CLIK ButtonGroup 클래스에 접근 할 수 있도록 액션 패널의 제일 상단에 다음의 코드를 삽입한다.

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

3. 그 다음 difficultyOption.dataProvider 코드의 라인 아래로 이동해서 새로운 버튼 그룹을 'aaGroup' 이라는 이름으로 생성하고 edgeaaBtn 과 hwaaBtn의 group 속성에 할당한다.

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

4. 라디오 버튼들을 사용가능 또는 사용불가능 상태로 만들 수 있는 함수를 추가한다.

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

5. 마지막으로, 그 함수를 불러낼 수 있는 Anti-Aliasing 체크박스 이벤트 리스너를 추가한다.

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

6. 무비를 테스트한다. Test the movie. 기본적으로 비디오 셋팅과 안티앨리어싱 라디오 버튼들은 현재 disable 상태이지만 안티앨리어싱 체크박스를 체크하면 enable 될 것이다.

4.3 옵션의 초기상태 설정하기

옵션은 사용자가 OK 버튼을 눌렀을 때 메인 메뉴로 돌아가고 그들의 상태를 계속 유지해야한다. 그리고 그들은 *Cancel* 버튼을 눌렀을때는 바뀐 모든 상태들은 무시되어야 한다.

1. 타임라인의 상단에(action 레이어의 위) 새로운 레이어를 생성하고 그 이름을 'vars'라 한다.
2. var 레이어의 첫번째 프레임을 선택한다.
3. 옵션의 상태를 저장할 새로운 객체를 'options'라는 이름으로 생성한다. 새로운 오브젝트가 이미 생성되었는지 검사하는 조건문을 삽입한다.

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

4. actions 레이어의 10프레임을 선택한다.
5. difficultyOption 의 초기화 상태를 설정한다. 만약 options.selectedDifficulty 속성이 아직 정의되지 않았다면 difficultyOption의 초기값을 0으로 넣어주고(0은 "Easy"에 해당함), 그렇지 않다면 options.selectedDifficulty 의 값을 넣어준다.

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

6. bloomBtn 과 aaBtn의 상태를 초기화 한다.

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

7. edgeaaBtn 과 hwaaBtn을 aaBtn 체크박스의 선택된 상태대로 disabled 인지 enable인지 설정해준다.

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

8. edgeaaBtn을 초기값을 넣는다. 만약 options.edgeaa의 속성이 아직 정의되지 않았다면, edgeaaBtn의 상태를 true 로 선택하고 그렇지 않으면 options.edgeaa의 값을 넣어준다.

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

9. 마찬가지로 hwaaBtn의 상태에 대한 초기값을 넣는다.

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

10. soundSlider의 초기값을 넣는다.

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

4.4 옵션 화면 종료

약간의 코드를 추가해서 *Cancel* 버튼이나 *OK* 버튼이 눌렸을 때 옵션화면을 종료하고 메인메뉴로 돌아가도록 한다.

1. actions 레이어의 10번째 프레임을 선택한다.
2. OK 버튼이 눌렸을 때 사용자를 첫번째 프레임(메인메뉴)으로 보내는 함수를 4.3에서 입력한 코드와 "stop();" 사이에 다음 함수를 삽입한다. 이 함수는 사용자가 옵션 화면으로 돌아갈 때 options 객체의 속성에 값을 넣준다. (그리고 저장한다.) 그것들은 다음과 같다.

```
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. Cancel 버튼이 눌렸을 때 사용자를 1 프레임으로 돌아오게 하는 함수를 삽입한다. 이 함수는 바뀌어진 어떠한 것도 저장하지 않는다.

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

4. 사용자가 OK 나 Cancel 버튼을 누르는지 체크하는 이벤트리스너를 추가한다.

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

5. 마지막으로, 기본 포커스는 Cancel버튼에 있도록 설정한다.

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

6. 무비 테스트, 모든 옵션화면 컨트롤들이 정상작동 하는지 체크한다.

5. 메뉴에 스킨 입히기

마지막으로 할 일은 메뉴와 UI 컴포넌트에 아티스트가 만든 스킨을 적용하는 것이다. 모든 컴포넌트에 스킨을 적용하지는 않을 것이고, 우리가 사용한 컴포넌트들에 대해서만 설명할 예정이다. 일단 작업공정을 이해하면 나머지 컴포넌트들에도 똑 같이 적용할 수 있다.

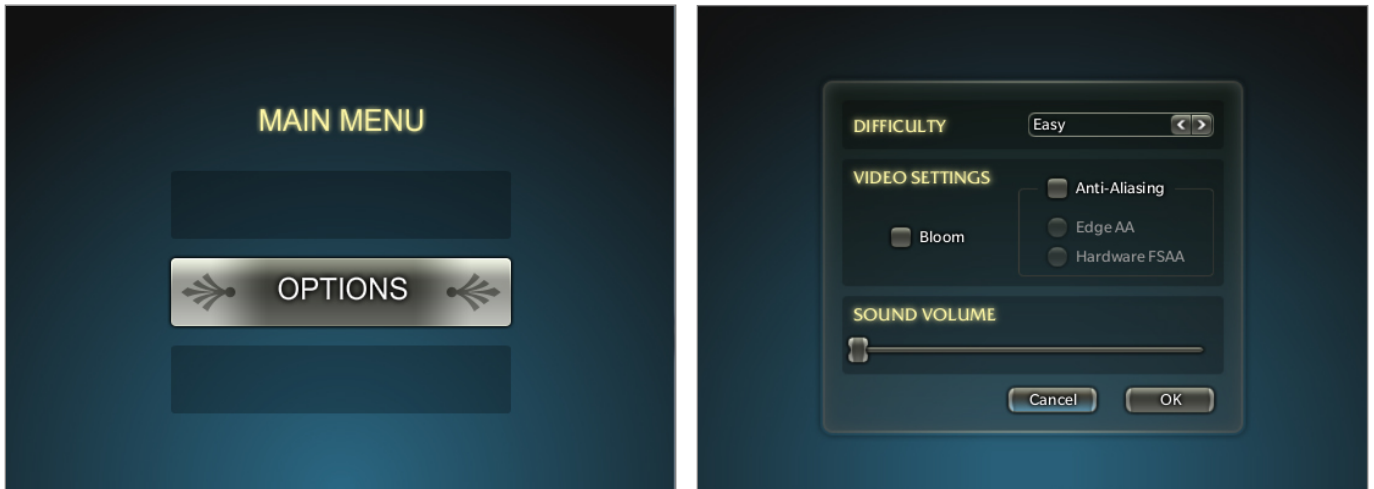


그림 29: 스킨 적용된 최종 메뉴

5.1 옵션버튼에 스킨입히기

여기서 설명하는 작업흐름은 포토샵®, 일러스트레이터® 와 플래시 CS3 를 기반으로 하고 있지만, 플래시 CS4 에서도 똑같이 적용 가능하다.

UI 에 있는 모든 오브젝트는 다양한 상태를 갖는다. 버튼을 예로 든다면 Up(기본), Over(선택되거나 마우스가 위에 있을 때), Down(눌렸을 때) 등이다. 이들 상태에 따라서 각기 다른 이미지의 비주얼 피드백을 사용자에게 제공해야 한다. 이들 상태들은 포토샵이나 일러스트레이터에서 만들 수 있다. 이처럼 상태를 나타내는 이미지 데이터들은 하나의 포토샵 혹은 일러스트레이터 파일에 들어가 있는게 좋다. 각각의 상태 별로 분리된 레이어에 데이터를 넣어두는 것이다. 이렇게 하면 플래시가 타임라인의 키프레임으로 합칠 수 있다. 이미지를 가져오기(import)전에 관련된 레이어들을 각 상태 별로 하나의 레이어에 합친다. 작업파일에는 원본 레이어를 유지한 채로 분리해서 작업함으로써 나중에 수정을 하기 쉽도록 해야 한다. 간단한 버튼의 경우에는 최종 이미지 파일에는 3 개의 레이어만 있으면 된다. (Up, Over, Down)

1. 버튼의 각 상태 이미지를 동일한 포토샵 문서에 만든다(PSD 파일).
 - a. 각 컴포넌트 별 상태리스트는 문서를 참고하거나 타임라인에서 컴포넌트의 *labels* 레이어를 참고하자.
 - b. **중요:** 똑같은 내용이 담긴 복제 레이어를 생성하지 말라. 우리가 여기서 필요한 것은 오직 Up, Over, Down 레이어 뿐이다.
2. 이미지 작업이 완료되었으면 PSD 파일을 저장한다. 하나의 상태에 여러 개의 레이어가 만들어졌다면, 동일한 상태는 하나의 레이어로 합치기 바란다. 레이어를 합치려면 CTRL 키를 누른채로 포토샵의 레이어 패널에서 레이어들을 선택하고 메뉴에서 레이어 합치기(*Merge Layers*)를 선택하면 된다.
 - a. 즉, over 상태를 나타내는 모든 레이어는 하나의 over 레이어로 합쳐야 한다.
3. 플래시 컴포넌트 타임라인의 첫번째 상태를 나타내도록 레이어를 정렬해야 하는데(일반적으로 UP 이 기본 상태) 포토샵 레이어 패널에서 가장 아래쪽에

위치시키도록 한다. 그리고 마지막 상태를 가장 위쪽에 위치시킨다. 이 순서는 플래시가 레이어를 가져올 때 매우 중요하다.

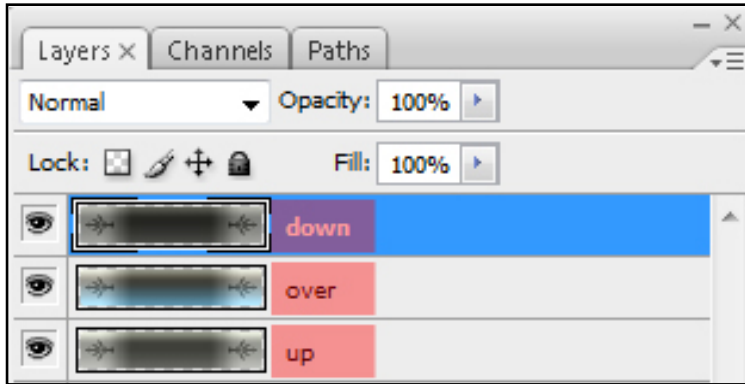


그림 30: 포토샵의 옵션버튼 레이어

4. (선택사항) 플래시에서 이름이 매칭될 수 있도록 각 레이어의 이름을 up, over, down 으로 바꾼다. 이것은 순전히 조직상의 목적이다.
5. 새로운 파일명으로 PSD 를 저장한다. 이때 파일명 끝에 '_mergedLayers'라고 추가한다. 기존 파일에 덮어쓰는 실수를 하지 않기 바란다.
6. 플래시에서 스킨적용이 안된 *Options* 버튼을 더블클릭해서 타임라인이 나타나게 한다.
7. 타임라인에 있는 *button* 레이어를 선택해서 모든 프레임을 지운다. 모든 프레임을 선택하려면 레이어 이름을 누르면 된다. 그리고 나서, 타임라인 프레임에서 오른쪽 버튼을 클릭하고 *Remove Frames* 를 선택한다.
8. 레이어에 빈 키프레임을 하나 추가해서 선택해둔다(step 9 에서 필요함)
9. 상단메뉴에서 [File→Import→Import to Stage]를 선택하거나 그냥 CTRL+R 을 누른다.
 - a. 만약 'Import to Stage'메뉴를 사용할 수 없다면, 레이어가 잠겨(locked)있는지 확인하기 바란다. 레이어 이름의 오른쪽에 잠금 아이콘이 보인다면 아이콘을 클릭해서 잠금해제(unlock)하기 바란다.
10. 스킨 UI 에 사용될 PSD 나 일러스트레이터(AI)파일을 찾는다(앞서 저장한 PSD 파일)

11. 가져오기 대화상자가 나타날 것이다. 모든 레이어가 체크되어 있는지 확인한다.
 - a. **주의:** Flash8 에서는 PSD 파일 가져오기 대화상자가 나타나지 않는다. CS3 와 CS4 에서는 잘 될 것이다. 플래시 8 을 사용할 때는 이 순서로 작업하면 안될 것이다.

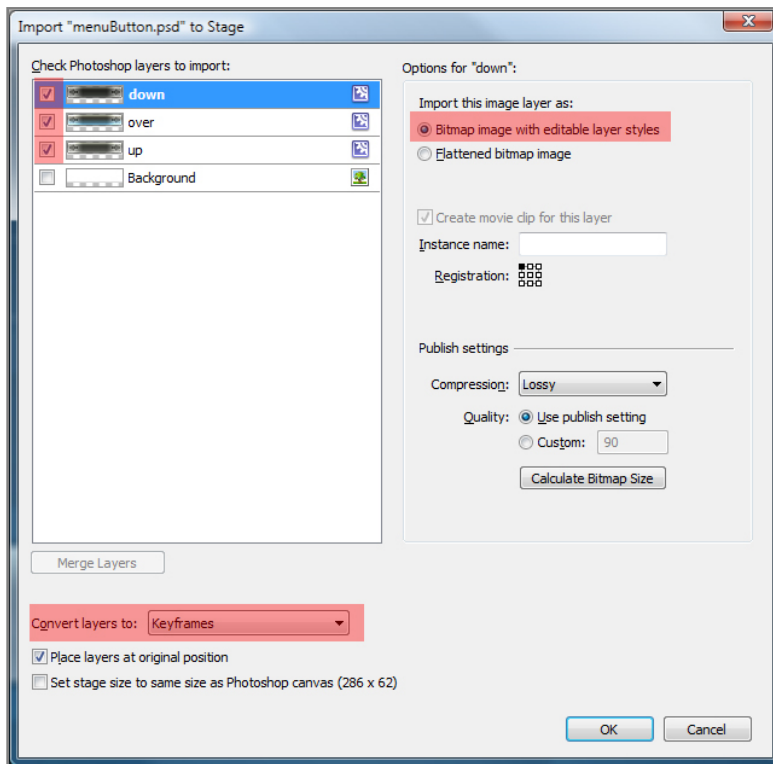


그림 31: 스테이지로 가져오기(The import to stage) 대화상자

12. *Convert layers to:* 항목을 'Keyframes'으로 선택한다. 이 항목은 대화상자 좌측 하단에 있다.
13. 리스트에 있는 각 레이어를 클릭하고 위쪽 *Import this image layer as*에 있는 *Bitmap image with editable layer styles*을 선택한다.
 - a. **중요:** 각 레이어 별로 이 과정을 처리하지 않으면 레이어 스타일이 적용된 레이어는 스테이지에 나타나지 않게 된다(비록 라이브러리에는 추가되겠지만 말이다). 게다가 첫 번째 레이어를 제외하고는 모든 레이어가 병합(flatten)되어 버린다.

14. 레이어가 타임라인의 각 키프레임 형태로 들어갈 것이다. 이 순서는 첫번째 상태에서 마지막 상태 순서로 들어간다.
 - a. **중요:** PSD 파일을 가져올 때는 타임라인의 최초부분에 새로운 레이어로 키프레임이 추가될 것이다. 예전 버튼 레이어를 삭제하고, 새로 생성된 레이어의 이름을 바꾸고, 새로운 버튼 레이어를 레이어 리스트의 제일 밑으로 이동시킨다.
 - b. AI 파일을 가져올 때는 선택된 키프레임이 레이어에 추가될 것이다.
15. 각 키프레임들을 클릭해서 끌어다가 적절한 레이블에 정렬한다(타임라인의 labels 레이어에 출력된다). 첫 번째 키프레임을 첫번째 프레임(up)으로 이동시킨다. 두 번째 키프레임을 over 키프레임 바로 밑으로 이동시킨다. 세번째와 마지막 키프레임을 down 키프레임 밑으로 이동시킨다. Disabled 는 이 문서에서는 설명하지 않는다.

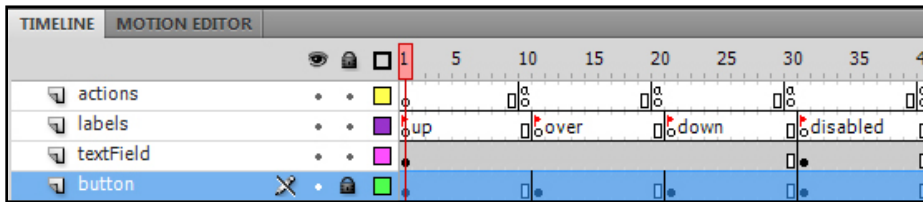


그림 32: 버튼 레이어에 키프레임 추가

16. 사용자가 버튼에 어떤 조작을 할 때마다 버튼에서 다른 레이블을 보여주려면 텍스트 레이어(up, over, down)의 각 키프레임별로 textField 인스턴스의 폰트설정을 수정한다. 그렇지 않다면 모든 상태에 대하여 같은 값을 갖도록 그냥 놔두면 된다.
17. 무비 테스트. 옵션버튼에 스킨이 적용되었을 것이다.



그림 33: 스킨이 적용된 옵션버튼과 메인메뉴 최종화면

5.2 볼륨 슬라이더에 스킨적용

볼륨 슬라이더는 2 개의 버튼으로 이루어져있다. 한 버튼은 슬라이더의 썸(슬라이더 좌우로 이동하는 놈)이고, 다른 하나는 썸이 이동할 트랙을 나타내는 버튼이다. 이번에 우리가 하려는 작업은 각각의 버튼(썸과 트랙)을 만드는 것인데, 앞서 설명한 것과 큰 차이가 없다.

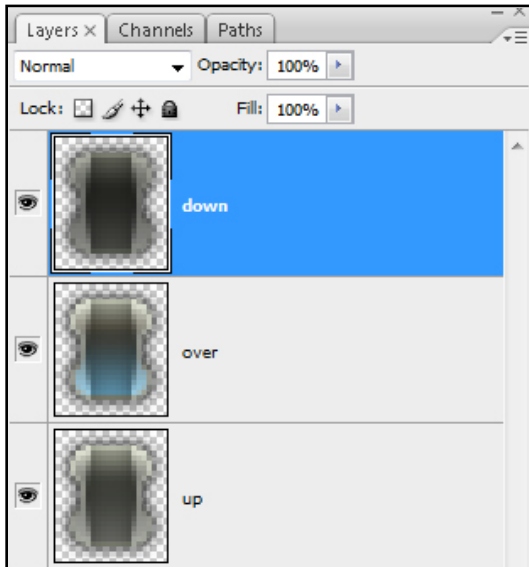


그림 34: 볼륨 슬라이더 썸 스킨의 포토샵 레이어

1. 스테이지에 있는 볼륨 슬라이더를 더블클릭해서 타임라인으로 들어간다.
2. 슬라이더의 *Track* 컴포넌트를 더블클릭해서 타임라인으로 들어간다.
3. Track 용으로 제작된 PSD 나 AI 파일을 가져와서 키프레임을 정렬한다. Track 의 등록지점(registration point)은 그래픽의 가장 좌측이어야 하며, 슬라이더 컴포넌트와 정확히 매치되어야 한다.
4. 슬라이더 타임라인으로 돌아간다(1 레벨 위로 이동)
5. 썸 컴포넌트를 더블클릭해서 타임라인으로 들어간다.
6. 썸 PSD 나 AI 파일을 가져온다.
 - a. **중요:** 썸 스킨을 가져올때는 등록지점이 가운데라는 것을 명심하자.

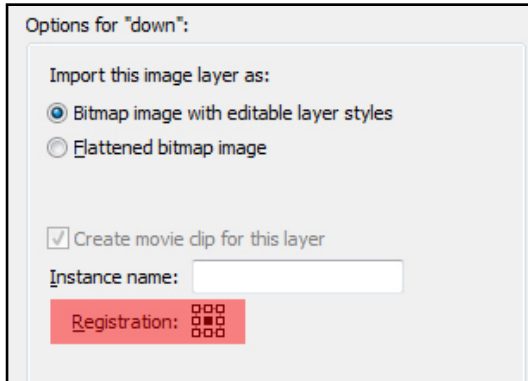


그림 35: 등록지점을 한 가운데로 설정

7. 각 키프레임의 씬 그래픽을 선택해서 *Property* 탭의 X 속성값을 0.0 으로 설정
8. 슬라이더 타임라인으로 돌아온다(1 레벨 위로 이동). 씬컴포넌트를 선택해서 바운딩 박스의 좌측상단이 슬라이더 컴포넌트의 등록지점과 일치하도록 맞춘다.

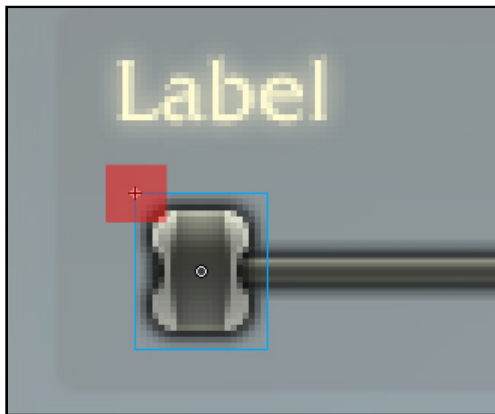


그림 36: 씬의 좌측상단을 슬라이더의 등록지점에 정렬

9. textField 레이블의 폰트 속성 수정
10. 무비 테스트



그림 37: 스킨 적용된 최종 슬라이더.

5.3 라디오버튼과 체크박스 스킨적용

라디오버튼과 체크박스는 다음과 같은 상태를 가진 버튼 컴포넌트다(selected_up, selected_over, selected_down, selected_out, selected_disabled) 이 컴포넌트에 대해서 어떻게 스킨을 적용하는지를 설명할 필요는 없을 것이다. 앞서 5.1 에서 설명한 것과 동일한 절차를 따르면 된다. 하지만, 새로 추가된 상태를 포함해야 한다는 것은 명심하도록 하자. 특히, 라디오버튼은 선택된 버튼을 나타내기 위해서 내부에 작은 점을 포함하고 있으며, 체크박스도 체크마크를 갖고 있다.

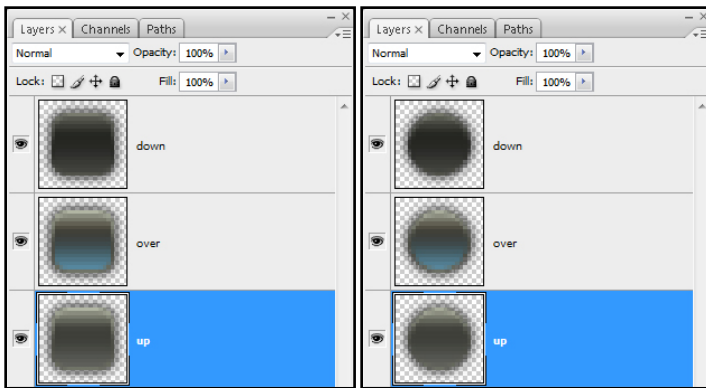


그림 38: 라디오버튼과 체크박스 스킨 포토샵 레이어

마지막으로 textField 레이블 인스턴스의 폰트속성을 수정하는 것을 잊지 말자.



그림 39: 스킨이 적용된 최종 edge AA 라디오 버튼

5.4 난이도 옵션 스텝퍼 스킨적용

난이도 스텝퍼는 2 개의 버튼 컴포넌트로 이루어져있다(좌측화살표, 우측화살표 버튼). 그리고 배경이나 경계면을 그리기 위한 그래픽도 필요할 수 있는데, 반드시 필요한 사항은 아니다.

1. 5.1 에서 설명한 방법으로 좌측, 우측 버튼을 만든다.
 - a. 하나의 버튼 이미지를 만들어서 플래시의 mirror 기능을 사용할 수도 있다.
2. 난이도 옵션 스텝퍼를 더블클릭해서 타임라인으로 들어간다
3. 디자인이 적절하면 바닥 배경과 경계면 이미지를 바꾼다. 밑에 있는 그림을 보면 회색 경계선을 사용하고 있다.
4. 좌측버튼을 더블클릭해서 타임라인으로 들어간다.
5. 앞서 5.1 에서 설명한 States 레이어에 PSD 파일을 가져와서 키프레임을 정렬한다.
6. 우측버튼도 4-5 과정을 반복한다.
7. 두 버튼의 부모 타임라인으로 돌아가서(난이도 옵션 스텝퍼의 타임라인) textField의 폰트속성을 수정한다.
8. 무비 테스트



그림 40: 스킨이 적용된 최종 난이도 스텝퍼

6. 결론



그림 41: 스킨이 적용된 최종 옵션메뉴

축하한다! 이 문서는 CLIK 사용에 있어서 한가지 길만을 제시했지만 실제로는 다양한 변종 수법들이 존재한다. 실제 프로젝트를 진행할 때는 각자의 독특한 방법들이 있을 것이다. 하지만 여기서 설명하는 기본적인 작업 흐름을 이해하는 것은 CLIK 프레임워크를 이용하는데 있어서 최소한으로 견비해야 하는 능력이다. 더 많은 학습이 필요하다면 더 많은 관련문서를 읽거나 CLIK 과 함께 제공되는 데모를 보고, 그 안에서 사용된 컴포넌트의 소스코드 들을 분석해보도록 하자.

데모와 소스코드는 Scaleform 4 SDK 가 설치된, 다음의 경로에 있음

Windows

- 데모 – *C:/Program Files/Scaleform/GFx SDK 4.3/Resources/AS2/CLIK/demos*
- 소스코드 – *C:/Program Files/Scaleform/GFx SDK 4.3/Resources/AS2/CLIK/gfx*

Mac

- 데모 – *scaleform_gfx_4.3_macos/Resources/AS2/CLIK/demos*
- 소스코드 – *scaleform_gfx_4.3_macos/Resources/AS2/CLIK/gfx*

추가 문서:

[CLIK 버튼 시작하기](#) – CLIK 의 버튼 컴포넌트 가이드
문서