

Autodesk® Scaleform®

Mobile Game Kit 개요

본 문서는 Scaleform 모바일 플레이어를 사용하여 모바일 및 태블릿 하드웨어용으로 설계된 범용 플랫폼의 터치/동작 기반 게임 데모인 Scaleform 4.2 Mobile Game Kit 의 구성, 소스코드 및 내용을 설명합니다..

작성자: Nate Mitchell

버전: 1.01

최종 수정일: 2012 년 6 월 5 일

저작권 안내

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.

How to Contact Autodesk Scaleform:

Document	Mobile Game Kit Overview
Address	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
Website	www.scaleform.com
Email	info@scaleform.com
Direct	(301) 446-3200
Fax	(301) 446-3199

차례

1	소개	1
1.1	Starforce Battlement	2
1.2	특징	4
2	개요	5
2.1	파일 위치 및 빌드 주의사항	5
2.2	데모 사용 방법	6
2.2.1	메인 메뉴	6
2.2.2	게임플레이 및 HUD	8
3	구성	10
3.1	Flash 파일	10
3.2	ActionScript 코드	10
3.2.1	구현 개요	11
3.2.2	ActionScript 패키지	15
3.2.3	프레임워크	15

1 소개

Scaleform Mobile Game Kit 은 Autodesk® Scaleform®을 엔진 및 렌더러로 사용하는 터치 기반 모바일 게임 빌드 구현의 모범 예제를 제공합니다. Starforce Battlement 는 ActionScript 3 으로 작성된 게임으로 Scaleform 의 'Shippable Mobile Player'의 장점을 활용한 범용 플랫폼을 제공합니다. 자세한 구현 정보는 이 문서에 포함되어 있습니다. 이 게임은 특별히 모바일 기기를 대상으로 설계되었지만 PC, Mac, Linux, iOS 및 Android 를 포함하여 마우스 또는 터치 입력을 지원하는 모든 플랫폼에서 플레이할 수 있습니다.

이 키트는 다해상도 지원, 사운드 재생, 장치에 저장 및 로딩, 업적, iOS Game Center 와의 통합을 위한 샘플 프레임워크 등 게임 및 메인 메뉴의 모든 ActionScript 3 소스 코드를 포함합니다. 이 키트는 또한 게임에 사용된 모든 Adobe® Flash® 콘텐츠와 자산을 포함합니다. 개발자는 키트의 모든 자원을 다시 사용하거나 Scaleform 을 사용한 자체 게임의 구성과 구현에 모범 예제 샘플을 활용할 수 있습니다.

게임에서는 모바일 기기에서 Scaleform 을 사용하여 재생하도록 설계되었지만 Adobe Flash Player 가 설치된 웹 브라우저에서도 재생이 가능해 다양한 플랫폼에서의 Scaleform 메모리 점유와 성능 벤치마크를 제공합니다.

개발/런타임 관점에서 Starforce Battlement 의 핵심 구성 요소를 간략하게 분류하면 다음과 같습니다.

1. Flash Professional 은 상호작용하는 아트, 애니메이션, UI 레이아웃 및 레벨 편집에 사용됩니다.
2. Gameplay 로직은 ActionScript 3 를 사용하여 작성되었습니다(권장 에디터는 FlashDevelop 4).
3. 내보내기한 .SWF 파일은 기기의 Scaleform 에서 재생됩니다.
4. Scaleform 렌더러는 기기의 GPU 를 통한 재생 결과 그리기가 효과적으로 화면에 표시되도록 해 줍니다.

1.1 Starforce Battlement



그림 1: Starforce - 레벨 1 스크린샷

Starforce Battlement 는 롤플레이팅 게임 요소를 포함한 타워 디펜스 게임입니다. 각 레벨의 주 목적은 몰려오는 적을 상대로 목적지에 도착하지 못하도록 막는 것입니다. 적이 목적지에 도달할 때 마다 플레이어의 생명이 1 점씩 내려갑니다. 적이 계속 목적지에 도달하여 플레이어의 생명이 0 점이 되면 게임이 종료됩니다.

플레이어는 맵에서 장소를 지정하여 유닛/타워를 지어 몰려드는 적이 목적지에 도달하는 것을 막을 수 있습니다. 적 제거 시 주어지는 점수로 플레이어는 새 타워를 짓거나 기존 타워를 업그레이드할 수 있습니다. 모든 공격에서 적을 제거하면 레벨이 끝납니다.

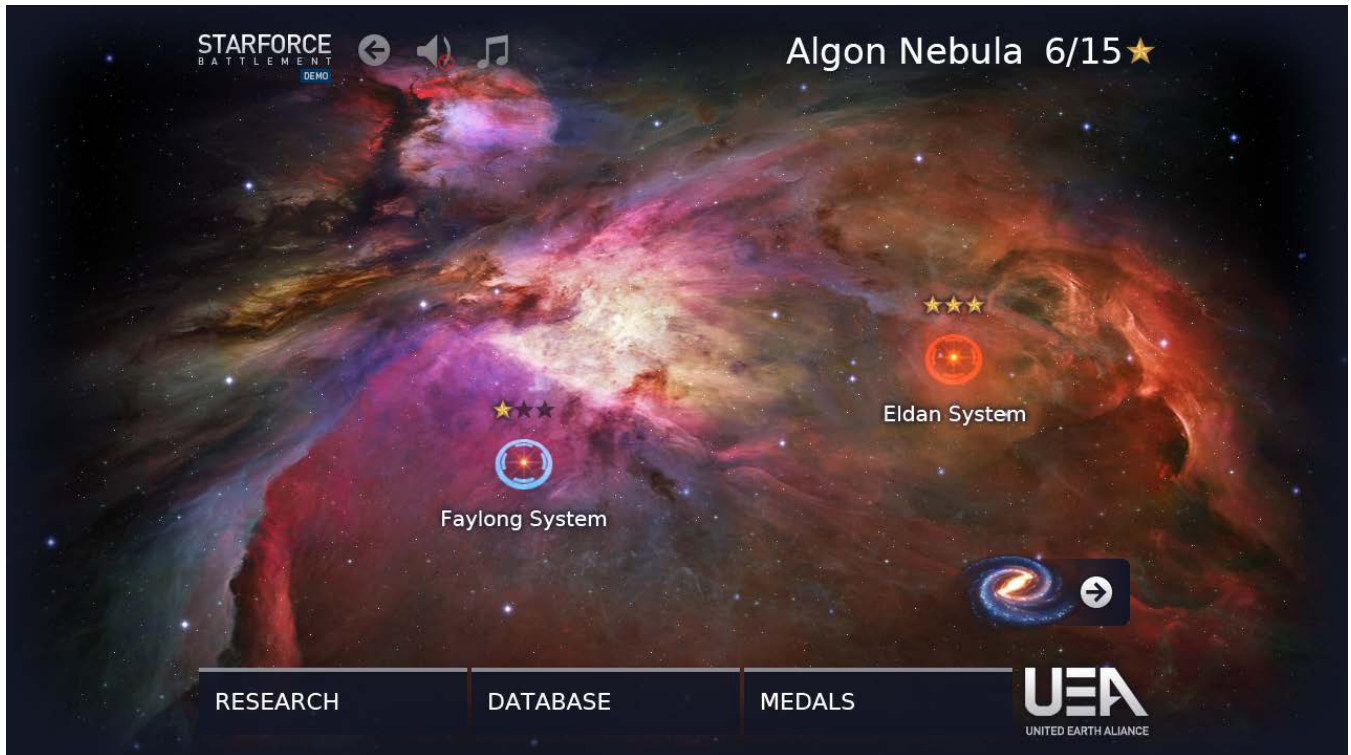


그림 2: Starforce - 메인 메뉴

Starforce Battlement 풀게임 데모는 또한 레벨 선택, 튜토리얼, 플레이어 통계/업적 보기가 있는 메인 메뉴를 포함하며 이들은 개발자가 다른 이름으로 재사용할 수 있습니다.

Starforce Battlement 는 2012 년 5 월부터 iOS App Store 에서 무료로 다운받을 수 있습니다. 개발자들은 이 게임을 다운받아 iOS 기기에서 Scaleform 이 어떻게 구동되는지 살펴보는 것을 권장합니다.

1.2 특징

Starforce Battlement 는 다음의 Scaleform 모바일 특징을 담고 있습니다.

1. 범용 플랫폼 모바일 지원
2. GPU 벡터 및 비트맵 렌더링
3. ActionScript 3 완벽 지원
4. 멀티 터치 및 동작 입력 처리
5. 데이터 저장 및 로딩
6. 사운드 및 음악 재생
7. iOS Game Center 과의 통합
8. 방향 고정 / 방향 처리

이 키트는 또한 다음에 사용되는 ActionScript 3 프레임워크 샘플을 제공합니다.

1. 플레이어 업적
2. 사운드 및 음악 재생
3. iOS Game Center 과의 통합
4. 데이터 저장 및 로딩

2 개요

2.1 파일 위치 및 빌드 주의사항

다음 위치에 본 데모와 관련된 파일이 있습니다.

- *Bin/Data/AS3/Kits/StarforceTD/* - ActionScript 3 소스 코드, Flash 콘텐츠 및 게임 플레이에서 사용되는 다른 리소스 포함(아이콘, 사운드 등).
- *Projects/Win32 /{Msvc80, Msvc90, or Msvc10}/GfX 4.0 SDK /* - Windows 용 FxPlayer Visual Studio 2005/2008/2010 빌드 및 디버그를 위한 Visual Studio 프로젝트 포함. 이 응용 프로그램은 Starforce 재생을 자동으로 설정하지 않습니다.
- *Projects/iPhone/Xcode4/GfX 4.0 iPhone SDK* - iOS "Shippable Mobile Player"용 Xcode 4 포함. 이 응용 프로그램은 Starforce 재생을 자동으로 설정하지 않습니다.
- *LocalApps\StarforceTD\Android* - make 를 통한 Scaleform 구성에 처음 성공하면 Android 용 StarforceTD 의 재생을 위해 이미 설정된 "Shippable Mobile Player"에 따라 적절하게 설정된 Eclipse 프로젝트가 이 디렉토리에 포함됩니다.

Bin/Data/AS3/Kits/ StarforceTD 에 미리 빌드된 Windows 용 실행 데모파일 *StarforceTD.exe* 가 있습니다. Windows 시작 메뉴나 Scaleform SDK 브라우저에서도 접근할 수 있습니다.

FxPlayer 또는 FxMobilePlayer 응용 프로그램용 프로젝트/솔루션 파일은 여러 플랫폼에서 모바일 게임 키트를 컴파일 및 배치, 실행하는데 사용됩니다.

Windows 에서는 디버깅을 위한 "작업 디렉토리"를 반드시 *Bin/Data/AS3/Kits/StarforceTD* 로 설정하고 "Command line arguments"를 *StarforceTD.swf*로 설정해 주십시오.

iOS 의 경우 Xcode 으로 응용 프로그램을 빌드하고 기기에 배치한 뒤 iTunes 를 사용하여 SWF 파일과 하위 디렉토리(오디오, 아이콘)을 응용 프로그램으로 옮깁니다.

2.2 데모 사용 방법

2.2.1 메인 메뉴

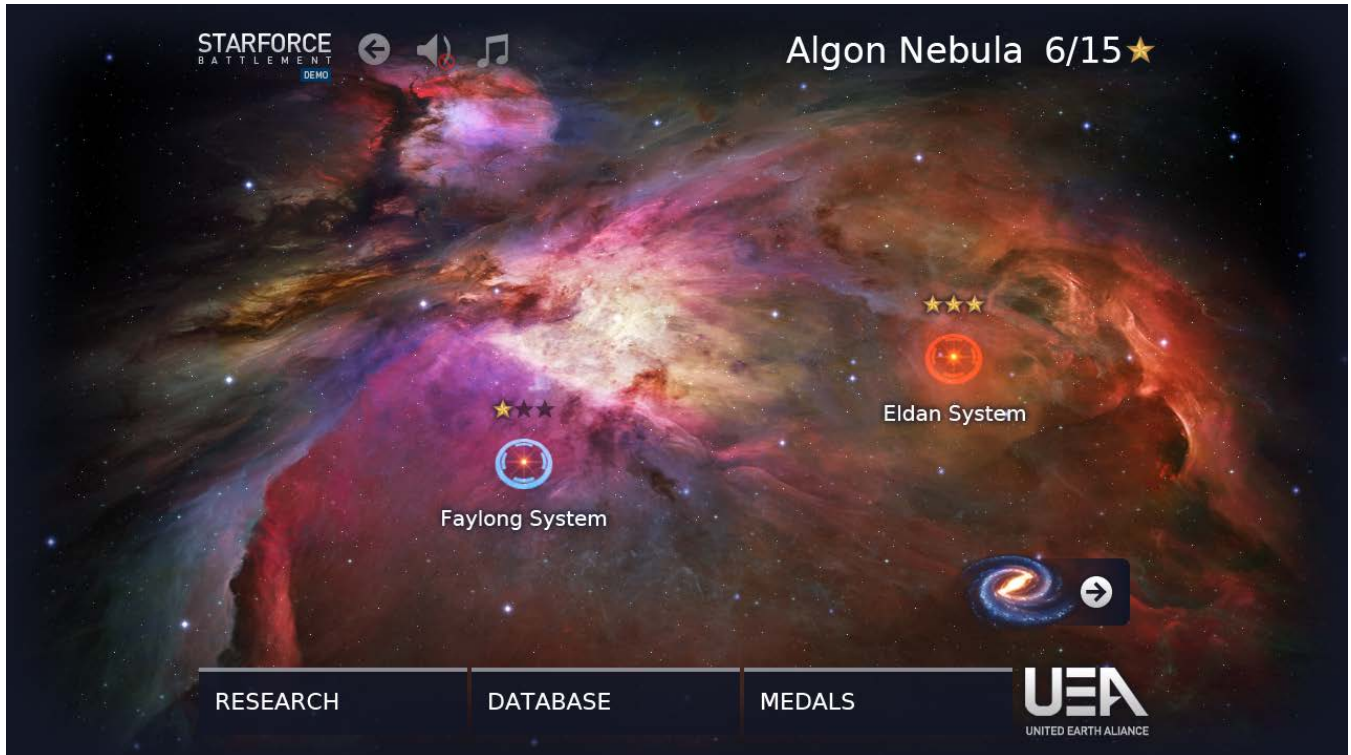


그림 3: Starforce - 메인 메뉴

StarforceTD.as 는 게임의 입구점(entry point)입니다. 이 클래스는 StarforceTD.fla 의 문서 클래스이며 초기화 시 StarforceMenu.swf 를 로드하는 MainMainState 를 함께 초기화합니다.

StarforceMenu.swf 는 메인 메뉴 아트 자산과 메인 메뉴를 움직이는 ActionScript 3 클래스 정의를 포함합니다.

메인 메뉴는 플레이어가 미션을 선택할 수 있도록 해 주며 플레이어의 기술 트리 변경 및 튜토리얼, 플레이어 통계 기록 및 업적을 볼 수 있는 추가 옵션을 제공합니다. 메인 메뉴의 왼쪽 상단에 있는 버튼들은 음악 및 음향 효과를 전환하거나 게임을 종료합니다.

레벨 선택은 플레이어가 선택한 Galaxy 또는 Nebula 맵의 표시기들로 구성되어 있습니다. 각 Nebula 는 직선형 진행에 따라 언록(unlock)되는 미션들을 포함합니다. 플레이어는 맵을 2D 공간으로 끌어당겨 특정 미션을 선택하기 전에 사용가능한 옵션을 볼 수 있습니다.



그림 4: Starforce - 미션 데이터

플레이어가 미션을 선택하면 미션 데이터 보기(MissionDataView.as)가 표시됩니다. 미션 데이터 보기는 해당 미션의 세부 정보를 제공합니다. 여기서 플레이어는 "Enter Mission" 버튼을 선택하여 미션을 시작하거나 오른쪽 상단의 "X" 버튼을 선택하여 Mission Select View 로 다시 돌아갈 수 있습니다.

2.2.2 게임플레이 및 HUD



그림 5: Starforce - UI 가 포함된 레벨 1 게임플레이

Starforce Battlement 의 레벨 및 게임플레이는 StarforceTD.swf 에 저장되어 있습니다.

StarforceTD.swf 를 처음 시작하면 게임 레벨의 모든 데이터를 메모리로 불러오지만 LevelGameState 이 생성되기 전에는 레벨 초기화가 되지 않습니다. 게임은 다섯 개의 미션/레벨로 이루어지며 각각 고유한 설정을 가집니다.

첫 공격은 위 스크린샷 오른쪽에 있는 "Rush Wave" 버튼(교차된 두 개 검은색으로 표시)을 클릭해야 시작됩니다. 첫 공격이 시작되면 레벨 내내 적들이 일정한 간격으로 몰려들며 모든 적이 제거되거나 플레이어가 패배할 때까지 지속됩니다. 현재 공격 번호 및 총 공격 횟수는 HUD 의 왼쪽 아래 부분에 있고 다음 공격에서 공격하는 적의 미리보기 정보가 함께 표시됩니다.

플레이어가 모든 공격을 성공적으로 클리어하면 목적지에 도달한 적 수에 따라 1, 2 또는 3 개의 별로 레벨 평가가 주어집니다. 해당 별은 플레이어의 기술 트리를 키우는데 사용할 수 있습니다.

타워 건축 지점은 맵에서 다이아몬드와 검정색 원으로 표시되며 레벨 진행 동안 산발적으로 나타납니다. 선택하면 지점 위에 원형 메뉴가 표시되고 플레이어가 새 타워를 지을 수 있습니다. 타워 건축 지점에는 한 번에 한 개의 타워만 지을 수 있습니다. 타워는 사용가능한 유효 지점에서만 세울 수 있고 플레이어는 타워 건축에 필요한 점수를 보유해야 합니다. 플레이어의 현재 점수는 HUD의 오른쪽에서 플레이어 체력 바로 위에 표시됩니다.

건축된 타워는 최대 두 번까지 업그레이드될 수 있습니다. 현재 타워 레벨과 다음 레벨 통계 정보는 해당 타워 선택 시 HUD 가운데 표시됩니다.

플레이어의 능력은 "Orbital Strike(외곽 공격)" 및 "Deploy Mercenaries(용병 배치)" 두 가지로 HUD 오른쪽 아래에 있는 해당 능력 버튼을 클릭하여 사용할 수 있습니다. 능력을 활성화하면 플레이어는 해당 능력을 적용할 위치를 맵에서 선택해야 합니다. 실행된 Orbital Strike은 마우스나 터치를 통해 누른 상태로 끌어당겨 움직일 수 있습니다. 사용한 능력은 '대기 모드(cool down)'로 되어 일정 시간 동안 사용할 수 없습니다. 대기 모드 유지 시간은 능력 버튼 위에 어두운 반투명 색으로 표시되며 시간이 따라 점차 사라집니다.

HUD의 오른쪽 상단에 있는 버튼들은 튜토리얼 보기, 게임 일시중지/계속, 음향 효과 전환, 음악 전환 및 현재 미션 다시 시작하기/종료 기능을 제공합니다.

3 구성

3.1 Flash 파일

대부분의 게임 아트 자산은 Flash .FLA 파일에 있으며 여기에는 레벨, 개체, UI, 게임 애니메이션 대부분이 포함됩니다. Starforce Battlement 는 비트맵 아트를 일차적으로 사용하는데 이는 유닛과 타워 모델이 편의를 위해 3D 로 제작되었기 때문입니다.

UI 에 아이콘을 중심으로 한 게임 아트의 하위 그룹은 FLA 파일 외부에 저장됩니다. 실행 시 ActionScript 에서 이러한 외부 이미지를 불러옵니다.

Starforce Battlement 는 세 가지 FLA 파일로 구성됩니다.

1. **StarforceTD fla** – 게임의 주 사용 .FLA. 이 파일은 Starforce Battlement 의 모든 레벨, UI 및 개체의 ActionScript 코드와 아트 자산을 포함합니다. 또한 메인 메뉴 / 로딩 화면을 로딩 및 언로딩함으로써 게임을 관리하는 매니저 역할을 합니다.
2. **StarforceMenu fla** – 이 파일은 메인 메뉴 및 미션 선택의 ActionScript 코드와 아트 자산을 포함합니다. 일부 사용자 인터페이스 클래스 정의를 StarforceTD.swf 와 공유하지만 전적으로 독립적이며 따로 테스트할 수 있습니다.
3. **LoadingView.swf** – 이 파일은 게임 상태 전환 중 표시되는 로딩 화면의 ActionScript 코드와 아트 자산을 포함합니다.

3.2 ActionScript 코드

3.2.1 구현 개요

Starforce Battlement의 게임플레이 및 UI 로직은 모두 ActionScript 3로 작성되었습니다. iOS Game Center과의 통합을 제외하면 Starforce에서 구현을 위한 특정 기기 요구가 없기 때문에 게임플레이 코드의 이동이 매우 간편합니다.

3.2.1.1 입구점(Entry Point)

게임의 입구점은 StarforceTD.as 클래스이며 Flash 문서 클래스 속성을 통해 StarforceTD.swf에 따릅니다. 문서 클래스에 대한 자세한 정보는 온라인의 Adobe's Flash 문서를 참조하십시오.

3.2.1.2 Game 상태

게임은 세 개의 'GameStates'로 구분되고 이들은 각각 다른 게임 상태를 나타냅니다. 한 번에 한 개의 GameState만 활성화되며 활성화된 GameState는 StarforceTD 클래스가 소유하고 관리합니다. 세 가지 게임 상태는 다음과 같습니다.

1. MenuGameState – 메인 메뉴 GameState.
2. LoadingGameState – 로딩 화면 GameState.
3. LevelGameState – 레벨 및 게임플레이 GameState.

StarforceTD 생성자는 하위 시스템(데이터, 사운드, 멀티 터치)을 초기화하고 StarforceMenu.swf 및 MainMenuView.as에서 정의된 메인 메뉴를 불러옵니다.

3.2.1.3 메인 메뉴 구현

메인 메뉴는 주로 com.scaleform.std.menu 패키지(패키지 세부 정보는 아래 참조)를 통해 구현됩니다. 메인 메뉴는 ActionScript 3의 자체 이벤트 시스템을 통해 이벤트를 메인 메뉴 보기 간에 이동하는 이벤트 기반의 구성을 중심으로 빌드되었습니다. 사용자가 선택하면 창 열고 닫기, 데이터 읽기 및/또는 완전히 새로운 상태로 전환하기(예: 미션 시작)를 위한 이벤트를 보내고 처리합니다.

미션이 시작되면 메인 메뉴에서는 GameEvent 를 보냅니다. StarforceTD.as 클래스는 이 이벤트를 받아 즉시 LoadingGameState 를 로딩하고 MenuGameState 를 언로딩 한 뒤 LevelGameState 를 로딩하여 레벨을 전환합니다. LevelGameState 는 선택한 레벨을 로딩하고 레벨 로딩이 완료되는대로 게임을 시작합니다.

LoadingGameState 은 기술적인 관점에서 봤을 때 StarforceTD.swf 내 메모리에 이미 로드된 레벨 콘텐츠를 가져오기 때문에 필수적이지 않은 요소입니다. 그러나 여기서 LoadingGameState 전환은 레벨 및 사용자 HUD 의 초기화에 사용되며 게임플레이 중 보다 부드러운 전환을 제공합니다.

3.2.1.4 게임플레이 구현

Level.as 는 게임 구현의 핵심 요소입니다. 이 클래스는 게임의 모든 레벨의 기본 클래스가 됩니다. Level.as 클래스는 레벨, HUD, 적, 타워 및 사용자 능력을 포함하는 거의 모든 게임 요소를 초기화하고 관리합니다. Level 은 모든 레벨의 기본 클래스입니다. 각 레벨은 고유한 하위클래스를 가지며 이 하위클래스는 적을 정의하고 행위 및 변수를 오버라이트하여 해당 레벨을 더욱 고유하게 만들어 줍니다(예: 기본 gold, 공격 간격, 적 유형, 공격 횟수 등).

Level 은 모든 개체의 생성, 제거 및 관리를 담당합니다. 게임 개체는 보통 타워와 적이 말합니다. 모든 개체는 레벨에 따라 관리되고 업데이트 됩니다(예: 맵 상 이동, 적 찾기, 공격 등).

Flash Professional 은 ActionScript 클래스와 심볼(그래픽, 애니메이션 및 ActionScript 로 구성)을 쉽게 바인딩하게 해 줍니다. 바인딩된 심볼은 ActionScript 를 통해 실행 시 참조 및 초기화가 가능합니다. Starforce Battlement 는 이 방법을 사용하여 대부분의 게임 및 메뉴 시스템 그래픽 콘텐츠를 인스턴스화합니다. 심볼과 ActionScript 을 바인딩하고 실행 시 인스턴스를 생성하는 자세한 정보는 Adobe Flash 문서를 참조하십시오.

예를 들어 TowerMech 심볼은 StarforceTD.fla 의 Flash 라이브러리에 있습니다. TowerMech 심볼의 속성에서 심볼은 com.scaleform.std.entities.TowerMech ActionScript class 에 바인딩됩니다. 이 클래스가 ActionScript 를 통해 인스턴스화될 때 마다 이 그래픽 심볼의 인스턴스가 생성되어 코드를 통해 조정될 수 있습니다.

Level.as 클래스는 이벤트 기반 프레임워크를 사용하여 게임을 관리합니다. 클래스는 개체, 사용자 인터페이스 또는 다른 시스템 프레임워크(사운드, 데이터, 업적) 등에서 보낸 이벤트를 받아 게임과 HUD 를 업데이트합니다. 예를 들어 타워가 유닛을 공격할 때는 공격받는 유닛의 체력을 직접 수정하는 대신 타워가 AttackEvent 를 보냅니다. 전체적인 AttackEvents 를 받는 Level 에서 해당 AttackEvent 를 받아 공격자와 수비자를 처리하고 수비자가 받는 데미지를 계산하여 해당 데미지를 수비자에 적용합니다.

게임 개체 간 상호 운용성은 대부분 Level 을 통과하여 다양한 클래스를 서로 대비하여 추상화하고 개체 범위 외부의 데이터를 계산에 포함합니다(예: 플레이어 기술 트리는 일부 경우에 수비자가 받는 데미지 양에 영향을 줄 수 있음). 이러한 설계 패턴은 중복된 코드를 제거하고 개체 클래스를 각각 독립적으로 유지하기 때문에 다른 여러 게임 내 변경도 간편하게 해 줍니다.

3.2.1.5 게임 데이터

모든 타워, 미션, 능력 및 적에 대한 통계를 포함하는 게임 데이터는 대부분 com.scaleform.std.data 패키지에서 정의됩니다. 이 데이터는 GameData static 클래스를 사용하는 구현 내내 액세스됩니다. 이에 따라 GameData 클래스가 데이터를 요청 클래스로 돌려보내기 전에 외부 변수에 의한 영향을 대비하여 데이터를 수정할 수 있도록 해줍니다(예: 플레이어 기술 트리가 타워의 표준 공격 데미지를 변경할 수 있음).

게임의 기본 데이터는 ActionScript 에 저장하지 않고 XML 또는 JSON 과 같은 일부 외부 데이터 정의로 저장하여 실행 시 불러오는 것이 이상적입니다. 하지만 Scaleform 4.0 에서는 XML 가 지원되지 않아 클래스 정의 내에 해당 데이터를 저장하는 것이 가장 빠른 방법이었습니다. 모범 사례를 반영하여 차후 배포 버전에서 업데이트할 예정입니다.

3.2.1.6 입력 처리

게임 입력은 ActionScript 이벤트 리스너를 사용하여 작성됩니다. 이에 따라 입력 로직이 범용 플랫폼 기능을 수행하여 Scaleform Player 가 플랫폼에서의 마우스/터치 입력을 추상화하고 ActionScript 로 보낼 수 있습니다.

게임은 Scaleform 응용 프로그램에서 터치 입력을 지원하는지 여부에 따라 마우스 또는 터치 입력을 받습니다. FxPlayer 가 올바르게 설정되었다고 가정했을 때 MultitouchInputState 이 MovieView 에 설치되었는지 그리고 터치에 관련된 운영 시스템 함수가 사용가능한지의 여부에 따라 응용 프로그램에서 실행 시 터치 지원 여부를 감지합니다.

Starforce 의 모든 상호작용 요소는 TouchEvents 및 MouseEvents 모두에 대한 로직을 가지지만 응용 프로그램 설정에 따라 한 가지 리스너 세트만 사용됩니다. TouchEvents 와 MouseEvents 유형은 다르지만 많은 TouchEvent 유형과 MouseEvent 유형은 서로 매우 유사합니다. 예를 들어 TouchEvent.TOUCH_TAP 는 MouseEvent.MOUSE_CLICK 와, TouchEvent.PRESS 는 MouseEvent.MOUSE_DOWN 와 비슷하게 사용할 수 있습니다.

하나의 함수를 TouchEvents 및 MouseEvents 모두에 대한 리스너로 사용하려면 함수 파라미터가 Event 유형이어야 합니다. *target* 또는 입력 이벤트의 위치 좌표 등 이벤트의 정보가 요구되는 경우 이벤트가 적절하게 캐스트되어야 합니다. 실행 시 유형을 감지하는 표준 방법 중 하나는 "is" 실행자이며 아래와 같습니다.

```
protected function onSetRallyClick( e:Event ):void {
    var point:Point;
    if (e is MouseEvent) {
        var me:MouseEvent = e as MouseEvent;
        point = new Point( me.stageX, me.stageY );
    }
    else
        var te:TouchEvent = e as TouchEvent;
        point = new Point( te.stageX, te.stageY );
    }
}
```

3.2.2 ActionScript 패키지

게임의 ActionScript 코드는 기능에 따라 기능에 따라 그리고/또는 포함된 클래스의 목적에 따라 ActionScript '패키지'로 나누어집니다.

다음의 모든 패키지는 "com.scaleform.std" 접두사를 가집니다.

1. **abilities** – 플레이어 능력(Orbital Strike, Hire mercenaries)에 대한 클래스 정의 및 게임플레이 로직.
2. **controls** – 계속 재사용되는 UI 요소 및 구성 요소에 대한 클래스 정의.
3. **core** – 상태 흐름 및 데이터 저장/복구의 기본 구성 요소에 대한 클래스 정의.
4. **data** – 플레이어, 개체, 레벨 및 능력을 포함하는 게임 데이터 정의.
5. **entities** – 플레이어, 개체, 레벨 및 능력을 포함하는 게임 데이터 정의.
6. **events**– 모든 게임, UI 및 상태 이벤트에 대한 이벤트 정의.
7. **fx** – 게임 내 특수 효과(폭발, 사망 시 애니메이션)에 대한 클래스 정의.
8. **hud**– 인게임 HUD 및 메뉴에 대한 클래스 정의.
9. **levels** – 각 레벨에 대한 클래스 정의. Level 클래스는 게임에서 주로 사용되는 클래스 중 하나입니다.
10. **loading** – 로딩 화면에 대한 클래스 정의.
11. **menu** – 모든 UI 요소, 미션 선택 및 입력 처리를 포함하는 메인 메뉴에 대한 클래스 정의.
12. **system** – 사운드 재생, 저장 및 로딩, 방향 처리 등의 작업을 위한 ActionScript 과 시스템 상호 작용을 위한 프레임워크.
13. **utils** – codebase 에서 사용되는 유틸리티 클래스, 함수 및 정의를 포함하지만 특정 패키지에 국한되지는 않음.

3.2.3 프레임워크

3.2.3.1 프레임워크 저장/로딩

프레임워크 저장 및 로딩은 com.scaleform.std.system.SaveSystem.as 에서 정의되며 디스크에 데이터를 쓰고 읽는 샘플 구현입니다. 플레이어 진행을 저장하는데 사용되는 코드 자체는 간단하지만 보다 복잡한 구현을 위한 기반을 마련해 줍니다.

이러한 기능을 위한 ActionScript 인터페이스는 SharedObject 클래스를 활용합니다. SharedObject 는 표준 ActionScript 3 클래스로 개발자들이 디스크의 ActionScript Object 에 데이터를 쓰고 읽을 수 있도록 해줍니다. SharedObject 클래스의 Scaleform 구현은 원형 유형 및 Array 를 지원합니다.

각 SharedObject 는 실행 시 읽기 및 쓰기에 사용되는 고유한 문자열 식별자로 정의됩니다. 아래 코드에서 SharedObject 의 고유 문자열 식별자는 "sb_player_data"입니다.

```
// Galaxy 맵의 진행에 대한 로컬 레퍼런스로 PlayerData 로부터 생성됩니다.
public static var galaxyProgress:Array = null;
// 수정하려는 SharedObject 의 속성에 대한 상수 문자열입니다.
protected static const GALAXY_PROGRESS_SO_PROP:String = "galaxyProgress";
// 가져온 SharedObject 에 대한 레퍼런스입니다.
protected static var _playerDataSO:SharedObject = null;

public static function load( pd:PlayerData ):void {
    _playerDataSO = SharedObject.getLocal("sb_player_data");
    galaxyProgress = _playerDataSO.data[ GALAXY_PROGRESS_SO_PROP ];
    pd.setGalaxyProgress( galaxyProgress );
}

public static function save( pd:PlayerData ):void {
    // SharedObject 에 PlayerData 를 작성합니다.
    _playerDataSO = SharedObject.getLocal("sb_player_data");
    galaxyProgress = pd.getGalaxyProgress();
    _playerDataSO.data[ GALAXY_PROGRESS_SO_PROP ] = galaxyProgress;
    _playerDataSO.flush();
}
```

개발은 단일 SharedObject 인스턴스로 제한되지 않습니다. 각 SharedObjects 이 고유한 이름을 가진다면 디스크에 다수 저장도 가능합니다. 이 방법을 통해 데이터를 카테고리로 간편하게 나눌 수 있습니다.

SharedObject 클래스에 대한 자세한 정보는 다음을 참조하십시오:

http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/net/SharedObject.html

3.2.3.2 사운드 프레임워크

사운드 프레임워크는 com.scaleform.std.system.SoundSystem 에서 정의되며 게임 내 음향 효과 및 음악 재생을 처리하는 샘플 구현입니다. 프레임워크는 게임에서 보낸 SoundEvent 에 따라 디스크에서 외부 사운드를 불러와 재생합니다(이 경우에는 .mp3). SoundEvent 는 스테이지에 따라 모든 클래스에서 보낼 수 있습니다.

```
dispatchEvent( new SoundEvent( SoundEvent.PLAY_SOUND, true, true, "infantry_attack.mp3",  
                                1, false ) );
```

사운드 프레임워크는 백그라운드 및 포그라운드 두 개의 사운드 채널을 사용합니다. SoundEvent 는 대상 사운드가 재생될 채널을 지정합니다. 게임의 모든 음향 효과는 포그라운드 채널에서 재생되고 배경 음악은 백그라운드 채널에서 재생됩니다. 이에 따라 SoundEvent 를 통해 개별적으로 음소거하거나 전환할 수 있습니다.

```
dispatchEvent( new SoundEvent( SoundEvent.MUTE_SOUNDFX, true, true );
```