

# Autodesk® Scaleform®

## 手机游戏工具箱概述

本文介紹 **Scaleform 4.2 手機遊戲工具箱 (Scaleform 4.2 Mobile Game Kit)** 的架構、原代碼和內容。此工具箱是一個跨平臺的、基於觸摸/手勢的遊戲演示,專用於採用 **Scaleform** 手機播放機的手機和平板電腦硬體。

作者：Nate Mitchell

版本：1.01

上次編輯：2012 年 6 月 5 日

## 版权声明

### 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	<a href="http://www.scaleform.com">www.scaleform.com</a>
Email	<a href="mailto:info@scaleform.com">info@scaleform.com</a>
Direct	(301) 446-3200
Fax	(301) 446-3199

# 目录

<b>1</b>	<b>引言 .....</b>	<b>Error! Bookmark not defined.</b>
1.1	Starforce Battlement .....	2
1.2	特性 .....	4
<b>2</b>	<b>概述 .....</b>	<b>5</b>
2.1	文件位置和构建说明 .....	5
2.2	演示用法 .....	6
2.2.1	主菜单 .....	6
2.2.2	游戏方式和 HUD .....	8
<b>3</b>	<b>架构 .....</b>	<b>10</b>
3.1	Flash 文件 .....	10
3.2	ActionScript 代码 .....	10
3.2.1	实现摘要 .....	10
3.2.2	ActionScript 程序包 .....	13
3.2.3	框架 .....	14

# 1 引言

Scaleform 手機遊戲工具箱提供觸控式螢幕手機遊戲的最佳做法實現示例,這種觸控式螢幕手機遊戲是將 Autodesk® Scaleform® 作為引擎和渲染器構建的。遊戲《Starforce Battlement》是完全採用 ActionScript 3 編寫的,並利用了 Scaleform 的「可裝運的手機播放機」(Shippable Mobile Player) 實現跨平臺部署。本概述詳細介紹具體實現方法。儘管該遊戲是專門針對行動裝置設計的,但它也可以在支援滑鼠或觸屏輸入的任何平臺(包括 PC、Mac、Linux、iOS 和 Android)上播放。

此工具箱包括針對遊戲和主功能表的完整 ActionScript 3 原始程式碼,包括支援多解析度的示例框架、聲音播放、保存和載入到設備、成績以及 iOS 遊戲中心 (iOS Game Center) 集成。此工具箱包括遊戲中使用的所有 Adobe® Flash® 內容和素材 (Asset)。開發者可以重複利用此工具箱的所有資源,並且/或者作為其採用 Scaleform 的遊戲的架構和實現的最佳示例。

儘管設計此遊戲的目的是利用 Scaleform 在行動裝置上播放,但此遊戲也可以在採用 Adobe Flash Player 的 Web 瀏覽器上播放,這使其成為 Scaleform 在各種平臺上的記憶體佔用空間和性能的一個很好的基準。

下面從開發/運行時觀點對構成 Starforce Battlement 的核心元件進行簡單細分：

1. Flash Professional 用於互動藝術、動畫、UI 佈局和關卡編輯。
2. 遊戲邏輯是用 ActionScript 3 編寫的(FlashDevelop 4 是我們的推薦的編輯器)。
3. 通過 Scaleform 在設備上播放 Exported .SWF 檔。
4. Scaleform 的渲染器負責將播放結果繪製到設備 GPU 上,因而可以有效地顯示在螢幕上。

## 1.1 Starforce Battlement



图 1：Starforce – 关卡 1 屏幕截图

鏡像的城垛是一個包含角色扮演元素的城防遊戲（防禦塔遊戲）。每個关卡的主要目的是防止一波波逼近的敵人到達其目的地。每當一個敵人到達目的地、就從玩家生命值中減去一分。如果足夠敵人到達目的地、玩家生命值就減為零、因而遊戲結束。

玩家可以在地圖上指定位置構建軍隊/城堡(單位/塔)、用來攻擊逼近的敵人、阻止他們到達目的地。消滅敵人玩家可贏得積分（信用）、這些積分可花費在構建新的城堡上、也可以花在升級現有的城堡上。消滅所有各波敵人後、关卡就通過了。

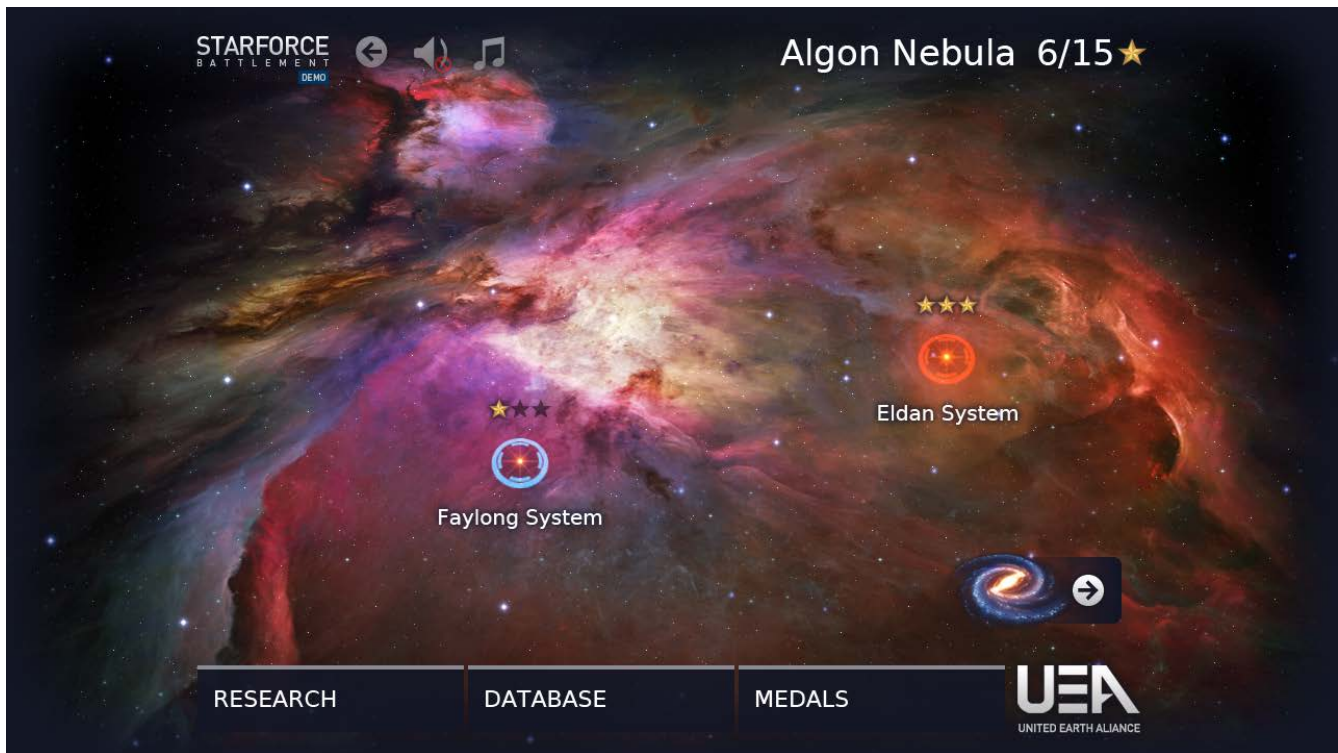


图 2：Starforce – 主菜单

作為一個完整的遊戲展示版本本,Starforce Battlement 還包括一個用於關卡選擇、教程和查看玩家統計資料/成績的主功能表,開發者可以在其自己的遊戲中重複利用它們。

2012 年 5 月,可在 iOS App Store 上免費下載 Starforce Battlement。歡迎開發者將該應用程式下載到 iOS 設備上,以便了解 Scaleform 在 iOS 上的工作情況。

## 1.2 特性

Starforce Battlement 展示下列 Scaleform 移動特性：

1. 跨平臺移動支援
2. GPU 向量和點陣圖渲染
3. 全面支援 ActionScript 3
4. 多點觸摸和手勢輸入處理
5. 保存和加载数据
6. 声音和音乐播放
7. iOS 遊戲中心集成
8. 定位閉鎖 (Orientation Lock) /定位处理 (Orientation Handling)

此工具箱還為下列特性提供 ActionScript 3 框架示例：

1. 玩家成績
2. 聲音和音樂播放
3. iOS 遊戲中心集成
4. 保存和載入資料



## 2 概述

### 2.1 文件位置和构建说明

與此演示關聯的檔位於下列位置：

- *Bin/Data/AS3/Kits/StarforceTD/* -包含 ActionScript 3 原始程式碼、Flash 內容以及玩遊戲期間使用的其它資源(圖示、聲音等)。
- *Projects/Win32 /{Msvc80, Msvc90, or Msvc10}/GfX 4.0 SDK /* -包含 Visual Studio 專案,用來在 Windows 上構建和調試 FxPlayer Visual Studio 2005/2008/2010。請注意,此應用程式沒有自動設定為播放 Starforce。
- *Projects/iPhone/Xcode4/GfX 4.0 iPhone SDK* -包含 Xcode 4 專案,用於 iOS 上的「可裝運手機播放起」(Shippable Mobile Player)。請注意,此應用程式沒有自動設定為播放 Starforce。
- *LocalApps\StarforceTD\Android* -一旦您首次通過 make 成功構建 Scaleform,此目錄就會包含一個正確配置的 Eclipse 專案,用於「Shippable Mobile Player」(已經配置為在 Android 上播放 StarforceTD)。

Bin/Data/AS3/Kits/ StarforceTD 中包含有用於 Windows 的演示的一個預構建可執行程式 StarforceTD.exe。它也可以通過 Windows 開始功能表或 Scaleform SDK 瀏覽器進行訪問。

用於 FxPlayer 的專案/解決方案檔或 FxMobilePlayer 應用程式用來在不同平臺上編譯、部署和運行手機遊戲工具箱。

在 Windows 上,確保將 Debugging(調試)的「Working directory」(工作目錄)設置為 Bin/Data/AS3/Kits/StarforceTD 目錄,而將「Command line arguments」(命令列參數)設置為 StarforceTD.swf。

在 iOS 上,通過 Xcode 將應用程式構建並部署到設備後,使用 iTunes 把 SWF 檔和子目錄(音訊、圖示)複製到應用程式。

## 2.2 演示用法

### 2.2.1 主菜单

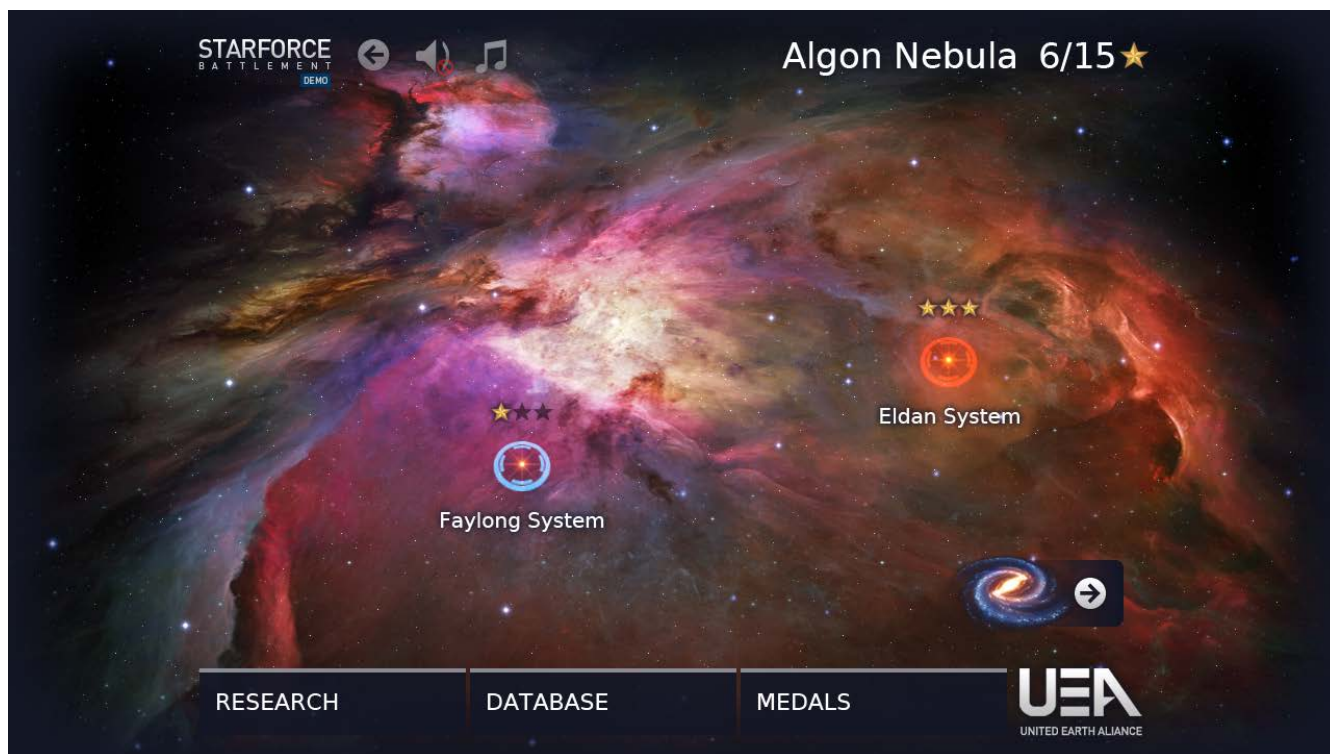


图 3：Starforce – 主菜单

StarforceTD.as 是遊戲的進入點。此類是 StarforceTD.fla 的 Document Class(文檔類),而且初始化時,會初始化 MainMainState。MainMainState 載入 StarforceMenu.swf。StarforceMenu.swf 包含「主功能表」美術素材,以及驅動主功能表的 ActionScript 3 類定義。

主功能表允許玩家選擇任務,並提供額外選項,用來更改玩家的技能樹,並查看教程、玩家統計資料歷史記錄以及玩家成績。主功能表左上部的按鈕允許玩家切換音樂、切換聲音效果或退出遊戲。

關卡選擇包含指示器,這些指示器位於玩家可以選擇的星系/星雲 (galaxy/nebula) 圖上。每個星雲包含一組以線性級數解鎖的任務。玩家可以在 2D 空間裡隨處拖動該圖,以便在選擇特定任務之前研究可用選項。



图 4：Starforce – 任务数据

當玩家選定一項可用任務時,就會顯示「任務資料檢視」(MissionDataView.as)。任務資料檢視 (Mission Data View) 為玩家提供有關任務的詳細資訊。從此視圖,玩家可以通過選擇「進入任務」(Enter Mission) 按鈕來啟動任務,也可以通過在視圖右上部選擇「X」按鈕來返回到「任務選擇視圖」(Mission Select View)。

### 2.2.2 游戏方式和 HUD



图 5：Starforce - 一级游戏（有用户界面）

Starforce Battlement 的關卡和遊戲方式封裝在 StarforceTD.swf 中。StarforceTD.swf 最初啟動時,會將遊戲關卡的所有資料載入到記憶體中,但在創建 LevelGameState 之後,才會初始化關卡。遊戲細分為五個任務/關卡,每個都有一個唯一的設置。

只有在點擊螢幕截圖右手側的脈動交叉劍按鈕 -- 「Rush Wave」(衝擊波)-- 按鈕後,才會啟動第一波敵人。啟動第一波後,敵人會在該關卡持續時間內以固定時間間隔到達,直到消滅所有敵人或者玩家被打敗。HUD 左下角顯示當前波數以及總波數,以及構成下一波敵人的預覽圖。

如果玩家成功消滅各波敵人,玩家將因其所過關卡而獲得一個 1 星、2 星或 3 星評級,具體星級取決於成功到達目的地的敵人的數量。這些星可用來發展玩家的技能樹。

城堡構建位置(用地圖上鑲有鑽石的黑圈表示)遍佈整個關卡。選定後,允許玩家構建新城堡的位置上面就會出現一個上下文徑向功能表。在一個城堡構建位置上一次只能構建一個城堡。只能在有效、可用的地點構建城堡,而且玩家必須擁有足夠積分來負擔城堡費用。玩家當前積分數顯示在 HUD 的右邊,就在玩家生命值 (Health) 上面。

一旦構建好一個城堡,最多可以升級城堡兩次。選定一個城堡時,當前城堡等級以及下一等級的統計資料就會出現在 HUD 中心。

玩家有兩項能力:「軌道炮打擊」(Orbital Strike) 和「部署雇傭兵」(Deploy Mercenaries),玩家可以通過點擊 HUD 右下部的相應能力按鈕隨時使用這兩項能力。啟動一項能力後,玩家必須在地圖上選擇一個啟動或定向該能力的位置。請注意,通過按住並拖動輸入(滑鼠或觸控式螢幕),可以拖動已經啟動的「軌道炮打擊」。一旦使用一項能力,該能力就會「冷卻下來」,並將在一限定時間段內處於禁用狀態。通過能力按鈕上方的一個深色調向使用者指明冷卻持續時間,該色調隨著時間的推移而逐漸消失。

HUD 右上部的那組按鈕允許玩家查看教程、暫停/恢復遊戲、切換聲音效果、切換音樂以及重啟/退出當前任務。

## 3 架构

### 3.1 Flash 文件

用於遊戲的多數美術素材均包含在 Flash .FLA 檔內,包括關卡、實體和 UI 以及遊戲的多數動畫。

Starforce Battlement 主要使用「點陣圖」(Bitmap) 美術,因為為了方便起見,單位和城堡的模型是用 3D 製作的。

包含遊戲的美術(主要是 UI 中使用的圖示)的一個子集不在這些 FLA 檔裡面。運行時通過 ActionScript 載入這些外部圖像。

Starforce Battlement 由三個不同的 FLA 檔組成：

1. **StarforceTD fla** –遊戲的主要 .FLA。此檔包含 Starforce Battlement 的所有關卡、UI 和實體的 ActionScript 代碼和美術素材。通過適當地載入和卸載「主功能表」(Main Menu) / 「載入屏」(Loading Screen),它還發揮遊戲管理器的作用。
2. **StarforceMenu fla** –此檔包含「主功能表」和「任務選擇」(Mission Select) 的 ActionScript 代碼和美術素材。儘管它與 StarforceTD.swf 共用一些使用者介面類定義,但它完全自包含,而且可以獨立測試。
3. **LoadingView.swf** –此檔包含「載入屏」(「遊戲狀態」轉換期間顯示)的 ActionScript 代碼和美術素材。

### 3.2 ActionScript 代碼

#### 3.2.1 实现摘要

Starforce Battlement 遊戲方式和 UI 的邏輯完全是用 ActionScript 3 編寫的。這使遊戲代碼高度可移植,因為除 iOS Game Center 集成之外,再沒有 Starforce 固有的針對設備的實現。

##### 3.2.1.1 进入点

遊戲的進入點是 StarforceTD.as 類,使用 Flash Document Class 屬性,該類被捆綁到 StarforceTD.swf。有關「文檔類」(Document Classes) 的更多資訊,請參閱 Adobe 的 Flash 說明文檔。



### 3.2.1.2 游戏状态

遊戲分為三種 ‘GameState’(遊戲狀態),每種代表遊戲可能處於的一個不同的狀態。一次只能啟動一種 GameState,而且由 StarforceTD 類所有,並且管理活動的 GameState。下面列出這三種遊戲狀態：

1. MenuGameState –用於「主功能表」的遊戲狀態。
2. LoadingGameState –用於「載入屏」的遊戲狀態。
3. LevelGameState –用於關卡和遊戲方式的遊戲狀態。

StarforceTD 的建構函式初始化子系統(資料、聲音、多點觸控),並載入 StarforceMenu.swf 和 MainMenuView.as 定義的主功能表。

### 3.2.1.3 主菜单实现

主功能表主要是用 com.scaleform.std.menu 套裝程式 (Package) 實現的(有關套裝程式的更多資訊,請參閱下文)。主功能表建立在一個事件驅動型架構之上,在這個架構上,採用 ActionScript 3 的本機事件系統在各種主功能表視圖之間傳遞事件。當使用者作出選擇時,就會發出並處理事件,以便打開和關閉視窗、讀取資料並且/或者轉換到全新的狀態(例如,啟動任務)。

啟動一項任務時,從主功能表發出一個 GameEvent(遊戲事件)。StarforceTD.as 類偵聽此事件,並通過載入 LoadingGameState、卸載 MenuGameState 以及載入 LevelGameState,立即轉換到該關卡。LevelGameState 載入選定的關卡,並且一旦完成載入該關卡,就立即開始遊戲。

請注意,從技術的觀點看,實際上不需要 LoadingGameState,因為將要繪製的關卡內容已經載入到 StarforceTD.swf 內的記憶體之中。不過,之所以在這裡使用 LoadingGameState 轉換,是為了向使用者隱藏關卡初始化和 HUD,並且更加順暢地過渡到遊戲當中。

### 3.2.1.4 游戏实现

Level.as 是實現遊戲的核心。此類充當遊戲的所有關卡的基類。Level.as 類初始化和和管理構成遊戲的幾乎所有元素,包括關卡、HUD、敵人、城堡和使用者能力。Level 是所有關卡的基類。每個關卡有其自己的子類,這些子類定義敵人,並可能會覆蓋其它行為和變數,以便使該關卡更加獨特(例如,預設黃金數量、各波之間的時間、敵人類型、波數,等等)。

Level 負責創建、銷毀和管理所有實體。一般情況下,遊戲中的實體為城堡和敵人。所有實體都按關卡打勾,這使實體得以更新(即跨越地圖、搜索敵人、攻擊,等等)。

Flash Professional 便利於將一個 ActionScript 類綁定到「符號」(Symbol),而符號由圖形、動畫和 ActionScript 構成。然後,在運行時,可以通過 ActionScript 對這些符號進行引用和具現化。這是 Starforce Battlement 在具現化遊戲和功能表系統中的幾乎所有圖形內容時所使用的方法。有關將「符號」綁定到 ActionScript 和在運行時創建實例的更多資訊,請參閱 Adobe 的 Flash 說明文檔。

例如,TowerMech 符號可在 StarforceTD.fla 的 Flash Library 中找到。在 TowerMech 符號的屬性中,請注意,該符號綁定到 com.scaleform.std.entities.TowerMech ActionScript 類。無論何時通過 ActionScript 具現化此類,都會創建此圖形符號的一個實例,然後通過代碼操控此實例。

Level.as 類使用一個事件驅動的框架管理遊戲。該類偵聽從實體、使用者介面或任一其它系統框架(聲音、資料、成績)發出的事件,並相應地更新遊戲和 HUD。例如,當一個城堡 (Tower) 攻擊一個部隊 (Unit) 時,該城堡不是直接修改其攻擊的部隊的生命值,而是發出 AttackEvent。全域地偵聽 AttackEvents 的「關卡」(Level) 收到該 AttackEvent,處理攻擊者和防禦者,計算防禦者應該受到的傷害的數量,並將該傷害情況應用到防禦者。

遊戲實體之間的大部分互通性是作為從實體彼此內部提取各種類並將這些實體範圍之外的資料包含在計算之內的一種手段來通過關卡的(例如,玩家的技能樹,它可能會影響某些情況下防禦者受到的傷害的數量)。此設計模式也便利於進行許多全遊戲範圍內的更改,因為它消除了冗餘碼,並使實體類彼此獨立。

### 3.2.1.5 游戏数据

com.scaleform.std.data 套裝程式中定義了遊戲的多數資料,包括所有城堡、任務、能力和敵人的統計資料。在整個實現過程中,使用 GameData 靜態類訪問這些資料。這樣,假如目前有外部變數影響這些資料(例如,玩家的技能樹可能會更改對某個城堡的標準攻擊力),在把資料返回到請求類之前,GameData 類就可以對這些資料進行修改。

理想的情況是,不把遊戲的預設資料存儲在 ActionScript 內,而是使用某種像 XML 或 JSON 這樣的外部資料定義格式,然後在運行時載入。令人遺憾的是,Scaleform 4.0 不支援 XML,因此,最便捷的解決辦法是將資料存儲在類定義內。這一問題可能會在未來的版本中得到修正,以便反映最佳做法。

### 3.2.1.6 输入处理

对游戏的输入是使用 ActionScript 事件侦听器编写脚本的。这使输入逻辑可以跨平台工作，因为 Scaleform 播放器将会跨越平台提取鼠标/触摸输入，并将其正确地传递到 ActionScript。



遊戲根據本機 **Scaleform** 應用程式當前是否支援觸摸輸入來偵聽滑鼠或觸摸輸入。假設 **FxPlayer** 設置正確,應用程式將會根據 **MovieView** 上是否安裝了 **MultitouchInputState**,以及與觸摸相關的適當作業系統功能是否可用,在運行時識別是否支援觸摸功能。

儘管根據應用程式設定只積極使用一組攔截器,但 **Starforce** 中的所有互動元素均同時具有 **TouchEvent** 和 **MouseEvent** 的邏輯。**TouchEvent** 和 **MouseEvent** 的類型是不同的,雖然許多 **TouchEvent** 類型都具有非常相似的 **MouseEvent** 類型,反之亦然。例如,**TouchEvent.TOUCH\_TAP** 可像 **MouseEvent.MOUSE\_CLICK** 一樣使用,而 **TouchEvent.PRESS** 也可像 **MouseEvent.MOUSE\_DOWN** 一樣使用。

要將單個函式宣告同時作為 **TouchEvent** 和 **MouseEvent** 的攔截器,該功能的參數必須屬於類型 **Event**(事件)。假如需要從該事件獲取更多資訊,例如,**target**(目標)或輸入事件的座標,則必須將該事件歸到適當類型中。運行時識別類型的一個標準方法是「**is**」操作符,如下所示：

```
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 程序包

根據所包含的類的功能和/或用途,遊戲的 **ActionScript** 代碼分成若干 **ActionScript**‘套裝程式’(Package)。

所有下列套裝程式均前面帶有「**com.scaleform.std**」：

1. **abilities** (能力) – 玩家能力(軌道炮打擊、雇用雇用兵)的類別定義和遊戲邏輯。
2. **controls** (控件) – 整個過程中重複使用的 UI 元素和元件。
3. **core** (核心) – 狀態流 (State Flow) 和資料存儲/檢索的基本元件的類定義。
4. **data** (数据) – 遊戲的資料定義,包括玩家、實體、關卡和能力。
5. **entities** (实体) – 遊戲的資料定義,包括玩家、實體、關卡和能力。
6. **events** (事件) – 所有遊戲、UI 和狀態事件的事件定義。
7. **fx** – 遊戲中特殊效果(爆炸、死亡動畫)的類定義。
8. **hud** – 遊戲內 HUD 和遊戲的類定義。
9. **levels** (关卡) – 每個關卡的類定義。請注意,**Level**(關卡)類是遊戲的基本類之一。

- 10. **loading** (加载) – 載入屏 (Loading Screen) 的類定義。
- 11. **menu** (菜单) – 「主功能表」的類定義,包括所有 UI 元素、任務選擇和輸入處理。
- 12. **system** (系统) – 執行聲音播放、保存和載入以及定位處理等任務時用來從 ActionScript 與系統進行互動的框架。
- 13. **utils** (实用工具) – 整個代碼庫(但不局限于特定套裝程式)中使用的公用程式類、函數和定義。

### 3.2.3 框架

#### 3.2.3.1 保存/加载框架

保存和載入框架(在 `com.scaleform.std.system.SaveSystem.as` 中定義)是在磁片上讀寫資料的實現示例。儘管用來存儲播放機進度的代碼非常簡單,但它可作為處理更複雜實現的基礎。

此功能的 ActionScript 介面採用了 SharedObject 類。SharedObject 是一個標準 ActionScript 3 類,它使開發者可以在磁片上存儲資料,並且可以從 ActionScript 物件檢索資料。Scaleform 對 SharedObject 類的實現支援圖元類型 (Primitive Type) 和陣列 (Array)。

每個 SharedObject 均由一個唯一字串識別碼定義,該識別碼用來在運行時讀寫資料。在下面的代碼中,SharedObject 的唯一字串識別碼為「sb\_player\_data」。

```
// Local reference to the galaxy progress. Populated from the PlayerData.
public static var galaxyProgress:Array = null;
// Constant String for the property of the SharedObject we'll modify.
protected static const GALAXY_PROGRESS_SO_PROP:String = "galaxyProgress";
// Reference to the SharedObject once we retrieve it.
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 {
    // Write PlayerData into SharedObject.
    _playerDataSO = SharedObject.getLocal("sb_player_data");
    galaxyProgress = pd.getGalaxyProgress();
    _playerDataSO.data[ GALAXY_PROGRESS_SO_PROP ] = galaxyProgress;
    _playerDataSO.flush();
}
```

請注意,開發者並不局限于單個 **SharedObject** 實例。只要每個 **SharedObject** 都使用一個唯一名稱,磁片上就可以存儲多個 **SharedObject**。這是一種把資料分成若干類別的便捷方式。

有關 **SharedObject** 類的更多詳細資訊,請訪問：

[http://help.adobe.com/en\\_US/FlashPlatform/reference/actionscript/3/flash/net/SharedObject.html](http://help.adobe.com/en_US/FlashPlatform/reference/actionscript/3/flash/net/SharedObject.html)

### 3.2.3.2 声音框架

聲音框架(**com.scaleform.std.system.SoundSystem** 中定義)是處理您的遊戲內的聲音效果和音樂播放的一個實現示例。該框架根據從遊戲發出的 **SoundEvent** 載入和播放外部聲音(在此案例中為 **.mp3**)。

**SoundEvent** 可從帶有對 **stage** 的引用的任何類發出。

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

聲音框架使用兩個聲音聲道,一個用於後臺,一個用於前臺。**SoundEvent** 指定目標聲音在哪個聲道上播放。遊戲的所有聲音效果均在前臺聲道上播放,而後台音樂則在後臺聲道播放。這樣就可以使用 **SoundEvent** 單獨設定每個聲道為靜音或在二者之間進行切換。

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