Autodesk®
Scaleform®

Menu Kit Overview

This document outlines the Scaleform 4.3 Menu Kit, a fully-featured, AAA user interface solution for PC and console games. It demonstrates splash to in-game content flow, screen resource management, background data loading, video streaming and PC/console input handling.

Author: Nate Mitchell, Prasad Silva

Version: 1.00
Last Edited: January 10, 2011

Autodesk®
GAMEWARE

Copyright Notice

Autodesk® Scaleform® 4.3

How to Contact Autodesk Scaleform:

Document	Menu 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

Table of Contents

1 Introduction

The Scaleform Menu Kit is the second in a series of high performance, full-featured AAA quality user interface kits which developers can use to learn and understand the Scaleform workflow and also integrate into their game. The Menu kit demonstrates how to create an efficient high performance menu system with video support and additionally shows how to implement common splash and loading screens using Scaleform.

The demo begins with an example splash screen, which contains a logo sequence, disclaimer message and an intro video. Each screen can be skipped via Escape key/circle button/B button on PC, PS3 and Xbox 360 respectively.

Figure 1: Menu Kit Introduction Video

The splash screen transitions to the main menu. The main menu contains a list of items that link to the Start Game, Options and Exit sub-screens. The items are part of a list element, and pressing on an item will transition the menu to the appropriate sub-screen.

Figure 2: Main Menu Screen

The Settings option in the main menu transitions to the settings screen, which provides a list of options. These options are fully interactive. Note that changes made here are not applied in the application. The options list demonstrates a list control that contains complex list item renderer elements (with option steppers, numeric steppers, and checkboxes).

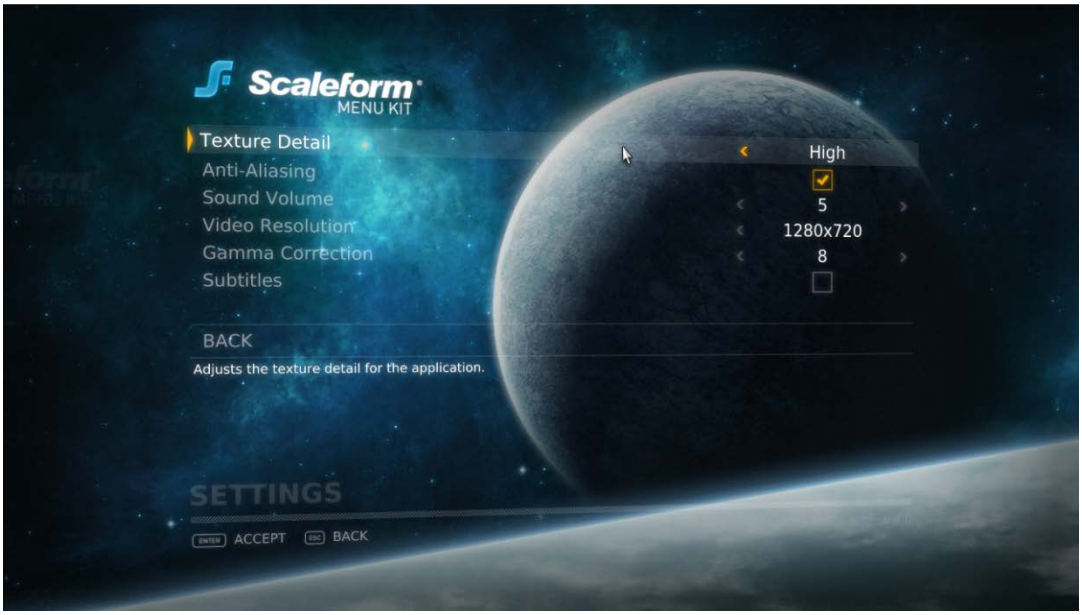

Figure 3: Options Screen

The Exit option in the main menu transitions to the exit dialog. Pressing Exit Game will cause the demo to quit. Pressing Cancel transitions back to the main menu.

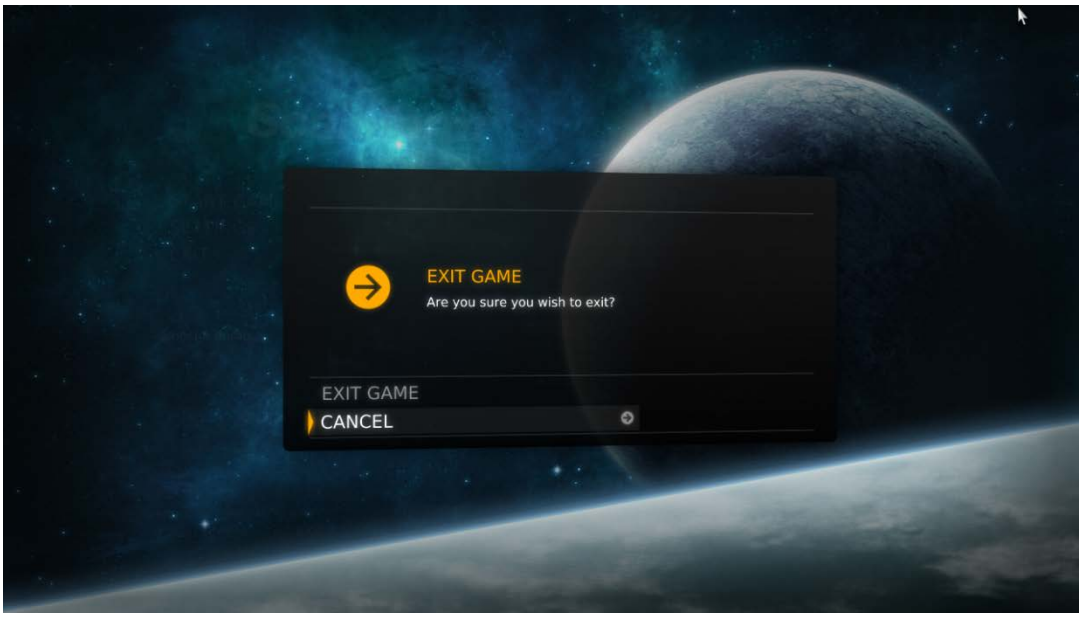

Figure 4: Exit Dialog

The Start Game option in the main menu transitions to the difficulty selection screen. Selecting any of the difficulty items will transition to the loading screen. The list items are also tied to image elements that animate in the z axis.

Figure 5: Difficulty Selection Screen

The loading screen mimics a background loading operation while a streaming video is displayed in the foreground. The background loading task loads actual file content from disk, and yields appropriately to the video I/O task. The progress bar at the bottom shows the current loading progress, and when the bar reaches 100%, the demo transitions to the in-game state.

Figure 6: Loading Screen

At any time during the demo, pressing the Pause key on PC or left-trigger button on consoles will generate a fake system error. This error causes a dialog with an appropriate message to display and also halts the content rendered below.

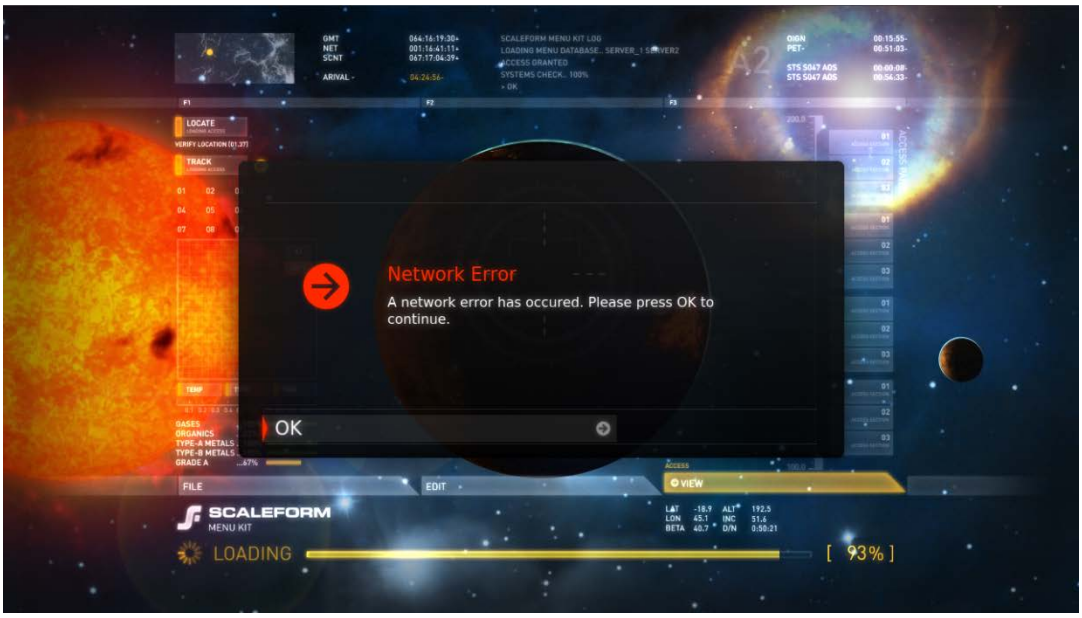

Figure 7: Error Dialog

The in-game state is a placeholder of possible game data. Pressing the appropriate button will restart the demo from the beginning.

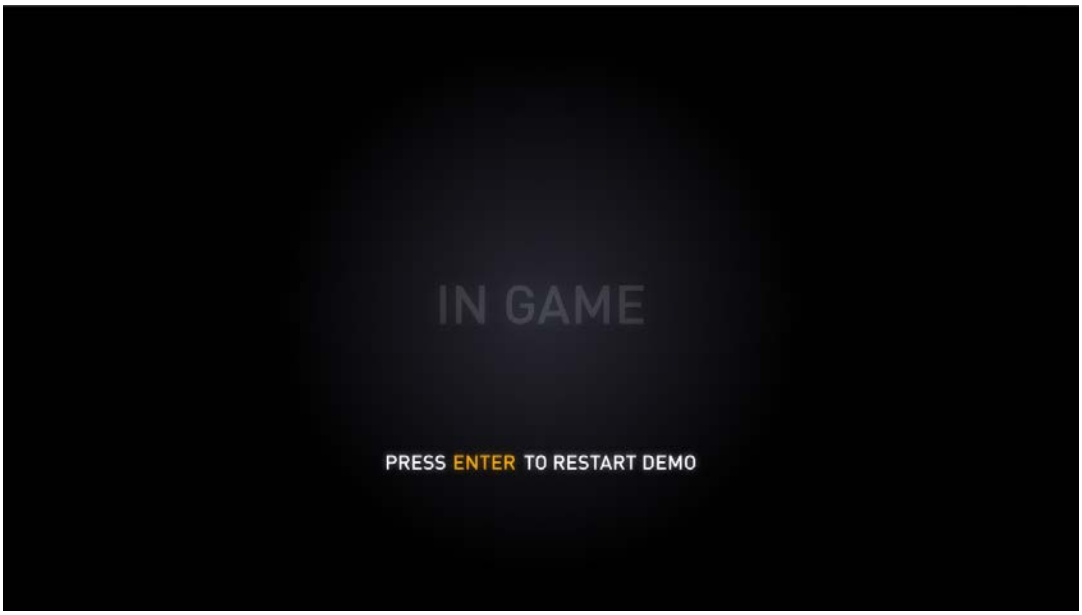

Figure 8: In-game State

2 Overview

2.1 Features

The menu kit demo features the following:

- A fully-functional menu built using complex CLIK widgets.
- Splash to in-game content flow.
- Screen resource management.
- Background data loading.
- Video streaming.
- PC and console input handling.
- Full source for application and menu.

2.2 File Locations and Build Notes

The files associated with this demo are located in the following locations:

- *Apps\Kits\Menu* - Contains C++ code for the Menu Demo executable.
- *Bin\Data\AS2 or AS3\Kits\Menu* - Contains Flash assets, ActionScript code and other binary data (AS2 directory contains ActionScript 2/Flash 8 assets, and AS3 directory contains ActionScript 3/Flash 10 assets).
- *Projects\Win32\{Msvc80 or Msvc90}\Kits\Menu* – Contains the demo projects for Visual Studio 2005/2008 on Windows.
- *Projects\Xbox360\Msvc90\Kits\Menu* – Contains the demo projects for Visual Studio 2008 on Xbox 360.
- *Projects\Common\MenuKit.mk* – PS3 makefile for the Menu Kit.

A pre-built executable of the demo Windows, MenuKit.exe, can be found in the *Bin\Kits\Menu* directory. It is also accessible via the start menu or the Scaleform SDK Browser.

On Windows, the Scaleform 4.3 Kits.sln file located in the *Projects\Win32\Msvc80\Kits (or Msvc90\Kits)* directory can be used to build and run this demo. Be certain that the "Working directory" for Debugging is set to the *Bin\Data\AS2\Kits\Menu* or *Bin\Data\AS3\Kits\Menu* directory before running the demo from the solution.

On Xbox 360, the Scaleform 4.3 Kits.sln file located in the *Projects\Xbox360\ Msvc90\Kits\Menu* directory can be used to build, deploy, and run this demo. All of the assets for the demo, located in *Bin\Data\AS2 or AS3\Kits\Menu*, will be deployed to the target Xbox 360 when compilation completes.

For PS3, the make command should be run from the root of the Scaleform installation directory. This will, by default, build all available demos including the Menu Kit. On PS3, the executable can be launched via the SN Systems toolset. The executable will be built to Bin\PS3 and should be launched using the following options:

- app_home/ Directory: *{Local Scaleform Directory}\Bin\Data\AS2 or AS3\Kits\Menu*

3 Architecture

The Menu Kit consists of two parts: the Flash assets that make up the different screens and the C++ code that drives them.

3.1 C++

The overall structure of the application consists of game states. A game state is a logical state in the application that provides a discrete set of functionalities. For example, the splash state shows the splash screen and handles appropriate input management. Another example is the loading state which manages the background loading of game data in addition to updating the loading view on screen.

The game states manage lifetimes themselves and register with the main game entity which is responsible for delegating system and user events to the active state. Memory consumption for each state is quarantined within. Releasing a state is expected to free all memory used by that state and its children.

Each state has access to the UI manager, which provides support to display multiple SWFs at the same time, similar to a compositing approach, as well as the resource manager, which provides interfaces to setup SWF loaders, create background data loaders and manage sound banks. The resource manager provides support to create a sound manager, which allows the UI to generate sound events that cause sound samples to be played in turn.

The menu kit demo includes the following states:

- Start – A bootstrapping state that transitions to the splash state.
- Splash – Displays splash screens. Transitions to the main menu state at the end.
- MainMenu – Displays and handles the main menu and its sub screens. Exiting from the main menu quits the application. Selecting a difficulty level transition to the loading state.
- Loading – Performs background loading of game data and displays load progress in the foreground. Transitions to the in-game state at the end.
- InGame – Placeholder for real game data. Continues transitions to the start state.
- System – Handles high priority system events, such as system errors. This state exists throughout the lifetime of the application.

The MainMenu is the only interactive state. It provides separate interfaces for communicating between widgets on screen. All main menu sub-screens (views) has a corresponding C++ class. This class keeps a

reference via GFx::Value to the SWF file that it loaded, allowing the application to interact with the Flash content dynamically at runtime.

3.2 Flash

3.2.1 Content

The Flash content for the Menu Kit can be found in the following directory:
Scaleform SDK Installation Directory/GFx 4.3/Bin/Data/AS2 or AS3/Kits/Menu

The .FLA and .SWF files can be divided into groups based on the game state where each is used:

- Splash State
    - Splash.fla/swf – The splash screen content. Shows the Scaleform intro (Flash animation) followed by the Scaleform text, and then the intro movie (Scaleform Video).

- MainMenu State
    - MenuAssets.fla/.swf –The core Flash file for the Menu Kit. Contains all symbols and view layouts for the Menu Kit. All other SWF files in the MainMenu State import symbols from this file.
    - ErrorDialog.fla/.swf – An information dialog with an 'OK' button that is used in the Main Menu.
    - MainMenu.fla/.swf – The first menu that appears when the MainMenu State begins.
    - OptionDialog.fla/.swf – A dialog with 'OK' and 'Cancel' options. Used for the Exit Game dialog.
    - Settings.fla/.swf – The Settings view, which can be accessed from the Main Menu.
    - StartGame.fla/.swf – The Start Game view, which can be accessed from the Main Menu.

- Loading State
    - Loading.fla/.swf – The loading screen that follows the Main Menu. This file plays a GFxVideo with a Flash overlay while loading data. An accurate progress bar and loading percentage is shown at the bottom of the view.

- Game State
    - InGame.fla/.swf – A placeholder for the actual game. This file is displayed at the end of the demo.

- System Error State

- SystemUI.fla/.swf – A system error dialog.

The most important file in this directory is MenuAssets.fla which contains all of the components and view layouts for the menu portion of the Menu Kit. The FLA files for the main menu views (e.g. MainMenu.fla, SettingsView.fla, etc…) import all of their content from MenuAssets.fla. This architecture creates a single location for symbols across all views and therefore simplifies updating of assets.

When changing a component or view, the ActionScript class and/or associated symbol should be changed in MenuAssets.fla. After doing so, MenuAssets.swf must be republished. Republishing MenuAssets.swf will cause the other SWF files to be automatically updated, since these SWF files will load the updated content dynamically from MenuAssets.swf at runtime.

Note that the one exception to this rule is SystemUI.swf, which contains all of its own symbol definitions and does not import anything from MenuAssets.swf to avoid loading MenuAssets.swf into memory when the SystemUI.swf is displayed. The system state exists throughout the lifetime of the demo and necessitates a low memory footprint.

The Menu Kit is built using CLIK 4.0, included with the Scaleform 4 SDK. All custom CLIK components used in the Menu Kit can be found in the following directory:
Scaleform SDK Installation Directory/GFx 4.3/Bin/Data/AS2 or AS3/Kits/Menu/com/scaleform/

The standard CLIK codebase can be found in the following directory:
Scaleform SDK Installation Directory/GFx 4.3/Resources/AS2 or AS3/CLIK/

Note that the standard CLIK classpath must be added to your ActionScript 2.0 /3.0 Preferences to republish any of the Menu Kit's SWF files.

3.2.2 Implementation

To obtain C++ references to CLIK components, the Menu Kit uses an initialization callback framework. When any CLIK component is first initialized, it attempts to call the function _global.CLIK_initCallback() with its path, name, and reference as parameters. Note that this only occurs if the function is defined in the VM.

_global.CLIK_initCallback() is not defined by default, but can be defined via a function object in C++ using the Direct Access API. When ActionScript invokes the method, the parameters are sent to the C++ function that was assigned as the function object. More information on function objects and the Direct Access API can be found in the Game Communication documentation.

In the case of the Menu Kit, the **GameStateWidgetSupport** class provides a robust framework that is used to receive and process these CLIK initialization callbacks.