

Autodesk® Scaleform®

MMO 工具箱概述

本文介绍 **Scaleform 4.2 MMO** 工具箱的架构和内容。该工具箱是一个用于 PC MMORPG 的功能齐全、可重复利用的 **3A** 级用户界面解决方案。

作者: Nate Mitchell, Prasad Silva

版本: 1.00

上次编辑: 2012 年 1 月 19 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.

如何联系 Autodesk Scaleform:

文档	MMO 工具箱概述
地址	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
网站	www.scaleform.com
电邮	info@scaleform.com
电话	(301) 446-3200
传真	(301) 446-3199

目录

1	引言	1
1.1	特性	1
2	概述	3
2.1	文件位置和构建说明	3
2.2	演示用法	3
2.2.1	此演示从屏幕上加载和显示的名牌 (Nameplate)、库存 (Inventory)、动作条 (Action Bar)、窗口菜单 (Window Menu) 和聊天日志 (Chat Log) 开始。名牌	3
2.2.2	库存	4
2.2.3	动作条	5
2.2.4	施法条	5
2.2.5	工具提示	6
2.2.6	窗口菜单	6
2.2.7	纸娃娃（玩家设备和统计管理器）	7
2.2.8	魔法/能力书	8
2.2.9	聊天日志	9
3	架构	10
3.1	C++	10
3.1.1	C++ 文件	10
3.2	Flash	11
3.2.1	MMOKit.flas.....	12
3.2.2	拖放框架	12
3.2.3	窗口化框架	13

1 引言

Scaleform MMO（大型多人在线游戏）工具箱，旨在实现最佳性能和节约内存，为使用 Autodesk® Scaleform® 的大型多人在线游戏用户界面 (UI) 提供一个基础。

该工具箱包括基于 CLIK 的 MMO 界面元素（如库存 (Inventory) 和纸娃娃 (Paper Doll)）以及用于拖放 (Drag and Drop)、窗口化 (Windowing)、动态图标资源管理和组件数据绑定的框架。开发者可以重复利用资源 (Asset)、ActionScript 小部件和示例 C++ 代码，并且/或者在其游戏用户界面的架构和实现中用作一个最佳做法示例。



图 1：MMO 用户界面概览

尽管此工具箱为 MMORPG

提供一个现成的用户界面解决方案，但用户并不会受限于所提供的內容。我们期望用户自定义或扩展此工具箱的各个元素，以便于为任何类型的游戏或应用程序创建新的具有创意的界面。

1.1 特性

MMO 工具箱以下列可重复利用的 UI 小部件为特色：

1. 播放器和目标名牌
2. 魔法/能力书 (Spell / Ability Book)
3. 纸娃娃（玩家设备和统计管理器）
4. 库存
5. 工具提示
6. 速动 (Quick-Action) 条
7. 施法条 (Cast Bar)
8. 窗口菜单 (Window)
9. 带选项卡的聊天日志 (Chat Log with Tabs)

此工具箱还为下列框架提供逻辑：

1. 拖放
2. 窗口化
3. 动态外部资源管理
4. 游戏数据绑定到小部件

2 概述

2.1 文件位置和构建说明

与此演示 (Demo) 关联的文件位于下列位置：

- *Apps/Kits/MMO/* - 包含用于 MMO 工具箱演示可执行程序的 C++ 代码。
- *Bin/Data/AS3/Kits/MMO/* - 包含 Flash 资源和 ActionScript 代码。
- *Projects/Win32 /{Msvc80, Msvc90, or Msvc10}/Kits/MMO/* – 包含针对在 Windows 上运行的 Visual Studio 2005/2008/2010 的演示项目。

可在 *Bin/Kits/MMO* 中找到一个针对 Windows 的演示的预构建的可执行程序 MMOKit.exe。它也可以通过开始菜单或 *Scaleform SDK 浏览器* 进行访问。

在 Windows 上，位于 *Projects/Win32/Msvc80/Kits*（或 *Msvc90/Kits* 或 *Msvc10/Kits*）目录的 *Scaleform 4.2 Kits.sln* 文件可用来构建和运行此演示。从解决方案运行演示之前，务必将调试用 “Working directory”（工作目录）设置为 *Bin/Data/AS3/Kits/MMO* 目录。

2.2 演示用法

2.2.1 此演示从屏幕上加载和显示的名牌 (Nameplate)、库存 (Inventory)、动作条 (Action Bar)、窗口菜单 (Window Menu) 和聊天日志 (Chat Log) 开始。名牌



图 2：玩家和目标名牌

玩家 (Player) 和目标名牌 (Target Nameplate) 位于用户界面顶部左侧。左侧名牌显示玩家姓名、等级、肖像、健康状况和魔力（神奇的魔法力量）。右侧的名牌显示玩家的当前目标的同样的信息。姓名和等级信息捆绑到游戏数据，并将自动更新。不捆绑健康状况和魔力条，因为目前在后端模拟中它们不受影响。

2.2.2 库存



图 3：库存条

底部右侧的库存条可用来打开和关闭玩家的库存。库存分为单独的背包，每个背包均由库存条上的一个切换按钮表示。当打开一个包时，显示一个新的库存窗口，其中显示该包的内容。

每个包都可以移动到库存条中的另一个插槽。如果一个包中没有任何道具，就可以把它放在另一个包内。如果一个包不空，并且移动到库存条上一个已占用的插槽中，包之间就会交换插槽。最右面的包是背包 (Backpack)，不可以从其插槽中移动。



图 4：库存包

每个包均有一个可用于道具 (Item)

的库存插槽的预定义数量。如果同时打开多个包，库存将自动回流包的窗口的位置，因而使它们可以同时可见。

可以在库存周围移动或拖动道具，这可以在单个包内进行，也可以在包与包之间进行。要移动一个道具，请在该道具的图标上左击或者启动鼠标拖动（按住鼠标左按钮，并移动鼠标光标）。在此移动期间，图标将会跟着鼠标光标。如果移动是左击鼠标启动的，那就左击目标库存插槽来完成移动。如果移动是由鼠标拖动启动的，那就在目标库存插槽上释放鼠标左按钮来完成移动。

某些道具可以一个接一个地堆叠起来。包含堆叠的道具的插槽将在插槽底部右侧和中心显示一个小数字，表示该插槽的堆中有多少个道具。可能被堆叠的道具的示例可能包括像药水或食物这样的消耗品。在 MMO 工具箱中，包中的保健药水（红瓶图标）和魔力药水（蓝瓶图标）可以彼此堆叠在一起，这将会增加堆叠大小。

如果把某个道具放到了一个已经占用的库存插槽中，而且无法将其与另一个道具合并，这两个道具就会交换库存插槽。任何道具，如药水，均可拖动到一个动作条插槽中，以便于以后使用。这在动作条插槽与库存道具之间创建一个直接的链接。假如为一个可消耗道具或堆创建一个动作条道具，那么该道具将是玩家的库存中该类型道具的总数。可以将可装备的道具（如剑或盾）拖动到要装备的纸娃娃中。

2.2.3 动作条



图 5：动作条 1 和 2

每个动作条均由 12 个动作条插槽组成。每个插槽都充当一个拖放目标（用来更改插槽中的能力或道具）以及一个按钮（用来使用上述道具或能力）。默认情况下，有三个单独的动作条：界面底部中心的两个横条和屏幕右边上的一个竖条。左击某个动作条插槽中的一个图标将会尝试使用该技能或道具。

通过将一个图标拖动到一个不同的插槽中，可以移动或交换动作条插槽中的技能和道具。可以通过将技能或道具拖离动作条并完成移动来从一个动作条中删除这些技能或道具。这并不会毁坏技能或道具，而是毁坏链接。通过将技能从魔法书拖动到一个动作条插槽中，可以将技能添加到动作条。通过将道具从库存拖动到一个动作条插槽中，可以将道具添加到动作条。

请注意，动作条插槽支持全局和具体“冷却时间”(Cooldown)，此即无法在其中使用能力的时间期间。冷却时间由插槽上的一个径向滑动表示，该插槽在其从插槽顶部中心开始顺时针方向移动时，从一个完整的有色方块变为什么也没有了。

2.2.4 施法条



图 6：施法条

施法条向用户显示正在使用的技能的名称以及剩余“施法时间”，即直到动作实际执行的时间，表现为一个数字和一个进度条。当玩家使用一项不会立即执行的能力时，就显示施法条。当施法时间完成且执行动作时，施法条就会淡出，并保持隐藏状态，直到执行下一个动作。

2.2.5 工具提示

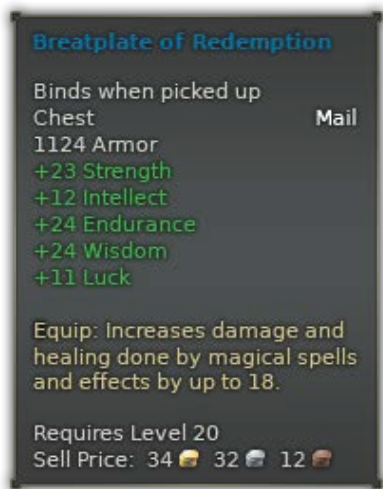


图 7：工具提示

在多数 MMORPG 中，所有道具和能力均有与其关联的数据，这些数据与用户相关，但需要比小图标所能提供的更多的不动产。例如，玩家可装备的一个剑道具可能拥有关于其速度、损害和重量的内部数据，玩家需要确定这些数据才能使用此剑。向用户显示此信息的一种常见的方法是：在用户在此道具上点击鼠标时显示一个工具提示。当用户的鼠标离开此道具/能力的空间时，工具提示就会被隐藏起来。

MMO

工具箱的工具提示是由来自后端的数据填充的，并且根据道具/能力进行自定义。工具提示主要通过 HTML 显示彩色文本和图像，但可根据游戏的需要进行进一步自定义。默认情况下，工具提示显示在该图标左上部位，不过，假如工具提示由于图标的位置而被设置为部分显示在屏幕之外，就会调节其位置，以确保在可见空间里显示该工具提示。

2.2.6 窗口菜单



图 8：窗口菜单

左下部位的窗口菜单可为用户提供一种切换并不总在屏幕上的各种小部件（如纸娃娃、魔法/能力书等）的方法，而不需要使用键盘快捷方式。窗口菜单由一组切换按钮组成，每个按钮有一个唯一图标，代表该按钮打开的屏幕。

在 MMO

工具箱中，窗口菜单可用来打开和关闭纸娃娃和魔法书。尽管目前禁用其它按钮，但是可以非常容易地启用它们，并进行挂接，以切换用户新建的小部件。

2.2.7 纸娃娃（玩家设备和统计管理器）



图 9：纸娃娃

纸娃娃 (Paper Doll)，或设备管理器 (Equipment Manager) 向玩家显示关于其角色及其当前装备的武器和盔甲的详细信息。它使玩家可以从其库存装备或卸下道具。

纸娃娃由三个部分组成：左侧的一个 ScrollingList（滚动列表）允许您切换中央面板中显示的信息；一个设备管理器，它包含一组纸娃娃的插槽，用来管理装备的道具（默认情况下，此面板显示 – 尽管此面板会根据左侧的用户选择发生变化）；以及右侧的一个 ScrollingList，它显示一系列箱子，其中包含 HTML TextField，用来显示有关玩家的角色的相关数据。请注意，玩家统计数据目前没有绑定到后端的数据。

纸娃娃的插槽也允许开发者限制可以装备的道具子集。例如，由于只有一个 “Head”（头）插槽供玩家使用，因此，“Helm of Redemption”（赎回头盔），一个 “Head” 插槽型道具，是玩家可以在该插槽中装备的唯一设备。假如用户试图在头盔插槽中装备一把剑，就不会发生任何变化，并将该剑归还给其前一个插槽。

纸娃娃的插槽还支持智能交换功能。例如，假如玩家在其左手装备一把单手剑，而在其右手装备一只盾，然后装备一把双手剑，那么那把单手剑和盾就会被放在用户的库存中，而双手剑就会占据左手和右手插槽。此功能可进一步扩展，以满足游戏的需要。

2.2.8 魔法/能力书



图 10：魔法/能力书

魔法/能力书是一个玩家可用的魔法和能力的列表。这些能力可从“魔法书”拖动到“动作条”上，以便于使用。点击“魔法书”中的能力，可启用上述能力。

能力是使用窗口左侧的一个 `ScrollingList` 进行分组的。选择一个组将会打开一个新的能力页面，而且每个组都可以有多个页面。每个能力都有一个图标、一个名称和一个头衔，它们都由后端数据进行填充。根据玩家的等级，某些能力可能无法使用，因此被禁止拖动或使用。

2.2.9 聊天日志

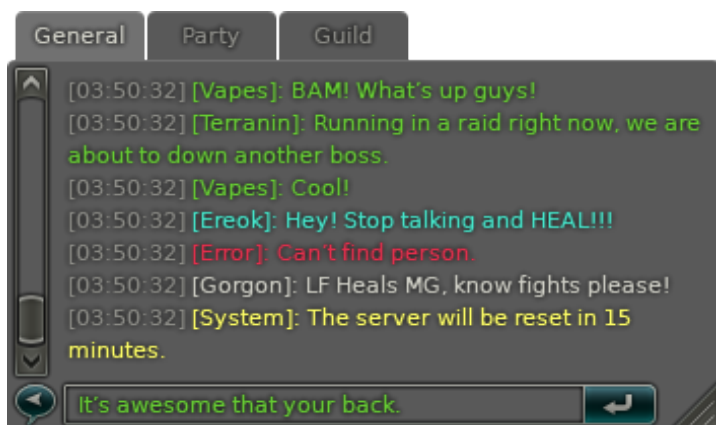


图 11：聊天日志

MMO 工具箱的聊天日志 (Chat Log)

是一个用来进行玩家通信的全功能接口和日志，由服务器数据进行填充。该日志支持多个聊天通道，这些聊天通道可划分并组织成为单独的聊天选项卡。“聊天日志”的输入字段也支持使用下列前缀将文本发送到某个通道：/s (Say)、/l (World)、/p (Party)、/g (Guild)。目标为其他玩家的消息可通过下面的语法进行发送：/t <player name> <message>。

“聊天日志”显示一个包含所有聊天消息的历史记录。该历史记录的长度和大小由应用程序确定。

3 架构

MMO 工具箱包含 Flash UI 资源以及 C++ 代码，C++

代码使用一个事件模型传送服务器的数据，而该事件模型最终驱动客户端视图。为了演示 MMO 工具箱与一个现有的客户端-服务器后台的互动，该工具箱建立在一种简单化客户端-服务器实现的基础之上，其中，服务器端模拟在适当时间生成数据，并将更新发送到客户端。

3.1 C++

C++ 代码的两个主要部分是：

1. 向客户端提供数据的环境（服务器）。
2. 客户端的与 Flash 内容的通信接口。

在 MVC

范式中，客户端与内容之间的通信接口是控制器，客户端的数据（最初由服务器提供）是模型，而 Flash 元素和 `ActionScript` 小部件构成视图。

本文详细介绍的代码提供一种最佳的 `Scaleform` 实现方式，即利用 `Scaleform Direct Access API` 完成完全从本机代码驱动的 UI 更新。不过，与 HUD 和菜单工具箱 (`Menu Kit`) 不同的是，MMO 工具箱将基本 `Direct Access API` 通信与一个数据绑定框架结合在一起，该框架将代表后端数据的唯一字符串绑定到 `ActionScript UIComponent`。倘若后端数据发生变化，就会立即自动告知绑定到特定标识符的 `UIComponent` 发生数据变化，因而 `UIComponent` 就可以适当更新其视图。

3.1.1 C++ 文件

C++ 代码包含下列文件（位于 `Apps/Kits/MMO/` 及其子文件夹）。

3.1.1.1 演示

- **MMOKitDemo.cpp** – 建立在标准 `Scaleform` 播放器基础之上的核心应用程序。处理平台应用程序实现和游戏初始化。
- **Game.cpp** – 处理客户端和服务器初始化以及它们之间的通信。

3.1.1.2 C++ 和 `ActionScript` 通信接口

- **GameUIDataBindings.cpp** – 用于 **GameUIAdapter** 类的多数定义。此类负责为 Flash 内容中的 **UIComponent** 创建绑定插槽。它处理 C++ 与 Flash 内容之间的大部分运行时通信。同时包含大部分插槽（库存插槽、动作条插槽、设备插槽等）处理，以便进行更新、拖动、放置和交换。
- **GameUIAdapter.cpp** – **GameUIAdapter** 类的核心。用来注册插槽、与客户端通信以及更新施法条的高级函数定义。

3.1.1.3 客户端和服务端

- **GameClient.cpp** – **GameClient** 类的定义。**GameClient** 从服务器接收事件，并通过与 **GameUIAdapter** 进行互动来处理这些事件，**GameUIAdapter** 使服务器的变化反映在 UI 中。
- **GameServer.cpp** – 一种针对 MMORPG 的简单的服务器实现。模拟 MMO 服务器，并为 **GameClient** 提供在 UI 中显示的数据。根据用户与 Flash UI 的互动，处理来自 **GameClient** 的消息。

3.1.1.4 运行时地图集汇编

- **GameUIRuntimeAtlasing.cpp** – 提供示例逻辑，该逻辑将一组外部图像打包到一个纹理地图集，以便于减少到达 GPU 的绘制调用 (Draw Call) 的总数。对于 MMO 工具箱，运行时将构成道具、魔法和能力的图标的所有单独的、外部的图像都打包到单个纹理地图集之中。这使美术师可以容易地添加新的图标，而不需要创建和管理纹理地图集。

3.2 Flash

用于 MMO 工具箱的 Flash 内容可在 *Bin/Data/AS3/Kits/MMO/* 目录中找到。本节提供的所有目录路径均与此目录相关。提供了一个 **FlashDevelop** 项目 **MMOKit.as3proj**，该项目可用来更加方便地导航和修改关联的 **ActionScript** 代码库。

构成 UI 的小部件（例如，纸娃娃、库存、动作条）分割到各个 **FLA** 文件中。每个 **FLA** 文件均为一个小部件提供图形布局、图像和动画。使小部件 **FLA** 互动的所有 **ActionScript** 逻辑均位于 *com/scaleform/mmo* 目录。每个小部件均有其自己的 **ActionScript** 类，这些类根据关联小部件被分成若干子文件夹。

例如，用于 **Inventory** 小部件的 Flash 内容可在 *widgets/InventoryView.fla* 中找到。用于 **Inventory** 的 **AS** 位于 *com/scaleform/mmo/inventory/*，其中，**InventoryView.as** 是 **Document Class**（文档类）。

3.2.1 MMOKit.fla

该工具箱的主要 FLA 可在此目录的根部找到：MMOKit.fla。这是运行时 MMOKitDemo 应用程序加载的第一个文件。MMOKit.as 作为工具箱中所有 ActionScript 逻辑的进入点，并且主要负责加载所有小部件和初始化像拖放这样的 ActionScript 框架。

用于 MMOKit.fla 的所有 ActionScript 均在 FLA 外部。主要 AS 文件是 Document Class for the FLA，com.scaleform.mmo.MMOKit，位于 com/scaleform/mmo/ 目录。

MMOKit 类负责在其配置 (configUI()) 期间初始化下列框架和子系统，此配置过程在第一个帧 (Event.ENTER_FRAME) 之后立即发生：

- TooltipManager
- WindowManager
- DragManager
- DataBinding
- GameDataModel

configUI()

方法也配置自定义鼠标光标，并加载初始小部件：聊天日志、窗口菜单、库存、动作条、名牌和施法条。

3.2.2 拖放框架

MMO 工具箱包括一个建立在以下两个主要类基础之上的基于 CLIK 的拖放框架：DragManager 和 DragSlot。

DragSlot

是基于按钮的类，该类充当一个可保存数据的按钮，这些数据可能包含，也可能不包含一个图标。按钮本身可用来与内部数据进行互动，或者从 UI 移动到其它 DragSlot。DragSlot 负责在用户互动时初始化初始拖动事件。DragManager 负责将 DragSlot 的数据/图标附加到原始 DragSlot（拖动开始的地方）与目标 DragSlot（拖动结束的地方）之间的数据移动和通信。请注意，要使 DragSlot 组件工作，必须初始化 DragManager。

DragSlot 类是 UI 中所有互动插槽元素的基础功能和骨架。DragSlot

应加以扩展，以使事件与游戏的后端适当地进行接口连接。许多重要函数，包括 startDrag() 和 endDrag() 均属于存根。这些存根可被开发者实现的子类所覆写。这使 DragSlot 成为整个 UI 内的许多不同类型插槽的基类。

例如，开发者可能需要一个在某个 `ActionBarSlot` 中开始而在某个 `InventorySlot` 中结束的拖动，什么目的也没有；需要一个在某个 `InventorySlot` 中开始而在某个 `ActionBarSlot` 中结束的拖动，目的是创建一个与库存道具的链接；以及需要一个在某个 `InventorySlot` 中开始而在某个 `InventorySlot` 中结束的拖动，目的是移动或交换该道具。通过给 `DragSlot` 划分子类，用户可以为 UI 中每个插槽类型创建独特的行为。

就 MMO 工具箱而言，每个小部件（库存、魔法书、动作条等）均有自己的 `DragSlot` 实现。请注意，所有这些具体的类均扩展 `com.scaleform.mmo.core.MDragSlot`，即一个扩展 `DragSlot` 的基类，其中包含在后端的 `GameUIAdapter` 上注册和注销 `MDragSlot` 的基本逻辑。

3.2.3 窗口化框架

MMO 工具箱包括一个基于 CLIK 的基本窗口化框架，该框架支持打开、关闭、移动并将焦点交换到 UI 内的窗口。此框架的核心是 `Window Manager` 类，该类为打开一个新窗口、更改焦点以及根据鼠标点击的最后一个窗口交换窗口深度。

窗口化框架将 `com.scaleform.mmo.core.MWindow` 类用于使窗口互动的逻辑。`MMOKit.fla` 中的 `MWindow`

符号用于基本窗口的视觉外壳：标题栏，以及可以重新调整窗口大小和关闭窗口的按钮。纸娃娃和魔法书小部件使用 `MWindow` 类将小部件嵌入一个 UI 窗口的上下文内。

`MWindow Symbol` 使用 `Scale9Grid`

嵌入任意大小的内容，而不会扭曲窗口的边缘和角落。窗口的背景是一个中型位图，根据内部的内容的大小进行缩放。假如窗口内容比原始背景位图的大小大出很多，背景就会出现微小伪影，不过这只在靠近观察时才会注意到。出于这个原因，开发者应该拥有多种大小的背景位图，以便于根据窗口中内容的大小使用这些背景位图。