

Autodesk® Scaleform®

AMP 使用者指南

本文介紹 Scaleform 記憶體和性能分析器 (AMP) 以及如何使用該系統來分析應用程式。

作者 : Alex Mantzaris
版本 : 3.03
上次編輯時間 : 2013 年 4 月 10 日

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.

如何聯繫 Autodesk Scaleform :

文檔	AMP 使用者指南
位址	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
網站	www.scaleform.com
電子郵件	info@scaleform.com
電話	(301) 446-3200
傳真	(301) 446-3199

目錄

1	引言	1
1.1	AMP 伺服器	1
1.2	AMP 用戶端	2
2	AMP 入門	10
2.1	進行連接	10
2.2	分析性能指標	11
2.3	分析渲染指標	12
2.4	分析記憶體消耗	13
2.5	使用透支分析	19
2.6	使用批量分析	19
3	平臺和集成說明	21
4	構建注釋	22
5	添加 AMP 支援	23
5.1	Debug 檔生成	23
5.2	應用程式控制	23
5.3	連接狀態	24
5.4	標記	24
6	與 AMP 相關的常見問題解答	25
7	更多資訊	26

1 引言

AMP™ 是 Scaleform® 公司的遠端分析系統,能夠監控 CPU 使用狀況、圖形渲染以及 Scaleform 應用程式記憶體消耗情況。即時的圖表提供一種快速查明記憶體和性能瓶頸的方法,而每幀統計資料功能允許深入分析和確定根源。AMP 利用其每幀 ActionScript 函數和原始程式碼計時,還有助於查找腳本問題所在。

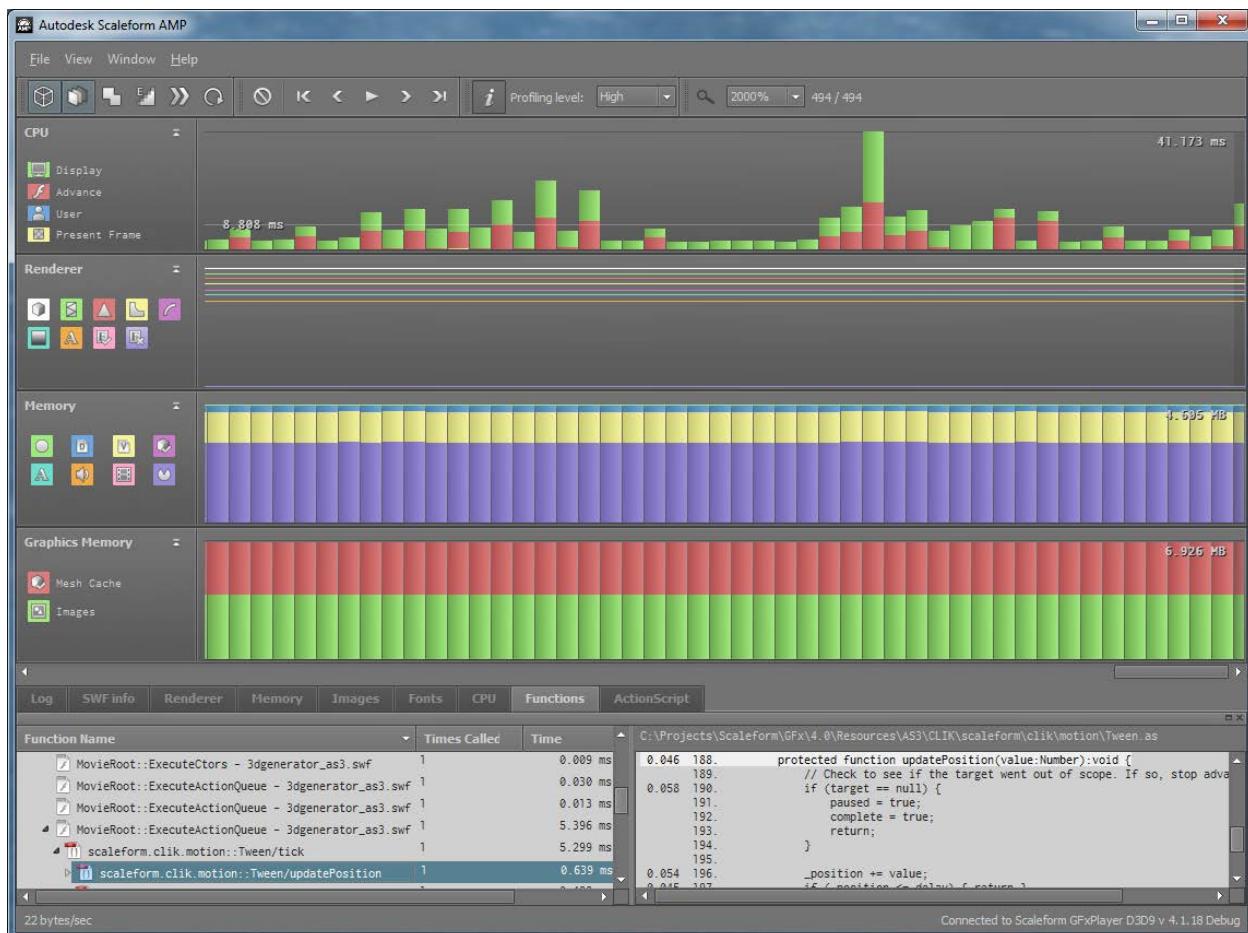


图 1 : Scaleform AMP

每個 Scaleform 應用程式都可以充當一個 AMP 伺服器,這就意味著,它可偵聽分析器進行的入站連接請求,收集每個幀的資訊,並通過網路將其發送給分析器。分析器可充當一個 AMP 用戶端,這意味著它可連接到特定的 AMP 伺服器,處理來自伺服器的每個幀的資訊,併發送請求以控制所分析的應用程式。

AMP 可用於 Scaleform 3.2 和更高版本。

1.1 AMP 伺服器

被分析的 Scaleform 應用程式(AMP 伺服器)通過網路將一定數量的資訊發送到分析器。可以根據函數分析級別對每個 ActionScript 函數和一些重要的 C++ Scaleform 函數進行計時。如果啟用 ActionScript 2 中的每

行原始程式碼分析,就會對每次 ActionScript 位元組碼執行進行計時(ActionScript 3 原始程式碼分析不需要計時每一個位元組碼)。此外,還記錄渲染統計資料、詳細記憶體堆和圖像資訊、Flash 檔資訊以及 ActionScript 標記。請注意,每個幀的這一額外工作都會減緩應用程式運行速度並增加其記憶體消耗 --- 在啟用每行計時的情況下尤其如此。

為了不讓 AMP 幷擾所分析的應用程式,將所有 AMP 伺服器資訊都保存在一個單獨的記憶體堆上。如果此記憶體堆的大小超出預設極限,就會暫停應用程式,直到通過網路發出待發幀資訊,並隨後將其刪除。 Scaleform Shipping 構造(Builds) 不充當 AMP 服務器,而且可以根據需要針對一個應用程序內的任何構造,容易地禁用 AMP 支持。

伺服器上的性能統計資料是按 Movie 維護的。這樣一來,倘若有多個視圖與特定 SWF 檔關聯,使用者就可以容易地查明問題點。它還使 AMP 可以為 ActionScript 在單一執行緒上執行多個 Movies 產生有意義的調用圖表。

除向用戶端報告記憶體和統計資料外,AMP 用戶端可以通過多種方式控制伺服器以便於進行分析。例如,分析器可以發送如下請求:線框模式渲染請求、暫停、快進或重啟 Flash 電影請求、切換防混疊(Anti-Aliasing)和筆劃模式請求、更改當地語系化字體請求以及更改向量圖形曲線公差。此外,用戶端可以切換特殊 AMP 渲染模式,以指出顯示性能瓶頸,例如,遮罩、篩檢程式和圖元無效渲染突出。

儘管在 Scaleform Player 中實現了上述所有功能,但其中某些功能還需要針對應用程式加以實現,而且可以作為 Scaleform 集成的組成部分進行執行。為了“優雅地”處理 Scaleform 應用程式中沒有實現其中某些功能的情形,AMP 伺服器向連接中的用戶端報告那組支援的應用程式控制功能。它還向用戶端報告其當前狀態(比如線框模式),進行視覺回饋。

最後,AMP 伺服器能夠通過網路將 Flash 調試資訊(ActionScript 2 only)發送到分析器,這樣一來,假如用戶端無法在本地訪問此資訊,也可以顯示原始程式碼和每行計時。為了實現這種傳輸,該 SWD 文件(ActionScript2)或 AS 文件(ActionScript3)應處在 AMP 可找到它的位置。

1.2 AMP 用戶端

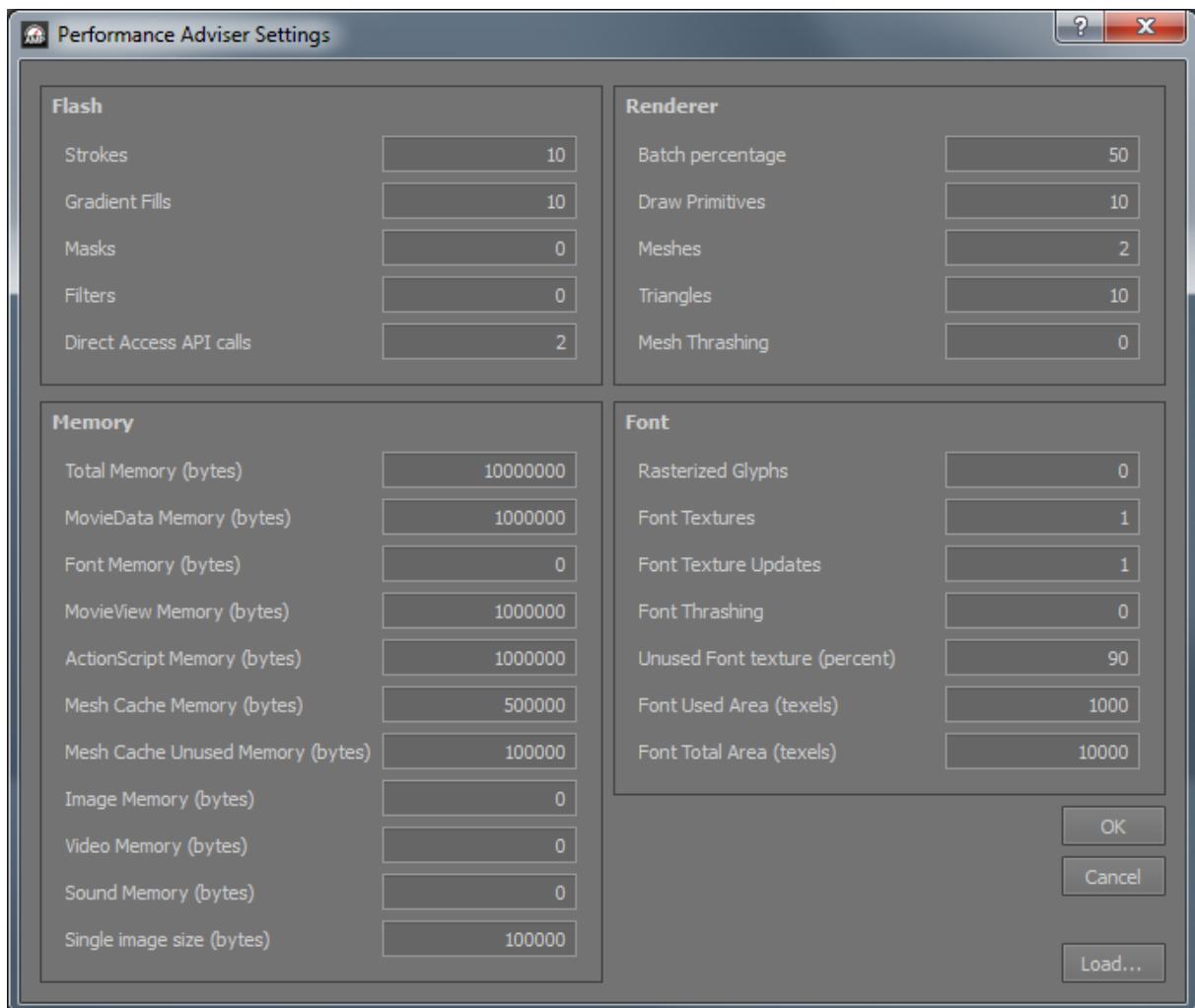
遠端分析器(AMP 用戶端)是一個連接某個 AMP 伺服器的獨立應用程式,接收每個幀的概要(Profile)資訊,並以一種對使用者友好的方式顯示此資訊,以便於有效地分析性能和記憶體瓶頸。分析器本身就是一個 Scaleform 應用程式,它是利用 Scaleform CLIK 實現的。

AMP 用戶端擁有多種特性,使開發人員可以查看其 Flash 資源的多個方面並查明具體問題所在:

- CPU 圖表：顯示在 ActionScript (Advance)、Scaleform 渲染 (Display)、Direct Access API (User) 和圖形緩衝區交換 (PresentFrame) 中每個幀花費多少時間。此圖表對於由於 Scaleform 引起的 CPU 使用峰值尤其有用。當選定一個帶有峰值的幀時，就可以通過檢查該幀的詳細統計資料來查明原因。使用者可以通過在此圖表上左擊並拖動滑鼠直到全部選中所需的幀來選擇多個幀。右擊並拖動此圖表可滾動圖表視窗。通過在圖例框中按一下相應的切換按鈕，可以顯示或隱藏此圖表的每個元件。
- **渲染圖表**:Scaleform 最佳做法指南 文檔標識多個與渲染相關的區域，這些區域可能導致性能影響，例如，繪製圖元、三角形和線條數量太多，或者存在遮罩、篩檢程式、筆觸或漸變填充。渲染圖表中顯示上述每個區域的計數器，因此，開發者可以更加深入地瞭解 Scaleform 內部所發生的情況。
- 記憶體圖表:記憶體圖表跟蹤 Scaleform 的記憶體使用情況，這使開發者可以按類別(例如，渲染、圖像、聲音、字體、電影資料等)監測系統記憶體使用情況，並包括所有載入的檔。
- 圖形記憶體圖表:Scaleform 圖形記憶體使用情況大部分也是通過 AMP 跟蹤的，而 AMP 報告針對圖像和網格緩存使用的 GPU 記憶體。
- “日誌”選項卡:點擊“日誌”(Log) 選項卡，可顯示被分析的應用程式在連接過程中產生的 Scaleform 日誌。
- **性能顧問(Performance Adviser)**：此選項卡提供如何提高 Flash 內容性能方面的指南。它是一個警告清單，當超出某個與內容相關的屬性的使用者定義的閾值時就會觸發這些警告。與性能相關的屬性有：
 - 針對可讀性而優化的文字欄位:假如您將「針對可讀性消除鋸齒」(Anti-alias for readability)用於大型文字欄位，那麼字體緩存中就包含每個字元的各種字體大小的單獨字形條目。如果一個文字欄位使用「動畫消除鋸齒」(Anti-alias for animation)，那麼不管字體大小如何，僅有一個條目。
 - 繪製圖元與網格數量之比:用來衡量批次處理效率。在沒有批次處理的最差情況下，每個繪製圖元將只有一個網格(比率為 1.0)。當某些網格作為同一繪製圖元的組成部分時，該比率下降，這表明批次處理效率提高了。
 - 繪製圖元數量:繪製圖元數量大表明出現渲染時間延長的複雜場景。
 - 網格數量:網格數量大也表明出現複雜場景。
 - 三角形數量:三角形數量大會影響記憶體和性能。
 - 網格緩存抖動:這是每個幀為給新條目騰出空間而從緩存中去除一個專案的次數。理想情況下，此數位為零，即渲染器所需的一切都可以在緩存中找到。
 - 筆觸次數:如「最佳做法」(Best Practices) 指南中所示，Flash 內容中的筆觸會影響性能。
 - 漸變填充數量:如「最佳做法」指南中所示，漸變填充數量也會影響性能。
 - 遮罩數量:如「最佳做法」指南中所示，使用遮罩會影響性能。
 - 篩檢程式數量:如「最佳做法」指南中所示，使用篩檢程式會影響性能。
 - DirectAccess API 呼叫數量:Direct Access API 呼叫太多會影響性能。
 - 點陣化字形數量:字形數量太多會極大地佔用記憶體空間。
 - 字體紋理數量:有些平臺可能有不止一個字體紋理。

- 字體紋理更新次數:每個幀創建新字體紋理的次數。這會影響性能。
- 字體緩存抖動:這是每個幀從緩存中去除一個字形以為一個新字形騰出空間來的時間數量。
理想情況下,此數位為零,即字體緩存足夠大,可以容納所有光柵。
- 未用字體紋理百分比:百分比小表明給不用的字體緩存留有空間。
- 已用字體緩存面積:不同的字體和樣式太多可能導致佔用過大記憶體空間。
- 字體緩存總面積:所有字體紋理的大小。字體緩存以消耗記憶體空間為代價來提高性能。
- 總記憶體:Scaleform 總記憶體佔用空間。
- MovieData 記憶體:這與導入的電影檔案的大小直接相關。
- 字體記憶體:通過讓文字欄位共用字體可以減小字體記憶體。
- MovieView 記憶體:用於 GFx::Movie 的記憶體,包含其時間表和實例資料。MovieView 記憶體受內部垃圾回收制約。
- ActionScript 記憶體:ActionScript 內容太多會導致佔用過大記憶體空間。
- 網格緩存記憶體:網格緩存提高性能,但會極大地消耗記憶體空間。
- 未用網格緩存記憶體:未用的網格緩存會佔用大量記憶體,而不會提高性能。
- 圖像記憶體:用於圖像的系統和圖形卡記憶體。
- 視頻記憶體:這是用於 Scaleform 視頻的記憶體,而不是 GPU 記憶體。
- 聲音記憶體:用於聲音的記憶體。
- 超出一定大小的圖像的數量:使用者可以設置圖像閾值,而 AMP 報告超出該閾值的圖像數量。

為了給所有上述閾值設置值,AMP 讀取一個位於其工作目錄的 XML 檔,稱為「`adviser.xml`」。如果 AMP 找不到這樣的檔,就把所有閾值設置為零。通過在「性能顧問」(Performance Advisor)選項卡上按一下「設置」(Settings) 按鈕,使用者可以修改任何閾值。這就會顯示設定對話方塊：



按一下「確定」(OK)，就把所修改的設置保存到一個 XML 檔，下次執行時 AMP 會讀取該檔。使用者可以為不同的內容類型創建不同的顧問設置，並通過從設置對話方塊進行載入來選擇要使用的那個設置。通過一個命令列參數 「 - adviser <AdviserFilePath> 」來啟動 AMP，可使程式讀取一個除位於工作目錄的那個預設檔之外的任何顧問檔</AdviserFilePath>。

- “SWF 資訊” (SWF info) 選項卡:此選項卡提供有關正在分析的所有 flash 檔的一些靜態資訊。這些資訊包括檔案名、flash 版本、電影尺寸以及電影畫面播放速率。
- “渲染器” (Renderer) 選項卡:此選項卡顯示與該圖表相同的資訊，但是採用數位形式。它還包括一些其它渲染統計資料，例如:網格和字體緩存抖動(在幀期間逐出的專案的數量)、字體填寫率(所有字形的面積占字體緩存紋理總面積的百分比)、字體緩存失敗率(幀期間有多少字形未能被分配，且被向量形狀所取代)、幀期間字體緩存紋理更新次數以及當前字體紋理數量。
- “記憶體” (Memory) 選項卡:此選項卡使使用者可以詳細確定 Scaleform 足跡、Scaleform 要求的記憶體有多少實際在用，以及如何按用途分配記憶體。當以完整記憶體報告模式(該模式通過AMP工具列上的“i”按鈕進行切換)分析時，記憶體報告變得更加詳細，顯示按堆、按檔和按用途分的明細資料。

- “圖像” (Images) 選項卡:顯示每個圖像及其尺寸和格式所佔用的記憶體。點擊清單中的每個專案可在預覽窗格中顯示該圖像的一個縮略圖。請注意,圖像資料不是作為分析資訊的組成部分發送,而是根據要求進行發送。因此,可能的情況是,假如 Scaleform 已經卸載該圖像,就不會顯示任何圖像。AMP 目前並非支援預覽所有圖像格式,例如,DXT 壓縮圖像。
- “字體” (Fonts) 選項卡:顯示字體緩存佔用的全部記憶體以及每個 SWF 的字體清單。當選中該選項卡中的每個專案時,就會顯示代表當前字體緩存的一個圖像,具體方式與“圖像”選項卡中的圖像預覽方式相似。
- CPU 選項卡:將“推進” (Advance)、 “顯示” (Display) 和“使用者” (User) 時間細分為子類別,以便於更加詳細地考察針對所選幀的時間花費在了什麼地方。假如選中多個幀,則值就會是那些幀之間的平均值。
- “函數” (Functions) 選項卡:顯示一個樹狀控制項,其中包含每個函數內花費的時間數量,這些函數也包含從該函式呼叫的函數。當選中多個幀時,調用時間和數量就是每個遊戲幀的平均值。
ActionScript 和 C++ 函數混合在同一個調用圖表中,因為代碼路徑可以在 ActionScript VM 及該應用程式的其餘部分之間移動。擴展每個函數就會顯示從該函式呼叫的函數。
- ActionScript 選項卡:此選項卡顯示與“函數” (Functions) 選項卡相似的資訊,但資訊不是分層的,不包括 C++ 函數,而且為每個函數花費的時間不包括花費在清單中其它函數中的時間。此選項卡便於快速識別最昂貴的腳本函數。
- “源” (Source) 窗格:此視窗是“函數”選項卡和 ActionScript 選項卡的組成部分。對於 AS3,始終為 AS 檔中駐留的代碼啟用按行計時。對於嵌入 SWF 中的 AS3 代碼,不顯示原始程式碼。對於 AS2,要查看每行計時情況(默認情況下,不顯示),需要從工具欄上的下拉控件中選擇“高”(high) 級函數分析。AS2 中的指令分析非常耗時,而且會減緩所分析的 Scaleform 應用程序的運行速度,最好在已經確定問題所在後再進行指令分析。倘若不止一個其他函數調用某個函數,緊接每行顯示的計時數據不會加到該函數調用圖表中報告的時間數量內。這是因為每行計時數據是花費在該函數中的總時間,而不管該函數是從哪裡調用的。此外,行計時數據不包括函數調用內所花費的時間,而調用圖表計時數據則包括被調用的函數中所花費的時間。要顯示源代碼和每行計時數據,AMP 服務器 (SWF 位置) 上找不到的任何 Flash 調試文件都需要位於 AMP 客戶端可執行文件的工作目錄之中,或者,如果是 AS3 文件,則位於已加載的 SWF 的相同相對路徑中。
- **檔(File) 功能表**：此功能表包括下列選項：
 - **連接(Connection)**：引出連接對話方塊,便於連接到一個不同的伺服器。
 - **斷開連接**：終止當前連接。
 - **調試資訊路徑 (Debug Info Paths)**:調出一個對話方塊,列出檔案系統中的所有路徑,其中,AMP 搜索 SWD 檔(針對 AS2)或 AS 檔(針對 AS3)。可以分別使用“添加” (Add) 和“刪除” (Remove) 按鈕來添加或刪除路徑。
 - **載入概要幀 (Load Profile Frames)**：載入要重新檢查的以前保存的概要運行。選擇此菜單項可引出一個文件加載對話框,該對話框用來定位概要數據文件。

- **保存概要幀(Save Profile Frames)**：將當前概要運行保存到磁片。將當前概要運行保存到磁盤。此運行可通過選擇上面的“加載概要幀”選項來重新加載。選擇此菜單項可引出一個文件保存對話框，該對話框用來選擇包含概要數據的文件的路徑和名稱。
- **轉儲記憶體報告 (Dump Memory Report)**:將當前設定檔幀保存到 AMP 的工作目錄中的一個名為 “AmpMemReport.txt” 的檔中。
- **退出(Exit)**：關閉應用程式。

- **視圖” (View) 功能表**:此功能表包括下列選擇:
 - 圖表工具提示 (Graph Tooltips):選擇此功能表,當把滑鼠放在 AMP 中的圖表上時,可切換上下文敏感的彈出式說明。
 - 時間單位 (Time Units):允許使用者選擇用於報告所有時間數量的單位,例如,“顯示”時間和“推進”時間。選擇可以是毫秒,也可以是微秒。
 - 記憶體單位 (Memory Units):允許使用者選擇“記憶體”選項卡和“記憶體”圖表的單位。選擇可以是位元組,可以是千位元組,也可以是百萬位元組。
- **“視窗” (Window) 功能表**:此功能表允許使用者顯示和隱藏上面所述的任何選項卡(日誌、SWF 資訊、渲染器、記憶體、圖像、字體、CPU、函數、ActionScript)。它還以一項 Restore UI(恢復 UI)選擇為特色,該選項可以取消使用者以前對 AMP 應用程式所做的任何視窗自訂。
- **“帮助”(Help) 菜单**: 此菜单目前有两个选项：“用户指南”(User Guide) -- 链接到 Scaleform.com 开发者中心中的此文档, 以及“关于”(About) -- 调出一个包含 AMP 版本和积分的屏幕。

- **應用程式控制工具列**：此 UI 元素包含一系列按鈕,它們用來控制所分析的應用程式。
 -  **線框 (Wireframe)**:向被分析的應用程式發出一個請求以便於以線框模式渲染內容。當應用程式已處於線框模式時,按一下此按鈕可使其退出此模式。
 -  **透支模式**:向被分析的應用程式發出一個請求,以進入一種渲染模式,該模式突出圖元渲染無效、遮罩和篩檢程式區域。AMP 伺服器用綠色渲染圖元渲染無效區域(顏色強度與渲染無效次數成比例),以紅色渲染遮罩,以藍色渲染篩檢程式。當應用程序已經處於透支模式時,單擊此按鈕可使其退出此模式。
 -  **批量分析模式**：向所分析的應用程序發送一個請求以進入某種渲染模式，該渲染模式以不同顏色繪製每個渲染批次。當應用程序已經處於批量分析模式時，單擊該按鈕可使其退出此模式。
 -  **防混疊 (Anti-aliasing)**:向被分析應用程式發出一個請求,以迴圈通過防混疊模式。三個可用模式為 Scaleform 的邊緣防混疊技術 (EdgeAA)、硬體全景防混疊 (HW FSAA) 或不防混疊 (None)。

-  **快進 (Fast-forward)**: 向被分析的應用程式發出一個請求,以便以快進方式播放 Flash 內容。這使 SWF 就像渲染幀一樣快速前進,而不管其文檔 FPS 設置如何。當該應用程式已經處於快進模式時,按一下此按鈕可使其退出此模式。
 -  **重啟 (Restart)**: 向被分析的應用程式發出一個重啟 Flash 內容的請求。
- “幀控制” (Frame control) 工具列:此工具列包含下列按鈕。
 -  **清除 (Clear)**: 放棄當前概要運行。丟棄分析幀,由此為新資料釋放 AMP 用戶端記憶體。
詳細分析(源行計時及詳細記憶體明細): 選定此選項時,AMP 收集每個幀的更詳細的內存統計數據,這些數據顯示在“內存”(Memory) 選項卡中。此選項還使 AMP 顯示“圖像”(Image) 選項卡中的完整圖像列表,而不是最多十個圖像。
 -  **第一個幀 (First Frame)**: 選定當前概要運行中的第一個幀,由此更新底部面板選項卡上顯示的資訊。
 -  **前一個幀 (Previous Frame)**: 從當前選定的一個幀選擇前一個幀,由此更新底部面板選項卡上顯示的資訊。
 -  **暫停 (Pause)**: 向被分析的應用程式發出一個停止發送記憶體和性能資料的請求。此操作不會暫停應用程式本身。
 -  **下一個幀 (Next Frame)**: 從當前選定的一個幀選擇下一個幀,由此更新底部面板選項卡上顯示的資訊。
 -  **下一個幀 (Last Frame)**: 選擇並鎖定到當前概要運行中的下一個幀上。接收到新幀的資訊時,選擇更改為始終是最後一個收到的幀,並連續更新底部面板選項卡上顯示的資訊。
- “分析” (Profiling) 工具列:此元素包含用於進行分析的一系列按鈕。
 -  **詳細分析 (Detailed Profiling)**(詳細的記憶體明細):選中此選項時,AMP 收集每個幀的更加詳細的記憶體統計資料,這些統計資料顯示在“記憶體”選項卡中。此選項還使 AMP 顯示“圖像”選項卡中的圖像的完整清單,而不是最多十個圖像。
 -  **“縮放” (Zoom)** 工具列:包含一個下拉式控制項,用來更改圖表的水準縮放比例。放大 (Zooming out) 功能允許查明該運行的問題領域,而縮小 (Zooming In) 功能則允許更加精確地檢查這些領域。也可以使用滑鼠滑輪進行縮小和放大,並提供通過下拉方式所達不到的較高解析度。此工具列還顯示當前選定的幀數。
- “狀態” (Status) 欄:分析器底部有一個“狀態”欄,它顯示當前連接以及通過網路接收資訊的位元速率。

AMP 允許使用者自訂其外觀。可以通過將工具列和選項卡拖動到其新位置來固定它們的位置。可以從應用程式取消固定這些選項卡和工具列。圖表和選項卡的尺寸可以調節，而且圖表可以部分或全部折疊。這些 UI 更改的狀態在 **AMP** 分析會話過程中都是不變的，但使用者可以從“視窗”功能表中的“重置 UI”(Reset UI) 選項中將幾何圖形重置回到其預設狀態。

2 AMP 入門

要開始使用 AMP 分析器,請運行 Scaleform 主安裝目錄下 Bin/AMP 中的 GFxAmpClient.exe 可執行檔。應用程式啟動,並出現連接對話方塊。

2.1 進行連接

用戶端啟動時會自動出現一個連接對話方塊。它提示使用者要麼從一個發現的 AMP 伺服器清單中進行選擇,要麼手動鍵入要連接的 IP 位址和埠。如果相關 AMP 伺服器尚未運行,就選第二種方法,就像需要分析啟動中的應用程式時的情況一樣。

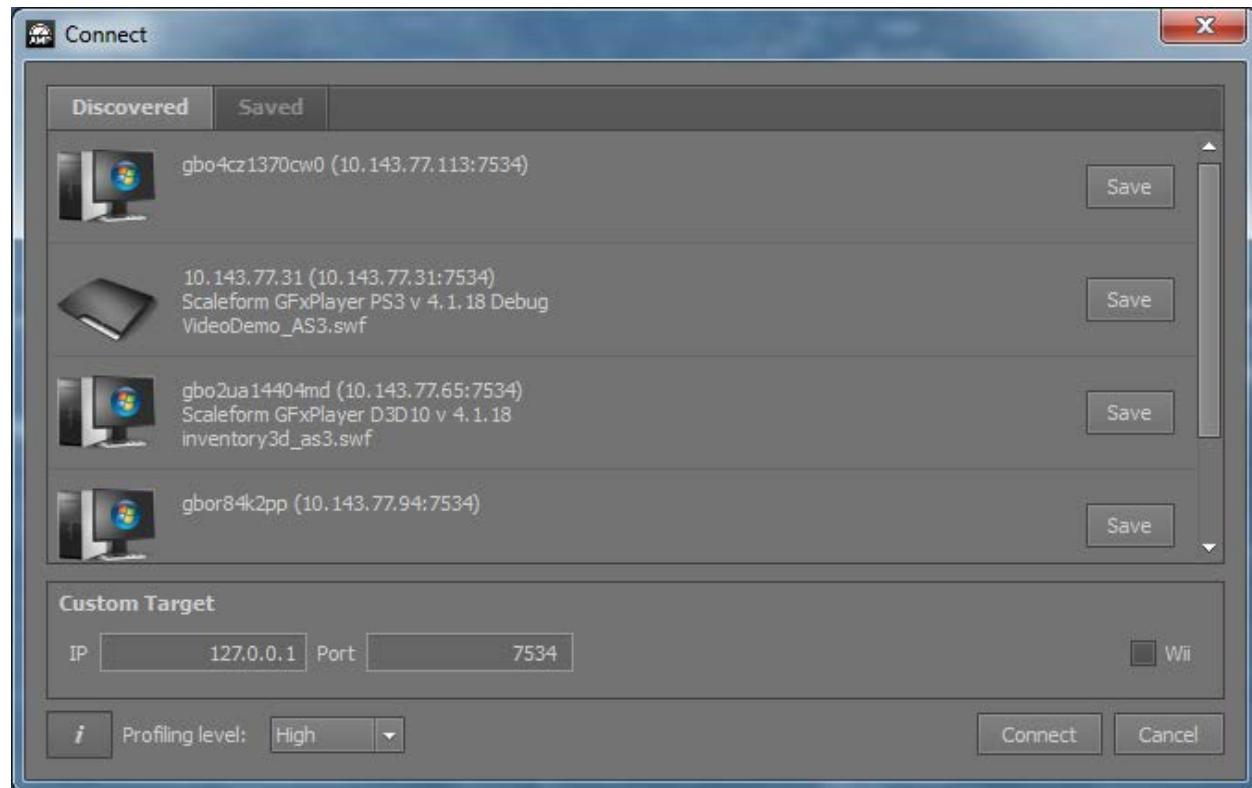


圖 2:AMP 連接對話方塊

“檔”(File) 功能表下的“連接”(Connection) 項隨時可用來引出連接對話方塊。發現的清單中的每個 AMP 伺服器都有一個醒目的圖示,用以識別其平臺(PC、Xbox360、PS3 等)。緊接該圖示的是應用程式的標題、伺服器正在偵聽的 IP 位址和埠以及目前播放的 Flash 內容。如果其中某些資訊丟失,這是因為相應的 AMP 伺服器不在發送這些資訊。本文檔後面部分將會提供一個如何向任何 Scaleform 應用程式添加完全 AMP 支

援,包括廣播應用程式和內容資訊。通過按兩下一個發現的伺服器或選擇一個伺服器並按一下“連接”(Connect) 按鈕來建立連接。

從連接對話方塊,可以選擇連接上的分析等級以及詳細記憶體報告,這樣就可以保證立即在連接上設置這些專案。當正在分析某個 Scaleform 應用程式的啟動情況時,這尤其重要。

請注意,與 Wii 開發工具箱 (NDEV) 的連接不使用上面的伺服器探索方法。相反,使用內置到 NDEV 中的 USB 介面(COM 埠)將 Wii 從物理上連接到 PC。通過在 IP 位址欄位中指定 “wii” 來從連接對話方塊連接到 Wii AMP 伺服器。

2.2 分析性能指標

一旦建立連接,分析資訊就開始流入用戶端,而圖表就開始獲得填充。最上面的圖表顯示每個幀在渲染 Scaleform (Display) 和執行 ActionScript (Advance) 以及在 Direct Access API (User) 中所花費的 CPU 時間數量。



圖 1：CPU 圖表

如果在分析期間 CPU 圖表在某個點沒有任何意義,則可以將該圖表最小化,或者重新調整大小,以便為其它 UI 元素留出更多空間。如果對某個圖解數量不感興趣,可以通過點擊其圖例旁邊的圖示來將其刪除。

與 CPU 圖表相關的是分析器底部的「CPU 摘要」(CPU Summary) 選項卡、「函數」(Functions) 選項卡以及 ActionScript 選項卡。“CPU 摘要”選項卡顯示更詳細的圖表資訊,“函數”選項卡顯示選定幀的調用圖表,而 ActionScript 選項卡顯示選定幀期間調用的所有腳本函數的非分層視圖。

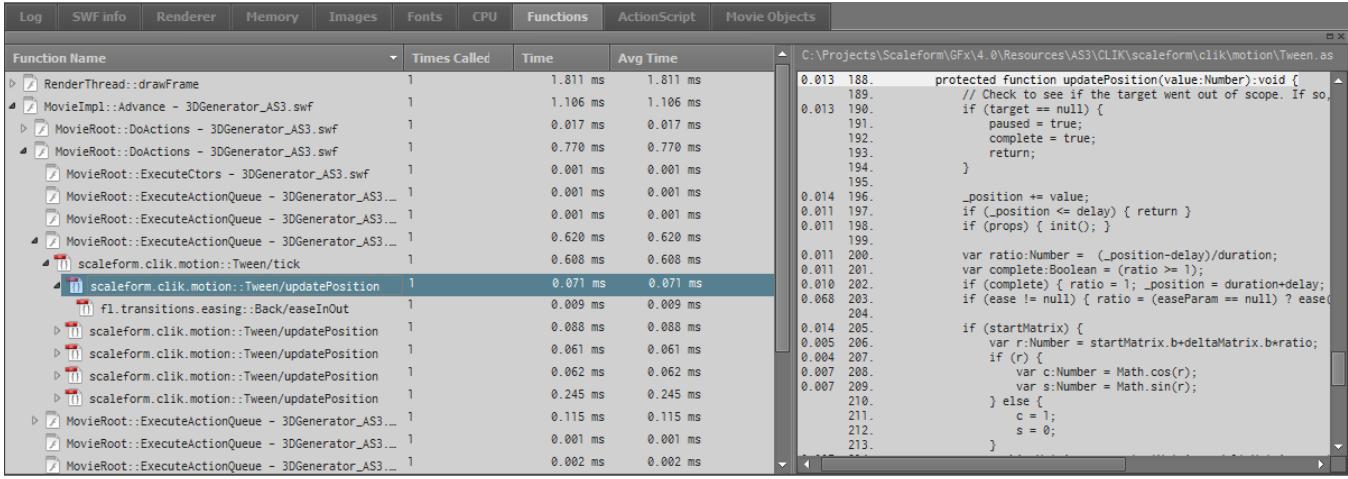


圖 2：CPU 摘要

通過檢查 CPU 詳細資訊選項卡在 CPU 圖表中找到某個有峰值的幀，就可以確定腳本性能瓶頸的原因。另一方面，通過檢查渲染圖表，可以更容易地查明渲染性能瓶頸。

2.3 分析渲染指標

渲染圖表顯示繪畫單元數目、三角形數目、線條數目、使用過的遮罩數目、筆劃數目、梯度填充數目、網格緩存中的字形數目以及針對選定幀進行的渲染期間發生的更新次數。[最佳做法指南\(Best Practices Guide\)](#) 詳細說明最大限度地減少上述數量對於性能的重要意義。

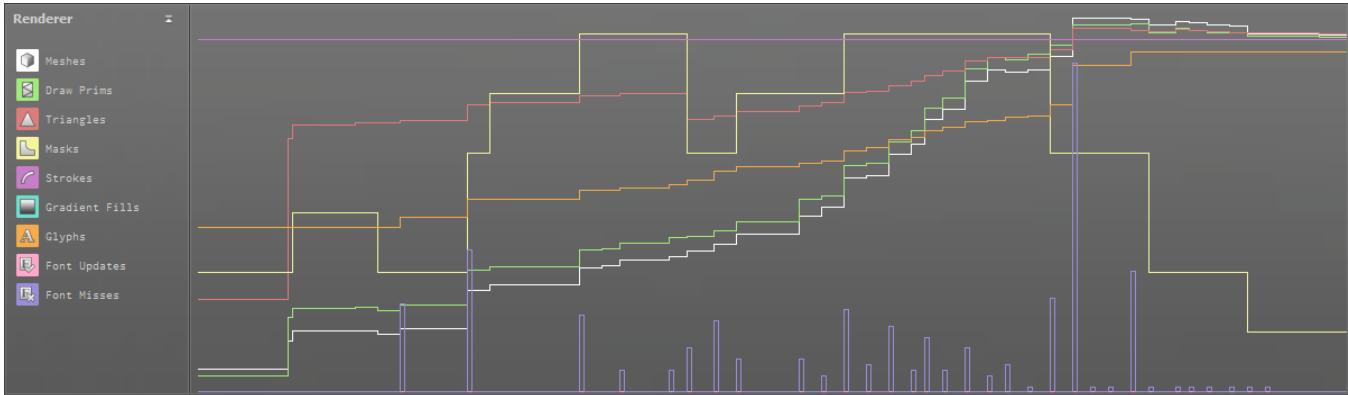


圖 3：渲染圖表

與渲染圖表相關的是應用程式底部的「渲染摘要」(Render Summary) 選項卡。

Meshes:	182	
Draw Primitives:	183	
Triangles:	5124	
Masks:	6	
Filters:	0	
Strokes:	21	
Gradient Fills:	0	
Rasterized Glyphs:	412	
Font Cache Updates:	0	
Font Textures:	1	
Font Cache Thrashing:	0	
Font Cache Fill:	28%	
Font Cache:	1048576 pixels	
Font Cache Used:	298294 pixels	
Font Failures:	0	
Font Cache Misses:	5	
Mesh Thrashing:	0	

圖 4：渲染摘要

2.4 分析記憶體消耗

從頂部數的第三和第四個圖表分別顯示 Scaleform 系統和圖形記憶體使用方式：

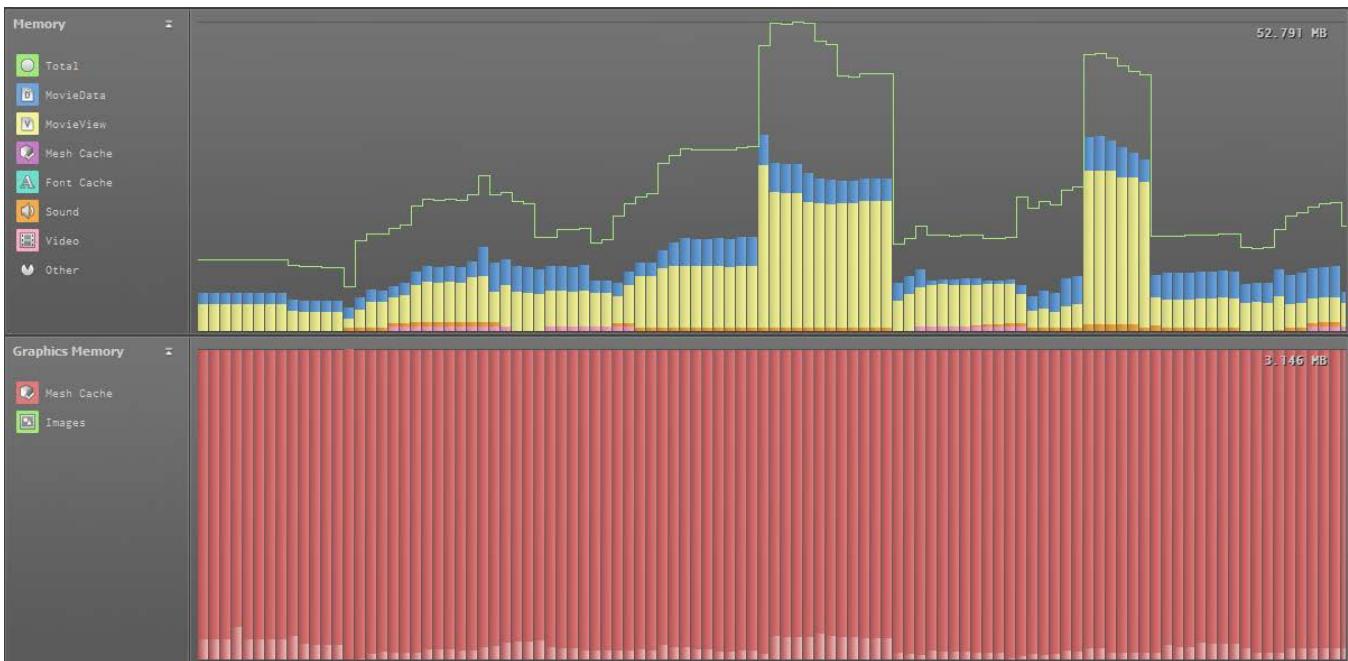


圖 5：記憶體圖表

- **總計(Total)**：Scaleform 要求的全部記憶體(足跡)。總記憶體畫為一個線形圖,包括下列全部類別,以及由於系統開銷、被請求的分配的細微性以及磁碟重組而未使用的記憶體。請注意,由於任何類別中都不顯示未使用的記憶體,因而總記憶體大於使用的記憶體類別的總和。
- **電影資料 (MovieData)**:所有載入的 SWF 或 GFX 檔所佔用的記憶體。複雜向量圖形物件、嵌入式字體以及時間線動畫會影響這一數位。
- **電影視圖 (Movie view)**:GFX::Movie 實例佔用的運行時記憶體,分配給時間線動畫支援、螢幕物件和 ActionScript。
- **網格緩存 (MeshCache)**:為緩存的形狀網格資料佔用的記憶體,這些資料是由向量拼嵌器 (Tesselator) 和邊緣防混疊功能生成的。可以從系統或圖形記憶體分配網格緩存記憶體,具體取決於平臺。實際使用的記憶體以較暗的陰影顯示,因此,使用者可以容易地確定是否可以設置一個較小的網格緩存。
- **字體緩存 (Font Cache)**:字體緩存使用的記憶體。
- **聲音 (Sound)**:用於嵌入式聲音資料的記憶體。它受嵌入的聲音示例的數量、長度或品質影響。
- **視頻 (Video)**:用於視頻記憶體重播暫存器的記憶體。這些暫存器是在視頻啟動時分配的,然後在其停止時釋放。
- **其它 (Other)**:由 Scaleform 使用但並非上述類別組成部分的記憶體。有關記憶體類別的更加詳細的細分資訊,請切換下文介紹的詳細記憶體報告。
- **圖像 (Images)**:用於圖像(紋理)的圖形記憶體。

有關記憶體系統及其配置、優化和管理方式的更多資訊,請參閱[記憶體系統概述 \(Memory System Overview\)](#) 文檔。

“記憶體摘要”選項卡更詳細地顯示上述記憶體的類別。“記憶體”(Memory)選項卡中的所有值的單位都相同(位元組、千位元組或百萬位元組),與在視圖功能表下的記憶體單位選擇中指定的相同。。請注意,“Used Space”(已用空間)是 GFX 有效使用的記憶體。“Unused Space”(未用空間)是 GFX 已經保留但主要由於碎片而尚未使用的記憶體。Debug Data(調試資料)表示 GFX 為跟蹤記憶體和性能統計資料而在內部使用的記憶體。

當以完整記憶體報告模式(該模式通過 AMP 工具列上的「i」按鈕進行切換)分析時,記憶體報告變得更加詳細,顯示按堆、按檔和按用途分的詳細資料。Scaleform 庫的調試版本將堆分配分割成若干記憶體類別,使用者可以由此更好地理解 Scaleform 的當前記憶體使用方式。下面介紹報告的記憶體類別：

- **Renderer (渲染器)** : Scaleform 渲染引擎分配的記憶體。此類別進一步細分為下列子類別：
 - **Buffers (緩衝區)** : 為網格暫存緩衝區 (Mesh Staging Buffer) 分配的記憶體,該緩衝區充當一個短期緩存,用來在將生成的網格具現化並複製到頂點和索引緩衝區之前存儲這些網格。
 - **RenderBatch (渲染批次)** : 為實體(例如:網格、圖元,或與渲染樹中即將繪製在一起的節點關聯的渲染命令)的集合分配的記憶體。
 - **Primitive (圖元)** : 用來保存在一起繪製的頂點的集合的記憶體。
 - **Fill (填充)** : 用來存儲和管理描述硬體填充的資料的記憶體。

- **Mesh** (网格)：給網格緩存分配的記憶體。網格緩存用來饋給包含頂點和索引緩衝區資料的渲染器。
- **MeshBatch** (网格批次)：給「網格緩存」的頂點和索引緩衝區中緩存的各個資料塊分配的記憶體。
- **Context** (上下文)：渲染支援系統佔用的記憶體，該系統用來管理渲染樹的壽命。此上下文維護渲染樹的多個邏輯上獨立的快照，因而執行緒可作用於渲染資料，而不會彼此干擾。
- **NodeData** (节点数据)：用於幀狀態資訊(如應用到所渲染的物件的當前矩陣或效果)的記憶體。
- **TreeCache** (树缓存)：給渲染器的場景圖表(所渲染的物件的一個樹狀結構)分配的記憶體。
- **TextureManager** (纹理管理器)：給管理紋理圖像分配的記憶體。實際圖像資料通常不存儲在系統記憶體中，因此，不是此記憶體類別的組成部分。
- **MatrixPool** (矩阵池)：渲染器所維護的矩陣池佔用的記憶體。
- **MatrixPoolHandles** (矩阵池句柄)：矩陣控制碼分配所佔用的記憶體。
- **Text** (文本)：給文本管理和渲染(如格式化資訊、樣式和顏色資訊、html 解析、富文本資料和行緩衝區)分配的記憶體。
- **Font** (字体)：與字體渲染(如字形點陣化和字體緩存存儲)相關的記憶體。
- **MovieDef** (电影定义)：給代表電影元素範本的不可變數據分配的記憶體。它包含下列子類別：
 - **CharDefs** (字符定义)：用於字元定義(如按鈕和文本)的記憶體。
 - **ShapeData** (形状数据)：用於形狀定義的記憶體。
 - **Tags** (标签)：用於 ActionScript 標籤的記憶體
 - **Fonts** (字体)：用於字體資源的記憶體。
 - **Sounds** (声音)：用於聲音資料的記憶體。
 - **ASBinaryData** (**ActionScript** 二进制数据)：ActionScript 緩衝區資料佔用的記憶體。
 - **MD_Other** (其它电影定义数据)：上述類別中不包含的其它 MovieDef 資料。
- **MovieView** (电影视图)：給時間表動畫支援、螢幕上物件以及 ActionScript 分配的記憶體。該類別進一步細分為下列子類別：
 - **MovieClip** (电影剪辑)：用於螢幕上動畫物件的記憶體。
 - **ActionScript** : **ActionScript** 代碼和資料佔用的記憶體。
 - **ASString** (**ActionScript** 字串)：ActionScript 字串佔用的記憶體。
 - **Text** (文本)：用於與文本相關的顯示物件佔用的記憶體。
 - **XML**：用於 XML 物件的記憶體。
 - **MV_Other** (其它电影视图)：不符合上述類別的其它電影視圖記憶體。
 - **VM** (虚拟机)：分配給 ActionScript 虛擬機器的記憶體。此類別進一步細分如下：
 - **AS3 VM** (**AS3** 虚拟机)：ActionScript 3 虛擬機器佔用的但不屬於下列子類別組成部分的記憶體。

- **CallFrame**（调用帧）：執行 ActionScript 期間給調用棧中每個實體分配的記憶體。
 - **Vtable**（虛擬表）：ActionScript 虛擬表佔用的記憶體。
 - **SlotInfo**（插槽信息）：用於描述 ActionScript 類成員的結構的記憶體。每個 ClassTraits 和 InstanceTraits 都包含 SlotInfo 結構，以便於描述「類」(Class) 或「實例」(Instance) 記憶體佈局。有關 ClassTraits 和 InstanceTraits 的更多資訊，請參閱下文。
 - **SlotInfoHash**（插槽信息哈希表）：加快 ActionScript 的屬性查找的雜湊表 (Hash Table) 使用的記憶體。
 - **ClassTraits**（类特征）：給保存 ActionScript 類的記憶體佈局分配的記憶體。
 - **Class**（类）：給保存 ActionScript 物件的靜態成員分配的記憶體。
 - **InstanceTraits**（实例特征）：給保存 ActionScript 物件的非靜態記憶體佈局分配的記憶體。
 - **Instance**（实例）：給保存 ActionScript 物件的執行個體變數和方法分配的記憶體。
 - **AbcFile (ABC 文件)**：給代表位元組代碼及其相應資料的資料結構分配的記憶體。
 - **AbcConstPool (ABC 常数池)**：用於存儲常數(如數位、字串、命名空間和 Multiname)的記憶體。這只是位元組代碼，因此它是上面的 AbcFile 的一部分。
 - **VMAbcFile (虚拟机 ABC 文件)**：保存內部資料結構的記憶體，而內部資料結構存儲從 AbcFile 生成的優化位元組代碼、依存關係以及其它資訊。
 - **Tracer (跟踪器)**：給代碼優化分配的記憶體。
- **IME (输入法编辑器)**：用於支援輸入法編輯器 (IME) 的記憶體。

Log	SWF info	Renderer	Memory	Image
Memory		Size		
▲ Total Footprint			4849.402 KB	
▲ Used Space			4633.715 KB	
▷ Video Heaps			0.000 KB	
▷ Other Heaps			325.621 KB	
▲ Movie View Heaps			1226.094 KB	
▲ MovieView "3dgenerator_as3.swf"			1226.094 KB	
String			0.828 KB	
▷ Renderer			12.672 KB	
▷ MovieView			1176.781 KB	
General			35.812 KB	
▷ Movie Data Heaps			261.066 KB	
▲ Global Heap			2820.934 KB	
Video			264.176 KB	
String			20.633 KB	
Sound			0.141 KB	
▷ Renderer			2310.914 KB	
▷ MovieView			8.039 KB	
▷ MovieDef			116.676 KB	
Image			0.141 KB	
IME			9.914 KB	
General			90.301 KB	
▷ Debug Memory			8.000 KB	
▷ Unused Space			38.820 KB	
Debug Data			176.867 KB	

圖 6：記憶體摘要

展開每個類別會顯示負責該類別直接分配的子類別或記憶體堆(如果有的話)。以這種方式深入探討,就會更好地瞭解任何特定幀中的 Scaleform 記憶體使用情況。

“圖像”(Images) 選項卡還與記憶體分析相關,因為它顯示在選定的幀內所載入的圖像,以及其大小。

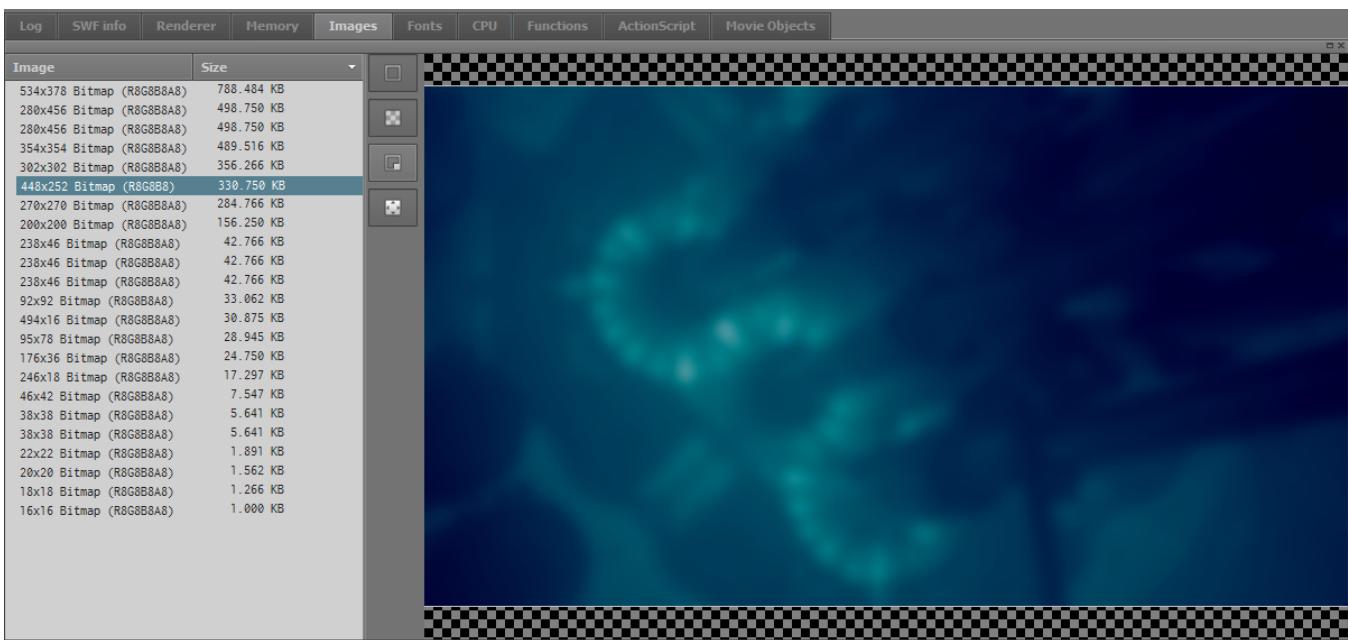


圖 9：“圖像記憶體”選項卡

使用此選項卡可快速檢查是否存在增加目標 Scaleform 應用程序內存消耗的大圖像。在圖像列表中單擊一個項目，可在預覽窗格中顯示該圖像的縮略圖。請注意，不是作為分析信息的組成部分來發送圖像數據，而是在請求時發送。因此，如果連接的 Scaleform 應用程序已經卸載了該圖像，就可能不會顯示任何圖像。

與記憶體分析相關的最後一個選項卡是「字體」(Fonts) 選項卡，該選項卡顯示選定的幀的字體資訊。

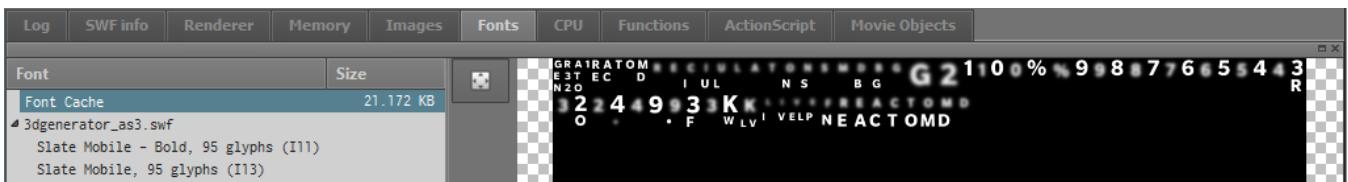


圖 10：“字體記憶體”選項卡

使用此選項卡檢查每個字體類型的嵌入式字形的數量(按每個 SWF 檔顯示)。

嵌入式字體可消耗相當大的記憶體，這會在 AMP 中按照 MovieData 堆和 CharDefs 類別的增加的大小顯示出來。這些可分別在“堆”選項卡和“記憶體摘要”選項卡中進行檢查。可通過在多個 SWF 檔上共用字體來減小字體記憶體。

2.5 使用透支分析

AMP 可用來遠端觸發一種特殊的渲染模式、這種模式可突出（突出顯示）被分析的應用程式中的潛在問題領域。請按一下應用程式控制工具列中的適當按鈕要切換渲染模式。

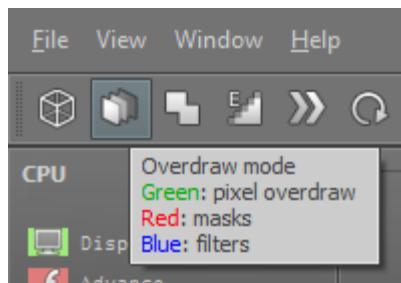


圖 11：切換按鈕

一旦切換此模式，就會以綠色渲染所有物件。綠色表示無效渲染(16 層達到全亮)，遮罩為紅色，篩檢程式為藍色(8 層遮罩或篩檢程式達到全亮)。這些顏色混合在一起，因而一個明亮的黃色區域由於帶有嵌入的遮罩而嚴重渲染無效。

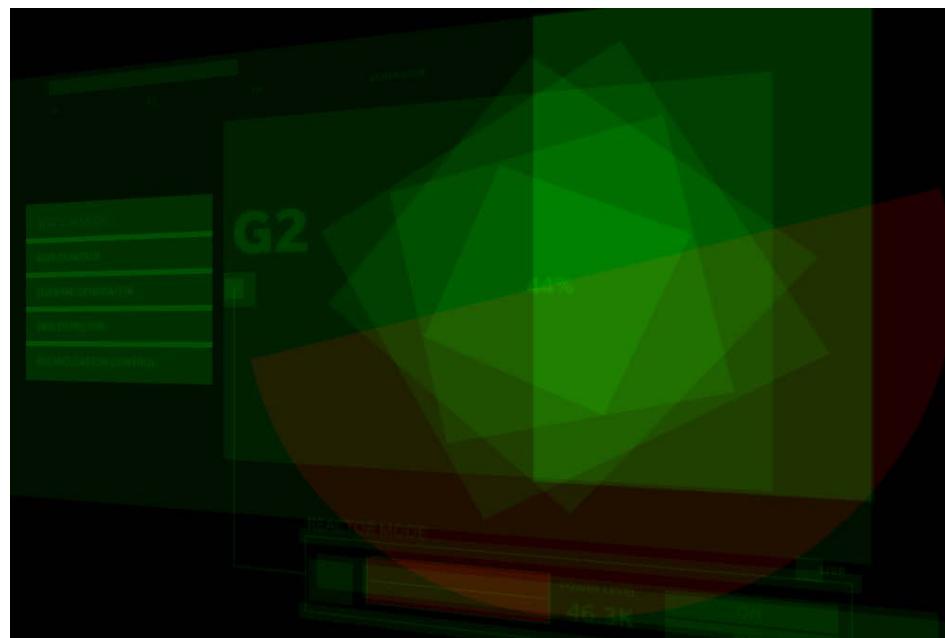
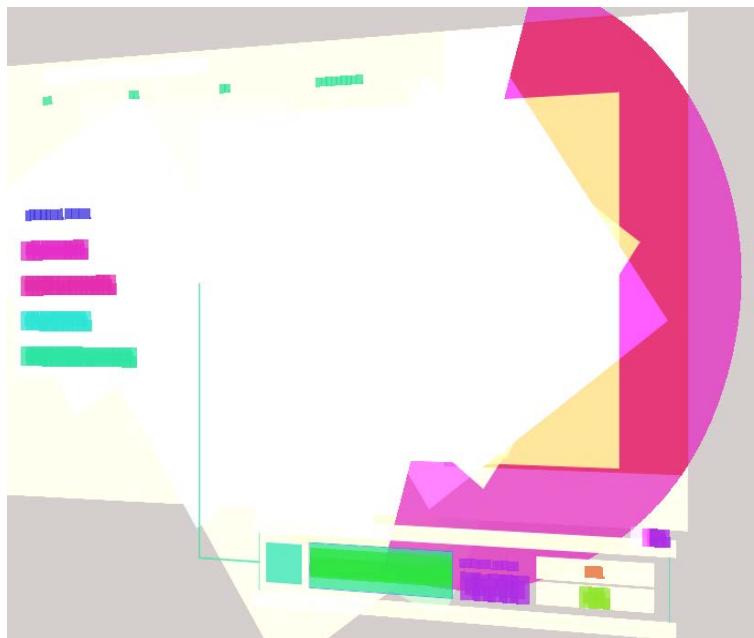


圖 12：用於性能突出的渲染模式

無效渲染、遮罩和篩檢程式都會影響性能，應盡可能減小這種影響。

2.6 使用批量分析

支持的另一個特殊渲染模式是批量分析。要切換此渲染模式，請單擊位於應用程序控制工具欄上的適當按鈕。



一旦切換此模式，就會以一種不同的顏色渲染每個渲染批次。如有可能，通過最大限度地減少批次來實現最佳性能。

3 平臺和集成說明

1. AMP 通常是在 GFx::System 物件構造期間或在 GFx::System::Init 函式呼叫期間初始化。或者,假如一個 Saleform::System 用來初始化 Scaleform,程式師就需要通過分別調用 GFx::AMP::Server::Init() 和 GFx::AMP::Server::Uninit() 來明確初始化和關閉 AMP。
1. AMP 伺服器目前支援 Windows、Xbox360、PS3、Linux、Mac, Wii, Android 和 iPhone/iPad, 3DS, PS Vita, WiiU。使用 AMP 用戶端的自動發現功能,連接非常簡單。不過,不經自動發現的連接必需知道 AMP 伺服器的 IP 位址,這會導致混亂,因為某些平臺擁有多個 IP 位址。
 - 對於 Xbox360,請務必指定標題 IP 位址,而非調試 IP 位址。標題 IP 位址可通過查看 Xbox 360 包含的開發或測試工具箱的屬性來確定。
 - 對於 PS3,如果開發工具箱配置為使用兩個 IP 位址,請務必指定應用程式 IP 位址,而非用於調試的 IP 位址。
 - 對於 Android,需要在應用程序級啟用互聯網許可。為此,需要把下面的代碼添加到應用程序的 AndroidManifest.xml :

```
<uses-permission android:name="android.permission.INTERNET" />
```
 - 請通過主機 PC 橋(而不是 WiFi 連接)來連接到 WiiU。主機 PC 通過自動發現埠 6003 來檢測遊戲機。一旦進行連接請求,就將通信從 TCP/IP 插槽切換到主機橋。
 - ° Wii 不使用網路套介面與 AMP 進行通信,而是使用內置到 NDEV 中的 USB 介面 (COM 埠) 來把主控台物理上連接到一台 PC。使用者可以通過在 IP 位址欄位中指定"wii"來從連接對話方塊連接到 Wii AMP 伺服器。重要提示:使用者必須確定 RVL_SDK/X86/bin 在環境變數裡。這是為確保 AMP 能運行 hio2.dll。
2. 某些遊戲引擎(如 Unreal Engine 3)採用一種特定的方式打包資源,使得 AMP 伺服器無法查找與所分析的 Flash 內容相對應的 AS2 SWD/AS3 AS 檔。倘若如此,請務必將 SWD 檔放在 AMP 用戶端工作目錄中,而不是依靠 AMP 伺服器在與 SWF/AS 相同的目錄中找到它們並通過網路將其發送出去。

4 構建注釋

1. 預設情況下，AMP 是所有非裝運 Scaleform 构建。要从 Scaleform 刪除 AMP，请确保没有在 GFxConfig.h 中定义 SF_AMP_SERVER，并重新构建 Scaleform 库。如果 Scaleform 源訪問不可用，可以通过调用 `Ptr<GFx::AMP::ServerState> serverState = *SF_NEW GFx::AMP::ServerState(); serverState->StateFlags &= GFx::AMP::Amp_Disabled; AmpServer::GetInstance().UpdateState(serverState);` 在運行時禁用 AMP。
2. AMP 使用網路套介面 (Network Socket) 與分析器進行通信。因此，請確保應用程式與適當針對平臺的網路庫相連結 (Ws2_32.lib 適用于 Windows, Xnet.lib 適用于 Xbox360, libnetctl 適用于 PS3, 套介面 適用于 Linux, 等等)。
3. 如果 AMP 只是使用網路套介面的系統，它就會初始化套介面庫，並在完成後發佈它。在某些系統上，如果初始化的次數大於發佈的次數，則發佈是無效的。即使出現這種情況，也沒有大礙。不過，在其他平臺上，發佈網路庫立即生效 --- 即使該庫初始化次數大於發佈次數，而且 AMP 可能會干擾應用程式其他部分中使用的套介面連接。倘若如此，可以強制 AMP 跳過套介面初始化，並通過調用 `AMP::Server::GetInstance().SetInitSocketLib(false)` 来使用以前初始化的套接口庫。此调用需要在初始化 AMP 后执行。
4. 可以針對 AMP 記憶體堆設置一個極限。當超過此極限時，就會暫停 Scaleform，直到將任何等候的消息發送到分析器。此極限是通過調用 `AMP::Server::GetInstance().SetHeapLimit()` 來設置的。
5. 預設情況下，AMP 伺服器將在埠 7534 上創建一個偵聽網路套介面。可以通過調用 `AMP::Server::GetInstance().SetListeningPort()` 來設置一個不同的埠。
6. AMP 可以配置為暫停執行，直到與分析器用戶端建立一個連接。這對於在啟動時獲得統計資料非常有用，而且是在啟動時通過進行以下調用來完成的：
`AMP::Server::GetInstance().SetConnectionWaitTime.`

5 添加 AMP 支援

按照本節所述的步驟,任何 Scaleform 應用程式都可以配置為通過 AMP 來支援遠端分析。另外,還可以檢查 Scaleform Player 源來查找到一個配置齊全的 AMP 伺服器實例。

5.1 Debug 檔生成

AMP 能夠使用 ActionScript 2 Flash 調試資訊(SWD 檔)來顯示原始程式碼和每行計時資料。通過在 Flash CS3 或 CS4 內運行偵錯工具 (Ctrl+Shift+Enter) 來生成 SWD 檔。請將 SWD 檔要麼放在與相應的 SWF 相同的位置,要麼放在 AMP 用戶端的工作目錄之中。

對於 ActionScript 3, 沒有 SWD 文件, 而且調試信息直接嵌入 SWF。需要使用 AS 文件顯示源碼。通過從 Flash Studio 內運行調試程序來生成調試 SWF 文件。

5.2 應用程式控制

AMP 能夠遠程控制被分析的應用程序中的設置,例如,線框模式或批量分析模式。許多支援的設置都是針對應用程式的,而且可以通過本應用程式加以實現。不實現的任何此類功能都只是不能通過 AMP 用戶端來使用。

請按照下述步驟處理發自 AMP 的應用程式控制消息：

- 實現一個從 `AMP::AppControlInterface` 派生的自訂應用程式控制回檔類,並改寫 `HandleAmpRequest` 方法來處理來自 AMP 的 `AMP::MessageAppControl` 消息。
- 通過在啟動時調用 `AMP::Server::GetInstance().SetAppControlCallback` 來安裝自訂處理常式。
- 通過調用 `AMP::Server::GetInstance().SetAppControlCaps` 來告知 AMP 支援的功能。此方法的參數是一個 `AMP::MessageAppControl` 消息,將支援的功能設置為 `True(真)`。例如：

```
AMP::MessageAppControl caps;
caps.SetCurveToleranceDown(true);
caps.SetCurveToleranceUp(true);
caps.SetNextFont(true);
```

```

    caps.SetRestartMovie(true);
    caps.SetToggleAaMode(true);
    caps.SetToggleAmpRecording(true);
    caps.SetToggleFastForward(true);
    caps.SetToggleInstructionProfile(true);
    caps.SetToggleOverdraw(true);
    caps.SetToggleBatch(true);
    caps.SetToggleStrokeType(true);
    caps.SetTogglePause(true);
    caps.SetToggleWireframe(true);
    AMP::Server::GetInstance().SetAppControlCaps(&caps);

```

5.3 連接狀態

AMP 用戶端能夠在狀態列中顯示連接的應用程式的名稱。此資訊是通過調用 `AMP::Server::GetInstance().SetConnectedApp` 來傳遞的。另外,AMP 用戶端還可以顯示當前應用程式狀態,但是為此,無論何時應用程式狀態發生變化,都需要告知 AMP 用戶端。使用下列方法來更新狀態

`: AMP::Server::GetInstance().SetState`, 將第一個參數作為下列 `ServerStateType` 枚舉類型之一：

- `Amp_RenderOverdraw`
- `Amp_App_Wireframe`
- `Amp_App_Paused`
- `Amp_App_FastForward`
- `AMP::Server::GetInstance().SetAaMode`
- `AMP::Server::GetInstance().SetStrokeType`
- `AMP::Server::GetInstance().SetCurrentLocale`
- `AMP::Server::GetInstance().SetCurveTolerance`
- 或者,可用上面的全部資訊填充一個 `AMP::ServerState` 物件,並將該物件傳遞到 `AMP::Server::GetInstance().UpdateState`

5.4 標記

AMP 可以在 CPU 圖表上顯示與在被分析的應用程式內進行的 C++ 或 ActionScript 調用相對應的特殊指標、標記。按如下所示方式在 ActionScript 中添加標記：

```
Amp.addMarker(1)
```

注意:編譯包含 `Amp.addMarker()` 代碼的 .as 檔可能會產生某種編譯錯誤。要避免這一點,務必使用 Resources/AS2/CLIK 目錄下提供的固有 Amp 類。

按如下所示方式在 C++ 中添加標記：`GFx::MovieImpl:: AdvanceStats->AddMarker(1)`

AMP 就會在進行過調用的幀上顯示一個標記。將來將使用整型參數顯示一種以上類型的標記。

6 與 AMP 相關的常見問題解答

本節包含開發者在使用 AMP 工具時可能會遇到的其中幾個問題。

1. 遊戲端使 AMP 工作所需的最低限度是什麼？

您需要一個啟用 AMP 的內部版本(即定義了 `GFX_AMP_SERVER`)。預設情況下,GFx 的所有非發售版本均啟用 AMP。另外,如果您的遊戲不調用 `Platform::RenderThread::drawFrame`,您就需要確保每個幀都更新了 AMP,這是通過調用 `AmpServer::GetInstance().AdvanceFrame` 實現的。

2. 使用 AmpClient 連接我們的遊戲時,我們的遊戲的畫面播放速率變得非常緩慢。我們發現 `AmpServer::AdvanceFrame()` 花費大量時間。這正常嗎？

就多數分析器而言,連接 AMP 時出現性能損失屬於正常情況。要最大限度地減小此性能影響,請確保以較低分析等級啟動,這可通過從 AMP 用戶端工具列上的適當下拉控制項中進行選擇來設定。然後,一旦查找到瓶頸,您就可以在瓶頸處提升分析等級,以更加詳細地確定根源。每個幀都請求完整記憶體報告也可能會對性能產生重大影響,因此,請確保僅在需要查閱時才生成這些報告。您可以通過 AMP 用戶端工具列上的 "i"(小寫 I)來切換詳細記憶體報告的開與關。

3. 我在 Windows/PS3/Xbox 上運行遊戲,而且能夠在發現視窗中找到 AMP 用戶端後與其連接。不過,所有統計資料均為空白。

AMP 每個幀都需要收到資訊,以便於顯示記憶體和性能統計資料。如果您的遊戲不調用 `Platform::RenderThread::drawFrame`,您就需要確保每個幀都更新了 AMP,這是通過調用 `AmpServer::GetInstance().AdvanceFrame` 實現的。

4. AMP 為何在更新時暫停一秒鐘或更長時間(`AMP::Server::WaitForAmpConnection`)？

正常情況下,AMP 伺服器在佔用記憶體太大時會暫停一秒鐘,以便於將積聚的資料發送到用戶端。如果這種現象繼續發生,那就可能是您的內容產生過多分析資訊,以至於 AMP 如果不暫停 Scaleform 就沒有時間將這些資訊發送到用戶端。

您可以通過關閉原代碼行計時和詳細記憶體報告(即 AMP 用戶端上的 'i' 按鈕)來減少所發送的分析資訊的數量。AMP 用戶端中也有一個'分析等級'(profile level) 下拉控制項,也可以進行降低。最後,您可以升高觸發此問題的記憶體閾值。預設數值設置為 1MB,但您可以通過調用 `GFx::AMP::Server::SetHeapLimit` 進行修改。

5.

預設情況下,AMP 堆限制為 1MB,儘管這實際上不是一個硬性限制,因此它是可以超過一點的。您可以通過調用 AmpServer::SetHeapLimit 在您的遊戲中更改此限制。此限制越大,在將資料發送到 AMP 用戶端時 GFx 必須暫停的可能性越小。

7 更多資訊

有關 AMP 的更多資訊,請查閱以下資源:

- [AMP 使用案例\(AMP Use Cases\)](#) 文檔
- [Scaleform Reference 文檔](#) – 查閱有關代碼/類的詳細資訊。
- [AMP Forum](#) – 關於社區問題和支援。