

Autodesk® Scaleform®

AMP 사용자 설명서

이 문서는 Scaleform AMP(Analyzer for Memory and Performance)에 대한 소개 및 응용 프로그램 프로파일링을 위한 시스템 사용 방법을 기술한 문서입니다.

작성자: Alex Mantzaris

버전: 3.02

최종 수정일: 2012 년 6 월 15 일

Copyright Notice

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.

Autodesk Scaleform 연락 방법

문서	AMP 사용자 설명서
주소	Autodesk Scaleform Corporation 6305 Ivy Lane, Suite 310 Greenbelt, MD 20770, USA
홈페이지	www.scaleform.com
전자 우편	info@scaleform.com
직통 전화	(301) 446-3200
팩스	(301) 446-3199

차례

1	소개	1
1.1	AMP 서버	2
1.2	AMP 클라이언트.....	3
2	AMP 시작하기.....	10
2.1	연결하기	10
2.2	성능 매트릭스 분석	11
2.3	렌더링 매트릭스 분석.....	12
2.4	메모리 사용 분석.....	13
2.5	오버드로우 프로파일링사용하기	21
2.6	배치 프로파일링 사용	22
3	플랫폼 및 통합 관련 주의사항.....	23
4	빌드 주의사항.....	25
5	AMP 지원 추가.....	27
5.1	Debug 파일 생성	27
5.2	응용 프로그램 제어	27
5.3	연결 상태.....	28
5.4	마커	29
6	AMP 관련 FAQ.....	30
7	추가 정보.....	32

1 소개

AMP™는 Scaleform®의 원격 프로파일링 시스템으로, Scaleform 응용 프로그램의 CPU 사용율, 그래픽, 렌더링 및 메모리 사용을 모니터링할 수 있습니다. 실시간 그래프를 통해서 신속히 메모리 및 성능 저하를 확인할 수 있는 방법과 프레임별 통계를 제공하며, 이러한 기능을 통해 상세한 분석을 얻고 문제 원인을 파악할 수 있습니다. 또한, AMP는 ActionScript 함수와 소스 코드의 프레임별 시간을 측정해서, 스크립트 문제 영역을 찾을 수 있도록 도와줍니다.

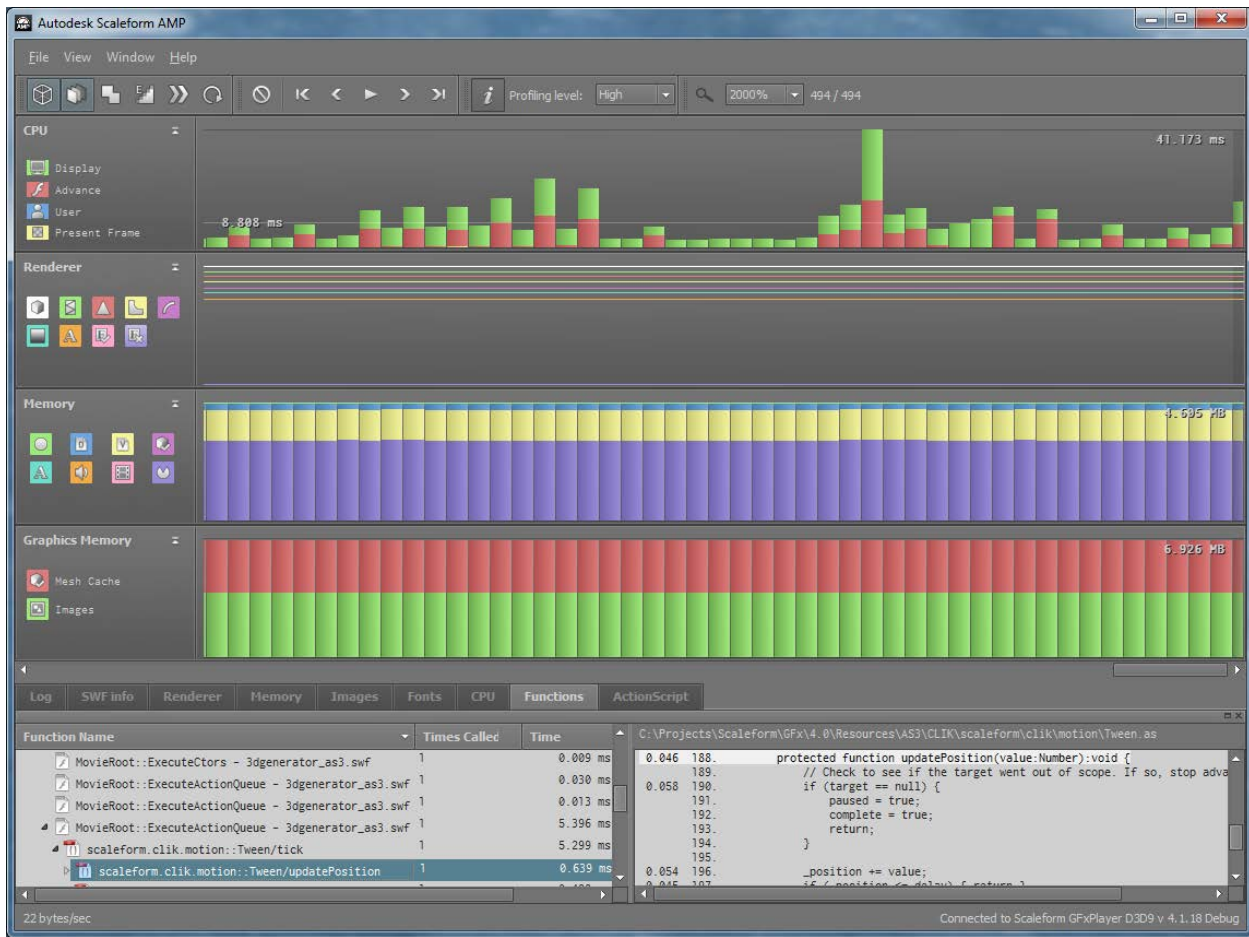


그림 1: Scaleform AMP

Scaleform 응용 프로그램은 AMP 서버 역할을 수행합니다. 이것은 응용 프로그램이 프로파일러의 접속 요청을 받아서 매 프레임 정보를 수집하고, 수집한 정보를 네트워크를 통해 프로파일러로 보내기 때문입니다. 프로파일러는 AMP 클라이언트 역할을 수행합니다. 이것은 프로파일러가 특정한 AMP

서버에 접속하여 해당 서버에서 프레임별로 보내는 정보를 처리하여, 프로파일링된 응용 프로그램을 제어하기 위한 요청을 보내기 때문입니다.

AMP 는 Scaleform 3.2 및 상위 버전에서 작동합니다.

1.1 AMP 서버

프로파일링 중인 Scaleform 응용 프로그램(AMP 서버)은 일정한 양의 정보를 네트워크를 통해서 프로파일러로 전달합니다. 함수 프로파일링 레벨에 따라 각각의 ActionScript 함수와 일부 중요한 C++ Scaleform 함수의 시간을 측정할 수 있습니다. ActionScript 2 의 라인별 소스코드 프로파일링이 활성화 되면 액션스크립트의 실행 시간을 측정한다. (ActionScript 3 의 소스코드 프로파일링은 각각의 실행 타이밍이 필요없다.) 이외에도 렌더링 통계, 메모리 힙 및 이미지에 대한 상세 정보, flash 파일 정보, ActionScript 마커가 기록됩니다. 이러한 프레임별 추가 작업은 응용 프로그램의 속도를 저하시키고 메모리 사용을 증가시킬 수 있습니다. 라인별 시간 측정을 수행할 경우는 이러한 문제가 심화될 수 있습니다.

APM 가 프로파일링 중인 응용 프로그램의 간섭을 받지 않도록 하기 위해서 모든 AMP 서버의 정보는 분리된 메모리 힙에 보존됩니다. 힙 사이즈가 미리 설정된 한계를 넘으면, 처리되지 않은 프레임 정보를 네트워크를 통해 전달받은 다음 삭제할 때까지 응용 프로그램의 작동이 중지됩니다. Scaleform Shipping 빌드는 AMP 서버의 역할을 하지 않으며, 요청이 있어도 응용 프로그램 내의 어떤 빌드에 대해서든 AMP 지원이 쉽게 불가능할 수 있습니다.

서버의 성능 통계는 Movie 별로 유지됩니다. 이를 통해 사용자는 특정한 SWF 파일과 연결된 몇 가지 view 가 존재하는 상황에서 문제 지점을 보다 쉽게 파악할 수 있습니다. 또한, AMP 는 개별 스프레드에서의 다중 Movies 의 ActionScript 실행에 대한 중요한 호출 그래프를 생성할 수 있습니다.

클라이언트에 대한 메모리 보고 및 통계 외에도, AMP 클라이언트는 여러 가지 방법으로 서버를 제어하여 프로파일링에 대한 지원을 얻습니다. 예를 들어, 프로파일러는 와이어프레임 모드 렌더링, flash movie 빨리 감기, 일시 중지 및 재시작, 안티 알리아싱 및 스트로크 모드 전환, 로컬라이제이션

폰트 변경, 벡터 그래픽의 curve tolerance 변경을 요청할 수 있습니다. 또한, 클라이언트는 마스크, 필터, 픽셀 오버드로우 하이라이팅과 같은 성능 저하 항목을 보여주는 특수한 AMP 렌더링 모드를 전환할 수 있습니다.

위에서 명시한 모든 기능은 Scaleform Player 에서 실행되며, 일부는 응용 프로그램에 특정한 실행이 필요하고 Scaleform 통합의 일부로서 수행될 수 있습니다. 이러한 기능 중 일부가 Scaleform 응용 프로그램에서 실행되지 않는 경우를 능숙하게 처리하기 위해서, AMP 서버는 지원 응용 프로그램 제어 기능 셋을 연결된 클라이언트로 보고합니다. 또한, 비주얼 피드백으로 클라이언트에 현재 상태(예: 와이어프레임 모드)를 보고합니다.

마지막으로, AMP 서버는 클라이언트가 Flash 디버그 정보 에 대해 로컬 액세스를 하지 못할 경우 소스 코드와 라인별 시간 측정이 표시될 수 있도록, 네트워크를 통해서 프로파일러로 해당 정보를 보냅니다. 이러한 정보 전달을 위해서는 해당 SWD 파일(ActionScript2) 또는 AS 파일(ActionScript3)이 AMP 가 찾을 수 있는 위치에 있어야 합니다.

1.2 AMP 클라이언트

원격 프로파일러(AMP 클라이언트)는 단일 실행 응용 프로그램으로, AMP 서버와 연결하여 매 프레임 프로파일 정보를 받고, 이러한 정보를 사용자 친화적인 방법으로 표시하여 성능 및 메모리 저하 분석이 효과적으로 이루어지도록 합니다. 프로파일러는 그 자체가 Scaleform 응용 프로그램이며, Scaleform CLIK 을 통해서 실행됩니다.

AMP 클라이언트가 제공하는 몇 가지 기능으로 개발자들은 Flash 자산의 다양한 상을 확인하고 문제 영역을 파악할 수 있습니다.

- **CPU graph:** ActionScript(Advance), Scaleform 렌더링(Display), Direct Access API (User), 그래픽 버퍼 스와핑(PresentFrame)에서 게임 프레임당 CPU 소요 시간을 보여줍니다.이 그래프는 특히 Scaleform 에 따른 CPU 사용율의 급격한 상승을 파악하는 데 유용합니다. 급격한 상승을 보인 프레임을 선택하면, 해당 프레임에 대한 상세 통계를 검사하여 원인을

파악할 수 있습니다. 사용자는 그래프를 왼쪽 마우스 버튼으로 클릭하고 원하는 프레임 범위가 선택될 때까지 마우스를 드래그하여 여러 개의 프레임을 선택할 수 있습니다. 그래프를 오른쪽 마우스 버튼으로 클릭하고 드래그하면 그래프 창을 스크롤합니다. 그래프의 각 항목은 범례(legend) 상자에서 해당되는 전환(toggle) 버튼을 클릭하여 표시하거나 숨길 수 있습니다.

- **Rendering graph:** [Best Practices Guide](#) 문서에는 성능 저하를 일으킬 수 있는 렌더링 관련 영역(과도하게 많은 원형(primitive), 삼각형, 선 그리기 또는 마스크, 필터, 외곽선의 사용, 또는 그라디언트 채우기 등)이 나와 있습니다. 이러한 영역 각각의 수치는 렌더링 그래프로 표시되며, 개발자들은 이를 통해 Scaleform 에서 발생하는 사안을 더욱 폭넓게 이해할 수 있습니다.
- **Memory graph:** Scaleform 의 메모리 사용은 메모리 그래프로 추적되며, 개발자들은 이를 이용하여 카테고리(렌더링, 이미지, 사운드, 폰트, 영상 데이터 및 전체 불러온 파일 등) 별로 메모리 사용을 빠르게 모니터링할 수 있습니다.
- **Graphics memory graph:** 대부분의 Scaleform 그래픽 메모리 사용 또한 AMP 를 통해 추적되며, 이미지 및 매시 캐시에 할당된 GPU 메모리를 보고합니다.
- **Log tab:** Log 탭을 클릭하면 프로파일된 응용 프로그램이 연결된 동안 생성한 Scaleform 로그가 표시됩니다.
- **SWF info tab:** 이 탭은 프로파일되는 모든 플래시 파일에 대하여 부분적인 정적 정보를 제공합니다. 이 정보는 파일 이름, 플래시 버전, 무비의 크기 및 무비의 Frame rate 를 포함합니다.
- **Renderer tab:** 이 탭은 그래프와 동일한 정보를 수치화하여 보여줍니다. 여기에는 또한 매시 및 폰트 캐시 스래싱(프레임 수행 중 강제로 제거된 항목 수), 폰트 채우기 비율(폰트 캐시 텍스처 영역의 합계에서 모든 문자 영역이 차지하는 비율), 폰트 캐시 실패율(프레임 중 할당 실패한 문자 수와 벡터 셰이프로 교체된 문자 수), 프레임 중 폰트 캐시 텍스처 업데이트 수, 현재 폰트 텍스처 수 등 일부 추가 렌더링 통계 정보를 포함합니다.
- **Memory tab:** 이 탭을 통해 사용자가 Scaleform 의 메모리 사용량, 즉 Scaleform 이 실제로 사용 중인 메모리량과 메모리의 용도별 할당 상태를 자세하게 확인할 수 있습니다.
- **Images tab:** 각 이미지와 그 크기 및 형식에 따라 사용되는 메모리를 표시합니다. 목록에서 각 항목을 클릭하면 미리보기 창에 해당 이미지의 축소판 그림이 표시됩니다. 이미지 데이터는 프로파일링 정보에 함께 전송되지 않으며 요청에 따라 전송됩니다. 따라서 Scaleform 을 통해






이미지를 이미 불러온 경우 이미지가 표시되지 않을 수 있습니다. 현재 AMP 는 모든 이미지 형식의 미리보기를 지원하지 않습니다(DXT 압축 이미지를 포함).

- **Fonts tab:** 폰트 캐시가 사용한 총 메모리와 각 SWF 의 폰트 목록이 표시됩니다. 이 탭에서 항목 선택 시, 현재 폰트 캐시를 나타내는 이미지가 Images 탭에서의 이미지 미리보기와 비슷한 방식으로 표시됩니다.
- **CPU tab:** Advance, Display, User 시간을 하위 카테고리로 분류하여 선택한 프레임에 대해 시간이 소요된 부분의 상세 개요를 제공합니다. 프레임을 여러 개 선택한 경우에는 선택한 모든 프레임의 평균 수치가 표시됩니다.
- **Functions tab:** 각 함수에서 호출한 함수를 포함하여 모든 함수 내부에서 걸린 시간을 계층 구조로 보여줍니다. 프레임을 여러 개 선택한 경우에는 시간과 호출 횟수가 게임 프레임 별 평균치로 표시됩니다. ActionScript 와 C++ 함수는 같은 호출 그래프에 함께 들어가 있습니다. 이것은 코드 경로가 ActionScript VM 과 나머지 응용 프로그램 간에 이동할 수 있기 때문입니다. 각 함수를 확장하면 해당 함수에서 호출된 함수가 나타납니다.
- **ActionScript tab:** 이 탭은 Functions 탭과 비슷한 정보를 제공하지만, 해당 정보는 계층적이지 않으며 C++ 함수는 제외되고 각 함수의 시간에는 목록의 다른 함수에서 소비한 시간이 제외됩니다. 가장 많은 시간이 소요되는 스크립트 함수를 빠르게 확인하는 데 유용합니다.
- **Source pane:** 이 창은 Functions 탭과 ActionScript 탭의 일부입니다. AS3 의 경우 AS 파일 내 코드에 대한 라이별 소요 시간이 항상 활성화됩니다. SWF 에 임베드된 AS3 코드의 경우 소스 코드는 표시되지 않습니다.. AS2 의 경우 라인 별 소요 시간(기본적으로 꺼짐)을 확인하려면 툴바의 드롭다운 컨트롤에서 "높은" 레벨의 함수 프로파일링을 선택해야 합니다. AS2 의 명령 프로파일링은 시간을 소요하며 프로파일링 중인 Scaleform 응용 프로그램의 속도가 느려지므로, 문제 영역을 완전히 확인한 다음에 실행하는 것이 좋습니다. 1 개 이상의 다른 함수로부터 호출된 함수의 경우, 각 라인 옆에 보이는 시간은 함수 호출 그래프에서 보고된 전체 시간에 추가되지 않습니다. 이것은 라인 별 시간이, 해당 함수를 호출한 곳과 관계 없이 함수 내에서 소요된 전체 시간이기 때문입니다. 또한, 함수 그래프 시간이 호출된 함수에서 소요된 시간에 포함되는 반면, 라인 시간은 함수 호출에서 소요된 시간에 포함되지 않습니다. 소스 코드와 라인 별 시간을 표시하려면, *AMP 서버(SWF 위치)에서 찾을 수 없는* 모든 flash


디버그 파일이 실행 가능한 AMP 클라이언트의 작업 경로 또는 AS3 소스 파일의 경우 로드된 SWF 에 대해 동일한 상태 경로 안에 있어야 합니다.

- **File Menu:** 이 메뉴에는 다음 항목을 선택할 수 있습니다.
 - **Connection:** 다른 서버로의 연결에 관한 연결 대화 상자를 표시합니다.
 - **Disconnect:** 현재 연결이 종료됩니다.
 - **Debug Info Paths:** 파일 시스템에서 AMP 가 SWD 파일(AS2 의 경우) 또는 AS 파일(AS3 의 경우)을 검색하는 모든 경로가 나열된 대화 상자를 불러옵니다. 경로는 Add 및 Remove 버튼을 사용하여 추가하거나 제거할 수 있습니다.
 - **Load Profile Frames:** 재검수를 위해 이전에 저장한 프로파일 실행을 불러옵니다. 이 메뉴 항목을 선택하면 프로파일 데이터 파일을 찾기 위한 파일 로드 대화 상자가 표시됩니다.
 - **Save Profile Frames:** 디스크에 현재 프로파일 실행을 저장합니다. 저장한 데이터는 위의 "Load Profile Frames" 옵션을 선택해서 다시 불러올 수 있습니다. 이 메뉴 항목을 선택하면 파일 저장 대화 상자가 표시되고 여기서 프로파일 데이터가 포함된 파일의 경로 및 이름을 선택합니다.
 - **Dump Memory Report:** 현재 프로파일 프레임을 AMP 작업 폴더에 "AmpMemReport.txt" 파일로 저장합니다.
 - **Exit:** 응용 프로그램을 종료합니다.







- **View menu:** 이 메뉴에서는 다음 항목을 선택할 수 있습니다.
 - **Graph Tooltips:** 이 메뉴를 선택하면 AMP 의 그래프에 마우스를 올릴 때 컨텍스트 위주 팝업 도움말을 활성화하거나 비활성화합니다.
 - **Time Units:** Display 및 Advance 시간 등 모든 시간 요소 보고에 사용할 단위를 선택할 수 있습니다. 제공 단위는 밀리초(millisecond)와 마이크로초(microsecond)입니다.
 - **Memory Units:** Memory 탭 및 Memory 그래프에서 사용할 단위를 선택할 수 있습니다. 제공 단위는 바이트(byte), 킬로바이트(kilobyte), 메가바이트(megabyte)입니다.

- **Window menu:** 이 메뉴에서 상기 표시된 탭(Log, SWF Info, Renderer, Memory, Images, Fonts, CPU, Functions, ActionScript)을 표시하거나 숨길 수 있습니다. 또한 선택 시 AMP 응용 프로그램에서 창에 관련된 사용자 정의 사항을 제거하는 Restore UI 을 제공합니다.
- **Help menu:** 이 메뉴는 현재 두 가지 옵션을 제공합니다. Scaleform.com 개발자 센터의 이 문서로 연결해 주는 User Guide 와 AMP 버전 및 개발진 정보를 확인할 수 있는 About 을 선택할 수 있습니다.
- **Application Control Toolbar:** 프로파일링 중인 응용 프로그램을 제어하는 일련의 버튼이 포함된 UI 구성 요소입니다.
 -  **Wireframe:** 와이어프레임 모드에서 콘텐츠를 렌더링하기 위해 프로파일된 응용 프로그램으로 요청을 보냅니다. 응용 프로그램이 이미 와이어프레임 모드일 때 버튼을 클릭하면 해당 모드가 종료됩니다.
 -  **Overdraw mode:** 픽셀 오버드로우, 마스크, 필터 영역을 강조하는 렌더링 모드로 들어가기 위해 프로파일된 응용 프로그램으로 요청을 보냅니다. AMP 서버는 픽셀 오버드로우 영역을 녹색(색 농도는 오버드로우의 수에 비례), 마스크를 적색, 필터를 청색으로 렌더링합니다. 응용 프로그램이 이미-오버드로우 모드일 때 버튼을 클릭하면 해당 모드가 종료됩니다.
 -  **Batch profiling mode:** 서로 다른 색상으로 각 렌더 배치를 드로우하는 렌더링 모드로 들어가기 위해 프로파일된 응용 프로그램으로 요청을 보냅니다. 응용 프로그램이 이미 배치 프로파일링 모드일 때 버튼을 클릭하면 해당 모드가 종료됩니다.
 -  **Anti-aliasing:** 안티 알리아싱 모드로 들어가기 위해(순환) 프로파일된 응용 프로그램으로 요청을 보냅니다. Scaleform에서는 EdgeAA(에지 안티 알리아싱) 기술, HW FSAA(하드웨어 전화면 안티 알리아싱) 및 None(안티 알리아싱 없음)의 세 가지 모드를 제공합니다.
 -  **Fast-forward:** 빨리 감기로 flash 콘텐츠를 재생하기 위해 프로파일된 응용 프로그램으로 요청을 보냅니다. 이렇게 해서 SWF 파일이 렌더링 프레임만큼 빨리


진행되도록 하며, 문서 FPS 설정을 무시합니다. 응용 프로그램이 이미 빨리 감기 모드일 때 버튼을 클릭하면 해당 모드가 종료됩니다.

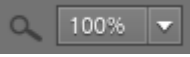
-  **Restart:** flash 콘텐츠를 다시 시작하기 위해 프로파일된 응용 프로그램으로 요청을 보냅니다.

- **Frame control toolbar:** 이 툴바는 다음 버튼으로 구성되어 있습니다.

-  **Clear:** 현재 프로파일 실행을 중지합니다. 이렇게 하면 프로파일 프레임이 삭제되므로 새 데이터를 저장할 AMP 클라이언트 메모리를 확보할 수 있습니다
-  **First Frame:** 현재 실행한 프로파일의 첫 번째 프레임을 선택하여 아래쪽 패널 탭에 표시되는 정보를 업데이트합니다.
-  **Previous Frame:** 현재 선택된 프레임의 바로 전 프레임을 선택하여 아래쪽 패널 탭에 표시되는 정보를 업데이트합니다.
-  **Pause:** 메모리 및 성능 데이터 전송을 중단하기 위하여 프로파일된 응용 프로그램으로 요청을 보냅니다. 이 동작으로 응용 프로그램이 중단되는 것은 아닙니다.
-  **Next Frame:** 현재 선택된 프레임의 바로 다음 프레임을 선택하여 아래쪽 패널 탭에 표시되는 정보를 업데이트합니다.
-  **Last Frame:** 현재 실행한 프로파일의 마지막 프레임으로 이동한 다음 이를 선택합니다. 새 프레임 정보를 계속 수신해도 항상 맨 마지막에 수신된 프레임을 선택하여 아래쪽 패널 탭에 표시되는 정보를 끊임없이 업데이트합니다.

- **Profiling toolbar:** 프로파일링에 사용되는 일련의 버튼이 포함된 구성 요소입니다.

-  **Detailed Profiling**(메모리 상세 분류): 이 옵션을 선택하면 AMP 가 더욱 자세한 프레임별 메모리 통계를 수집하여 Memory 탭에 표시합니다. 또한 이 옵션을 선택하면 AMP 가 Images 탭에 최대 10 개의 이미지 목록 대신 전체 목록을 표시합니다.

-  **Zoom toolbar:** 그래프의 수평크기를 변경하는 드롭다운 컨트롤로 구성되어 있습니다. 줌아웃하면 해당 실행의 문제 영역을 확인할 수 있고, 줌아웃하면 문제 영역에 대한

더욱 정밀한 조사가 가능합니다. 마우스 휠로도 줌인 또는 줌아웃할 수 있으며, 드롭다운 중에는 높은 해상도를 사용할 수 없습니다. 또한 현재 선택된 프레임의 개수를 툴바에 표시합니다.

- **Status bar:** 프로파일러 하단에는 현재 연결과 네트워크를 통해 받은 정보의 비트레이트를 보여주는 Status Bar 가 있습니다.

AMP 는 외형에 대한 사용자 정의를 지원합니다. 툴바 및 탭의 정렬 위치는 새로운 위치로 드래그하여 변경할 수 있습니다. 탭과 툴바는 응용 프로그램에서 제거할 수 있습니다. 그래프와 탭의 크기는 조정 가능하며 그래프를 부분적 또는 완전히 펼칠 수 있습니다. 이러한 UI 의 변경은 AMP 프로파일링 세션보다 우선되나 사용자가 Window 메뉴에서 Reset UI 을 선택하여 지오메트리를 초기화할 수 있습니다.

2 AMP 시작하기

AMP 프로파일러를 사용하려면 GfxAmpClient.exe 파일을 실행하십시오. 해당 파일은 Scaleform 주 설치 경로 아래의 *Bin/AMP*에 있습니다. 응용 프로그램이 시작되고 연결 대화 상자가 나타납니다.

2.1 연결하기

클라이언트가 시작되면 연결 대화 상자가 자동으로 나타납니다. 검색된 AMP 서버 목록에서 선택하거나 수동으로 IP 주소와 연결할 포트를 입력합니다. 대상의 AMP 서버가 아직 실행되지 않았고, 시작하면서 응용 프로그램을 프로파일해야 한다면 두 번째 방법을 선택합니다.

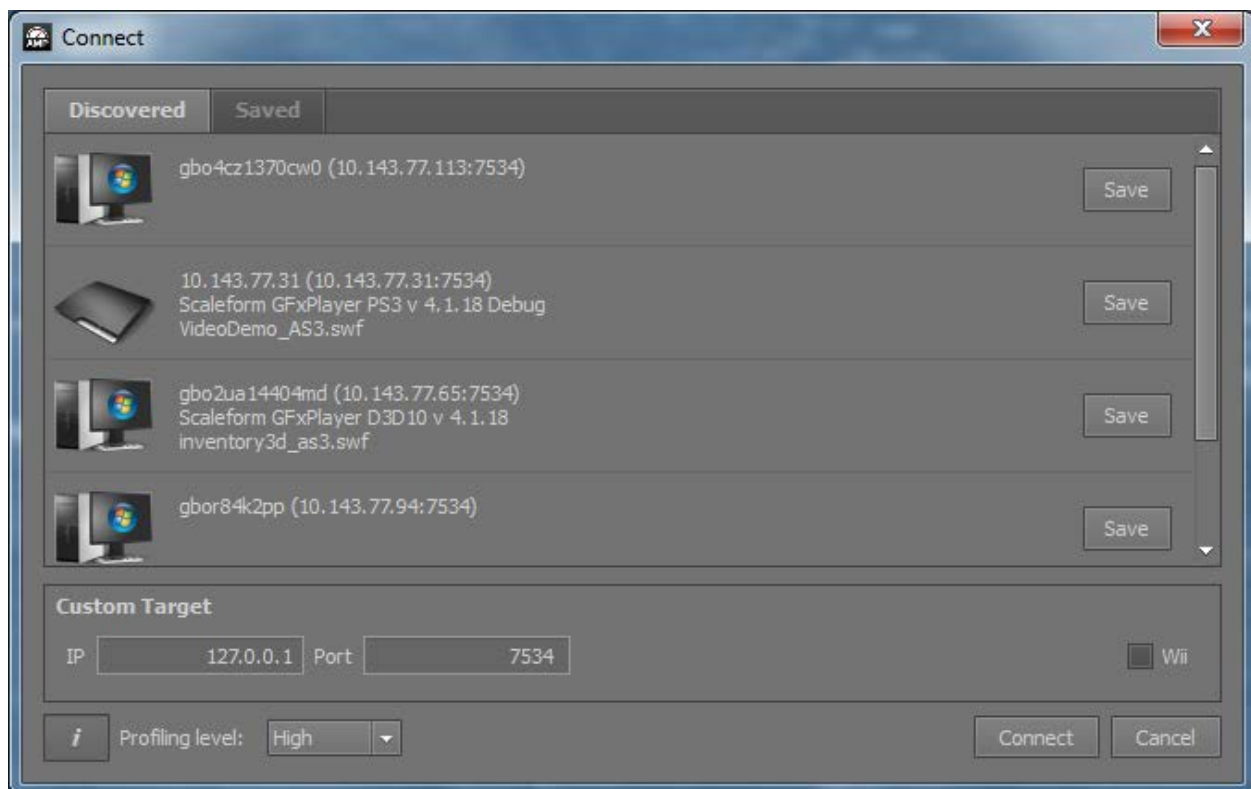


그림 2 : AMP 연결 대화 상자

File 메뉴 하위의 Connection 항목은 언제든지 연결 대화 상자를 여는 데 사용할 수 있습니다.

검색된 목록 내의 AMP 서버에는 플랫폼(PC, Xbox 360, PS3 등)을 확인할 수 있는 특유의 아이콘이 붙어 있습니다. 아이콘 옆에는 응용 프로그램 이름, IP 주소와 서버가 연결 대기 중인 포트, 현재 재생 중인 Flash 콘텐츠가 표시됩니다. 이러한 정보의 일부가 사라질 경우, 그 원인은 연관된 AMP 서버가 해당 정보를 전송하지 않기 때문입니다. Scaleform 응용 프로그램에 AMP 지원 전체를 추가하는 방법(응용 프로그램 및 콘텐츠 정보 브로드캐스팅 포함)에 대해서는 이 문서 후반부에 설명되어 있습니다. 검색된 서버를 두 번 클릭하거나 서버 선택 후 "Connect" 버튼을 클릭하여 연결을 구성합니다.

연결 대화 상자에서 프로파일링 레벨과 연결에 대한 상세 메모리 보고를 선택해 연결 즉시 설정되도록 할 수 있습니다. 이 작업은 Scaleform 응용 프로그램의 시작을 프로파일링할 경우에 특히 중요합니다.

Wii 개발 키트(NDEV)에 연결할 때는 위의 서버 검색 방법을 사용하지 않습니다. 대신, Wii 는 NDEV 에 내장된 USB 인터페이스(COM 포트)를 사용하여 PC 에 직접 연결합니다. IP 주소 필드에 "wii"라고 명확히 하여 연결 대화 상자에서 Wii AMP 서버로 연결합니다.

2.2 성능 매트릭스 분석

연결이 구성되면 프로파일링 정보가 클라이언트로 전달되어 그래프 생성이 시작됩니다. 상위 그래프는 프레임(Scaleform (Display) 렌더링 시, ActionScript (Advance) 실행 시, Direct Access API (User) 내부) 별로 소요되는 CPU 시간이 어느 정도인지를 보여줍니다.

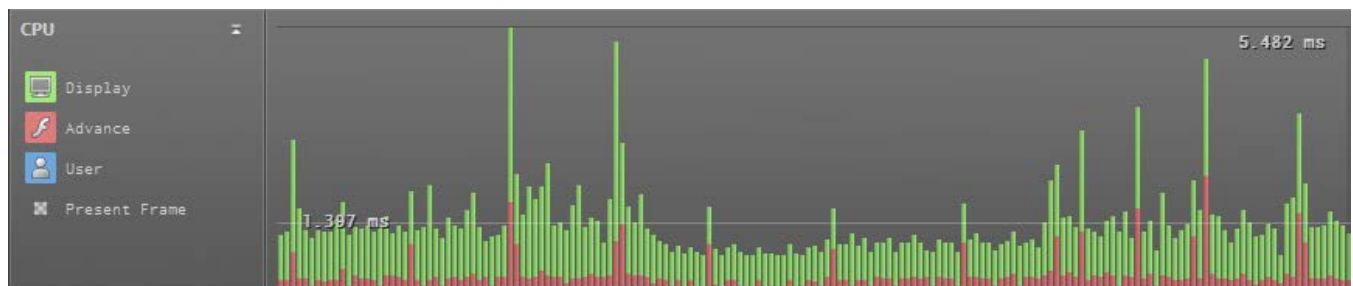


그림 3: CPU 그래프

분석이 진행되는 동안 어떤 지점에서 중요하지 않은 그래프는 최소화되거나 크기가 조절되어 다른 UI 구성 요소가 더 많은 공간을 사용할 수 있습니다. 중요하지 않은 그래프는 범례 옆에 있는 아이콘을 클릭하여 제거할 수 있습니다.

CPU 그래프와 관련된 CPU Summary, Functions 및 ActionScript 탭이 프로파일러의 하단에 표시됩니다. 탭은 그래프 정보를 좀 더 자세하게 보여주며, Function 탭은 선택된 프레임에 대한 호출 그래프를 보여줍니다. 또, ActionScript 탭은 선택된 프레임 동안에 호출된 모든 스크립트 함수를 비계층적으로 보여줍니다.

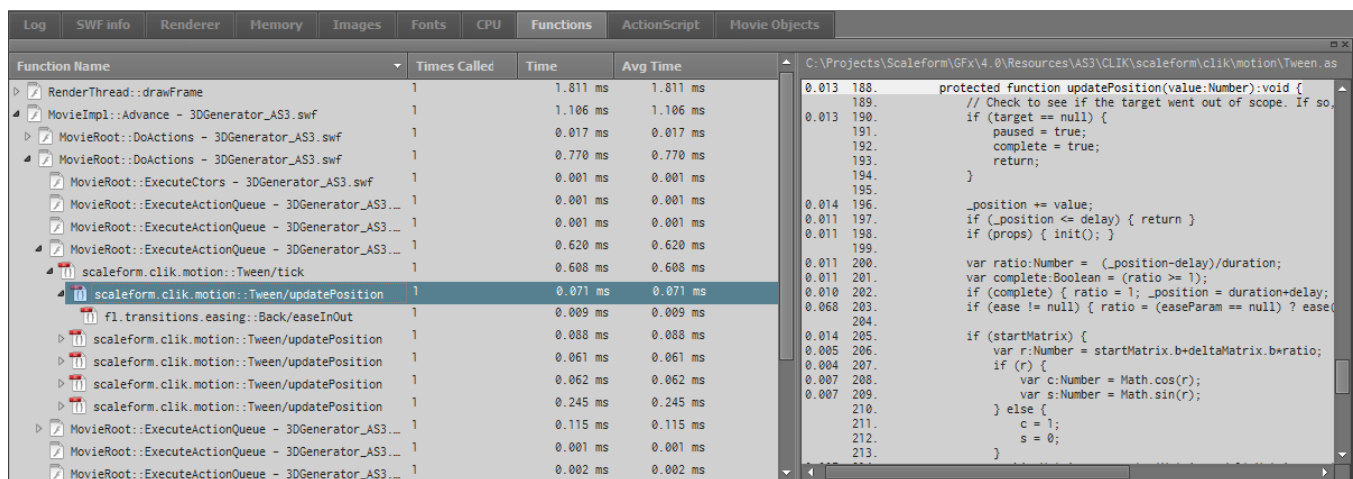


그림 4 : CPU Summary

CPU 그래프에서 스파이크가 있는 프레임의 CPU 상세 탭을 검사하면, 스크립트 성능 저하의 원인을 확인할 수도 있습니다. 반면, 렌더링 성능 저하는 렌더링 그래프를 조사하여 훨씬 쉽게 발견할 수 있습니다.

2.3 렌더링 매트릭스 분석

렌더링 그래프는 선택된 프레임을 렌더링하는 동안 발생한 그리기 원본 개수, 삼각형 개수, 선 개수, 사용된 마스크 개수, 스트로크 개수, 그라디언트 채움 개수, 매쉬 개수의 그림 개수 및 폰트 업데이트 개수를 보여줍니다. [Best Practices Guide](#)에 상기 요소들을 최소화하는 것이 성능에 중요한 이유가 자세히 설명되어 있습니다.

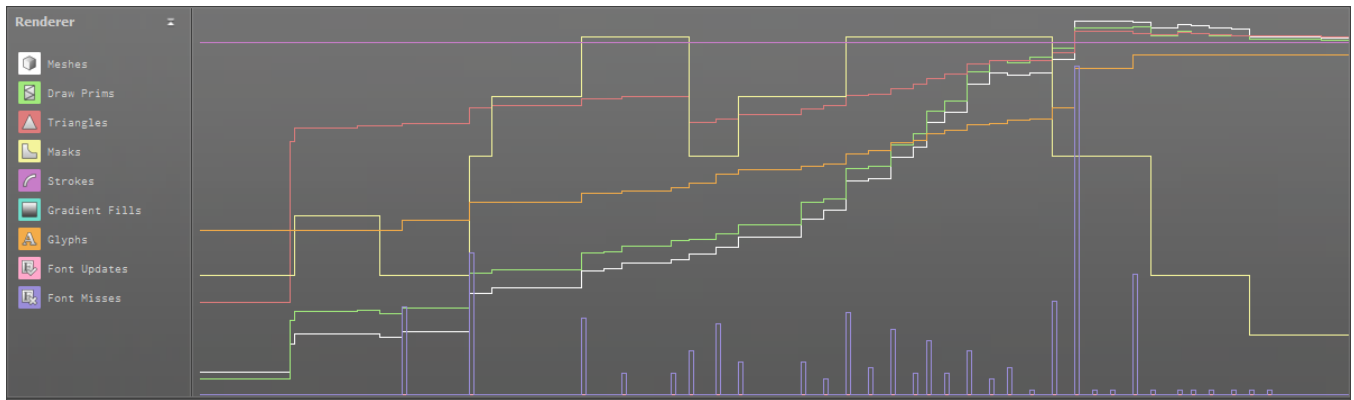


그림 5: 렌더링 그래프

렌더링 그래프와 관련된 Render Summary 탭이 응용 프로그램의 하단에 표시됩니다.

Log	SWF info	Renderer	Memory
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			

그림 6: Rendering Summary

2.4 메모리 사용 분석

위에서 세 번째와 네 번째 그래프는 Scaleform 시스템 및 그래픽 메모리 사용량을 각각 표시합니다.

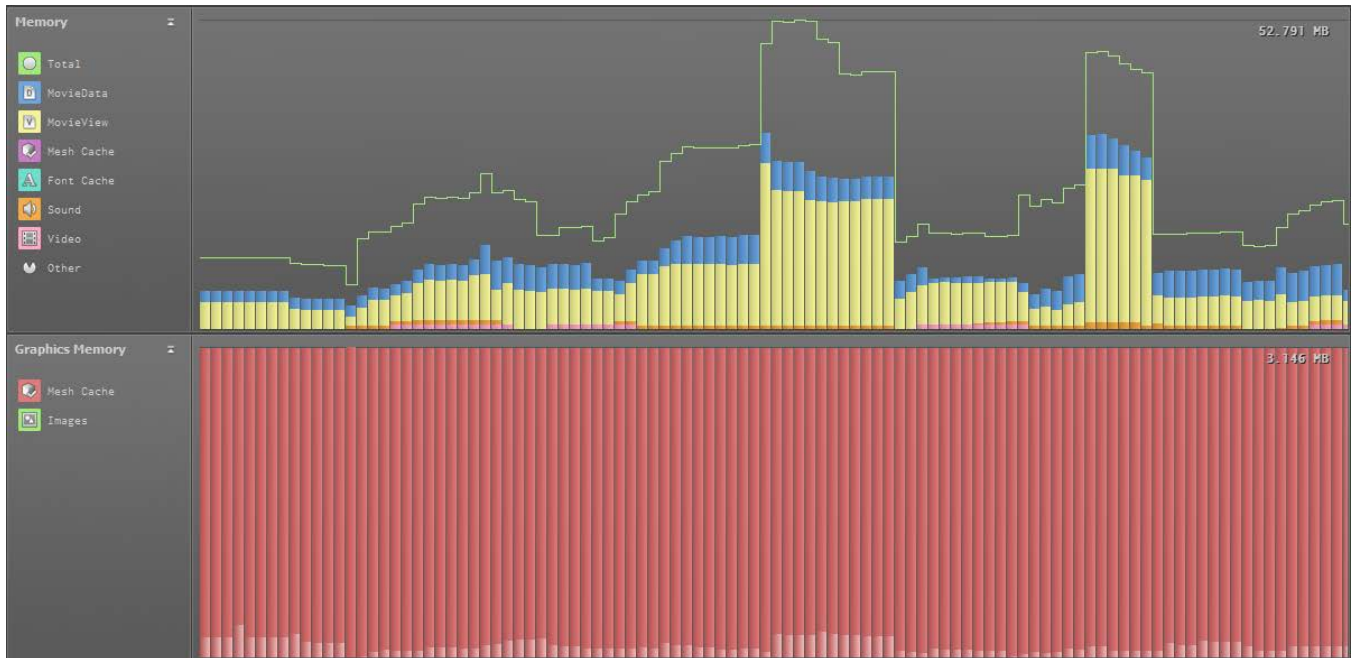


그림 7: Memory 그래프

- **Total:** Scaleform 가 점유한 전체 메모리입니다(차지 공간). 전체 메모리는 선 그래프로 나타나며, 아래에 나열된 모든 카테고리 및 오버헤드, 요청된 할당의 입상 및 단편화 등으로 사용하지 않은 메모리를 모두 포함합니다. 사용되지 않은 메모리는 따로 카테고리로 표시되지 않으므로, 전체 메모리의 양은 하위 카테고리의 메모리 합보다 많습니다.
- **MovieData:** 불러온 모든 SWF 또는 GFX 파일이 사용한 메모리입니다. 복잡한 벡터 그래픽, 임베드된 폰트, 타임라인 애니메이션이 이 수치에 영향을 줍니다.
- **Movie view:** 타임라인 애니메이션 지원, 온스크린 오브젝트 및 ActionScript 에 할당된 Gfx::Movie 인스턴스에서 사용한 런타임 메모리입니다.
- **MeshCache:** 캐쉬된 셰이프 매쉬 데이터가 사용한 메모리이며, 해당 데이터는 벡터 tessellator 와 에지 안티 알리이싱을 통해 생성되었습니다. 웨입 메시 데이터를 위해 사용되는 메모리이며, 벡터 테셀레이터 및 edge antialiasing 에 의해 생성됩니다. 매쉬 데이터 메모리는 플랫폼에 따라 시스템 또는 그래픽 메모리에서 할당될 수 있습니다. 실제로 사용된 메모리는 더 어두운 음영으로 표시되므로 사용자가 더 작은 매쉬 캐시를 설정할 수 있는지를 쉽게 파악할 수 있어 메모리가 절약됩니다.
- **Font Cache:** 폰트 캐쉬가 사용한 메모리입니다.

- **Sound:** 임베드된 사운드 데이터가 사용한 메모리입니다. 임베드된 사운드 표본의 개수, 길이, 품질에 따라 영향을 받습니다.
- **Video:** 비디오 메모리 플레이백 버퍼에 사용된 메모리입니다. 이러한 버퍼는 비디오가 시작될 때 할당되고, 정지할 때 해제됩니다.
- **Other:** Scaleform 에서 사용한 메모리입니다. 상기 카테고리에 포함되지 않는 항목입니다. 메모리 카테고리 분류에 대한 자세한 내용은 아래에 설명된 자세한 메모리 보고를 확인하십시오.
- **Images:** 이미지(텍스처)에 사용된 그래픽 메모리.

메모리 시스템과 메모리 시스템 설정, 최적화 및 관리 방법에 대해서는 [Memory System Overview](#) 문서를 참조하십시오.

Memory 탭은 위 카테고리의 내용을 좀 더 자세하게 보여줍니다. 메모리 탭의 모든 값은 보기 메뉴 아래의 메모리 단위 선택에 지정된 것과 동일한 단위(바이트, 킬로바이트 또는 메가바이트)로 표시됩니다. "Used Space"는 Gfx 에 의해 실제로 사용되는 메모리입니다. "Unused Space"는 Gfx 에 의해 예약되었지만 주로 단편화에 의해 사용 중이 아닌 메모리입니다. 디버그 데이터는 메모리 추적 및 성능 통계를 위해 Gfx 에 의해 내부에서 사용되는 메모리를 표시합니다.

전체 메모리 보고 모드에서 프로파일링하는 경우(AMP 툴바의 "i" 버튼으로 전환) 메모리 보고를 힙별, 파일별 및 목적별로 분류하여 더욱 자세하게 제공합니다. Scaleform 라이브러리의 디버그 버전은 힙 할당을 메모리 카테고리에 따라 분류하여 사용자가 Scaleform 의 메모리 사용에 대해 좀 더 자세히 파악할 수 있습니다. 보고되는 메모리 카테고리는 다음과 같습니다.

- **Renderer:** Scaleform 렌더링 엔진에 할당된 메모리입니다. 이 카테고리는 다음 하부 카테고리로 다시 나누어집니다.
 - **Buffers:** 매쉬 스테이지 버퍼에 할당된 메모리이며 생성된 매쉬가 인스턴스되어 꼭지점 및 인덱스 버퍼로 복사되기 전에 매쉬를 저장하는 단기 캐시의 역할을 합니다.
 - **RenderBatch:** 함께 그리기하는 렌더링 트리의 노드와 관련된 매쉬, 원형 또는 렌더링 명령어 등의 집합체에 할당된 메모리입니다.

- **Primitive:** 함께 그릴 꼭지점의 집합을 보관하는데 사용된 메모리입니다.
 - **Fill:** 하드웨어 채우기를 나타내는 데이터의 저장 및 관리에 사용된 메모리입니다.
 - **Mesh:** 매쉬 캐시에 할당된 메모리입니다. 매쉬 캐시는 꼭지점 및 인덱스 버퍼 데이터를 렌더링하는데 사용됩니다.
 - **MeshBatch:** 매쉬 캐시의 꼭지점 및 인덱스 버퍼에 캐시된 각 데이터 청크(chunk)에 할당된 메모리입니다.
 - **Context:** 렌더링 트리의 수명 주기를 관리하는 렌더링 지원 시스템에서 사용한 메모리입니다. 컨텍스트는 렌더링 트리의 여러 독립 로직 스냅샷을 유지하여 스레드(thread)가 상호 간섭 없이 렌더링 데이터와 작동하도록 해 줍니다.
 - **NodeData:** 렌더링 중인 오브젝트에 현재 적용된 매트릭스 또는 효과 등 프레임 상태 정보에서 사용한 메모리입니다.
 - **TreeCache:** 렌더러의 장면 그래프, 즉 렌더링 중인 오브젝트의 트리 구조에 할당된 메모리입니다.
 - **TextureManager:** 텍스처 이미지 관리를 위해 할당된 메모리입니다. 실제 이미지 데이터는 시스템 메모리에 보관되지 않는 경우가 대부분이며 따라서 이 메모리 카테고리에 해당되지 않습니다.
 - **MatrixPool:** 렌더러에서 유지되는 매트릭스풀에서 사용되는 메모리입니다.
 - **MatrixPoolHandles:** 매트릭스 처리 할당에서 사용되는 메모리입니다.
 - **Text:** 형식 정보, 스타일 및 색상 정보, html 분석, 리치 텍스트 데이터 및 행(line) 버퍼 등 텍스트의 관리와 렌더링에 할당된 메모리입니다.
 - **Font:** 그래프 래스터화 및 폰트 캐시 저장과 같은 폰트 렌더링에 관련된 메모리입니다.
- **MovieDef:** 영상 요소 템플릿을 렌더링하는 변경이 불가능한 데이터에 할당된 메모리입니다. 이 메모리는 다음 하위 카테고리로 구성됩니다.
- **CharDefs:** 버튼 및 텍스트와 같은 문자 정의에 사용된 메모리입니다.
 - **ShapeData:** 셰이프 정의에 사용된 메모리입니다.
 - **Tags:** ActionScript 태그에 사용된 메모리입니다.
 - **Fonts:** 폰트 리소스에 사용된 메모리입니다.

- **Sounds:** 사운드 데이터가 사용한 메모리입니다.
- **ASBinaryData:** ActionScript 버퍼 데이터가 사용한 메모리입니다.
- **MD_Other:** 위 카테고리 외 다른 MovieDef 데이터가 해당됩니다.
- **MovieView:** 타임라인 애니메이션 지원, 온스크린 오브젝트 및 ActionScript 에 할당된 메모리입니다. 이는 다음의 하위 카테고리로 추가 분류됩니다.
 - **MovieClip:** 온스크린 애니메이션 오브젝트가 사용한 메모리입니다.
 - **ActionScript:** ActionScript 코드 및 데이터가 사용한 메모리입니다.
 - **ASString:** ActionScript 문자열이 사용한 메모리입니다.
 - **Text:** 텍스트 관련 디스플레이 오브젝트가 사용한 메모리입니다.
 - **XML:** XML 오브젝트가 사용한 메모리입니다.
 - **MV_Other:** 위 카테고리에 포함되지 않는 다른 Movie View 메모리입니다.
 - **VM:** ActionScript 가상 머신에 할당된 메모리입니다. 이 카테고리는 아래와 같이 세부 분류됩니다.
 - **AS3 VM:** ActionScript 3 가상 머신이 사용한 메모리 중 다음 하위 부류에 해당되지 않는 메모리입니다.
 - **CallFrame:** ActionScript 실행 중 호출 스택의 각 항목에 할당된 메모리입니다.
 - **VTable:** ActionScript 가상 테이블이 사용한 메모리입니다.
 - **SlotInfo:** ActionScript 클래스 멤버를 설명하는 구조에 할당된 메모리입니다. ClassTraits 및 InstanceTraits 는 각각 클래스 또는 인스턴스 메모리 레이아웃을 설명하는 SlotInfo 구조를 포함합니다. ClassTraits 및 InstanceTraits 에 대한 자세한 정보는 아래를 참조하십시오.
 - **SlotInfoHash:** ActionScript 의 속성 검색 속도를 높이는 해쉬(hash) 테이블이 사용한 메모리입니다.
 - **ClassTraits:** ActionScript 클래스의 메모리 레이아웃을 유지하는데 할당된 메모리입니다.
 - **Class:** ActionScript 오브젝트의 정적 멤버를 유지하는데 할당된 메모리입니다.

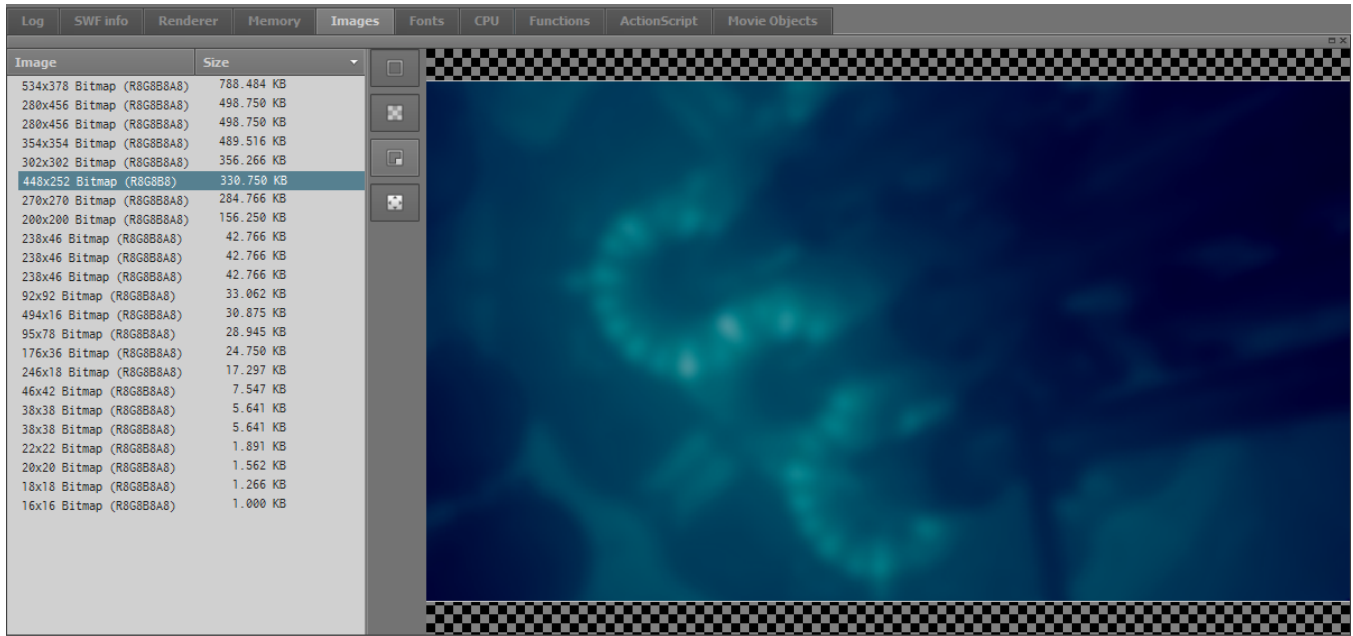
- **InstanceTraits:** ActionScript 오브젝트의 비정적 메모리 레이아웃을 유지하는데 할당된 메모리입니다.
- **Instance:** ActionScript 오브젝트의 인스턴스 변수 및 메소드를 유지하는데 할당된 메모리입니다.
- **AbcFile:** 바이트 코드 및 해당 데이터를 대표하는 데이터에 할당된 메모리입니다.
 - **AbcConstPool:** 숫자, 문자열, Namespaces 및 Multinames 과 같은 상수(constants) 저장에 사용되는 메모리입니다. 바이트 코드이므로 위의 AbcFile 에 포함됩니다.
- **VMAbcFile:** AbcFiles 에서 생성된 최적화된 바이트 코드, 파일간 종속성 및 다른 정보를 저장하는 내부 데이터 구조를 유지하는 메모리입니다.
- **Tracer:** 코드 최적화에 할당된 메모리입니다.
- **IME:** Input Method Editor (IME) 지원에 사용된 메모리입니다.

Log	SWF info	Renderer	Memory	Image
		Memory		
		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	

그림 8: Memory Summary

각 카테고리를 확장하면 하위 카테고리 또는 만약 있을 경우 해당 카테고리 내의 직접 할당을 책임지는 메모리 힙이 표시됩니다. 이러한 방법으로 계속 확장하면 특정 프레임에서의 Scaleform 메모리 사용을 좀 더 이해하는 데 도움이 됩니다.

Images 탭은 선택된 프레임에서 불러온 이미지와 함께 크기를 표시하므로 마찬가지로 메모리 프로파일링과 관련이 있습니다.



크기 9: Images 메모리 탭

대형 이미지가 출력되어 대상 Scaleform 응용 프로그램의 메모리 사용이 증가하였는지를 신속히 확인하려면 이 탭을 사용합니다. . 이미지 목록에서 항목을 클릭하면 미리보기 창에 해당 이미지의 축소판 그림이 표시됩니다. 이미지 데이터는 프로파일링 정보의 일부로 전송되지 않으며 요청 시 전송됩니다. 따라서 이미지가 Scaleform 에 의해 이미 내려진 경우 이미지가 표시되지 않을 수 있습니다.

메모리 프로파일링과 관련된 마지막 탭은 선택한 프레임의 폰트 정보를 표시하는 Fonts 탭입니다.



그림 10: Fonts 메모리 탭

이 탭을 사용하여 SWF 파일 별로 각 폰트 형식을 위해 임베드된 그림의 숫자를 검사합니다.

임베드된 폰트는 상당한 양의 메모리를 사용할 수 있으며, MovieData 힙과 CharDefs 카테고리의 증가된 크기에 의해 AMP 에 표시됩니다. 이러한 항목은 각각 Heaps 와 Memory Summary 탭에서 검사할 수 있습니다. 여러 SWF 파일에서 폰트를 공유하여 폰트 메모리를 줄일 수도 있습니다.

2.5 오버드로우 프로파일링사용하기

AMP 는 프로파일된 응용 프로그램에서 잠재적인 문제 영역을 강조하는 특수 렌더링 모드를 원격으로 시작하는 데 사용할 수 있습니다. 이 렌더링 모드의 활성화 상태를 변경하려면 응용 프로그램 제어 톨바에 있는 해당 버튼을 클릭하십시오.

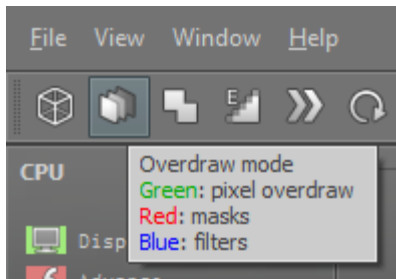


그림 11: 전환 버튼

이 모드가 활성화되면 모든 오브젝트가 녹색으로 렌더링됩니다. 녹색은 픽셀 오버드로우(최대 밝기는 16 레이어), 마스크는 적색, 필터는 청색(최대 밝기는 8 레이어)입니다. 이들 색깔은 서로 혼합되며, 따라서 연한 황색 영역은 마스크와 포개진 상위 오버드로우가 됩니다.

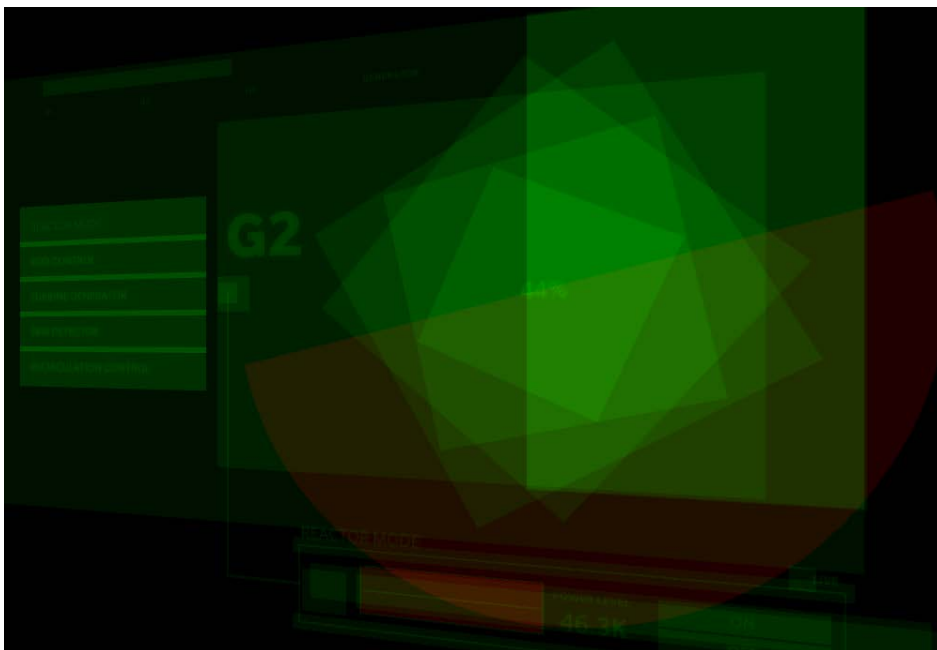


그림 12: Highlighting 을 위한 렌더링 모드

오버드로우, 마스크, 필터는 모드 성능에 영향을 끼치므로 가능하다면 최소화해야 합니다.

2.6 배치 프로파일링 사용

AMP 가 지원하는 또 다른 특별한 렌더링 모드가 배치 프로파일링입니다. 이 렌더링 모드의 활성화 상태를 변경하려면 응용 프로그램 제어 툴바에 있는 해당 버튼을 클릭하십시오.

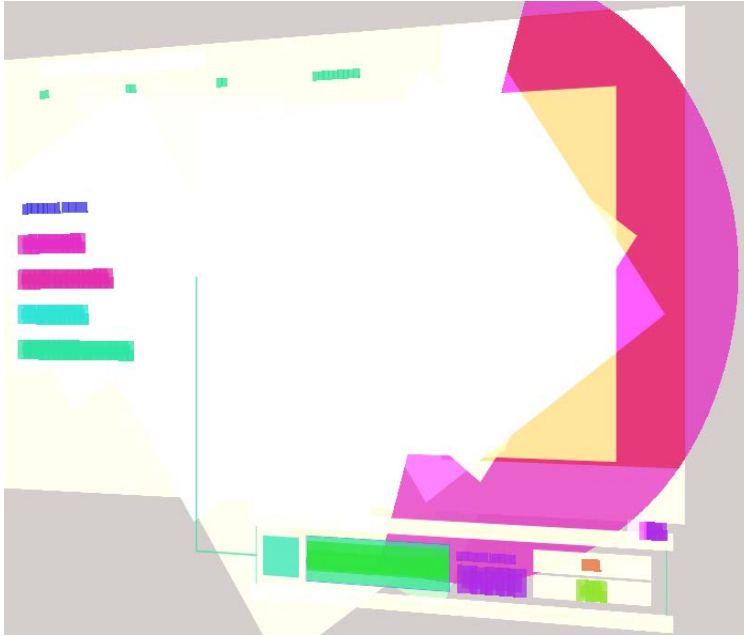


Figure 13: Batch profiling mode

이 모드가 활성화되면 각 렌더링 배치가 다른 색상으로 렌더링됩니다. 가능할 때마다 배치 수를 최소화하여 최상의 성능을 구현합니다.

3 플랫폼 및 통합 관련 주의사항

1. AMP 는 일반적으로 Gfx::System 오브젝트 구성 또는 Gfx::System::Init 기능 호출 중에 초기화됩니다. 또는 Scaleform::System 이 Scaleform 을 초기화하는 데 대신 사용되는 경우, 프로그래머는 Gfx::AMP::Server::Init() 및 Gfx::AMP::Server::Uninit()를 각각 호출하여 AMP 를 명시적으로 초기화하고 종료해야 합니다.
2. AMP 서버는 현재 Windows, Xbox360, PS3, Linux, Mac, Wii, Android 및 iPhone/iPad, 3DS, PS Vita, WiiU 를 지원합니다. AMP 클라이언트의 자동 검색 기능을 사용하여 바로 연결이 가능합니다. 하지만 자동 검색을 사용하려면 AMP 서버의 IP 주소를 알아야 하며, 여러 IP 주소를 가진 플랫폼의 경우 혼란이 있을 수 있습니다.
 - Xbox360 의 경우 디버그용 IP 주소가 아닌 타이틀의 IP 주소를 지정해야 합니다. 타이틀 IP 주소는 Xbox 360 Neighborhood 의 테스트 키트 또는 개발 속성에서 확인할 수 있습니다.
 - PS3 의 경우 개발 키트가 두 개의 IP 주소를 사용하도록 설정되어 있다면 디버그용 IP 주소가 아닌 응용 프로그램 IP 주소를 지정합니다.
 - Android 의 경우 응용 프로그램 레벨에서 활성화하기 위해 인터넷 권한이 필요합니다. 이를 위해 다음을 응용 프로그램의 AndroidManifest.xml 에 추가해야 합니다.

```
<uses-permission android:name="android.permission.INTERNET" />
```
 - WiFi 연결이 아닌 호스트 PC 브리지를 통해 WiiU 에 연결하십시오. 호스트 PC 가 포트 6003 을 자동으로 발견하고 콘솔을 감지합니다. 연결 요청이 완료되면 TCP/IP 소켓에서 호스트 브리지로 통신이 전환됩니다.
 - Wii 는 AMP 와 연결할 수 있는 네트워크 소켓을 사용하지 않습니다. 대신 NDEV 에 내장된 USB 인터페이스(COM 포트)를 통해 본체를 PC 에 직접 연결할 수 있습니다. IP 주소 필드에서 "wii"를 지정하고 연결 대화 상자에서 Wii AMP 서버로 연결할 수 있습니다.
주의: RVL_SDK/X86/bin 이 환경변수 경로에 있는지 확인하세요. 그러면 hio2.dll 은 AMP 에 의해 로드될 것입니다.
3. Unreal Engine 3 와 같은 일부 게임 엔진은 AMP 서버가 검색된 Flash 콘텐츠에 대응하는 SWD/AS3 AS 파일을 찾지 못하도록 자산을 패키징합니다. 이런 경우에는 AMP 가 SWF 와 같은

디렉터리에서 AS2 SWD 파일을 찾아 네트워크로 보내도록 두지 말고 SWD/AS 파일을 AMP 네트워크 디렉터리로 옮겨야 합니다.

4 빌드 주의사항

1. AMP 는 기본적으로 모든 non-shipping Scaleform 빌드의 일부로 구성되어 있습니다.
Scaleform 에서 AMP 를 제거하려면 GfxConfig.h 에서 SF_AMP_SERVER 를 정의하지 않고 Gfx 라이브러리를 다시 빌드해야 합니다. 만일 Scaleform 소스로 액세스가 불가능한 경우 다음을 호출하여 AMP 를 비활성화합니다.

```
Ptr<Gfx::AMP::ServerState> serverState = *SF_NEW  
Gfx::AMP::ServerState();  
serverState->StateFlags &= Gfx::AMP::Amp_Disabled;  
AmpServer::GetInstance().UpdateState(serverState);
```
2. AMP 는 네트워크 소켓을 통해 프로파일러와 연결됩니다. 그러므로 응용 프로그램이 적절한 플랫폼별 네트워크 라이브러리와 연결되어 있는지 확인이 필요합니다(Windows: Ws2_32.lib, Xbox 360: Xnet.lib, PS3: libnetctl, Linux: 소켓).
3. AMP 가 네트워크 소켓을 사용하는 유일한 시스템인 경우, AMP 는 소켓 라이브러리를 초기화했다가 작업이 끝나면 이를 해제합니다. 일부 플랫폼의 경우 라이브러리가 해제된 횟수보다 더 많이 초기화되면 라이브러리를 해제해도 영향이 없습니다. 이런 경우에는 아무런 문제가 없습니다. 하지만 다른 플랫폼은 네트워크 라이브러리가 해제된 횟수보다 더 많이 초기화되었다 해도 라이브러리를 해제하면 바로 효과가 나타나며, 이로 인해 AMP 가 응용 프로그램의 다른 부분에서 사용하는 소켓 연결과의 간섭을 일으킬 수 있습니다. 이런 경우 다음을 호출하여 AMP 가 소켓 초기화 과정을 건너뛰고 이전에 초기화한 소켓 라이브러리를 사용하도록 강제할 수 있습니다. `AMP::Server::GetInstance().SetInitSocketLib(false)`. 이 호출은 AMP 가 초기화되기 전에 실행되어야 합니다.
4. AMP 메모리 힙에 제한을 설정할 수 있습니다. 제한을 초과할 경우 Scaleform 는 대기 메시지가 프로파일러에 전송될 때까지 작동을 중지합니다. 시간 제한은 `AMP::Server::GetInstance().SetHeapLimit()` 호출을 통해 설정할 수 있습니다.
5. AMP 는 기본적으로 포트 7534 에 연결 대기 소켓을 생성합니다. 다음을 호출하여 다른 포트로 설정할 수도 있습니다. `AMP::Server::GetInstance().SetListeningPort()`;

6. AMP 가 프로파일러 클라이언트와의 연결이 완료될 때까지 실행을 멈추도록 설정할 수 있습니다. 이 기능은 프로그램 시작 시 통계를 구할 때 유용하며, 구동 시 다음을 호출하여 설정할 수 있습니다. `AMP::Server::GetInstance().SetConnectionWaitTime.`

5 AMP 지원 추가

모든 Scaleform 응용 프로그램은 다음 단계를 통해 AMP 를 통한 원격 프로파일링 지원이 가능하도록 설정할 수 있습니다. 또한 설정이 완료된 AMP 서버의 한 예시로써 Scaleform Player 소스를 확인할 수도 있습니다.

5.1 Debug 파일 생성

AMP 에는 ActionScript 2 Flash 디버그 정보(SWD 파일)를 사용하여 소스 코드 및 라인별 시간 측정을 표시하는 기능이 있습니다. Flash CS3 또는 CS4 에서 디버거(Ctrl+Shift+Enter)를 실행하여 SWD 파일을 생성합니다. SWD 파일을 해당 SWF 파일과 같은 위치 또는 AMP 클라이언트의 작업 디렉토리에 둡니다.

ActionScript 3 의 경우 SWD 파일이 없으며 디버그 정보가 바로 SWF 에 포함됩니다. 소스 코드를 표시하려면 AS 파일이 필요합니다. Flash Studio 내에서 디버거를 실행하여 디버그 SWF 파일을 생성합니다.

5.2 응용 프로그램 제어

AMP 는 와이어 프레임 모드나 배치 프로파일링 모드처럼 이미 프로파일된 응용 프로그램의 설정을 원격에서 제어할 수 있습니다. 지원되는 설정은 대부분 응용 프로그램별로 세부화되어 있으며, 응용 프로그램에서 실행할 수 있습니다. 사용하도록 설정되지 않은 기능은 AMP 클라이언트에서 작동하지 않습니다.

AMP 에서 보낸 응용 프로그램 제어 메시지는 다음 절차에 따라 처리할 수 있습니다.

- `AMP::AppControlInterface` 에서 얻은 사용자 정의 응용 프로그램 제어 콜백 클래스를 실행하고, AMP 에서 보낸 `AMP::MessageAppControl` 메시지를 처리하는 `HandleAmpRequest` 메소드를 오버라이드합니다.

- 시작할 때 `AMP::Server::GetInstance().SetAppControlCallback` 을 호출하여 사용자 정의 핸들러를 설치합니다.
- `AMP::Server::GetInstance().SetAppControlCaps` 를 호출하여 AMP 에 지원되는 기능을 전달합니다. 이 메소드의 전달 인자는 `AMP::MessageAppControl` 메시지이며, `true` 로 설정된 지원 기능을 포함합니다. 예제:

```
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 연결 상태

AMP 클라이언트의 상태 표시줄에는 연결된 응용 프로그램의 이름이 표시됩니다.

`AMP::Server::GetInstance().SetConnectedApp` 호출을 통해 이 정보의 통신이 이루어집니다.

AMP 클라이언트는 응용 프로그램의 현재 상태를 표시할 수 있지만, 이를 위해서는 응용 프로그램의 상태가 변경될 때마다 해당 정보를 전달 받아야 합니다. 다음 메소드를 사용하여 상태를 업데이트합니다.

- 다음 `ServerStateType` 열거형 중 하나를 첫 번째 전달 인자로 사용하는 `AMP::Server::GetInstance().SetState`.
 - `Amp_RenderOverdraw`

- Amp_App_Wireframe
- Amp_App_Paused
- Amp_App_FastForward
- AMP::Server::GetInstance().SetAaMode
- AMP::Server::GetInstance().SetStrokeType
- AMP::Server::GetInstance().SetCurveTolerance
- AMP::Server::GetInstance().SetCurveTolerance
- 또는 AMP::ServerState 오브젝트에 위의 정보를 입력하고
AMP::Server::GetInstance().UpdateState 로 전달할 수도 있습니다.

5.4 마커

AMP 는 프로파일된 응용 프로그램에서 C++ 또는 ActionScript 호출에 대응하는 CPU 그래프에 마커라는 특별한 표시자를 표시할 수 있습니다. ActionScript 에서 마커를 추가하는 방법은 다음과 같습니다.

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

참고: Amp.addMarker() 코드가 포함된 .as 파일을 컴파일하면 컴파일 오류가 일어날 수 있습니다. 오류를 방지하기 위해서는 Resources/AS2/CLIK 디렉토리 하위에 있는 내장 Amp 클래스를 사용합니다.

C++ 에서 마커를 추가하는 방법은 다음과 같습니다.

```
GFX::MovieImpl:: AdvanceStats->AddMarker(1)
```

호출이 이루어지면 AMP 는 해당 프레임에 마커를 표시합니다. 차후에는 한 종류 이상의 마커를 표시할 수 있도록 정수 인수가 사용될 예정입니다.

6 AMP 관련 FAQ

이 섹션은 AMP 도구를 사용할 때 생길 수 있는 몇 가지 질문을 포함합니다.

1. 게임 측에서 AMP 를 사용하는 데 필요한 최저 사양은 무엇입니까?

GFX_AMP_SERVER 를 정의한 빌드 등 AMP 가 활성화된 빌드가 필요합니다. 출시하지 않는 Gfx 빌드는 모두 기본적으로 AMP 가 활성화되어 있습니다. Platform::RenderThread::drawFrame 을 호출하지 않는 게임에서는 AmpServer::GetInstance().AdvanceFrame 을 호출하여 프레임마다 AMP 가 업데이트되도록 해야 합니다.

2. AmpClient 를 사용하여 게임을 연결하면 게임 프레임률이 현저하게 떨어집니다. 조사해보니 AmpServer::AdvanceFrame()을 실행하면 시간이 오래 걸리는데 이게 정상입니까?

프로파일러를 AMP 에 연결하면 성능 저하가 발생하는 것이 정상입니다. 이 성능 저하를 최대한 줄이려면 AMP 클라이언트 도구 모음에서 적절한 드롭다운 컨트롤을 선택하여 낮은 프로파일링 수준에서 시작하십시오. 그런 다음 성능 저하가 발생하는 부분을 찾아 그 부분의 프로파일링 수준을 높여 문제의 원인을 좀 더 자세하게 파악할 수 있습니다. 프레임마다 전체 메모리를 보고하면 심각한 성능 저하를 불러올 수 있으므로 보고서는 볼 필요가 있을 때만 생성하십시오. AMP 클라이언트 도구 모음에서 "i"(소문자 I)를 통해 세부 메모리 보고 기능을 켜거나 끌 수 있습니다.

3. Windows/PS3/Xbox 에서 게임을 실행합니다. 검색창에서 AMP 클라이언트를 검색해 AMP 에 연결할 수는 있습니다. 그런데 통계 정보가 전혀 없습니다.

AMP 가 메모리와 성능 통계를 표시하려면 프레임마다 정보를 받아야 합니다. 해당 게임이 Platform::RenderThread::drawFrame 을 호출하지 않으면 AmpServer::GetInstance().AdvanceFrame 을 호출하여 프레임마다 AMP 가 업데이트되도록 해야 합니다.

4. (AMP::Server::WaitForAmpConnection 을) 업데이트할 때 왜 AMP 가 일 초 이상 멈춥니까?

일반적으로 AMP 서버가 메모리를 너무 많이 사용하면 그동안 누적된 데이터가 클라이언트에게 전송될 수 있도록 잠시 작동을 멈춥니다. 계속 이런 현상이 발생한다면 사용자의 콘텐츠가 너무 많은 프로파일 정보를 유발하여 AMP 가 클라이언트에게 데이터를 모두 전송할 시간이 충분하지 않아 Scaleform 이 일시적으로 중단되는 것일 수 있습니다.

AMP 클라이언트의 'i' 버튼으로 소스 라인 타이밍과 세부 메모리 보고 기능을 끄면 전송되는 정보의 양을 줄일 수 있습니다. AMP 클라이언트에서 'profile level' 드롭다운 메뉴를 사용하여 프로파일링 수준을 낮출 수도 있습니다. 마지막으로 문제를 유발하는 메모리 임계값을 늘릴 수 있습니다. 기본 임계값은 1MB 이지만 Gfx::AMP::Server::SetHeapLimit 을 호출하면 이 크기를 변경할 수 있습니다.

5. 프로파일러를 사용할 때 예상되는 메모리 오버헤드는 어느 정도입니까?

AMP 의 힙 제한은 기본 1MB 이지만 고정된 값은 아닙니다. 그러므로 이 값은 제한을 약간 넘을 수 있습니다. 게임에서 힙 제한을 변경하려면 AmpServer::SetHeapLimit 을 호출하십시오. 제한치가 클수록 AMP 클라이언트로 데이터를 전송하는 동안 Gfx 가 잠시 중지될 확률이 줄어듭니다.

7 추가 정보

AMP 에 대한 보다 자세한 정보는 다음 리소스를 참조하십시오.

- [AMP Use Cases](#) 문서.
- [Scaleform Reference 문서](#) - 문서 - 코드/클래스에 대한 자세한 정보
- [AMP Forum](#) - 커뮤니티 질문 및 지원