

在安卓设备上 对移动游戏进行 性能分析和优化

Remi Breton

乐逗游戏



GAME DEVELOPERS CONFERENCE™ CHINA

SHANGHAI INTERNATIONAL CONVENTION CENTER

SHANGHAI, CHINA · OCTOBER 19-21, 2014

为什么要进行分析与优化

- 访问多个设备和用户
- 减少安卓安装包(APK)大小来增加下载量
- 提高游戏性能和改善加载时间
- 找出性能瓶颈
- 优化和减少开发工作量







中国市场 概况

中国排名前20位的设备

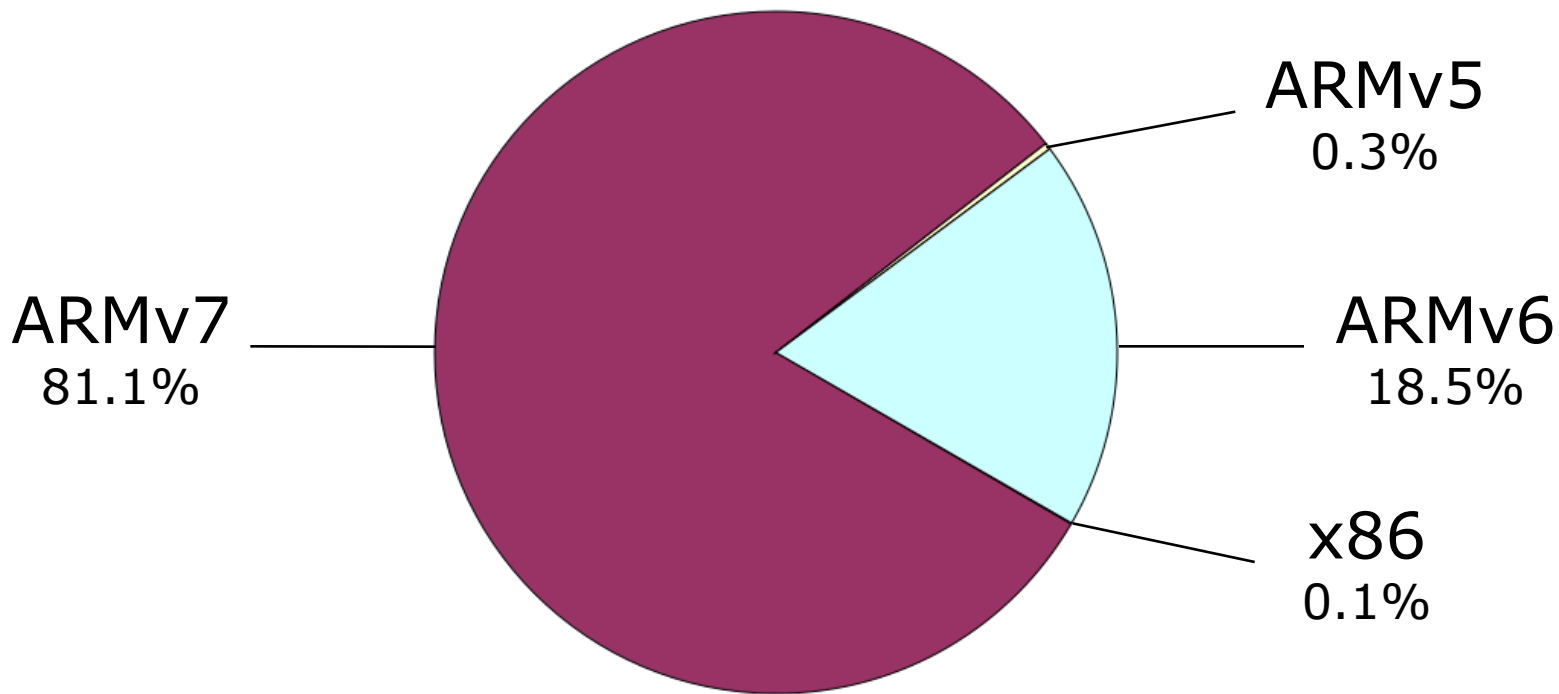
三星 - Galaxy S	三星 - Galaxy Note 2
三星 - Galaxy S 2	三星 - Galaxy Note
HTC - Desire	魅族 - M9
HTC - Desire HD	谷歌 - Galaxy Nexus
三星 - Galaxy S 3	小米 - Mi-One
三星 - Galaxy Ace	HTC - Wildfire S
HTC - Desire S	三星 - Galaxy SL
摩托罗拉 - Defy	三星 - Galaxy Tab
HTC - Incredible S	现代 - S800
HTC - Wildfire	三星 - Galaxy Gio

* 地瓜游戏中心在2011年到2013年之间收集的数据

图形处理器 (GPU) 供应商

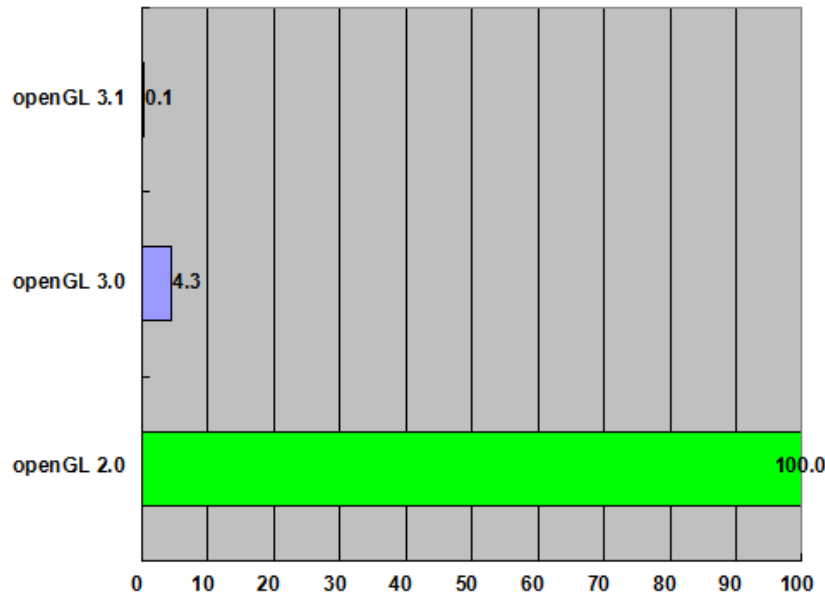
Adreno	-	44.7 %	
PowerVR	-	26.7 %	
Mali	-	19.1 %	
Tegra	-	1.6 %	
Vivante	-	1.6 %	
VideoCore	-	0.5 %	
* 软件渲染器	-	5.8 %	

中央处理器(CPU)指令集架构

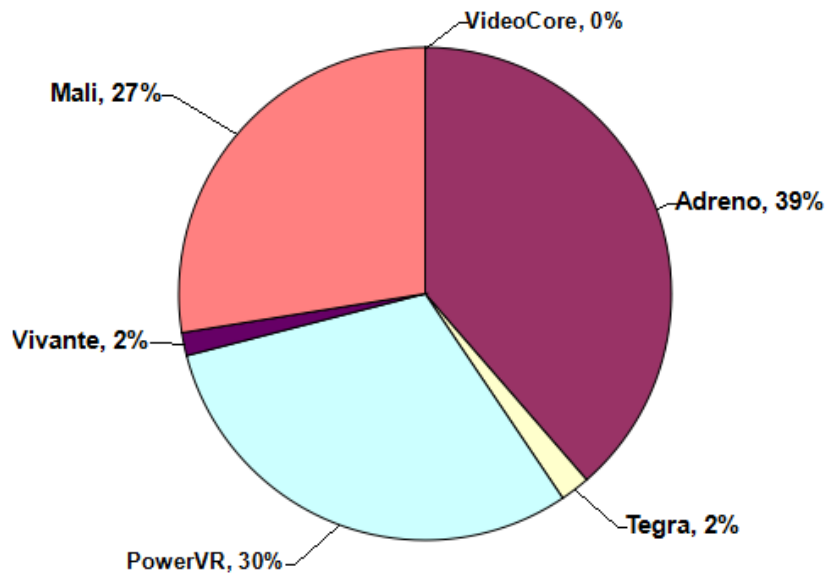


ARMv7 – 图形

OpenGL ES

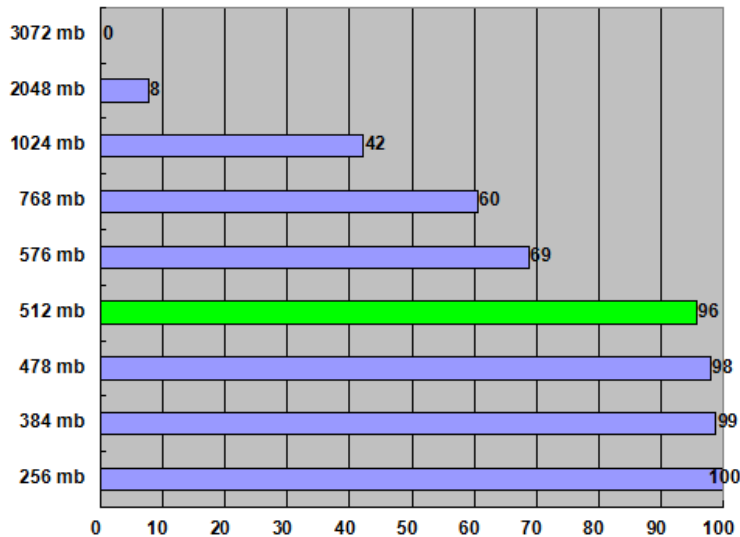


图形处理器

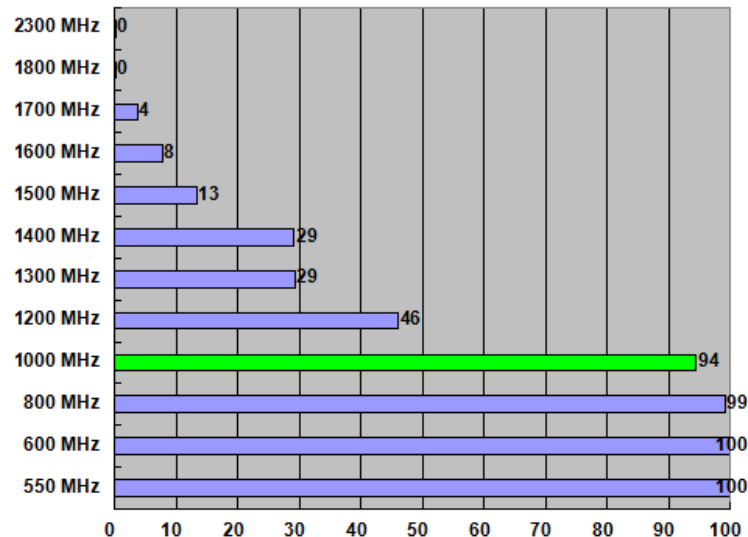


ARMv7 – 规格

随机存取存储器（RAM）



中央处理器（CPU）



* 97%的1000 MHz总线为单核CPU

ARMv7 目标设备

OpenGL ES : **2.0**

物理内存: **512 mb**

CPU内核 : 单核心

CPU速度 : **1000 MHz**

安卓API : **2.3**

覆盖中国74%的设备

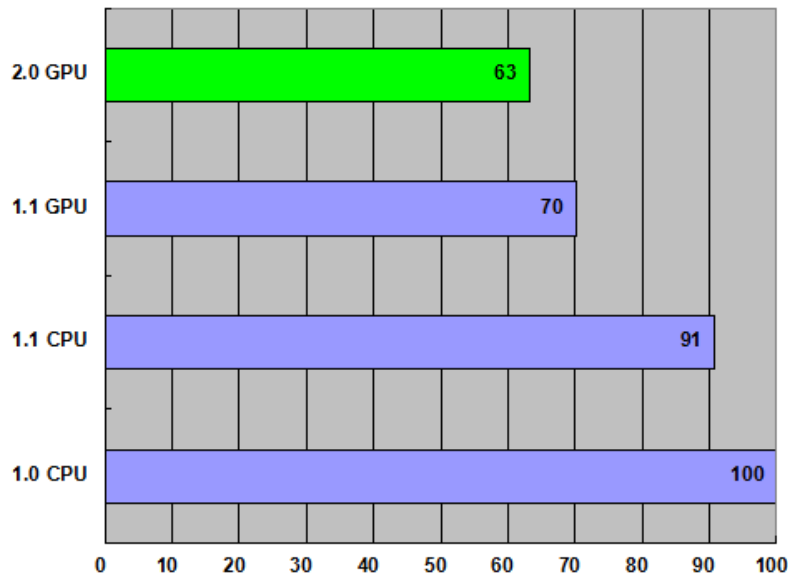
覆盖中国91%的ARMv7设备

前9位

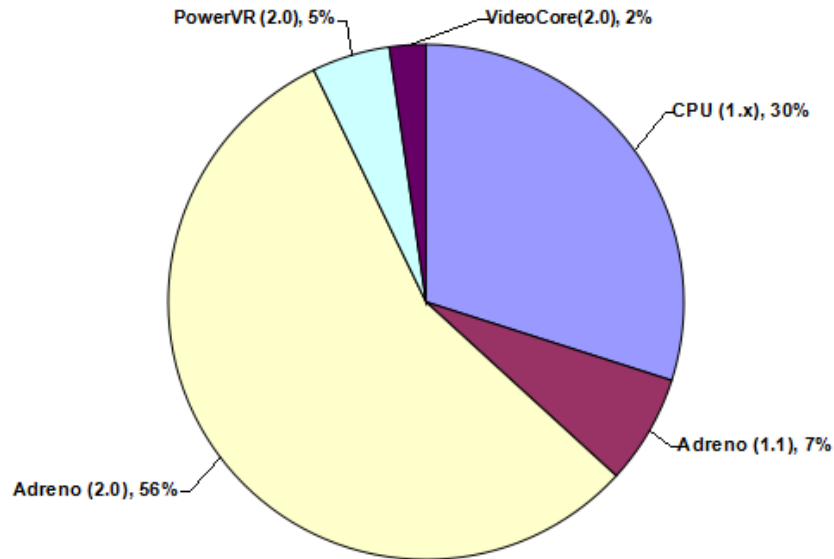
- 三星 **Galaxy S**
- 三星 Galaxy S2
- 三星 Galaxy S3
- 三星 Galaxy Note
- HTC Desire
- HTC Desire HD
- HTC Desire S
- HTC Incredible S
- 魅族 **M9**

ARMv6 – 图形

OpenGL ES

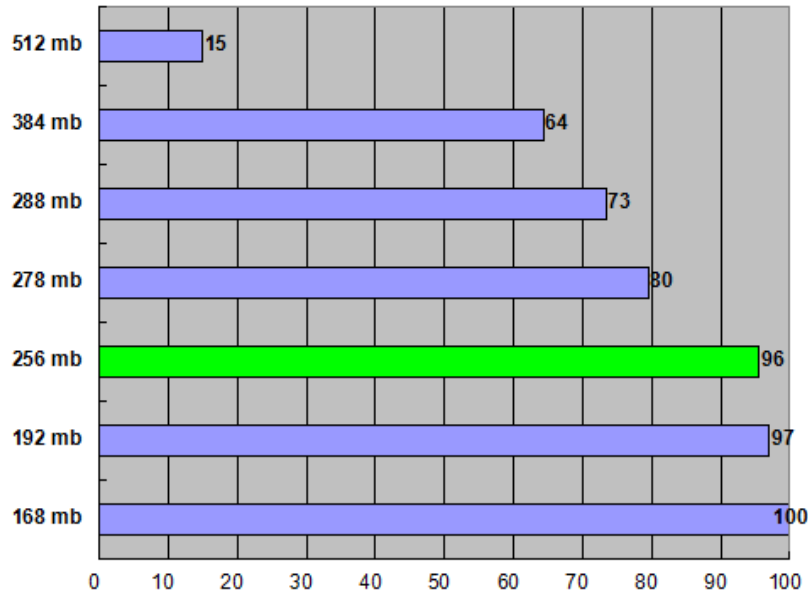


图形处理器供应商

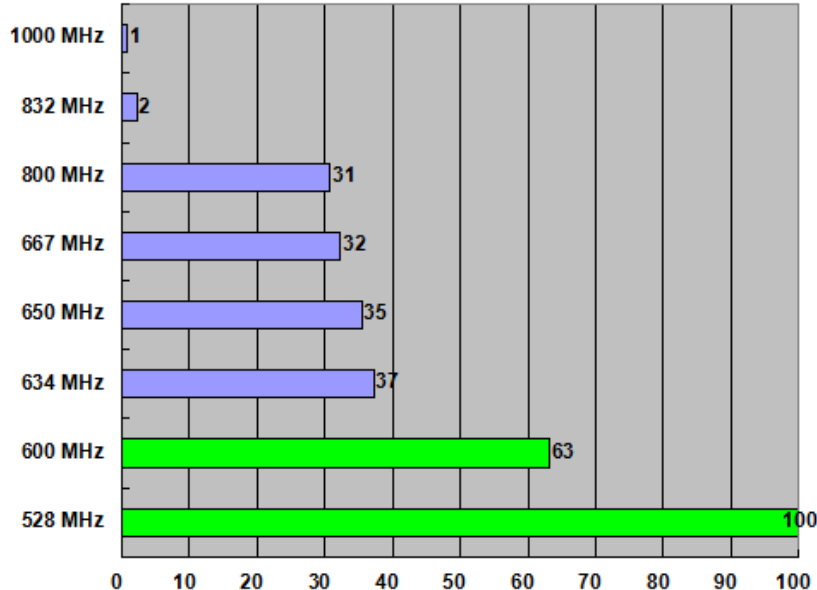


ARMv6 – 规格

随机存取存储器（RAM）



中央处理器（CPU）



ARMv6目标设备(高)

OpenGL ES : **2.0**

物理内存 : **256 mb**

CPU内核 : 单核心

CPU速度 : **600 MHz**

安卓API : **2.3**

覆盖中国93%的设备

覆盖中国60%的ARMv6设备

前**8**位

- 三星 Galaxy Ace
- 三星 Galaxy Gio
- 三星 Galaxy Mini
- **三星 Galaxy 550**
- HTC Wildfire S
- HTC Legend
- 东信MTW20
- **华为Sonic**

ARMv6目标设备(低)

OpenGL ES : **1.1**

物理内存 : **256 mb**

CPU内核 : 单核心

CPU速度 : **528 MHz**

安卓API : **2.2**

前**6**位

- HTC Wildfire *CPU
- HTC Hero *CPU
- HTC Magic *GPU
- **华为 C8500** *CPU
- 摩托罗拉 Backflip *GPU
- a摩托罗拉 Cliq *GPU

覆盖中国99%的设备

覆盖中国95%的ARMv6设备

* 75%的CPU渲染

Unity支持

	中央处理器指令集			OpenGL ES				安卓API	
	ARMv6	ARMv7	x86	1.x	2.0	3.0	3.1	最小	最大
Unity 3.5	●	●		●	●			2.0.1	4.2
Unity 4.0		●		●	●			2.0.1	4.2
Unity 4.1		●		●	●			2.0.1	4.2
Unity 4.2		●		●	●	●		2.0.1	4.2
Unity 4.3		●			●	●		2.3.1	4.3
Unity 4.5		●			自动	自动		2.3.1	4.4
Unity 4.6		●	*		自动	自动		2.3.1	4.4
Unity 5.0		●	*		自动	自动	?	2.3.1	4.4

了解 内存

RAM内存分配

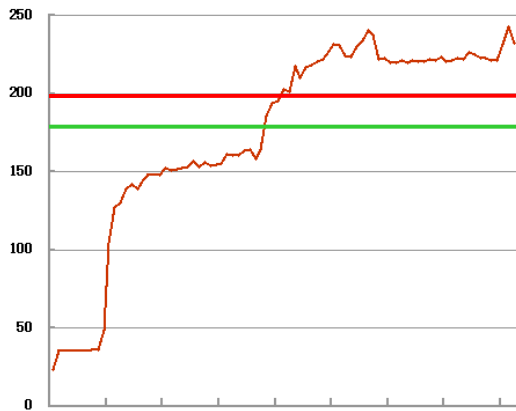
256 mb	512 mb	- 物理内存(统一)
~54 mb	~130 mb	- Linux内核
~72 mb	~72 mb	- 安卓操作系统进程(隐藏)
~34 mb	~120 mb	- 进程(持续&可见) <ul style="list-style-type: none">- 系统- android.process.acore- com.android.sysmui- com.android.phone- com.android.smspush- com.android.nfc
~95 mb	~190 mb	- 可用



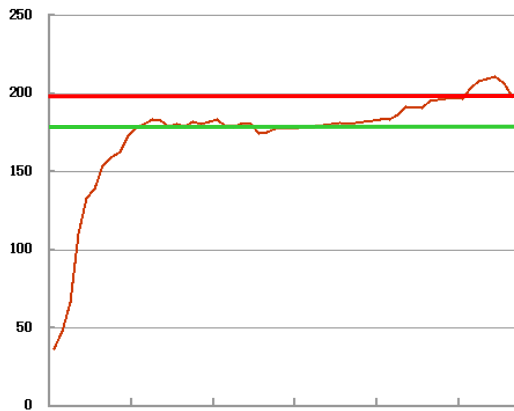
内存占用

同样的游戏在不同的设备上(安卓安装包大小为35mb)

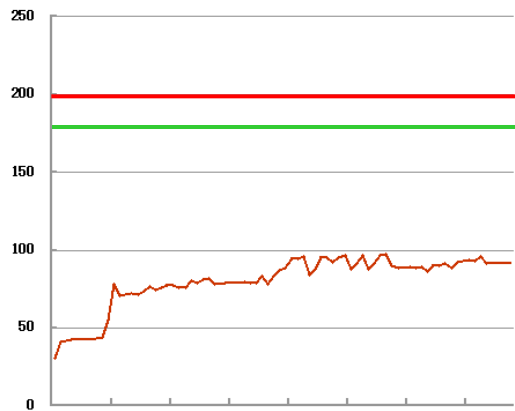
Adreno



PowerVR



Tegra



Shell命令

- `dumpsys meminfo` (所有进程和进程类)
- `dumpsys meminfo <PID>` (目标进程)
- `cat /proc/meminfo` (整个设备)
- `procrank` (包括系统进程在内的所有进程) *并不总是可用进程

- 可以通过USB远程桌面执行

```
adb shell dumpsys meminfo
```

- 可以记录到文件或剪贴板

```
adb shell dumpsys meminfo >FileName.txt
```

```
adb shell dumpsys meminfo |Clip
```

*返回数据会根据设备安卓版本的不同而有所不同

了解 纹理

OpenGL ES内部纹理格式

内部纹理格式	R	G	B	A	L	D	S	Bits	Unity
ALPHA8				8				8	Alpha 8
LUMINANCE8					8			8	
LUMINANCE4_ALPHA4				4	4			8	
LUMINANCE8_ALPHA8				8	8			16	
RGB565	5	6	5					16	RGB 16 Bit
RGB8	8	8	8					24	RGB 24 Bit
RGB10	10	10	10					30	
RGBA4	4	4	4	4				16	RGBA 16 Bit
RGB5_A1	5	5	5	1				16	
RGBA8	8	8	8	8				32	RGBA 32 Bit
RGB10_A2	10	10	10	2				32	
DEPTH24_STENCIL8						24	8	32	深度(玩家设置)
DEPTH_COMPONENT16						16		16	深度(玩家设置)
DEPTH_COMPONENT24						24		24	
DEPTH_COMPONENT32						32		32	

固定速度的纹理

- 100% • ETC1> 所有OpenGL ES 2.0 & 更高版本的图形处理器
- 45% • ATC> Adreno
- 27% • PVRT> PowerVR (iOS 100%)
- 7% • ETC2> 所有OpenGL ES 3.0 & 更高版本的图形处理器
- 3% • DXTC> Tegra & Vivante
- >1% • ASTC> 支持的 * OpenGL ES 3.0图形处理器 (苹果A8)

* Adreno 420 / Mali Midgard 2nd-Gen / Tegra K1 / PowerVR Series6XT / Vivante GC7000

RGBA 32

32 bpp

RGB ETC

4 bpp

RGB PVR

4 bpp

RGBA PVR

4 bpp

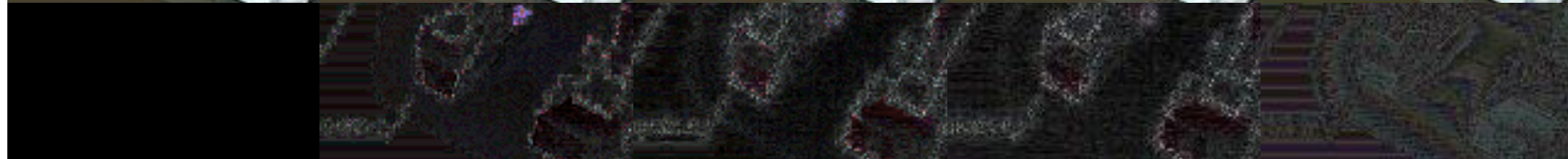
RGBA 16

16 bpp

颜色



比较



alpha



比较



大小

256 kb

(1 纹理 256kb)

64 kb

(2 纹理 32kb)

64 kb

(2 纹理 32kb)

32 kb

(1 纹理 32kb)

128 kb

(1 纹理 128kb)

着色器

修改与优化

- 经过修改需要“Split Alpha”的着色器
- 创建自定义着色器
- 使用Unity内置的“移动”着色器
- 减少指令集的着色器数量
- 避免“舍弃”固定功能(阿尔法测试)

限制

- Unity遗留的图形用户界面系统
- Unity遗留的字体系统

多次构建 创造

优点/缺点

优点

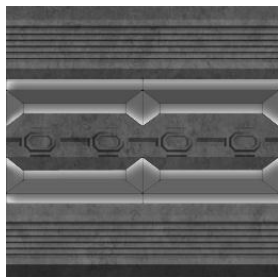
- 提高每个目标的性能
- 在更好的手机上实现更好的图形
- 每个目标的文件包更小

缺点

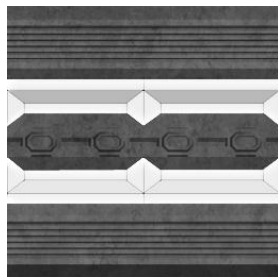
- 更多的资产创建和管理
- 更多的代码创建和管理
- 更多的着色器变化
- 构建分布的复杂性
- 更多的测试和调试

纹理 优化

颜色拆分Alpha通道



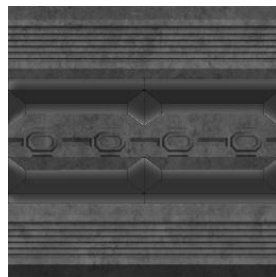
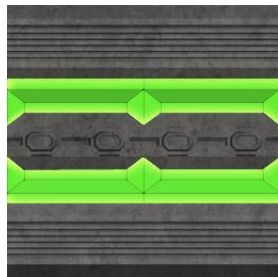
[R]



[G]



[R G B]



[B]



[A]



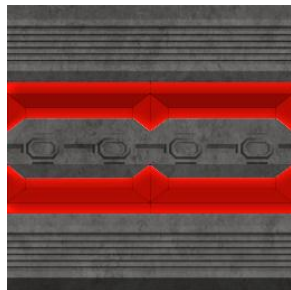
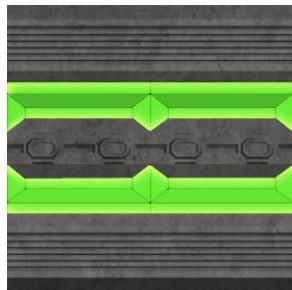
[R G B]



删除副本

前

颜色

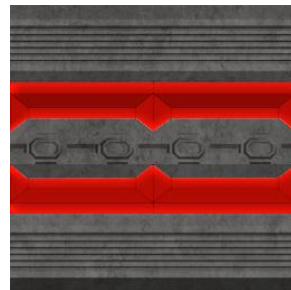
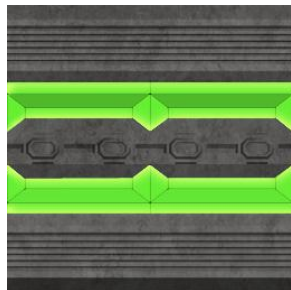


Alpha



使用的4种纹理

后



使用的3种纹理

调整纹理大小

根据设备屏幕大小调整纹理尺寸

颜色

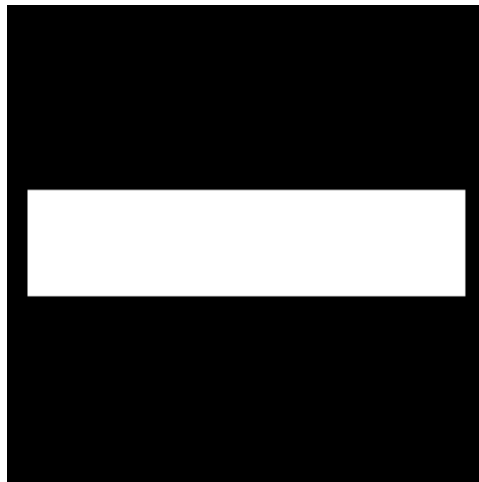
256 x 256



需要分辨率

Alpha

256 x 256



可以减少

Alpha

64 x 64



方形2次幂 (SPOT)

只使用SPOT纹理

32 x 32
64 x 64
128 x 128
256 x 256
512 x 512
1024 x 1024
2048 x 2048

转换非方形纹理

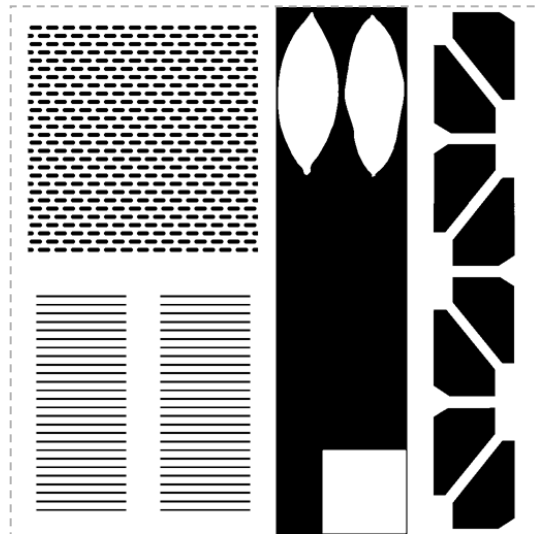
- 分割纹理
 - 创建新对象
- 合并纹理/创建贴图集
 - 改变对象UV坐标

减少文件包大小

[1024 x 1024]在内存当中总是采用512 kb



316 kb的安卓安装包



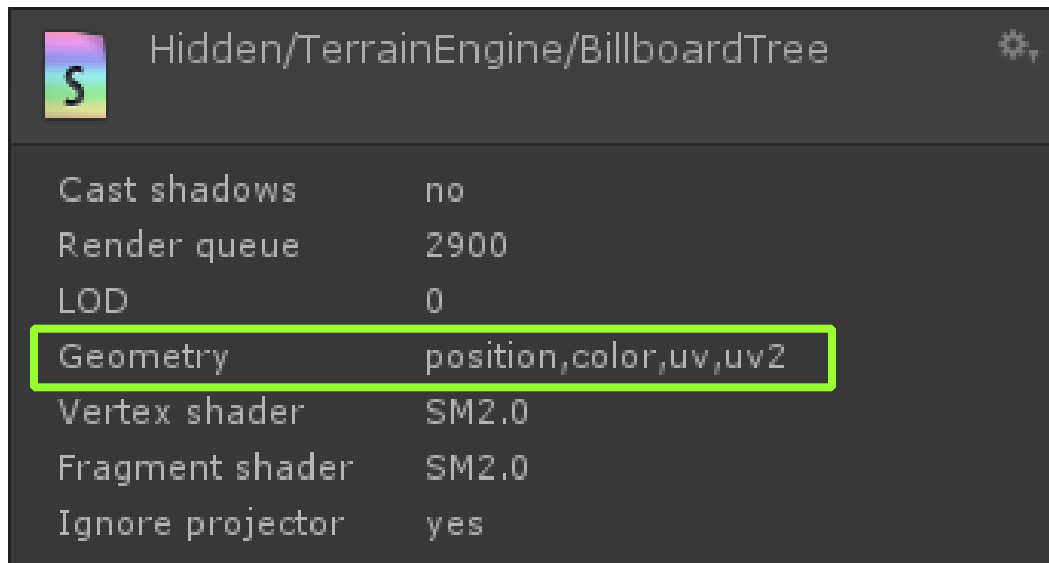
287 kb的安卓安装包

其他资产 优化

几何

数据缩减/删除

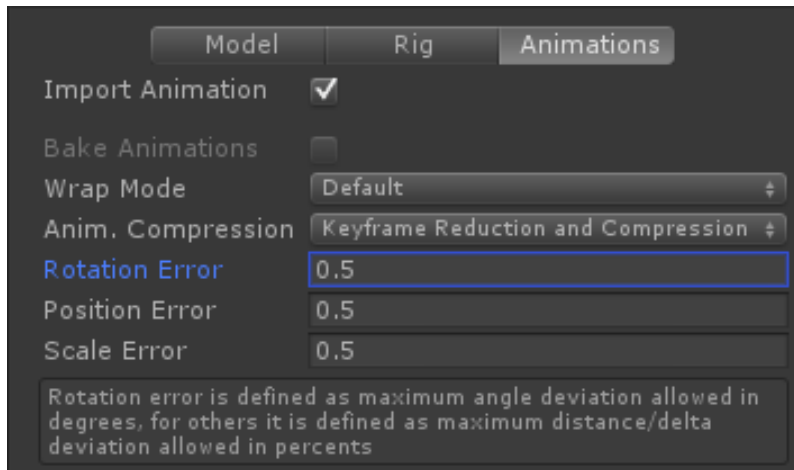
- 顶点索引
- 顶点颜色
- UV 1 / UV 2
- 正常
- 蒙皮权重



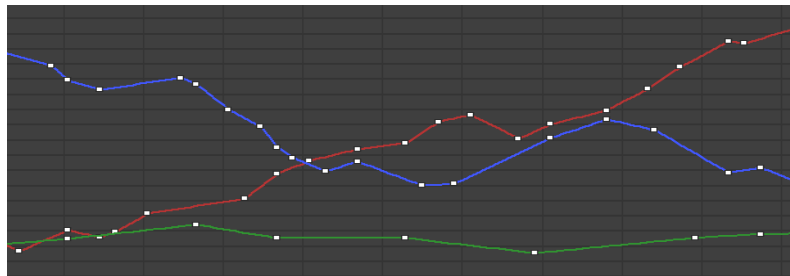
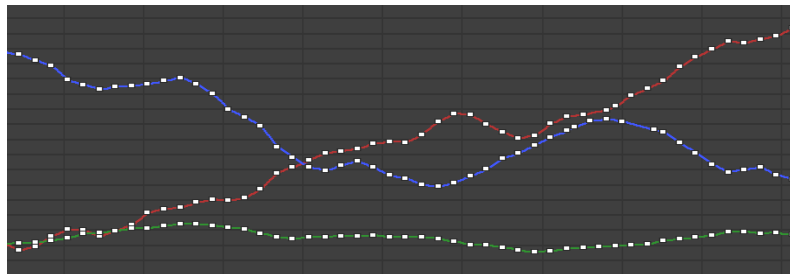
动画

键缩减

在Unity中



在FBX中



音频

视频

尺寸缩减

- 强制使用单声道
- 转换为OGG/MP3格式
- 减少比特率(kbps)

- 创建实时影像
- 转换为OGG格式
- 减少比特率(kbps)

(使用不同品质的视频/音频)

<http://developer.android.com/guide/appendix/media-formats.html>

性能分析

找出性能瓶颈

中央处理器

- 绘图调用太多
- 复杂的脚本或物理

顶点处理

- 顶点太多
- 每个顶点的计算太多

带宽

- 未压缩的大纹理
- 高分辨率的帧缓冲(framebuffer)

片元处理

- 片元太多, 超量绘制
- 每个片元的计算太多

优化类型

- 减少内存占用
 - 资产修改(纹理、音频、模型、动画...)
 - 代码修改(内存管理、内存泄露...)
- 解决性能瓶颈
 - CPU “中央处理器”
 - GPU “图形处理器”
- 减少文件包大小

分析工具

Adreno - Adreno Profiler

PowerVR - PVRTune / PVRTrace

Mali - Mali Graphics Debugger / ARM DS-5 Streamline

Tegra - PerfHUD / Tegra Graphics Debugger / Tegra System Profiler

Vivante - 无

VideoCore - 无

All - Android SDK tools (安卓软件开发工具包)

Unity - Unity Profiler(仅支持Unity专业版)

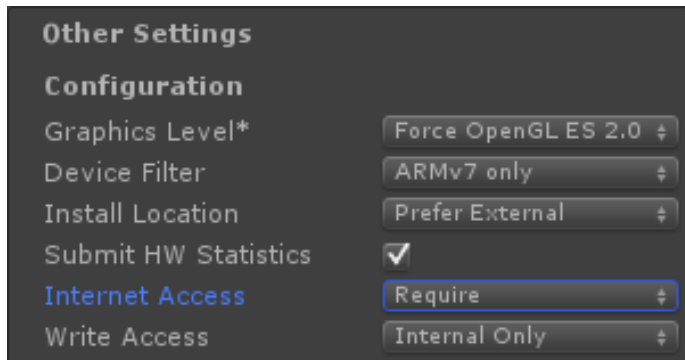
分析器要求

- 根设备（可选）
- USB调试启用（开发人员选项）
- 互联网权限启用的安卓安装包

- 在AndroidManifest.xml文件中

```
<uses-permission android:name="android.permission.INTERNET" />
```

- 在Unity中



PVRTune



连接过程

- 启动设备上的PVRPerfServer

```
su
```

```
cd /data/data/com.powervr.PVRHub/bin  
./PVRHubDaemon &
```

- 安卓调试桥（ADB）命令

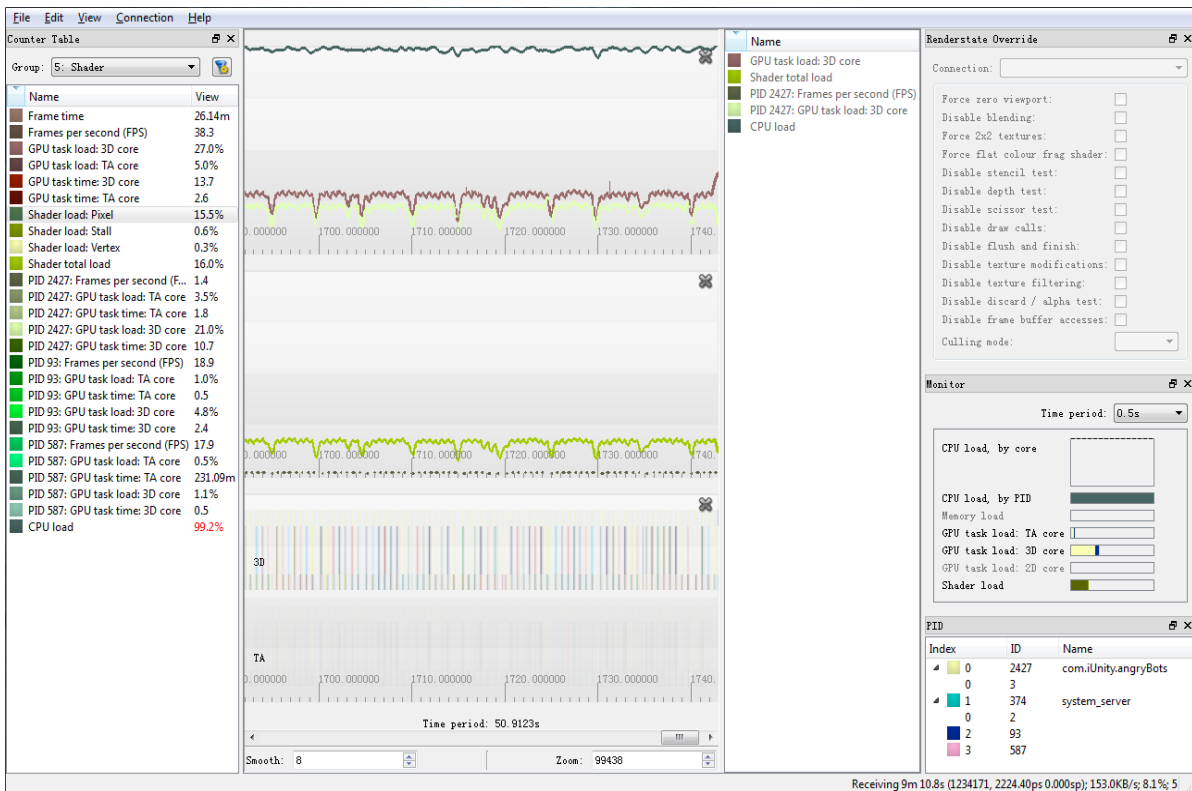
```
adb forward tcp:6520 tcp:6520
```

- 启动PVRTune并连接到:

```
pvrtune://localhost
```

- 启动目标应用程序

PVRTune



- Core Overview
 - Frames per second (FPS)
 - GPU task load
 - CPU load
 - Memory total / use
 - Shader Clock Cycle per pixel
 - Shader Clock Cycle per vertex
 - Shader load: Pixel
 - Shader load: Vertex
- Primitive Clipping
 - Primitives per frame: on-screen
 - Vertice per frame: on-screen
- Texturing
 - Texture unit(s) load

PerfHUD ES



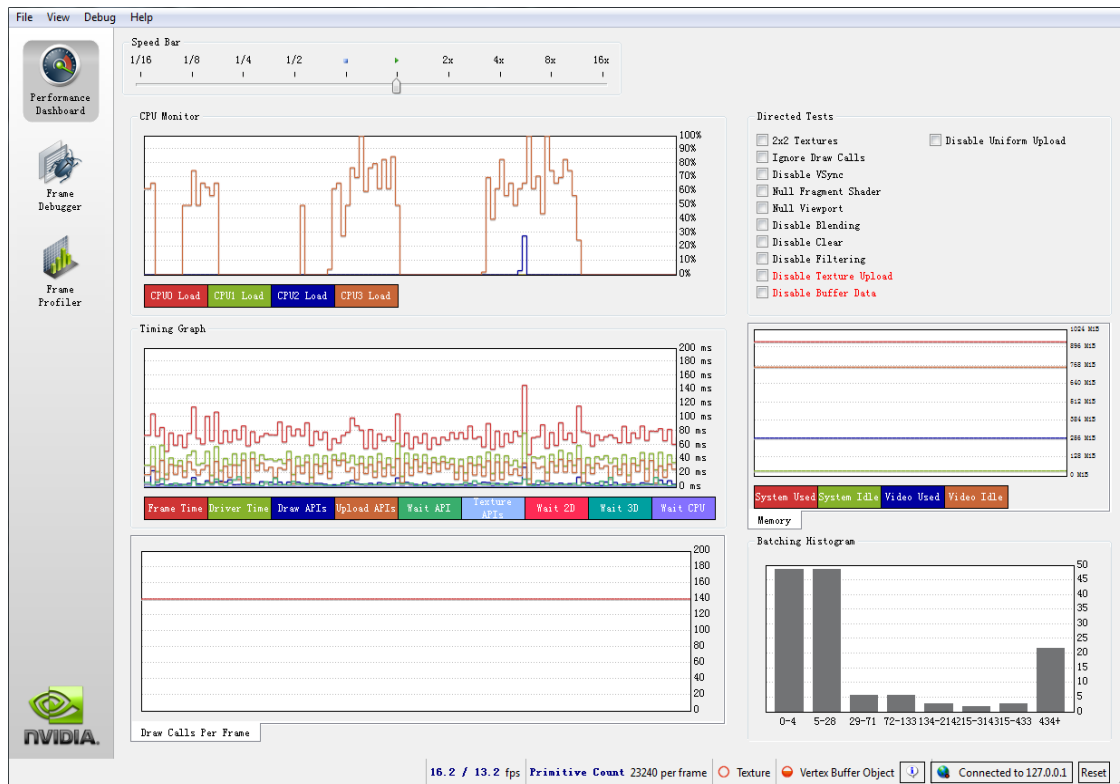
连接过程

- 关闭目标应用程序
- 安卓调试桥（ADB）命令

adb shell setprop debug.perfhudes 1

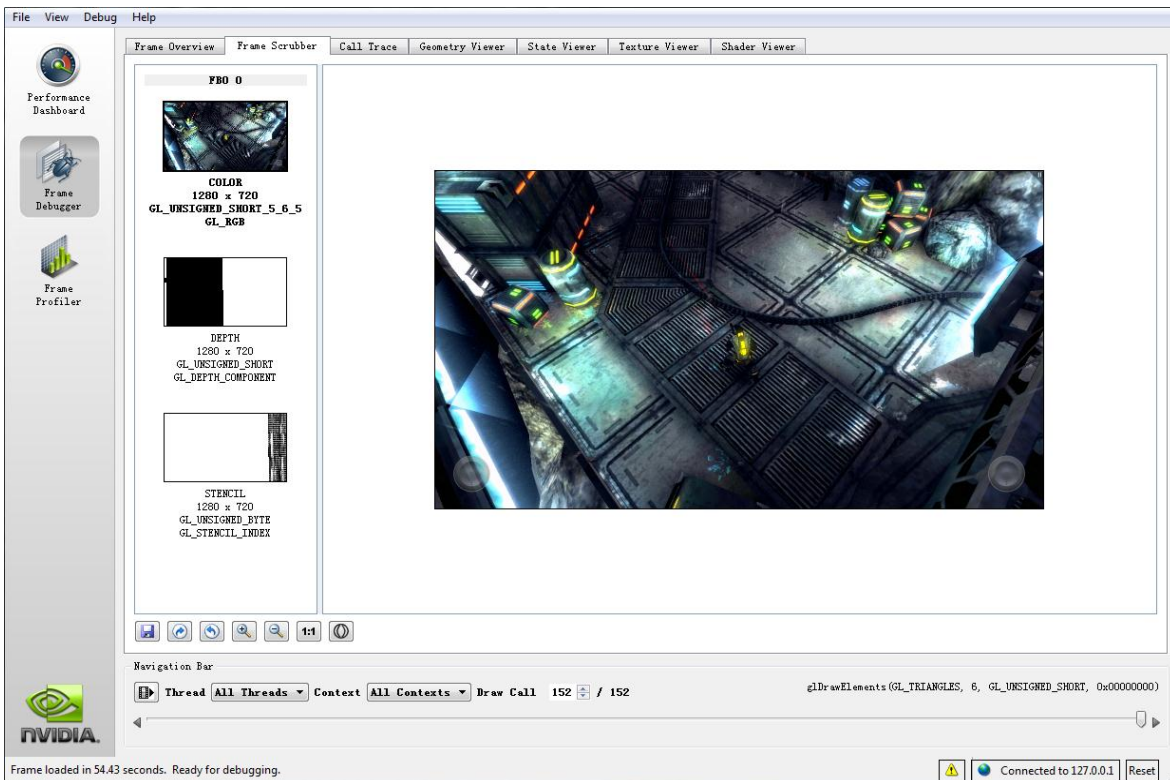
- 启动目标应用程序
- 启动PerfHUB ES

性能仪表盘 (Performance DashBoard)



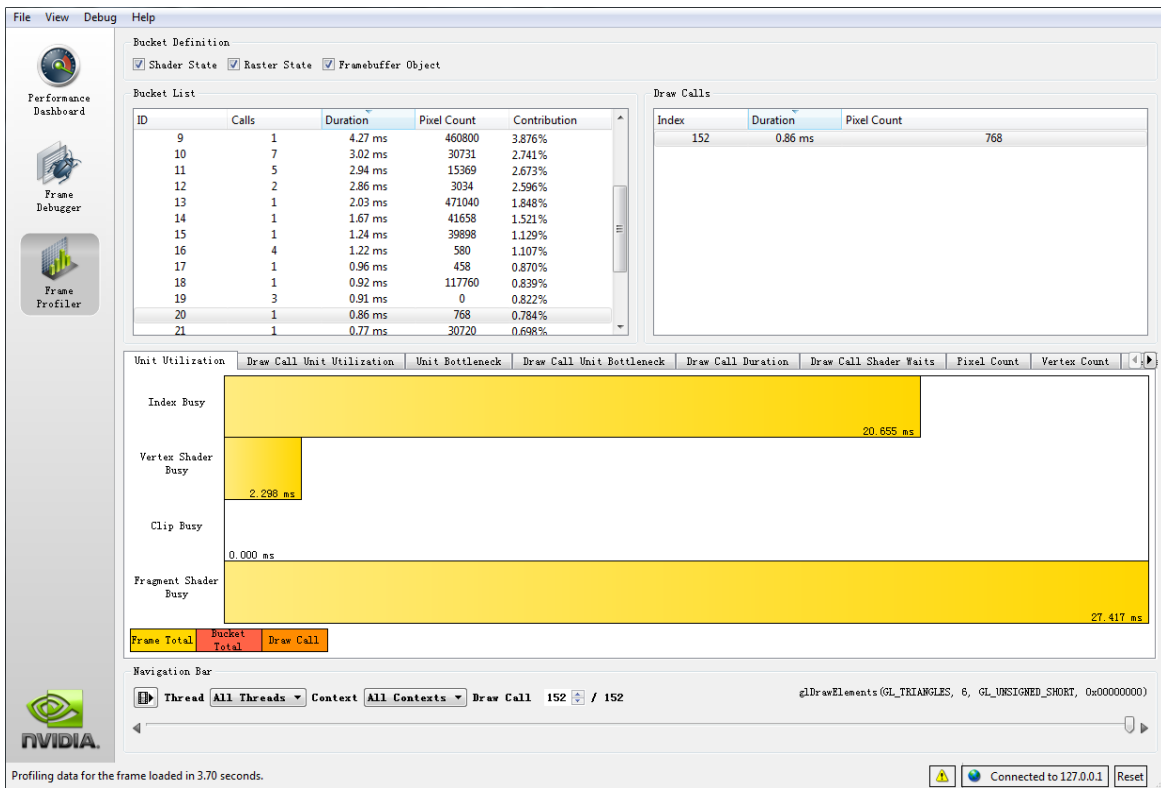
- Speed Bar
- Directed Tests
 - Ignore Draw Calls
 - Null Fragment Shader
- CPU Monitor
 - Draw Calls Per Frame
- Batching Histogram
 - Draw Calls Per Frame
- Status Bar
 - fps
 - Primitive Count

帧调试器



- Frame Scrubber
 - FBO (FrameBuffer Object)
- Frame Overview
 - Total Vertices / Primitives
 - Texture Number
 - Program Number
- Geometry Viewer
 - Attribute
- State Viewer
- Texture Viewer
- Shader Viewer

帧分析器



- Bucket Definition
- Unit Utilization
- Draw Call Unit Utilization
- Draw Call Unity Bottleneck
- Draw Call Duration
- Draw Call Shader Waits
- Vertex Count
- Triangle Count
- Vertex Attribute Count

Adreno分析器

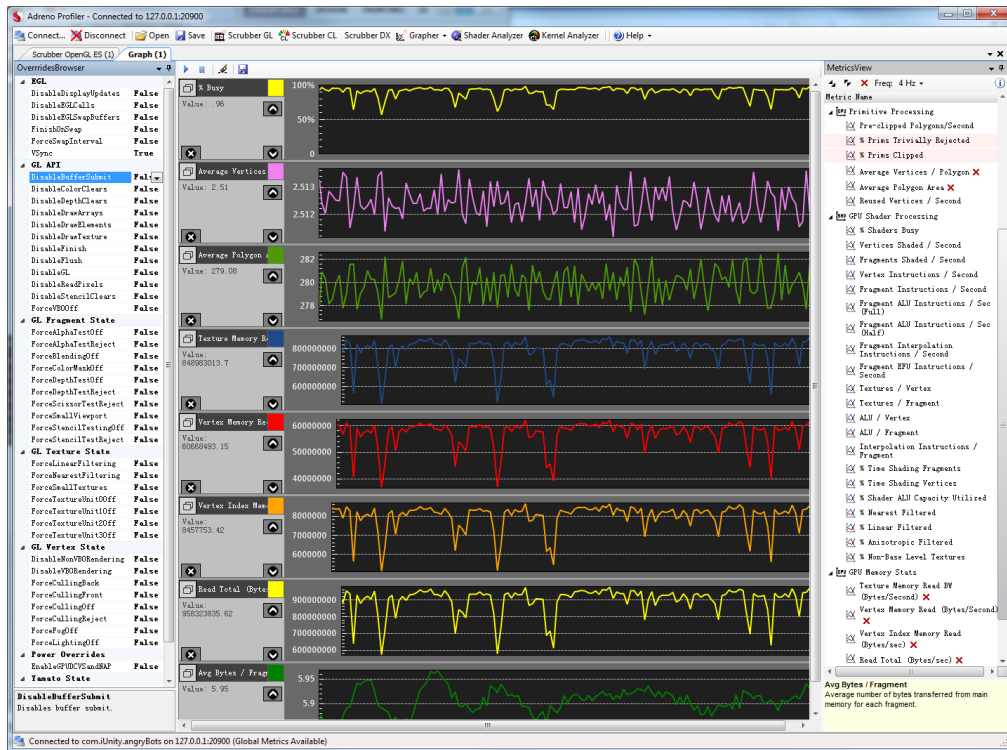


连接过程

- 启动目标应用程序
- 启动 Adreno分析器
- Click 'Connect...' to open this
 - *Select Target Application*
 - *Click 'Connect' button*



图示记录器



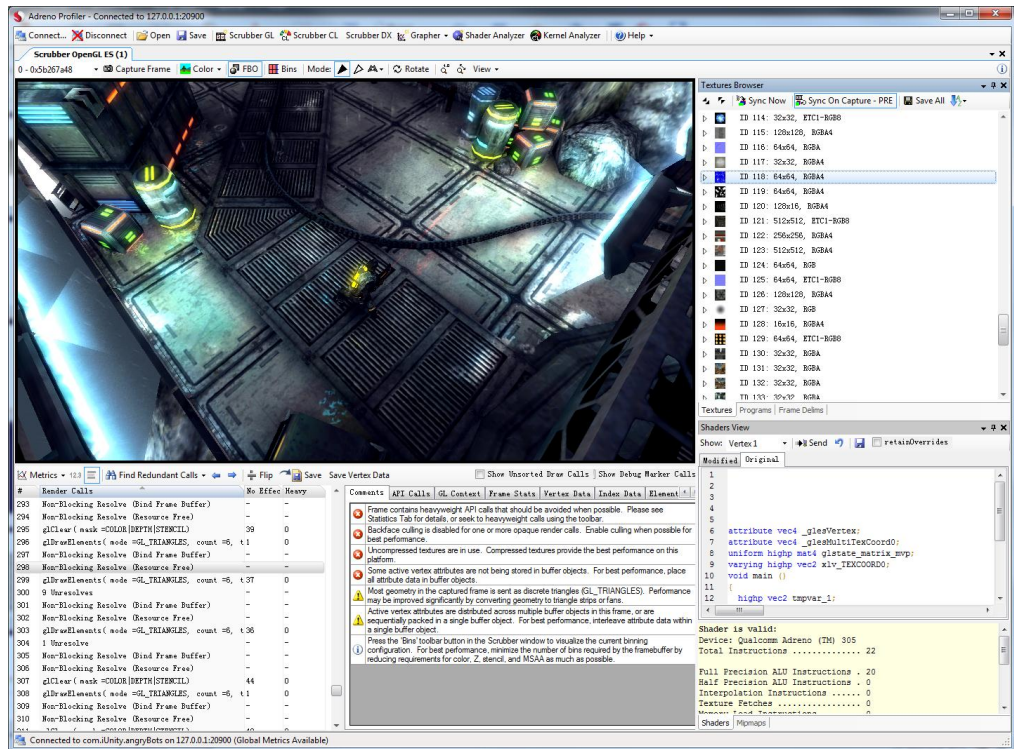
指标查看

- EGL (App Metrics Grapher)
 - FPS
- GPU General
 - % Busy
- GPU Shader Processing
 - % Shaders Busy
 - % Time Shading Fragment/Vertices








覆盖

- GL API
 - Disable Draw Elements
- GL Texture State
 - Force Linear/Nearest Filtering
 - Force Small Textures

Srubby



注解

Comments	API Calls	GL Context	Frame Stats	Vertex Data	Index Data	Element Data
	Frame contains heavyweight API calls that should be avoided when possible. Please see Statistics Tab for details, or seek to heavyweight calls using the toolbar.					
	Backface culling is disabled for one or more opaque render calls. Enable culling when possible for best performance.					
	Uncompressed textures are in use. Compressed textures provide the best performance on this platform.					
	Some active vertex attributes are not being stored in buffer objects. For best performance, place all attribute data in buffer objects.					
	Most geometry in the captured frame is sent as discrete triangles (GL_TRIANGLES). Performance may be improved significantly by converting geometry to triangle strips or fans.					
	Active vertex attributes are distributed across multiple buffer objects in this frame, or are sequentially packed in a single buffer object. For best performance, interleave attribute data within a single buffer object.					
	Press the 'Bins' toolbar button in the Scrubber window to visualize the current binning configuration. For best performance, minimize the number of bins required by the framebuffer by reducing requirements for color, Z, stencil, and MSAA as much as possible.					

GL 文本/帧统计

- 混合
 - 已启用
- 选取
 - 已启用
- 渲染调用
 - # glDraw调用
- 几何
 - 总顶点
 - 总图元
- 杂项
 - 总纹理使用
- 纹理格式

几何数据

顶点

UV1

UV2

顶点索引

顶点三角

顶点数据

	_glesVertex - idx:0	_glesMultiTexCoord0 - idx:3	_glesMultiTexCoord1 - idx:4
0	-0.221245, 3.753481, -132.8791	1.050042E+09, 1.017569E+09	1.050042E+09, 1.017569E+09
1	0.6291934, 2.043518, -132.702	3.205685E+09, 1.065009E+09	3.205685E+09, 1.065009E+09
2	-0.2212454, 3.306272, -133.0829	3.186065E+09, 1.017569E+09	3.186065E+09, 1.017569E+09
3	0.6262308, 3.512619, -131.3856	1.061298E+09, 1.064952E+09	1.061298E+09, 1.064952E+09
4	-0.2212454, 3.306272, -133.0829	3.198419E+09, 1.017569E+09	3.198419E+09, 1.017569E+09
5	-3.329978, 2.043514, -132.702	1.073218E+09, 1.065009E+09	1.073218E+09, 1.065009E+09
6	-2.479542, 3.306269, -133.0829	1.06804E+09, 1.017569E+09	1.06804E+09, 1.017569E+09
7	0.6291934, 2.043518, -132.702	3.211796E+09, 1.065009E+09	3.211796E+09, 1.065009E+09
8	-2.479542, 3.306269, -133.0829	3.186065E+09, 1.017569E+09	3.186065E+09, 1.017569E+09
9	-3.327009, 3.512615, -131.3856	1.061298E+09, 1.064952E+09	1.061298E+09, 1.064952E+09
10	-2.479542, 3.753479, -132.8791	1.050042E+09, 1.017569E+09	1.050042E+09, 1.017569E+09
11	-3.329978, 2.043514, -132.702	3.205685E+09, 1.065009E+09	3.205685E+09, 1.065009E+09
12	-3.327009, 3.512615, -131.3856	1.073201E+09, 1.064952E+09	1.073201E+09, 1.064952E+09
13	-0.221245, 3.753481, -132.8791	3.198419E+09, 1.017569E+09	3.198419E+09, 1.017569E+09
14	-2.479542, 3.753479, -132.8791	1.06804E+09, 1.017569E+09	1.06804E+09, 1.017569E+09
15	0.6262308, 3.512619, -131.3856	3.211764E+09, 1.064952E+09	3.211764E+09, 1.064952E+09
16	-4.111131, 3.127025, 19.11377	1.050042E+09, 1.017569E+09	1.050042E+09, 1.017569E+09
17	-4.961568, 1.466375, 18.70544	3.205685E+09, 1.065009E+09	3.205685E+09, 1.065009E+09
18	-4.111131, 2.59259, 19.39194	3.186065E+09, 1.017569E+09	3.186065E+09, 1.017569E+09
19	-4.958604, 3.219525, 17.80123	1.061298E+09, 1.064952E+09	1.061298E+09, 1.064952E+09
20	-4.111131, 2.59259, 19.39194	3.198419E+09, 1.017569E+09	3.198419E+09, 1.017569E+09
21	-1.002397, 1.466374, 18.70544	1.073218E+09, 1.065009E+09	1.073218E+09, 1.065009E+09
22	-1.852834, 2.592589, 19.39194	1.06804E+09, 1.017569E+09	1.06804E+09, 1.017569E+09
23	-4.961568, 1.466375, 18.70544	3.211796E+09, 1.065009E+09	3.211796E+09, 1.065009E+09
24	-1.852834, 2.592589, 19.39194	3.186065E+09, 1.017569E+09	3.186065E+09, 1.017569E+09
25	-1.005365, 3.219524, 17.80123	1.061298E+09, 1.064952E+09	1.061298E+09, 1.064952E+09
26	-1.852834, 3.127025, 19.11377	1.050042E+09, 1.017569E+09	1.050042E+09, 1.017569E+09
27	-1.002397, 1.466374, 18.70544	3.205685E+09, 1.065009E+09	3.205685E+09, 1.065009E+09
28	-1.005365, 3.219524, 17.80123	1.073201E+09, 1.064952E+09	1.073201E+09, 1.064952E+09

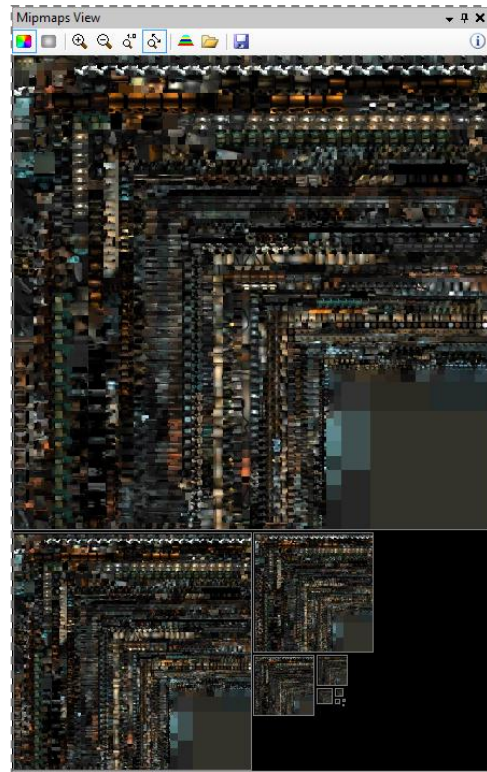
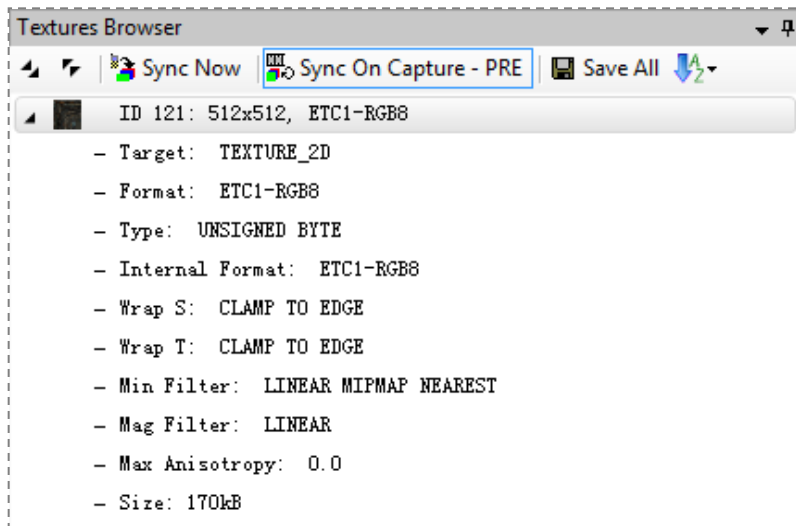
索引数据

	Indices
0	4092
1	4093
2	4094
3	4095
4	4096
5	4097
6	4095
7	4098
8	4096
9	4099
10	4100
11	4101
12	4102
13	4103
14	4104
15	4102
16	4105
17	4103
18	4106
19	4107
20	4108
21	4106
22	4109
23	4107
24	4110
25	4111
26	4112
27	4110
28	4113

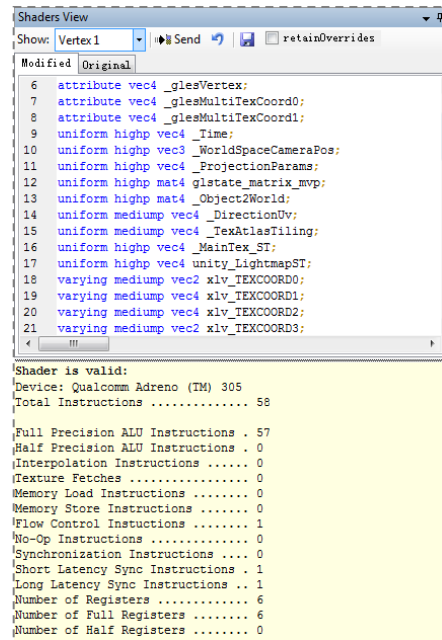
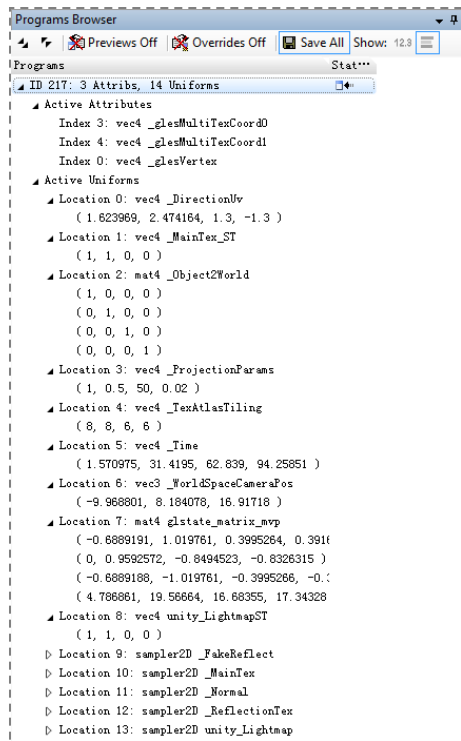
元数据

	GL_TRIANGLES
0	4092, 4093, 4094
1	4095, 4096, 4097
2	4095, 4098, 4096
3	4099, 4100, 4101
4	4102, 4103, 4104
5	4102, 4105, 4103
6	4106, 4107, 4108
7	4106, 4109, 4107
8	4110, 4111, 4112
9	4110, 4113, 4111
10	4114, 4115, 4116
11	4116, 4117, 4114
12	4117, 4118, 4114
13	4119, 4120, 4121
14	4119, 4122, 4120
15	4123, 4124, 4125
16	4123, 4126, 4124
17	4127, 4128, 4129
18	4127, 4130, 4128
19	4131, 4132, 4133
20	4131, 4134, 4132
21	4135, 4136, 4137
22	4135, 4138, 4136
23	4139, 4140, 4141
24	4139, 4142, 4140
25	4143, 4144, 4145
26	4143, 4146, 4144
27	4147, 4148, 4149
28	4150, 4151, 4152

纹理浏览器/Mipmap贴图视图



程序浏览器/着色器视图

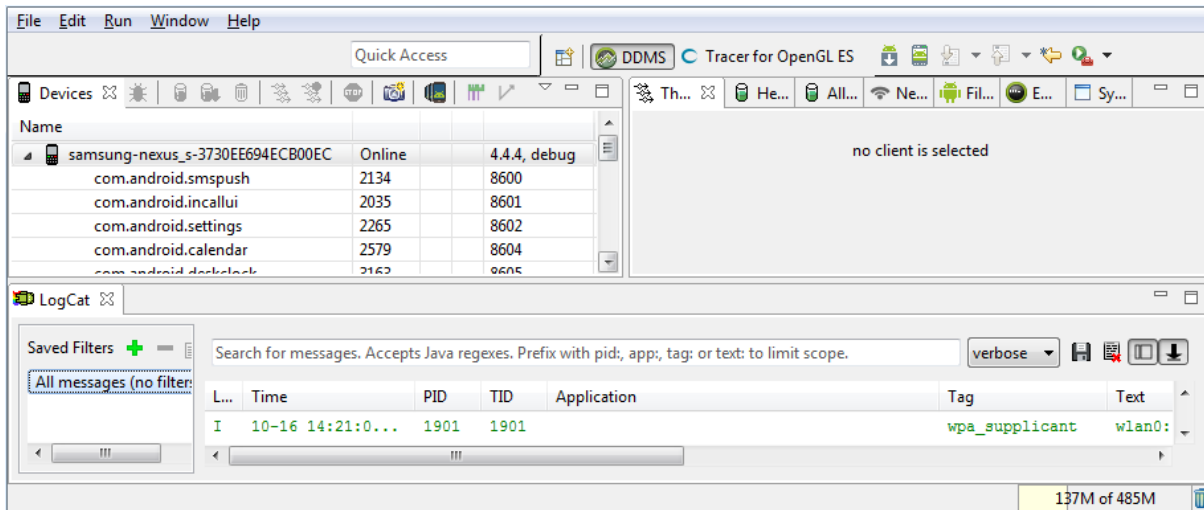


Android Device Monitor



连接过程

- 启动 Android Device Monitor

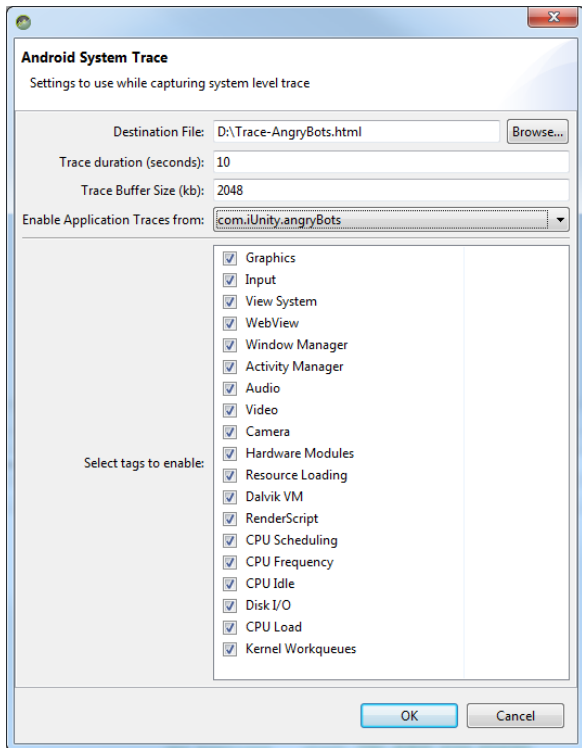


Requirement:

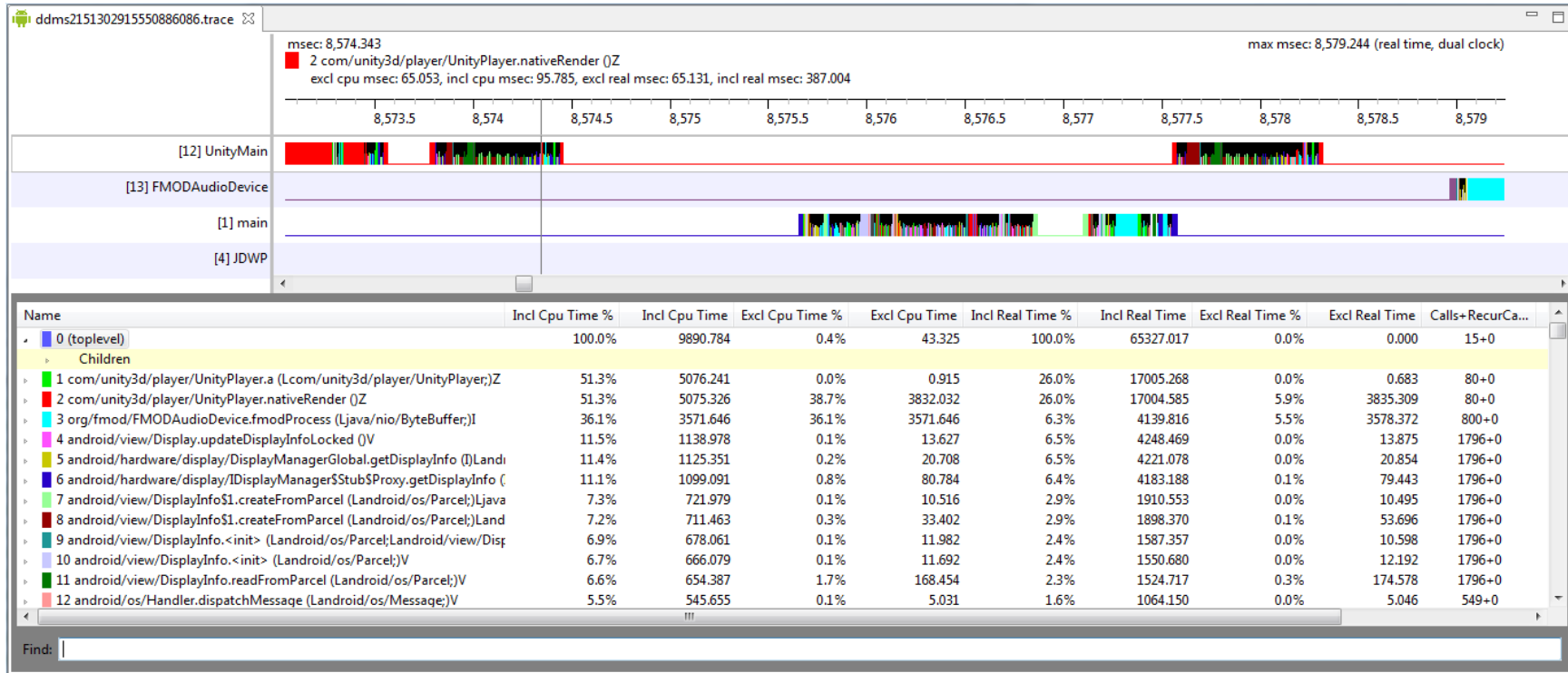
Android SDK Tool 20

Android 4.1 (API Level 16)

SysTrace



Trace

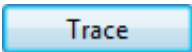


Tracer for OpenGL ES

连接过程

- Kill Target Application
- Get 'Application Package' & 'Main Activity' name

```
aapt dump xmltree OGLS2Water.apk AndroidManifest.xml
```

- Set names in options..
- Press  to start.



OpenGL ES Trace Options

Provide the application and activity to be traced.

Device:

Application Package:

Activity to launch:

☐ Activity name is fully qualified, do not prefix with package name

Data Collection Options:

- ☒ Read back framebuffer 0 on eglSwapBuffers()
- ☐ Read back currently bound framebuffer on glDraw()
- ☒ Collect texture data submitted using glTexImage()

Destination File:

Tracer for OpenGL ES

The screenshot displays the OGLES2Water.gltrace application interface, which is divided into several panels for analyzing OpenGL ES calls and state.

OGLES2Water.gltrace Panel:

- Select Frame:** A slider set to frame 16.
- Filter:** A text input field with the placeholder "Filter list of OpenGL calls. Accepts Java regexes."
- Table:** A table listing OpenGL functions, their wall time, and thread time. The function `glDrawElements` is highlighted in blue.

Function	Wall Time (ns)	Thread Time (ns)
<code>glUniform3fv(location = 5, count = 1, v = [1.0, 1.0, 1.0])</code>	4,542	2,291
<code>glUniform3fv(location = 6, count = 1, v = [0.0, 0.0, 0.0])</code>	4,125	2,125
<code>glUniform3fv(location = -1, count = 1, v = [0.0, 0.0, 0.0])</code>	3,625	1,541
<code>glBindBuffer(target = GL_ARRAY_BUFFER, buffer = 420006)</code>	5,708	3,541
<code>glBindBuffer(target = GL_ELEMENT_ARRAY_BUFFER, buffer = 1)</code>	4,500	2,375
<code>glEnableVertexAttribArray(index = 0)</code>	3,917	1,750
<code>glEnableVertexAttribArray(index = 1)</code>	3,583	1,541
<code>glEnableVertexAttribArray(index = 2)</code>	3,583	1,583
<code>glVertexAttribPointer(indx = 0, size = 3, type = GL_FLOAT, norm = false, offset = 0)</code>	5,000	2,916
<code>glVertexAttribPointer(indx = 1, size = 3, type = GL_FLOAT, norm = false, offset = 4)</code>	4,584	2,375
<code>glVertexAttribPointer(indx = 2, size = 2, type = GL_FLOAT, norm = false, offset = 8)</code>	4,750	2,708
<code>glDrawElements(mode = GL_MAP_INVALIDATE_RANGE_BIT, count = 28, type = GL_UNSIGNED_SHORT, offset = 0)</code>	44,667	41,958
<code>glDisableVertexAttribArray(index = 0)</code>	4,666	2,166
<code>glDisableVertexAttribArray(index = 1)</code>	3,792	1,666
<code>glDisableVertexAttribArray(index = 2)</code>	3,583	1,500
<code>glBindBuffer(target = GL_ARRAY_BUFFER, buffer = 0)</code>	4,375	2,333
<code>glBindBuffer(target = GL_ELEMENT_ARRAY_BUFFER, buffer = 0)</code>	4,083	2,000
<code>glActiveTexture(texture = GL_TEXTURE0)</code>	3,750	1,666
<code>glBindTexture(target = GL_TEXTURE_2D, texture = 840012)</code>	5,459	3,333
<code>glUseProgram(program = 2240032)</code>	3,875	1,750
<code>glUniformMatrix4fv(location = 2, count = 1, transpose = false, v = [1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0, 0.0, 0.0, 1.0])</code>	4,917	2,708
<code>glUniform3fv(location = -1, count = 1, v = [-482.7261, 68.22445, 0.0])</code>	3,833	1,666

GL State Panel:

- Name:** A tree view showing the current OpenGL state. The selected state is `Context 0 (ES2)`.
- Value:** A list of values for the selected state.

Details Panel:

- Frame Summary:** A summary of the frame's performance.
- Console:** A console window for logging.
- Water:** A 3D rendering of a ship on water.

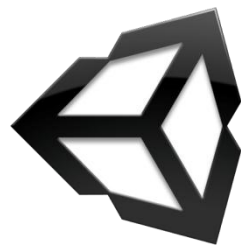
Performance Metrics:

- Cumulative call duration of all OpenGL Calls in this frame:**
 - Wall Clock Time: 17.16 ms
 - Thread Time: 8.44 ms
- Per OpenGL Function Statistics:**

Function	C...	Wall Time ...	Thread Ti...
<code>glDrawElements</code>	28	5,004,917	3,321,449
<code>glUniform3fv</code>	1...	2,786,919	769,023
<code>glBindBuffer</code>	1...	1,643,416	617,255
<code>glUniform1f</code>	87	1,314,086	352,182
<code>glClear</code>	3	1,103,290	324,455

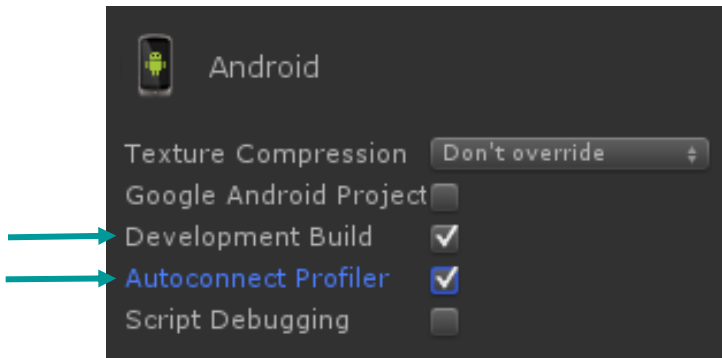
Bottom Bar: 212M of 500M

Unity Profiler




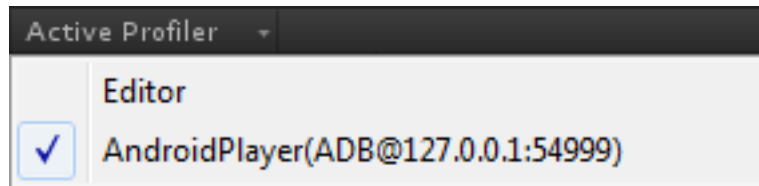
连接过程

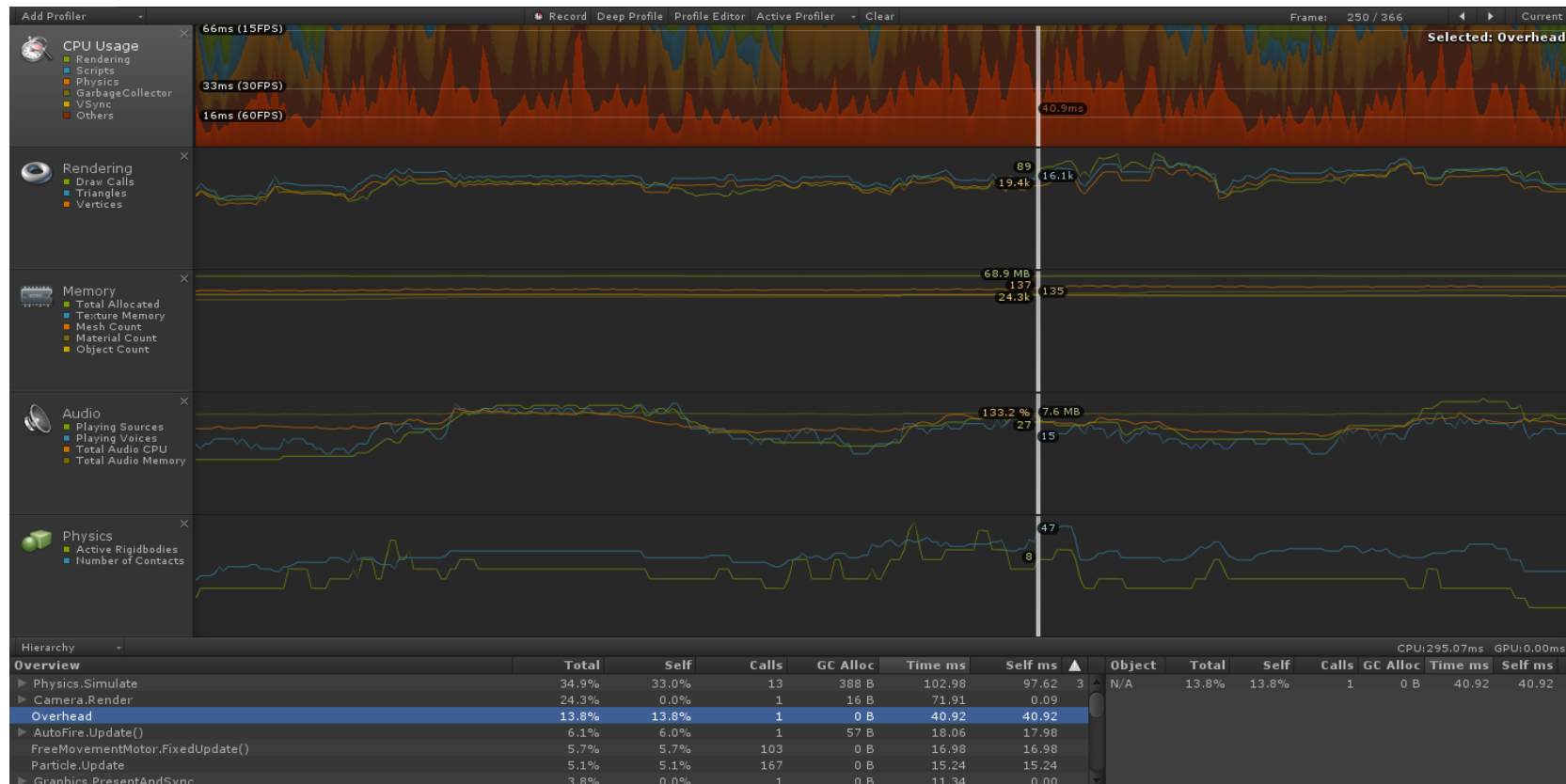
- Build APK using setting..
 - Development Build
 - Autoconnect Profiler
- ADB command



`adb forward tcp:54999 localabstract:Unity-com.iUnity.angryBots`

- Change Active Profiler to "AndroidPlayer(ADB@127.0.0.1:54999)"
- Press  Record to start.





谢谢