

Contents

1	前言	3
2	UnityProfiler	8
2.1	简介	8
2.2	命令手册	8
2.2.1	alloc	8
2.2.2	frame	11
2.2.3	next	11
2.2.4	prev	11
2.2.5	func	11
2.2.6	find	11
2.2.7	list	11
2.2.8	info	11
2.2.9	meta	11
2.2.10	lock	11
2.2.11	stat	11
2.2.12	seek	11
2.2.13	fps	11
2.2.14	help	11
2.2.15	quit	11
2.3	使用案例	11
2.3.1	追踪渲染丢帧	11
2.3.2	追踪动态内存分配	11
2.4	小结	11
3	MemoryCrawler	11
3.1	简介	11
3.2	命令手册	11
3.2.1	read	11
3.2.2	load	11

3.2.3	track	11
3.2.4	str	11
3.2.5	ref	11
3.2.6	uref	11
3.2.7	REF	11
3.2.8	UREF	11
3.2.9	kref	11
3.2.10	ukref	11
3.2.11	KREF	11
3.2.12	UKREF	11
3.2.13	link	11
3.2.14	ulink	11
3.2.15	show	11
3.2.16	ushow	11
3.2.17	find	11
3.2.18	ufind	11
3.2.19	type	11
3.2.20	utype	11
3.2.21	stat	11
3.2.22	ustat	11
3.2.23	list	11
3.2.24	ulist	11
3.2.25	bar	11
3.2.26	ubar	11
3.2.27	heap	11
3.2.28	save	11
3.2.29	uuid	11
3.2.30	help	11
3.2.31	quit	11
3.3	使用案例	11
3.3.1	检视内存对象	11

3.3.2

追踪内存增长

11

3.3.3

追踪内存泄漏

11

3.3.4

优化 Mono 内存

11

3.4

小结

11

1前言

Unity3D 是个普及度很高拥有大量开发者的游戏开发引擎，其提供的 Unity 编辑器可以快速的开发移动设备游戏，并且通过编辑器扩展可以很容易开发出项目需要的辅助工具，但是 Unity 提供性能调试工具非常简陋，功能简单并且难以使用，如果项目出现性能问题，定位起来相当花时间，并且准确率很低，一定程度上靠运气。

Profiler

目前 Unity 随包提供的只有 Profiler 工具，里面聚合 CPU、GPU、内存、音频、视频、物理、网络等多个维度的性能数据，但是我们大部分情况下只是用它来定位卡顿问题，也就是主要 CPU 时间消耗 (图1)。

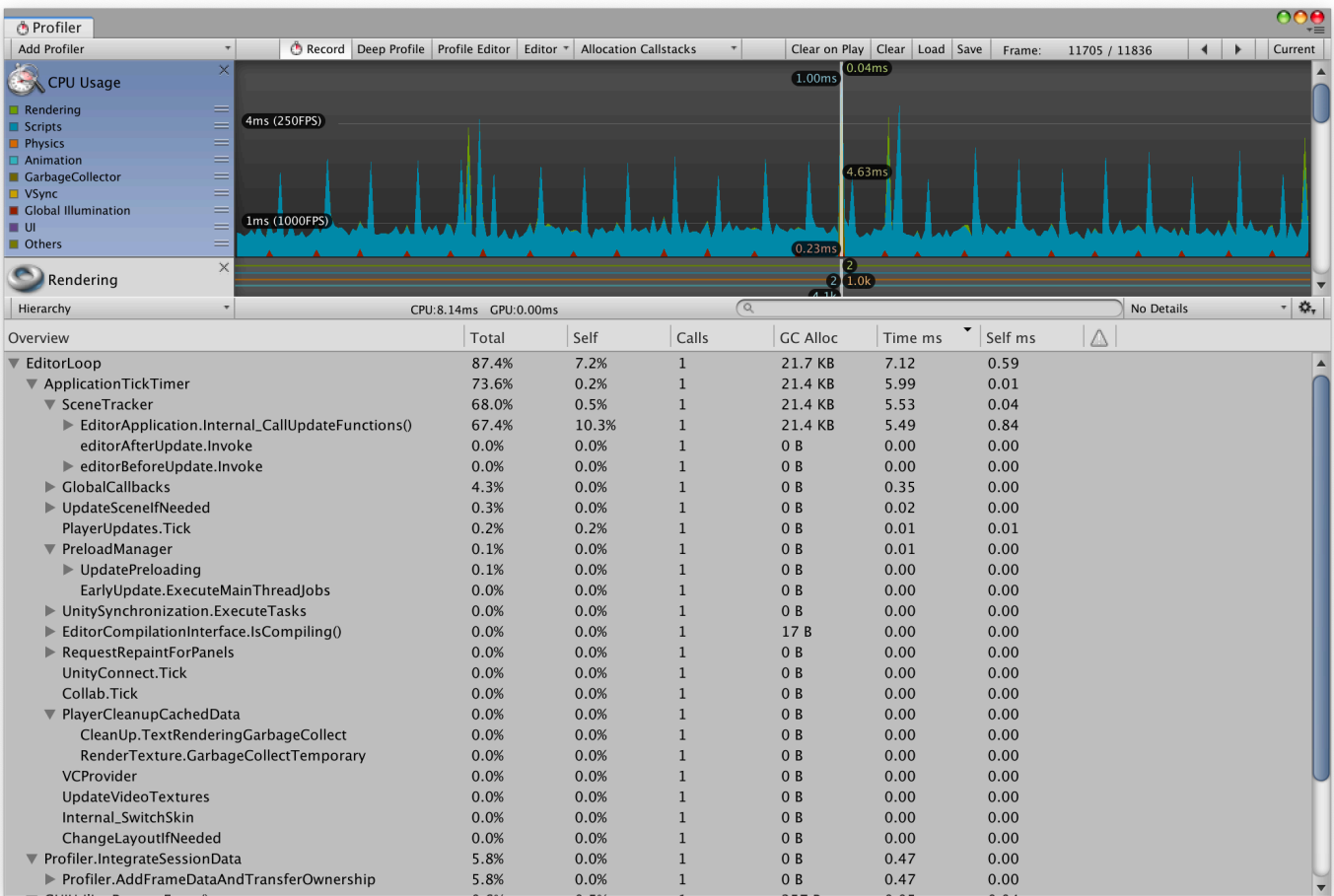


Figure 1: Unity 性能调试工具 Profiler

在 CPU 的维度里面，可以看到当前渲染帧的函数调用层级关系，以及每个函数的时间开销以及调用次数等信息，但是这个工具同一时间只能处理 300 帧左右的数据，如果游戏帧率 30，那么只能看到过去 10 秒的信息，并且需要我们一直盯着图表看才有机会发现意外的丢帧情况，这种设计非常的不友好，违反正常人的操作习惯，因为通常情况下如果我要调试游戏内战斗过程的性能开销，首先我要像普通玩家那样安安静静的玩一把，而不是分散出大部分精力去看一个只有 10 秒历史的滚动图表。这种交互带来两个明显的问题，

- 由于分心去看 Profiler，导致不能全心投入游戏，从而不能收集正常战斗过程的性能数据
- 为了收集数据需要像正常玩家那样打游戏，不能全神关注 Profiler 图表，从而不能发现/查看所有的性能问题

上面两个情形相互排斥，鱼与熊掌不可兼得。从这个角度来看，Profiler 不是一个好的性能调试工具，苛刻的操作条件导致我们很难发现性能问题，想要通过 Profiler 定位所有的性能问题简直是痴人说梦。

MemoryProfiler

Unity 还提供另外一个内存分析工具 MemoryProfiler(图2)

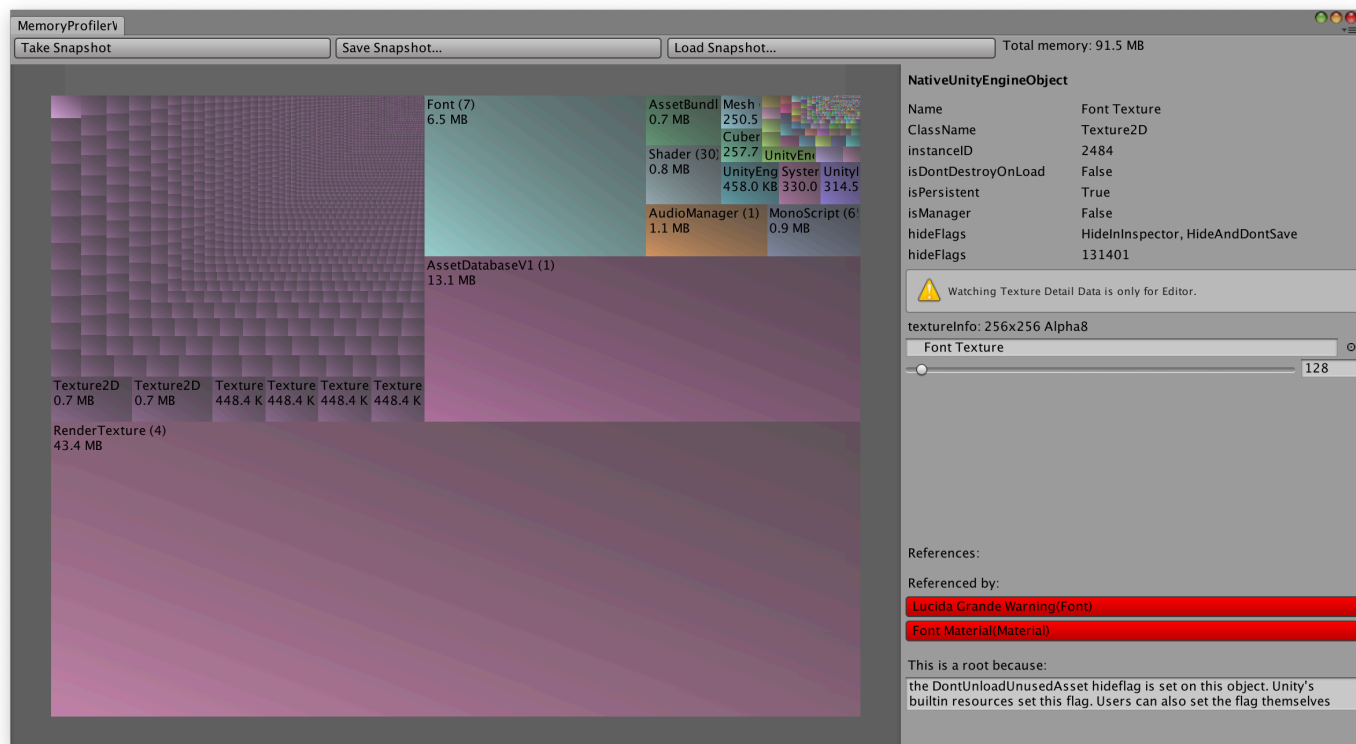


Figure 2: Unity 内存调试工具 MemoryProfiler

在这个界面的左边彩色区域里，MemoryProfiler 按类型占用总内存大小绘制对应面积比例的矩阵图，第一次看到还是蛮酷炫的，Unity 是想通过这个矩阵图向开发者提供对象内存检索入口，但是实际使用过程中问题多多。

- 内存分析过程缓慢

- 在众多无差别的小方格里面找到实际关心的资源很难，虽然可以放大缩小，但感觉并没有提升检索的便利性
- 每个对象只提供父级引用关系，无法看到完整的对象引用链，容易在跳转过程中迷失
- 引擎对象的引用和 IL2CPP 对象的引用混为一谈，让使用者对引用关系的理解模糊不清
- 没有按引擎对象内存和 IL2CPP 对象内存分类区别统计，加深使用者对内存使用的误解

MemoryProfiler源码托管在 Bitbucket，但是从最后提交记录来看，这个内存工具已经超过 2 年半没有任何更新了，但是这期间 Unity 可是发布了好多个版本，想想就有点后怕。

Unity Technologies / Unity / MemoryProfiler

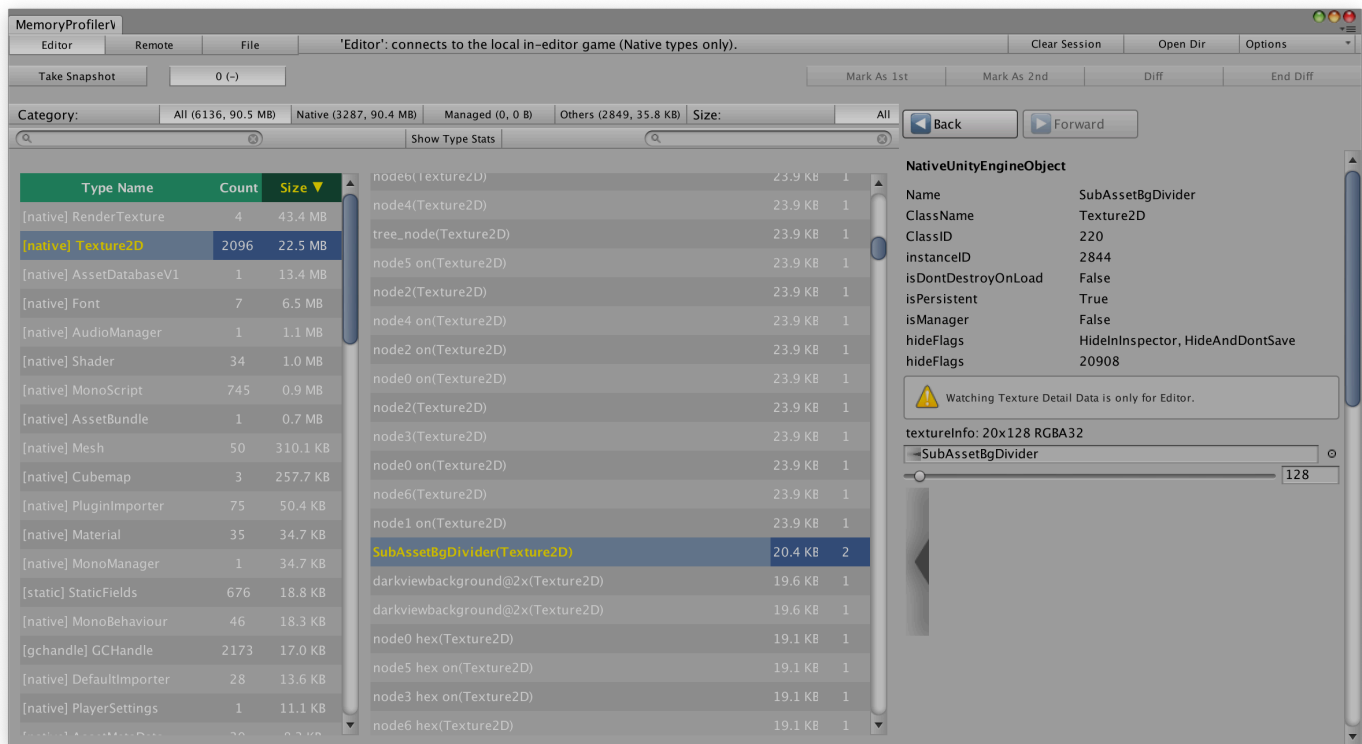
Commits

All branches

? Find commits

Author	Commit	Message	Date	Builds
Lukasz Paczko...	e8317df <small>M</small>	Merged in lukaszunity/include-mono-support-in-readme (pull request ...	2017-10-13	
Lukasz Paczko...	abd35a7	Close branch lukaszunity/include-mono-sup... <small>lukaszunity/includ...</small>	2017-10-13	
Lukasz Paczko...	8322cc4	Use Mono .NET 3.5 wording instead of 2.0 in ... <small>lukaszunity/includ...</small>	2017-09-27	
Lukasz Paczko...	11f359d	Include Mono support in README <small>lukaszunity/includ...</small>	2017-09-27	
Tautvydas Žilys	4b2f619 <small>M</small>	Merged in flassari/memoryprofiler/zero-positive-fix (pull request #30)...	2017-07-29	
Ari Arnbjörnss...	7e8a974	Treat zero as a positive number <small>zero-positive-fix</small>	2017-07-27	
Tautvydas Žilys	a0d5a31 <small>M</small>	Merged in aghogg/memoryprofiler/classid_remove (pull request #31) ...	2017-07-29	
Ashley Hogg	c3d91ad	Removed ClassID field from the object inspector, s... <small>classid_remove</small>	2017-07-20	
Tautvydas Žilys	7126ded <small>M</small>	Merged in aghogg/memoryprofiler/64bit_size_fix (pull request #32) C...	2017-07-29	
Ashley Hogg	6478c5d	Changed size members to 64-bit longs, to fix over... <small>64bit_size_fix</small>	2017-07-20	

有热心开发者也忍受不了 Unity 这缓慢的更新节奏，干脆自己动手基于源码在github上更新优化，并更改了检索的交互方式。



不过这也只是在 MemoryProfiler 的基础上增加检索的便利性，跟理想的检索工具还有很大差距，虽然在内存的类别上做了相对 MemoryProfiler 更加清晰的区分，但是没有系统化的重构设计，内存分析过程依然异常缓慢，甚至会在分析过程中异常崩溃。

Unity Bug Reporter

What is the problem related to*

Crash bug

How often does it happen*

Please Specify

Your email address*

Title*

Describe the problem

[How to report bugs](#)

[Public issue tracker](#)

[Unity answers](#)

[Unity forums](#)

[Unity community](#)

[Privacy policy](#)

unity3d.com

issuetracker.unity3d.com

answers.unity3d.com

forum.unity3d.com

unity3d.com

unity3d.com

Details*

1. What happened

2. How we can reproduce it using the example you attached

Attached files

Path	Type
/Users/larryhou/Documents/Unity/MemoryProfiler	folder
/Users/larryhou/Documents/Unity/MemoryProfiler/Packages/manifest.json	.json

Add File

Add Folder

Remove

Report strength:

In order to send the report you need to specify how often does the problem happen.

Preview

Cancel

Send

```
namespace MemoryProfilerWindow
{
    static class ManagedHeapExtensions
    {
```

```

public static BytesAndOffset Find(this MemorySection[] heap, UInt64 address,
    VirtualMachineInformation virtualMachineInformation)
{
    foreach (var segment in heap)
    {
        if (address ≥ segment.startAddress
            && address < (segment.startAddress + (ulong) segment.bytes.Length))
        {
            return new BytesAndOffset()
            {
                bytes = segment.bytes,
                offset = (int)(address - segment.startAddress),
                pointerSize = virtualMachineInformation.pointerSize
            };
        }
    }

    return new BytesAndOffset();
}
}
}

```

2 UnityProfiler

2.1 简介

2.2 命令手册

2.2.1 alloc

```

const char *basename(const char *filepath)
{
    auto offset = filepath + strlen(filepath);
    const char *upper = nullptr;

    while (offset ≠ filepath)
    {
        if (upper = nullptr && *offset == '.') {upper = offset;}
        if (*offset == '/')
        {

```



```

        ++offset;
        break;
    }
    --offset;
}

auto name = &*offset;
auto size = upper - offset;

char *filename = new char[size + 1];
memset(filename, 0, size + 1);
memcpy(filename, name, size);
return filename;
}

```


2.2.2 frame

2.2.3 next

2.2.4 prev

2.2.5 func

2.2.6 find

2.2.7 list

2.2.8 info

2.2.9 meta

2.2.10 lock

2.2.11 stat

2.2.12 seek

2.2.13 fps

2.2.14 help

2.2.15 quit

2.3 使用案例

2.3.1 追踪渲染丢帧

2.3.2 追踪动态内存分配

2.4 小结

3 MemoryCrawler

3.1 简介

3.2 命令手册

3.2.1 read

3.2.2 load

3.2.3 track