GOT 使用说明

安装环境：Windows & Mac OS
支持发布平台：Android & iOS & Windows

Contents

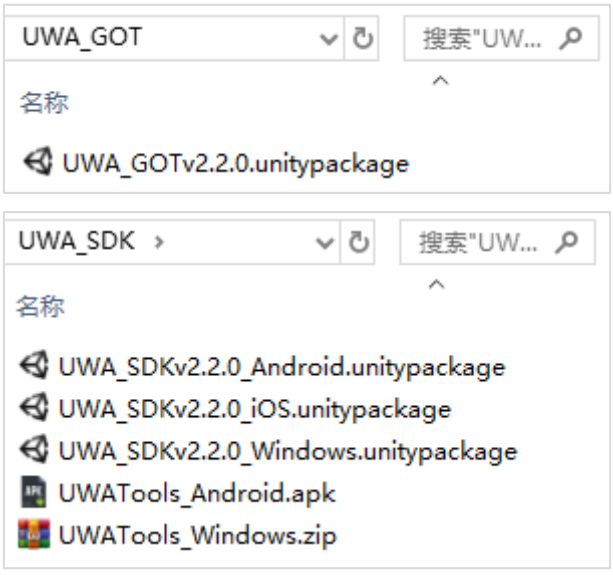
1. GOT 工具

1.1. 工具下载

1. 打开 UWA 网站，登录 UWA 账号；
2. 打开下载页面：https://www.uwa4d.com/index.html#download，可以看到本地测试产品 - GOT 的购买及文档下载链接，购买之后即可下载工具，文档可直接下载查看。

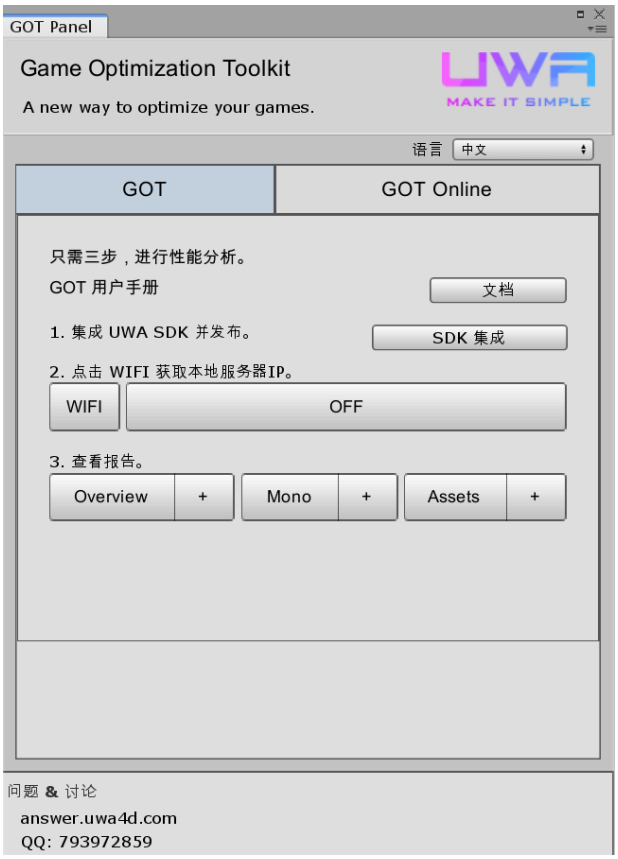
3. 下载包为一个 ZIP 文件，其中包含了两个文件夹：UWA_GOT、UWA_SDK。需要集成对应发布平台的 UWA_SDK_XXX.unitypackage 文件到项目中；在真机测试设备上安装 UWATools。

两个文件夹包含：



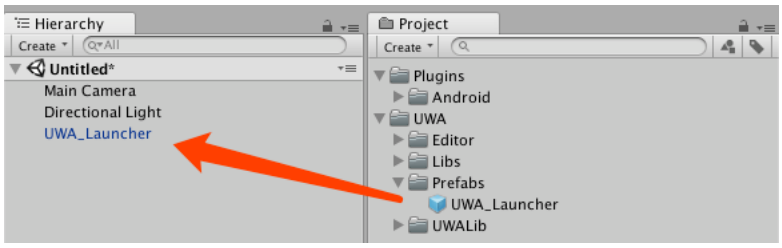
1.2. GOT 导入

1. 导入“UWA_GOT”文件夹中的 unitypackage 文件。
2. 导入后，可以通过菜单栏中的“Tools -> UWA GOT”选项打开“GOT Panel”界面。

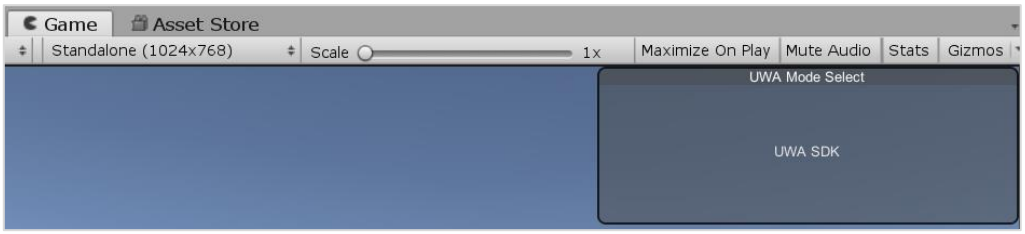


1.3.集成打包

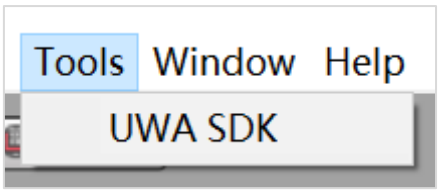
1. 导入“UWA_SDK”文件夹中对应平台的 UWA_SDK_XXX.unitypackage 文件。
2. 在 Unity Editor 中将 UWA/Prefabs 文件夹下的 Prefab 文件拖入到项目的首场景中，且确保不会被强制 Destroy，如下图所示。



3. 如在Game视图的右上角出现如下图所示的UI界面，且无报错信息，说明工具集成完成。

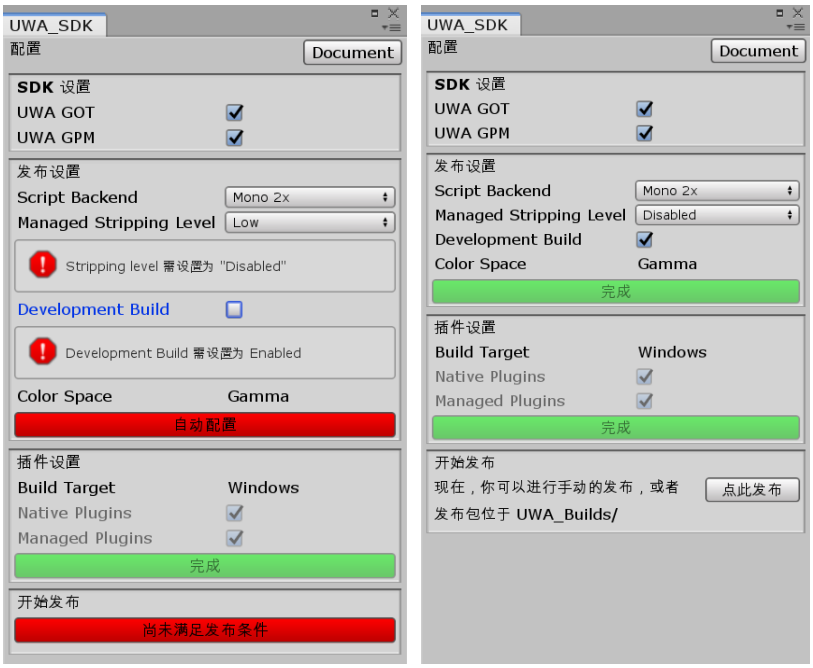


4. 点击菜单栏“Tools -> UWA SDK”，打开 UWA 工具栏。



5. 在“配置”界面上选择需要使用的 UWA GOT 和/或 UWA GPM 工具，按照指引指导配置直到两个按钮变成绿色且显示“完成”则表示配置成功。

   注：UWA GOT 只支持 Development Build，请确保配置勾选 Development Build。UWA GPM 支持 Development Build 或者 Release 版本。



6. 发布版本。

（1） **针对 Android 和 Windows 发布平台：**

建议直接点击“配置”界面上的“点此发布”按钮，完成一键发布操作，发布包存储于 UWA_Builds/Android 或者 UWA_Builds/Windows 文件夹。同时，也可以通过“Build Settings -> Build”进行手动发布。如果通过 BuildPlayer 接口发布，请确保添加 BuildOptions.Development 参数。

（2） **针对 iOS 发布平台：**

按 iOS 版本发布流程执行即可。

注意事项：

1. 支持同时集成多个平台的插件，且多个平台共用同一个 Prefab，切换平台后无需任何修改。
2. 对于 Windows 设备：
   a) 如果截图是黑色的，请尝试把 Color Space 改为 Gamma。
3. 对于 Android 设备：
   a) 无需 Root；
   b) 截屏记录功能只支持 Android 5.0 或以上的系统；
   c) 项目在真机设备上运行并开始 UWA 工具后，若屏幕左上角提示：Write Access Internal (True/False)，请按照以下 4 个方式依次排查：

i. Player Settings 中的 Write Access 需要设置为 External。在打开 UWA SDK 界面时会自动设置；
ii. 部分设备的外部存储权限需要动态开启。可以手动在手机上操作，设置->应用->权限->读写外部存储，选择“允许”，然后再次尝试；
iii. 发布时，外部存储权限添加上了 maxSdkVersion。通过 Android SDK 里的工具 aapt 打印一下 apk 的权限（aapt dump permissions XXX.apk）可确认。解决方案请参考：https://forum.unity.com/threads/gradle-maxsdkversion-read_external_storage.570370/；
iv. 发布时 Android Target API 为 29（Android 10）或以上，且运行在 Android 10 设备上。这种情况是由 Android 最新的“分区存储”机制引起，可以采用两种方法将其禁用：1）把 Target 改为 28（Android 9）或以下；2）在 Manifest 中添加特殊字段，见一下文档中的最末尾部分：https://developer.android.google.cn/training/data-storage/files/external-scoped?hl=zh-cn#opt-out-of-scoped-storage

2. 数据采集

2.1.测试模式

GOT 只支持 Development Build 打包方式。支持四种测试模式，包含：

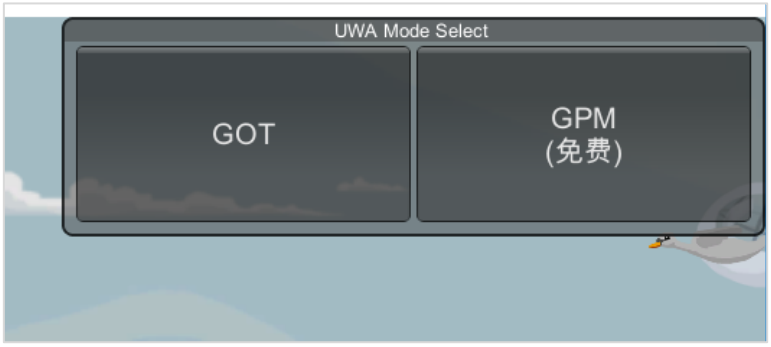
模式	Android	iOS	Windows
Overview（总体性能分析）	√	√	√
Mono（Mono 堆内存分析）	√	×	√
Assets（运行时资源）	√	√	√
Lua（Lua 性能分析）	√	×	√

注：若使用 il2cpp 发布设置时，各发布平台的 Mono 模式均不支持。

Lua 模式仅适用于使用 Lua 的项目。

2.2.数据采集流程

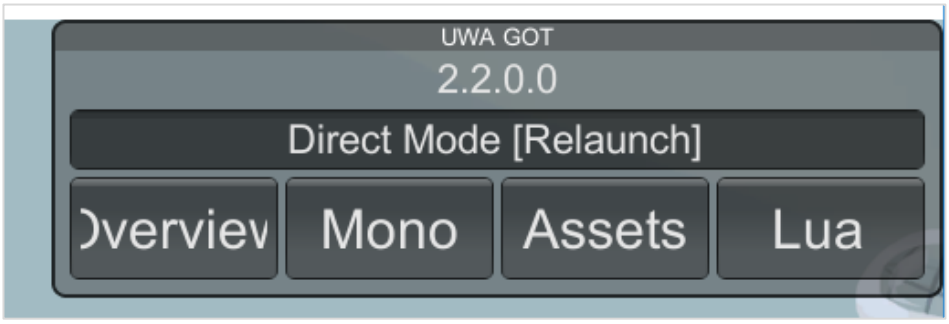
1. 项目集成 GOT 并生成发布包后， 将发布包安装在 Android、iOS 或 Windows 的真机测试设备上。在真机测试设备上打开项目，在界面右上角选择产品（GOT 或 GPM）。



注：若集成打包时“配置”界面（第 1.3 节）只勾选了“UWA GOT”或“UWA GPM”中的一个，则上图只显示对应的产品。

2. 当点击“GOT”按钮后，真机测试设备界面右上角会出现四种模式，点击选择需要的其中一个模式后即开始记录数据。

注：每次测试仅可点选一个模式。



特别介绍：Direct Mode

当点击 Direct Mode 并使按钮变绿后再点击某个模式，项目会先自动退出，并在下一次开启项目后立即自动开启已选的模式。

3. 此时界面相同位置会出现显示测试时长的读秒显示和 Stop 字样，该面板可以拖动。如果希望结束本次测试，点击“Stop”即可，采集的数据保存在设备本地。

3. 数据上传

3.1. 开启 GOT

点击“GOT Panel”界面上的“WIFI”按钮，红框处是本地设备的当前 IP（端口固定为 8099）。

注：在部分 Mac 设备上若 IP 无法获取，则可以通过将 IP（不需要填端口）写入 UWA/UWALib 下的 serverip.txt 文件中进行手动配置。

3.2. 针对 Android 和 Windows 的数据上传方式

1. 先确认用于测试的真机设备与 PC 处于同一网段，在测试设备上安装并打开“UWA Tools”，在以下输入框中输入第 3.1 节中获得的 IP（不需要填端口），点击“检测”按钮，如果左侧圆点变绿，即表示可以连接到本地服务器。

   注：请关闭 VPN、防火墙设置。

2. 在 UWA Tools App 中可查看本机已测试的数据列表，选择需要分析的 GOT 数据，先点击“GOT”按钮后再点击“提交数据”按钮上传数据。

3. 点击“提交数据”后，即可进入下图中的上传界面。当上传界面中的进度条结束，提示返回，即上传成功。

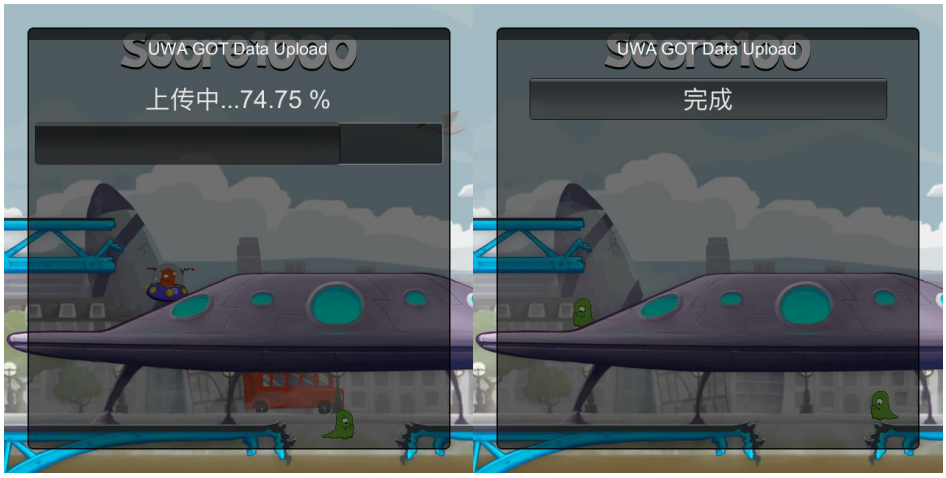
3.3. 针对 iOS 的数据上传方式

先确认用于测试的真机设备与 PC 处于同一网段，在“UWA GOT Data Upload”面板中选择“GOT”页签，在以下输入框中输入第 3.1 节中获得的 IP（不需要填端口），点击“确定”按钮。如果 IP 显示变绿，即表示可以连接到本地服务器。

完成测试点击“Stop”后，会自动呼出“UWA GOT Data Upload”面板。

注：iOS 平台的数据上传须在测试完成之后立即完成，才能确保测试数据被有效使用，不能退出项目。

点击“提交数据”后，即可进入下图中的上传界面。当上传界面中的进度条结束，出现“完成”按钮提示返回，即上传成功。



3.4. 通过 UWA API 的数据上传方式

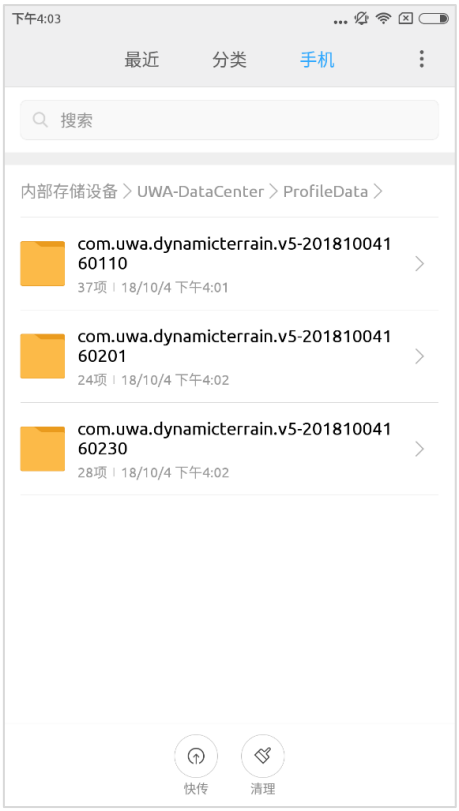
该方式仅支持 GOT 模式下测试的数据。

在游戏运行时，通过调用 UWA API 将测试数据上传至 GOT Online 上，具体的使用方法见附录 1 中对 **UWAEngine.Upload** 部分的介绍文档。

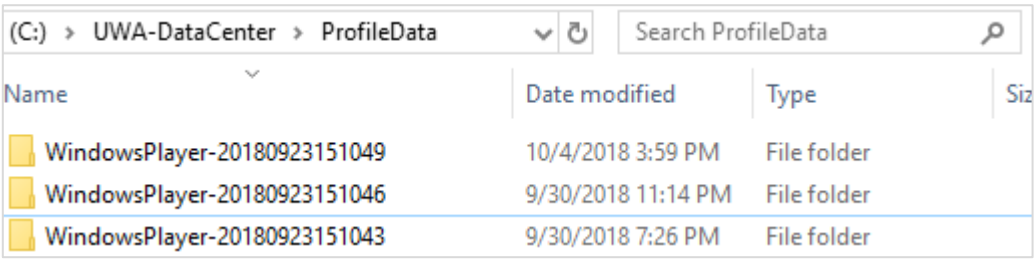
3.5. 注意事项

1. GOT 中的 Lua 模式仅限 GOT Online 服务可用。
2. 如果 UWA Tools App 无法与本地服务器连接，或其他原因导致数据无法传输，可通过以下路径找到测试数据：

Android 设备：UWA-DataCenter/ProfileData

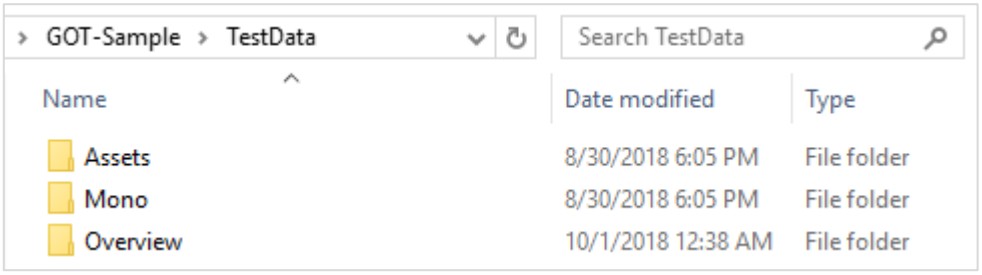


Windows 设备：C:/UWA-DataCenter/ProfileData



iOS 设备：Documents/UWA-DataCenter/ProfileData

并将其中的文件夹剪切至 GOT 所在的工程目录下与 Assets 同级的 TestData 目录中，TestData 中包含了若干个文件夹，对应了若干种测试模式，将上述的文件夹放入对应模式的文件夹即可。



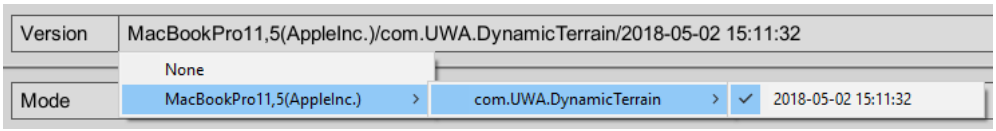
4. 查看报告（本地编辑器内查看）

在“GOT Panel”界面中选择测试模式，即可查看已上传至本地服务器的测试数据报告，共三种测试模式：Overview、Mono 和 Assets。若查看 Lua 模式的数据报告需要同时满足项目使用了 Lua 且已购买 GOT Online 测试时长（使用方式详见《UWA SDK 使用说明》第 5 节）。

4.1. Overview（总体性能分析）

1. 逻辑代码的 CPU 开销

（1） 在“Version”中选择需要查看的测试版本。



（2） 选择后，对应的数据将被载入并进行分析和展示。它主要包括以下视图：

i. **CPU 开销走势图**

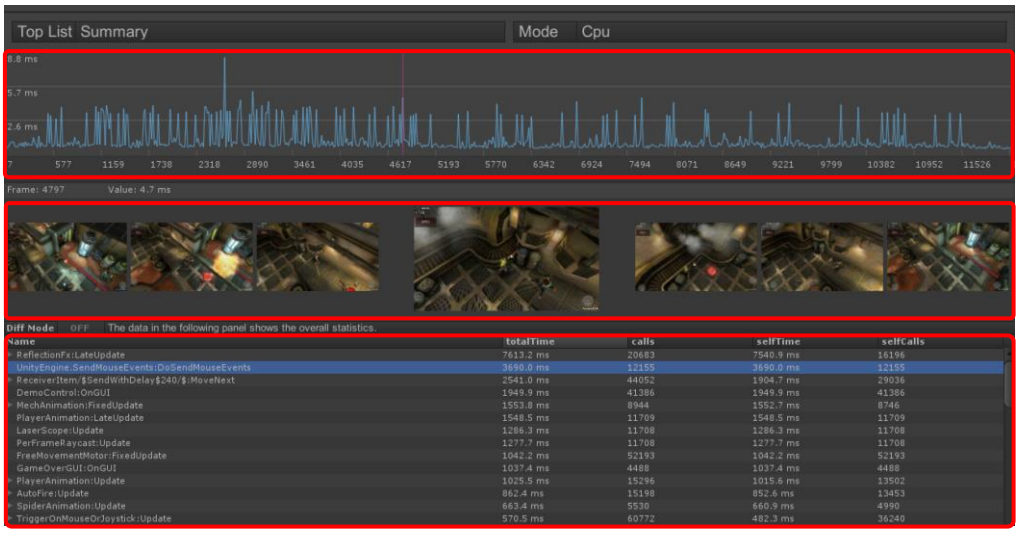
您可以选择任何一个函数，查看它在项目运行时的 CPU 开销。

ii. **截屏视图**

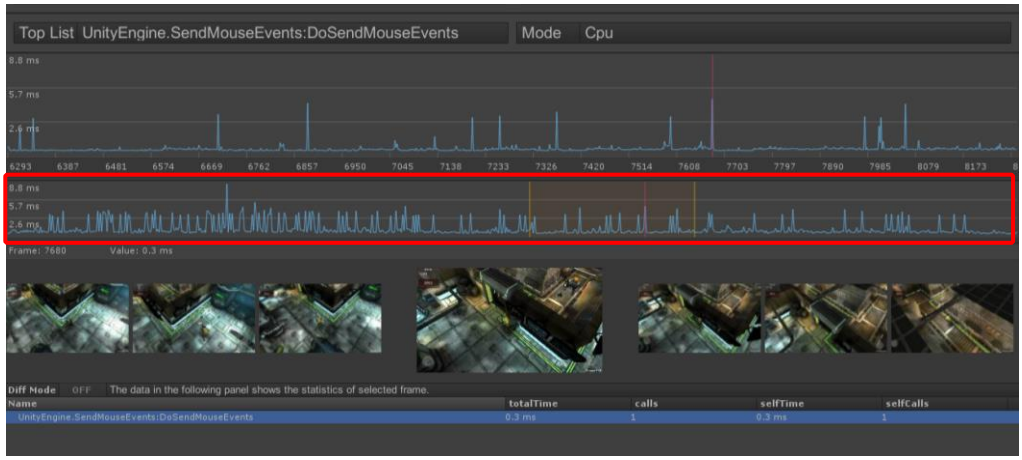
您可以在 CPU 开销走势图中选择任何一帧，截屏视图会随之切换到与其相对应的运行截屏。

iii. **CPU 耗时分析视图**

UWA 将逻辑代码的 CPU 耗时进行分析，并将最为耗时的代码展示在此。您可以通过 UWA API 统计指定的代码段的 CPU 耗时，具体用法见附录 1。



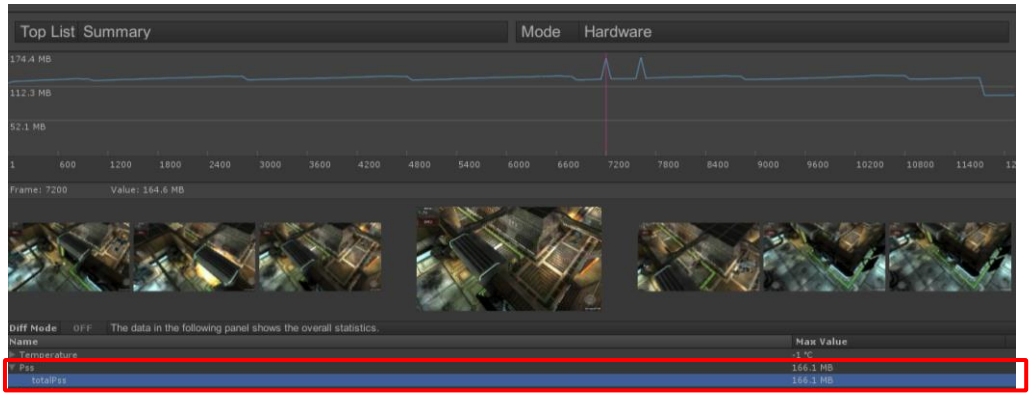
（3） 在此面板中，您既可以选择“Total”模式，查看逻辑代码的整体 CPU 耗时。也可以选择查看具体的逻辑代码。同时，您可以通过调整关注区域视图中的滑块，来重点查看您关注区域的 CPU 开销。



2. 硬件设备信息

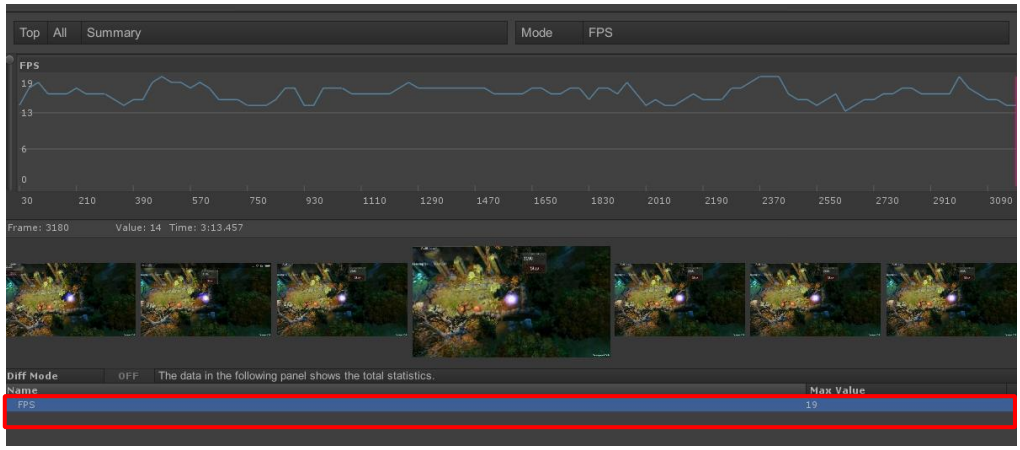
在“Mode”中选择“Hardware”，即可查看项目运行时的硬件设备运行信息，主要包括：硬件设备的内存信息。

注：在 Windows 上主要显示 WorkingSet 内存在项目运行时的走势。



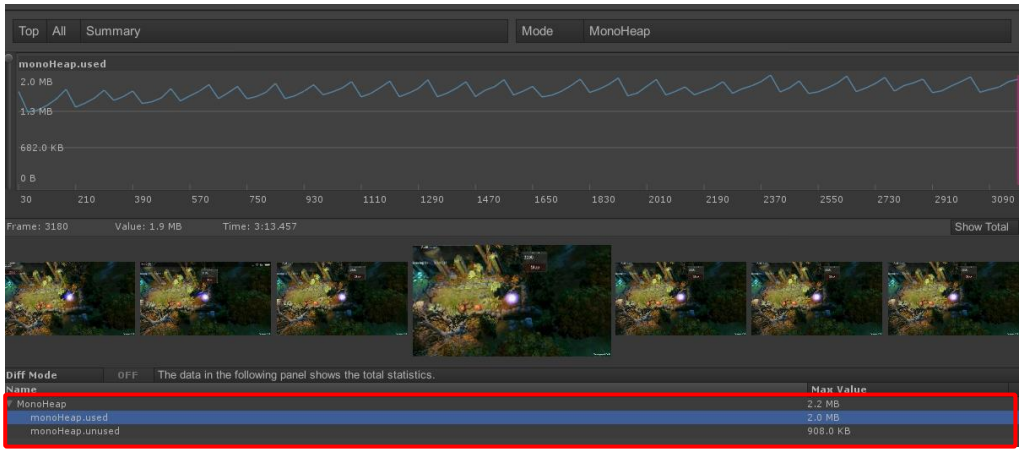
3. FPS 信息

在“Mode”中选择“FPS”，即可查看项目运行时的 FPS 统计。



4. Mono Heap 信息

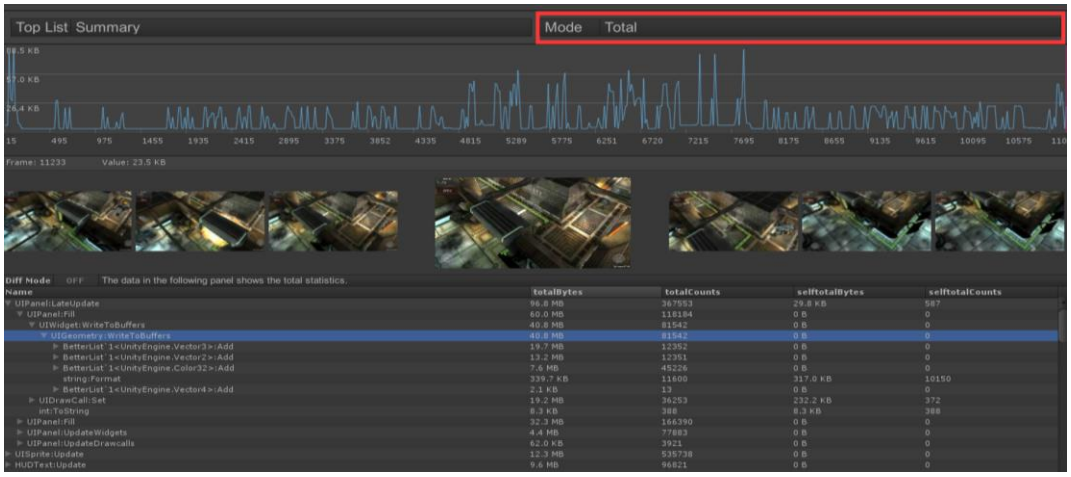
在“Mode”中选择“MonoHeap”，即可查看项目运行时的 Mono 堆内存总量统计，包括了使用中的和未使用中的部分。



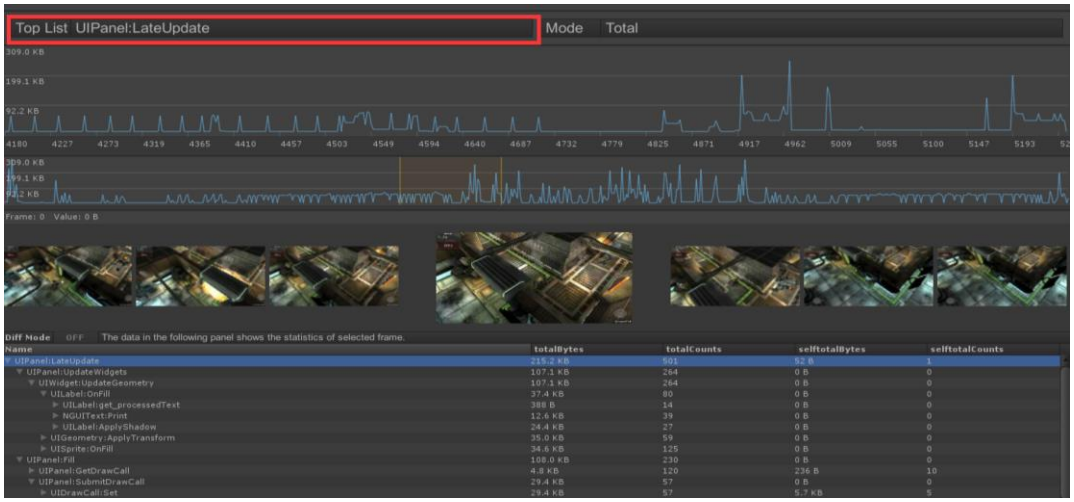
4.2. Mono（Mono 堆内存分析）

1. 代码堆内存累积分配

（1） 在“Mode”中选择“Total”，您即可查看项目运行时每个函数的总体堆内存分配情况；

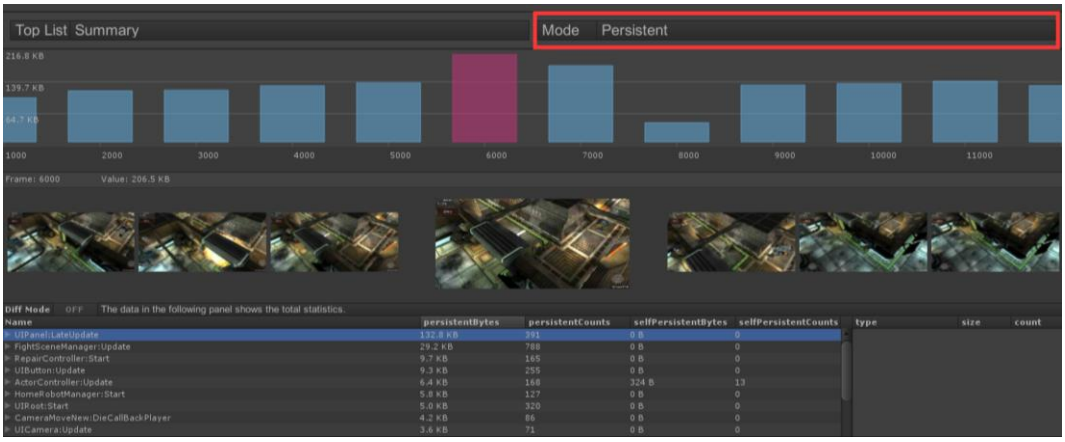


（2） 在“Top List”中选择具体的函数名称，您就可以看到相应函数的具体堆内存分配情况，并且通过与图表进行交互来查看任何一帧的具体堆内存分配。

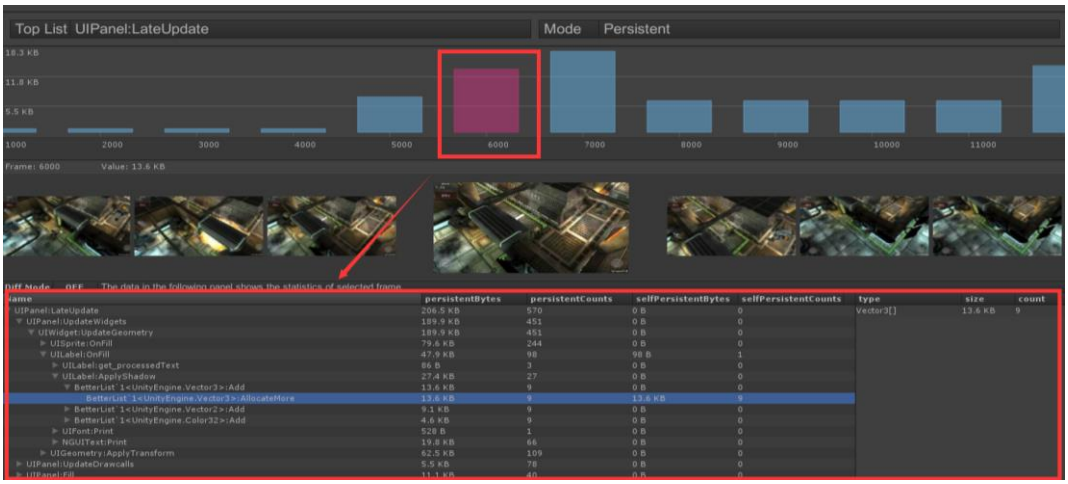


2. 代码堆内存泄露分析

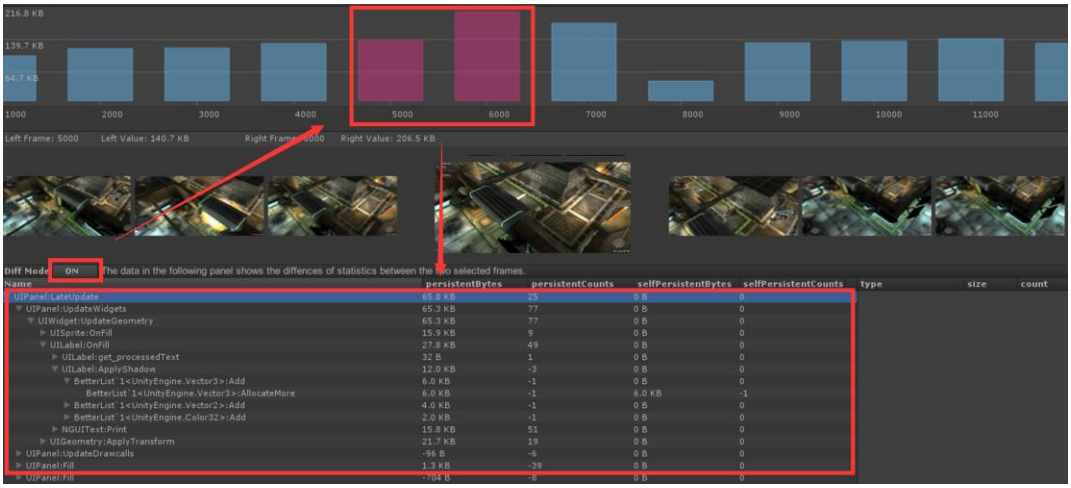
（1） 在“Mode”中选择“Persistent”，您即可查看项目运行时每个函数在 Mono 中的真实驻留情况。UWA 默认是每 1000 帧分析一次 Mono 堆内存快照，将函数真实的堆内存驻留情况以柱状图的形式进行显示。



（2） 在“Top List”中选择具体的函数名称，您就可以看到相应函数的具体堆内存分配情况，并且通过与图表进行交互来查看详细堆内存驻留情况。同时，当 selfPersistentCounts 不为 0 时，点击可以查看由该函数生成的、驻留在堆内存中的变量类型。



（3） 在 Persistent 模式下，您可以比较两次堆内存统计的差异，从而来快速定位堆内存变化的出处。在“Diff Mode”中选择“ON”，即可开启该功能。选择任意两个柱状图，您则可以快随比较两次堆内存占用的差异。

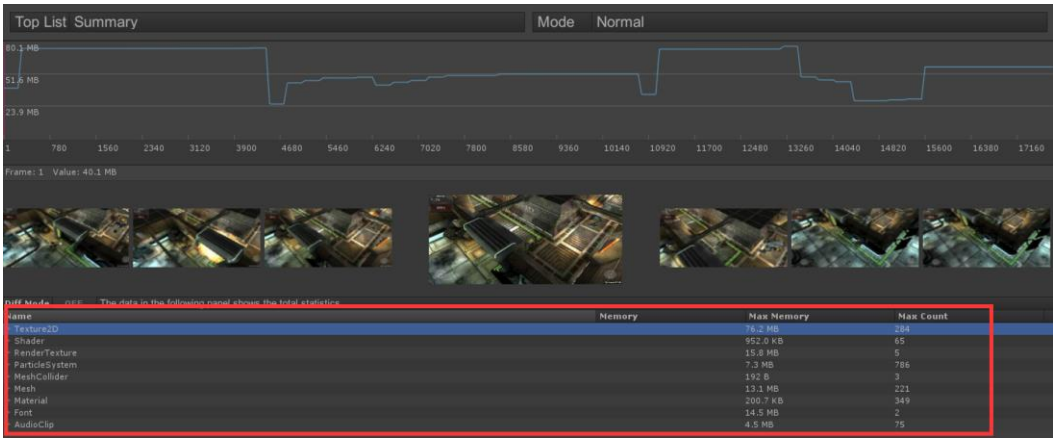


4.3. Assets（运行时资源）

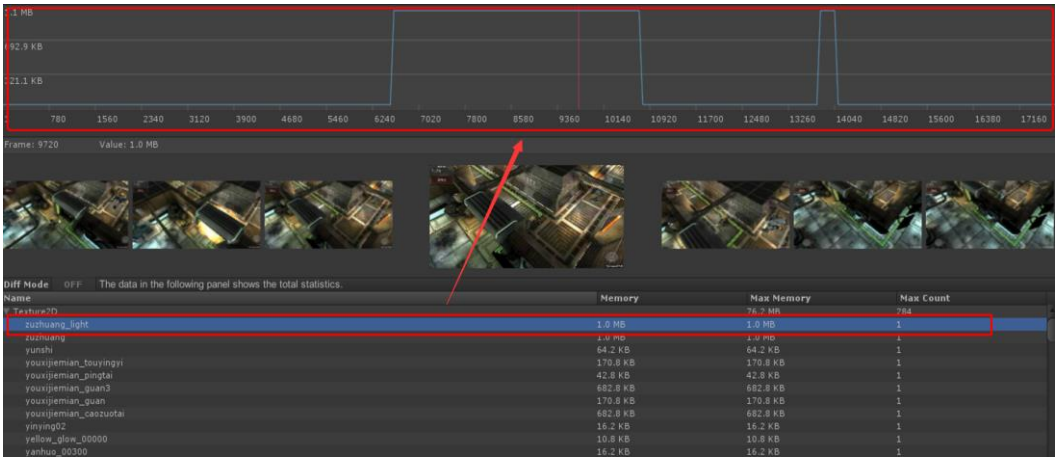
点击“Assets”按钮，即可查看项目运行时资源的具体使用情况。它主要包括以下功能：

1. 资源使用情况

（1） 可以查看重点资源在项目运行时的内存占用情况。



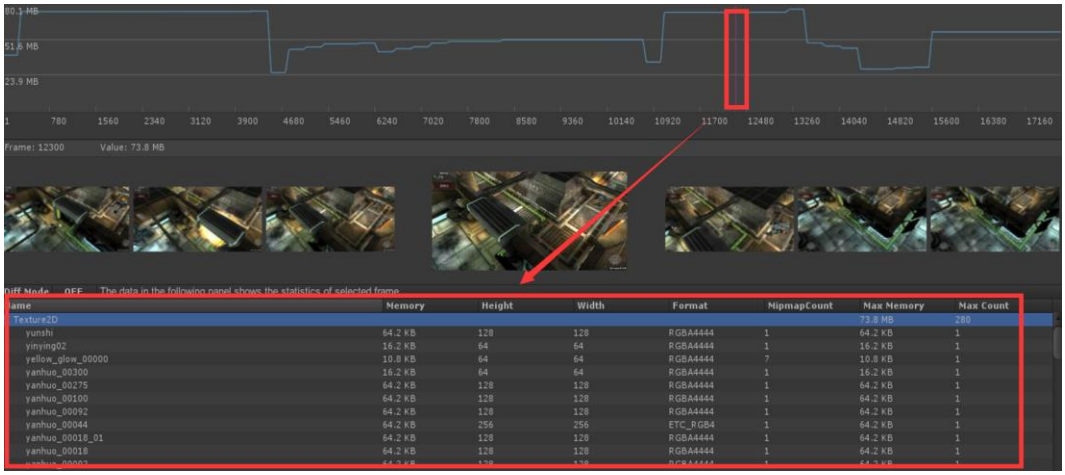
（2） 可以查看具体资源在项目运行时的使用情况。



2. 查看每帧中资源的具体使用情况

（1） 在 TopList 中选择您想查看的资源类型。

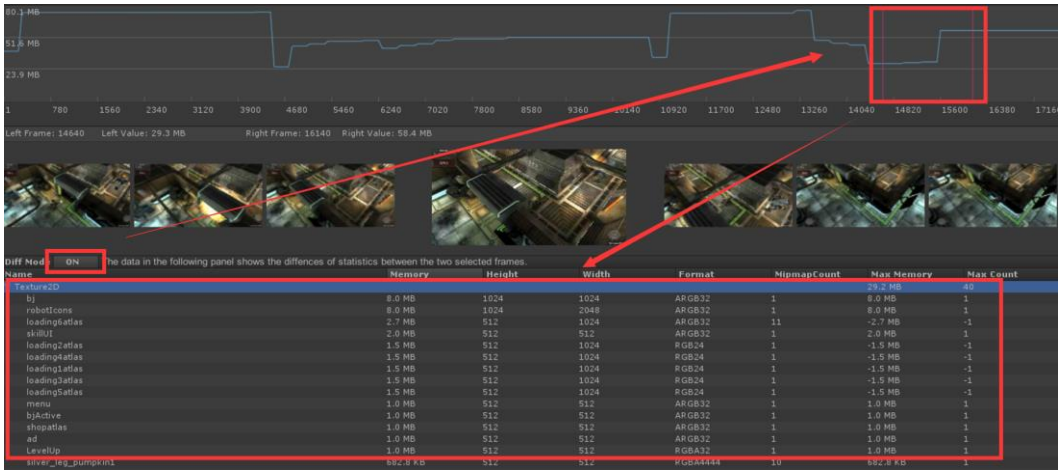
（2） 点击资源使用走势图，即可查看每帧该类资源或某个特定资源的具体使用情况。



3. 资源泄露分析

您可以通过比较任意两帧的资源变化情况，来分析是否存在资源泄露等问题。

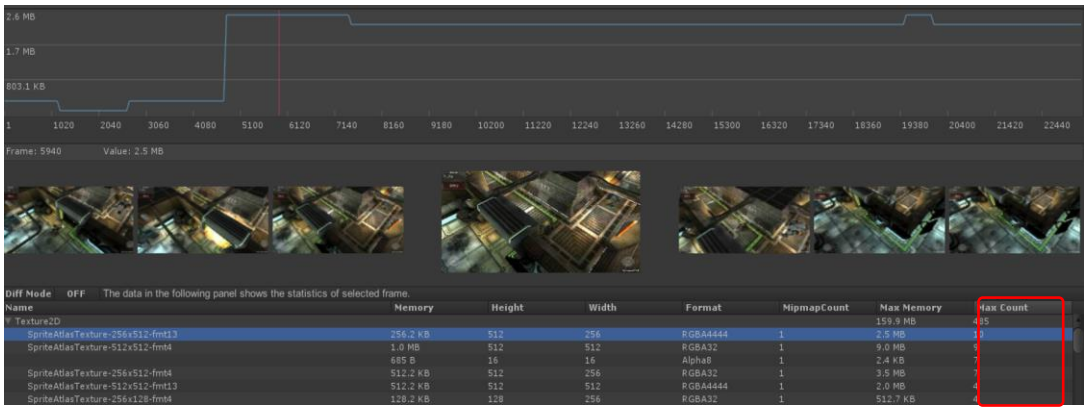
（1） 将“Diff Mode”设置为“ON”，选择资源使用视图中的任意两帧，即可查看资源的变化情况。



（2） 上图为第 16140 帧与第 14640 帧的 Texture 比较情况。其中，“Max Memory”中为正值的资源表示为第 16140 帧中的新增资源，而负值的资源则为第 16140 帧中的减少资源。通过这种比较，即可帮您快速定位具体的资源变化量和解决资源泄露等问题。

4. 资源冗余分析

项目运行过程中，内存中的资源很有可能出现冗余情况。对此，建议您详细查看资源数据展示界面中的“Max Count”数值， “Max Count”大于 1 的资源存在冗余问题的风险较高。Max Count 是指项目运行过程中，某一资源在某一帧中的最大资源使用数量。



注：Max Count 资源数量大于 1，并不能 100%说明该资源存在冗余，也有可能是内存中确实存在两个资源名称、内存以及各个属性均相同的资源。因此，我们将 Max Count 大于 1 的资源称为“疑似”冗余资源。

4.4. 通过 GOT 使用 GOT Online 的方式

对于已经存在于 PC 的 GOT 测试数据，可以通过以下方式将数据上传到 GOT Online（GOT Online 详见《UWA SDK 使用说明》第 5 节）。

1. 在“GOT Panel”界面中选择“GOT Online”。登陆 UWA 账号。

2. 选择任意一份在编辑器中已存在且打开的测试数据报告，先在左侧项目列表的下拉框中选择需要同步到网页的项目名称，再点击报告页的菜单栏“同步到 GOT Online”按钮。


New Project
剩余时间：4604分44秒
同步到 GOT Online
New Project
DynamicTerrain
2020-5-2 14:29:17

附录 1：UWA API 的介绍和用法

UWA.Tools.PrepareForBuild
public static void PrepareForBuild()

该函数仅用于 Editor 中：通过脚本设置 SDK 所需要的发布设置，而不需要手动打开 SDK Integration 界面进行配置。其中修改的设置包括：
1） Stripping Level 设置为 Disabled；
2） Strip Engine Code 设置为 Disabled；
3） Development Build 设置为 True；
针对安卓平台，还会额外将 Write Access 设置为 External (SDCard)。

注：如果使用 BuildPlayer 接口发布 App，则需确保添加 BuildOptions.Development 参数。

UWAEngine.StaticInit
public static void StaticInit();

支持 GOT 和 GPM 模式，该函数可用于：通过脚本动态挂载 UWA_Launcher，而不需要提前手动将 UWA_Launcher 的 Prefab 放入场景。

UWAEngine.Tag
public static void Tag(string tag);

只支持 GOT 模式，该函数可用于：标记测试区间，使用后将会覆盖 Unity 场景名。线上报告中看到的场景名，以及区间统计将会以 tag 为准。

区间	帧数	FPS(帧/秒)		
		均值	最小值	最大值
A	602	28.1	5	30
B	1831	13.44	11	27
C	5213	25.3	21	30

UWAEngine.PushSample/PopSample
public static void PushSample(string sampleName);
public static void PopSample();

只支持 GOT 模式，该函数可用于：统计自定义代码段 CPU 耗时，从而更快地定位脚本的性能瓶颈。

参数 sampleName 表示自定义的函数标签，UWAEngine 会对 PushSample 和 PopSample 之间的代码段统计 CPU 开销，并在 UWA GOT 中的统计面板中进行显示，该 API 支持嵌套调用。其具体用法如下

```
UWAEngine.PushSample("MyCode");
// some code …
UWAEngine.PopSample();
```

最终在 Overview 界面中，可以看到自定义的函数标签，及其具体耗时（下图中 A~E 都是自定义函数标签）。

Name	percent	selfPercent	totalTime	calls	selfTime	selfCalls
Perf:Update	100.00 %	0.04 %	90.5 ms	3	0.0 ms	1
A	54.40 %	0.03 %	49.2 ms	11	0.0 ms	1
B	54.37 %	0.27 %	49.2 ms	110	0.3 ms	10
C	54.09 %	2.98 %	49.0 ms	1100	2.7 ms	100
D	51.11 %	30.92 %	46.3 ms	11000	28.0 ms	1000
E	20.19 %	20.19 %	18.3 ms	10000	18.3 ms	10000

请确保 PushSample 和 PopSample 是成对使用的。如果两者之间使用了 return 语句提前退出代码段（或者在协程中使用 yield return 提前跳出代码段），则会造成 PushSample 和 PopSample 的配对不准确，从而导致数据错误。
另外，请注意在同一帧中 PushSample 和 PopSample 的调用次数不宜过多。初步统计，在中低端的设备上，10000 次的调用会导致接近 50ms 的额外开销。

UWAEngine.LogValue

```
public static void LogValue(string valueName, float value);
public static void LogValue(string valueName, int value);
public static void LogValue(string valueName, bool value);
public static void LogValue(string valueName, Vector3 value);
```

只支持 GOT 模式，该函数可用于：统计每帧中自定义标签的数值变化，从而可视化关键变量的走势。

参数 valueName 表示自定义的变量标签，value 表示对应的变量的当前值。

UWAEngine.AddMarker

```
public static void AddMarker(string valueName);
```

只支持 GOT 模式，该函数可用于：统计每帧中自定义标签被标记的次数，从而该接口可以用来统计如 Lua 调用 C#接口的次数。以 SLua 为例，适当修改 SLua 的代码生成器，自动在每个 Wrap 函数中插入对应语句即可：

```
[SLua.MonoPInvokeCallbackAttribute(typeof(LuaCSFunction))]
[UnityEngine.Scripting.Preserve]
static public int Find_s(IntPtr l) {
    UWAEngine.AddMarker("Shader.Find");
    try {
        System.String a1;
        checkType(l,1,out a1);
        var ret=UnityEngine.Shader.Find(a1);
        pushValue(l,true);
        pushValue(l,ret);
        return 2;
    }
    catch(Exception e) {
        return error(l,e);
    }
}
```

UWAEngine.SetOverrideLuaLib

public static void SetOverrideLuaLib(string luaLib)

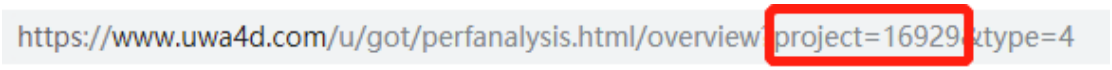
只支持 GOT 模式，在 Lua 模式中，该函数可用于：通过脚本指定自定义 Lua 库的名字，如 libgamex.so。如果使用 ulua/tolua/slua/xlua 的默认 lua 库，则不需要使用该接口来指定。

UWAEngine.Upload

public static void Upload(Action<bool> callback, string user, string pwd, int projectId, int timeLimitSec)

public static void Upload(Action<bool> callback, string user, string pwd, string projectName, int timeLimitSec)

callback 为上传结束后的回调，中的 bool 参数为上传结果；
user/pwd 为登录 UWA 网站的用户名和密码；
projectId/projectName 为上传项目的 ID 或名称；
timeLimitSec 为可上传的数据的时间上限（以防因为逻辑错误，自动地上传了时间过长的数据）。
项目 ID 的获取，为打开项目后，URL 中该字段：



该接口只支持 GOT 模式，可用于：在游戏运行时，通过传入账号信息，项目 ID/项目名来进行测试数据的上传。在 IL2CPP 模式下，该接口仅支持 2018 及以上版本。在上传过程中，以及上传成功后，都会在游戏界面上显示以下的上传界面：

注意事项：

1. 如果通过项目名上传，但该参数为空，那么会自动创建以“应用名”为名的项目；
2. 如果通过项目 ID 上传，需要确保测试数据的平台（Android/iOS/Windows）、模式（Overview/Assets/Mono/LUA）与该 ID 对应的项目是一致的，否则上传界面上会出现 InvalidProjectId 的报错；
3. 如果测试时间大于给定的上限，则上传界面上会出现 Duration [测试时长] has exceeded the limit [时间上限] 的报错。
4. 上传前，请确保“GOT Online”服务有足够的余额。

UWAEngine.Start

public static void Start(Mode mode)

只支持 GOT 模式，该函数可用于：通过脚本动态开启指定类型的测试，而不需要手动点击右上角的 UI 按钮。

UWAEngine.Stop

public static void Stop()

只支持 GOT 模式，该函数可用于：通过脚本动态关闭当前的测试，而不需要手动点击 Stop 按钮。

注：

UWAEngine.Start/Stop 在一次游戏运行中只有第一次调用会生效，无法反复使用。