

GOT (Game Optimization Toolkit) 使用说明

安装环境: Windows & Mac OS

支持发布/测试平台: Android & iOS & Windows & HarmonyOS NEXT

目录

1. GOT 工具.....	2
1.1. 工具下载.....	2
1.2. UWA GOT 导入	2
1.3. UWA SDK 集成打包	3
1.4. 本地测试 - 注意事项.....	6
2. 数据采集.....	8
2.1. 测试模式.....	8
2.2. 数据采集流程.....	8
2.3. 数据支持情况.....	16
3. 数据上传.....	18
3.1. 开启 GOT.....	18
3.2. UWA Tools 上传 (IOS 与 HarmonyOS NEXT 除外)	18
3.3. 游戏内直接上传.....	19
3.4. 注意事项.....	20
4. 查看报告 (本地编辑器内查看)	23
4.1. Overview (总体性能分析)	23
4.2. Mono (Mono 堆内存分析)	25
4.3. Resources (运行时资源)	27
4.4. 通过 GOT 使用 GOT Online 的方式	29
附录 1: UWA API 的介绍和用法	31

1.GOT 工具

1.1. 工具下载

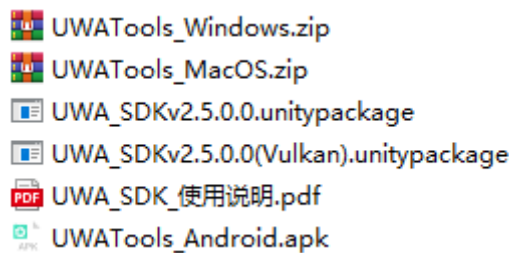
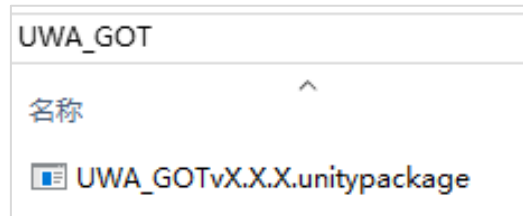
1. 打开 UWA 网站，登录 UWA 账号；
2. 打开下载页面：<https://www.uwa4d.com/#download>，可以看到“GOT”产品的“购买”及“文档下载”，购买之后即可下载工具，文档可直接下载查看。



3. 下载包为一个 ZIP 文件，其中包含了两个文件夹：UWA_GOT、UWA_SDK。



两个文件夹分别包含：



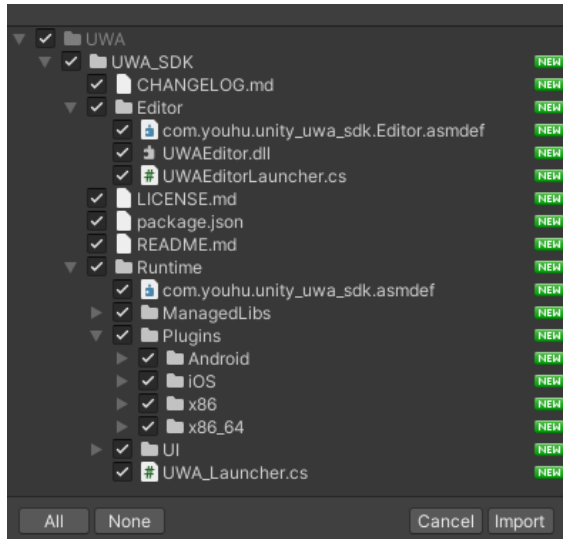
1.2. UWA GOT 导入

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。

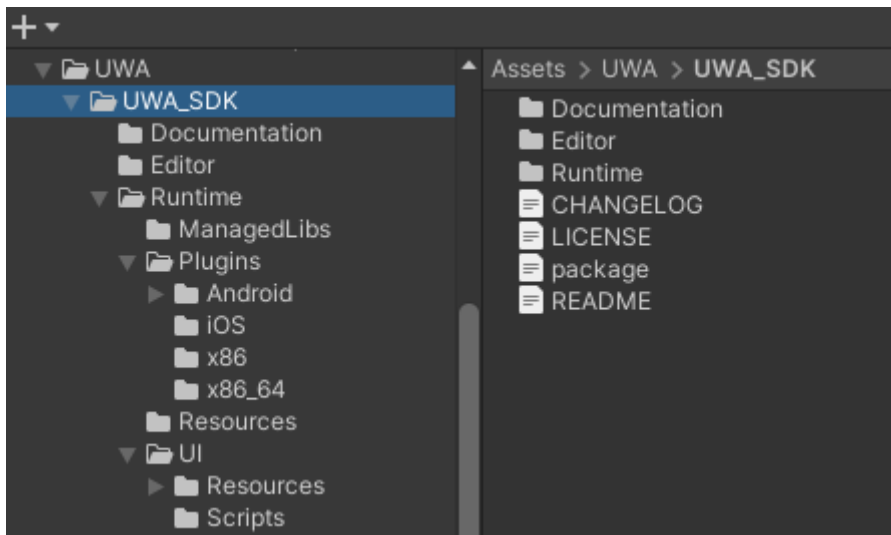


1.3. UWA SDK 集成打包

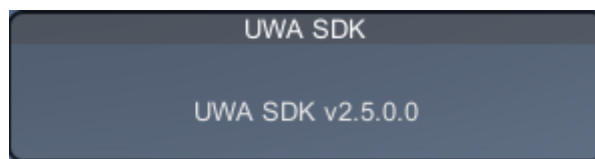
1. 选择“UWA_SDK”文件夹中的 UWA_SDKvX.X.X 或者 UWA_SDKvX.X.X(Vulkan).unitypackage，其中 Vulkan 版 SDK 可获取安卓端 Vulkan 渲染相关统计，但仅支持 Unity 2019.4 及以上，且 Android Minimum API Level 为 24 及以上的项目。
2. 将选中的 unitypackage 文件拖入项目中，并点击“Import”按钮进行导入。



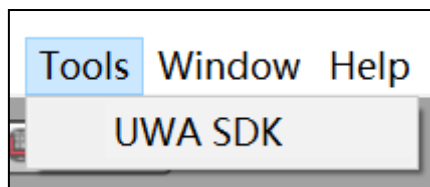
3. 导入之后，UWA SDK 的全部文件位于/Assets/UWA/UWA_SDK 文件夹下，便于管理。



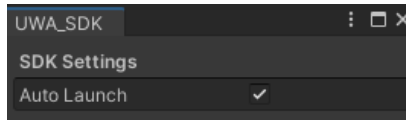
4. 点击 Play 运行后，如在 Game 视图的右上角出现如下图所示的 UI 界面，且无报错信息，说明工具集成完成。



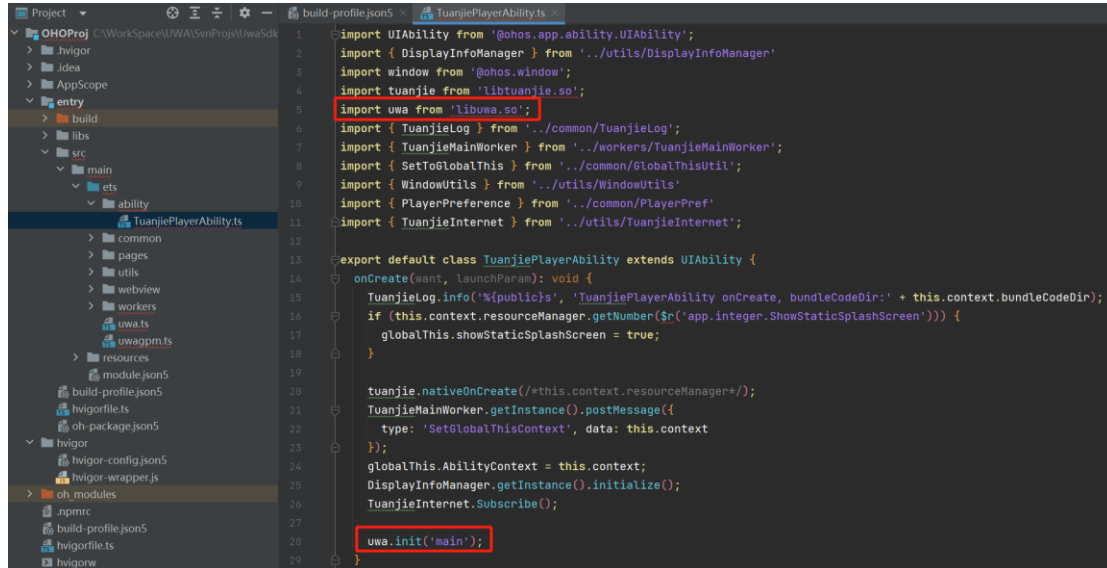
5. 点击菜单栏 “Tools -> UWA SDK”，打开 UWA 工具栏。



6. 在“配置”界面上调整 Auto Launch 的设置，默认勾选时，运行项目默认加载显示 UWA SDK UI 界面；否则不加载，需要使用 UWA API 启动测试，API 为 UWAEngine.StaticInit，具体使用方式见附录 1。



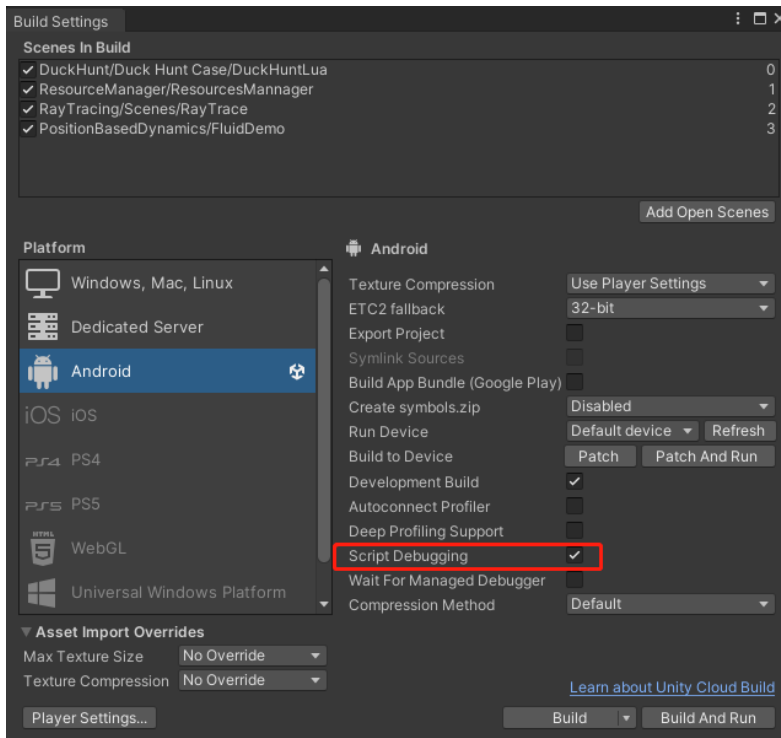
7. 对于 HarmonyOS NEXT 平台，需要在导出后的 DevEco Studio 工程的 TuanjiePlayerAbility.ts 中添加两行代码，具体位置见红框处：



8. 发布版本。

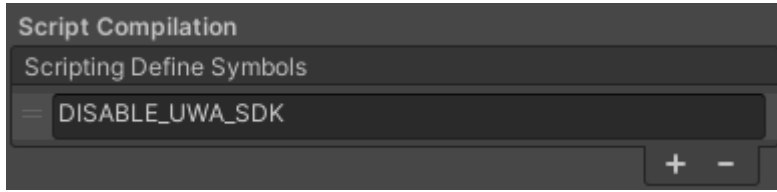
按正常发布流程执行即可。当通过 BuildPlayer 接口发布时，如需支持 UWA GOT 或者真人真机测试服务，请确保添加 BuildOptions.Development 参数。

如果在 IL2CPP 下进行 Android 的 Mono 模式测试，需要勾选 Script Debugging。当通过 BuildPlayer 接口发布时，请确保添加 BuildOptions.AllowDebugging 选项。



注：

在发布正式 Release 版本时,若需要移除 UWA SDK 进行打包,可以直接删除 UWA_SDK 文件夹,或者在 Player Settings 中添加 DISABLE_UWA_SDK,从而屏蔽 SDK 的集成。



在自动打包流水线中,可以通过以下方式决定 UWA SDK 是否集成:

PlayerSettings.SetScriptingDefineSymbolsForGroup 添加或删除 DISABLE_UWA_SDK 宏。

1.4. 本地测试 - 注意事项

1. 对于 Windows 平台

(1) 截图采集异常

如果截图是黑色的,请尝试临时把 Color Space 改为 Gamma。

2. 对于 Android 平台与 HarmonyOS NEXT 平台

(1) GPU Counter 采集异常

a. 确保设备在 GPU Counter 的支持列表中,具体型号参见:

<https://www.uwa4d.com/main/supported.html>

b. 在支持的型号中,若 Adreno GPU 且为 Android 12 及以上的设备,如果无法获取 GPU Counter 数据,可以尝试通过以下 adb 命令开启:adb shell "echo 1 > /sys/class/kgsl/kgsl-3d0/perfcounter"。但部分设备可能因为硬件厂商的限制无法进行该操作,如小米品牌设备。

c. 在支持的型号中,若 Mali GPU 的设备,如果无法获取 GPU Counter 数据,可以尝试重启设备(有可能需要多次)。

(2) SOC 数据采集异常

SOC 数据包含 GPU 频率、GPU 利用率等数据。

a. 目前仅支持 MTK 和 HISI 芯片,具体型号参见:

<https://www.uwa4d.com/main/supported.html>

b. 在 HISI 芯片的支持型号中,Android 系统要求最低为 Android 10,在部分设备上可能要求更高版本的 Android 系统。

(3) 截图采集异常

截图功能因为安卓权限等限制,只支持 Android 5.0 或以上的系统,SDK 中为了兼容不同的平台设置提供了 3 个 AAR 文件,这三个文件分别为:

uwa_pc.aar: 使用 PixelCopy 接口

uwa_mp.aar: 使用 MediaProjection 接口

uwa_mp28.aar: 使用 MediaProjection 接口

在 Unity Editor 打包时,SDK 会自动根据平台设置选择合适的 AAR 文件;如果使用其它打包方式,需要按照以下规则选择合适的 AAR 文件:

a. 当 TargetApi>=29 (Android 10) 时,勾选 uwa_pc.aar 和 uwa_mp.aar 的 Android 平台开关,取消勾选 uwa_mp28.aar 的 Android 平台开关;

b. 当 TargetApi<29 时，勾选 uwa_pc.aar 和 uwa_mp28.aar 的 Android 平台开关，取消勾选 uwa_mp.aar 的 Android 平台开关。

2. 数据采集

2.1. 测试模式

Development Build 打包方式。支持五种测试模式，包含：

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

Release Build 打包方式，仅支持 Overview 模式极简版，详见官网 [Demo 页面](#)。

注：Lua 模式仅适用于使用 Lua 的项目。

2.2. 数据采集流程

项目集成 UWA SDK 并生成发布包后，将发布包安装在目标平台的真机测试设备上。在真机测试设备上打开项目，即可看到当前支持的测试模式。

右侧 UI 中，信息按钮可查看当前设置信息，包括打包方式、图形 API、Unity 版本以及 SDK 版本；退出按钮则是关闭 SDK。



注：每次测试仅可点选一个模式。

1. 点击右边齿轮按钮，可以对各个模式进行配置，配置完成后点击“Save & Start”，保存设置并开启测试：

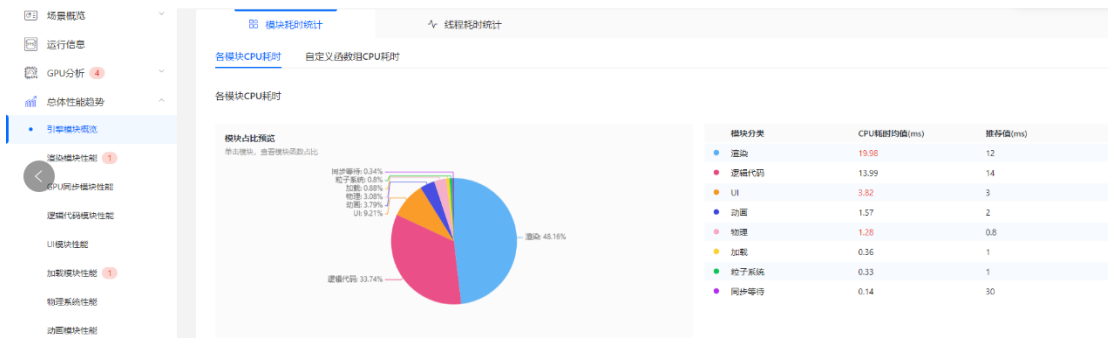
(1) Overview 模式



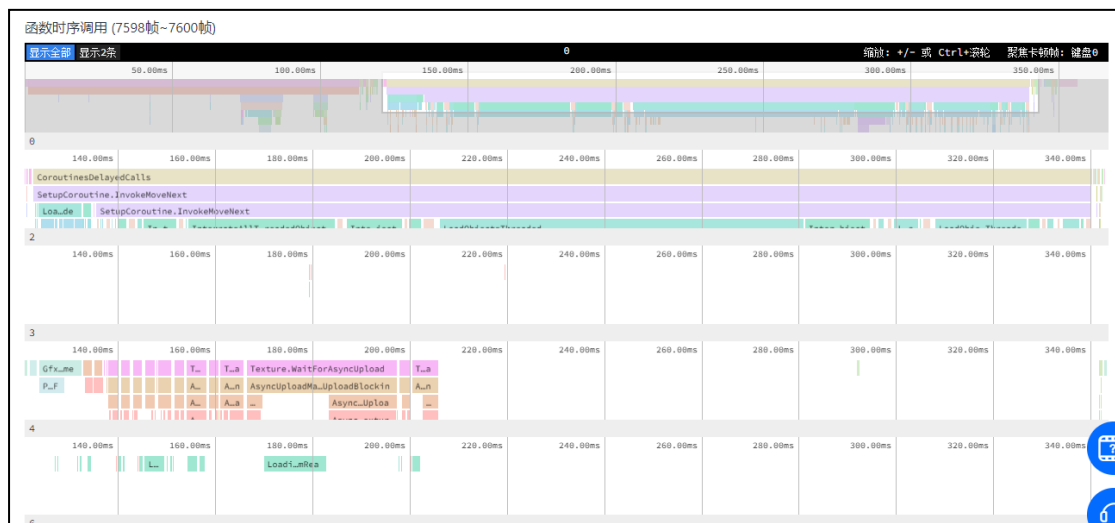
配置说明：

- 堆栈获取：
勾选时，将支持各模块堆栈信息的获取；

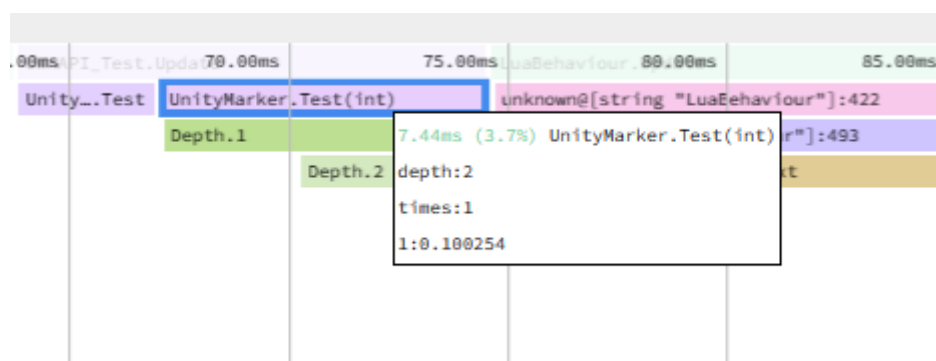
不勾选时，将关闭所有堆栈信息的获取。



- Timeline: 勾选后，将开启函数时序调用功能



同时，支持自定义参数功能：使用 ProfilerMarker API 标记参数后，会在报告 Timeline 界面出现对应数据统计（目前仅支持一条）。



注：在 Android 平台的部分设备上，开启函数时序调用（Timeline）功能会导致项目运行时的额外耗时偏高。

- Unity API（仅支持 Android）：勾选后，将开启 Unity API 调用统计功能
可以在函数总体堆栈信息中看到：AudioSource.PlayHelper、ParticleSystem.Simulate、CharacterController.Move_Injected、SystemInfo.GetBatteryLevel、SystemInfo.GetBatteryStatus、Application:get_internetReachability、GameObject.Internal_AddComponentWithType 等函数的

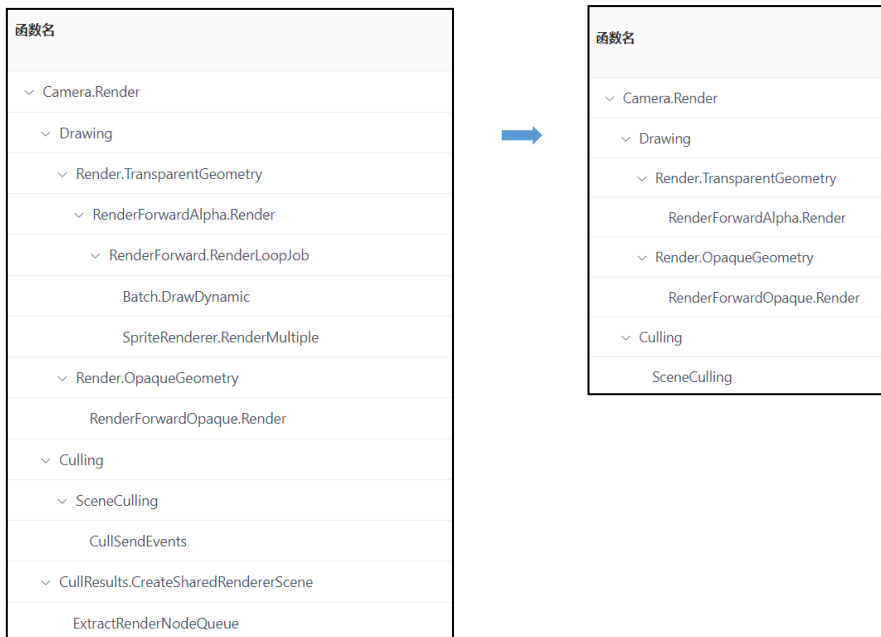
运行信息。

函数总体堆栈信息							
函数名	总体占比	自身占比	总耗时(ms)	自身耗时(ms)	调用次数	显著调用帧数	
<input checked="" type="checkbox"/> XGameAppCorMono.Update	100.00%	84.75%	63205.400	53567.810	74159	1271	
<input type="checkbox"/> > Transform.SetParent	6.05%	3.28%	3822.180	2072.010	28723	82	
<input type="checkbox"/> > GameObject.Activate	2.78%	0.04%	1759.810	23.530	8651	0	
<input type="checkbox"/> > GameObject.Deactivate	2.15%	0.04%	1361.960	27.620	8729	0	
<input type="checkbox"/> > Instantiate	1.51%	0.03%	956.820	19.960	2469	0	
<input type="checkbox"/> GameObject.Internal_AddComponentWithType	1.49%	0.61%	938.880	387.310	2267	1	

注：在部分设备上，开启 Unity API 调用统计功能会导致部分项目运行不稳定的问题。

● 详细堆栈：

堆栈差异的示例（左：勾选；右：不勾选）：



注：在 Android 平台低端设备上，堆栈的获取会产生一定的额外开销。

不勾选，减少堆栈获取的细节程度，减少 SDK 对帧率的影响；

勾选，帧率可能会有明显下降。

● Lua 堆栈：

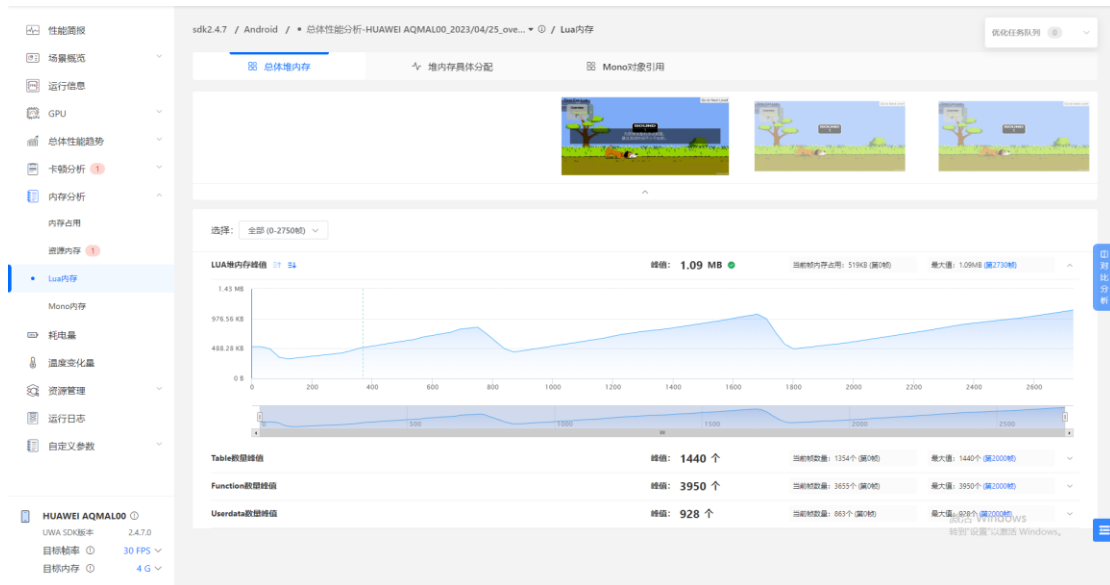
勾选时，开启 Lua 的函数耗时统计。

在报告中，Lua 的 CPU 耗时会出现在逻辑代码的调用堆栈中。

● Lua 内存：

勾选时，开启 Lua 内存数据采集，可设置 Lua 内存 Dump 的间隔（详见“Lua 模式”的说明）。

在报告中，Lua 的内存分配信息会出现在下图的“Lua 内存”面板中。



● Resources:

不勾选时，在测试开始、测试结束、发生场景切换，以及每 1000 帧采集当前帧的资源汇总信息，包括纹理资源、网格资源、动画资源、音频资源、材质资源等 11 种资源的总大小、总数量。

勾选时，在测试 Overview 模式的同时开启 Resources 模式的数据采集。

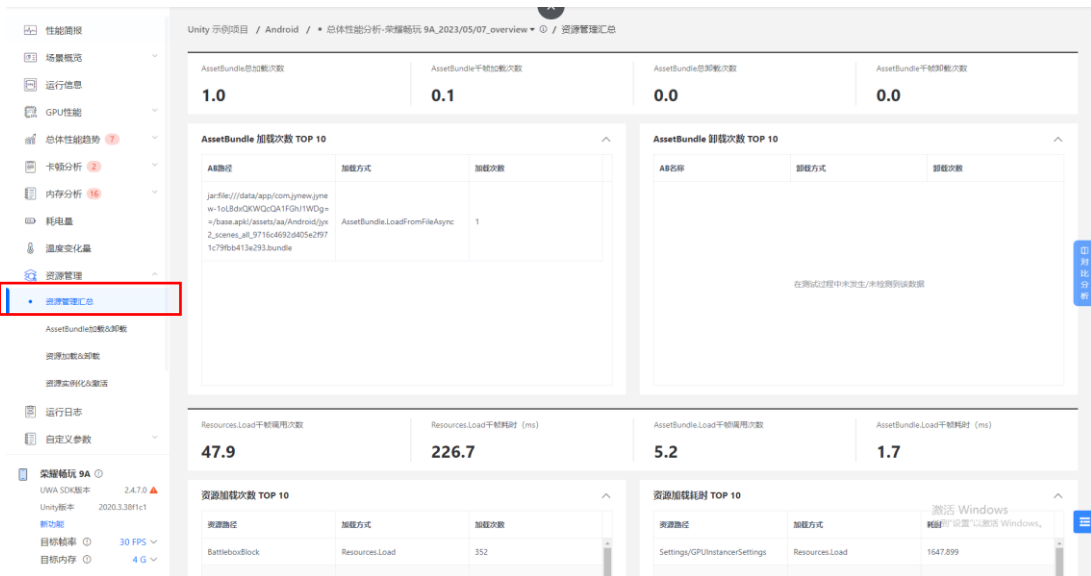
可在报告“资源内存”中查看相关数据：



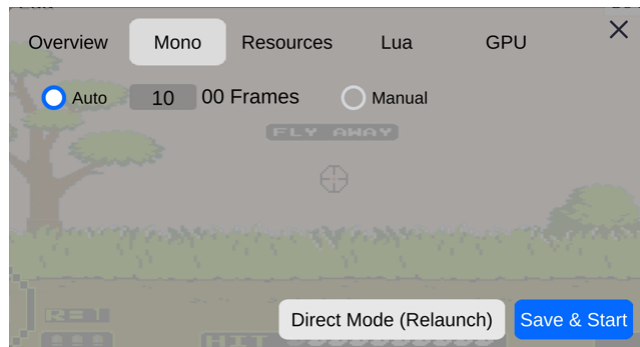
● 资源管理:

勾选时开启“资源管理”功能（详见“Resources 模式”的说明）。

可在报告“资源管理”中查看相关数据：



(2) Mono 模式

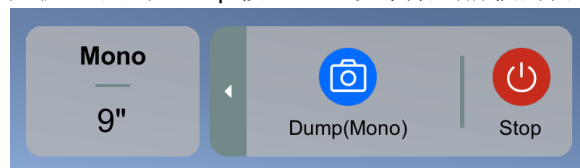


配置说明：

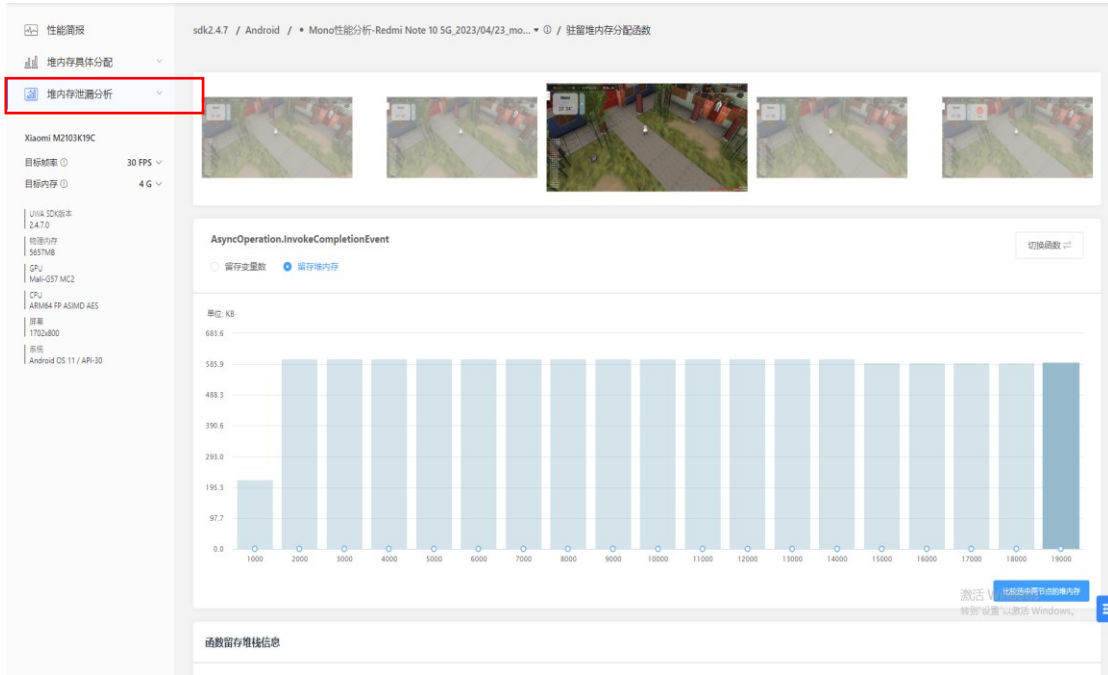
- Dump 间隔设置

勾选 **Auto**，则会按照指定的间隔帧数（可输入范围 1-20）自动采集；

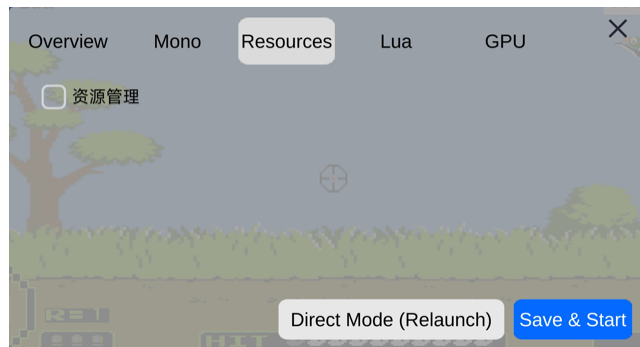
勾选 **Manual**，开始测试后，点击 **Dump** 按钮，可以采集当前帧的留存堆内存信息。



可在报告“堆内存泄漏分析”一栏中查看相关数据：



(3) Resources 模式



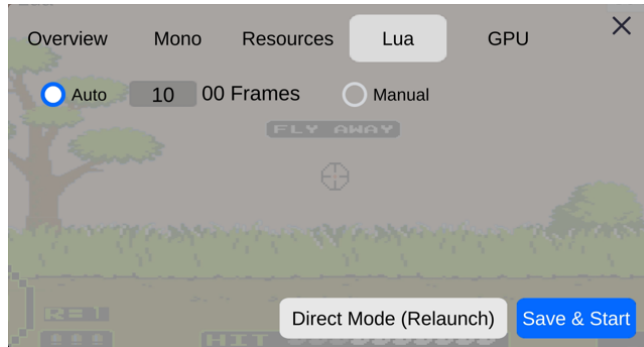
配置说明:

- 资源管理 (仅支持 Android): 勾选后, 将开启资源管理功能



注: 在部分设备上, 开启资源管理功能会导致部分项目运行不稳定的问题。

(4) Lua 模式

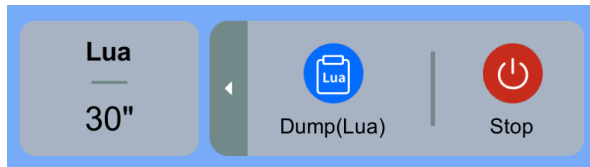


配置说明：

- Dump 间隔设置

勾选 **Auto**，则会按照指定的间隔帧数（可输入范围 1-20）自动采集；

勾选 **Manual**，开始测试后，点击 **Dump** 按钮，可以采集当前帧的 Lua Table 数量、Function 数量、Userdata 数量、Mono 对象引用数量等信息。



可在报告“总体堆内存”和“Mono 对象引用”栏目中查看相关数据：



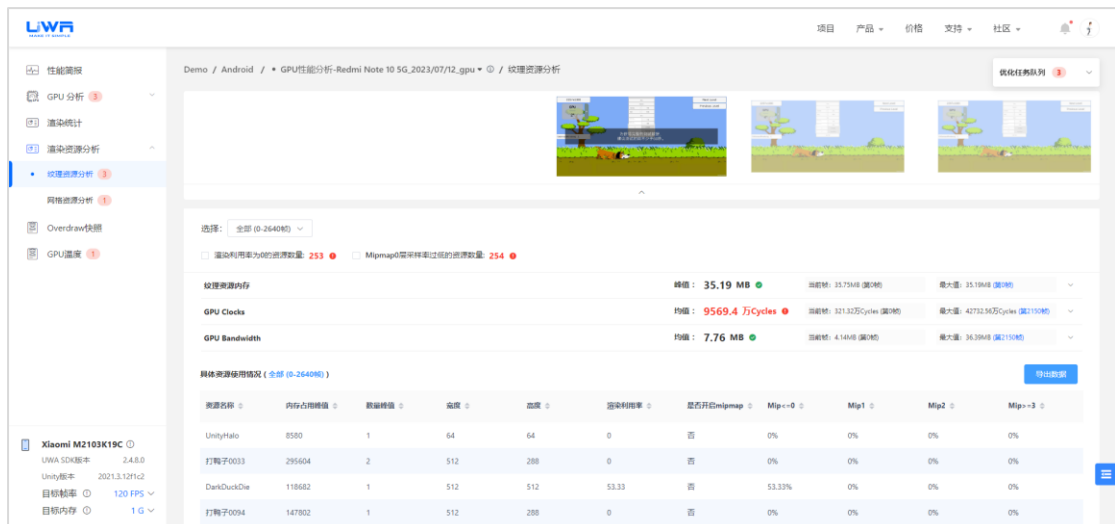


(5) GPU 模式

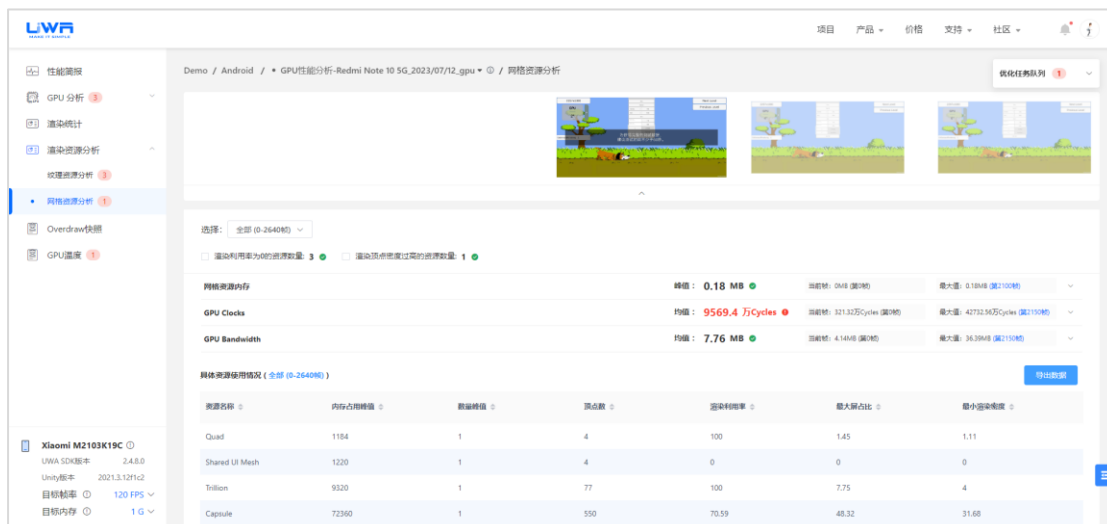


配置说明：

- 纹理资源分析（仅支持 Android）：勾选后，将开启纹理资源分析功能



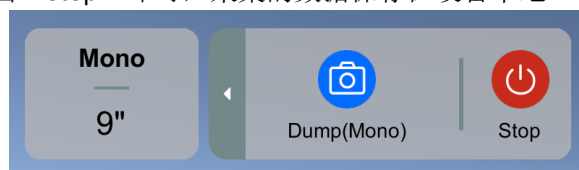
- 网格资源分析（仅支持 Android）：勾选后，将开启网格资源分析功能



2. Direct Mode (Relaunch): 点击 Direct Mode 按钮，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. 开启测试后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。



2.3. 数据支持情况

受 Unity 版本及发布设置影响的数据统计如下：

Android 平台

	版本与设置要求
Mono 模式 堆内存累积分配 堆内存泄漏分析	Mono 包无限制； IL2CPP 包需 Unity 2019.4 及以上版本，且勾选 Script Debugging
Mono 模式 堆内存对象快照	Mono 包，不支持； IL2CPP 包，需 Unity 2019.4 及以上版本，且勾

	选 Script Debugging
Overview 模式 Vulkan DrawCall/Triangle 统计	需 Unity 2019.4 及以上版本，且使用 Vulkan 版 UWA SDK
Overview 模式 Timeline 自定义函数参数	需 Unity 2021.x 及以上版本
GPU 模式 渲染资源分析	需 Unity 2018.2 及以上版本

Windows 平台

	版本与设置要求
Mono 模式 堆内存累积分配 堆内存泄漏分析	Mono 包，暂时仅支持 Unity 2018–2020 版本 IL2CPP 包，需 Unity 2019.4 及以上版本，且勾选 Script Debugging
Mono 模式 堆内存对象快照	Mono 包，不支持； IL2CPP 包，需 Unity 2019.4 及以上版本，且勾选 Script Debugging
Overview 模式 C#函数调用统计	暂时仅支持 Unity 2018–2020 版本
Overview 模式 Timeline 自定义函数参数	需 Unity 2021.x 及以上版本

iOS 平台

	版本与设置要求
Overview 模式 C#函数调用统计	暂时仅支持 Unity 2018–2020 版本
Overview 模式 Timeline 自定义函数参数	需 Unity 2021.x 及以上版本

HarmonyOS NEXT 平台

	版本与设置要求
Mono 模式 堆内存累积分配 堆内存泄漏分析	Mono 包，无限制； IL2CPP 包，需勾选 Script Debugging
Mono 模式 堆内存对象快照	Mono 包，不支持； IL2CPP 包，需勾选 Script Debugging

注：当“Mono 模式-堆内存累积分配”不被支持时，Mono 模式将被禁用（UI 面板上不可见）。

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. UWA Tools 上传 (IOS 与 HarmonyOS NEXT 除外)

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。

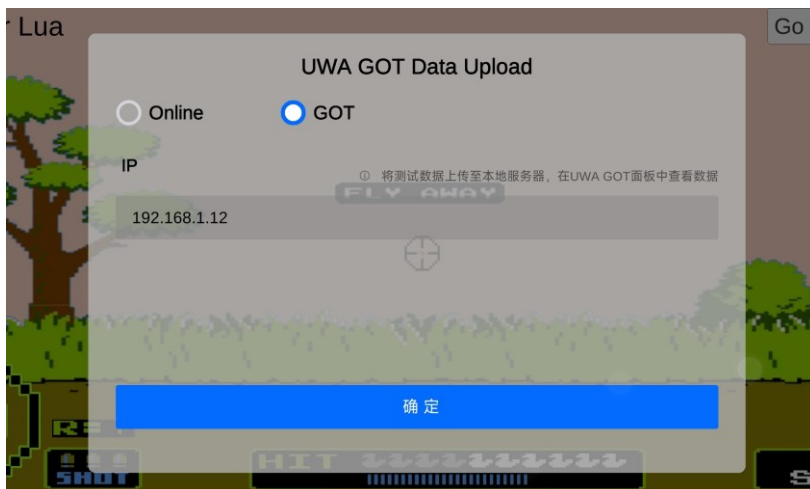


3. 点击“提交数据”后,即可进入下图中的上传界面。当上传界面中的进度条结束,提示返回,即上传成功。



3.3. 游戏内直接上传

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，“确定”按钮亮起时，即上传成功。



3.4. 注意事项

1. GOT 中的 Lua 和 GPU 模式仅限 GOT Online 服务可用。
2. GOT 中的 Overview 模式极简版仅限 GOT Online 服务可用。
3. 如果在测试中发生闪退，或者不正常的终止，此时可以重启游戏，在 SDK 启动后会自动检测到尚未上传的数据，并提示上传，如下图。
如果重启游戏后未出现图中提示，可以尝试使用 UWA Tools 查看最新的测试数据并上传。

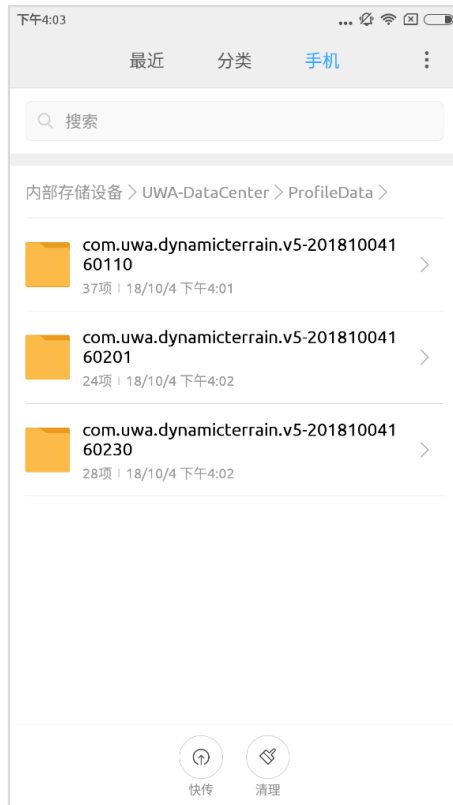


4. 如果 UWA Tools App 无法与本地服务器连接，或其他原因导致数据无法传输，可通过以下路径找到测试数据：

Android 设备：/sdcard/UWA-DataCenter/ProfileData

Android 10 及以上设备默认存储路径：

/sdcard/Android/data/com.xxxx.xxxx（游戏包名）/files/UWA-DataCenter/ProfileData



Windows 设备: C:/UWA-DataCenter/ProfileData

Name	Date modified	Type	Size
WindowsPlayer-20180923151049	10/4/2018 3:59 PM	File folder	
WindowsPlayer-20180923151046	9/30/2018 11:14 PM	File folder	
WindowsPlayer-20180923151043	9/30/2018 7:26 PM	File folder	

iOS 设备: Documents/UWA-DataCenter/ProfileData

HarmonyOS NEXT 设备: /data/app/userid/base/游戏包名/haps/entry/files/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。

> GOT-Sample > TestData

Search TestData

Name	Date modified	Type
Assets	8/30/2018 6:05 PM	File folder
Mono	8/30/2018 6:05 PM	File folder
Overview	10/1/2018 12:38 AM	File folder

4. 查看报告（本地编辑器内查看）

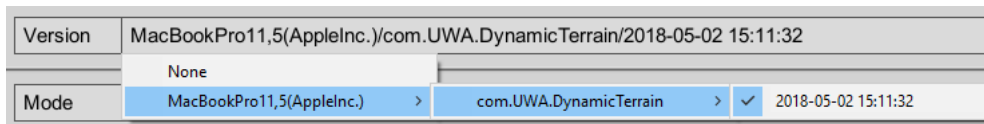
在“GOT Panel”界面中选择测试模式，即可查看已上传至本地服务器的测试数据报告，共三种测试模式：Overview、Mono 和 Resources。若查看 Lua 模式的数据报告需要同时满足项目使用了 Lua 且已购买 GOT Online 测试时长；若查看 GPU 模式的数据报告也需要通过 GOT Online 实现。



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：

a. CPU 开销走势图

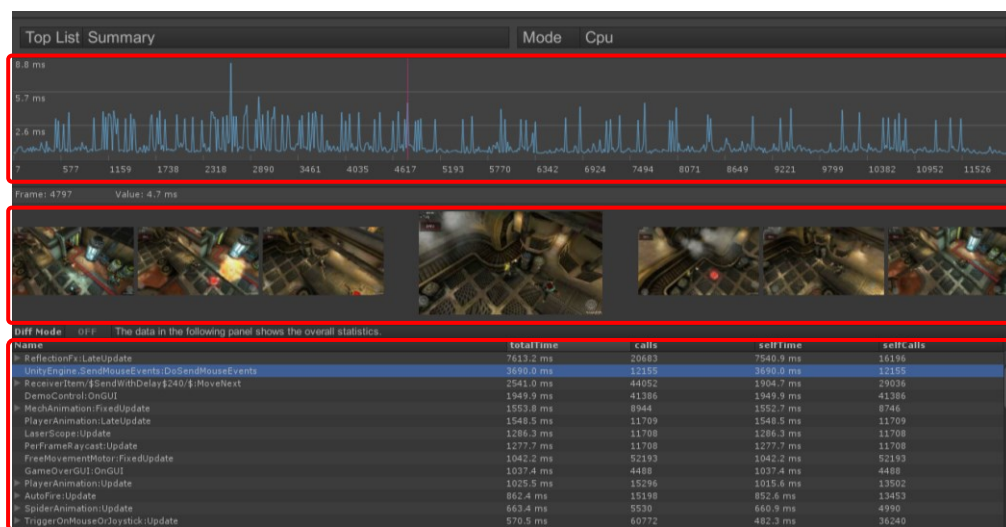
您可以选择任何一个函数，查看它在项目运行时的 CPU 开销。

b. 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。

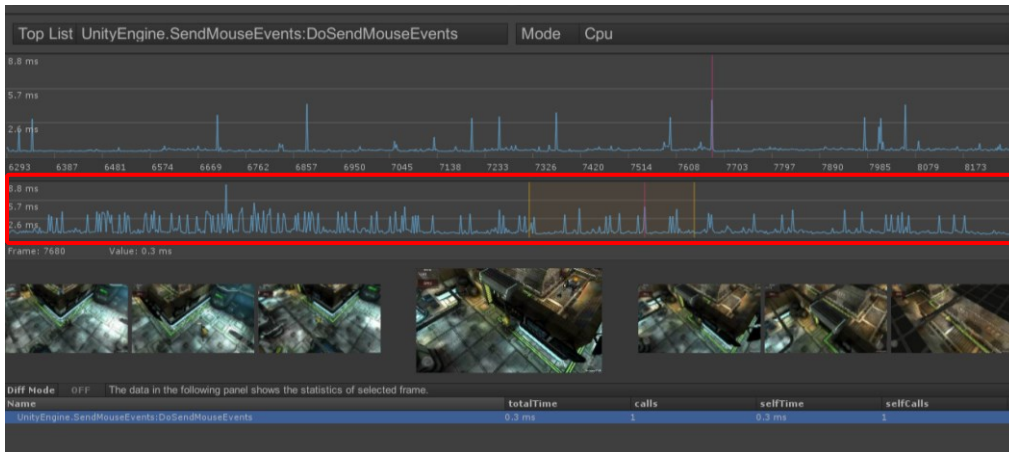
c. CPU 耗时分析视图

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

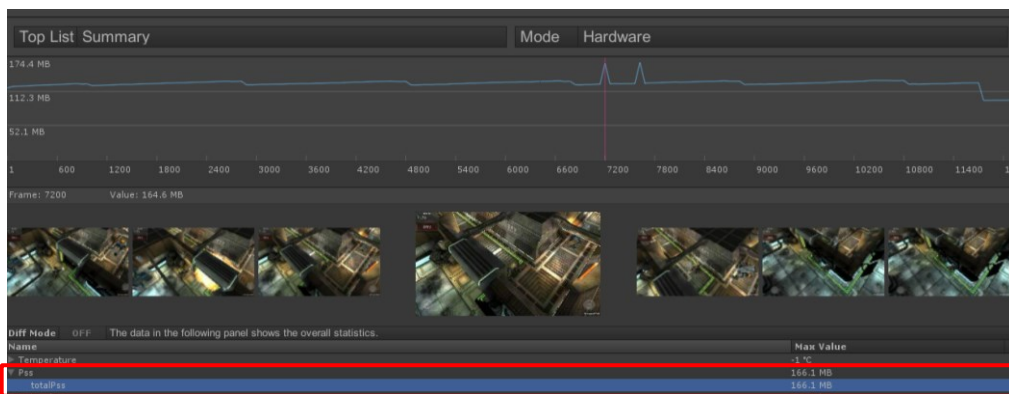
选择查看具体的逻辑代码。同时，您可以通过调整关注区域视图中的滑块，来重点查看您关注区域的 CPU 开销。



2. 硬件设备信息

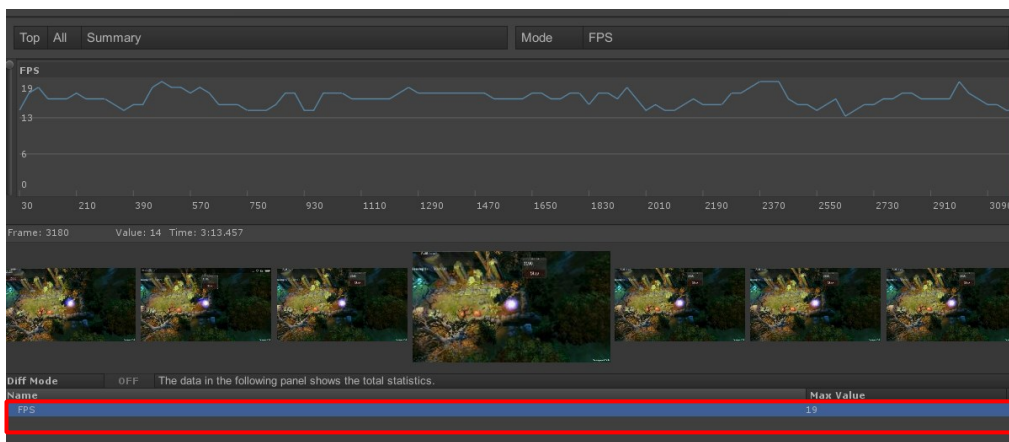
在“Mode”中选择“Hardware”，即可查看项目运行时的硬件设备运行信息，主要包括：硬件设备的内存信息。

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



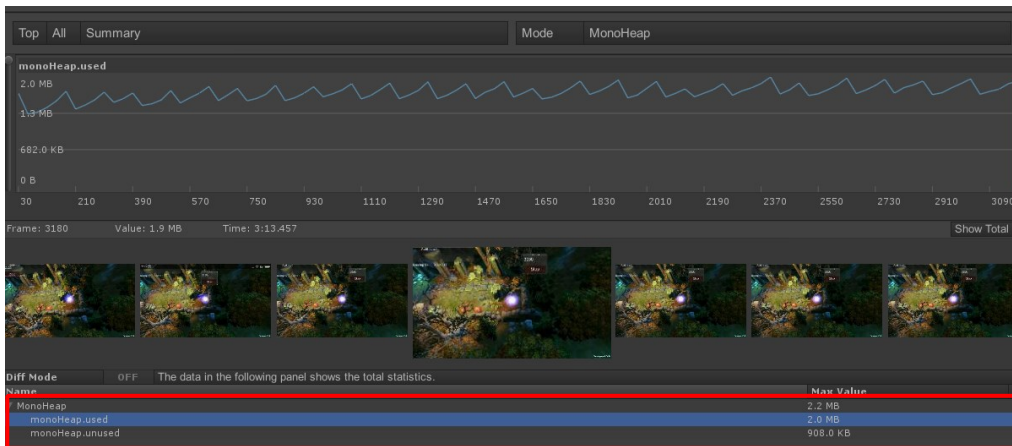
3. FPS 信息

在“Mode”中选择“FPS”，即可查看项目运行时的 FPS 统计。



4. Mono Heap 信息

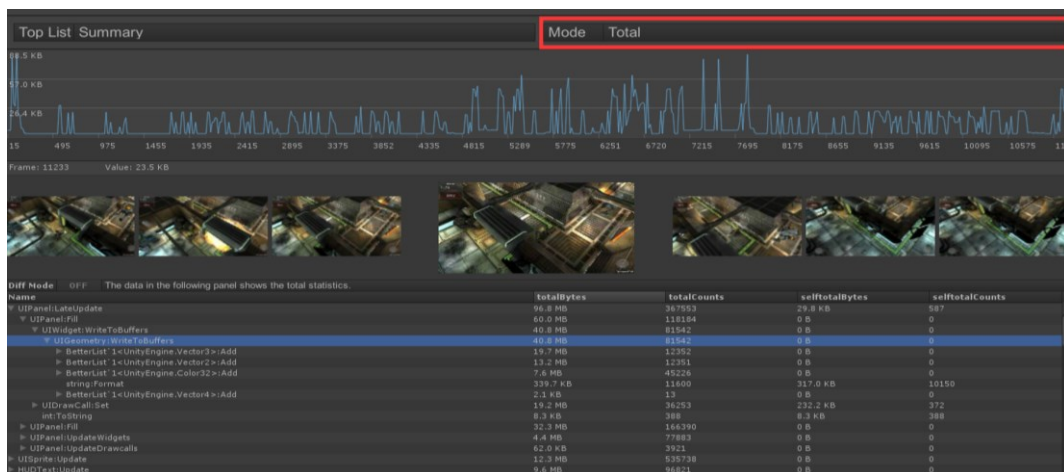
在“Mode”中选择“MonoHeap”，即可查看项目运行时的 Mono 堆内存总量统计，包括了使用中的和未使用中的部分。



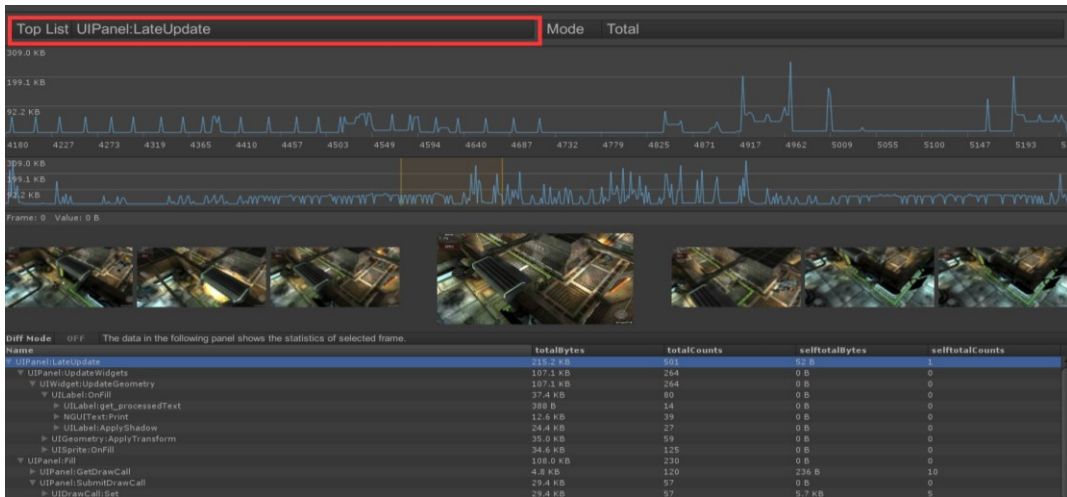
4.2. Mono（Mono 堆内存分析）

1. 代码堆内存累积分配

- 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；

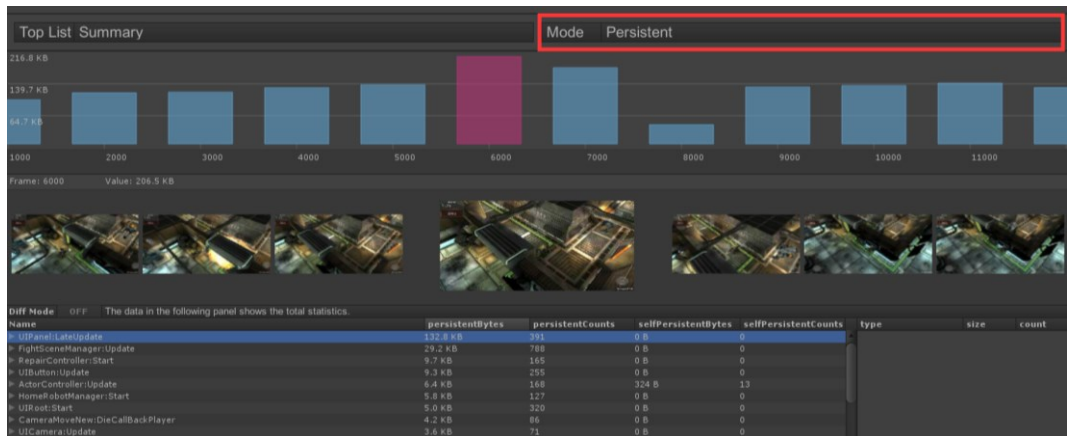


- 在“Top List”中选择具体的函数名称，您就可以看到相应函数的具体堆内存分配情况，并且通过与图表进行交互来查看任何一帧的具体堆内存分配。

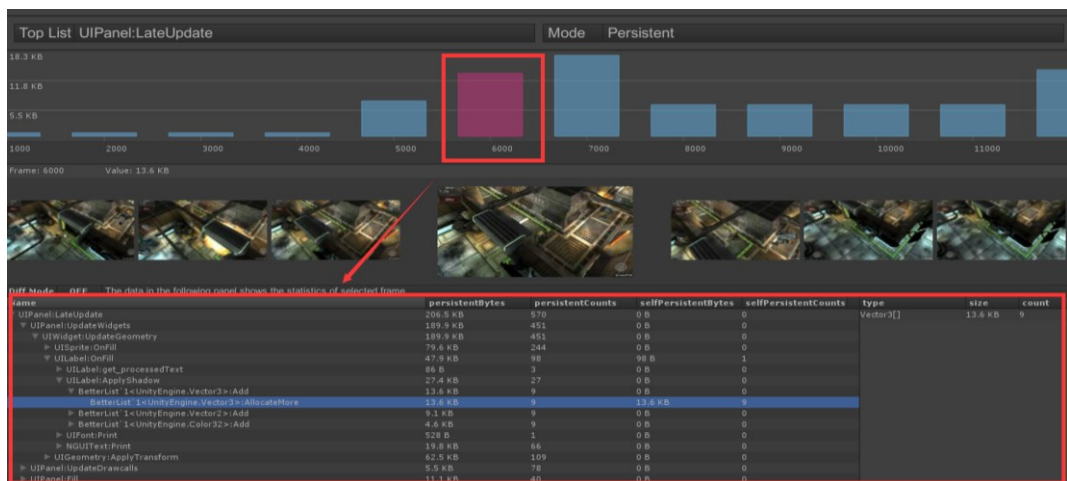


2. 代码堆内存泄露分析

(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。

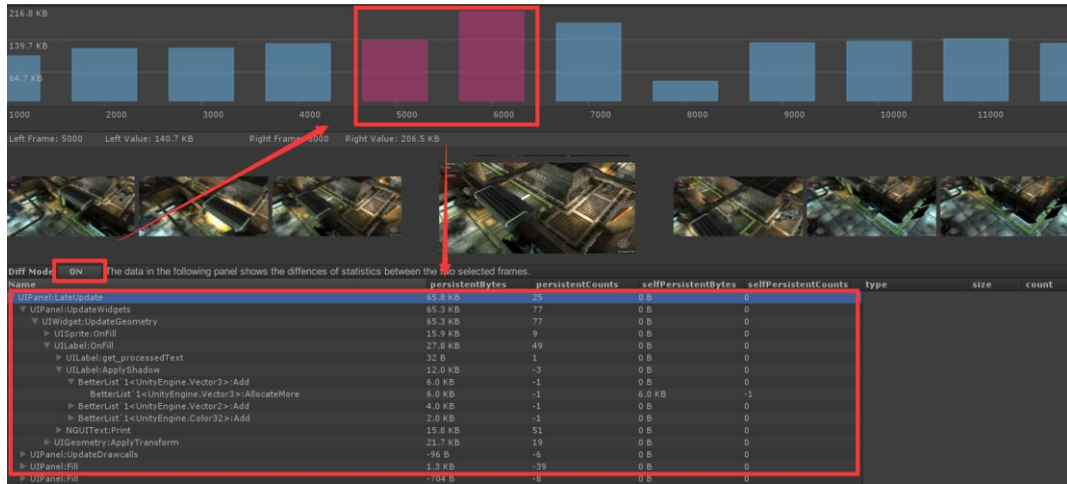


(2) 在“Top List”中选择具体的函数名称，您就可以看到相应函数的具体堆内存分配情况，并且通过与图表进行交互来查看详细堆内存驻留情况。同时，当 selfPersistentCounts 不为 0 时，点击可以查看由该函数生成的、驻留在堆内存中的变量类型。



(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，您则可以快速比较两次堆内存占用的差异。

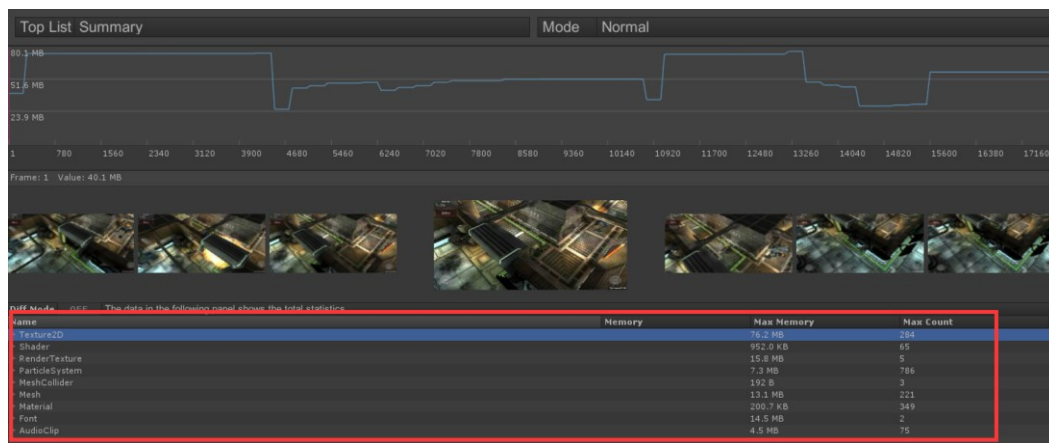


4.3. Resources（运行时资源）

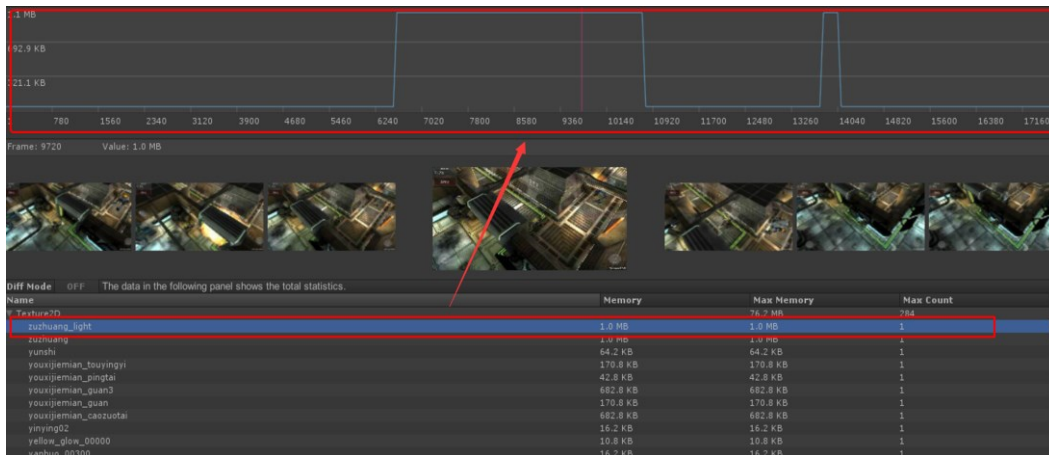
点击“Resources”按钮，即可查看项目运行时资源的具体使用情况。它主要包括以下功能：

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。

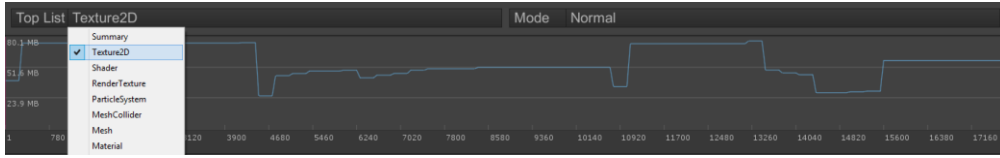


(2) 可以查看具体资源在项目运行时的使用情况。

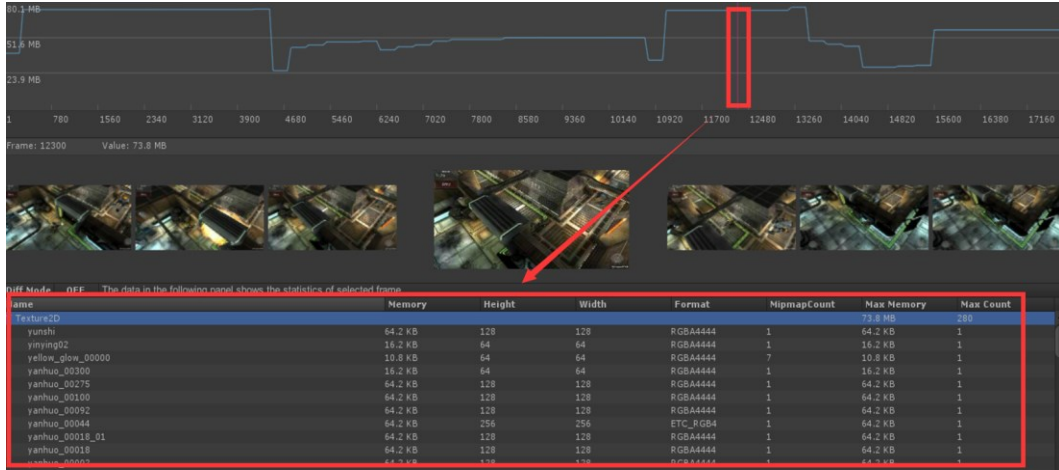


2. 查看每帧中资源的具体使用情况

(1) 在 TopList 中选择您想查看的资源类型。



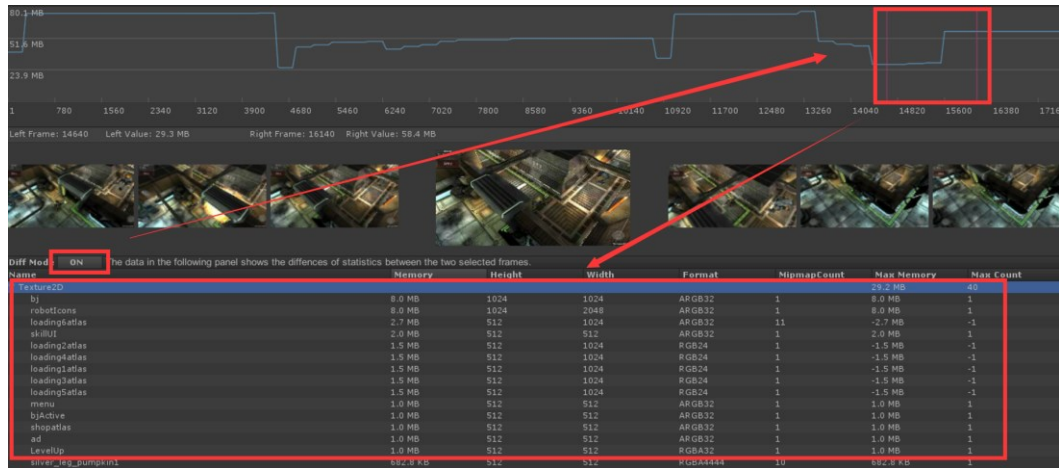
(2) 点击资源使用走势图，即可查看每帧该类资源或某个特定资源的具体使用情况。



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。

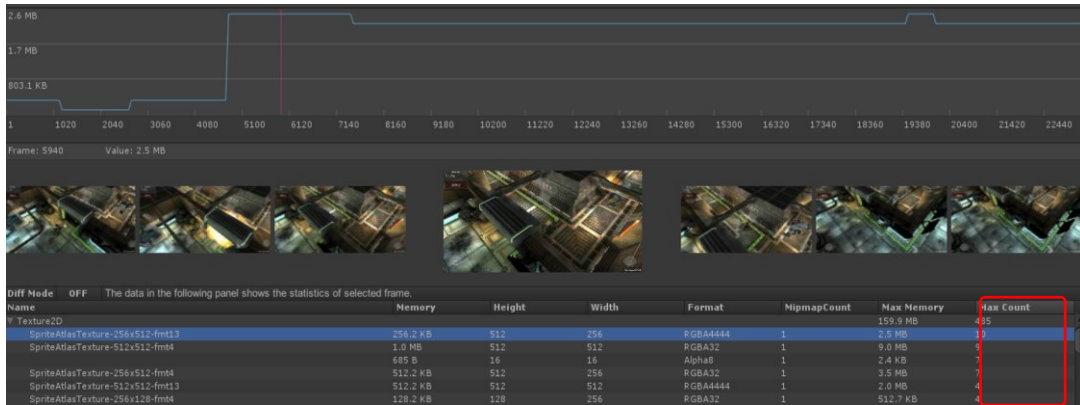


(2) 上图为第 16140 帧与第 14640 帧的 Texture 比较情况。其中，“Max Memory”中为正值资源表示为第 16140 帧中的新增资源，而负值的资源则为第 16140 帧中的减少资源。通过这种比较，即可帮您快速定位具体的资源变化量和解决资源泄露等问题。

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。

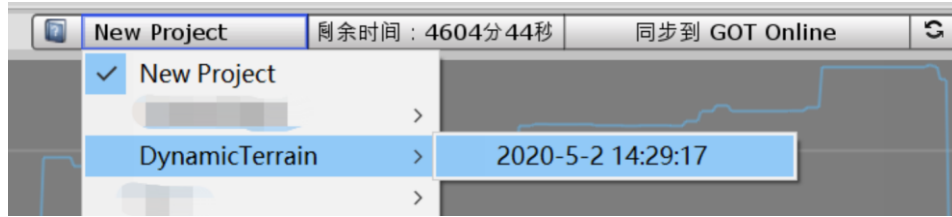
4.4. 通过 GOT 使用 GOT Online 的方式

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_User_Guide_For_Unity》（UWA SDK For Unity 使用说明）。

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

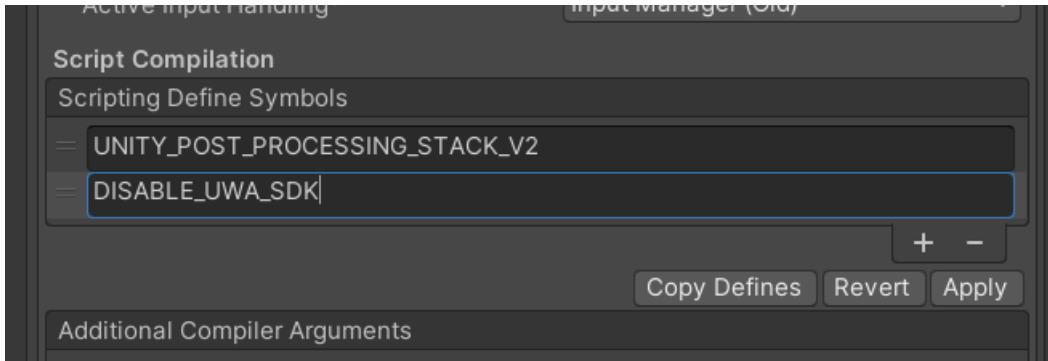


2. 选择任意一份在编辑器中已存在且打开的测试数据报告，先在左侧项目列表的下拉框中选择需要同步到网页的项目名称，再点击报告页的菜单栏“同步到 GOT Online”按钮。



附录 1: UWA API 的介绍和用法

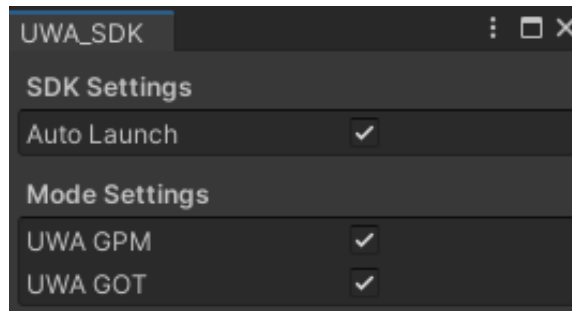
DISABLE_UWA_SDK



当编辑器设置中定义了 `DISABLE_UWA_SDK`，在打包编译 apk 时，会把所有 UWA SDK 相关的代码屏蔽，最终 apk 中不存在 UWA 相关的 dll 或者 so。可在发布正式包时使用。

UWAEngine.StaticInit

```
public static void StaticInit();
```



当 UWA_SDK 面板设置不勾选“Auto Launch”选项的时候，需要通过该函数在测试开始前初始化插件。

注：如果 SDK 是通过 `StaticInit` 进行初始化的，那么在使用 `Direct Mode` 时，重启游戏后也只会 `StaticInit` 被调用之后才会自动开始采集数据。

UWAEngine.SetUIActive

```
public static void SetUIActive (bool active);
```

该函数可用于：隐藏或显示测试过程中 SDK 本身的 GUI 界面。

UWAEngine.Note

```
public static void Note (string note);
```

该函数可用于：为本次测试的性能报告添加备注。

UWAEngine.SetConfig

```
public static void SetConfig(string config);
```

该函数可用于：通过脚本自定义所有模式的测试配置，而不需要手动设置。

具体配置内容以及使用方式，请查阅函数注释和说明。

UWAEngine.Tag

```
public static void Tag(string tag);
```

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.Dump

```
public static void Dump (eDumpType type);
```

仅支持 Development 环境，该函数可用于触发各种类型 Dump 操作：

1. Overview 模式支持 Lua 和 Resources 类型的 Dump 操作；
 2. Mono 模式支持 ManagedHeap 类型的 Dump 操作；
 3. Lua 模式支持 Lua 类型的 Dump 操作；
 4. GPU 模型支持 Overdraw 类型的 Dump 操作。
-

UWAEngine.PushSample/PopSample

```
public static void PushSample(string sampleName);
```

```
public static void PopSample();
```

仅支持 Development 环境，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，同时也支持子线程使用。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);

```

仅支持 Development 环境，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);

```

仅支持 Development 环境，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)
```

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，其中 bool 参数为上传结果；

user/pwd 为登录 UWA 网站的用户名和密码；

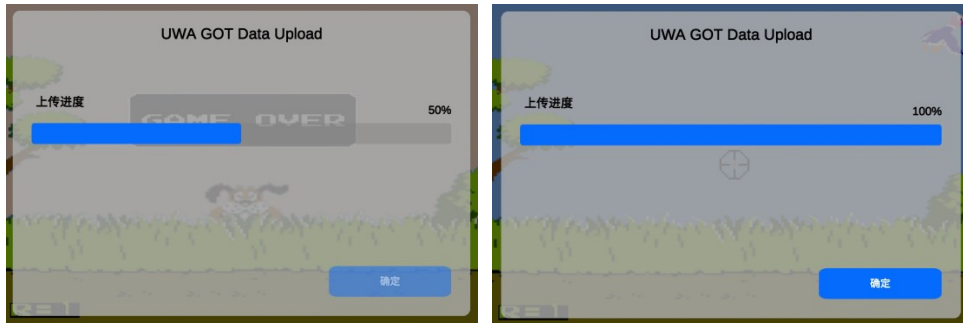
projectId/projectName 为上传项目的 ID 或名称；

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：

```
https://www.uwa4d.com/u/got/perfanalysis.html/overview?project=16929&type=4
```

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:

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/Resources/Mono/Lua/GPU）与该 ID 对应的的项目是一致的，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)`

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()`

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。
