

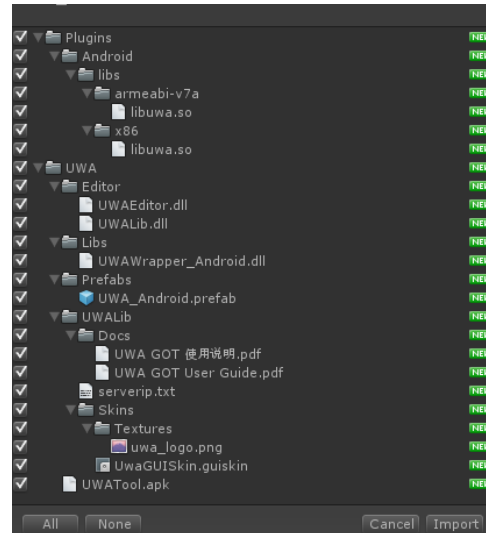
# UWA GOT 使用说明

## Contents

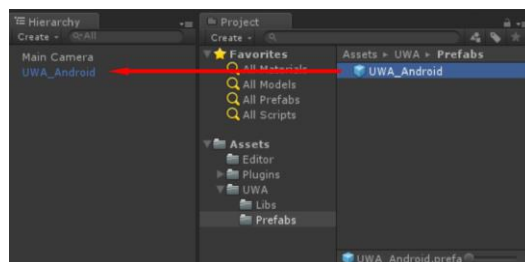
1. GOT SDK 集成 .....	2
2. 性能测试与数据上传 .....	4
3. GOT Editor 使用说明 .....	8
3.1. Performance Overview .....	8
3.2. Mono Memory Analysis .....	11
3.3. Runtime Asset Checker .....	13
4. GOT Online 使用说明 .....	16
4.1. 付费方式 .....	16
4.2. 通过 GOT App 上传性能数据 .....	16
4.3. 通过 GOT Editor 上传性能数据 .....	19
4.4. 在线查看 UWA GOT 性能报告 .....	21
附录 1: UWA API 的介绍和用法 .....	24

# 1. GOT SDK 集成

- 1、 将 ZIP 文件中的 unitypackage 文件拖入到待测试的 Unity 项目中，当出现下图窗口时，点击“Import”按钮导入相关文件。



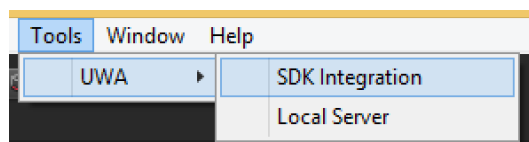
- 2、 在 Editor 中将 UWA/Prefabs 文件夹下的 Prefab 文件拖入到项目的首场景中，如下图所示。



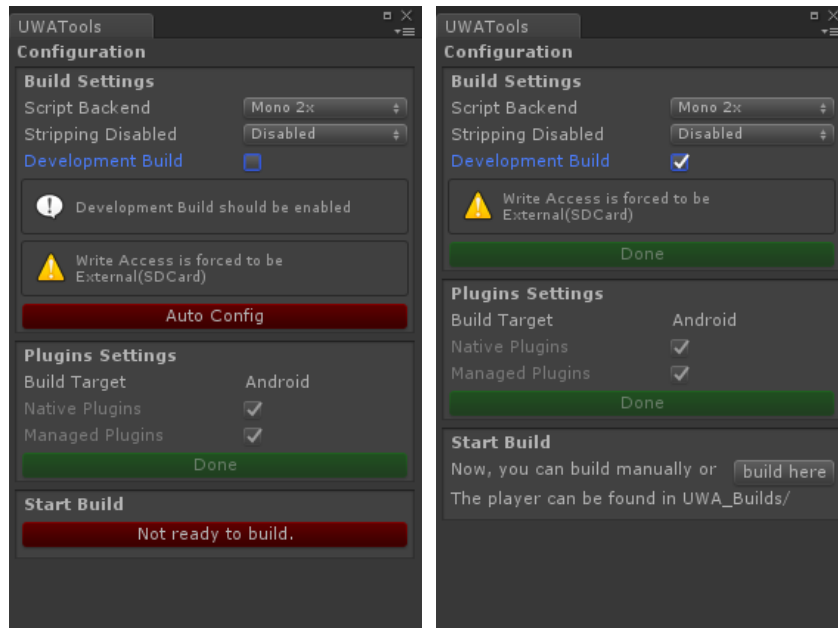
- 3、 如在 Game 视图的右上角出现如下图所示的 UI 界面，且无报错信息，则说明 UWA 插件已经集成到项目中。



- 4、 点击编辑器菜单栏中的“Tools->UWA->SDK Integration”选项，打开 UWA 工具栏。



- 5、 在出现的 UWA 工具设置窗口中，请按指引确保界面中的两个按钮都变绿。



- 6、发布游戏版本。建议用户直接点击 UWA 界面上的“build here”按钮来一键发布游戏，发布后存放于 UWA\_Builds/Android 文件夹中。同时，用户也可通过“Build Settings->Build”或 BuildPlayer 接口来手动发布游戏版本，但须确保添加 BuildOptions.Development 参数。

**注意事项：**

- 1、Android 设备无需 Root 权限。
- 2、UWA 工具设置界面按钮为绿色状态。
- 3、请确保发布的是 Development Build。
- 4、目前不支持采用 il2cpp 进行发布的项目。
- 5、截屏的记录只支持 Android 5.0 或以上的系统。

## 2. 性能测试与数据上传

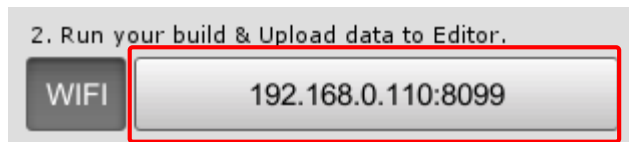
- 1、在 UWA SDK 集成并发布后，您即可将其安装在 **Android** 设备上并进行测试。
- 2、打开 App 后，在界面右上角选择您想进行的测试类型。点击后，测试就会立即开始。  
**Direct Mode:** 当点击 **Direct Mode** 并使左侧按钮变绿后再点击某个模式，App 会自动退出，而在下一次开启时则立即开启对应模式的测试。



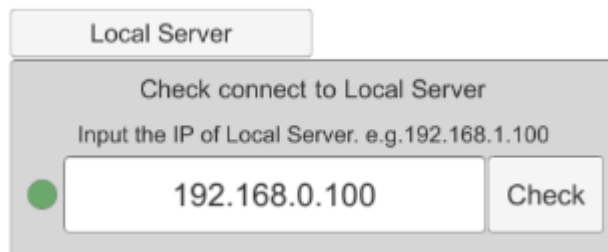
- 3、**Stop** 面板可以进行拖动，如果想结束本次测试，点击“**Stop**”即可。UWA SDK 会将搜集到的性能数据保存在设备本地。



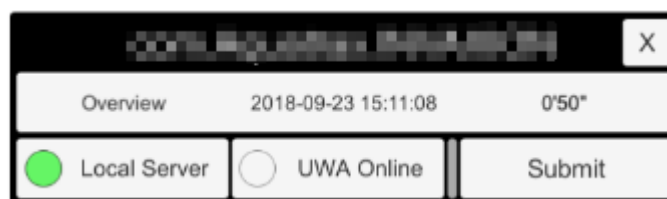
- 4、测试完成后，在测试机上安装 ZIP 中的 UWATool.apk 文件并打开，该 App 用于数据的上传。登录账号后，可选择两种方式查看数据：
  - a) **GOT Editor 本地离线查看(免费使用):** 点击编辑器菜单栏中的“UWA->Local Server”选项，点击面板上的“**WIFI**”按钮开启本地服务器。红框处即为您本地机器的当前 IP（端口固定为 8099）。若 IP 无法获取，则可以通过将 IP 写入 UWA/UWALib 下的 serverip.txt 文件中来手动配置。



在 App 中以下界面，输入 Local Server 的 IP 地址，点击“**Check**”按钮，如果左侧圆点变绿，即表示可以连接到服务器。若无法连接成功，请尝试关闭防火墙、VPN 等设置。

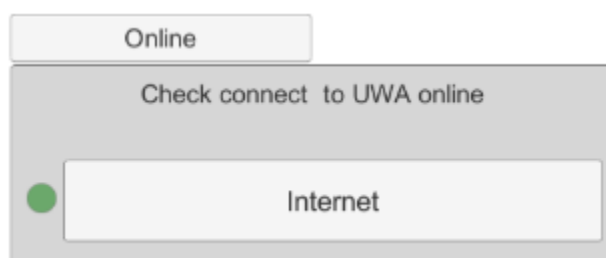


点击“**Upload Data**”，即可查看测试数据，再点击“**Local Server**”使其圆点变绿，最后点击“**Submit**”即可开始上传数据。

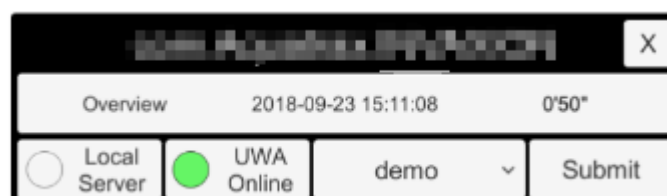


上传完毕后，可在 Editor 中查看对应的数据，具体方式可见本文档中的“GOT Editor 使用说明”部分。

- b) GOT Online 网页版查看（付费使用）：在 App 中以下界面，确认左侧圆点为绿色（即联网状态）。



点击“Upload Data”，即可查看测试数据，再点击“UWA Online”使其圆点变绿后，选择合适的项目进行上传，最后点击“Submit”即可开始上传数据。

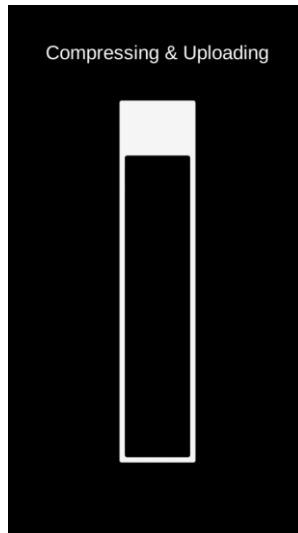


通过该方式上传与查看报告，以及付费方式的具体说明可见本文档中的“GOT Online 使用说明”部分。

- c) 以上两种方式支持同时进行。

对于无用的测试数据，点击右上角“x”删除即可。

- 5、点击“Submit”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，回到测试数据查看界面，即上传成功。



6、当上传完成时，既可在 Local Server 上，点击相关的按钮，选择您想查看的性能报告。或者在 UWA 网站上浏览您的性能报告。

**注意事项：**

1. 目前 Lua 测试的数据只能上传至 GOT Online 进行查看。
2. 如果 GOT App 无法与 Local Server 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 设备上：UWA-DataCenter/ProfileData



Windows 设备上：C:/UWA-DataCenter/ProfileData

(C:) > UWA-DataCenter > ProfileData			
Search 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	

并将其中的文件夹剪切至 GOT Editor（Local Server）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	

最后点击 Local Server 界面中的任意模式按钮，以刷新手动剪切的数据，并开始查看对应的报告。（不支持 Lua 模式的数据）



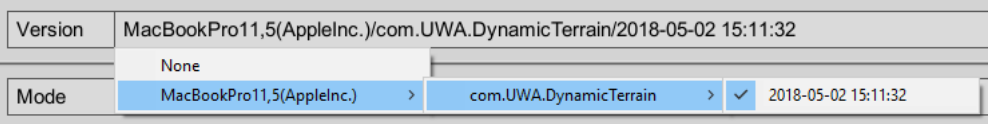
### 3. GOT Editor 使用说明

#### 3.1. Performance Overview

点击“Overview”按钮，即可查看 App 运行时的整体性能情况。

##### 1、逻辑代码的 CPU 开销

(1) 在“Version”中选择您想查看的测试版本。



(2) 选择后，UWA 将为您载入相应数据并进行分析和展示。它主要包括以下视图：

i. **CPU 开销走势图**

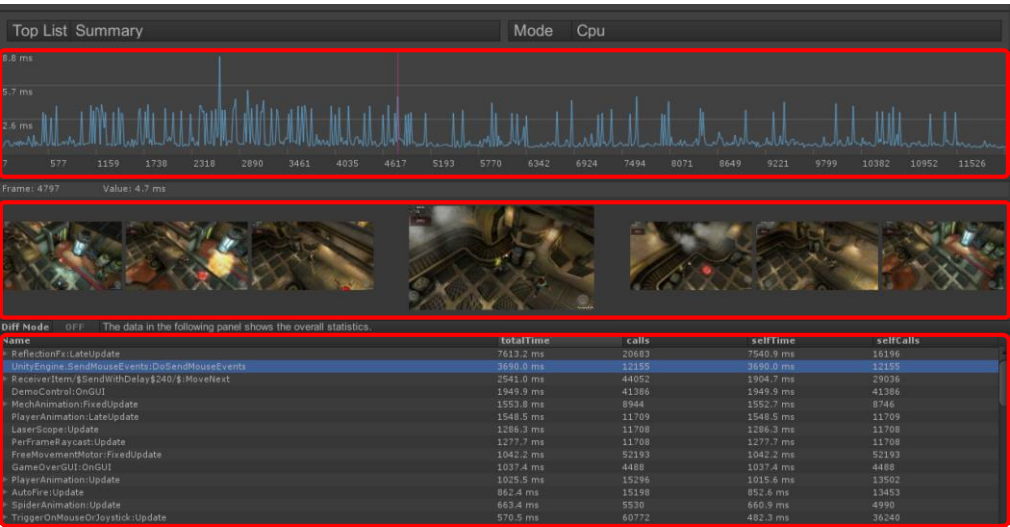
您可以选择任何一个函数，查看它在项目运行时的 CPU 开销。

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。

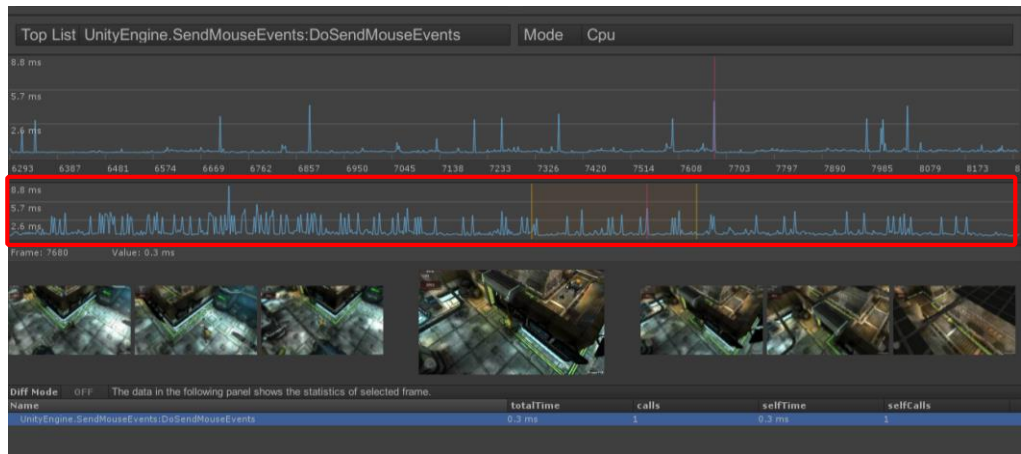
iii. **CPU 耗时分析视图**

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。

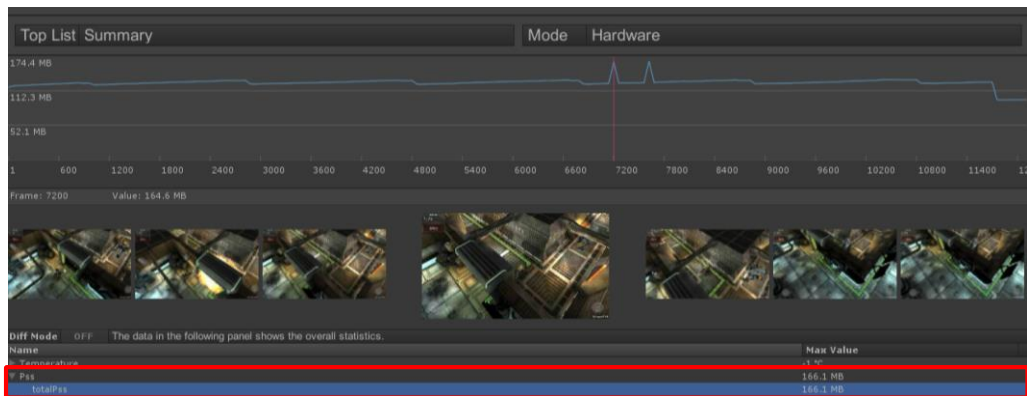




## 2、硬件设备信息

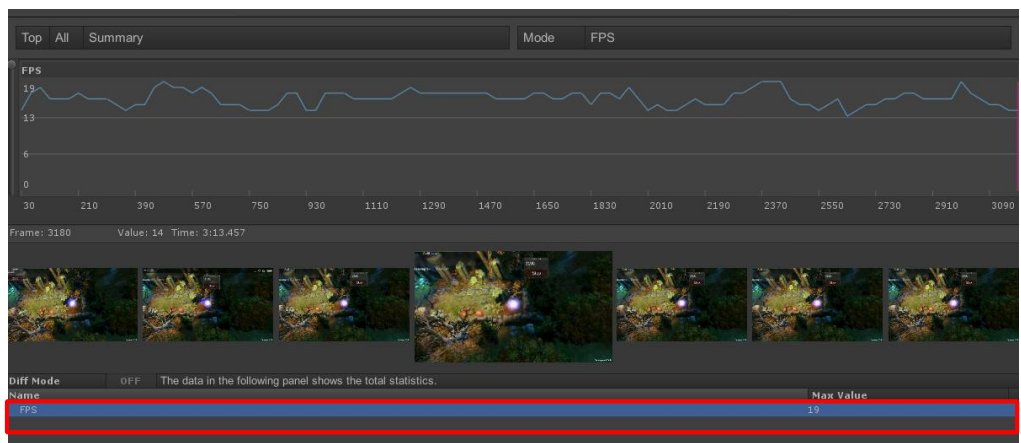
在“Mode”中选择“Hardware”，即可查看 App 运行时的硬件设备运行信息，主要包括：

- (1) 硬件设备的内存信息。在 Windows 上主要显示 WorkingSet 内存在项目运行时的走势。



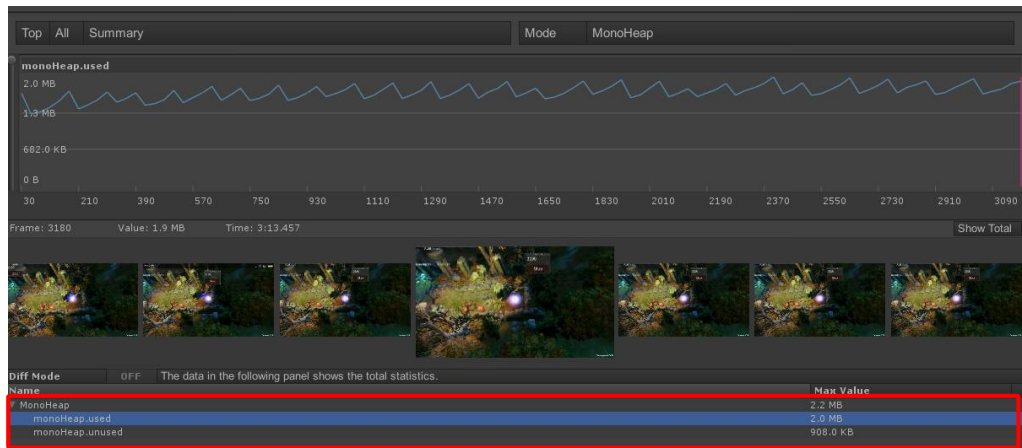
## 3、FPS 信息

在“Mode”中选择“FPS”，即可查看 App 运行时的 FPS 统计。



## 4、Mono Heap 信息

在“Mode”中选择“MonoHeap”，即可查看 App 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。

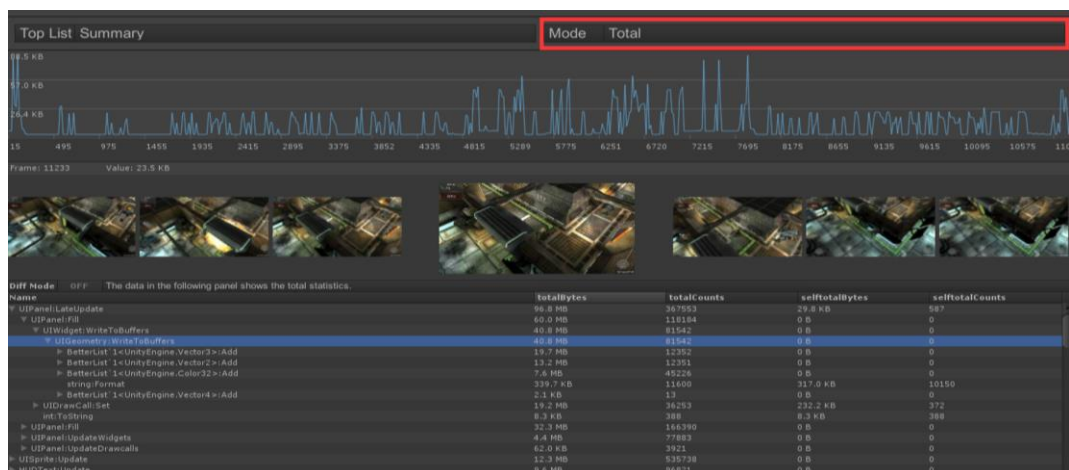


## 3.2. Mono Memory Analysis

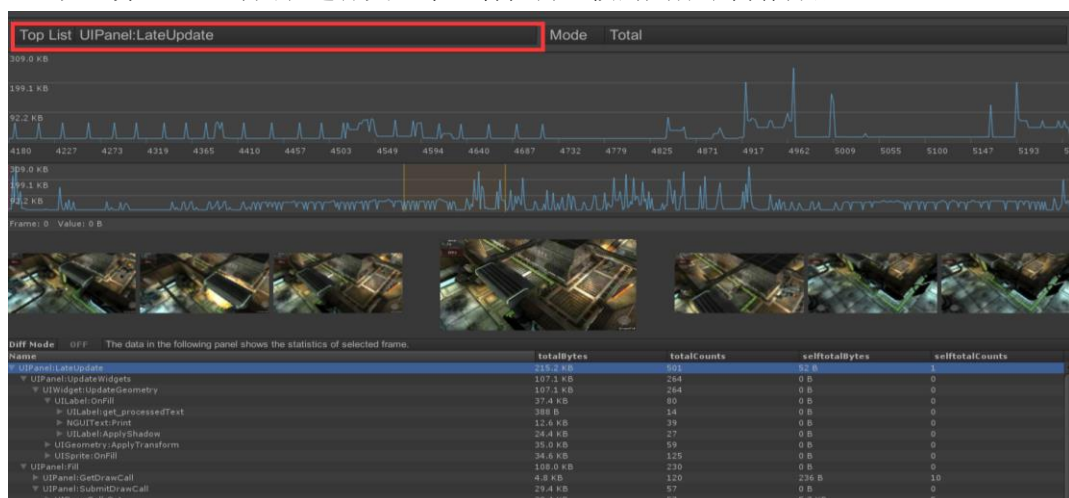
点击“Mono”按钮，即可查看 App 运行时的 Mono 堆内存分配情况。它主要包括以下功能：

### 1、代码堆内存累积分配

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

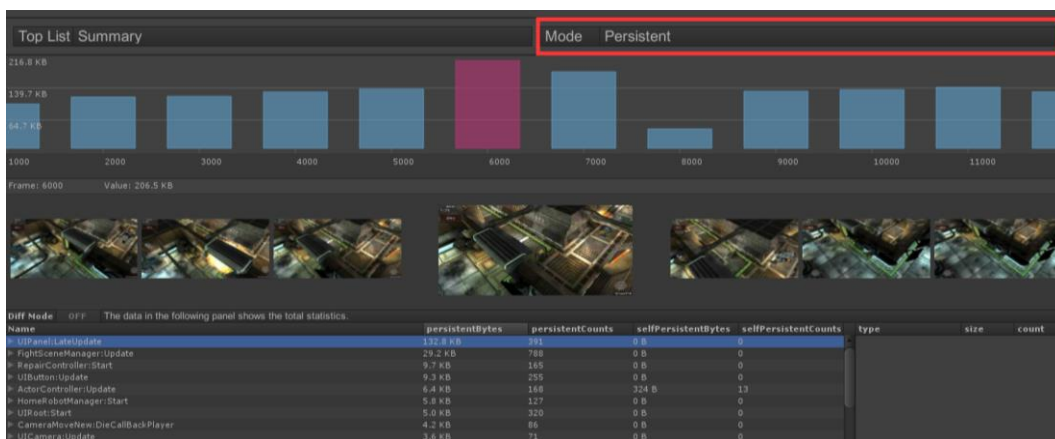


- （2）在“Top List”中选择具体的函数名称，您就可以看到相应函数的具体堆内存分配情况，并且通过与图表进行交互来查看任何一帧的具体堆内存分配。

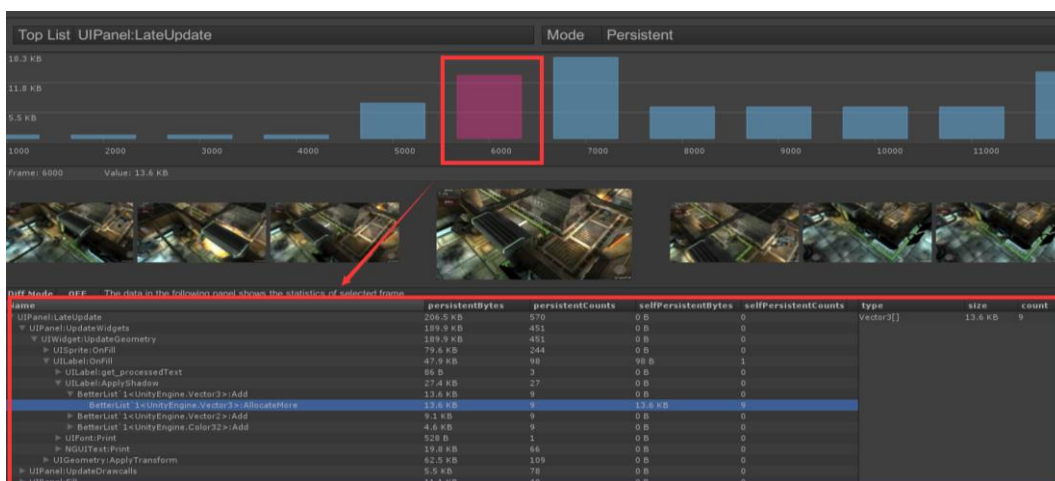


### 2、代码堆内存泄露分析

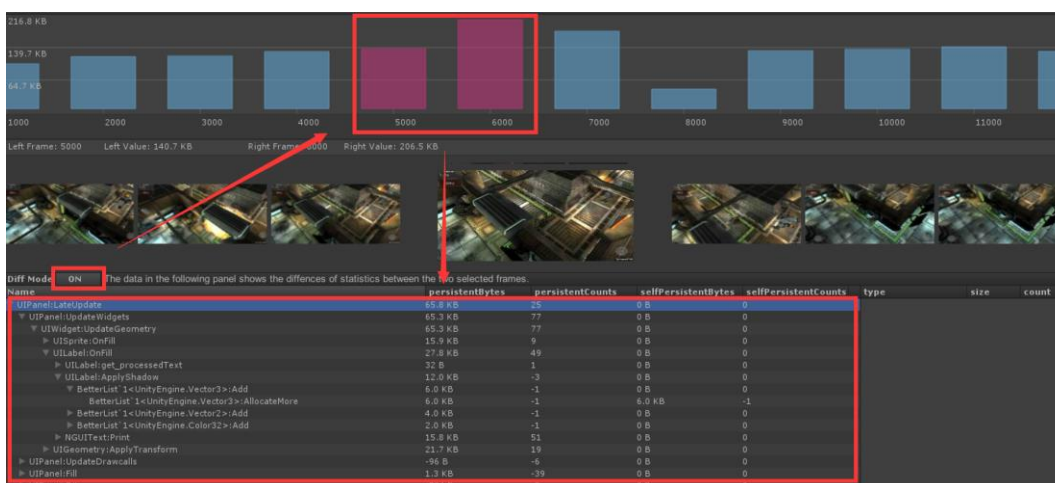
- （1）在“Mode”中选择“Persistent”，您即可查看 App 运行时每个函数在 Mono 中的真实驻留情况。UWA 默认是每 1000 帧分析一次 Mono 堆内存快照，将函数真实的堆内存驻留情况以柱状图的形式进行显示。



- (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。

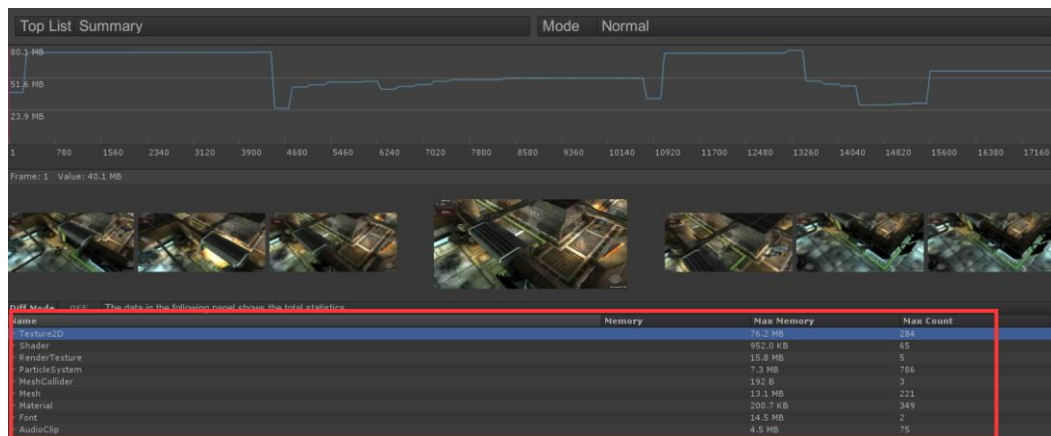


### 3.3. Runtime Asset Checker

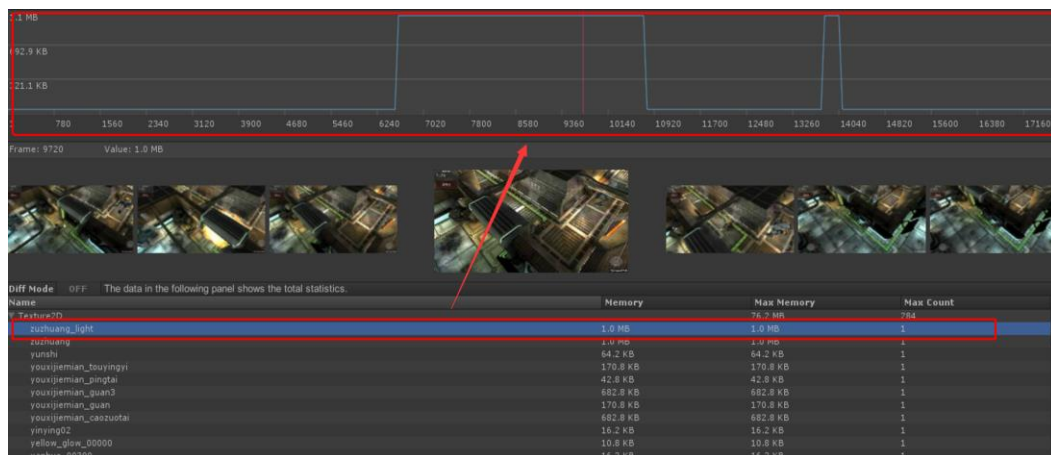
点击“Asset”按钮，即可查看 App 运行时资源的具体使用情况。它主要包括以下功能：

#### 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。



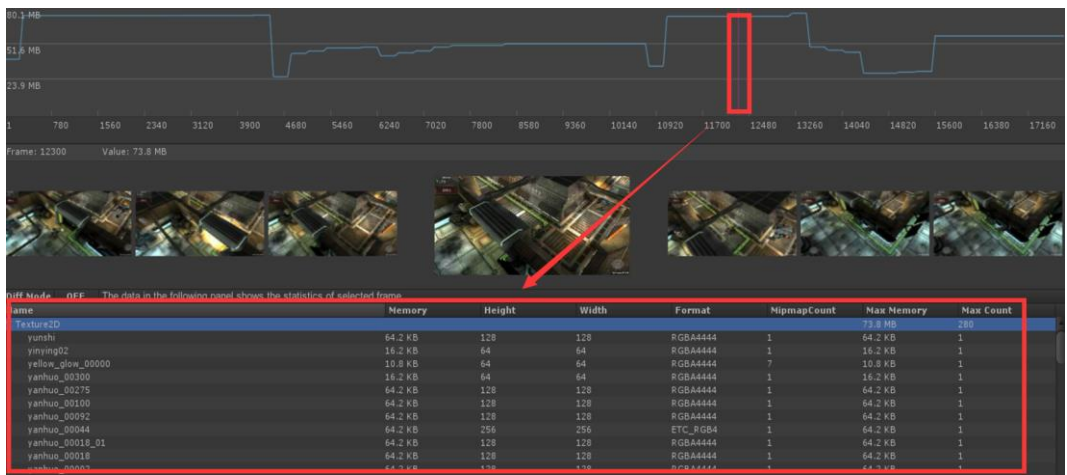
#### 2、查看每帧中资源的具体使用情况

(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。

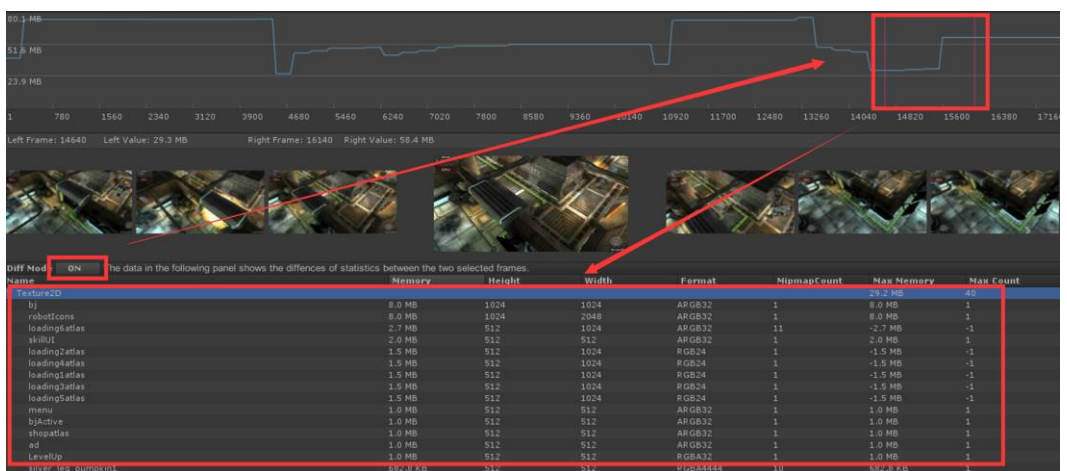




### 3、资源泄露分析

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。

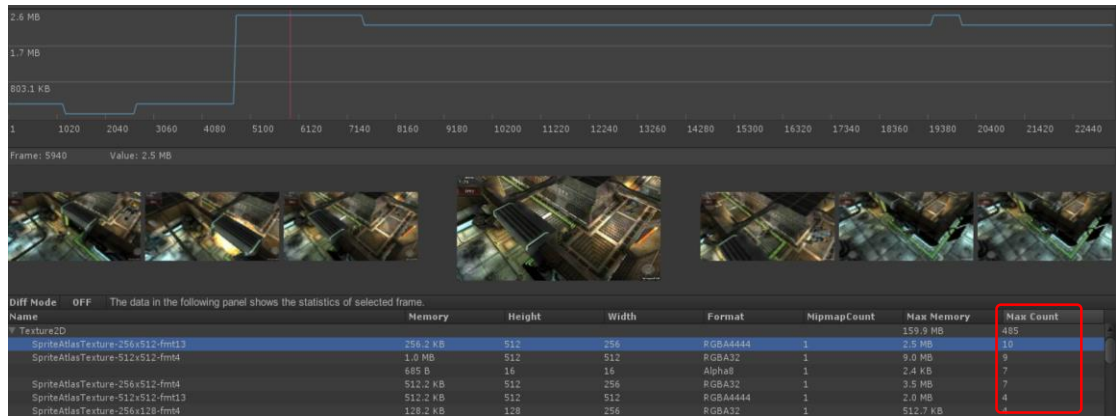
- (1) 将“Diff Mode”设置为“ON”，选择资源使用视图中的任意两帧，即可查看资源的变化情况。



- (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。



**注意：**MaxCount 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 MaxCount 大于 1 的资源称为“疑似”冗余资源。

## 4. GOT Online 使用说明

为了让您可以更加直观地分析和管理项目运行时的性能数据，UWA GOT 本地测评工具在原有基础上，新增了 Online 功能。该功能可将您本地测评所获的数据上传到 UWA 网站，生成与在线测评同样类型的数据报告，为您提供更为高效地性能分析数据及有针对性的优化建议和解决方案。同时，还可以为您提供质量监控功能，监测各个测试版本中的重要性能参数变化趋势，为您的项目研发进度和质量管控保驾护航。

具体的使用说明介绍如下：

### 4.1. 付费方式

GOT Online 服务根据上传数据的测试时长进行收费，每分钟 5 元，当前活动期间收费半价，即 2.5 元每分钟。

在官网（[www.uwa4d.com](http://www.uwa4d.com)）登陆后，可通过账户设置/产品购买/UWA GOT Online 进入如下页面进行购买。



### 4.2. 通过 GOT App 上传性能数据

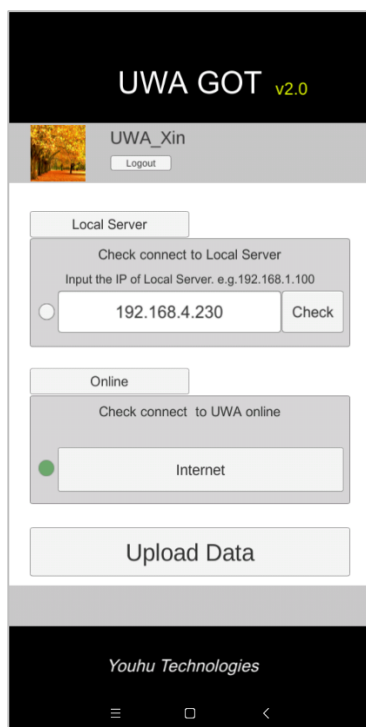
当项目在真机上测试过后，测试人员可通过 App 直接上传性能数据到 UWA 服务器进行分析，其具体步骤如下：

- (1) 在需要测试的 Android 或 Windows PC 设备上，安装 UWA GOT 插件中的 UWATool.apk（Windows 下为 UWATool.zip 中的 UWATool.exe）文件。
- (2) 打开 UWA GOT App，提示您可先进行登录。登陆账号为 UWA 网站上的注册账号，如未曾注册过，请点击“Signup”进行注册，如已有账号，则直接登录即可。





- (3) 登录后，您会看到如下界面。其中，**Local Server** 是连接本地搭建的服务器，其用法与前述文档一致。而 **Internet** 则表示是连接到 UWA 服务器，如果圆点为绿色，则表示可连接到 UWA 服务器。下一步请点击“**Upload Data**”按钮。



- (4) 点击“**Upload Data**”按钮后，则会进入到如下界面中。下方内容则为您测试过的游戏数据，您选择指定的文件并点击“**Submit**”按钮上传即可。
- 您可以通过“**Local Server**”或“**UWA Online**”来将指定数据上传到本地或 UWA

服务器；

- ii. 您可以将数据传到新的项目“New Project”中，也可以传到已有的项目例如“Demo1”中。

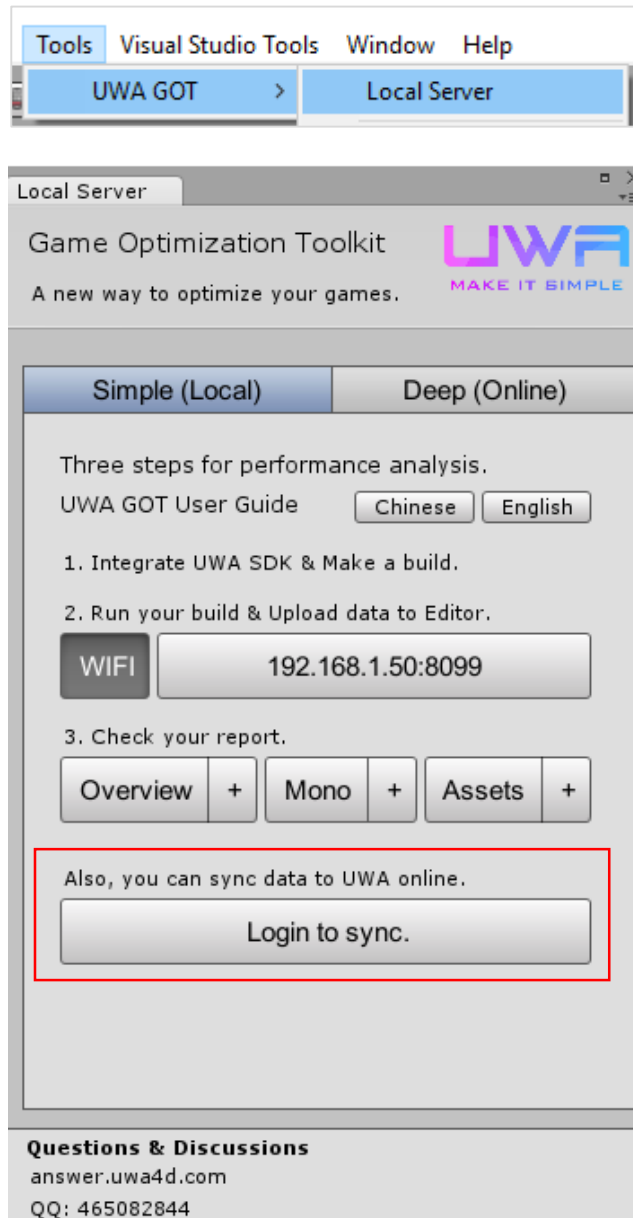


### 4.3. 通过 GOT Editor 上传性能数据

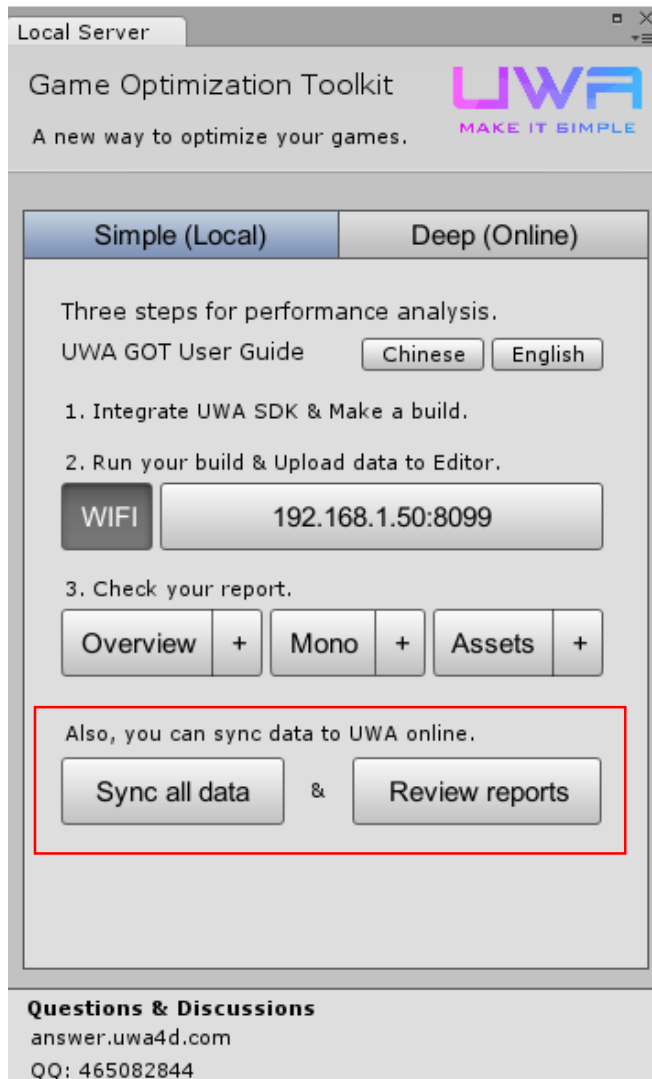
对于已经存在于 Unity Editor 中的性能数据，可以通过以下两种方式将数据同步到 UWA 服务器上进行分析。

(1) 通过 Local Server 面板批量上传

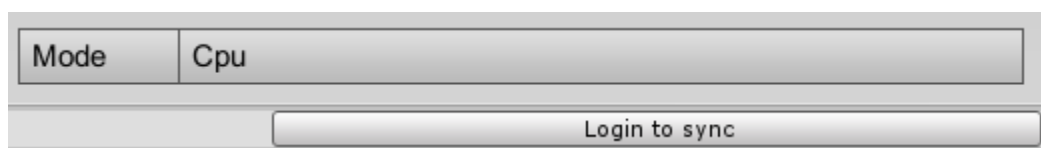
点击 Local Server 面板，并点击界面下方的 Login to Sync 按钮进行登录。



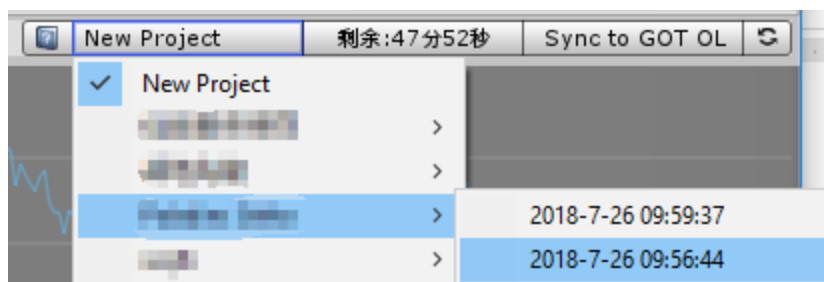
登陆后，点击 Sync All Data 即可一键同步本地所有测试数据，然后点击“Review Reports”，即可到 UWA 网站上查看性能分析报告。



- (2) 通过数据查看面板逐个上传  
 点击数据查看面板中 **Mode** 选择框下方的按钮进行登录:



登陆后, 点击 Sync to GOT OL 按钮来同步正在查看的数据, 同步单个数据之前, 可以通过中间的下拉框选择同步到哪个具体项目中去。



注意：

- i. “Sync all data” 一键同步所有数据时，每个数据都会被自动同步到与其应用名相同的项目中；
- ii. 同步单个数据时，当选择 **New Project**，会自动新建一个与其应用名相同的项目，可在 UWA 网站修改项目名称。

## 4.4. 在线查看 UWA GOT 性能报告

当数据上传到 UWA 服务器后，登录 UWA 网站（[www.uwa4d.com](http://www.uwa4d.com)），即可在本地 GOT 测试服务的相关界面中查看项目性能数据，具体步骤如下：

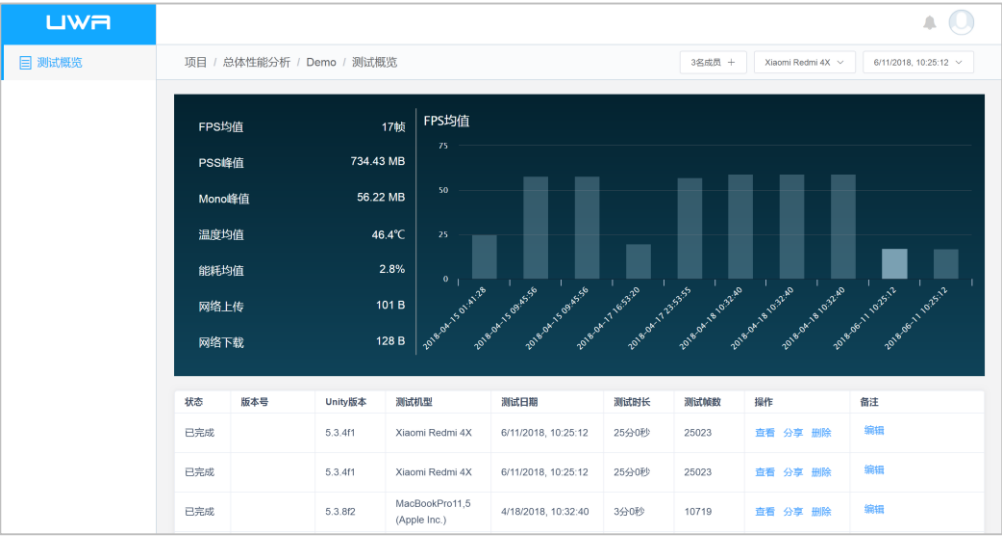
- （1）登录 UWA 网站后，即可看到“本地 GOT 测试”服务，如下图所示。



- （2）点击上图中的“总体性能分析”，你会进入如下界面。其中，项目显示时间为该项目最新一次测试的提交时间。



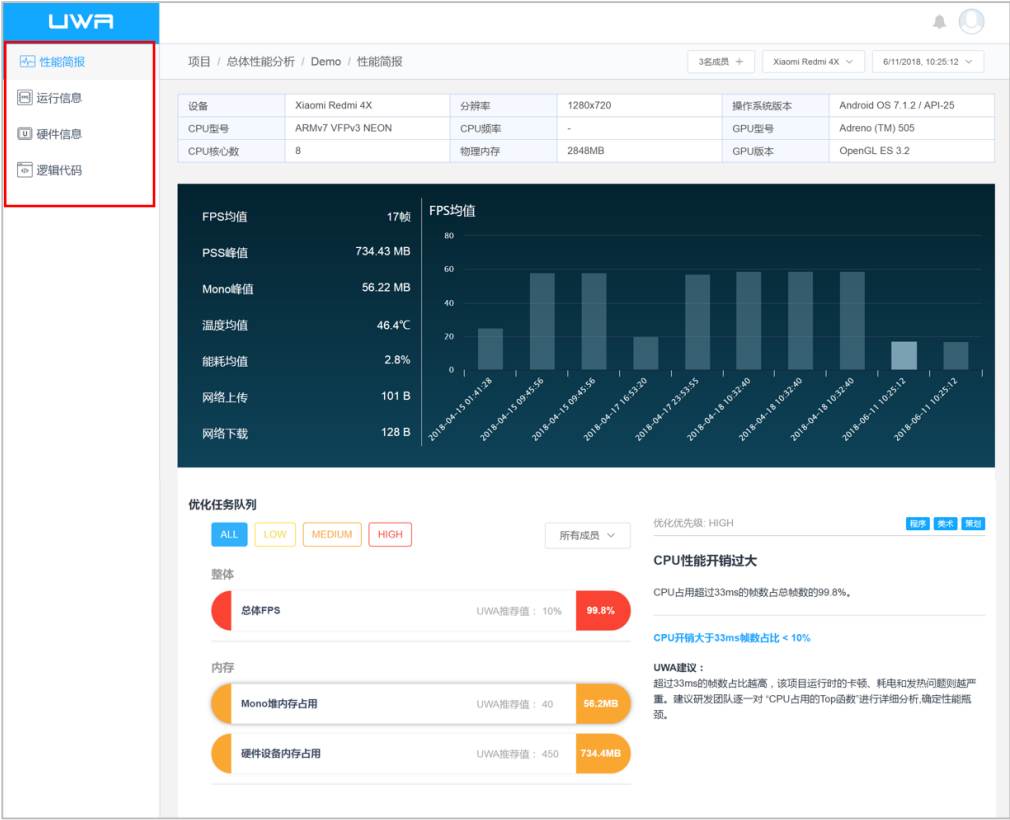
- （3）点击需查看的项目名称，即可进入该项目数据界面：



上图中主要分为上下两部分，上半部分为性能趋势，您可以点击左边栏中的重要性能参数，即可查看各次测试中的性能走势，从而判断当前版本中是否出现了重要的性能问题；下半部分为测试记录，通过它，您可以知道各次测试的状态、Unity 版本、测试机型、具体日期和测试时长。同时，您还可以邀请其他成员来一起查看该性能报告，也可以对报告进行查看、分享和删除。最后，您也可以对指定测试进行备注，记录这次测试的具体情况、修改内容等。

(4) 点击测试记录中的“查看”按钮，即可进入性能报告页面。一般来说，报告页面的内容会随着性能测试类型的不同而不同，但布局则都是一致的。下面，我们以“总体性能分析”为例来进行说明。

左边为导航栏，您可以点击这些模块来查看具体的页面信息。比如，在性能简报中，您可以看到测试设备的具体硬件参数、重要参数的性能趋势和具体的优化队列。而在其他页面中，您同样可以看到测试过程中重要性能参数具体走势、相应的运行截图和详细的逻辑代码调用堆栈。这些信息，可以帮助您对性能问题进行快速定位，并对其进行有针对性地优化。



以上为 UWA GOT Online 功能的使用说明, 您对此有任何问题, 欢迎您与我们联系。可发送邮件至: [support@uwa4d.com](mailto:support@uwa4d.com), 或加入 UWA 官方 QQ 群: 465082844。

# 附录 1：UWA API 的介绍和用法

## UWAEngine.StaticInit

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

该函数可用于：通过脚本动态挂载 UWA\_Windows，而不需要提前手动将 UWA\_Windows 的 Prefab 放入场景。

## UWAEngine.PushSample/PopSample

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

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);
```



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);
```

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.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。

---

备注：

1. 目前 UWA API 都添加了条件编译：[Conditional("ENABLE\_PROFILER")]. 因此，当发布为非 **Development** 版本时，UWA API 不会参与编译，从而对运行效率无任何影响。
2. **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。