对于memory和rendering这块，目前是获取每一帧所有的数据。然后抽取比较关键的数据进行处理，对于rendering目前已经处理的数据是Draw Calls、Tris；对于memory而言处理的主要是Used Total、Used Unity、Used GFX、Reserved Total、Reserved Unity、Reserved GFX。而且对于这两个数据，现在存放的是原始数据，均值、最低最高是画图是计算的。

目前需要对memory中mono的内存消耗以及资源内存数据进行分析和处理。首先原始数据存储到数据库，并在图表中展现。然后根据这几天对unity游戏内存的调研，得到一些主要的敏感数据结果。包括且不仅限于：

1、通过观察mono内存的变化得到无效mono堆内存的大小，并分析mono内存泄漏的可能性，比如无效mono堆内存不断增大的情况、mono峰值内存持续较高的情况；

2、一些内存分配原则（目前只是作为参考）

a、150M的总体内存标准

将Reserved Total设定在150MB，这是Unity引擎的自身内存分配，以保证App在使用到的系统库后，其OS中的整体内存也在200MB以下。

b、内存分配原则

　纹理资源： 50 MB 动画片段：15 MB 音频片段：15 MB  
　　 网格资源： 20 MB Mono堆内存：40 MB 其他：10 MB

3、还有一些敏感函数：比如UIPanel.LateUpdate。这是NGUI中CPU和堆内存开销最大的函数。

4、资源内存泄漏问题

比如切场景之后资源内存没有完全清理（AssetBundle、Resource.UnloadAsset()等）

困难：其实memory中一些detail的数据很重要，昨天看了下还没找到抓取的办法，到时候再研究下。比如WebStream、SerializedFile、AssetBundle等

参考网站：<http://www.58kaifa.com/article/537>

<http://www.jianshu.com/p/a823e70eca5c>

<http://wetest.qq.com/cube/Report?testid=5> <https://www.uwa4d.com/demo/pa.html?v=5x&platform=oculus&name=unity-chan#cpu>