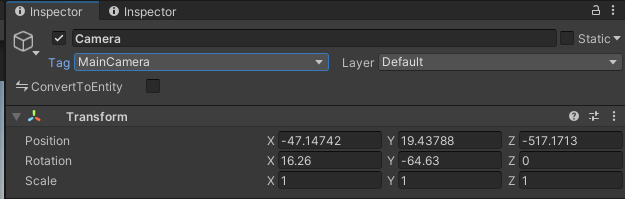
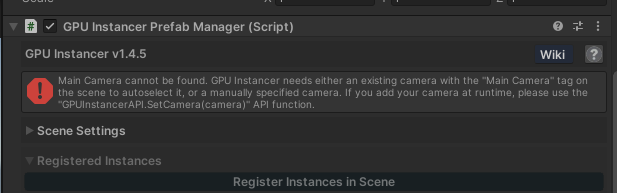
**注意主摄像头必须设置为MainCamera,**

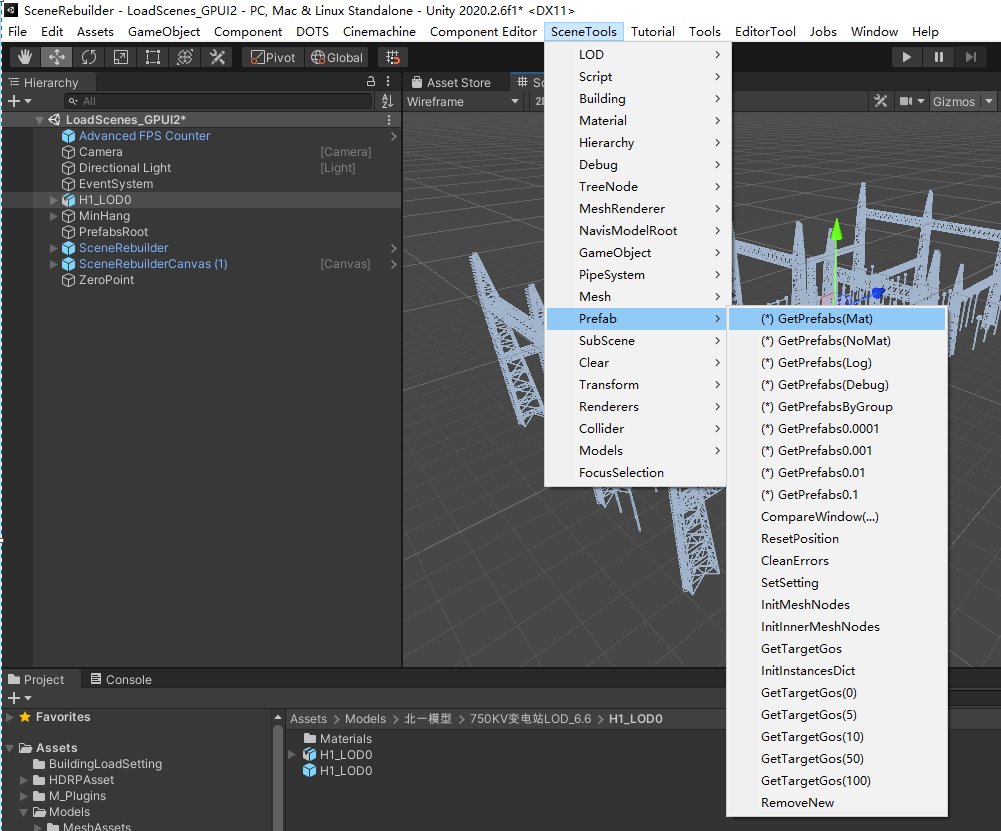


**不然运行时会出现提示，并且不会显示GPUI渲染的模型。**

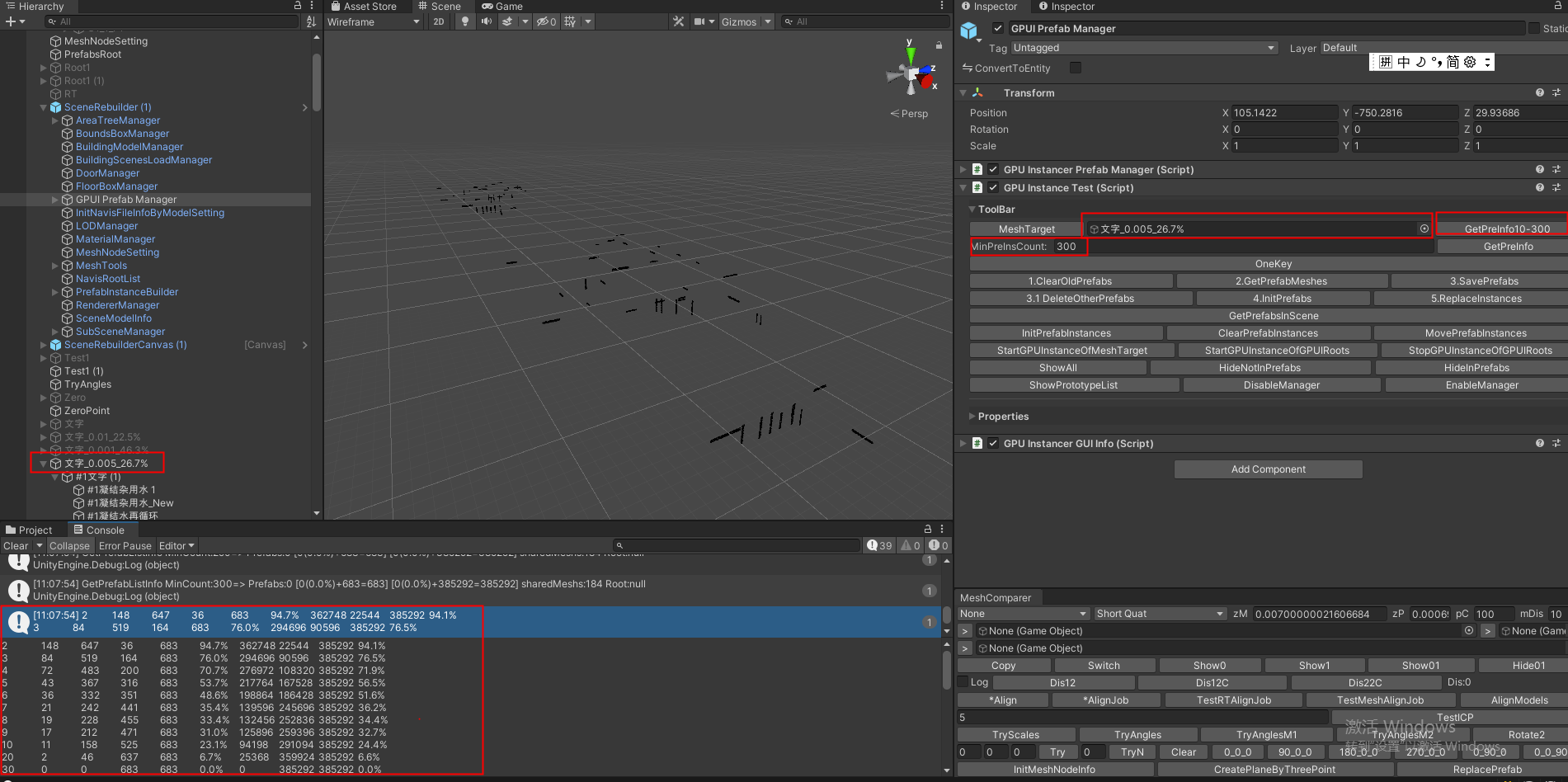


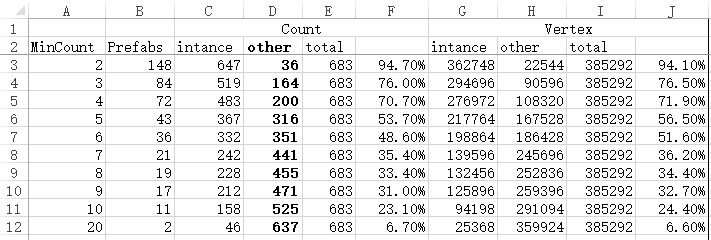
一、操作说明（从新处理）

1. 将模型进行过复用处理

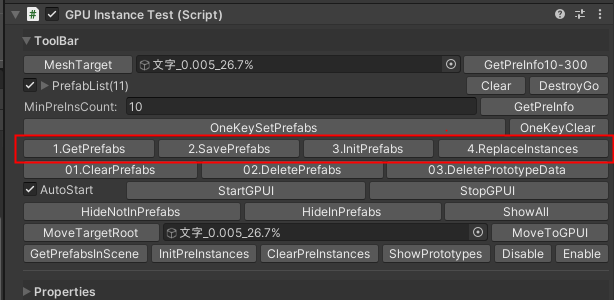


1. 将需要处理的模型的对象根节点拖到GPUInstanceTest.MeshTarget中，点击GetPreInfo10-300，判断一下不同的MinPreInsCount设置的情况下，有多少模型放入GPUI渲染。





1. 设置一个MinPreInsCount，这个参数指的是一个模型的复用数量，大于这个数值的模型设置成GPUI渲染，其他的模型继续默认渲染。复用率高的模型放到GPUI渲染中，复用率的模型就不用放了。比如这里可能应该设置为3或者4吧，先设置10测试一下。
2. 分步设置操作

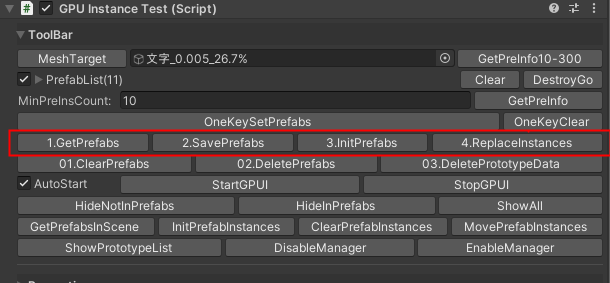


* 1. 1.GetPrefabes

MinPreInsCount设置太高的情况下，或者说目标模型的复用数量很低的情况下，没有效果，会提示



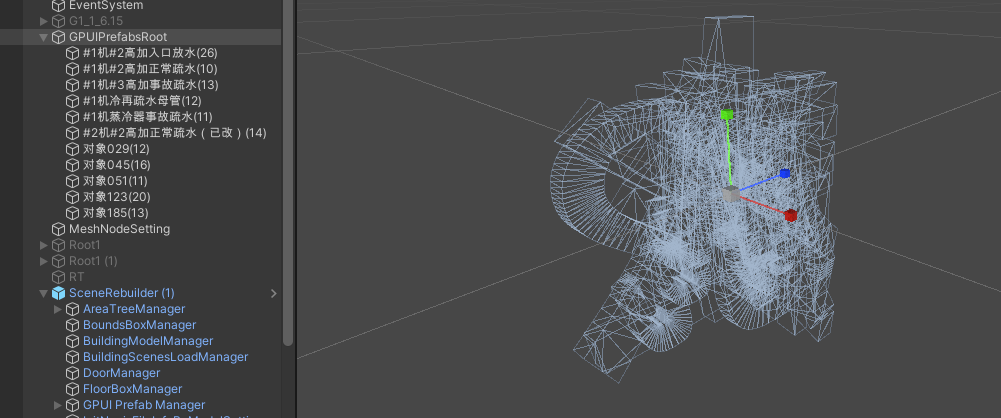
设置的数值合适的情况下，会把大于该值的模型获取并列出来。





这个信息代表有11个模型的复用数大于10，这些模型的总数是158，占原来模型的总体数量的23.1%。

同时会在场景中创建一个PrefabsRoot根节点，拷贝一份这些模型，放到该节点下面。

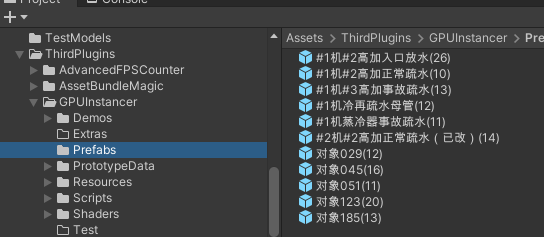


重复点击的情况下，不会有影响，会判断是否已经存在，没有存在则添加的。

* 1. SavePrefabs

将上述的模型创建预设文件，放入文件夹中。因为GPUInstance功能似乎是基于预设的。

预设文件会被保存在Assets/ThirdPlugins/GPUInstancer/Prefabs/



文件名称是模型Mesh名称(复用数量)。

同时会删除场景中GPUIPrefabsRoot下面的模型，PrefabList内关联的变成了这些预设文件了。

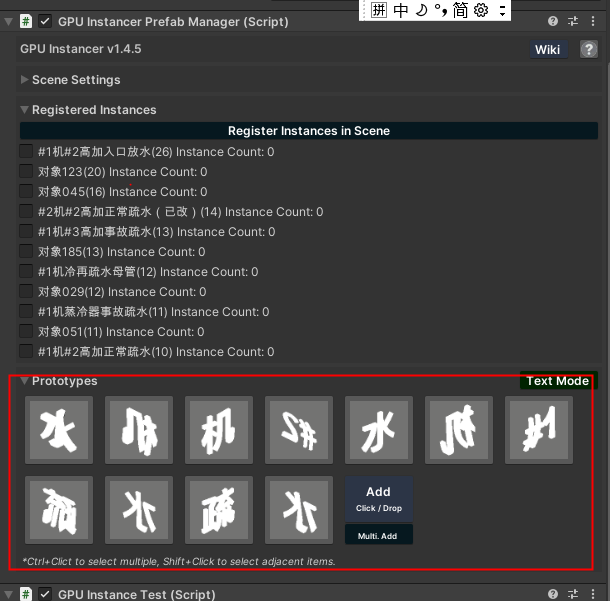
* 1. DeleteOtherPrefabs

一般不用操作，清理以前的老的预设文件。

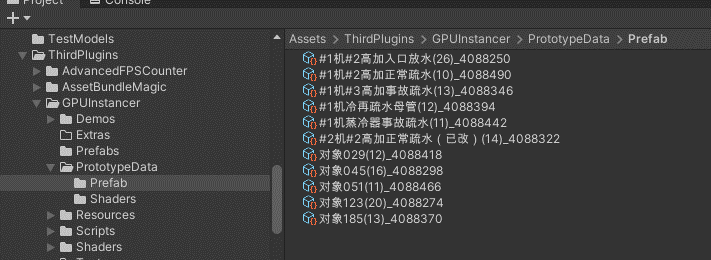
* 1. InitPrefabs

将这些预设文件放入GPUInstance脚本GPUInstancerPrefabManager。

设置的结果就是Prototypes列表中加上了刚刚的模型预设，Instances中也有。



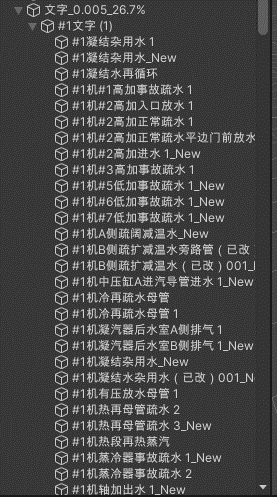
在GPUInstancer/PrototypeData/Prefab/文件夹中也有了相应的文件。



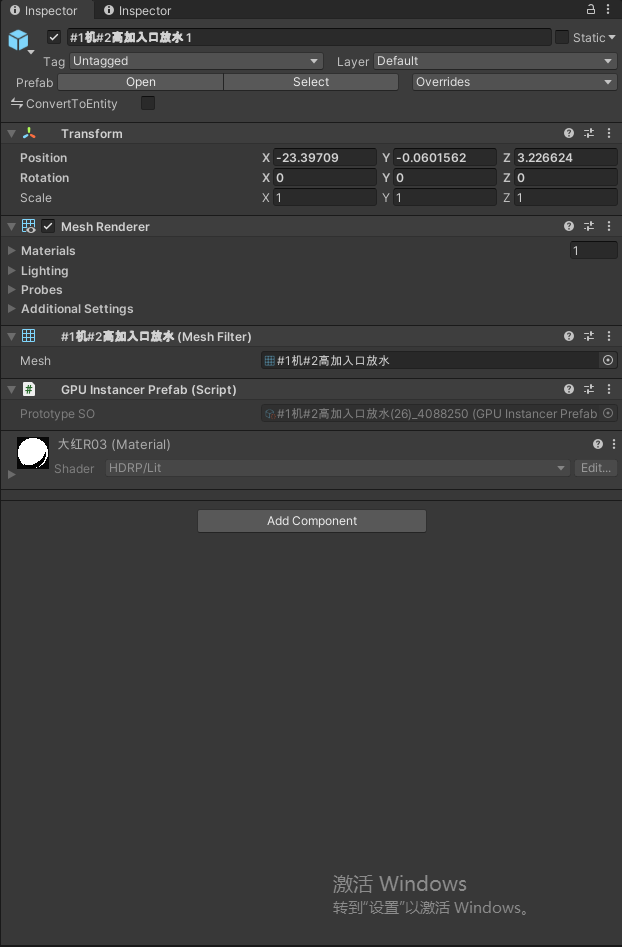
* 1. ReplaceInstances

替换目标模型关联到预设上。

替换前后

替换后的模型属性



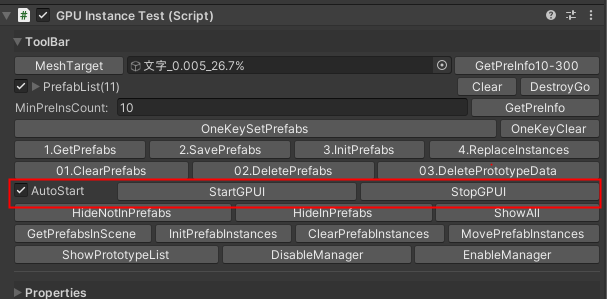
所以存在一个问题，对于其他脚本如果引用了某个替换掉的模型的话需要重新绑定或者根据名称或者Id等方式来查找关联。

1. 一键设置操作

OneKeySetPrefabs,将前面的4步自动执行一遍。

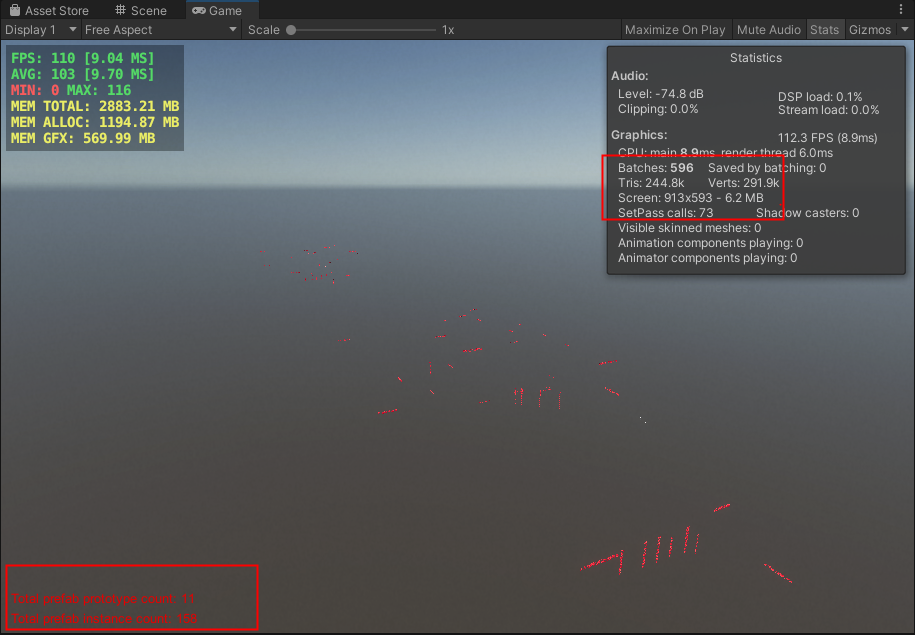
1. 运行中启动/停止GPUInstancer渲染

点击StartGPUI/StopGPUI按钮，或者设置为启动就调用，或者程序中代码调用开始/停止。



启动后：





1. 重置清理工具
   1. ClearPrefabs：

清空PrefabList列表，GPUInstancerPrefabManager. ClearPrefabsAndPrototypes()

* 1. DeletePrfabs：

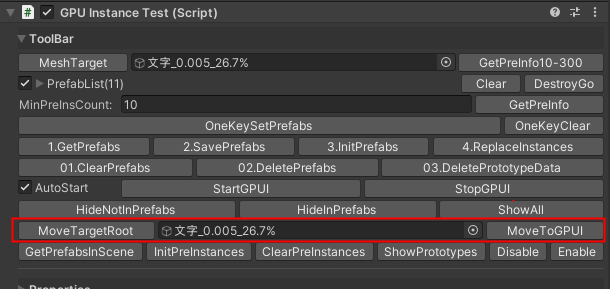
删除预设文件。Assets/ThirdPlugins/GPUInstancer/Prefabs/。

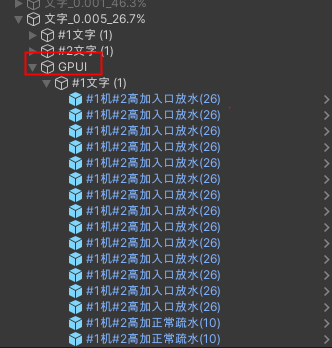
* 1. DeletePrototypeData：

删除GPUI的Prototype文件。GPUInstancer/PrototypeData/Prefab/

1. 模型处理工具
   1. MoveToGPUI：

将模型移动到GPUI节点下统一区分进行处理，保持原来的路径。



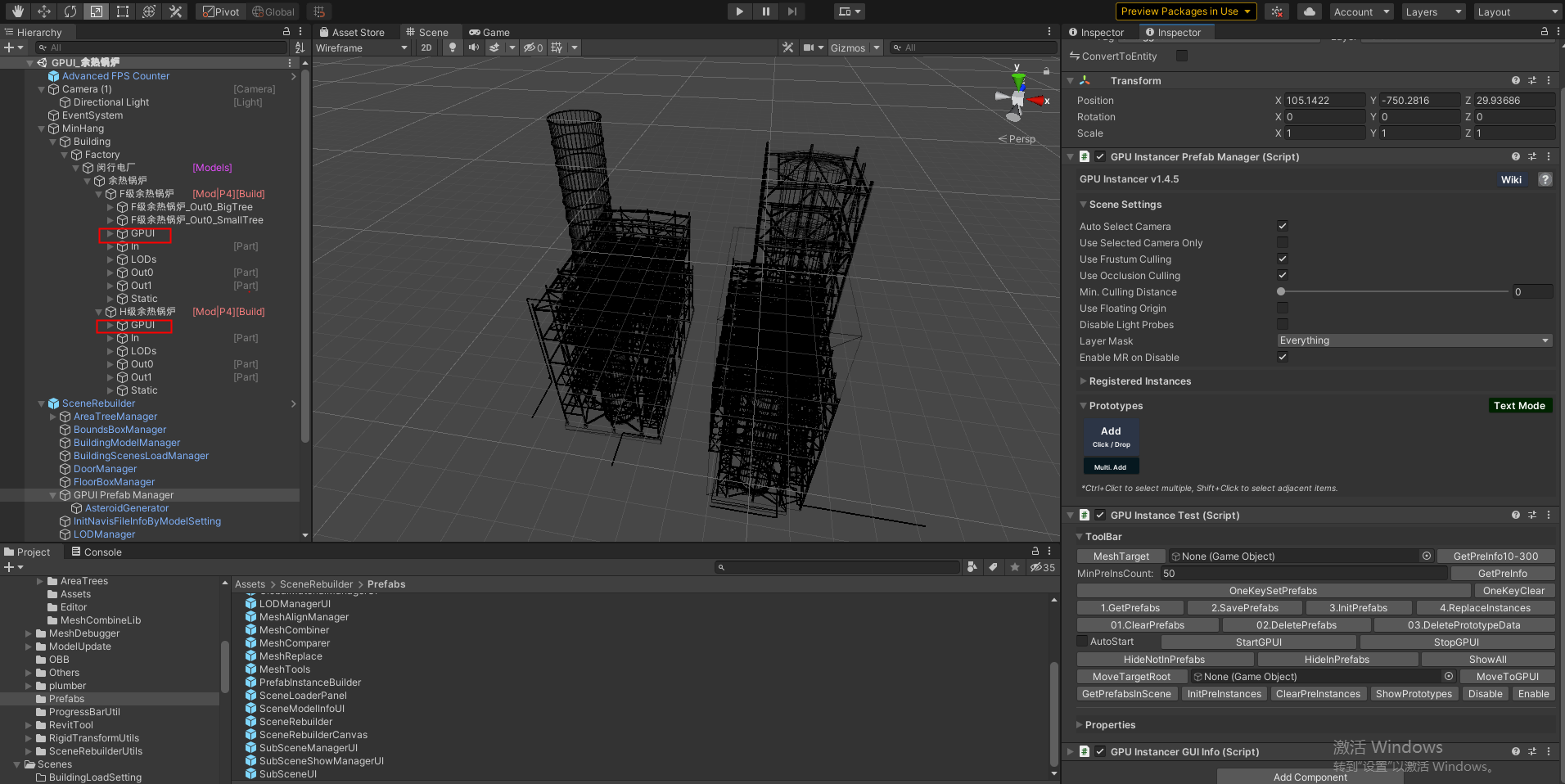


* 1. 。。。

1. 。。。。
2. 。。。。。

二、操作说明（恢复Prefabs设置）

上下文：已经做过处理，场景移动到另一台电脑上，GPUInstance没同步过来。

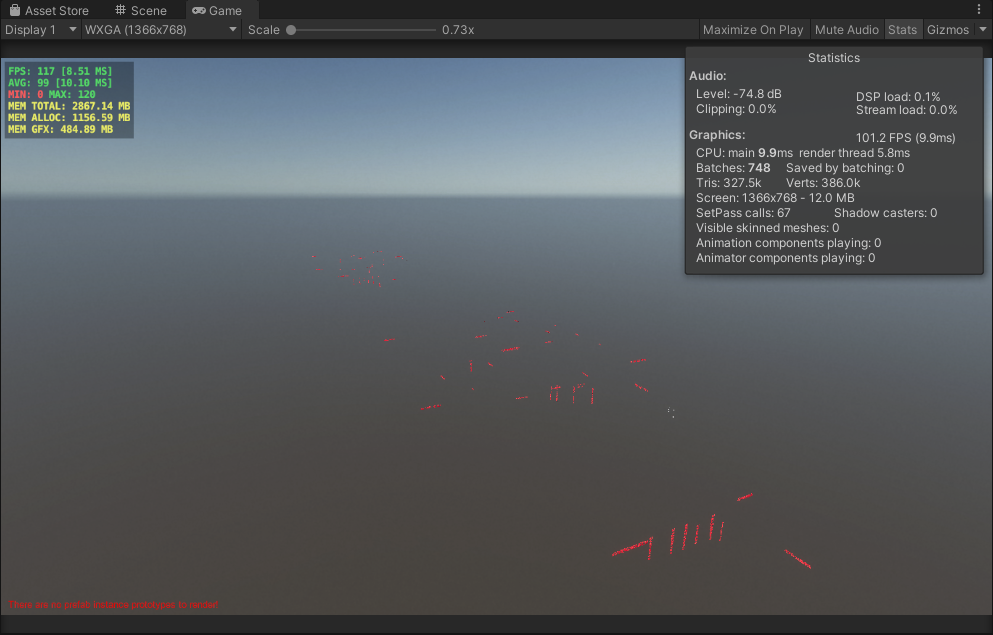


H级余热锅炉和F级余热锅炉大于50个复用模型的模型都已经分别移动到不同的GPUI节点下了。

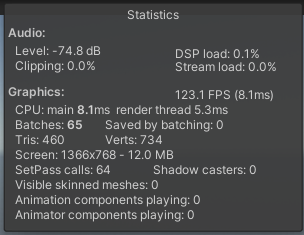
1. 设置GPUIRoots。
2. 获取Prefabs
3. 保存Prefabs
4. 设置到GPUInstanceManager
5. 替换Instances。

三、性能测试

1. 默认渲染情况下的性能参数。



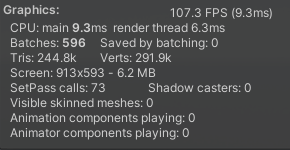
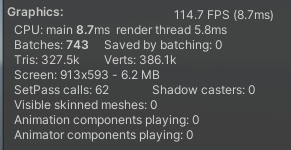
无模型时是



模型复用前后参数是一样的， 模型的复用在不结合GPUI的情况下只是减少了内存占用。

1. GPUI渲染的性能参数。

前后



这里数量少还看不出区别来。参数上Batches和Tris少了，Calls多了。