

URP中SSAO优化 - Update GTAO

H项目中使用了SSAO用于提升场景立体感。但是URP中的SSAO使用了DepthPrePass将场景中所有Opaque物件渲染了一遍获取depth贴图，然后再进行SSAO计算，最后在渲染场景物件的时候采样SSAO计算结果贴图加入到光照计算中。我们对该方案进行了优化，去掉了PreDepthPass，取而代之的使用OpaquePass之后的CopyDepthPass来获取depth贴图，并计算SSAO。最终用一个blit将SSAO附加到OpaquePass结果上。节省了将所有Opaque物件渲染一遍的开销。由于CopyDepthPass与MSAA有冲突，因此必须关掉MSAA，采用其他AA算法替代。

代码

<http://svn.ifunplus.cn/svn/common/UrpSsaoOptimize>

<<http://svn.ifunplus.cn/svn/common/UrpSsaoOptimize>>

性能比较

我们在项目中测试了SSAO的优化，目前在高端机上可以使用，以下的优化前后的数据比较：
内城中用copydepth替代DepthPrePass，节省了7ms

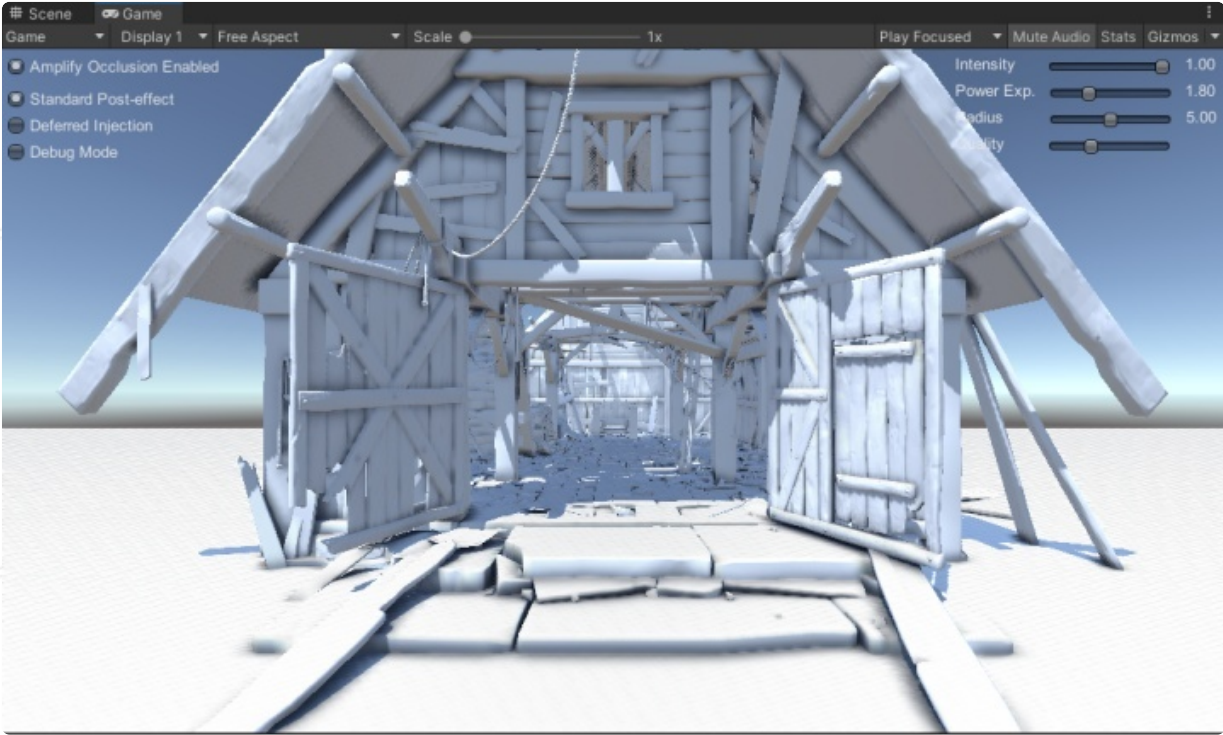
MI8 (Snapdragon 845)	11.4版本 GPU 耗时	方案	11.1版本 GPU 耗时
MainLightShadow	11.1	Static物件cache	MainLightShadow 12.2
Opaque	47.9	优化pbr shader	AdditionalLightShadow 2.7 已优化掉
CopyDepth	0.7		DepthPrePass 7.7 已优化掉
SSAO	8.9	降低SSAO分辨率 @立鑫	SSAO 8.7
Transparent	1.7		Opaque 69.3 美术资源优化一部分
SMAA	4.6	TAA或者其它替代方案	Transparent 5.5
Bloom	0.9		Bloom 0.9
UberPost	.3		UberPost .3
blit	0.8		blit 2
UI	0.3		UI 5.4
FinalBlit	0.5		FinalBlit 0.5
	80.4		117.9

此外，经测试降低SSAO分辨率后有进一步的提升。

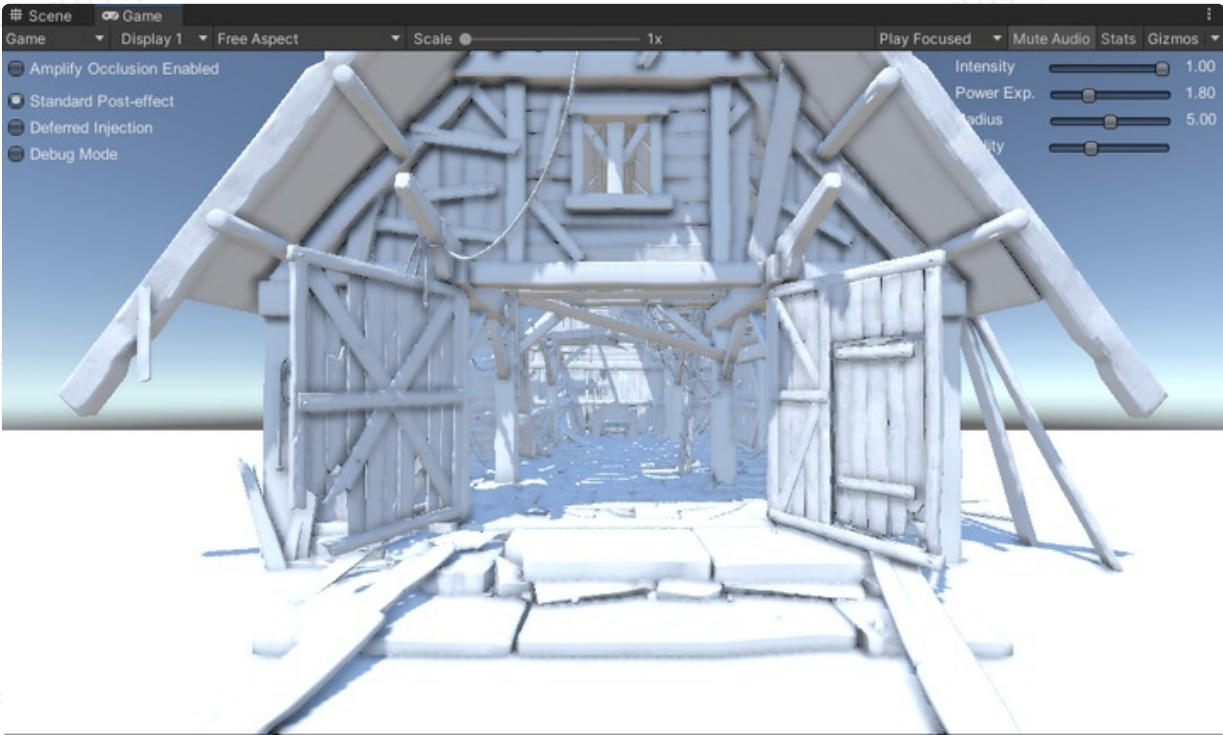
----- update 2022.3.1 -----

我们把Amplify的GTAO移植到了URP，并在h项目中使用，效果比URP自带SSAO更好：

Amplify GTAO：



URP SSAO:



H内城： GTAO vs noAO



代码

<http://svn.ifunplus.cn/svn/common/GTAO> <<http://svn.ifunplus.cn/svn/common/GTAO>>