

自动化性能优化检测工具

检测对象：Unity-based VR APP Project

美术检查

程序检查

配置检查

模型减面（物件背面、内部不会看见的面删掉，所有物体不超过20000个三角面，人模模型面数控制在3000-4000面，移动端单个人体骨骼数量控制在30个~50个左右）

合并网格：将多个网格合并为一个网格，以减少Tris数量。

地形优化（T4M插件转换，地形LOD）

贴图

动画

使用遮挡剔除（Occlusion Culling）：根据场景中物体的可见性，只渲染可见的物体，从而减少不必要的渲染操作。这样可以减少渲染的Tris数量，提高游戏性能。

使用 GPU Instance（GPU 实例化）：通过使用 GPU 实例化技术，可以将多个相同的网格实例化为一个批次进行渲染。这样可以减少不必要的绘制调用和渲染开销，提高游戏性能

内存分配

降低或禁用 Accelerometer Frequency（加速度计频率）

Unity 每秒钟会以一定次数统计移动设备的加速度计状态。如果应用并不会用到加速度计，我们完全可以禁用该功能或降低统计频率来获得更好的性能。

贴图尺寸减小

贴图使用Bitmap贴图类型，尺寸主要为2的N次方

贴图压缩，对贴图进行PVRT（iOS）或是ETC（Android）格式的压缩可以减少大量内存消耗。建议场景的材质可以用Substance材质系统转成.sbsar 格式材质。 Substance材质可以大大压缩贴图数据的大小但不损失贴图质量。

检查纹理读/写标记，检查Inspector -> Advanced -> Read/Write Enabled选项，默认必须要关闭，开启纹理资源的读/写标志会导致双倍的内存占用。

检查Mipmap标记，检查Inspector -> Advanced -> Generate Mip Maps选项，未压缩的纹理资源启用Mipmap标志会增加内存占用。

检查纹理资源的过滤模式，检查Inspector -> Filter Mode选项，纹理的过滤模式一般不建议使用Trilinear，会占用较高的计算资源。

动画资源压缩方式 查看Inspector -> Animation Tab -> Anim. Compression选项，动画资源使用最佳压缩方式可以提高加载效率。off：表示不采用压缩处理。keyframe Reduction：使用关键帧进行处理（旧版本unity：keyfram, reduction and compression表示关键帧减少和压缩）。Optimal：（新版本unity）自动选择一个最优的压缩方式。

检查动画资源的Optimize Game Objects选项，查看Inspector -> Rig Tab -> Optimize Game Objects选项，动画资源应该勾选Optimize Game Objects选项来减少CPU消耗。

评价指标

FPS（Frames per Second）：显示当前游戏帧率。帧率越高，游戏画面越流畅。为提高帧率，可减少游戏对象数量、优化渲染方式等。

Draw Calls：显示当前帧需要渲染的次数。过多的Draw Calls会导致游戏运行缓慢。降低此指标可通过优化场景、材质或使用批处理技术等方式。

Tris：显示当前帧需要渲染的三角形数量。过多的三角形数量同样会导致游戏运行缓慢。减少此指标可采取降低模型细节、使用LOD技术等方法

Memory：显示当前游戏所使用的内存大小。过高的内存占用会导致游戏运行缓慢或崩溃。降低此指标可通过优化资源加载和释放方式等方法。

实验方法

数据集：UnityList 开源VR项目

通过对比优化前后的指标提升，评估性能

实现方法

形式：Unity 编辑器扩展、python脚本程序

算法：代码静态分析？模型减面？

难点

如何高效提升性能表现

参考

性能蓝皮书

性能优化UWA

Unity3D 实用技巧 - 优化实践之路（一）

Unity游戏优化与性能提升