

图形管线渲染性能研究报告

小组编号：1120213081 撰写人：介应奇

测试用 GPU 参数

项目	参数
显卡型号	NVIDIA GeForce GTX 1650
流处理器数量	896 个
显存规格	4GB GDDR5
主频	1125—1560 MHz
支持 DX 版本	DX12 及以下
内存接口	128 位
最大功耗	75W
核心面积(mm^2)	200
纹理单元	56
光栅单元	32
显存频率(Gbps)	8
单精度浮点性能 (TFLOPS)	3
纹理填充率(GT/s)	93.3
显存宽位 (GB/s)	128

图形管线性能分析

作业 2 字符森林

字符数目	图元数	顶点数	帧率
6720	322560	967680	60fps 左右
21000	1008000	3024000	21fps 左右
210000	10080000	30240000	2fps 左右

表格 1 顶点和图元数量与帧率关系表

结论

GPU 处理的图元和顶点数目越多帧率就越低
尤其从后两行我们可以明显的看到，计算力是一定时，在字符数上升了一个数量级的情况下，帧率就下降了一个数量级。

作业 3 林中飞翔

相机位置朝向与帧率的关系

$$\text{视野中没有字符} - \text{视野中有字符} = 2 \sim 4$$

结论

当在 PS 之前的光栅化阶段裁剪掉顶点之后，帧率会有所上升，也就是说相机中的顶点越少帧率越高，但是提高并不大，这也很合理，因为前半部分还是都做了一遍了，要保证能在转头的几帧里能看到顶点，这部分保障所需的计算还是做了，虽然看不到，因此帧率不会大幅变化。

作业 4 光照效果

光源类型	帧率
平行光	65-67fps
点光源（3 盏）	64-66fps
方向光	65-68fps

表格 2 光源（光源类型、数量）与帧率的关系

结论

由于三个光源我设置的基本上都要与所有字符作用，因此帧率没有太大差异，按理来说应该方向光的帧率最高，因为有裁剪掉找不到的部分，其他两个类型的数量越多帧率越低。

作业 5 镜像森林

GS 使用情况	帧率
不使用	65-67fps
使用	64-65fps

表格 3 使用平行光字符数同为 5600 个字符前提下使用 GS 与不使用 GS 帧率表

结论

使用 GS 进行渲染顶点数量翻倍的效果对于 GPU 性能影响不大，几乎可以忽略不计从而达到低成本渲染大量重复几何体的效果。

作业 7 镜中世界

添加了一面镜子后要达到 60 帧就只能绘制 945 个原像字符了，这里由于加入法线方向的时候采用了拆分顶点的方式，因此顶点数量和第 3 次作业之前的顶点数量基准不同了，一个字符应该等同与原先的 3 个字符，也就是 2835 个字符，加上光照等其他操作的影响，正好基本上是之前不做反射的一半的字符数目，因为相当于要将整个场景渲染两边，因此能支撑的顶点数变为原来的一半，和理论相符。

结论

做一个完全的镜中世界的话，使用真正地再渲染一遍的方式的话就是会对性能产生对半的影响。

总结

这几次作业中分别主要涉及到的图形管线阶段有：IA 阶段、VS 阶段、PS 阶段、GS 阶段、RS 阶段，而其中重点由这些作业操作的部分其实只有 IA、VS、PS、GS 阶段，RS 只是设置好参数，并没有进行对具体的裁剪方式进行编程，所以分析主要也是在这几个可操作的部分上；

我们可以看到在作业中最显著地影响到图形管线性能的就是待绘制的顶点图元数量，因为这个是最根本的，基本影响全局的性能的，因为每一个顶点都要做一遍全流程，要处理的对象越多，对性能的要求必然更大，具体来说是在 IA 装配阶段和 VS 顶点着色阶段会有很大的性能影响这一点在最后一次作业中也可以得到体现，因为是要将整个场景全部重新绘制一遍所以只能绘制之前一般数量规模的字符森林了，做镜子确实是很费资源的一件事；

从作业三我们可以看到，后半段的光栅化处理中裁剪操作对于减少 GPU 的性能压力作用不是很大，因为这只节省了最后一步的光栅化中所需的部分计算资源，但是无论是否在视野，是否要做光栅化计算，都需要做 VS 阶段，这个阶段应该才占据了更多的 GPU 资源；

PS 阶段会随着光源的数量增加而对性能产生越来越大的需求；

而夹在 VS 和 PS 之间的 GS 阶段就对性能影响就不是很大了，可以用于做生成大量重复图形的工作。