Opengl 渲染流程：

Cpu: FBX -🡪Meshrender

Fbx obj : 模型文件 里面包含了 uv 顶点位置 ，法线 切线 等 渲染所需要的信息。

MeshRender : 将这些信息 传递到GPU 。 skin mesh render / mesh render mesh filter

skin mesh render : 带蒙皮的骨骼 。

mesh render : mesh render 主要是将 顶点等信息传递到gpu .

mesh filter : 表示 将 哪个模型 信息 传递给 gpu .

GPU:

渲染管线：

顶点着色器 -🡪 光栅化 -🡪 片段着色器 --🡪 alpha 测试 -🡪 模板测试

----🡪 深度测试 ---🡪 Blend -🡪 Gbuffer --🡪 frontBuffer -🡪 frame buffer -🡪显示器

顶点着色器：

1, 计算顶点的颜色。

2，将物体坐标系 转换到 相机坐标系。

光栅化 ：

将顶点 转换成 像素 。

100\*100

（0，0，0，0）

（1，1，1，1）

插值：

片段着色器：

1, 已经是像素点 。

像素： RGBA 4 通道组成 。 RGBA 8888 8bit .

屏幕 ： 720\*1280 横 有 720 个像素 竖排 12080 个像素。

Float[720 \*1280]

顶点 4 个 到了 片段着色器 100\*100 .

结论 ：

顶点着色器 会运行 4次 片段着色器 会 运行 10000.

顶点着色器 和 片段着色器 运算次数 不是一个量级 。

能 把 运算 放在 顶点着色器 就放这里。

片段着色器 ：

1, 纹理采样 。 从 纹理 像素 赋给 像素 。

2， 像素 跟 灯光计算 。

Alpha 测试： 挑选合格的alpha 像素显示。

模板测试： 像素 还可以携带 模板 信息 。 达到条件的模板值 。

深度测试 ： 符合条件的 像素 就通过 不然就丢弃。

Blend : 将 当前要渲染的 像素 和 已经 渲染出来的像素 混合运算。

GBuffer : RGBA 模板值 深度值 等 。

Float[ 720\*1280\*4 ]

Front Buffer : float[720\*1280]

framebuffer : float[720\*1280]