



STEAM VR™

Tracking 培训



STEAM® VR
Tracking 培训

渲染模型

关键概念

- 渲染模型是 SteamVR™ 中显示的图像
- 渲染模型均构建于同一坐标系中
- 渲染模型至少包含以下项目：
 - Wavefront OBJ 网格
 - 材料文件 MTL
 - 纹理文件 PNG 或 TGA
- OBJ 要求
 - 单个形状
 - 少于 65,000 个顶点

用户注意

- 此处所示流程为简化后的流程
- 目标是用于定位评估的代表性渲染模型
- 生产渲染模型有更多功能
 - 更好的 UV 图
 - 更精细的纹理
 - 对 SteamVR™ 做出反应
 - 专业 3D 设计师可执行此工作
- 工程师需要定位结果
 - 可参考应用程序渲染模型
 - 能够制作一个代表对象的简单模型会更好

流程概述

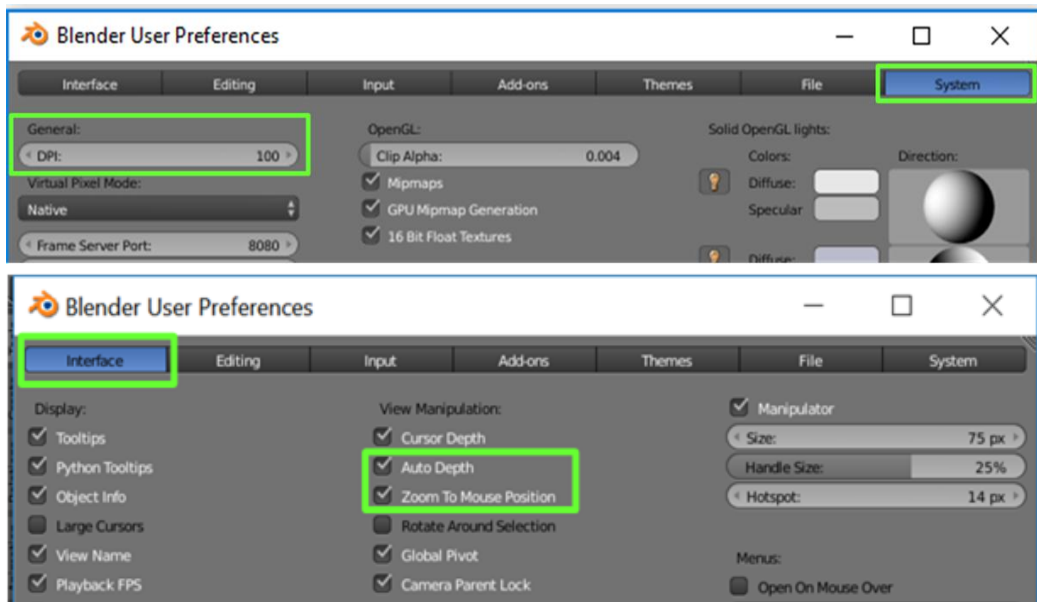
- 创建参考控制器的 STL 文件
- 创建对象的 STL 文件（适用于渲染模型）
- 将两者都导入同一坐标系
- 将您的对象与参考模型对齐
- 导出对齐的 STL，并用作渲染模型 OBJ 的基础
- 将对齐的 STL 导入到 Blender 中
 - 创建材料
 - 创建纹理
 - 创建 UV 图
 - 导出 OBJ、MTL 和 PNG
- 在 MeshLab 中试用渲染模型
- 在 VR 中试用实体渲染模型！

开始练习

- 我们需要将我们的渲染模型与参考渲染模型对齐
- VR 应用程序替换控制器渲染模型
- 使用哪个模型？
 - 模型大部分内容的编写方式都具有良好的支持性
 - HTC Vive 控制器
- 让我们制作 Vive 控制器的 STL...

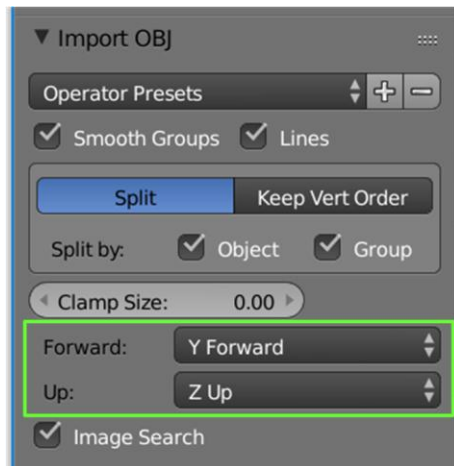
在 Blender 中设置首选项

- 打开 Blender
- 文件 > 用户首选项...
- 调整：系统
 - 增加空间大小的 DPI
- 检查：界面
 - 自动调整深度
 - 缩放至鼠标位置



创建参考 STL

- 删除默认立方体
- 导入 Vive 控制器 OBJ
 - 150_the_render_model\vr_controller_vive_1_5\vr_controller_vive_1_5.obj
 - 注意导入设置
 - 导入屏幕的左下方
 - Y = 向前, Z = 向上



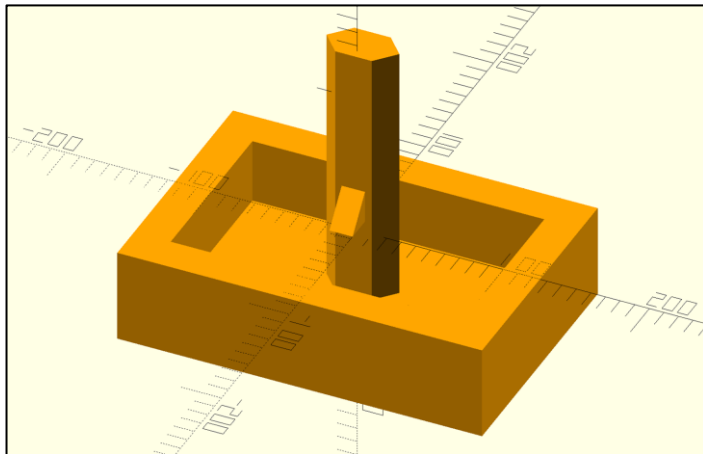
导出参考 STL

- 这是渲染模型坐标系
 - 相对于手部来说，这个原点和方位分别是什么？
- 将 STL 导出到练习文件夹
 - 150_the_render_model\htc_vive_controller.stl



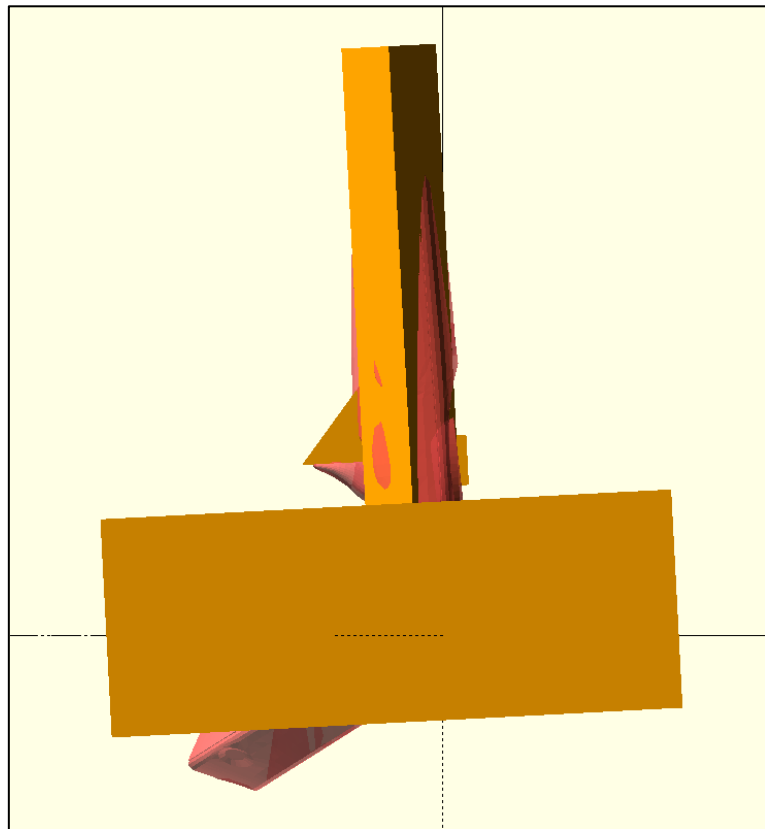
将两者都导入到 OpenSCAD

- Thor_hammer.stl 是我们对象的 STL 文件
 - 进行简化以：
 - 降低顶点计数
 - 展开 UV 图
- 将两个 STL 都导入到 OpenSCAD
- 在 OpenSCAD 中打开
 - 150_the_render_model\align_controllers.scad
- **Vive 控制器在何处？**



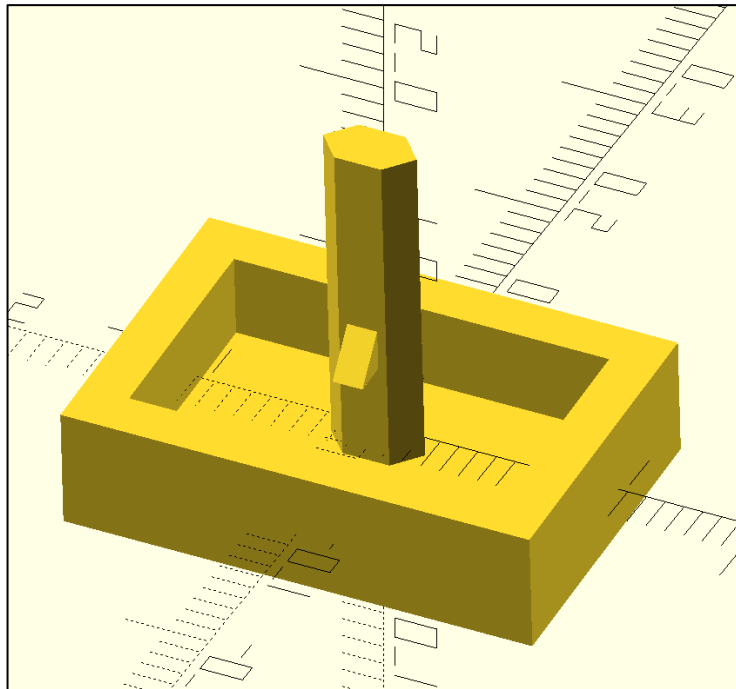
对齐模型

- 将比例尺调整至米
- 让 Vive 控制器变为透明
 - 在模型之前使用 ‘#’
- 调整雷神之锤的平移和旋转
 - 对齐触发器
 - 对齐触控板和按钮
 - 使把手相互平行



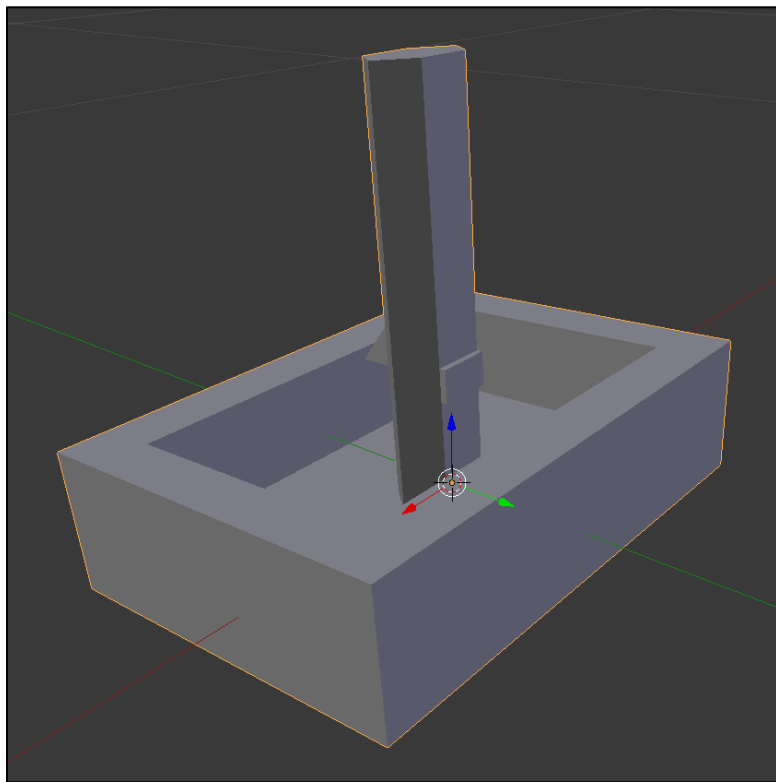
导出渲染模型 STL

- 单独的雷神之锤
 - 在锤子前方使用 ‘!’
- 构建对象
 - 按 F6
- 构建更复杂的网格需要较长时间
- 文件 > 导出 STL...
 - 150_the_render_model\thor_aligned.stl



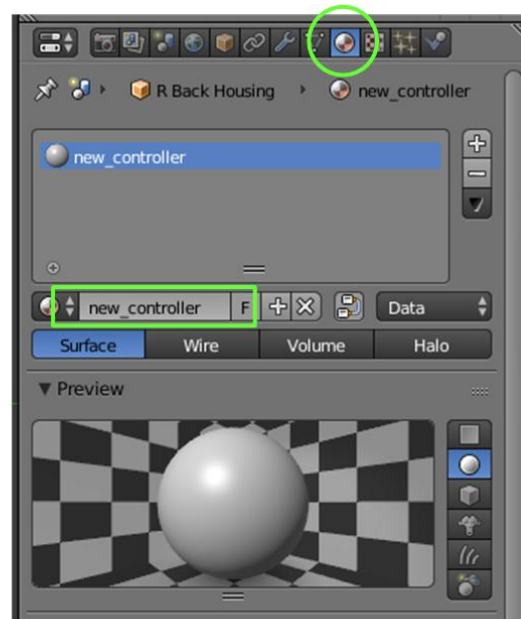
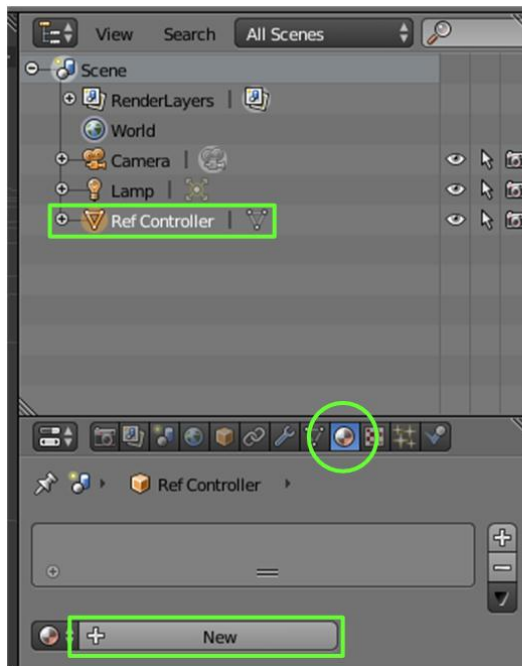
将 STL 导入到 Blender

- 打开一个新的 Blender 文件
- 删除该立方体
- 导入雷神之锤
 - 150_the_render_model\thor_aligned.stl



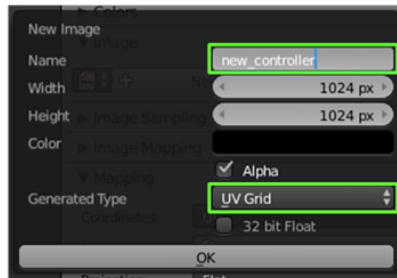
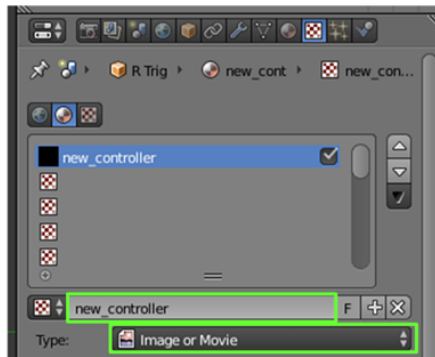
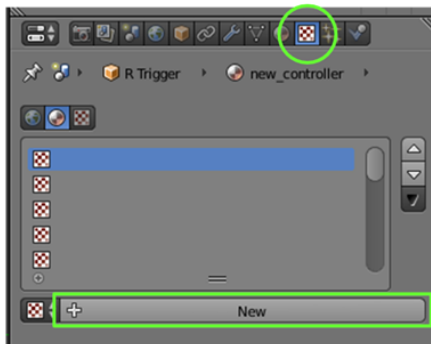
添加材料

- 选择网格
- 单击材料按钮
- 为材料命名



添加纹理

- 单击纹理按钮，“+New（新增）”、“Image（图像）”或“Movie（影片）”、“+New（新增）”
- 为纹理命名，并选择“UV 网格”

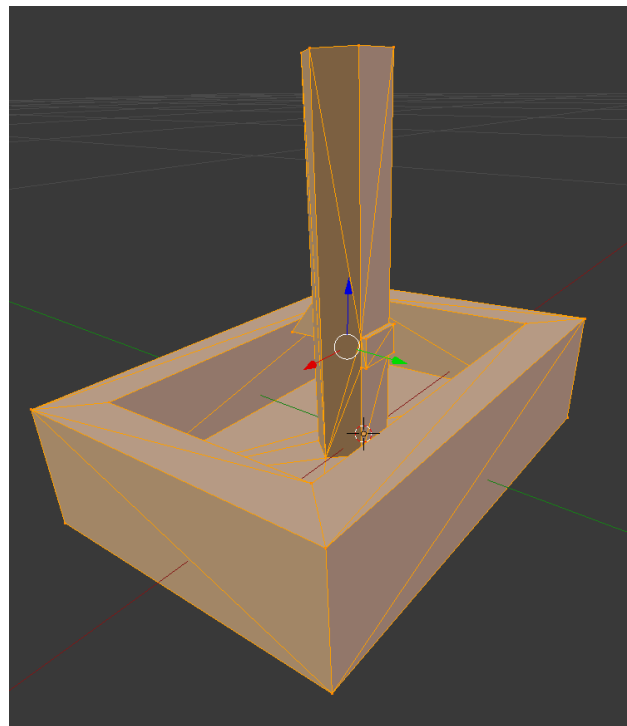


创建 UV 图

- 谁能解释一下什么是 UV 图？
- 这个 UV 图可能不怎么样...
 - 但如果我们不创建 UV 图，就无法加载渲染模型
- 切换至“编辑”模式

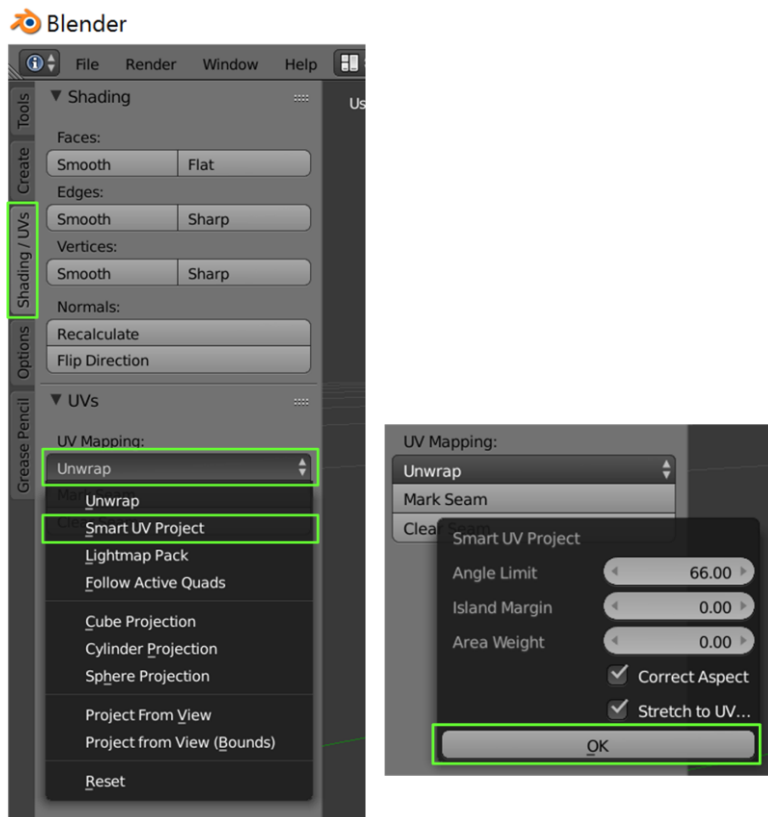


- 按 ‘a’ 全选



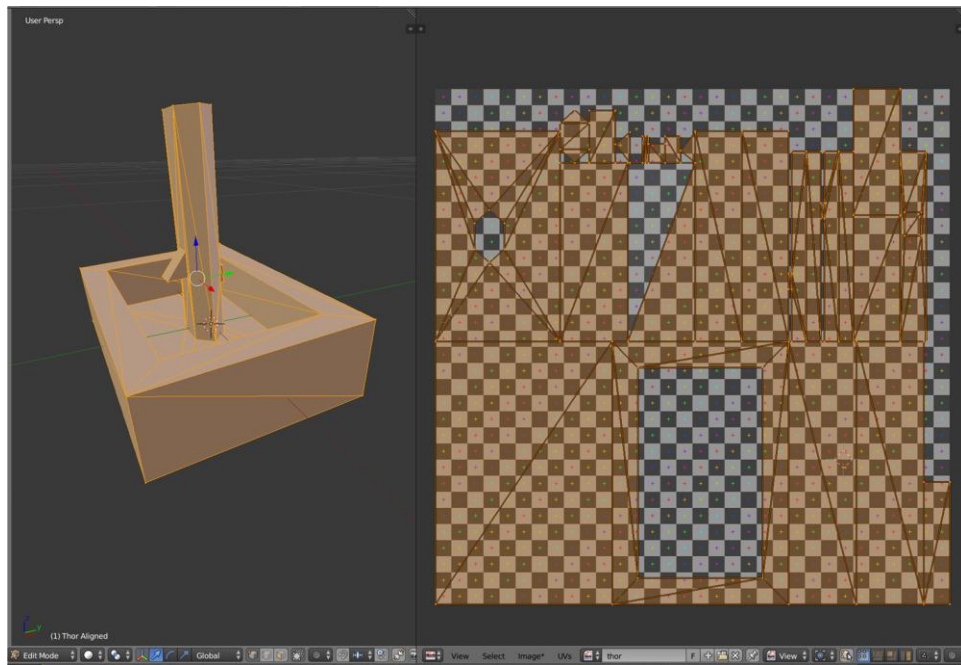
创建 UV 图

- 展开对象
- 单击
 - 遮蔽/UV
 - 展开
 - 智能 UV 项目
 - 确认



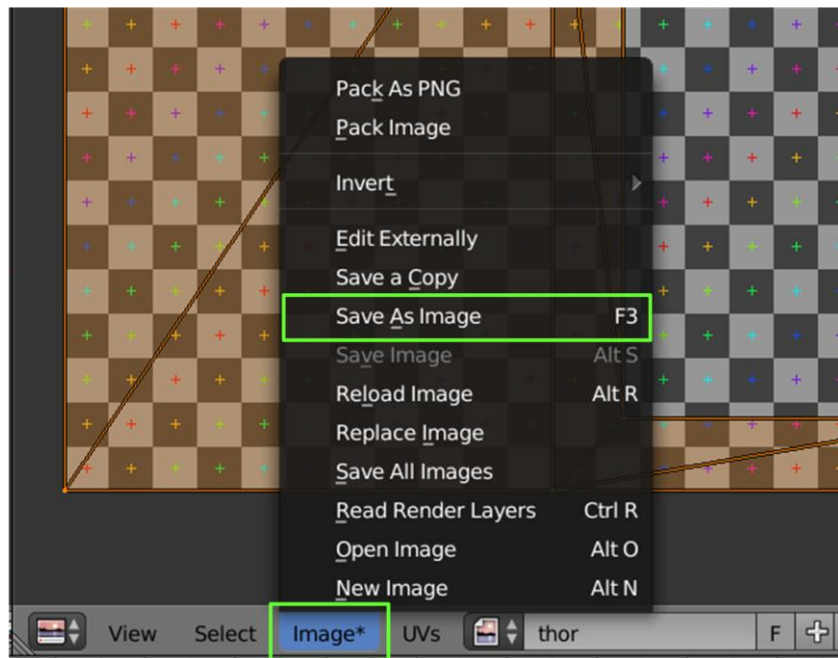
并排显示这两者

- 拖出一个新窗口
- 选择 UV/图像编辑器
- 选择所有 ‘a’



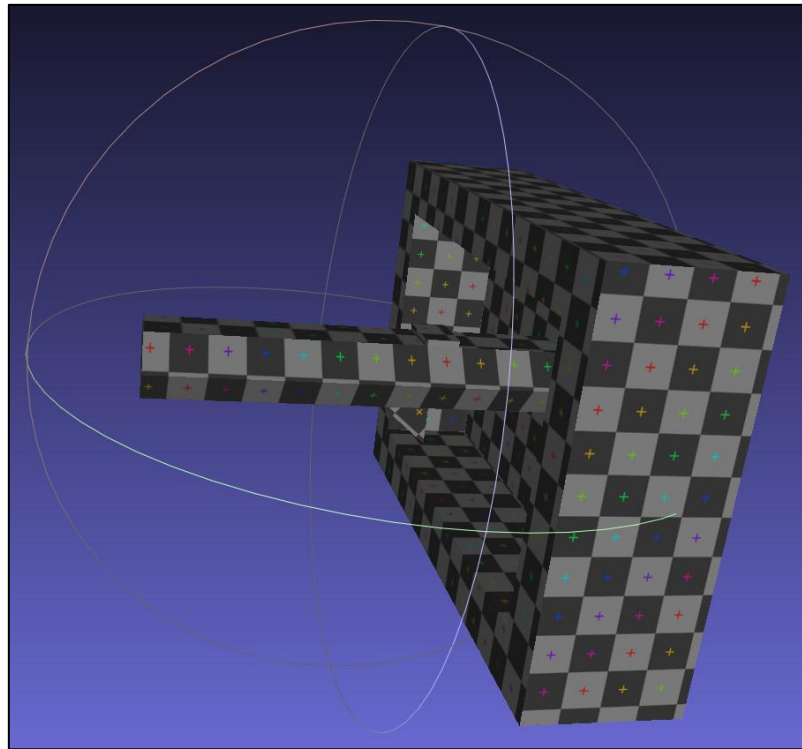
保存纹理图像

- 单击
 - 图像
 - 另存为图像
- 将文件保存在:
 - 150_the_render_model\



导出 OBJ

- 保存 Blender 文件
 - 文件 > 另存为...
- 导出 OBJ
 - 文件 > 导出 > 波前 (.obj)
 - Y = 向前, Z = 向上
 - 保存到: 150_the_render_model
- 在 Meshlab 打开 OBJ!



SteamVR™ !

- 渲染模型存储在以下目录中：
 - C:\Program Files (x86)\Steam\steamapps\common\SteamVR\resources\rendermodels
- 用您的姓名创建一个文件夹
 - 添加 OBJ、MTL 和 PNG
 - OBJ 名称必须与文件夹名称匹配
 - 如果更改其他文件名，注意引用
- 将它放到前面，让我们试试这个文件！
- 定位真实对象时精确度如何？
- 如何更好地调整它？

渲染模型故障排除

- 渲染模型可能未加载
- 在合成器日志文件中查找线索...
 - C:\Program Files (x86)\Steam\logs\vrclient_vrcompositor.txt
 - 搜索您的渲染模型名称
- 超过一个形状？太多顶点？没有 UV 图？

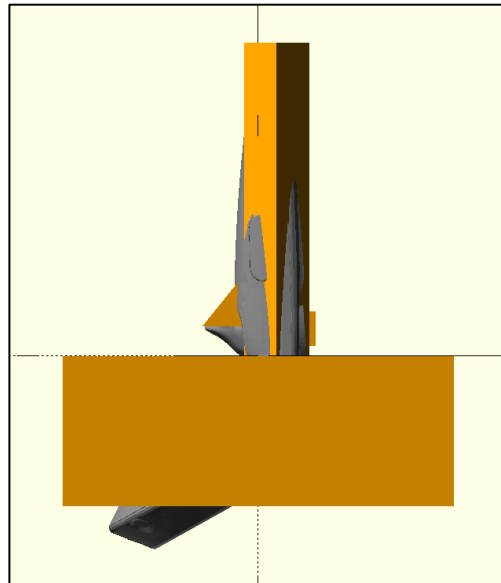
```
<date> - Render model from <rendermodels>\ref_controller\ref_controller.obj  
contained more than one shape. We only support one.
```

```
<date> - Render model <rendermodels>\ref_controller\ref_controller.obj has 313189  
vertices. Only 65k are supported
```

```
<date> - Render model <rendermodels>\ref_controller\ref_controller.obj has 0 texture  
coordinates, expected 82994
```

调入 Head 变量

- 还记得“head”变量有何意义吗？
- 使渲染模型与原始 STL 保持对齐
 - 反转之前的平移和旋转
- 使用“head”变量中的位置
- 将旋转转变为单位向量组件
- 上载新的 JSON 文件
- 渲染模型现在与现实情况的符合程度如何？



总结

- 渲染模型是 SteamVR™ 中显示的图像
- 所有渲染模型都构建于同一坐标系中
- 要求
 - 单个形状
 - 少于 65,000 个顶点
 - OBJ 与 UV 图、MTL、PNG
- 将其部署在 rendermodels 文件夹
- 在 JSON 文件中引用该文件
- 检查 vrclient_vrcompositor.txt 是否有错误
- 与 3D 设计师交谈，了解实际材料的属性和纹理！