

Final project: 参考选题

Computer Graphics Teaching Stuff, Sun Yat-Sen University

这是图形学期末大作业的参考选题，同学们以组为单位选一个题目共同完成。同学们可以根据自己的能力、兴趣爱好选一个题目然后以组内自主分工合作的形式完成。

一、大作业选题

考虑到同学们的基础，我们提供的大作业选题难度不会太高，基本上网上都能找到不少的参考资料和参考代码（特别是[Github](#)）。对于每个题目，除了我们给出的参考实现的功能（不一定要全部实现），同学们可发挥自己的想象力添加一些其他的功能特性（尽可能用本学期所学知识）。我们希望同学们最终做出来的效果**尽可能地美观**，否则打分会大打折扣。

请注意1-8选题人数有限制，自由发挥选题不限。各小组选好之后，请在腾讯文档的组队名单里面填上所选的课题，我们按照先到先得的顺序，若有小组已经选了同一个选题，后面的小组就不能再选这个选题了。我们鼓励大家选择自由选题，实现更多天马行空的创意。请注意，这里要求使用的OpenGL版本是3.3以上的可编程渲染管线版本。

1、用OpenGL搭建并渲染一个场景（名额：2）

在这个选题中，请你使用OpenGL/Vulkan/DirectX3D（任选一）搭建并渲染一个场景（什么场景由小组自行决定，充分发挥你们的想象力），**参考实现的功能**：

- **模型文件加载**：使用Assimp库加载静态模型文件
- **动画模型加载**：使用Assimp库加载骨骼动画模型文件
- **天空盒背景**：使用立方体贴图实现天空盒背景
- **Blinn-Phong光照**：实现Blinn-Phong光照明模型
- **交互摄像机**：通过鼠标的拖动可以实现模型的旋转，鼠标滚轮实现拉近效果
- **实现阴影效果**：使用Shadow Map方法为物体构建一个阴影
- **禁止搭建宇宙等常见场景！**
- **延迟渲染**：在多光源的环境下，延迟渲染性能更优，参考[这里](#)
PBR光照模型：physicall based rendering是一种基于物理的光照模型，这种光照模型更加真实、更加物理准确，参考[这里](#)

.....

参考资料：

[LearnOpenGL CN](#)

2、用OpenGL实现简单的场景编辑器（名额：2）

在这个选题中，你将使用OpenGL/Vulkan/DirectX3D实现一个简单的可交互的场景编辑器，这个场景编辑器主要用于构建自然场景，**参考实现的功能**：

- **加载天空盒**：从用户选定的路径加载天空盒图片作为场景的背景
- **高度图构建地形**：高度图就是一张灰度图，每个像素的灰度值代表高度，你需要读取高度图然后构建地形网格

- **实现鼠标与地形的交互**：鼠标点击向场景中发射一条射线，射线与地形相交，求这个相交点（这里有个小技巧，可以考虑使用二分查找方法）
- **实现物体的放置**：实现了上一条功能之后，你就可以很容地实现将一棵树、一棵草、一个箱子或其他模型的物体通过鼠标点击来摆放到地形上了
- **模型加载**：通过选定模型文件的路径，加载该模型文件，然后使用鼠标点击来放置该模型
- **交互摄像机**：不建议实现fps摄像机，一般场景编辑使用的摄像机类似于第三人称摄像机
- **实现局部光照**：实现传统的Blinn-Phong局部光照明模型
- **也可以制作一个原神千星奇域小游戏！**
- **实现延迟渲染**：延迟渲染着色，把光照计算统一起来，参考[这里](#)

.....

参考资料：

[高度图地形读取与漫游](#)

[射线与地形的碰撞检测](#)

[LearnOpenGL CN](#)

3、实现烟花粒子系统（名额：2）

这个选题主要是关于粒子特效的构建方面，你可以构建一个三维的粒子系统，也可以构建一个二维的粒子系统（使用的渲染库不限，但不得直接套用游戏引擎库的粒子系统，使用游戏引擎的话，粒子系统必须是自己构建的）。在本选题中，你将构建一个烟花的粒子系统，烟花的类型由你自己定，**参考实现的功能**：

- **构建烟花粒子模型**：不要求你使用GPU加速，你只需在CPU上实现烟花粒子系统即可
- **添加天空盒**：添加一个黑夜的天空盒作为背景
- **添加一个地面**：为了使得场景不那么突兀，你只需添加一个平面作为地面
- **实现烟花光照**：为了实现烟花爆炸瞬间的光辉璀璨，你需要在每朵烟花的爆炸中心激活一个点光源，将该点光源的颜色选定为烟花的颜色，爆炸结束后再删除这个点光源。地面需要实现光照（传统Blinn-Phong光照即可），从而凸显烟火闪烁的效果
- **实现辉光特效**：使用高斯模糊之类的后处理技术实现烟花周围的光晕特效，可参考[这个](#)
- **添加音效**：在烟花爆炸的时候添加一个Boom的音效

参考资料：

[Coding Challenge #27: Fireworks!](#)（需要科学上网）

[Particle System](#)

4、构建L-system分形树（名额：1）

L-system是一种简单而有用的分形系统，它常用于树的构建。在这个选题中，你需要实现一个L-system分形系统，构建一颗漂亮的树。**参考实现的功能**：

- **L-system文法**：构建随机三维L-system文法，从一根树干产生分枝，迭代得越多则产生得越多
- **附上纹理**：从将L-system产生的分支构建出树，并加上叶子，给树干、纸条和叶子附上纹理
- **天空盒背景**：场景的背景用立方体贴图表示，并相应地用一个平面表示地面
- **传统光照**：用传统的局部光照模型实现光照效果
- **实现阴影**：使用Shadow map技术为树增加阴影轮廓
- **交互摄像机**：实现第三人称摄像机，摄像机的观察中心是树
- **实现四季更换**：树干的纹理、叶子的纹理可以随着时间流逝而更换，产生一年四季变化的效果（当然冬季可能没有叶子了）
-

参考资料：

[\[OpenGL\] L系统 分形树的实现 \(L-System植物建模\)](#)

[L-系统语法规则构建三维分形树](#)

[opengl L系统 三维分形树 \(有纹理\)](#)

[Tree procedural generation using L-System](#)

5、完善光线追踪渲染器（名额：2）

在Assignment4你们跟着一个教程实现了一个简单的光线追踪渲染器，在这个选题中，你需要对其做进一步的完善。参考实现的功能：

- **光源**：更多的光源类型，例如面光源、点光源、平行光源等；
- **纹理**：为场景物体添加纹理；
- **模型加载**：支持网格模型的加载（例如.obj）；
- **积分器**：使用蒙特卡洛积分方法求解渲染方程求解；
- **多线程**：支持多线程加速渲染计算过程；
-

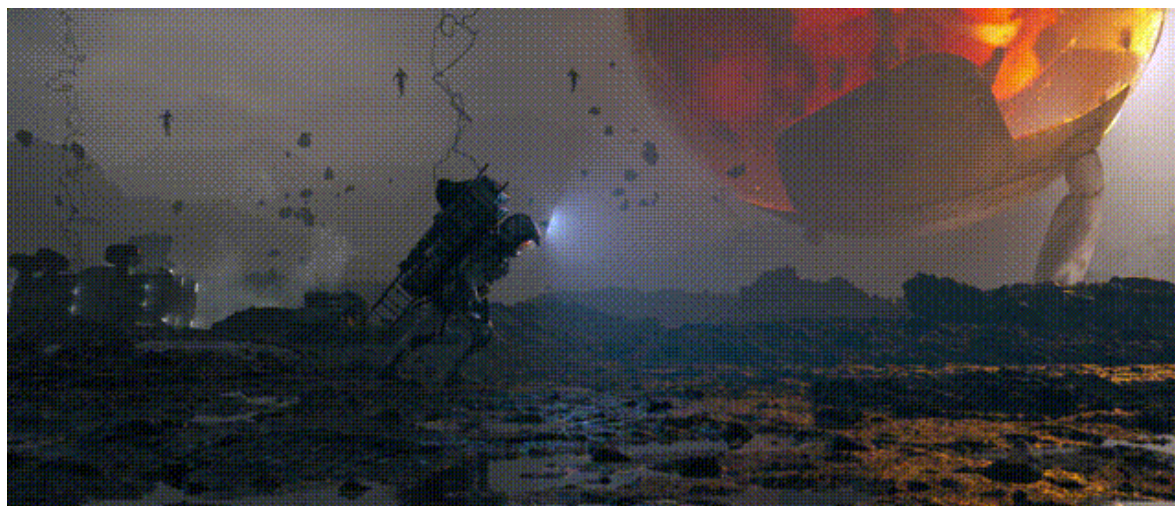
6、玩转ShaderToy（名额：2）

你需要多少个三角形来表示下图所示的场景并进行渲染？答案是仅仅需要两个，这个两个三角形其实构成了恰好覆盖整个屏幕空间的四边形，所有的一切都是在屏幕空间的像素上进行。回顾光线追踪，它也是这样，在每个屏幕空间的像素上发射光线然后与场景中的物体求交然后反射、折射、散射，最后返回该追踪光线得到的辐射率。但这里要使用的是与ray tracing类似的技术——ray marching，即光线步进。场景中的物体采用距离场表示。这里有个[教程](#)详细地教你如何使用距离场表示几何图形，以及如何在片段着色器中实时渲染他们！注意，此项目要求你使用GPU的着色器，你可以使用OpenGL/DirectX/Vulkan来完成，甚至也可以直接在[ShaderToy](#)上完成。此选题难度稍微有点大，实现什么完全靠自身想象力，但ShaderToy上有很多代码可以参考。



7、Unity!（名额：2）

如果觉得ShaderToy太局限，你可以使用Unity来进行更复杂的场景的搭建以及shader的编写，甚至你可以自己去实现一些课程相关的**小动画或者小游戏**！通过Unity，你可以实现一些基础的光照模型，如之前作业中所做的Blinn-Phong光照模型，同样你也可以创作一些光照的衰减与阴影等。此外，你也可以实现一些复杂的纹理效果，包括玻璃效果、水波效果等。总而言之，你可以通过你天马行空的想象力，使用shader更自由地做自己想做的场景以及渲染效果。当然，你做的东西必须要涉及课程所学的内容，并且大部分为自己制作，**我们不能容忍直接从网上直接复制Unity项目**。此选题难度大，请酌情选择。（下图来源于渲染艺术家Jerome Stephan）

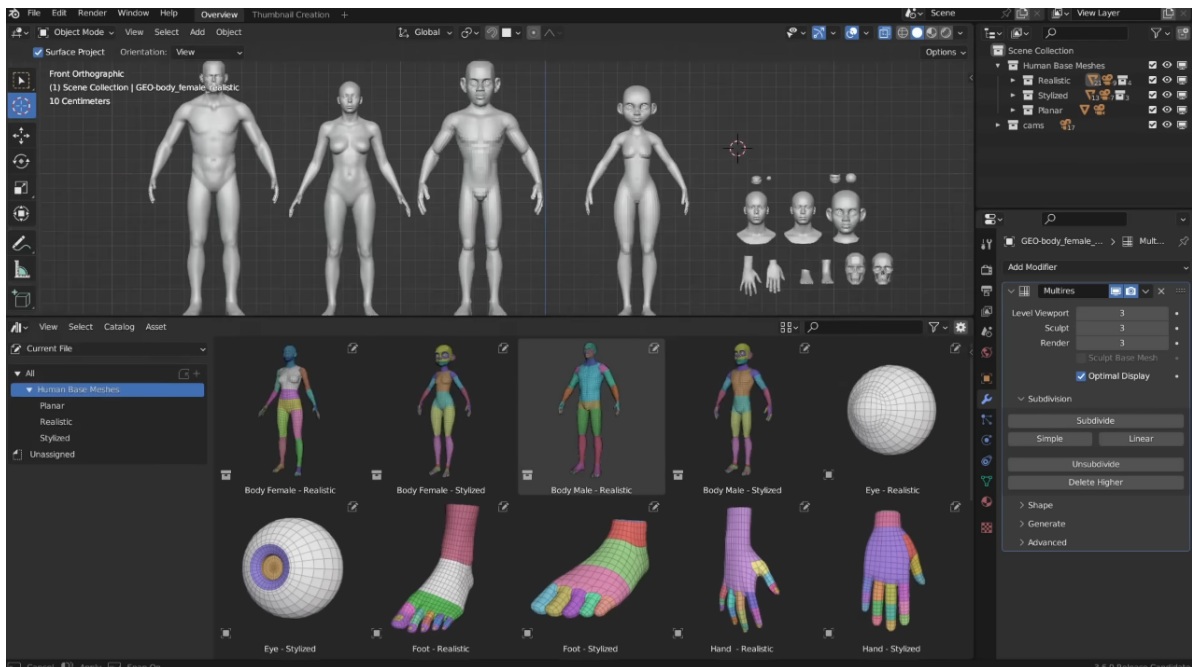


8、三维建模与动画（名额：2）

B站二次创作视频十分热门，许多创作者发挥自己的创意，用自己喜欢的游戏角色、动漫角色等进行了二创。这个选题要求使用Blender、Maya等三维建模软件实现三维物体或三维角色的建模和动画。参考实现的功能：

- **模型构建**：可使用网上公开的三维模型与人体并进行修改，或尝试手动构建。同时搭建出对应场景环境。
- **材质与纹理**：为模型添加合适的材质和纹理，实现真实感或卡通化外观。
- **绑定与动画**：通过骨骼绑定，实现关节、动作、表情的控制和动画。
- **照明与渲染**：实现光照、阴影、动画等效果的渲染。

该选题难度偏大，需参考网上教程。请将实现的重点放在与课程相关的内容上。（下图来源于Blender官方demo教程）



9、自由发挥

觉得上述选题都不太符合预期！

那你们小组可以自己来制定选题，并且自己完成！尽情发挥你们的想象力，设计更多有趣的内容！

但请注意，你们所做的内容必须与课程内容相结合，**要体现课程内容所学**。设计的知识面越广越深，我们的评分也会更高噢！

二、项目展示安排

请注意1-8选题至多人数有限制，自由发挥选题不限。小组选好之后，请在腾讯文档的组队名单里面填上所选的课题，我们按照先到先得的顺序，有2个小组选了同一个选题，后面的小组就不能再选这个选题了。完成之后需要制作展示的ppt，展示ppt主要展示小组完成了哪些基础功能，实现了哪些额外的功能，然后还需要现场演示一下程序运行结果，最后老师和助教会提一些问题，大作业太过敷衍的不会得分。

三、项目提交安排

项目提交包括**两次提交**。

第一次提交时间：所有小组务必在第一次课堂展示的前一天晚上24:00提交（待确定）。该次提交要求**已基本完成整体项目**。提交的内容如下：

- 首先清理一下你的项目，清除编译过程中得到的中间文件，删除不相干的文件，将相关**代码、项目文件、模型**（如网格文件.obj，纹理贴图.png等）放到code目录中；
- 创建一个release目录，将**可执行程序**放到这个目录；
- 同时，请添加一个**README.md**文件或**pdf文档**，写清楚自己完成的项目是什么，介绍如何实现，实现过程遇到的问题和解决过程，每个小组成员完成的内容等；
- 再请附上一些结果图片与**演示视频**，视频不用太长，放到demo目录下；
- 课堂展示**PPT**
- 将以上目录文件打包，组长发到提交作业的邮箱，若文件过大，可同步上传至**github**的public仓库，视频可在文档中提供**B站链接**。

综上所述，你需要提交的压缩文件 小组编号-组长姓名-期末.zip 下应有

- /code目录存放项目文件、代码与模型（如网格文件.obj，纹理贴图.png等），
- /release目录存放可执行文件，
- /README.md作为报告文档，
- /demo目录下存放演示视频和相关的图片，展示PPT，必要的访问链接。

第二次提交时间：暂定第19周周末。该次提交为可选提交，本次提交主要是对第一次提交的**补充和完善**，逾期则视为放弃该次提交。若第一次提交已完成所有项目内容，无需提交第二次。提交的内容、方式与第一次一致。文件名与邮件名为 小组编号-组长姓名-期末v2.zip

最终的评分将会综合两次的提交结果与课堂展示结果。