

# Final project

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。大作业的完成时间预计3周左右。我们对每个课题的难度等级进行了估计，同学们可以根据自己的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、大作业说明

1. 同学们可以选择我们提供的7个题目。

**请注意每个选题至多可被两个小组选择，各小组选好之后，请在腾讯文档里面填上所选的课题，我们按照先到先得的顺序，若有两个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。**

2. 若自行选题，请于2020年11月21日前将题目告知TA，确认选题是否可行。

**请注意，选题内容应与计算机图形学相关，且需体现出一定的工作量。若只是简单地选定一篇论文并运行作者提供的源码做一些简单实验，这是不行的。**

## 二、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，除了我们给出的可实现的特性，同学们可发挥自己的想象力添加一些其他的特性。以下的课题基本涵盖了图形学渲染、模拟、建模的三大话题。在要求实现的特性中，除了视觉效果之外，我们还有一些性能的要求，这是因为在图形学中，视觉效果固然重要，但性能也是不可忽略的一点。

### 1、完善软光栅化渲染器（难度等级：1.5）

在平时作业当中，我们已经给出了一个软光栅化渲染器的基本框架。在这个选题中，你需要在我们提供的框架上做进一步的完善。**当然你也可以不在我们提供的框架上实现软光栅化渲染器，而是参考框架代码自己从头开始写。你必需要完成的功能有如下几点：**

- **线框模式**：除了三角形填充光栅化，请实现线条光栅化
- **更多的光源类型**：除了平行光源，请实现点光源类型
- **可交互的摄像机**：实现一个可交互的摄像机，例如第一人称摄像机或者第三人称摄像机
- **消除背向面**：对于那些正面背朝我们视线的三角形，可以直接剔除而不进行光栅化，这只需做一个点乘就能实现
- **不可见剔除**：完全超出屏幕之外的三角形亦不需要光栅化，你需要在光栅化之前计算三角形的二维包围盒然后判断它是不是完全在屏幕外面
- **多线程优化**：在原本的框架上，所有的算法都是串行执行的。可以考虑使用多线程优化算法效率。我们提供的代码目前耦合度有点高，为了易于并行化，你可能需要其中的算法代码进行解耦

**可选择完成的功能有如下几点：**

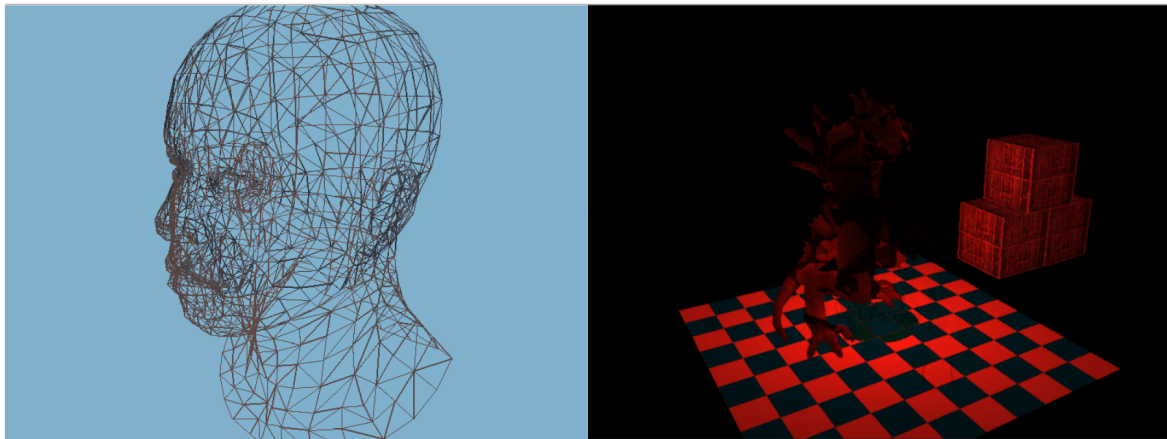
- **实现alpha融合**：实现alpha blending，这需要额外的alpha通道以实现透明效果（例如RGB变成RGBA）；
- **多种模型文件加载**：可以把原来自己写的.obj文件加载类换成Assimp库进行加载，能够加载的模型文件更加广泛
- .....

**参考资料：**

[软渲染器Soft Renderer: 光栅化篇](#)

[软渲染器Soft Renderer: 进击三维篇](#)

[软渲染器Soft Renderer: 光照着色篇 \(完结\)](#)



## 2、完善光线追踪渲染器（难度等级：2.0）

在Assignment 5中，你们跟着教程实现了一个简单的光线追踪渲染器。在这个选题中，你需要在这个作业的基础上做进一步的功能完善。**你必须要完成的功能有如下几点：**

- **添加纹理：**为物体的表面添加纹理
- **添加光源：**目前场景中只有一个背景光，请添加自定义的光源（例如球形光源）
- **渲染三角形：**平时作业中你们只实现了光线与球体的求交，进一步的，你们需要实现光线与给定的三角形求交，从而使之渲染出来
- **加载网格模型：**三角网格模型都是由三角形构成，在实现了渲染三角形的基础上，你们利用Assimp库导入三角网格，然后一个一个三角形渲染，这样就能够实现渲染给定的模型
- **构建BVH树：**为每个物体构建一个AABB包围盒，然后以BVH树的形式把这些包围盒组织起来，加速光线与物体的求交性能
- **多线程加速：**可以很容易看到，不同像素之间的渲染是没有依赖的关系的，可以很容易地并行化

**可选择完成的功能有如下几点：**

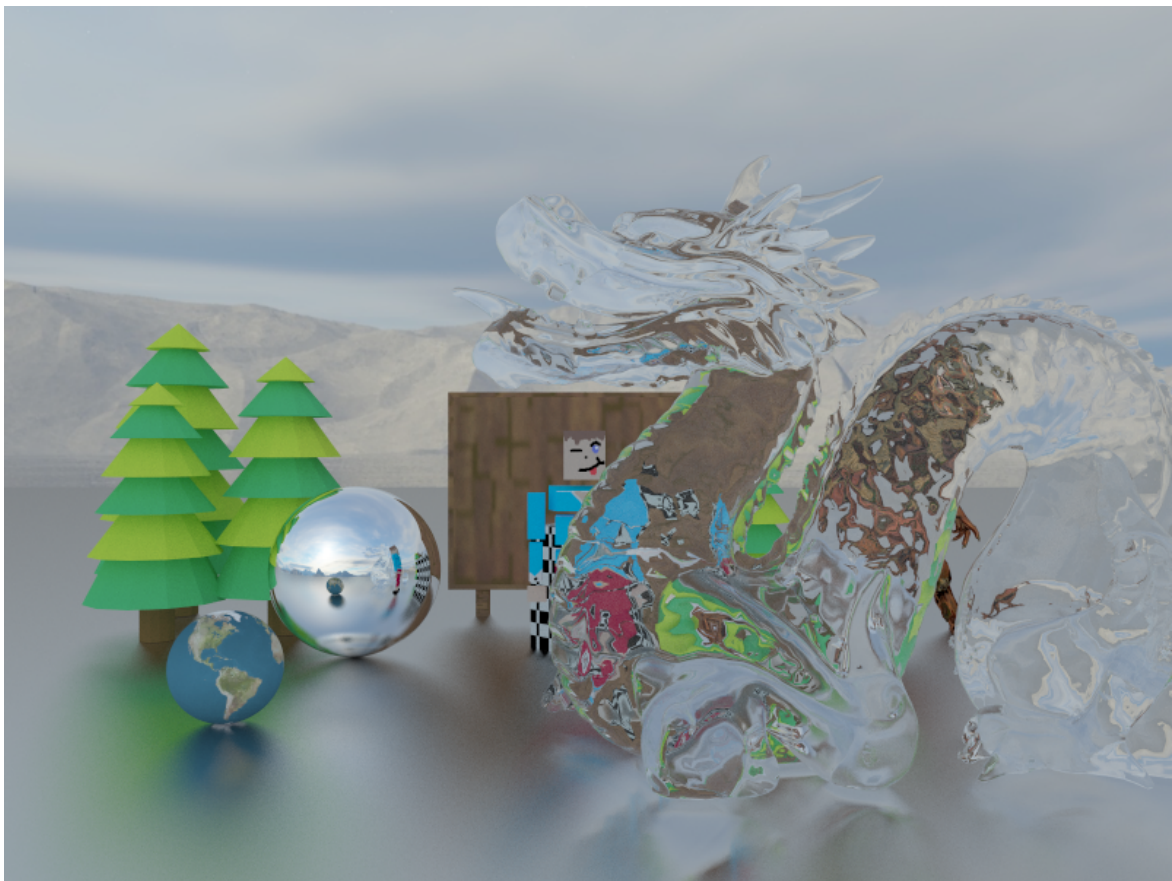
- **添加天空盒背景：**目前的背景光都是手动定义的颜色RGB值，添加一个立方体天空盒，使背景更加真实
- **实现蒙特卡洛积分：**使用蒙特卡洛方法对光线的采样积分，这个理解起来较难，但实现起来不会很难
- .....

**参考资料：**

[Ray Tracing: The Next Week](#)（重点参考bvh实现、纹理）

[光线追踪器Ray Tracer: 进阶篇](#)（重点参考）

[Learn Computer Graphics From Scratch!](#)



### 3、实现模型加载/查看器（难度等级：1.5）

在这个选题中，请你使用OpenGL/Vulkan/DirectX3D（任选一）实现一个模型加载器，你必须要完成的基础功能有如下几点：

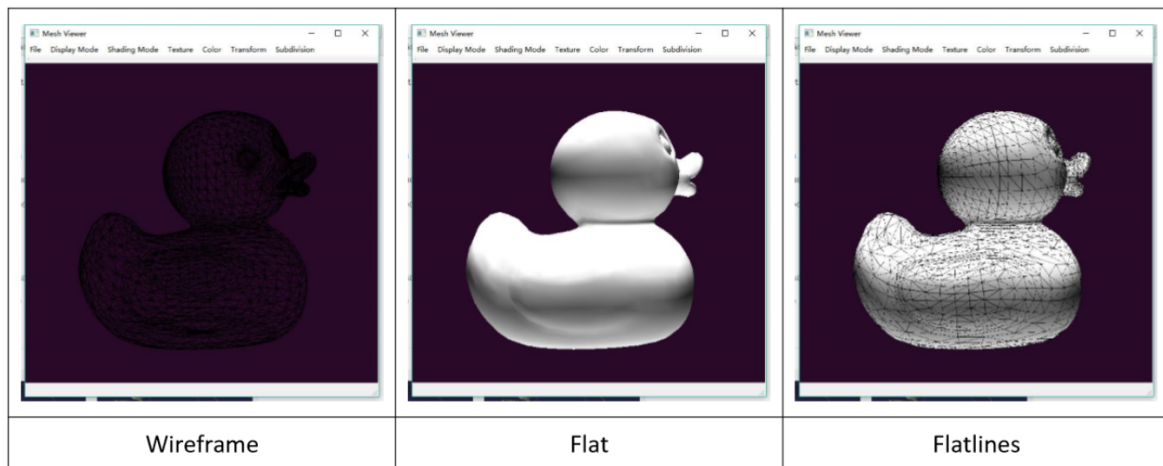
- **模型文件加载**：使用Assimp库加载静态模型文件
- **动画模型加载**：使用Assimp库加载骨骼动画模型文件
- **天空盒背景**：使用立方体贴图实现天空盒背景
- **Blinn-Phong光照**：实现Blinn-Phong光照明模型
- **交互摄像机**：通过鼠标的拖动可以实现模型的旋转，鼠标滚轮实现拉近效果
- **实现阴影**：使用Shadow Map方法为物体构建一个阴影

可选择完成的功能有如下几点：

- **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](http://LearnOpenGL.CN)



## 4、简单的场景编辑器（难度等级：1.5）

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，**你必须实现的基础功能有如下几点：**

- **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

**可选择完成的功能有如下几点：**

- **实现局部光照：**实现传统的Blinn-Phong局部光照明模型
- **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](#)
- .....

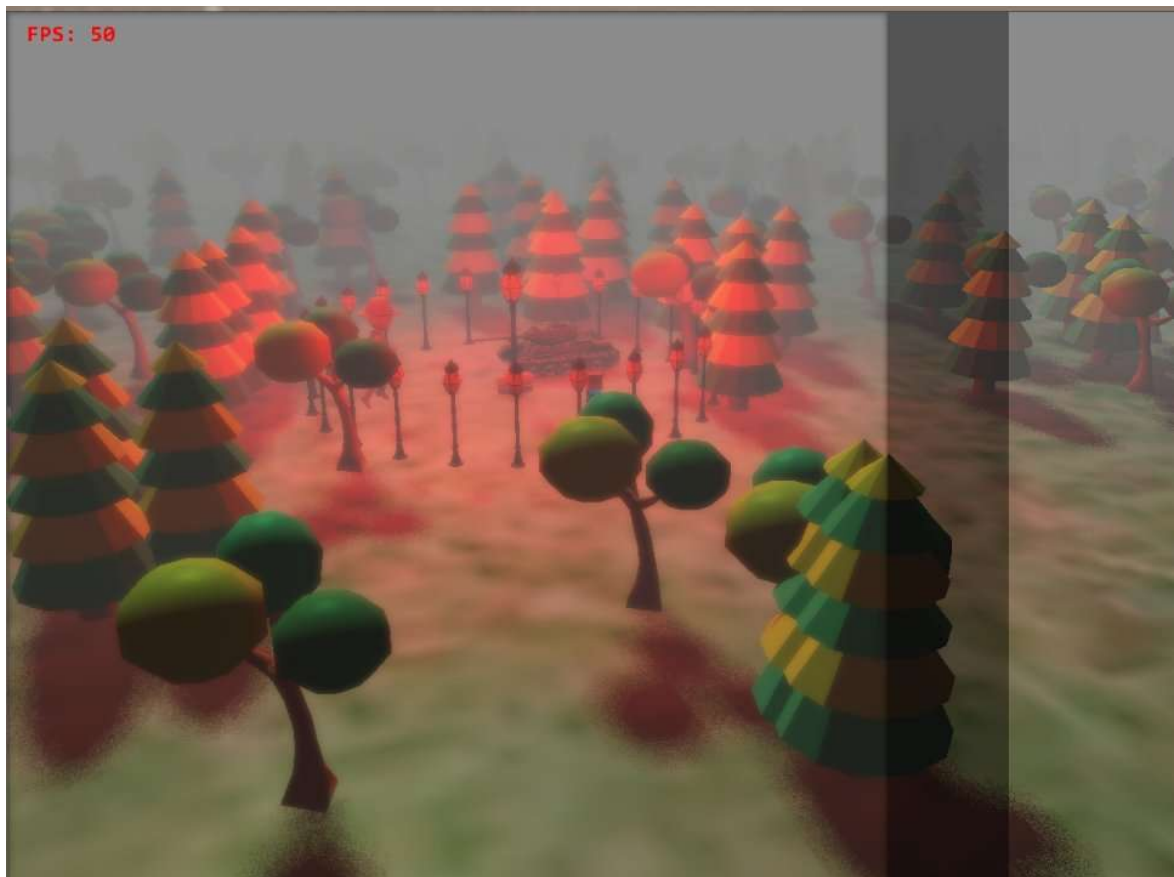
**参考资料：**

[高度图地形读取与漫游](#)

[射线与地形的碰撞检测](#)

[LearnOpenGL CN](#)





## 5、实现烟花粒子系统（难度等级：1.5）

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，**你必须实现的基础功能有如下几点：**

- **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

**可选择完成的功能有如下几点：**

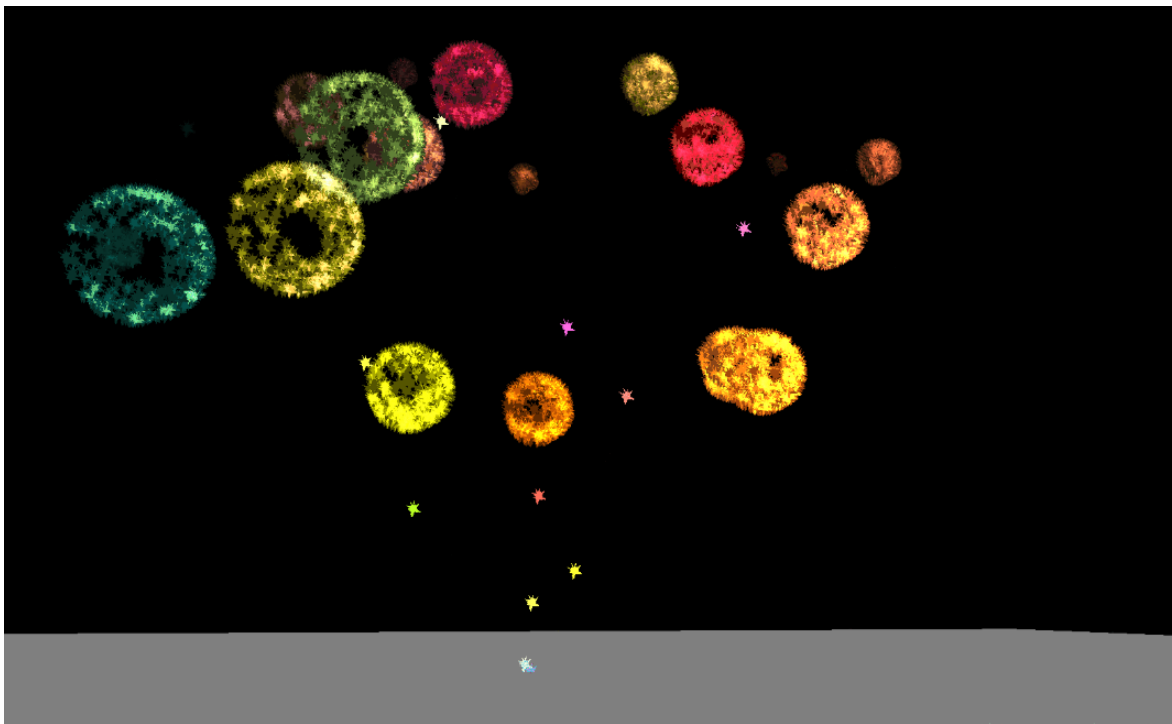
- **实现辉光特效：**使用高斯模糊之类的后处理技术实现烟花周围的光晕特效，可参考[这个](#)
- **添加音效：**在烟花爆炸的时候添加一个Boom的音效

**参考资料：**

[粒子系统--烟花](#)

[Coding Challenge #27: Fireworks!](#)（需要翻墙）

[Particle System](#)



## 6、构建L-system分形树（难度等级：1.5）

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。**你必须实现的基础功能有如下几点：**

- **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

**可选择完成的功能有如下几点：**

- **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](#)



## 7、用两个三角形渲染世界（难度等级：2.5）

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，甚至你可以复现ShaderToy上的效果，只要你理解了其中的算法原理。



## 三、课堂展示内容

1. 实现了哪些基础功能；
2. 实现了哪些额外功能；
3. 现场演示程序运行结果（如光追等离线渲染器就不用演示了，展示相关渲染图片就好）。
4. 回答老师和TA的问题。

我们大作业综合**实现的功能+实现的效果+实现的难度+ppt展示**四项作为最终的评分标准。

## 四、时间节点

---

2020年12月27日 12pm：将展示PPT发到邮箱[zhangzk3@mail2.sysu.edu.cn](mailto:zhangzk3@mail2.sysu.edu.cn)。邮件标题及PPT命名格式：组别\_期末大作业课堂展示PPT。

2020年12月31日：课堂展示。

2021年1月7日：课堂展示。

2020年1月31日 12pm：按下面的要求提交项目材料：

- 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；
- 创建一个demo目录，附上一些结果图片与演示视频，视频不用太长；
- 添加课堂展示的最终版PPT xxxxx.ppt文件；
- 添加一个README.md文件，写清楚自己完成的项目是什么，并简述是如何实现的；
- 将以上的文件一起打包，发送到邮箱[zhangzk3@mail2.sysu.edu.cn](mailto:zhangzk3@mail2.sysu.edu.cn)，邮件标题格式：组别\_期末大作业项目材料。

综上所述，你需要提交的文件下有/code目录存放代码、/release目录存放可执行文件、/demo目录下存放演示视频和相关的图片、/README.md作为报告文档、/xxx.ppt作为课堂展示PPT