

和豆包的对话_0325

课程项目5：实时渲染器 — 第一次汇报稿

(简单明了版，可直接用于课堂汇报)

一、项目背景与目标

大家好，我今天汇报的是**实时渲染器**课程项目的整体规划与算法实现方案。

本项目目标是：

- 实现一个**完整的实时渲染器**，不依赖任何游戏引擎
- 集成核心实时全局光照与后处理技术：阴影映射 (Shadow Mapping)、屏幕空间环境光遮蔽 (SSAO)、屏幕空间反射/全局光照 (SSR/SSGI)，并完成对应的时域降噪
- 保证渲染效果与实时帧率的平衡，完成可交互演示

二、整体项目规划

我们将项目分为三个阶段推进：

1. 基础框架搭建 (第1-2周)

- 选定技术栈：计划使用 **C++ + OpenGL** (或wgpu) 搭建渲染管线
- 实现基础几何渲染、纹理加载、相机交互与窗口系统
- 完成深度缓冲、帧缓冲等核心渲染对象的封装

2. 核心算法实现 (第3-5周)

- 分步实现：Shadow Mapping → SSAO → SSR/SSGI
- 为每个算法编写独立的着色器/计算着色器模块
- 集成时域降噪 (Temporal Anti-Aliasing/Reprojection)，解决SSAO/SSR/SSGI的噪点问题

3. 优化与联调（第6-7周）

- 优化着色器性能、带宽占用，保证实时帧率
- 完成多场景测试，验证光照与阴影效果
- 整理代码与文档，准备最终演示

三、核心算法实现方案 ✨

1. 整体渲染流程

几何输入 → 深度/颜色缓冲 → 阴影映射 → 基础光照 → SSAO → SSR/SSGI → 时域降噪 → 最终输出

2. 阴影映射（Shadow Mapping）

- **实现思路：** 先从光源视角渲染场景得到深度纹理（Shadow Map），再在相机视角渲染时对比深度值，判断像素是否处于阴影中
- **关键步骤：**
 1. 生成光源视角的投影矩阵，渲染深度到纹理
 2. 相机视角片元着色器中采样Shadow Map，进行深度比较
 3. 加入PCF（Percentage Closer Filtering）优化阴影边缘锯齿

3. SSAO（屏幕空间环境光遮蔽）

- **实现思路：** 在观察空间中，对每个像素采样周围半球方向，统计被遮挡比例，模拟环境光遮蔽效果
- **关键步骤：**
 1. 预生成随机采样核与旋转噪声纹理
 2. 从线性深度缓冲重建观察空间位置
 3. 采样周围点，计算遮蔽因子，最后做模糊降噪

4. SSR/SSGI（屏幕空间反射/全局光照）

- **实现思路：**利用屏幕空间的法线、深度信息，沿反射方向做光线步进，找到最近交点，采样颜色作为间接光照
- **关键步骤：**
 1. 从观察空间位置与法线计算反射方向
 2. 屏幕空间光线步进，找到相交像素
 3. 对结果做模糊与边缘处理，避免走样

5. 时域降噪（Temporal Denoising）

- **实现思路：**利用历史帧信息对当前帧结果做累积，降低SSAO/SSR/SSGI的噪点
 - **关键步骤：**
 1. 用前一帧的相机矩阵重投影当前像素，找到历史对应点
 2. 对历史与当前像素做颜色、深度、法线校验，避免重影
 3. 指数平滑累积，得到降噪后的结果
-

四、当前进度与下一步计划

当前已完成：

- 技术栈选型（C++ + OpenGL）
- 基础窗口、相机交互与几何渲染框架
- 深度缓冲与帧缓冲对象的封装

下一步计划（1周内）：

- 实现Shadow Mapping完整流程
 - 完成基础光照与阴影效果验证
 - 开始SSAO模块的编码与调试
-

五、总结

本次汇报主要介绍了：

- 实时渲染器的项目目标与三阶段开发规划
- 从Shadow Mapping到时域降噪的完整算法实现流程
- 清晰可落地的技术选型与开发步骤

后续我们将按计划稳步推进，保证按时完成一个功能完整、效果稳定的实时渲染器。

我的汇报完毕，谢谢大家。

要不要我帮你把这份稿子**精简成1分钟超短版**，或者拆成PPT大纲方便你做幻灯片？

(豆包AI生成)