1 项目配置

使用c++17编写，用cmake(>=3.17)配置

可以在windos10和linux(majaro)上成功编译和运行

windows10使用x64的visual studio编译器，linux上使用gcc-10.2

依赖的第三方库有：

1.SDL2 用于创建窗口、处理鼠标键盘事件

2.glm 一个图形学常用的数学库

3.tiny\_obj\_loader 读取obj格式的文件

2 项目实现内容

本项目实现了一个软光栅器，包括4种不同的z-buffer算法，分别是

1.普通的z-buffer算法

2.层次z-buffer算法

3.层次z-buffer+场景八叉树

4.扫描线z-buffer算法

3 程序使用说明



其中15表示重复加载15个该模型（会自动加偏移不会重叠在一起）

-l表示使用扫描线z-buffer算法

-h表示使用层次z-buffer算法

-o表示使用层次z-buffer+场景八叉树算法

没有该参数则使用普通z-buffer算法

使用w前进，s后退，a左移，d右移，q上移，e下移，鼠标左键点击移动相机视角

4 算法核心设计

5 算法测试结果

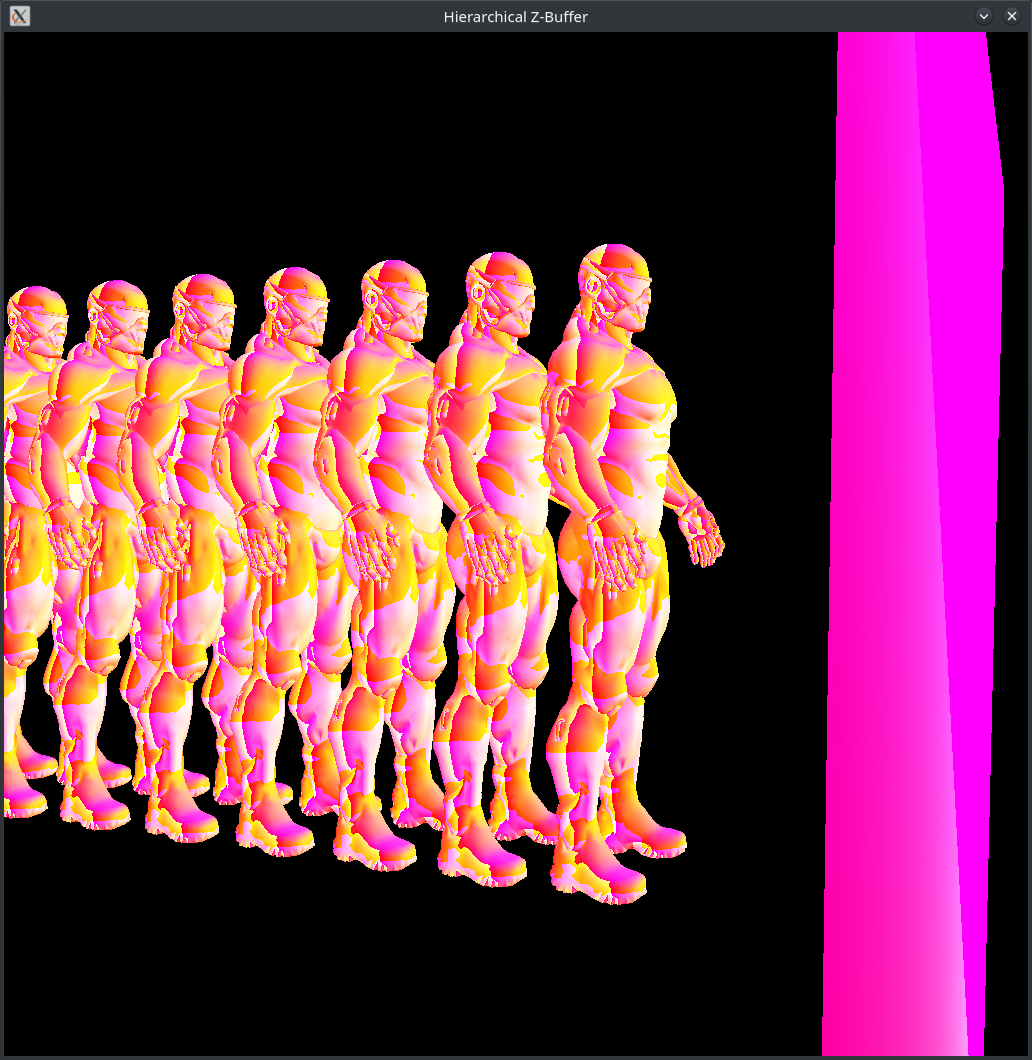
测试使用模型nanosuit.obj，个数为15个，测试场景图如下



(1)普通场景

(2)特殊场景

为了体现层次z-buffer和场景八叉树的加速效果，在15个nanosuit模型与相机前加了一堵墙，侧看场景图如下：



当正视时，由于只能看到墙，后面的nanosuit模型全都被遮挡，可以最大程度体现加速效果。