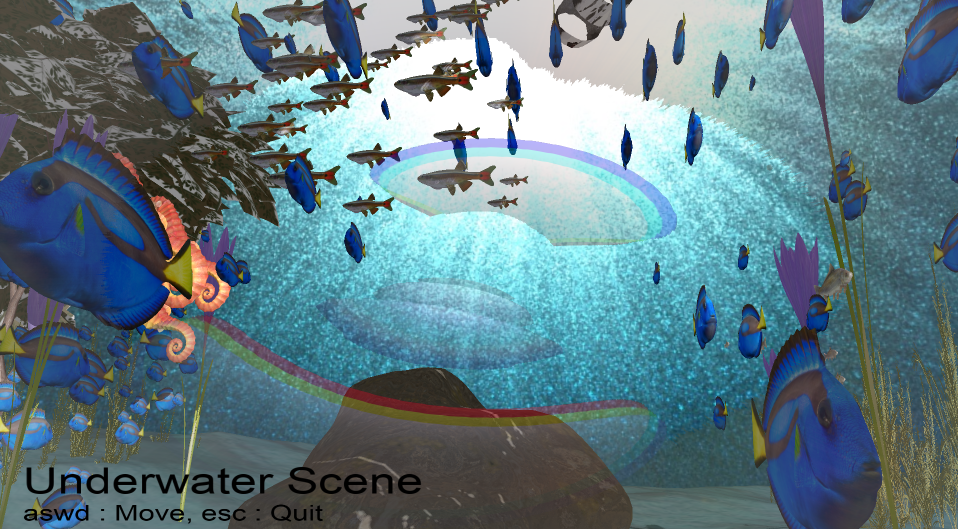
****

实现功能列表

# Basic

1. Camera Roaming

# Bonus

1. Sky Box
2. Display Text
3. Anti-Aliasing
4. Lens Flare

实现功能点介绍

# Camera Roaming

OpenGL 本身没有摄像机的概念，需要通过把场景中的所有物体往相反方向移动的方式来模拟出摄像机，产生一种移动的感觉。用 glfwGetKey 获得按键输入，修改摄像机的位置向量，glfwSetCursorPosCallback 获得鼠标位置，修改摄像机的前向方向向量，将这两者相结合，利用 glm::lookAt 计算出 lookAt 矩阵并传给场景中物体的端点着色器，将它们转换到摄影坐标系中，实现摄像机漫游。

# Sky Box

天空盒是一个包含了整个场景的立方体，它包含周围环境的六个图像，让玩家以为他处在一个比实际大得多的环境当中。天空盒是一个立方体贴图，需要六个面的纹理，还要在片段着色器用 samplerCube 类型的采样器采样立方体贴图。在绘制天空盒时，需要在场景中第一个渲染，并禁用深度写入，使得天空盒绘制在其它物体的背后。还要去掉观察矩阵的位移部分，使得天空盒以玩家为中心，不随玩家位置移动。

# Display Text

使用FreeType库加载一些常用字符的TrueType字体，渲染到位图并生成单颜色纹理和度量值。单颜色纹理在片段着色器中作为 alpha 值与颜色 uniform 混合，使得字形的背景透明而字形是给定的颜色；度量值用于在渲染一行文字时计算每个字的偏移量。再用正射投影渲染到屏幕上。

# Anti-Aliasing

使用OpenGL中内建的多重采样抗锯齿(Multisample Anti-aliasing, MSAA)技术抗锯齿。光栅器将图元的所有顶点作为输入转换为一系列的片段，光栅器不会受到窗口分辨率的影响。



在光栅器中，用这样的屏幕像素的网格来采样，每个像素的中心包含有一个采样点，来决定这个三角形是否遮盖了某个像素，如果采样点没有被三角形内部所遮盖，这个像素就不会生成片段。



这样有些边缘的像素能够被渲染出来，而有些则不会，产生了边缘锯齿的现象。OpenGL中的多重采样将单一的采样点变为多个采样点，以特定图案排列，因此颜色缓冲的大小会随着子采样点的增加而增加。



无论三角形遮盖了多少个子采样点，每个像素只运行一次片段着色器，片段着色器所使用的顶点数据会插值到每个像素的中心，所得到的结果颜色会被储存在每个被遮盖住的子采样点中，当颜色缓冲的子样本被图元的所有颜色填满时，所有的这些颜色将会在每个像素内部平均化。因此上面像素的颜色将会是三角形颜色与其他两个采样点的颜色（无色）的平均值，最终形成一种淡蓝色。



三角形的边缘被稍浅的颜色所包围后，边缘就变得平滑了。

# Lens Flare

镜头眩光是光线经过相机的透镜系统时，一部分光在多个镜片之间经过多次的折射和反射，在照片上形成独特的雾霾和光线效果。在计算机图形学中，模拟这种效果可以增强艺术感和真实感。

实现这种效果需要一系列的帧缓冲和对应的基于屏幕坐标系的端点着色器和片段着色器进行一系列采样。

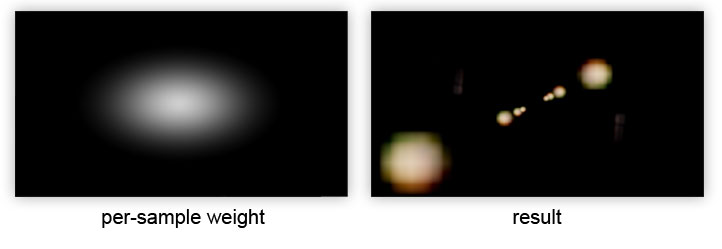
首先在当前的场景中下采样较亮的区域作为光源。



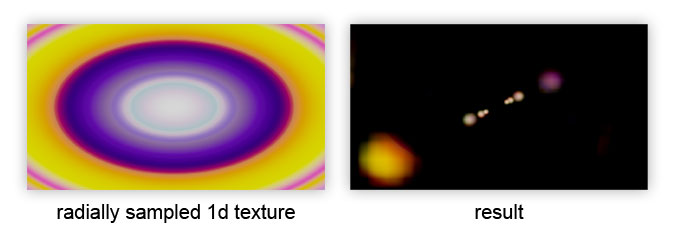
然后在以光源和场景中心形成的直线上对它进行多次偏移的采样，形成多个亮片。



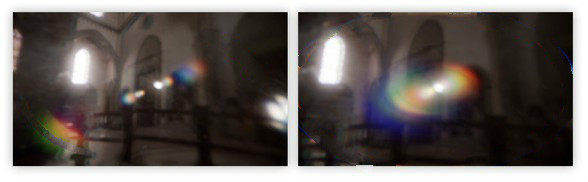
可以让越靠近场景中心的区域的采样权重越大使得效果更加真实。



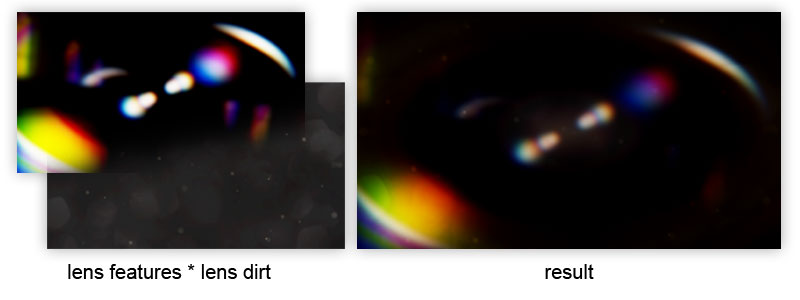
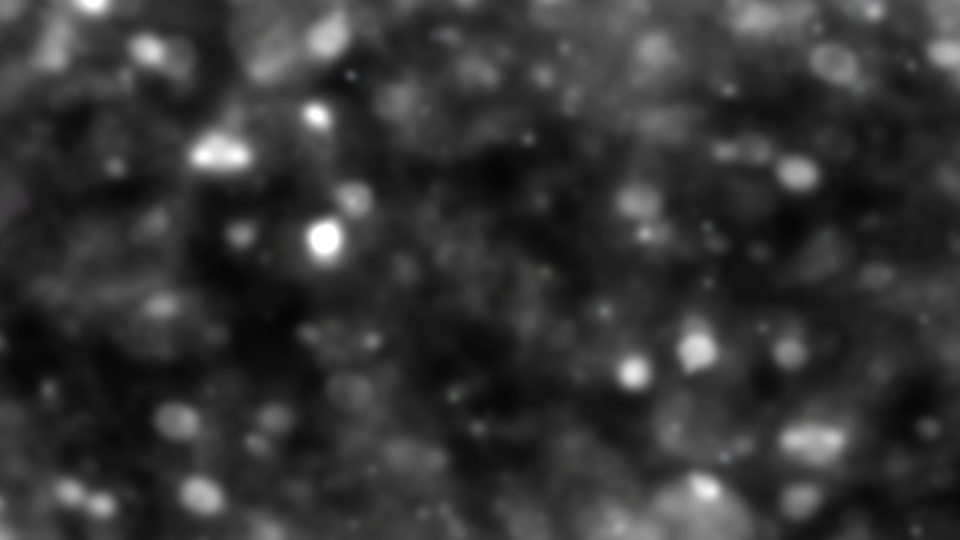
然后贴上一维纹理。



由于光线经过镜片折射会产生色散，可以通过在采样向量中给红绿蓝通道不同的偏移值来实现。



然后贴上一张噪声纹理模拟由镜头上的灰尘导致的光线不干净的现象。



对于一般透镜系统来说，通光孔是光圈结构，由数个叶片组成，难以把通光孔做成标准圆的形状，而是多边形，因此会产生光圈衍射时星芒的现象。



可以贴上一张星芒的纹理来模拟这种现象，并根据摄像机移动时位置的偏移量旋转一定角度来产生动态的效果。



遇到的问题和解决方案

在渲染镜头眩光时只看到眩光，后面的场景都是黑色的。

在片段着色器中眩光背景部分的 alpha 值是 0，为了渲染出不同的透明度级别，需要通过 glEnable(GL\_BLEND); 开启混合，并用 glBlendFunc 设置好混合模式，如 glBlendFunc(GL\_SRC\_ALPHA, GL\_ONE\_MINUS\_SRC\_ALPHA);，渲染出半透明的眩光纹理。这一点在显示文字也有体现。