

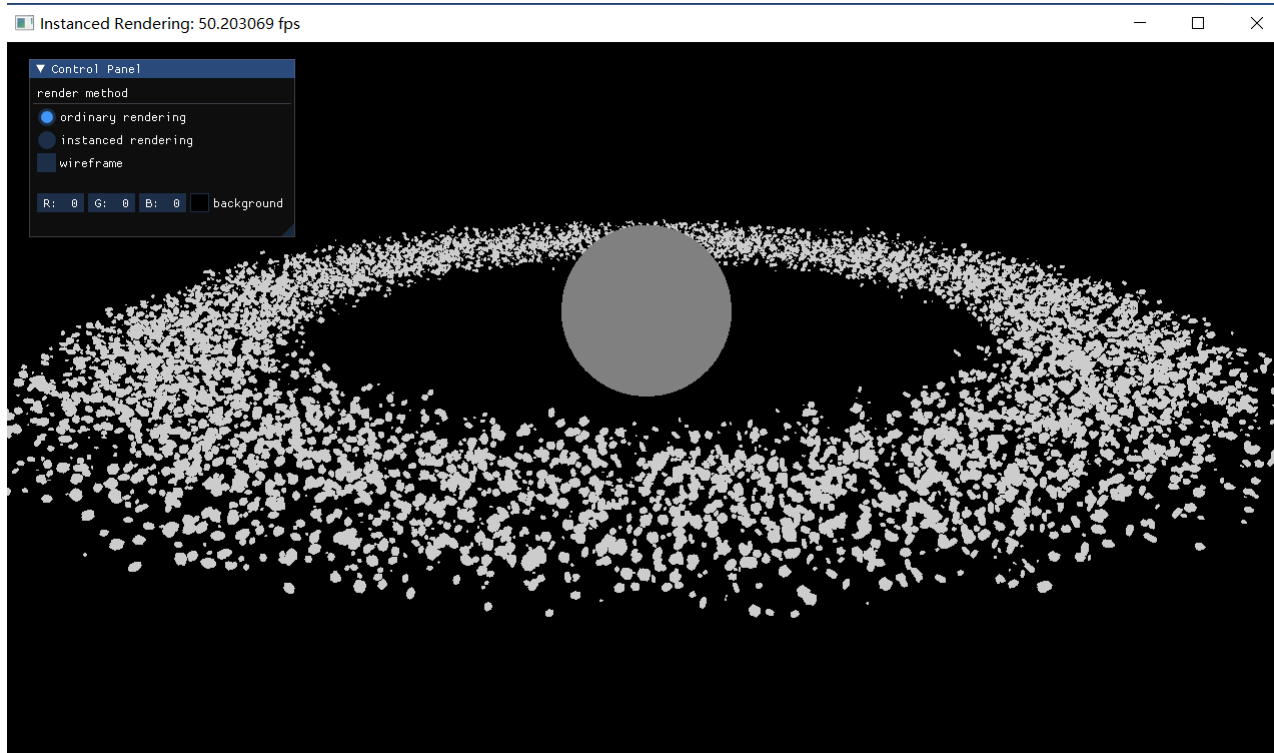
浙江大学实验报告

专业： 计算机科学与技术
姓名： 方彦祺
学号： 3220102829
日期： 2024.12.3
地点： 无

课程名称： 计算机图形学 指导老师： 童若锋 成绩：
实验名称： OpenGL 实例化渲染 实验类型： 基础实验 同组学生姓名： 无

一、实验目的和要求

掌握 OpenGL 中实例化渲染的方法。在此基础上，利用实例化渲染技术提升下列场景绘制的帧率。



二、实验内容和原理

实例化渲染： 在传统渲染中，每个物体通常会独立地进行绘制，每个物体的顶点数据需要多次传输到 GPU。而实例化渲染技术通过一次性传递物体的顶点数据，只需发送一次顶点数据和索引数据，然后通过不同的变换矩阵（如位置、旋转、缩放等）来渲染多个相同的物体。这样可以减少 CPU 到 GPU 的数据传输量，降低绘制多个实例的成本。

三、主要仪器设备

Visual Studio 2022

四、操作方法和实验步骤

实验有两段 TODO 部分需要我们完成，分别完成如下的代码：

```

glGenBuffers(1, &_amp;instanceBuffer);
glBindBuffer(GL_ARRAY_BUFFER, _amp;instanceBuffer);
glBufferData(GL_ARRAY_BUFFER, sizeof(glm::mat4) * _amp;modelMatrices.size(), &_amp;modelMatrices[0], GL_STATIC_DRAW);

glBindVertexArray(_amp;asternoid->getVao());
GLuint mat4AttribLoc = 3; // 顶点着色器中的aInstanceMatrix
for (GLuint i = 0; i < 4; ++i) {
    glVertexAttribPointer(mat4AttribLoc + i, 4, GL_FLOAT, GL_FALSE, sizeof(glm::mat4), (GLvoid*)(sizeof(GLfloat) * 4 * i));
    glEnableVertexAttribArray(mat4AttribLoc + i);
    glVertexAttribDivisor(mat4AttribLoc + i, 1);
}
glBindBuffer(GL_ARRAY_BUFFER, 0);
glBindVertexArray(0);

```

在上述代码中，glGenBuffers 函数创建一个缓冲区对象，用于存储实例化变换矩阵，glBindBuffer 函数绑定缓冲区对象到当前上下文的数组缓冲区，glBufferData 函数为实例化变换矩阵分配内存并将数据传入缓冲区。变换矩阵（glm::mat4）被存储在一个缓冲区中，该缓冲区包含了多个物体（或实例）的变换信息。每个实例都有一个变换矩阵，这些矩阵通过 glVertexAttribPointer 被传递到顶点着色器，计算每个物体的最终变换。

glVertexAttribPointer: 设定如何从缓冲区中获取数据并将其传递给顶点着色器。顶点属性包含变换矩阵的列向量。

glVertexAttribDivisor: 通过设置 glVertexAttribDivisor(mat4AttribLoc + i, 1)，使得每个实例（而非每个顶点）都使用一个新的变换矩阵。也就是说，变换矩阵会在每次实例化时更新，而不是每个顶点都会更新。这使得每个物体能够独立地使用自己的变换矩阵。

glBindBuffer(GL_ARRAY_BUFFER, 0);

glBindVertexArray(0);

这两句代码解除对缓冲区和 VAO 的绑定。

以及：

```

glBindVertexArray(_amp;asternoid->getVao());

glDrawElementsInstanced(GL_TRIANGLES, _amp;asternoid->getVertexCount(), GL_UNSIGNED_INT, 0, _amp;amount);

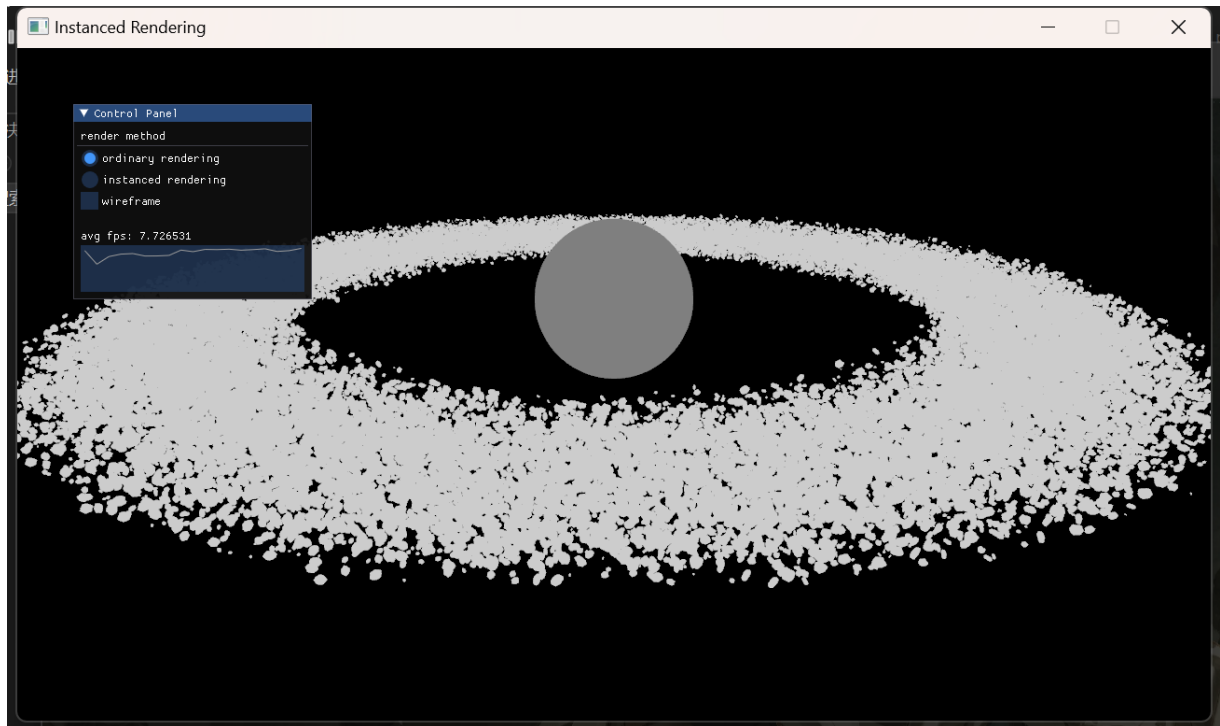
glBindVertexArray(0);

```

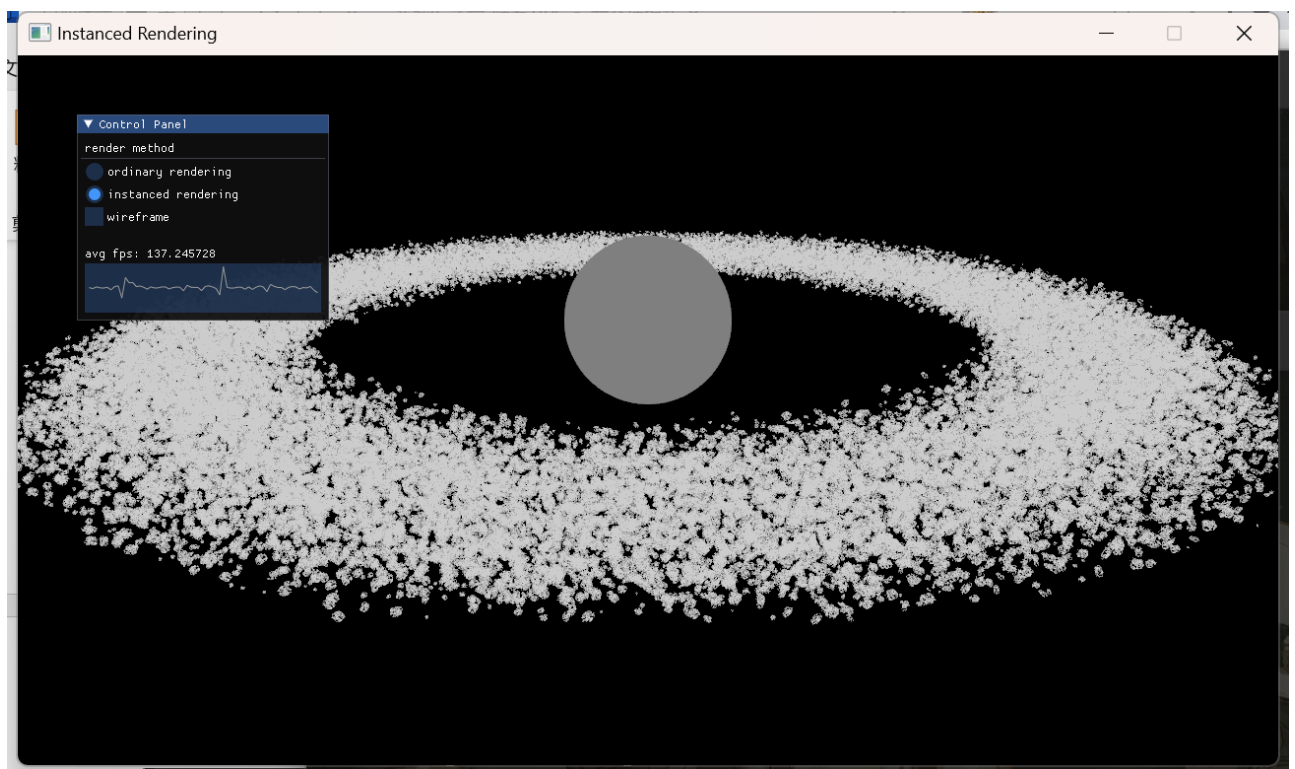
glBindVertexArray 函数绑定 VAO 进行绘制操作，glDrawElementsInstanced 用于绘制多个实例。_amount 参数表示要绘制的实例数量。GL_TRIANGLES 表示绘制的是三角形。0 表示从索引缓冲区的起始位置开始绘制。glBindVertexArray(0);这一句解除对 VAO 的绑定。

五、实验结果与分析

原渲染：



实例化:



可以看到帧数得到了明显的提升。

六、讨论、心得

本次实验需要完成的内容较少，使用参考链接中提供的几个函数进行操作即可。

七、参考链接

[1] [实例化 - LearnOpenGL CN \(learnopengl-cn.github.io\)](https://learnopengl-cn.github.io)