

练习题报告

课程名称 计算机图形学

项目名称 Arcball 的实现

学 院 计算机与软件学院

专 业 软件工程（腾班）

指导教师 熊卫丹

报 告 人 洪子敬 学号 20221550333

一、练习目的

1. 了解和掌握通过四元数进行旋转变换操作。
2. 掌握 Arcball 的实现原理。

二. 练习完成过程及主要代码说明

要求:在实验 2.3 的代码基础上(已编写好 TriMesh.cpp 文件),实现 mouse_pos_callback 函数,在鼠标左键按下时,鼠标的移动将实现 arcball 的旋转功能;在鼠标松开时,鼠标的移动不会触发 arcball 的移动。在实现上,设定了两个旋转矩阵,arcball_pre 与 arcball。Arcball_pre 记录已经发生的旋转过程,arcball 记录正在进行中的旋转过程。完成代码缺失部分。

解答: 首先定义全局变量 LastX、LastY 和 LastZ 用于记录上次鼠标的所在位置,以及 is_dragging 用于判断是否拖拽,mouse_pos_callback 函数的补充步骤如下:

(1) 获取鼠标坐标并转换坐标系,计算当前鼠标三维坐标

函数传入 xpos 和 ypos,利用 glfwGetWindowSize 获取当前窗口的大小,并通过下述变换得到转换坐标系后的鼠标坐标:

$$x = 2 * xpos / width - 1; y = -(2 * ypos) / height - 1;$$

(由于坐标系 y 轴方向相反所以变换后的 y 坐标要加负号)

接着通过勾股定理计算 z 坐标,不过首先要保证其不为负数,所以要多一步条件判断,若其小于 0,则设置 z 坐标为 0;反之则 $z = \sqrt{1 - x * x - y * y}$;最后,用 glm 的三元组将 x、y、z 组装成当前的鼠标坐标;代码如下所示:

```
int width, height;
glfwGetWindowSize(window, &width, &height);
//转换坐标系
double x = 2 * xpos / width - 1;
double y = -(2 * ypos / height - 1);
//计算z值
double z_tmp = 1 - x * x - y * y;
double z = (z_tmp < 0) ? 0 : sqrt(z_tmp);
//当前鼠标位置
glm::vec3 currentPoint(x, y, z);
```

HZJ

(2) 鼠标左键的点击效果实现

首先判断图形是否发生拖拽,若未发生,则将 is_dragging 置为 true,并记录当前鼠标位置为上次鼠标的位置,进入下次判断;若发生,则首先计算上次鼠标位置,此时我们就有了两个向量(lastMousePoint 和 currentPoint),接着是 arcball 的主体实现:

- a. 通过 dot 操作计算两个向量的内积 s;

- 通过 cross 操作计算两个向量的外积 v;
- 通过对内积 s 取 arccos 操作得到旋转角度;
- 通过 angleAxis 函数创建四元数 q 同时 normalize 函数进行归一化;
- 通过 $V = \left(\frac{q \cdot v}{\sqrt{1 - q \cdot s \cdot q \cdot s}} \right)$ 公式计算旋转轴; (注意 0 的 chuli1chu' li)
- 更新 arcball 和 arcball_pre。

上述步骤的代码实现如下所示:

```
if (glfwGetMouseButton(window, GLFW_MOUSE_BUTTON_LEFT) == GLFW_PRESS)
{
    if (!is_dragging) {
        lastX = x;
        lastY = y;
        lastZ = z;
        is_dragging = true;
    }
    else {
        glm::vec3 lastMousePoint(lastX, lastY, lastZ);
        //计算内积
        float dot = glm::dot(lastMousePoint, currentPoint);
        //计算外积
        glm::vec3 cross = glm::cross(lastMousePoint, currentPoint);
        //计算旋转角度
        float angle = acos(dot);
        //创建四元数并归一化
        glm::quat rotation = glm::angleAxis(angle, glm::normalize(cross));

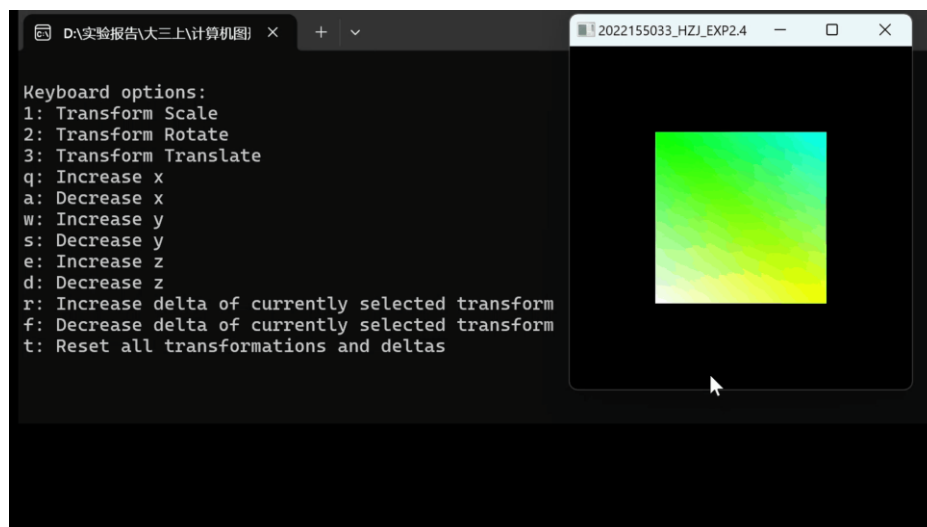
        //计算旋转轴
        glm::vec3 v = glm::normalize(cross);
        float s = rotation.w;
        glm::vec3 axis;
        //防止除以0
        if (1 - s * s > 0) {
            axis = v / sqrt(1 - s * s);
        }
        else {
            axis = glm::vec3(0.0f);
        }
        //更新arcball和arcball_pre
        arcball_pre = arcball;
        arcball = glm::mat4_cast(rotation) * arcball;
    }
}
```

(3) 鼠标右键的效果实现

鼠标右键的代码实现较为简单,只需要设置 is_dragging 为 false 代表停止拖拽即可。

```
else if (glfwGetMouseButton(window, GLFW_MOUSE_BUTTON_LEFT) == GLFW_RELEASE)
{
    is_dragging = false;
}
```

结果展示: 在 display 函数中已经添加好 arcball 作用在变换矩阵上的代码,主函数中也绑定好鼠标响应事件,此时运行程序即可得到理想结果如下:



(转换后的 gif 像素不是很大,详细运行可见提交的 main.exe 文件)