计算机图形学期末大作业

EDWARD RAYMOND HE 518030990022

星号：核心逻辑

斜体：代码与教程参考

一、几何建模及显示

1. 圆柱形**

1.1 几何建模

1.1.1 圆柱形切片构建点

对于一个有 N - 1 个顶点 及 1 个圆心的圆，对此进行X轴上的坐标变换（切片厚度），即可获得 2 N - 2 个顶点 及 2个圆心，并构造一个圆柱形切片。

1.1.2 圆柱形切片渲染

将圆柱形切片分为两块进行渲染：(1) 切片侧面表面，(2) 切片两端圆表面

(1) 通过 2 N 个顶点，通过将圆心作为 N - 1 个三角形公用的顶点，构造圆

(2) 将侧面展开有 2 (N - 1) 个 三角形构造圆柱形切片侧面

1.1.3 圆柱形渲染

将圆柱形切片通过 glm::translate() 函数，在循环中进行渲染（细节将在后续进行说明）。

1.2 纹理

对于同样的加工零件，加工过程中可分为两种材质：

1.2.1 光滑表面（圆柱形两端）

1.2.2 粗糙表面（圆柱形侧面）

1.3 形变

1.3.1 主要数据结构：

1.3.1.1 Slice类

(1) float posA, posB （两者之差绝对值为切片厚度）

(2) float rad （切片半径）

1.3.1.2 Cylinder类

(1) vector slices（同时Cylinder类有slices的个数、半径与厚度）

1.3.1.3 Knife类

(1) vec3 position

(2) float width

1.3.2 形变思路：

以刀尖处于x轴的位置作为中心，2倍切片厚度作为切割范围内的切片将收到波及。

对 slices 中的每个slice，在渲染前调用同样的shader，仅进行半径与位置的变化。

在该程序中，第一层切片的第一个圆所处横坐标 x = 0，因此对往后的每层切片：

(1) 位移：调用 glm::translate(model, glm::vec3(slice_distance, 0.0f, 0.0f)) ，切片仅在x轴上移动，y轴与z轴均不变；

(2) 半径：调用 glm::scale(model, glm::vec3(1.0f, slice_rad, slice_rad)) 进行切片的半径变换，切片大小再x轴上不变，y轴与z轴进行对应放缩。

2. *背景*

   2.1 天空盒[1]

   (1) loadCubemap函数用于加载所有构成天空盒的图片

3. *光照*[2]

4. **粒子系统**

   4.1 Particle类

   粒子运动信息相关数据结构；[3]

   每个Particle都将有生成时的初速度、重力加速度等运动信息，在每次遍历完ParticleSystem中的vector时，每个顶点将根据其运动信息进行一定量的位移。

   (1) 位移：通上述 glm::translate() 等函数进行每个Frame之间运动信息变更的重新渲染与生成；
   (2) 纹理：由于粒子效果多数为原材料的碎片，因此将圆柱形光滑面的纹理映射在粒子块上，以区分并合理化不同材质间的粒子效果。

   3.2 ParticleSystem类

   该类控制切割时：
   (1) 生成Particle的数量，并对每次Particle的生成进行合理且随机的运动信息变量进行赋值；
   (2) 对每个 lifetime 过期的粒子进行重新赋值。

二、交互操作

1. 输入控制
   该程序操作方式由注册回调函数的方式进行刀具的移动、材质切换、视角调节等。
2. 交互曲线
   交互曲线的切割由，与切片同样Array大小的曲线取值集合进行限制，即，
   bezierArray[slices_size]，并以 1/slices_size 作为步长，依次获取对应函数值并存入；随后，在进行切割时，被切割的切片组中，切片半径正好抵达该切片位置对应函数值附近时，则禁止改变其半径大小，并固定切片半径。

三、经验与教训

在尝试以不同形式进行圆柱形渲染时，意识到各类数据结构边界问题，如插入、排序、边界条件判定等，都是对自身编程基础的基本考验。
在此次项目中最大的问题在于GUI的实现未到位，暂时只能以键盘鼠标的形式进行操作。
先前自己喜欢尝试用Unity编辑器进行小型游戏开发，当时的自己对其中的 Particle System还有诸多例如 Prefab、Model、World Space、Camera 等概念处于非常模糊但能意会的边界，经过这次图形学的练手，例如在Shader中对向量的运算和粒子系统等效果的实现加深了自己对图形学的认知！

四、参考网址

[1] https://learnopengl-cn.github.io/04%20Advanced%20OpenGL/06%20Cubemaps/
[2] https://learnopengl-cn.github.io/02%20Lighting/02%20Basic%20Lighting/
[3] https://blog.csdn.net/dcba2014/article/details/52290521