计算机图形学考试大作业报告

519030990009 BRIANHAN 韩亦多

一．题目描述

总体工程需求：有一款中国古风游戏，需要为玩家提供一个自己创意设计 3D 场景的工具。

- 本作业要求提供玩家自己通过编辑轮廓曲线生成茶具（支持建模三才杯、茶杯，桌凳，茶盘）；
- 玩家可选择添加贴图和环境光照，并能将自己设计的茶具模型存储;
- 玩家可将自己设计建造的 3D 模放置在 3D 场景中;
- 系统可提供相机动画实现场景的不同角度观察;
- 系统还能自动在场景中添加粒子特效（例如桃花飞落、落叶飘零）；

评分标准

（1）提供交互编辑界面，供玩家建模并存储模型，提供模型文件的读入和模型渲染；（30 分）

（2）为玩家提供环境贴图、青花瓷贴图、木纹、石纹等的贴图各 2 种以上，供用户选择使用；（15 分）

（3）为玩家提供场景和图形对象光照参数的修改；（15 分）

（4）为玩家提供场景的渲染和相机动画；（10 分）

（5）提供桃花花瓣飘落的粒子系统的特效实现（无需用户设计，渲染场景时自动生成）。（20 分）


图1


图2


图3


图4

二． 项目简介

1. 功能简介

本项目实现了一个在天空盒场景下编辑 bezier 曲线建模茶具的系统。

其中场景可通过按钮切换，提供三套不同的场景贴图。进入场景后默认生成桃花飘落的粒子特效。使用者进入系统中会生成默认的三段式线框（即 bezier 曲线的控制顶点），使用者可编辑线框绘制曲面来生成自己想要的茶具模型。生成后的模型提供了光照和纹理的渲染效果。使用者在操作过程中，可自由移动视角，提供相机动画。同时在建模过程中也可以随时存储和加载指定模型文件达到读入存储效果。

2. 操作键位说明

KEY_A：镜头左移

KEY_B：镜头下移

KEY_W：镜头上移

KEY_S：镜头下移

KEY_M: 激活鼠标光标，固定视角

KEY_F5：隐藏鼠标光标，全方位移动视角

KEY_K: 将目前编辑的模型曲线状态写入文件存储到指定目录

KEY_L: 从指定目录中加载模型文件生成指定模型

KEY_ESC：退出程序

鼠标滚轮：放大缩小视角

鼠标中键：在线框上添加顶点

鼠标左键选中顶点拖拽编辑曲线

鼠标右键选中顶点：删除顶点，两点一线编辑曲线

注意，镜头无法离开天空盒的范围。

3．项目代码简介

使用了OpenGL 的核心模式，并利用了一些外部的库与代码。每一部分的目录下都有对应绘制部分的vs与fs文件，为模型对应的顶点着色器和片段着色器。

项目主要代码文件有

main.cpp：主程序，包括了创建窗口，创建、绘制场景中的物体，接收使用者输入交互执行相应模块功能

camera.h：响应用户鼠标的滚轮和移动事件，转换视角。同时根据绑定的物体的运动状态修改相机的位置与方向。

shader.h：用于定义并编辑构造着色器程序

particle.h: 粒子生成器，根据模式生成不同的粒子效果

snow.h：粒子系统实现，根据主程序需要控制桃花飘落的粒子系统

SkyBox.h：绘制场景天空盒

Button.h：绘制 UI 界面按钮

bezierface.h：定义 bezier 曲面，根据控制顶点生成顶点索引数组和曲面数据，为 Curvemodel 的绘制模型提供接口

Curvemodel.h：实现了通过 bezier 曲面拼接渲染模型，并提供曲线编辑功能修改更新模型状态的功能，程序最关键部分。

三． 功能实现

1、场景：

场景建模采用天空盒，即画一个立方体，然后在六个面贴上相应

面的图片，在内部观察可以看到整个场景连起来的立体的效果。 场景尺寸为30*30*30。程序提供按钮点击可以切换场景图片（即纹理贴图）

2、粒子系统：

进入程序场景后自动渲染。默认效果为以玩家初始视角斜向从左上方中速飘落桃花粒子，飘落过程中通过 ParticleGenerator 类实现。初始化类时，会从文件读取粒子模型、粒子纹理，读取参数以限制粒子的最大数目，确定初速度方向和生命周期等初始化参数，生成系统中的第一批粒子。之后，由 snow 类（粒子系统实现参考下雪的表现方法）实例化 ParticleGenerator，通过运行模式（表现效果）的不同来初始化具体的参数。运行时窗口循环每一帧定时调用 update 方法，并传入距离上一次调用经过的时间作为参数，更新系统状态。

粒子运动轨迹按照抛物线进行物理仿真，使用上一刻的状态计算下一刻的状态以提高仿真的真实性，同时使用中点法计算位移以减小误差。粒子的生命期从产生开始计算，到落地结束。利用随机数对初始速度和角度作一个小的扰动，增加真实感。 在 update 方法中，根据传入的时间间隔，逐一更新所有粒子的状态，删除死亡的粒子，产生一定数量的新粒子。

创建粒子生成器需要以下参数：

粒子的 shader 与 mesh。

粒子系统的初始粒子数目、最大粒子数目、每次 update 复活的粒子数目（grow）及其改变的速度。

粒子生成器的生成的位置及其范围、速度及其范围、角度及其范围、初始颜色及其范围、

结束颜色、大小及其范围、旋转范围与旋转轴。粒子变化基本采用了随机分布，少数采用了高斯分布。

粒子有在与坐标轴平行的矩形平面（或立方体）生成和任意方向矩形生成两种模式。与坐标轴平行实现，只需要在粒子生成器的位置生成之后，增加 xyz 三个方向的随机量。任意方向的矩形，需要宽度、深度两个向量，然后在粒子生成器的位置生成之后，分别在两个方向用随机量偏移，且 xyz 的随机量需要相同。

桃花粒子使用了小型三角片模型，PG 渲染时定义为单个 mesh，贴上了桃花的纹理。根据定义模式的不同设置不同的运动和更新参数。

3、绘制曲线建模：

模型整体通过多个 bezier 曲线旋转得到的对称曲面拼接而成。绘制过程中的属性与方法皆由 Curvemodel 类来实现。

绘制过程中设有 pvalues， tvalues 和 nvalues 来存储模型的顶点坐标，纹理坐标和法线数据，而 fvalues 用于存储控制顶点的坐标数据。

Curvemodel 初始化时会使用默认的一组控制顶点来初始化锚点的坐标数据，并以此调用 makeFaces，通过 bezierface 类来生成 bezier 曲面。Bezierface 类接收 3 个为一组的控制顶点数据，使用 generator 函数来生成曲面。由于茶具模型都为对称形状，所以生成方法是将 3 个顶点生成的二次曲线从上至下旋转一周得到一块 bezier 曲面，生成完后 bezierface 更新他自己的 v、t、n 顶点数组数据，并生成一维的顶点数据索引 indices。这样初始化后，Curvemodel 类便有了一组 bezier 曲面数据。

进入绘制阶段，Curvemodel 需要先将曲面的模型顶点数据加载进一维数组中，这里调用了 load_model 方法，将所有曲面的顶点数据存入了 p、t、nvluaes 中（借用 bezierface 类中的 indices），并返回一个 totalindex，即总的顶点数。接下来将 p、t、nvluaes 与对应的 VBO 绑定，在固定管线中以 glDrawArrays(GL_TRIANGLES, 0, totalindex);拼接三角面片绘制模型。

在绘制模型的同时，Curvemodel也会用load_frame(&fvalues)函数以类似的方法绘制控制顶点线框。用户可以直接对线框上的点进行交互操作，以此更新Curvemodel中的控制顶点数据，以备接下来的重渲染。同时，Curvemodel也提供了各种点修改回调函数：

```
void split_point(int pid);

void remove_point(int pid);

void modify_point(float dx,float dy);
int get_point(float x,float y,Camera camera);

int get_line_start_point(float x,float y,Camera camera);
```

来应对用户对线框的不同操作，以更新线框的状态并存储数据。

当用户更新过线框的点后，主函数中的全局变量 noChange 会被修改，程序会在下一帧的窗口渲染前调用 Curvemodel 的 remake 函数进行控制顶点的更新修改，重新生成曲面数据，从而在下一帧的渲染中绘制出更新后的模型。

4. 模型存储和读入

在上述介绍中可以得知，由于模型的绘制关键数据在于控制顶点数据的变更，因此要做到模型状态的保存关键在于控制顶点的数据存储。

因此在 Curvemodel 的 load/save_file 的实现中主要采用了 fstream 提供的文件读写打开操作。调用存储读入功能时，系统会打开会话框要求用户选择目录，随后将当前状态的顶点数据（即 Curvemodel 中的 vertices）写入到文件中，即完成了模型存储。而模型读入只需要将从文件流中读到的数据进行分割过滤，以此保存到控制顶点数据中即可，剩下绘制操作就交给 Curvemodel 自身和 opengl 管线就行。

5、交互：包含鼠标、键盘两钟输入方式：

场景交互

场景交互：通过鼠标移动，可以改变观察视角；使用鼠标滚轮，可以拉近或远离键，可以拉近或远离视野中的场景。点击线框中的顶点可以进行拖动和增删操作。这些操作的实现主要通过 glfw 中对输入信号的回调绑定实现。

如鼠标移动时系统便会调用预设好的 mouse_fixed_callback 函数，实时传输鼠标坐标，计算差值并调用 camera 类的镜头移动函数；鼠标左键按下时会调用 mouse_button_callback 函数，这时会根据输入键型进行判断，调用对应的点修改函数同时也会对视角移动进行锁定。

普通输入交互

其余的按键交互通过每帧渲染调用的 processInput 函数接收信号实现。其中按钮的点击实现是由于 opengl 没有按钮，使用贴上纹理的矩形面片和对点击区域的判断组合起来实现的。因为按钮的位置不能随着镜头移动，所以按钮的顶点渲染器中没有与 view，projection 等剪裁矩阵相乘，这样它就可以一直显示在画面的右边而不随镜头移动。贴上不同的二维纹理后，就可以表示不同的按钮了。首先用 glfwGetMouseButton 来获取鼠标左键的状态，如果发现它点击了，则修改全局变量 Select 的值，防止重复点击。根据点击区域的不同，Select 的值也不同，从而判断现在需要执行的任务。松开鼠标后，Select 的值会被修改为 NONE，结束点击事件。

6．纹理与光照贴图

这里主要参考了 canvas 上 lightMap 的示例，主要体现在片元着色器的程序编写上。加入了基本的环境光照，镜面光照与漫反射光照的计算。而模型的纹理则提供了石纹，粘土，青花瓷三种不同的纹理。

四． 目标完成度

（1）提供交互编辑界面，供玩家建模并存储模型，提供模型文件的读入和模型渲染；

由于未知原因，预设的模型渲染程序不起作用，生成图形时只能生成线框，无法看到预期效果，所以设计思路与效果只能通过报告与代码赘述，这里非常遗憾。但曲线编辑，存储读入建模与交互界面都完成了实现。

（2）为玩家提供环境贴图、青花瓷贴图、木纹、石纹等的贴图各 2 种以上，供用户选择使用；

提供了三套环境贴图。模型贴图提供了三种。

（3）为玩家提供场景和图形对象光照参数的修改；

暂无环境光照参数修改

（4）为玩家提供场景的渲染和相机动画；

提供了天空盒场景和相机交互动画效果

（5）提供桃花花瓣飘落的粒子系统的特效实现（无需用户设计，渲染场景时自动生成）。

提供了粒子特效的自动渲染，使用纹理贴图和特定参数的修改达到预期的桃花飘落效果。

五、外部辅助与库

1. 使用了助教提供的glfw大生态环境，包括camera, stb_image, model, shader等各种功能完备的库函数；

2. 在原理、代码等方面参考了learnOpenGL，特别是模型渲染的shader和光照部分

3. 代码框架和各种底层实现参考了 canvas 上的 LightMap 示例