

# 报告

2019年11月24日 18:00

## 1. 环境配置

a. Windows10 + Visual Studio 2017

b. 使用的库

i. GLFW

ii. Glad

iii. Glm

iv. Glew

v. Stb\_image

vi. Assimp

## 2. 具体工作

a. 导入并显示炸弹模型

i. 使用Assimp进行模型的导入

ii. 模型从网上下载

iii. 编写了model与mesh模块，前者进行文件（模型与纹理）的读取处理，后者进行模型的渲染

b. 实现引线点燃的效果

i. 实现了粒子模块particle，计算重力，落地有反弹

1) 参考了网上喷泉粒子模型的实现

ii. 更改粒子发射位置来实现移动（实现时移动的是炸弹模型）

c. 实现炸弹爆炸效果

i. 使用几何着色器来对模型的每个三角形进行操作

d. 额外的工作

i. 造了一个位置固定，视角可变的摄像机

ii. 修了一块地板

1) 原来的想法是爆炸后把模型藏到地板下面，后来发现了让模型直接粉碎的办法

## 3. 技术方案

a. 整体架构

i. 模型显示与渲染

1) Mesh

a) 处理并渲染网格

2) Model

a) 读取模型

3) Shader

a) 包装了着色器

4) Texture

- a) 导入纹理
- ii. 视角
  - 1) Camera
    - a) 可以进行视角的变动
- iii. 地板
  - 1) Floor
- b. 如何渲染模型
  - i. 将模型文件中的顶点读到VAO中
  - ii. 使用glDrawElements按顺序画三角形
- c. 如何构造粒子系统
  - i. 单个粒子
    - 1) 属性
      - a) 位置、速度、大小、年龄、类型
    - 2) 分类
      - a) 发射器粒子
        - i) 这些粒子是不会动的
        - ii) 产生粒子时, 这些粒子会"分裂", 新粒子会射出去, 并重置原粒子的寿命
      - b) 空中的粒子
      - c) 落地的粒子
        - i) 落地时会根据速度产生一个弹起来的粒子
- d. 粒子系统
  - 1) 初始化
    - a) 在原点附近的球内随机生成一定数量的粒子(发射器粒子), 为他们申请GPU内存空间
  - 2) 更新
    - a) 使用几何着色器随机在发射器粒子处产生新的粒子
    - b) 为新粒子随机生成属性
  - 3) 删除
    - a) 删除寿命不足的粒子
    - b) 删除越界的粒子
  - 4) 绘制
    - a) 使用glDrawTransformFeedback来绘制粒子
- e. 引线移动
  - i. 简单的改变了炸弹的位置
- f. 炸弹爆炸
  - i. 在几何着色器中
    - 1) 使用叉乘来获得三角形的法向量
    - 2) 根据法向量确定三角形的运动方向
    - 3) 三角形的当前位置+运动方向\*时间参数得到下一次的位置

#### 4. 如何操作

##### a. RELEASE运行

- i. 直接在运行 REALEASE/exe/Bomb.exe , 相关依赖已在相同路径下

##### b. VS工程运行

- i. 使用VS打开Code/Bomb/Bomb.sln

#### 5. 反思

- a. 同往常一样, 这次作业参考了大量资料, 深深发现自己有太多不会的东西了。以至于到现在才意识到, 自己并没有使用传统的固定管线编程, 这让自己在寻找教程与跟同学交流的过程中产生了一定的困扰。但学到了新东西还是很开心的。

#### 6. 参考资料

##### a. 粒子系统的实现

- i. [https://blog.csdn.net/qq\\_31615919/article/details/78961229](https://blog.csdn.net/qq_31615919/article/details/78961229)
- ii. <http://wiki.jikexueyuan.com/project/modern-opengl-tutorial/tutorial28.html>

##### b. 着色器的使用

- i. <https://learnopengl-cn.github.io/04%20Advanced%20OpenGL/09%20Geometry%20Shader/>

##### c. opengl教程网站

- i. <http://www.opengl-tutorial.org/cn/intermediate-tutorials/billboards-particles/particles-instancing/>
- ii. <http://ogldev.atspace.co.uk/>
- iii. <https://learnopengl-cn.github.io/>