Projection report for the CG 100433 course

Project Title

SmashHIT aka. S-HIT

Team member

胡轩 刘子康 罗嘉胤 孙惠宇

Abstract

该项目基于一款移动端游戏 Smash Hit，意在运用所学的图形学知识实现游戏的基本逻辑、场景渲染、碰撞检测和碰撞效果等方面，最终实现了一套完整的场景渲染引擎和碰撞检测。该项目用到的图形学技术有图形变换、天空盒渲染和 Blinn-Phong Shading 等。

Motivation

1. 小组成员对 Smash Hit 这款游戏的规则和机制较为了解；
2. Smash Hit 是一款有趣的游戏，小组成员有兴趣就此进行研究和开发；
3. Smash Hit 中包含了全面的图形变换，例如平移、旋转、缩放等，可以较好地把课上学习的知识运用到实际的开发中；
4. Smash Hit 中也有一些比较复杂的效果，例如碰撞检测、破碎效果等，在实现基础功能之后，小组成员能够针对复杂效果进行探索研究并尝试实现它们。

The Goal of the project

1. 实现地图的设计和基本渲染；
2. 实现物体的组合和变换；
3. 在基本渲染的基础上加上天空盒、折射、反射、Blinn-Phong Shading 等更高级的渲染；
4. 实现碰撞检测；
5. 如果前面的目标顺利实现，尝试实现几何体破碎效果、物体材质区别以及多光源渲染等效果。

The Scope of the project

时间所限，本项目的游戏成分不多。主要是作为一个较为简陋的游戏渲染引擎，提供了基本几何体的组合、空间变换、渲染、碰撞效果和高级光照等。

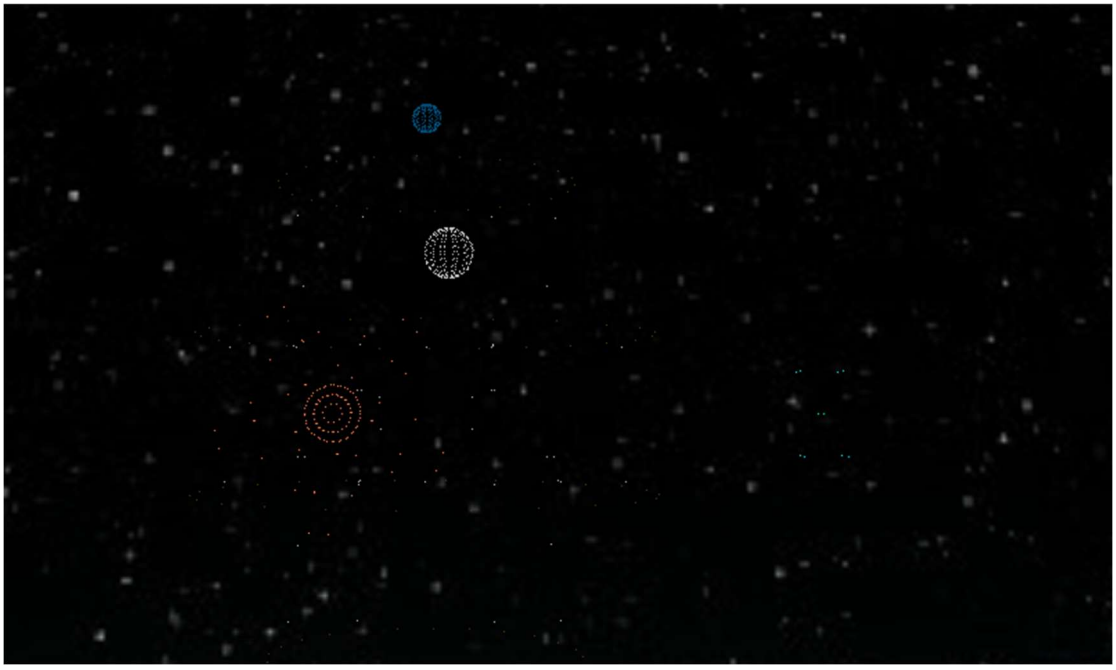
Involved CG techniques

1. 图形变换。对一个物体进行旋转、缩放、平移以及它们的复合变换。
2. 天空盒渲染。利用立方体贴图绘制天空盒，并用天空盒作为物体折射盒反射的纹理。
3. Blinn-Phong Shading. 在 Phong Shading 的基础上进行了一定的改进，计算镜面反射时用法向量点乘半程向量，免去了 Phong Shading 中反射向量的计算，也提升了镜面反射的效果。
4. 碰撞检测。使用 AABB 碰撞盒包围物体，在每一帧时都对碰撞盒进行 Scale 和 Translate 操作，但是不能随物体旋转而旋转。
5. 几何着色器。传入一个距离值，让每个三角形向法线方向移动。(由于代码遗失，只保留了点阵爆炸的效果）组员提出的思想是三角形向物体中心到三角形中心的方向移动，而

不是向法线方向移动。这样爆炸效果更自然。此外在爆炸开始时，关闭物体的旋转，否则物体将较为奇怪。

Project contents

该项目最终实现了目标 1-4，并对目标 5 中的破碎效果、法线贴图（物体材质区别）和多光源渲染进行了研究。由于法线贴图和多光源渲染在该项目中作用不大，因此没有实现；由于 GitHub 上 merge 时覆盖掉了原有的爆炸部分的代码（几何着色器也部分遗失），此部分未能在最后进行展示。



Implementation

Figure

Figure 类及其子类 FigureCuboid, FigurePyramid, FigureSphere, FigureCylinder 实现了基本图形正方体、四棱锥、球和圆柱体的模型构建和基本渲染。

渲染的流程如下：首先，初始化 VAO, VBO 和 EBO（如有必要）；然后，计算各顶点、法线（如有必要）、纹理（如有必要）的坐标数据，计算 EBO 的值，并将它们绑定至第一部中的初始化过的相应对象上；最后，调用 render()函数以线框模式绘制基本图形。

Object

Object 类及其子类 Ball, Pyramid, Cylinder, Cuboid, Fan, Diamond 等构成了在游戏场景中运动或静止的所有物体。Object 类的设计参考了 Unity3d 中有关物体组合及其空间变换的实现方法。

物体组合方面。mainFigure 成员作为当前物体实际拥有的基本 Figure，subobjv 成员作为附着的所有子物体的集合，其中包含其他 Object，fobj 成员则指向父物体。

空间变换方面。若没有父物体，世界矩阵属性用于确定物体在世界坐标系中的位置和姿态，速度、加速度、转轴和加速度属性确定物体的运动状态。若有父物体，相对矩阵属性用于确定子物体相对父物体的空间位置和姿态，速度、加速度、转轴和加速度属性则确定物体相对于父物体的运动状态。以 Fan 为例，Fan 类的 mainFigure 为空，自身不维护 Figure，子物体包括可变个数的立方体（扇叶）和一个圆柱，立方体和圆柱分别设定相应的相对矩阵属性维护相对 Fan 对象的位置。空间变换时，只需对父物体做平移和旋转变换，子物体的世界

矩阵由相对矩阵和父物体的世界矩阵相乘即可求得。

Shader

该项目含两个顶点着色器和两个片段着色器，第一种是用于渲染天空盒的，第二种用于渲染一般的物体。用于天空盒的着色器很简单，只需要在基础着色器的基础上绑定纹理即可。用于一般物体的顶点着色器也比较简单，只负责把顶点位置、法线、纹理坐标传入片段着色器。片段着色器中设置了多个 uniform 用来在 Object 类中赋值以实现不同的渲染效果。能够实现的渲染效果有：着色、反射和（或）折射天空盒的纹理以及应用 Blinn-Phong Shading 实现方向光照效果的渲染。Bool 型 uniform 用来设定是否实现对应效果，而其他的 uniform 则负责设定各效果的细节，例如光照强度、方向、反射和折射效果的比例等。

Collision

该部分，首先使用我自创的算法对多面体进行碰撞检测，检测多面体的每一个面和球之间的距离，但该算法效率太低，哪怕使用卡马克的快速平方根倒数算法也依然卡顿明显，故放弃。

最终，使用 AABB 碰撞盒，该碰撞盒只需要检测 min 和 max 的 xyz 坐标即可，实现较为简单，效果不差。

Results

实现效果见附件中的视频文件。

Roles in group

胡轩：代码框架构建，物体组合、空间变换和底层渲染引擎实现。
刘子康：代码框架构建，碰撞检测实现，破碎效果研究。
罗嘉胤：高级渲染实现，底层渲染引擎扩充，法线贴图、多光源渲染研究。
孙惠宇：点击抛球机制实现，地图设计，游戏逻辑实现。

References

https://learnopengl-cn.github.io/
http://www.math.ucsd.edu/~sbuss/MathCG2/OpenGLsoft/