

Final Project Report

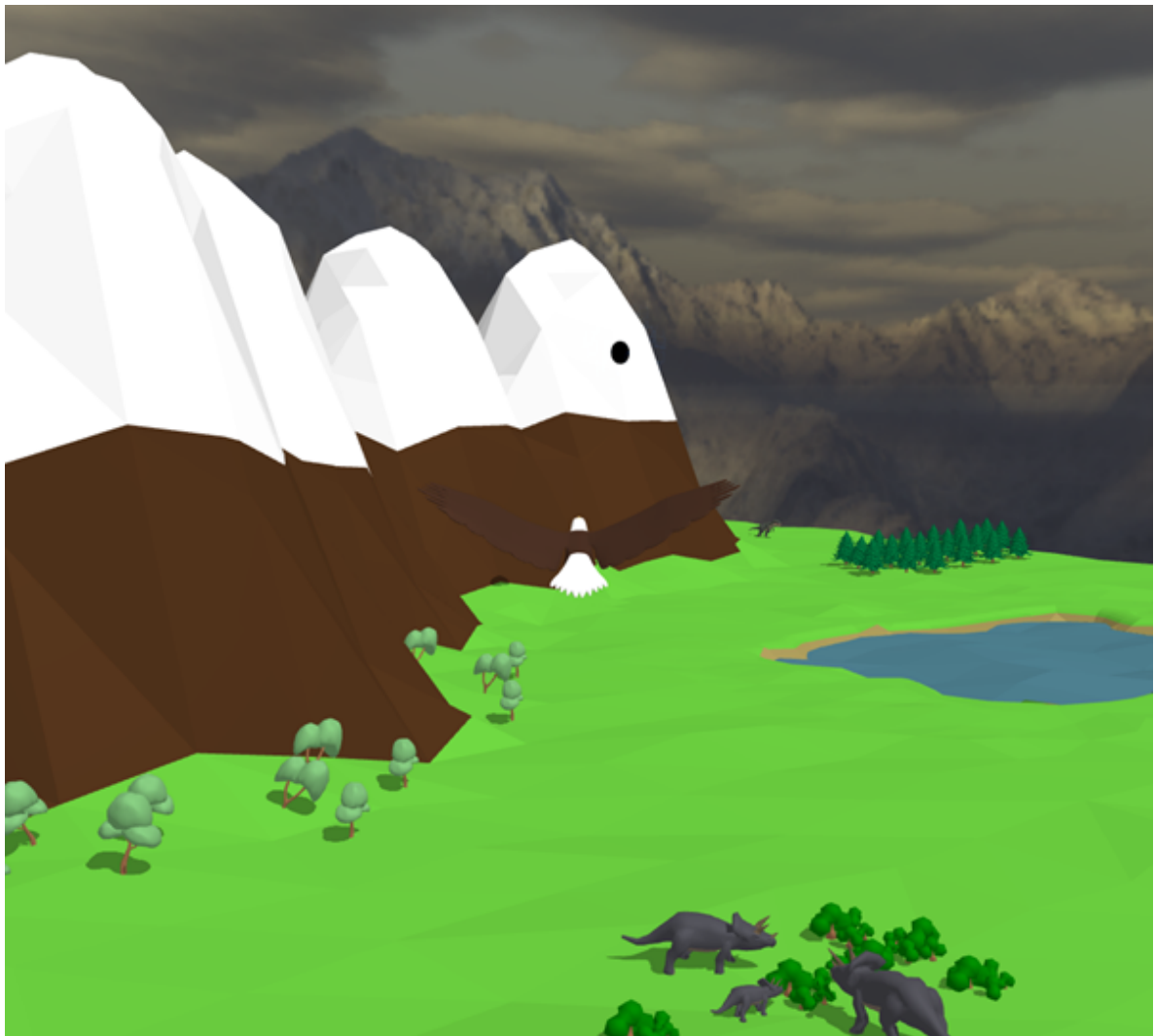
项目介绍

该项目受老师上课展示的“春夏秋冬”以及霍金的纪录片所启发，是一个时空穿梭题材的场景漫游项目。
该项目包含2个场景：过去和现在。

虫洞猜想：生活中充满了黑洞，但它们太小了以至于我们人眼看不见。霍金的一部纪录片表明，通过虫洞，我们可以穿越时空，回到过去，或者进入未来。

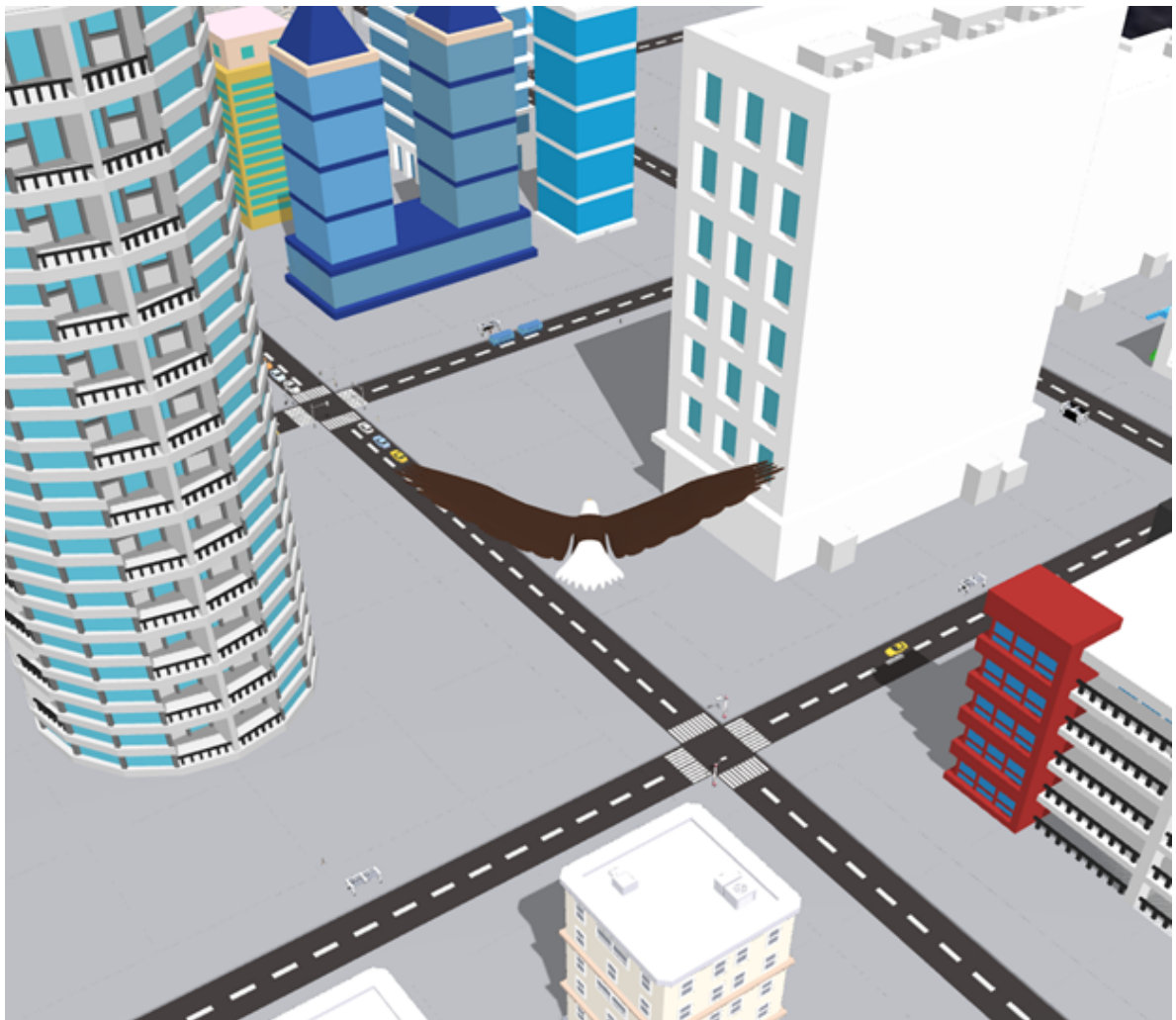
过去

“过去”的场景中，我们用恐龙作为场景中的标志物。



现在

“现在”的场景中，我们选择城市森林作为主场景。



开发环境以及使用到的第三方库

1. 开发环境: *windows10, Microsoft Visual Studio Community 2017*
2. 使用到的第三方库:
 - `glfw3`: 提供创建窗口、上下文等功能
 - `imgui`: 提供GUI界面
 - `assimp`: 加载模型

实现功能列表

Basic

1. Camera Roaming
2. Simple lighting and shading
3. Texture mapping (与天空盒结合)
4. Shadow mapping
5. Model import & Mesh viewing

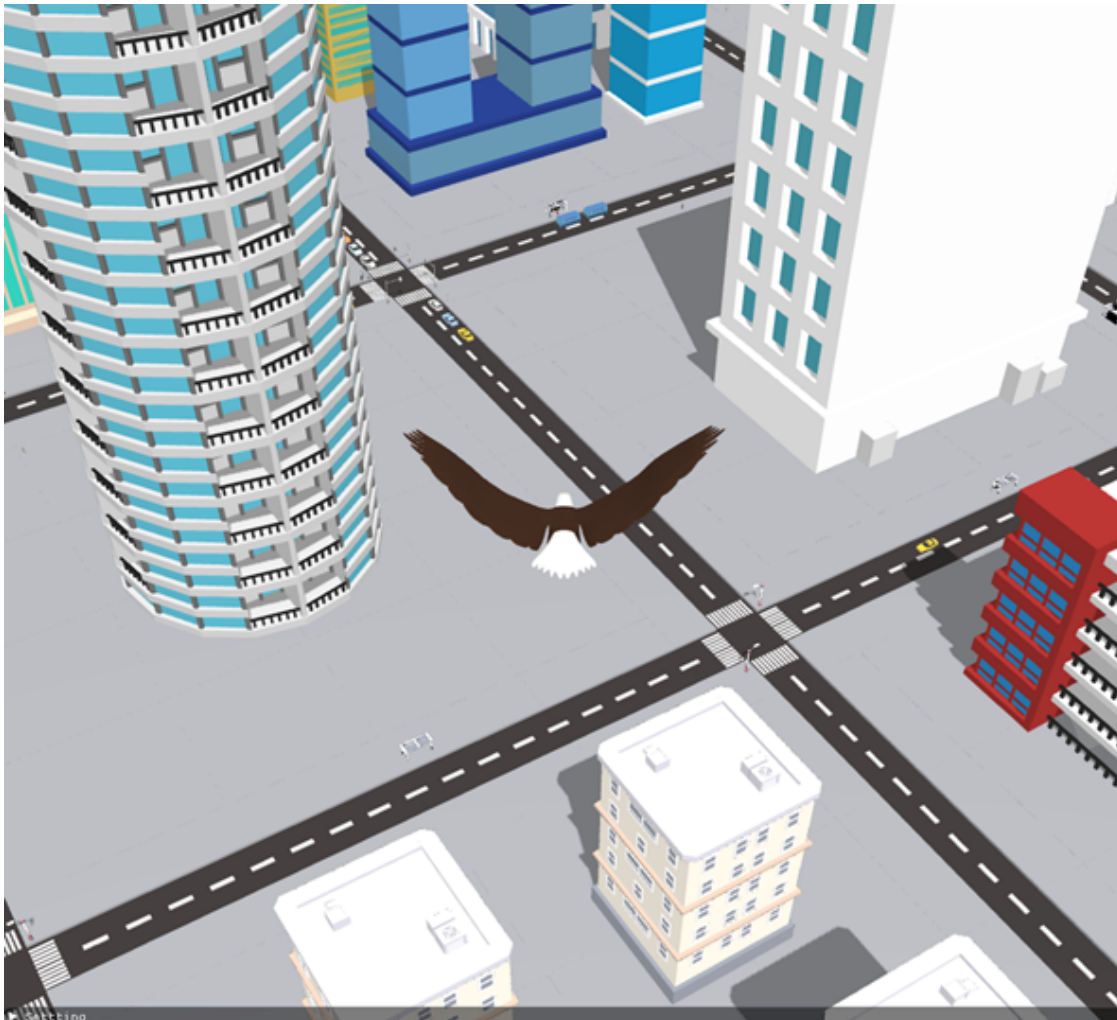
Bonus

1. Sky Box
2. Skeletal Animation
3. Gamma Correction
4. Anti-Aliasing
5. Display Text

功能点介绍

Basic

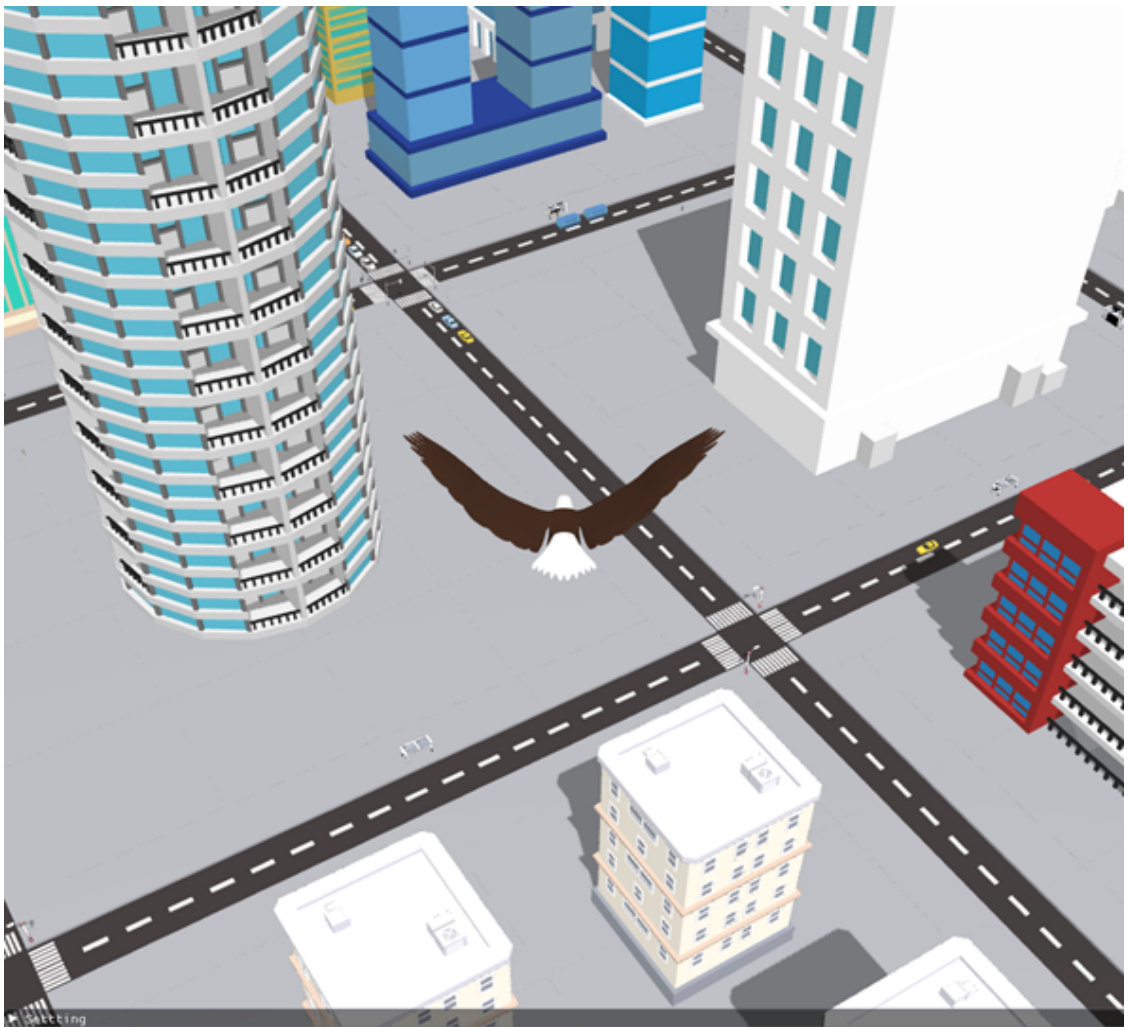
1. Camera Roaming: 将带有骨骼动画的鹰放置于摄像头的前方，这样就可以实现第三人称视角，同时鹰在摄像头移动过程中，会有些许角度偏转，达到更真实的效果。



2. Simple Lighting: 使用 Phong 光照模型，但是不论怎么调光照颜色都很难达到我们想要的效果，后面我们使用伽马矫正修正了它，下面是未修复时的效果。



3. Shadow Mapping: 阴影部分我们使用了 Shadow Mapping, 与之前 LearnOpenGL 的教程一致。

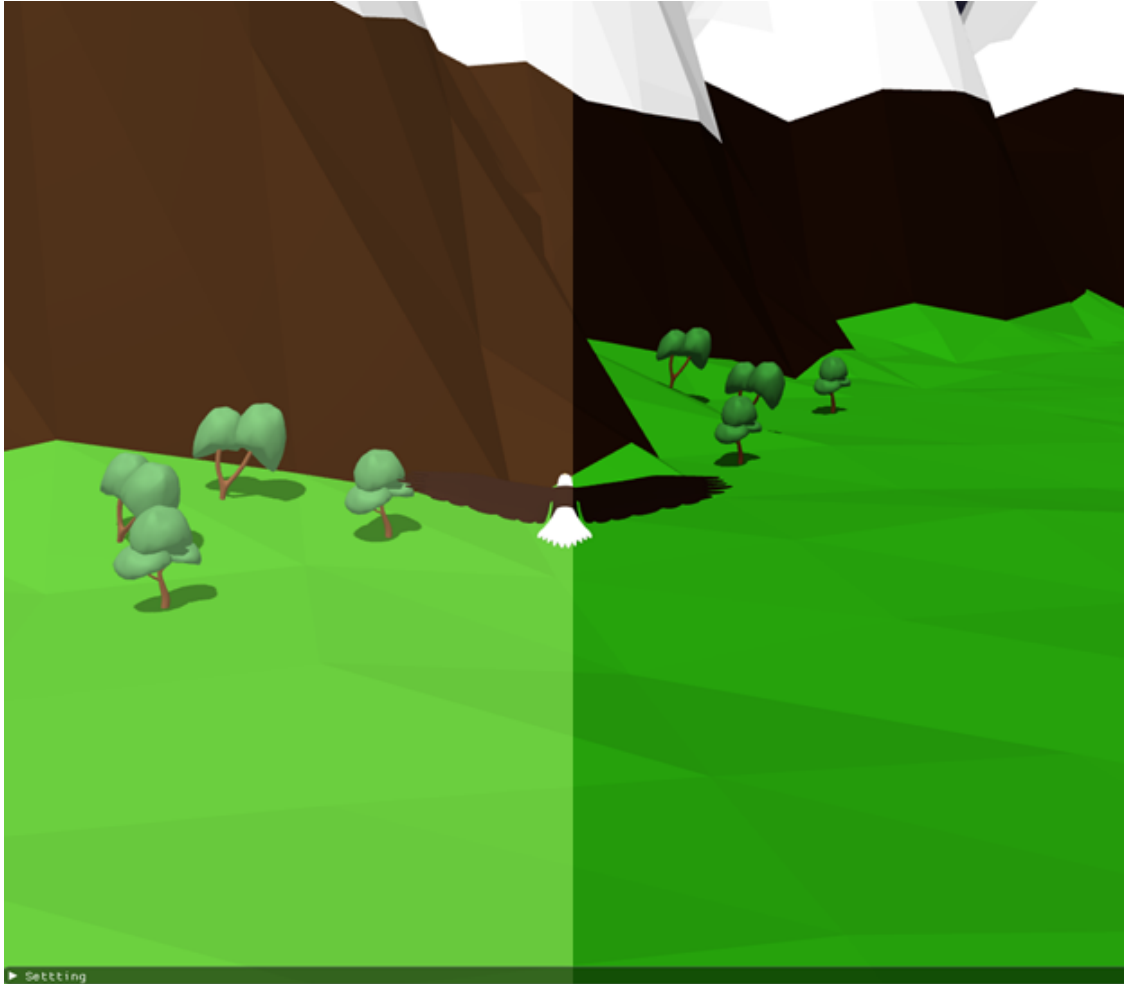


- 4. Texture mapping: 与天空盒结合
- 5. Model Import

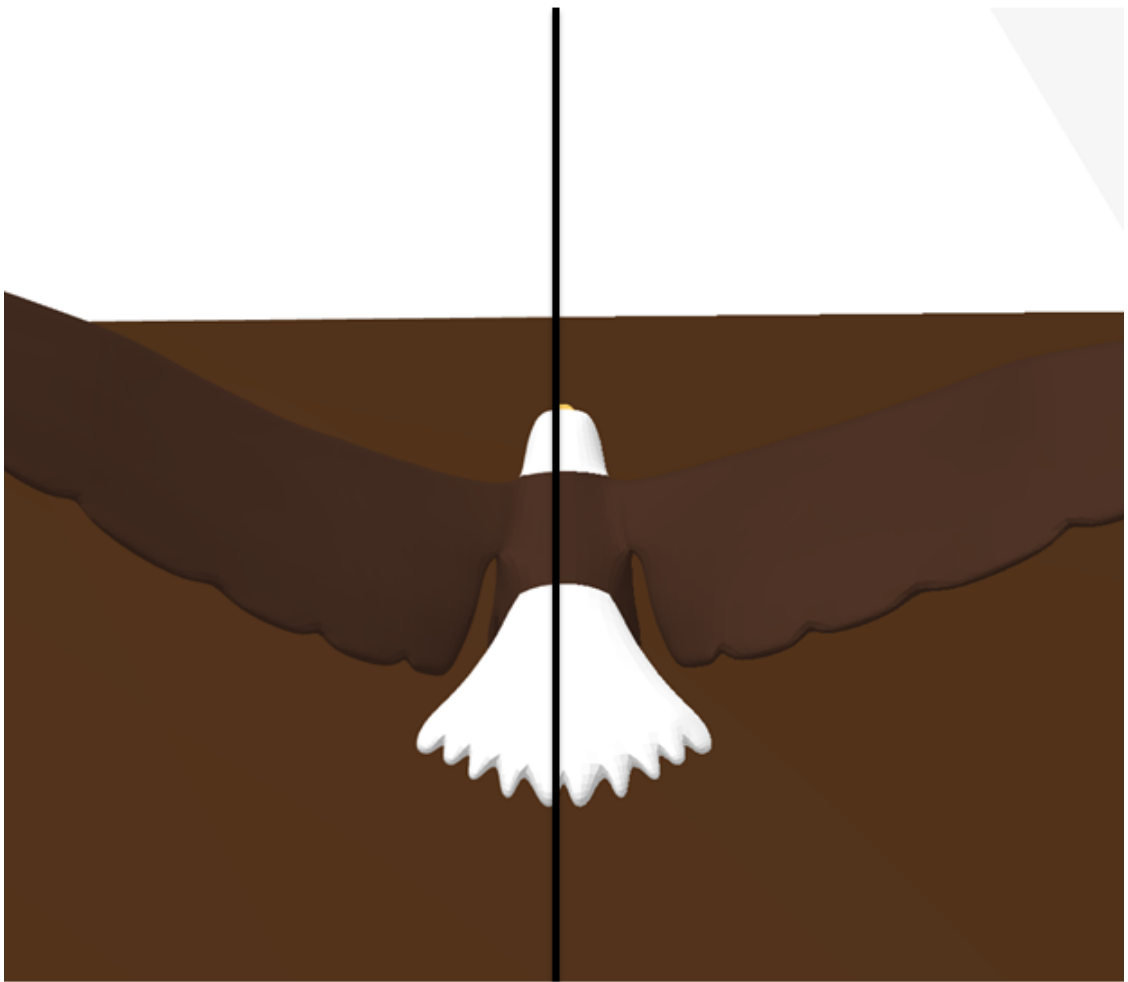
Bonus

- 1. Gamma correction: 前面说到的光照模型没有得到我们期望的颜色效果，这里使用伽马矫正修复了这个问题，但实际上，我们只知道使用伽马矫正能让我们更方便地调节光照，为何需要它才能达到与Blender中建模地效果我们还不清楚。下面的效果图中，左边是做了伽马矫正的，右边是原

图。



2. Anti-Aliasing: 抗锯齿我们使用的是MSAA，即用多个子采样点来代替原来的一个。下面的效果图中，左边是做了抗锯齿处理的，具体看白色和棕色的交界处即可。



3. Display Text: 文本渲染我们使用的是 FreeType 这个库，当我们需要渲染一个字符时，会先拿到它的位图，将位图数据作为 α 通道，然后开启混合模式，即可实现背景透明的文本渲染。



4. Sky Box: 天空盒的实现非常简单, 原先我们实现 Camera Roaming 时, 会将物体的坐标与view 矩阵相乘, 但如果想要让玩家觉得自己始终是在世界中, 无法靠近地图的边缘, 只需要将与天空盒相关的view矩阵变成3x3即可, 我们知道, 线性变换无法实现位移操作。



5. Particle System: 我们使用粒子系统来增加黑洞的科幻感, 要实现这个功能, 我们需要一个粒子发生器, 它维护一个粒子数组, 不断地查看粒子的生命值, 如果生命值耗尽则将其重置, 即可大大节省资源。而粒子的颜色位置等都是与生命值相关的函数, 如生命值越小, 颜色越浅。注: 可看同

文件夹的 ParticleSystem.gif 图。



6. Skeletal Animation: 带有骨骼动画的模型除了有skin（即一系列的网格），还有骨骼 `aiBone` 和动画 `aiAnimation`。骨骼没有大小和位置，只有一个名字和初始旋转矩阵（决定了骨骼的初始姿态），除此之外，每一个骨骼还存储了它所影响的顶点的ID以及影响的权重；动画 `aiAnimation` 存储了一系列的关键帧和当前动画的持续时间。关键帧存储的是从初始姿态到当前姿态，所有骨骼要经过的旋转平移和缩放。在骨骼动画播放的时候，首先根据当前时间找到动画的前一个关键帧和后一个关键帧，然后根据到这两个关键帧的时间距离进行线性插值，得到当前关键帧。再将当前关键帧的旋转平移和缩放应用到所有相关的骨骼上，从而改变当前的骨骼姿态。最后在着色器中，将骨骼当前的姿态，按照权重作用到受其影响的顶点上，从而改变顶点的位置。更详细的可以阅读小组成员胡嘉鹏同学的个人博客 [OpenGL加载骨骼动画](#)。

遇到的问题和解决方案

Blender

1. **物体合并**：在blender中合并物体时，如果先选中场景，再选择其它物体进行合并，导出的.fbx会出现size过小的情况 解决方案：只需先选中物体再选择场景即可
2. **场景显示不完整**：在对大场景进行建模时，会出现场景一部分被挡住显示不了 解决方案：原因在于投影变换的远平面设置得太近（只是场景太大），只需要调大 View 的 End 即可

OpenGL

1. **光照**：导入的模型Ka为零导致整体场景偏暗 解决方案：如果Ka为零，则令Ka等于Kd即可
2. **光照**：场景颜色不符合 Blender 建模预期 解决方案：使用伽马矫正即可
3. **模型加载**：骨骼动画能加载，但动作不对 解决方案：原因是位置没有对应好，导致没有将对应的骨骼信息传入着色器

小组成员分工

成员	分工	占比
黄俊凯 16340082	场景建模	25%
黄俊杰 16340081	负责光照、阴影、摄像机等	25%
黄嘉杰 16340080	场景建模	25%
胡嘉鹏 16340076	负责模型加载、场景切换、文字渲染等	25%

个人报告

黄俊凯 16340082

本次项目中，我主要负责了场景建模、协调组员工作和制作PPT等工作。在进行场景建模时，我们选择了 Blender 作为建模工具，在学习使用 Blender 过程中，因为版本问题导致的快捷键不通用，耗费了我们建模组很多时间。建完模后发现模型导入出现了很多问题，如：物体的大小没有达到预期，动画出现问题，这其中有建模的问题，同时也有OpenGL程序的BUG。上面的这些问题中，很多在百度、Google上都找不到答案，我们的解决方案也很奇葩，非常不优雅，这也给我们团队带来一次次低谷。但是在小组成员的共同努力下，我们一步步见证我们的程序绘制出的动画愈发生动优美，这也是我们前行的动力。同时审美这个东西，真的是天赋吧，我们的场景风格是 lowpoly，在网上有很多现成的模型、图片可以参考，尽管如此，我们建出来的模型还是有点小丑。在本次项目中，改变了我之前一个人就能顶一片天的想法，队友真的能让项目增色不少，大家都在共同学习进步。

黄嘉杰 16340080

在本次项目中，我主要负责的是现代城市场景的建模，在使用 Blender 的时候，由于网上没有很好的文字教程，很多操作都需要看视频学习，学习 Blender 的使用耗费了很多时间。然后在用 Assimp 库导入模型的时候又遇到了很多问题，比如如果模型过大就会报错、模型的材质如果选择了"使用节点"的方式上色导入之后没有颜色、加了路径跟随约束的 fbx 模型导入之后会放大 100 倍等等，这些问题我本来以为是我在建模过程中的操作错误，但是这些问题 Google 都查不到，然后我们转变方向从代码入手，通过断点调试发现有些确实是 Assimp 库的问题并做了对应的解决方案。通过这次项目，我体会到了3D建模的困难，另外，我觉得我们小组成员间的分工协作都做得很好，队友们都很给力，很多难题都一起解决了。

胡嘉鹏 16340076

开发终于结束，这里对开发做一些总结。模型加载是用时最长的一个部分，因为项目的设计中要支持骨骼动画。理解assimp库加载的数据结构和骨骼动画的实现原理用了很长的时间，这里也让我感受到我工程能力的不足，遇到复杂的结构就头脑一片混乱，画了好几张图一系列视频，才勉强理解骨骼动画的原理，最终结合leanopengl和ogldev教程中的代码，才实现了骨骼动画。碰撞检测遇到的问题是所有模型都要进过model, view, projection三个矩阵变换，才得到我们最终看到的样子，而模型原本的尺寸不知道从哪里获取，且每个模型所用到的model矩阵并不完全相同，这使得对于模型大小的估量存在困难。难以使用需要用到模型大小的AABB碰撞检测。这个问题最终也没有解决，只是使用了简单的基于中心距离的碰撞检测，效果一般。文字渲染还是比较简单，跟着leanopengl很快就做完了。此外，我队友负责的部分也有许多技术的难点，绝大部分的技术难点被攻破，队友很给力，感谢队友。

黄俊杰 16340081

在本次项目中，我主要负责的是光照、阴影、摄像机和粒子系统。对于光照，我是直接根据opengl教程采用冯氏光照模型，光源采用的是平行光，因此传入片段着色器的是光的方向而不是位置。光照的3个分量中材质信息来自于模型中的Ka、Kd、Ks和Ki，而gamma校正只需要将原本的颜色应用一个 $1.0/\gamma$ 的幂运算即可，光照这部分并没有什么难点。对于阴影映射，由于采用的是平行光，在生成深度贴图时，光源空间的变换采用的是正交投影。阴影映射的难点在于需要调节许多参数，比如光源空间的变换时投影视锥的各个参数，以及为了获得一个良好的阴影效果，还需要调整光的方向使得光的方向与物体形成一定的夹角。这里的一个技巧是绘制出深度贴图，深度贴图的各个物体的层次明显，则阴影效果良好。通过绘制深度贴图以及使用gui，最终调节出一个较满意的参数。对于摄像机，我们增加了一只鹰一直保持在摄像机前。这里的实现方式是，始终将鹰的位置放置在摄像机前方，而鹰的旋转角度随摄像机的欧拉角进行旋转。为了给鹰赋予一定的灵动行性，我们监听了鼠标的移动事件，使得鹰具有可以拐弯的效果。对于粒子系统，按照opengl教程定义一个粒子类和粒子生成器类，然后在每个事件循环调用粒子更新粒子以及绘制粒子即可。在这次项目中，我的工作是在队友的工作的基础之上的，多亏了队友创建以及加载出模型，我只需要在整体框架中，添加些基础的代码，来实现我们项目的基本功能。