1.1选题的背景

1、人们对于3D场景的设计与物品3D展示的需求日趋增长

1.1、3D展示允许受众能以更加全面的视角对观察目标进行观察，而不是局限于常规的2D图片和视频这些有限的观测手段

观察者能够观察观察目标的任意部分，并对自己关注的重点部分着重观察。观察者能够根据自己的意愿随时调整对

观察目标的角度、距离，并能模拟不同的光线条件，使得观察者的观测能更加接近真实的环境，甚至突破现实一些

环境的约束。

2、目前对于3D展示的途径有限

2.1、像3DMAX、Unity、blender、solidWorks等软件对于使用者和硬件设备有着一定的要求，而且虽然功能强大但是操

作复杂，对普通用户的使用成本比较高，并不友好

2.2、现阶段国内专门用于3D展示的门户网站并不多，许多对于3D展示有需求的使用者在使用Sketchfab等国外的网站

（国内网站。。。）

3、随着浏览器性能以及网络速度的提高，在线形式3D展示的门槛已经不再存在

1.2国内外研究的现状

随着信息技术浪潮的不断推进，人们对于信息获取渠道提出了越来越高的要求，用户观看阅浏览信息是其中一个重要的途径。除了以传统的二维平面的途径之外，人们更希望能够以三维的方式，以更加贴近现实的方式来获取信息以得到更加的体验，比如人们在购买商品的时候，三维展示的商品比传统的图片、视频的手段而言更具优势，不仅仅可以增加商品的可交互性，还允许用户从各个角度观察，大幅度的提高用户的使用体验。

目前，webgl已经从1.0发展到2.0，使用的着色器版本为OpenglES3.0。新版本的webgl能够提供更快的实时渲染，更新的纹理、渲染器以及更低的内存消耗。除了性能的提升和新的渲染手段外，新版本的webgl还提供了一套标准测试手段来保证不同的浏览器的兼容性。

虽然webgl以浏览器为平台可以解决大部分的兼容性问题，但是由于各大浏览器厂商的相互竞争，实际上还有一部分的浏览器对webgl的支持程度并不高，如IE浏览器。同时，由于加载模型所需的内存资源较多，对用户电脑硬件的要求较高，以及网速的限制，目前webgl技术还没有被大规模的使用，不过随着5G时代的到来以及个人电脑性能的快速提升，相信webgl作为一种优秀的展示技术会被快速普及。

1.3发展趋势

目前，有90%的web浏览器支持WebGL1.0，在3DCG进入网络世界后，webgl已经成为事实上的标准。

3D内容的展示还需要与web高度兼容的文件格式，而标准组织Khronos工作组制定的GLTF的文件格式极大的促进了WebGL的传播。GLTF并不特定于某个动或是工具，它被开源在github上，任何人都可以获得他的规格。

作为一项无需插件就可以在浏览器上实现丰富的3D表达的技术，webgl毫无疑问是当前网络3DCG开发的首选。

随着互联网的发展，桌面应用向web应用、移动应用的过渡是无法避免的现状。伴随这WebGL标准被广泛的接受，市场上出现了很多基于HTML的

三维引擎，如ThingJS、THREEJS、SceneJS...可以说WebGL技术打发展颠覆了传统浏览器页面的表现形式，为使用者带来了全新的体验。

然而浏览器平台固有的限制同样制约了webGL的推广。为了适应浏览器在计算能力、内存受限方面的限制，我们必须要对现有的，基于桌面应用

的模型数据组织形式和加载方式进行调整。我们需要对三维模型进行轻量化的处理，处理的手段包括图元合并、几何对象构件的对象化、场景八叉

树划分、Lod、模型文件压缩等等

2.1基本框架

本论文将使用react为前段开发的框架搭建前端页面系统，使用node作为后端处理请求，使用mysql作为数据库存储必要的数据，使用three作为3D内容开发的主要框架。主要的开发工具使用vscode，主要的环境平台使用chrome。

本论文的研究目的是构建一套简易的在线3D场景编辑的展示的系统，该系统允许用户无需使用专业的软件平台，也不需要有专业的3D软件的操作知识就可以完成建议3D场景的搭建的编辑，可以预见的，该系统完成后满足普通人们对于3D场景搭建和编辑的需求，对于降低信息多元化展示的成本有着显著的促进作用。

由于WebGL原生的api十分接近底层，是非常低级的接口，并且这些接口的使用往往要求使用者具备专业的数学矩阵和图形学的知识，对于没有相

基础的人而言难以入门。而THREE.js在WebGL的api的基础上又进行了一层封装，隐藏了页面3D渲染的细节，将常用的目标对象抽象出来。它将3D场

景拆解为网格、材质、光源等等对象，方便了使用者调用。Three.js的在开发的时候使用了很多图形引擎的高级技巧，能极大的提升3D场景的运行

性能，而且THREE.js完全开源在Github上，个人可以很方便的获得它的源码。

2.2研究的重点和难点

当前对于web端的3D展示并没有统一的模型格式，导致需要使用不同的方法加载处理不同的模型。而且对于当前一些通用的模型而言，如.max后缀的模型文件，是3dmax的专属模型文件格式，由于没有开源的原因，个人无法实现加载，除非购买3dmax的sdk才能加载相应的模型。由于无法规范所有上传模型文件的规范以及提前得知模型文件的内容，因此很难在直接加载，不做其他调整的情况下完美的展示模型文件。

由于在浏览器平台做3D应用的缘故，可以加载的模型资源受限，过大的模型会导致浏览器崩溃，卡顿。而且过大的模型文件也会导致网页加载时间过长，严重影响用户体验。而且当在场景中存在过多的网格对象时，网格对象之间的碰撞检测，物体的选择等都会是需要解决的问题。

在复杂3D场景中的webGL应用有着代码逻辑复杂，开发周期漫长等无法避免的问题，这就要求在开发的前期就预先做好规划、书写规范化、组件

化、工程化的标准代码，尽量增强代码的可阅读性，降低开发难度。

2.3需要解决的关键问题

1、解决不同规范模型的加载，保证加载效果

2、场景中网格对象的碰撞检测

3、关于大模型的加载问题

4、平台编辑器性能优化

5、模型纹理替换与光线渲染

3研究的方法和途径

1、关于3D功能的开发方面，three与webgl的官网以及github上都能找到许多优秀的案例以供参考，学习别人的优秀案例是降低学习成本和加快开发进度的有效手段

2、收集不同的模型类型，以保证测试用例的全面

3、在研究遇到瓶颈时及时联系导师寻求帮助

4、在遇到难题时学会查找资料，从资料中寻求解决问题的思路，避免闭门造车

4预期的研究成果

完成系统的搭建，其中包括以下功能：

1、场景/模型的加载展示

1、用户可以上传模型、场景、贴图等资源到个人素材库或是公共素材库

2、用户在上传资源的时候可以预览、分类、重命名

2、场景的搭建与编辑（简单房屋场景的搭建编辑）

1、用户可以直接使用素材库中的模型场景贴图等进行场景搭建

2、用户可以通过2D作图的方式设计房间的户型图，确定房间的基本结构

3、用户在操纵模型的时候提供辅助操纵功能

4、用户可以使用历史版本功能进行回退

5、系统提供场景中模型的简单碰撞检测

6、支持场景云端保存和多人协作

3、简单模型物体的设计创造（\*）

1、用户拥有有限的简单模型设计能力，包括对模型进行合并、切削、贴图等

4、场景的在线存储，修改和三方展示

1、支持不同终端一种状态，基本信息存储在服务器并通过账号系统进行状态的统一

2、可以为指定的场景生成公开的展示页面，供用户方便向第三方展示

5、用户系统，管理员系统的权限控制

1、确定游客、注册用户、管理员之间的权限，以及不同角色之间的联系

6、同一场景的多人协同操作（\*）

1、用户可以查找并添加其他注册用户为好友

2、用户可以为自己创建的场景邀请好友一同创作

7、个人作品的发布（\*）

1、用户可以选择将自己的作品保存到个人素材库或是发布到公共素材库

2、系统会自动将公共素材库中的场景作品推送到主页进行展示

5研究进度计划

1、第七学期 1 - 8 周 文献的收集与整理

2、第七学期 9 - 12 周 文献综述和外文翻译

3、第七学期 12 - 15 周 完成开题报告和基础数据收集整理

4、第七学期 16、17 周 和第八学期 1 - 9周 设计、开发

5、第八学期 10 - 12 周 撰写毕业设计论文

6、第八学期 13 -14 周 论文的答辩与修改