

**江 苏 科 技 大 学**

**基 于 Three.js 的**

**3D 模 型 共 享 平 台**

学 院： 计算机学院

专 业： 计算机科学与技术

姓 名： 朱庆壁

学 号： 192210701144

指导教师： 宦婧

**2023年4月**

摘 要

随着3D模型在工业设计、建筑设计、游戏开发等领域的应用越来越广泛，3D模型共享成为了一个重要的话题。本文提出了一个基于Three.js的3D模型共享平台，旨在为用户提供高效、便捷的3D模型分享和浏览方式。

当前，WebGL技术得到了广泛的应用，基于WebGL的3D库如Three.js、Babylon.js等也日益受到关注。其中，Three.js是一个非常流行的基于WebGL的3D库，它通过提供简单易用的API，使得在Web中创建和展示3D图形变得更加容易和高效。随着HTML5在2011年成为W3C推荐标准，WebGL技术也随之被广泛支持。同时，移动设备的普及让越来越多的用户通过手机和平板电脑来访问Web，使得基于Web的3D应用需求日益增加。在这种背景下，基于WebGL的3D库应运而生，Three.js也成为了其中最受欢迎的一种。

本平台使用了基于Three.js的WebGL技术，能够在现代浏览器中运行。用户可以通过上传自己的3D模型或浏览其他用户上传的3D模型。同时，本平台还支持3D模型的在线预览、旋转、缩放等基本操作，使用户能够更好地了解和展示自己的作品。

在实现中，本平台采用了Vue 3框架和Spring MVC作为前后端分离的架构，同时使用了jQuery封装的Ajax技术来实现客户端和服务器之间的异步通信，以实现实时的数据传输和页面更新。此外，本平台使用MySQL数据库来存储用户信息、3D模型存放位置等信息，使得数据的组织和管理更加高效、可靠。还使用了MyBatis-Plus框架来简化数据持久化操作。

目前，本平台已经完成了所有的开发和测试，希望能为3D模型的分享和展示提供新的途径。

Abstract

With the increasingly widespread use of 3D models in fields such as industrial design, architectural design, and game development, 3D model sharing has become an important topic. This paper proposes a 3D model sharing platform based on Three.js, aiming to provide users with efficient and convenient ways to share and browse 3D models.

Currently, WebGL technology has been widely used, and 3D libraries based on WebGL such as Three.js, Babylon.js, etc. are also receiving increasing attention. Among them, Three.js is a very popular WebGL-based 3D library that makes it easier and more efficient to create and display 3D graphics in the web by providing simple and easy-to-use APIs. With HTML5 becoming a W3C recommended standard in 2011, WebGL technology has also been widely supported. At the same time, the popularity of mobile devices has led to an increasing demand for web-based 3D applications. In this context, 3D libraries based on WebGL have emerged, and Three.js has become one of the most popular ones.

This platform uses WebGL technology based on Three.js and can run in modern browsers. Users can upload their own 3D models or browse other users' uploaded 3D models. Additionally, the platform also supports basic operations such as online preview, rotation, and scaling of 3D models, allowing users to better understand and showcase their work.

In terms of implementation, this platform adopts Vue 3 framework and Spring MVC as the front-end and back-end separation architecture, and uses Ajax technology encapsulated by jQuery to realize asynchronous communication between the client and server, achieving real-time data transmission and page updates. In addition, this platform uses MySQL databases to store user information, 3D model storage locations, and other information, making data organization and management more efficient and reliable. The MyBatis Plus framework is also used to simplify data persistence operations.

Currently, this platform has completed all development and testing, and hopes to provide a new way for sharing and showcasing 3D models.

目录

[摘 要 I](#_Toc130840041)

[Abstract II](#_Toc130840042)

[目录 III](#_Toc130840043)

[第一章 绪论 1](#_Toc130840044)

[1.1 选题的目的与意义 1](#_Toc130840045)

[1.2 国内外研究现状及存在的问题 2](#_Toc130840046)

[1.3 主要研究内容 4](#_Toc130840047)

[第二章 系统总体设计 5](#_Toc130840048)

[2.1 系统开发环境 5](#_Toc130840049)

[2.2 服务器架构 5](#_Toc130840050)

[2.2.1 Linux系统中的服务器架构 5](#_Toc130840051)

[2.2.2 Windows系统中的服务器架构 6](#_Toc130840052)

[2.3 数据库设计 6](#_Toc130840053)

[2.4 用户模型文件的管理 6](#_Toc130840054)

1. 绪论
   1. 选题的目的与意义

Web开发在过去几年中发生了很大变化。JavaScript库已经成熟，浏览器变得更加强大，并且大多数系统都很好地支持HTML5规范。不过，目前Web主要关注2D来表示信息、创建游戏和构建网站。但是，当前也有一种创建3D图形的出色且标准化的方法。那就是WebGL；它提供了一个API来创建硬件加速的3D图形。

然而，WebGL API并不是很容易使用。它需要使用C语言进行编程，并且需要确实地深入了解WebGL的内部工作方式才能使用此API。不过幸运的是，有许多可用的JavaScript库隐藏了WebGL的复杂性，并且提供了一个易于使用的API来创建3D应用程序和可视化。目前，这些JavaScript 库中最优秀的，便是Three.js。Three.js提供了一个易于使用的API和一系列可用于访问WebGL的高级材料和形状。

在现代社会，随着3D技术的不断发展，3D模型应用范围越来越广泛。在工业设计、建筑设计、游戏开发等领域，3D模型已经成为必不可少的工具。然而，由于3D模型的复杂性和体积较大，如何高效地共享和浏览3D模型成为了一个重要的话题。

基于此，本文提出了一个基于Three.js的3D模型共享平台。该平台能够为用户提供高效、便捷的3D模型分享和浏览方式。用户可以通过上传自己的3D模型或浏览其他用户上传的3D模型，同时支持在线预览、旋转、缩放等基本操作，以及灯光、材质等高级设置。该平台不仅能够满足用户的3D模型共享需求，同时还能为用户提供更好的展示自己作品的途径。

本文的选题目的是为了解决3D模型共享的问题，提供一个高效、便捷的3D模型共享平台。该平台的意义在于为用户提供了一个全新的3D模型共享和展示方式，可以帮助用户更好地展示自己的作品，同时也可以促进3D模型在各领域的应用发展。

* 1. 国内外研究现状及存在的问题

当今社会，随着科技的不断进步，3D技术应用领域也在不断拓展。3D模型作为3D技术应用的基础，被广泛应用于工业设计、建筑设计、游戏开发等领域。在这些领域，3D模型的共享和交流已经成为一项重要的工作。而基于Web技术的3D模型共享平台，成为实现3D模型共享的一种有效途径。

在国内，已经有一些基于Web技术的3D模型共享平台，如3D溜溜、3D鱼塘等，但这些平台存在一些问题。首先，这些平台的3D展示效果不够优秀，用户体验较差。其次，这些平台对于上传的3D模型的格式和大小有一定的限制，不够灵活。此外，这些平台缺少对3D模型的高级操作，如灯光、材质等的设置，使得用户无法充分展示自己的作品。总之，国内的基于Web技术的3D模型共享平台仍有一些不足之处。

与此相比，在国外，有很多公司和组织已经开发了各种类型的3D模型共享平台。例如，Sketchfab是一个总部位于法国的平台，用户可以上传、展示和共享自己的3D模型，同时也支持一些高级特性，如3D模型在线编辑和VR浏览等。此外，Thingiverse是一个致力于3D打印的社区，用户可以上传和分享自己的3D打印模型，同时也可以浏览和下载其他用户共享的模型。此外，这些平台还支持社交化的功能，使得用户可以方便地分享和交流自己的作品。

然而，国外的这些平台在国内并不十分流行，主要是因为这些平台的语言和界面都是英文的，不够贴近国内用户的需求。同时，这些平台在国内的使用可能会面临一些网络访问问题，如访问速度慢、不能访问等。

针对上述问题，国内外学者们已经进行了大量的研究和探索，提出了许多解决方案。

首先，有一些研究集中在3D模型压缩和传输技术方面，旨在减小3D模型的文件大小并提高传输效率。例如，研究人员在3D模型的网格数据中使用分层细节技术，使得3D模型在不同的分辨率下都可以呈现出良好的效果，从而减小文件大小。另外，也有研究利用多边形网格简化算法来减少模型的三角面数，从而进一步降低文件大小。

其次，还有一些研究集中在3D模型共享和交互方面，旨在提高用户对3D模型的共享和交互体验。例如，研究人员利用虚拟现实技术和手势识别技术来实现对3D模型的自由浏览和操作，从而提高用户的交互体验。同时，也有研究提出了基于社交网络的3D模型共享方案，使得用户可以在社交网络上分享和交流自己的3D作品。

尽管在上述领域中已经取得了一些进展，但是仍然存在一些问题和挑战。例如，在3D模型压缩和传输方面，如何在减小文件大小的同时保证模型的质量和细节，仍然是一个难以解决的问题。在3D模型共享和交互方面，如何实现多用户之间的实时共享和交互，以及如何提高用户对共享模型的信任度和安全性，也是亟待解决的问题。在3D模型版权保护方面，如何避免数字水印和加密算法被破解或绕过，从而保护作者的知识产权，也是一个需要考虑的问题。

* 1. 主要研究内容

1. Three.js技术的应用：本系统通过使用基于WebGL的Three.js技术，实现了一个3D模型共享平台，可以在现代浏览器中运行。为此需要掌握前置知识。它们包括基础的三维数学和几何学知识，如矩阵、向量、坐标系转换等；Three.js 库的基本结构和使用方法，包括如何创建场景、相机、灯光、材质等；还有Three.js 中常用的几何体、贴图、动画等相关知识。
2. 前后端分离架构：本系统采用了Vue 3框架和Spring MVC作为前后端分离的架构，通过使用jQuery封装的Ajax技术来实现客户端和服务器之间的异步通信，以实现实时的数据传输和页面更新。采用前后端分离架构可以将前端和后端的开发分开，降低代码的耦合度，使得代码维护和开发更加方便和高效。
3. MyBatis-Plus框架的应用：这其中包括MyBatis-Plus框架的基本使用方法，包括配置、实体类映射、查询、更新、删除等操作；MyBatis-Plus框架与Spring框架的整合，包括配置、事务管理等；MyBatis-Plus框架与MySQL数据库的整合，包括数据库连接配置、数据源管理等以及MyBatis-Plus框架的高级特性，例如Wrapper查询。
4. MySQL数据库系统与Linux或Windows文件系统的结合设计：这需要研究SQL语言，包括SQL的基本语法和常用的操作命令，如查询、插入、更新、删除等；数据库设计和规范，包括数据表设计、数据字段设计、数据完整性约束等；Linux和Windows系统的文件结构，包括Linux文件系统的树状结构和Windows文件系统的分层结构等。
5. 3D模型共享平台的设计与实现：本研究设计并要实现一个基于Three.js的3D模型共享平台，用户可以通过上传自己的3D模型或浏览其他用户上传的3D模型，同时还支持3D模型的在线预览、旋转、缩放等基本操作，使用户能够更好地了解和展示自己的作品。
6. 系统总体设计
   1. 系统开发环境

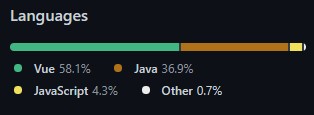
本系统开发部分全程在Windows系统中进行，对于测试部分则分别在Linux系统和Windows系统中进行。如图2.1所示，系统中主要的代码文件为vue文件和java文件。其中vue文件内的代码使用Visual Studio Code进行编写。对于Java代码，开发时使用的集成开发环境为IntelliJ IDEA。

图2.1系统设计代码语言占比

* 1. 服务器架构
     1. Linux系统中的服务器架构

在Linux系统中，本系统用到了两个服务器，分别是Nginx和Apache Tomcat。其中Nginx因为需要提供两类服务，所以占用了两个服务端口。

对于Nginx，其主要功能为负责文件的传输和以代理的方式解决跨域问题。它提供的一类服务是返回用户请求的静态页面以及转发部分部分用户请求至Apache Tomcat。由于该系统是一个单页面应用（Single Page Application，SPA），所以返回的静态页面其实只包括一个html文件、一个js文件以及若干个图片和模型文件。另一类服务是处理用户预览和下载3D模型文件的请求。这部分请求并不会被送至Apache Tomcat，而是直接由Nginx处理，这样做的一个优势在于有效降低了Apache Tomcat的服务压力，且加快了该类请求的响应速度。

对于Apache Tomcat，它主要负责为Java Servlet提供运行环境，并且不与用户直接进行交互，用户所有需要Apache Tomcat处理的请求由Nginx以代理的形式转发。Java Servlet则负责处理客户端发来的 HTTP 请求并返回相应的响应，这其中就包括用户上传模型文件的请求。

* + 1. Windows系统中的服务器架构

在Windows系统中，本系统同样使用了Apache Tomcat作为Java Servlet的运行环境。与在Linux系统中不同的是，在Windows系统中并没有使用Nginx作为与用户直接进行交互的服务器。原先在Linux系统中由Nginx提供的两类服务改为分别由webpack-dev-server和Live Server提供。Live Server和webpack-dev-server占用两个不同的端口。

对于webpack-dev-server，它是由Webpack提供的开发服务器。在Windows系统中，它主要功能为提供一个开发环境下的服务器，需要让页面随着代码的变化而变化。它只负责返回用户请求的静态页面，本系统并未使用webpack-dev-server具备的代理功能，用户发送的需要Apache Tomcat处理的请求由浏览器直接发送至Apache Tomcat的服务端口。因此在Windows系统中，本系统所采用的跨域问题处理方式与Linux系统中也不相同。

对于Live Server，它是一款Visual Studio Code插件。在Windows系统中，Live Server负责处理用户预览和下载3D模型的请求，这部分请求同样不会由Apache Tomcat进行处理。

* 1. 数据库设计

在数据库方面，本系统使用的是MySQL数据库，其中创建了三张数据库表，分别是用户信息表，其中存储了用户的用户名和密码；模型文件信息表，该表用于存储用户上传的3D模型文件信息，包括上传用户的id、模型名等；用户收藏信息表，本系统为用户提供有收藏功能，除了下载外，用户还可以收藏其他用户上传的模型。该表用于存储所有用户的模型收藏信息。三张表的字段信息如图2.2所示。

Java程序无法直接访问数据库，它需要通过一些持久层框架或者API来间接对数据库进行操作。本系统选择了Mybatis-Plus框架来简化数据库访问层的开发，该框架是Mybatis的一个增强工具，而Mybatis则是一个建立在JDBC之上的持久层框架，它几乎消除了所有的JDBC代码。JDBC（Java Database Connectivity）则是Java平台用于连接数据库的标准API，真正使得Java程序能够访问到数据库的便是JDBC，MyBatis可以看作是对JDBC的封装和简化。

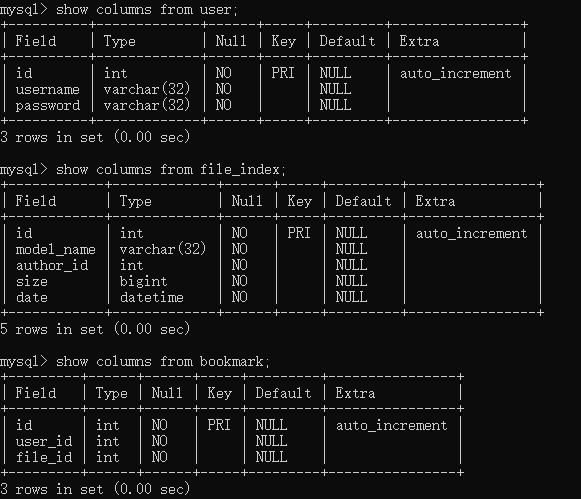
* 1. 用户模型文件的管理

图2.2数据表字段信息

从图2.2中可以看到数据库中并没有直接存储3D模型文件。实际上，这类文件以及它们所对应的用于展示的图片文件存在了Windows或者Linux的文件系统中。原因之一是存储在数据库中的文件通常需要进行二进制存储，因此占用的存储空间相对较大。而如果将文件存储在文件系统中，则可以采用原始的文件格式进行存储，需要的存储空间也相对较少。而另一个更重要的原因是文件系统通常具有较高的访问速度，能够更快地读取和写入文件。但也因此，当用户传来需要下载或者预览的文件id时，我们无法直接通过id将对应的文件传回给用户，我们需要先从数据库中找出该id对应的文件在文件系统中的存储路径，然后将该路径返回给用户，让用户通过返回的路径去访问Nginx以获得对应的文件。

1. 系统详细设计
   1. Three.js模型库应用
      1. 场景（Scene）

在Three.js中，场景（Scene）是所有Three.js 3D对象的容器，它是一个抽象的概念，用于表示一个虚拟的三维环境。它是Three.js应用程序的核心，所有的渲染都是在场景中进行的。如表3-1所示，为了让场景显示任何内容，我们需要三种类型的组件。

场景可以通过以下代码创建：

const scene = new THREE.Scene();

表3-1场景中的三种组件

|  |  |
| --- | --- |
| **组件** | **描述** |
| 相机 | 相机决定了屏幕上渲染的内容 |
| 光源 | 光源影响着创建阴影效果时材质的显示和使用方式 |
| 物体 | 物体是从摄影机的角度渲染的主要对象，包括立方体、球体等。 |

* + 1. 几何体、纹理和材质（Geometries, Textures and Materials）

几何体（Geometries）是3D模型的形状，例如盒子、球体、圆柱体、平面等。每种几何体都由一组顶点和连接它们的线或面组成。在Three.js中，可以通过创建一个几何体对象来定义一个3D模型的形状。

例如我们可以通过以下代码分别创建一个正方体几何体和一个正方形几何体：

const geometry = new THREE.BoxGeometry(1, 1, 1, 128, 128, 128);

const planeGeometry = new THREE.PlaneGeometry(1, 1, 128, 128);

纹理（Textures）是指映射到3D对象表面的图像或视频等内容。它可以用来为几何体和材质添加外观、颜色和纹理等特性，纹理还可以用来实现各种效果，例如木纹、金属质感、云层、雾等等。

材质（Materials）是3D模型的外观特性，例如颜色、纹理、反光度、透明度等。在Three.js中，可以通过创建一个材质对象来定义一个3D模型的外观特性。Three.js中有十多种材质类型，这里只介绍其中的两种。分别是MeshStandardMaterial和MeshBasicMaterial，MeshBasicMaterial是最简单的材质类型，它只有基本的颜色和透明度属性。这个材质不考虑场景中的光照，也不会产生阴影。MeshStandardMaterial则是一种更高级别的材质类型，它考虑了光照、阴影和反射等因素。它支持漫反射、镜面反射、法线贴图等高级特性，可以用于更真实的渲染效果。但相比之下，MeshStandardMaterial更加复杂，需要更多的计算资源。

纹理可以作为材质的组成部分，不同的纹理可以应用于材质的不同属性中，例如颜色、透明度、环境光遮蔽、置换、法线等。可以通过Texture类来创建不同的纹理，利用TextureLoader类加载外部图片作为纹理。

（帅气的地方）

* + 1. 网格（meshes）

添加到场景当中

* + 1. 不同用途的相机（Cameras）
    2. 使用不同的光源（Light sources）
    3. 动画（Animations）

相机移动

* 1. MyBatis-Plus框架应用
  2. Vue.js框架应用
     1. Node.js
     2. Webpack
  3. Spring MVC框架应用
  4. Windows环境测试
     1. webpack-dev-server配置
     2. Apache Tomcat配置
  5. Linux环境测试
     1. Nginx配置
     2. Apache Tomcat配置

文件解压。。。tar.gz

* + 1. MySQL配置
    2. 阿里云服务器内部署

1. 系统功能展示