Web development has been changing a lot in the last couple of years. JavaScript libraries have matured, browsers have become more powerful, and the HTML5 spec is well supported on most systems. Currently, though, the Web mostly focuses on 2D to represent information, create games, and build websites. There is also, however, a great and standardized way to create 3D graphics. This is called WebGL; it provides an API to create hardware-accelerated 3D graphics.

However, the WebGL API isn't very easy to use. It requires you to program in C, and you really need to have an in-depth knowledge of how WebGL works internally to work with this API. Luckily though, there are a number of JavaScript libraries available that hide the complexity of WebGL and provide you with an easy-to-use API to create 3D applications and visualizations. Currently, the best of these JavaScript libraries, the one that is explained in this book, is called Three.js. With Three.js, you're provided with an easy-to-use API and a whole range of advanced materials and shapes that you can use to access WebGL.

In this book, we'll walk you through the most essential features of Three.js. We do this by creating some advanced visualizations that cover various parts of the Three.js API.

**选题的目的和意义 ：**

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。

目前，政府有大量的新基建项目，比如数字孪生、智慧城市、智慧园区、智慧工厂、智慧消防等等，都涉及到了3D可视化技术。

**国内外研究现状：**

这十年来，Web技术得到了快速的发展，随着WebGL的普及，网页的表现能力越来越强大，网页上已经可以开始做出很多复杂的动画、精美的效果，还能通过WebGL，在网页中绘制高性能的3D图形。

随着浏览器的性能和网络带宽的大幅度提升以及WebGL的实现，使得3D的技术不再是桌面程序的专利，越来越多的外部应用使用了3D技术。

比如说对于网站而言3D技术实现了企业网站的三维呈现，让企业形象更加直观、更立体地展现给客户。打破传统平面展示的模式，打造一个智慧的、个性的、创新的企业形象。比如说腾讯的腾讯云服务的官网。

**主要研究内容：**

本文主要通过使用WebGL技术中的第三方引擎框架——Three.js，并结合后端SSM框架及Ajax交互技术开发一套基于Three.js的3D模型共享平台。采用Browser/Server浏览器和服务器架构，综合利用Three.js，Spring MVC，MySQL等技术，实现3D模型文件在浏览器和服务器之间的传输。

·主要研究内容如下：

1. 基于Browser/Server浏览器和服务器结构开发分布式浏览系统的架构。
2. 三维图形数据的存储和研究，由于glb文件中包含大量的三维模型数据信息，为了保证模型的上传与下载速度，必须设计有效的数据存储方法。
3. 三维场景的设计与呈现。为用户呈现三维模型是，要分别调整物体的缩放、旋转以及场景的光照等参数。
4. 数据库系统开发，实现对用户登录及权限的管理，glb文件的存储与管理。
5. 前端页面的设计与实现，主要包括用户与模型的交互，实现用户模型的权限管理以及模型的上传与下载。

**研究方法、步骤和措施：**

1. 学习Three.js框架的使用，包括场景（Scene）、相机（Camera）和渲染器（Renderer）等。
2. 学习glb文件的加载方式。在对基础场景完成搭建后，需要在三维空间中添加三维模型。三维模型主要以glb、gltf等文件格式为主，每种文件格式都有其对应的前端模型加载插件库。
3. 研究三维模型在服务器的存储方式。由于不少三维模型都需要较大的存储空间，数据库并不仅仅用来存储数据，数据库还要保证它的查找效率，所以并不能直接将模型存入数据库，而应存入服务器硬盘中。而为此便需要掌握Spring框架操作文件的对应类。

[Three.js Essentials (dominictran.com)](https://www.dominictran.com/pdf/ThreeJS.Essentials.PACKT.pdf)

[Learning Three.js – the JavaScript 3D Library for WebGL - Second Edition - Jos Dirksen - Google Books](https://books.google.co.jp/books?hl=en&lr=&id=Xja9BwAAQBAJ&oi=fnd&pg=PP1&dq=three+js+webgl&ots=Rs1elC4d6i&sig=Atn5MZLldl-Bp_bB_5lPXB36vyE&redir_esc=y#v=onepage&q=three%20js%20webgl&f=false)