基于 Web3D 的多人沉浸式学习平台

一、选题概述

在这个课程项目选题中，你需要实现一个基于 Web3D 的多人在线沉浸式学习平台。该虚拟环境可以是一个博物馆、校园、展览馆等学习环境。 用户可以通过浏览器登录该虚拟场景，拥有自己的虚拟形象作为化身，可以在这个环境中行走，和周围的环境交互，以及和其他用户进行交流等。

该学习平台具有以下特点:

- 采用 Web3D 技术模拟真实世界中的环境，用户可以通过浏览器不受时空限制地沉浸式学习。
- 多个用户可以选择化身进入同一个环境，引入社交因素。
- 实现教学场景。比如博物馆中，玩家可以漫游博物馆浏览文物学习知识，并进行交流。又如可以实现一个多人虚拟环境，支持让学习者移动虚拟的汉诺塔，并进行相互讨论。

二、系统需求和分析

2.1 功能要求

2.1.1 基本功能

- 前端页面
    - 支持用户注册和登录;
    - 选择相应的 Web3D 场景，比如可以通过课程或者学习房间的方式。
- 用户后端管理页面
    - 记录用户个性信息如虚拟形象等;
    - 根据所选择的学习场景，可提供历史学习行为查看和分析功能。
- 支持多人加入同一个虚拟世界中实现协同学习:
    - 用户的虚拟化身之间相互可见，并且行为共享;
    - 可以进行一定方式的交流;
    - 环境中可以有一些可交互性的实体，如可以进行操作的汉诺塔。
- 维护虚拟世界的一致性:
    - 同一时刻各个化身能看到的场景应该是一致并且最新的。
- 系统部署在云服务器上，提供可以访问的公网地址。

2.1.2 进阶功能

- 加入一些人工智能因素，增加一个由计算机控制的智能虚拟人(NPC)，比如智能教师，虚拟导游等。可以根据用户化身的行为做一些简单智能的响应，比如介绍展品等。
- 在环境中的实体可以采用语义 Web 技术加入描述，甚至可以支持推理。
- 支持多模态的交互，比如在虚拟世界中通过音视频交流等。

2.1.3 附加说明

- 3D 建模和动画制作不是项目重点，不要求建模非常逼真，可以采用公开的可免费下载使用的模型。

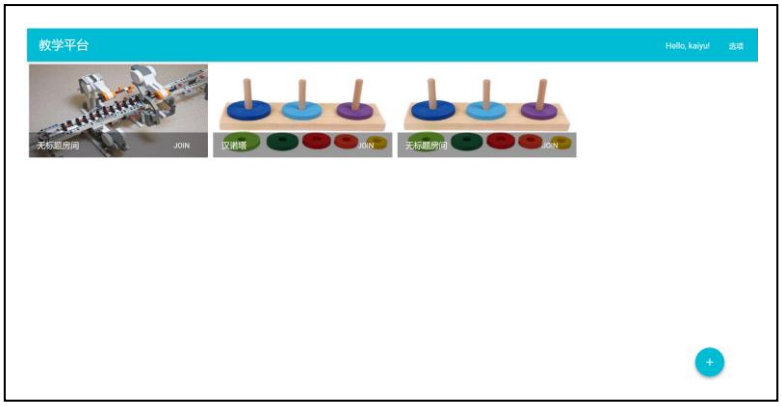
2.2 非功能性要求

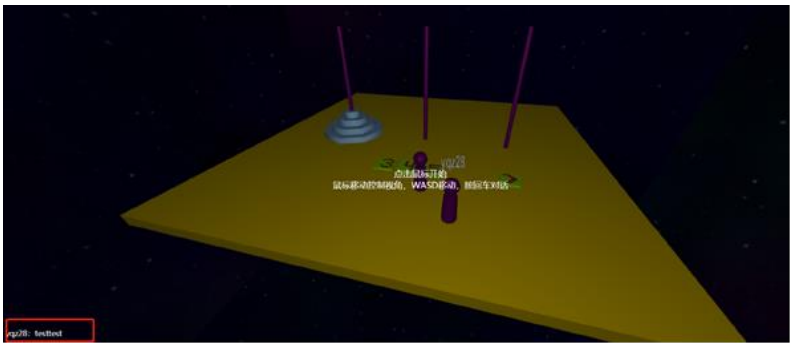
除了功能需求外，平台还有以下几个非功能性要求：

- 实时性：虚拟环境中的用户交互、行为同步实时准确。
- 可用性：具有较好的用户体验性和可用性。页面返回在合理的范围内，比如通常需要在 3 秒内返回。如果有音视频通讯，则音视频聊天具有合适的帧率，比较流畅。
- 易用性：操作简单，提示清晰，让用户可以简单、快速地上手，并且具有很好的用户体验。
- 稳定性：平台应在某些服务出现问题后只是部分功能缺失，比如视频聊天功能在某些网络环境下无法正常运行，但是其他功能仍能稳定运行。

2.3 示例场景截图

以下为一个计算思维虚拟仿真教学平台的场景截图，演示了多个用户进入该虚拟场景，共同讨论和解决汉诺塔问题，从而实现可视化地学习计算思维中的递归思维。如下第一个图中，平台列出所有的学习房间。用户可以选择房间，从而进入某个虚拟教学场景，如第二个图所示，该三维显示的虚拟场景支持多人行为同步和实时文字沟通等。

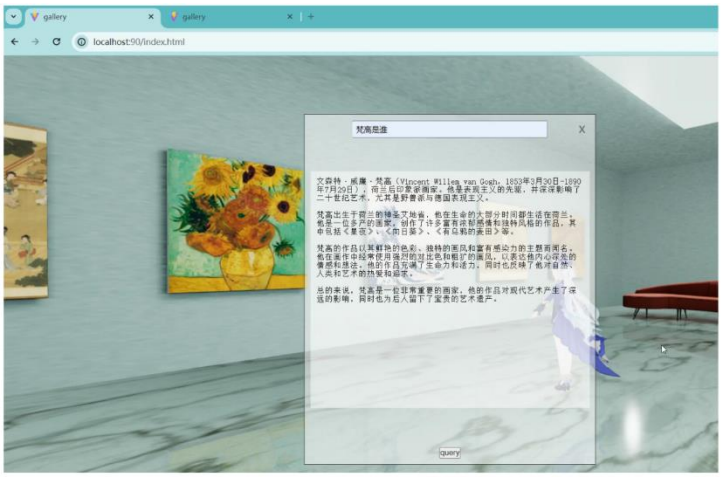




也可以是一个艺术品展览馆，如下面截图所示，用户可以在馆中漫游，并且相互可见：

可以点击看到艺术品的介绍：

语音交互：双方同时输入需要对讲的用户名，可以进行语音沟通。还可以接入智能系统 API（如文心一言、ChatGPT 等），实现智能问答（可选进阶功能）：



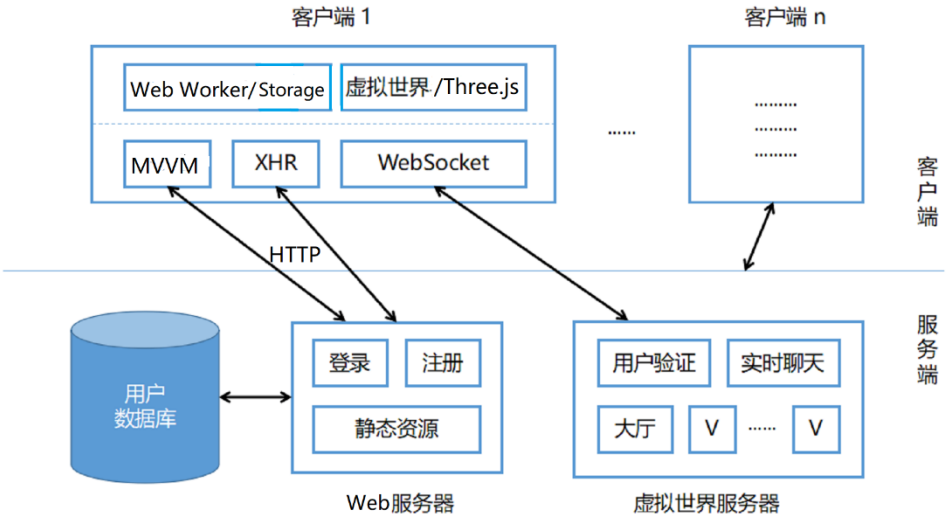
三、技术要求和参考

3.1 技术方案

- Web3D 建议使用 Three.js 框架。多用户的位置同步、协同聊天等功能，建议使用 WebSocket 技术，可以采用框架 Socket.io。
- 采用前后端分离的架构，后端提供 RESTful风格的 API 给前端调用，前后端之间通过 JSON 或者 XML 传递数据。
- 后端采用Spring Boot 框架，MyBatis 作数据库持久化层，数据库不限制，可以采用 MySQL、Redis、MongoDB 或者图数据库如 Neo4j。
- 前端与后端的程序均部署在云服务器上，在线上环境直接演示。

3.2 参考技术架构

应用课程所学习的 HTML5 高级技术，包括 XHR，websocket，web storage，web worker，以及 Web3D 技术，如封装了 webGL 的 Three.js 框架，技术架构图如下所示：



四. 评分细则

4.1 分数组成

- 基本功能分: 完成系统基本功能，如下面量表中所给出的基本功能项，满分 100 分。
- 进阶任务分:包括但不限于 AI 相关功能、多模态交流，云服务应用，以及其他具有创新性的功能等。最多 30 分。
- 根据小组分工及个人完成工作量，由小组成员间协调，给出贡献比例，该比例以相对方式提供，比如某位贡献度最大的同学是 100%的话，其他组员按照相对于他的得分给出比例，比如另外一位组员 90%表示贡献度最大同学 PJ 得分是 100 分的话，这位组员是 90 分。

4.2 评分量表

	功能项	评分指标	分值
基本功能	UI 和交互	UI 设计合理，具有较好的用户体验	5
	基本页面与流程	登录和注册功能使用正常	5
		后端用户管理功能	8
	虚拟场景以及交互	可交互的 3D 场景建模合理，用户体验好	10
		场景功能的完成度和交互的丰富程度	15
		支持多人加入该场景，并实现行为共享	17
		支持用户间的文本、动作等方式交流	10
	工程能力	文档说明清晰、详细，图文并茂，图示准确	10
		系统架构设计合理规范	5
		代码清晰，风格合理，具有良好的设计模式	10
		服务部署在云平台上，具有很好的可访问性	5
进阶功能	AI 能力	响应用户虚拟行为的智能导师	5
		虚拟场景中的实体添加语义描述	5
	交互性	创新交互与多模式交流支持（如 WebRTC）	5
	云计算应用	合理采用 Docker 以及多种云服务	5
	其他	以上仅做参考，支持增强功能和用户体验的其他创新设计和开发，进阶功能不超过 30 分。	每项 5 分

4.3 评分点说明

1）不得抄袭，否则得分为 0 并且后果自负！

2）每一项的分数取决于该项功能的完成度。完成度和可用性越好，分数越高。

3）项目完整度和易用性评价标准:

- A. 功能残缺，不能完整运行，有明显 bug。
- B. 完成规定的用户功能和操作，无明显瑕疵。
- C. 界面舒适，操作合理，响应迅速，鲁棒性强。

A、B、C 分别对应 分数的 0 - 30% ，30% - 70% 分，70% - 100% 分。

4）附加功能必须在文档中明确写出，概述该功能并描述实现原理。

5）项目设计文档需要至少包含:

- 项目组织以及其中每个文件的说明。
- 关键功能实现的细节。
- 服务器部署配置的详细介绍。

6）团队分工文档需要至少包含:

- 团队成员、分工、具体完成工作，列出每个人的贡献比例。
- 其他需要补充说明的问题，比如创新之处的思考。

五. 提交

- 源代码:推荐使用 Git 进行协作，提交到 GitHub 等 Git 托管平台上。
- 文档:推荐使用 Markdown 编写项目文档，与源代码一同提交到 Git 托管平台上。

可供访问的公网地址，以及系统的使用说明文档。