

实验二：创建基于 Three.js 和 socket.io 的多人在线系统

在该基于 Web 的虚拟世界中，系统可以随机给用户化身进行场景漫游，以及可以进行多人之间的互动。我们部署好的 Demo 在线访问网址：<http://124.223.224.204:3000>。



本次 Lab 的目标首先是通过提供的源代码和教学视频，学习如何基于 Three.js 和 Socket.io 构建和部署这样的 Web3D 多人环境；另外，就是引导各位同学基于该源代码进行修改，朝完成自己的课程项目前进。

评分规则：

- 1) **基本任务：满分 70 分。**能按照我们下文给出的文档进行 demo 的部署，并且记录你觉得可以补充的一些经验，记录你的源代码学习笔记，提交记录的 Word 文档以及部署的网址。
- 2) **进行的改进：满分 30 分。**包括针对本组 PJ 最后设计的场景进行的改进，比如改为最后共同游览计算机发展历史的场馆，并可以选择化身而不是 DEMO 中的随机选定，创建不同的房间等；如果本次不是选的 Web3D 课程项目，可以是选择不同房间，中间有数字藏品的展示，客户可以一起观赏和聊天等。在 Word 文档中增加记录下你这部分的工作。这些记录最后可以放入课程设计的文档。
- 3) **提交截止期：5 月 5 号 11:59 之前交。**迟交按照每天 10%扣除部分分数，直到扣完。

各位同学一定要遵守时间，Lab 帮助大家完成整个课程项目，给了四周时间；大家一定要认真按时完成，避免后面 PJ 完成时间紧张！

DEMO 源码地址：

<https://github.com/PMLS3/3D-multi-player>

或者文件夹中的“3D-multi-player-main.zip”文件

和 DEMO 配套的 Bilibili 教学地址（建议去掉中文字幕）：

https://www.bilibili.com/video/BV1th411J7zo/?spm_id_from=333.337.search-card.all.click&vd_source=1068ea6e2b9d5d54342d2f2dfc0faa0f

DEMO 安装部署步骤：

1. 安装 NodeJS 环境

- a) 下载 node 包：<https://nodejs.org/en/download/>，选择适用于 Linux 版本文件，以'.xz'后缀结尾文件；（如果你是用的 ubuntu18，文件夹中的“node-v18.15.0-linux-x64.tar.xz”就直接可以使用）
- b) 将下载好的 node 文件放到云服务器上的/usr/local/路径下；
- c) 解压文件：tar -xvf node-vxx.xx.xx-linux-x64.tar.xz
- d) 重命名：mv node-vxx.xx.xx-linux-x64 node
- e) 建立软链接：
 - i. sudo vim /etc/profile，编辑该文件加入以下内容：
 - ii. export NODE_HOME=/usr/local/node
 - iii. export PATH=\$NODE_HOME/bin:\$PATH
- f) 编译：source /etc/profile
（以上步骤如果无效，或者嫌麻烦，也可以采用简化为
安装 node: **sudo apt install nodejs**
安装 npm: **sudo apt-get install npm**
）
- g) 通过显示版本号，检查 node 和 npm 安装：node -v; npm -v

2. 从上面给出的 DEMO 源代码网址 <https://github.com/PMLS3/3D-multi-player> 下载源代码，放到云服务上的某个目录下，比如 home 目录下。安装项目依赖，包括 express 和 socket.io

- a) `cd home/3D-multi-player-main/nodeapps/blockland;`
 - b) 删除原先的 `package-lock.json` 和 `package.json` 文件;
(`rm -rf package-lock.json; rm -rf package.json.`)
 - c) 初始化: `npm init;`
 - d) 安装 `express` 包: `npm install express@4.15.2 --save`
 - e) 安装 `socket.io` 包: `npm install socket.io@2.2.0 --save`
3. 运行项目: 在 `/3D-multi-player-main/nodeapps/blockland` 路径下执行: `node app.js`
- a) 默认为 2002 端口, 可在 `app.js` 的 50、51 行自行修改;
4. 使用 `forever` 运行 `node` 后端, 为了防止终端关闭导致后端断开:
- a) 安装 `forever`: `npm install forever -g`
 - b) 运行: `forever start app.js` (在 `app.js` 路径下)

友情提示:

- 1) 某些命令需要管理员权限, 可以采用前面加 `sudo` 获得管理员权限进行操作。
- 2) 修改某个目录的权限成为完全控制, `ubuntu` 下面的命令是:
`sudo -R chmod 777 目录名`
如果你要删除和进行修改说权限不够的时候采用该命令。
- 3) `source` 命令执行无效时, 先执行下 **`sudo -s`**
- 4) 建议安装 **FinalShell** 来进行远程操作, 也可以用 **WinSCP** 来进行文件传输。

其他参考学习资料:

除开以上提供的 B 站的教学视频, 以及配套的源代码; 本次实验 2 还提供了另外一份从零开始构建简单多人环境的文档“lab2-基础学习文档”, 以及配套的源代码“lab2-基础学习源码”。

期待各位同学做的有趣、有用的虚拟世界!