

联机五子棋

3. 前端交互的展示

五子棋-联机版



房间号

请输入游戏房间号

加入游戏

联机五子棋

4. 五子棋游戏部分的展示与制作

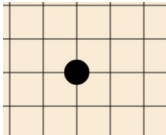
1. 棋盘的制作
2. 棋子的逻辑绘制

参考以下目录代码

名称	修改日期	类型	大小
identity.html	2020/6/4 18:11	Chrome HTML D...	1 KB
index copy.html	2020/6/4 16:14	Chrome HTML D...	3 KB
index.html	2020/6/6 9:05	Chrome HTML D...	2 KB
joinGame.html	2020/6/4 18:15	Chrome HTML D...	2 KB
room.html	2020/6/6 10:52	Chrome HTML D...	5 KB
room.js	2020/6/6 10:57	JavaScript 文件	3 KB
server.js	2020/6/6 10:37	JavaScript 文件	4 KB

5. 判断五子棋输赢的算法

1. 五子棋如果赢了的话，肯定是因为最后落的子才赢的
2. 我们从最后落得子开始推算
3. 穿过每个子能成的有四种情况（横线，竖线，两条斜线）



4. 根据该点和轴线方向，筛选出所有符合条件的落子
5. 根据顺序s个为一组进行拆分
6. 分别判断每组中是否有匹配成功的
7. 参考以下目录文件

名称	修改日期	类型	大小
identity.html	2020/6/4 18:11	Chrome HTML D...	1 KB
index copy.html	2020/6/4 16:14	Chrome HTML D...	3 KB
index.html	2020/6/6 9:05	Chrome HTML D...	2 KB
infoScreen.js	2020/6/8 14:43	JavaScript 文件	5 KB
joinGame.html	2020/6/4 18:15	Chrome HTML D...	2 KB
room.html	2020/6/8 14:22	Chrome HTML D...	5 KB
room.js	2020/6/6 10:57	JavaScript 文件	3 KB
server.js	2020/6/8 14:38	JavaScript 文件	5 KB

websocket和http如何配合使用

让ws服务器和http服务器公用一个端口，可以使用如下的写法：

1. 创建http服务器

```
server-http.js  server-ws.js  index.html
server-http.js > <unknown>
1 const http = require('http');
2
3 const httpServer = http.createServer((req, res) => {
4   res.end('ok')
5 })
6
7 module.exports = httpServer
```

2. ws服务器，并传入http服务器实例

```
server-http.js  server-ws.js  index.html
server-ws.js > wsServer.on('connection') callback
1 const ws = require('ws');
2 const httpServer = require('./server-http');
3 const wsServer = new ws.Server({
4   server: httpServer
5 })
6
7 wsServer.on('connection', (client) => {
8   client.send('ws服务器已经建立链接')
9 })
10
11 httpServer.listen(3000, () => {
12   console.log('基于3000端口的http服务器和ws服务器都已经运行')
13 })
```

在ws环境下如何获取客户端信息

```
const ws = require('ws');
const httpServer = require('./server-http');
const wsServer = new ws.Server({
  server: httpServer
})

wsServer.on('connection', (client, req) => {
  console.log(req);
  client.send('ws服务器已经建立链接')
})

httpServer.listen(3000, () => {
  console.log('基于3000端口的http服务器和ws服务器都已经运行')
})
```

和http服务器中的req是同一类对象