附件1

QQTang软件测试用例集

# 大厅模块测试

## 模块功能描述

大厅模块的主要功能是维护一个游戏大厅的状态，支持任意玩家进行“进入/离开大厅”、“进入/离开房间”、“准备/取消准备”、“发言”等操作。由于服务器需要将每个客户端的行为告知其他客户端，大厅模块还需要提供一个接口，用来查询大厅当前的状态。此外，大厅模块维护了游戏开始、结束的状态，并及时通知游戏模块，所以还有一些与游戏模块交互的接口。

综上，大厅的主要功能有四部分：

1）支持获取大厅的实例

2）支持一个用户的行为，如“进入大厅”等

3）支持查询大厅状态

4）支持与游戏模块的交互

### 1.2获取大厅实例

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 函数描述 | static HallInfo HallInfo.getInstance(); | | | |
| 函数功能 | 获取大厅实例 | | | |
| 测试原则 | 可以得到一个非空的实例 | | | |
| 测试目的 | 输入/动作 | 期望输出/响应 | 实际输出/响应 | 状态 |
| 测试是否能获取大厅实例 | - | 非null的值 | 非null的值 | 通过 |

### 1.3支持用户行为

该功能主要通过以下函数实现：

boolean execute(JSONObject query) throws Exception;

其中，参数query是一个字典，它一定包含type域，以表示操作类型，例如login表示进入大厅，这个将在后面具体介绍。若返回值为true，则表示操作成功；若返回值为false表示操作失败；若抛出异常则表示输入数据格式不合法。

### 1.3.1 支持用户进入大厅

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean execute(JSONObject query) throws Exception  其中query必须包含：  “type” 值为“login”  “user” 表示用户名  “details” 表示用户信息 | | | | |
| 函数功能 | 一个用户进入大厅 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false  数据格式错误时，抛出异常 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteLoginSuccess-1 | 用户可进入大厅 | 正常数据 | true | true | 通过 |
| testExecuteLoginFail-1 | 测试已经在大厅中的用户无法重新进入大厅 | 一个已经在房间中的用户 | false | false | 通过 |
| testExecuteLoginException-1 | 测试格式错误时抛出异常 | 一个缺少details的query | 异常 | 异常 | 通过 |

### 1.3.2 支持用户进入房间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean execute(JSONObject query) throws Exception  其中query必须包含：  “type” 值为“enter”  “user” 表示用户名  “room” 表示房间号（0~49），999表示随机  “pos” 表示座位号（0~3），999表示随机 | | | | |
| 函数功能 | 支持用户进入房间，有三种模式：  a) 支持用户进入指定房间指定座位；  b) 支持用户进入指定房间随机座位；  c) 支持用户进入随机房间随机座位。 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false  数据格式错误时，抛出异常 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteEnterSuccess-1 | 用户可进入指定房间指定座位 | 正常数据  room=1 pos=1 | true | true | 通过 |
| testExecuteEnterSuccess-2 | 用户可进入指定房间随机座位 | 正常数据  room=1 pos=999 | true | true | 通过 |
| testExecuteEnterSuccess-3 | 用户可进入随机房间随机座位 | 正常数据  room=999 pos=999 | true | true | 通过 |
| testExecuteEnterFail-1 | 当用户不在大厅中，要求进入房间，不予处理 | user为一个不再大厅内的用户 | false | false | 通过 |
| testExecuteEnterFail-2 | 当用户已经在房间中，要求进入房间，不予处理 | user已经在某个房间中 | false | false | 通过 |
| testExecuteEnterFail-3 | 用户要求进入的房间下标溢出 | room=101 | false | false | 通过 |
| testExecuteEnterFail-4 | 用户要求坐下的位置下标溢出 | pos=-1 | false | false | 通过 |
| testExecuteEnterFail-5 | 用户要求坐在一个有人的位置 | room和pos是一个有人的位置 | false | false | 通过 |
| testExceptionEnterFail-6 | 用户要求进入指定房间随机座位，但该房间已满 | room是一个满的房间 | false | false | 通过 |
| testExceptionEnterException-1 | 当query不合法时抛出异常 | query缺失room和pos | 异常 | 异常 | 通过 |

### 1.3.3支持用户离开房间

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean execute(JSONObject query) throws Exception  其中query必须包含：  “type” 值为“leave”  “user” 表示用户名 | | | | |
| 函数功能 | 一个用户离开所在房间 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false  数据格式错误时，抛出异常 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteLeaveSuccess-1 | 用户可离开房间 | 正常数据 | true | true | 通过 |
| testExecuteLeaveFail-1 | 当用户不在大厅内要求离开房间，返回false | user为一个不在大厅内的用户 | false | false | 通过 |
| testExecuteLeaveFail-2 | 当用户不在房间内要求离开房间，返回false | user为一个不在房间内的用户 | false | false | 通过 |

### 1.3.4支持用户准备

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean execute(JSONObject query) throws Exception  其中query必须包含：  “type” 值为“ready”  “user” 表示用户名 | | | | |
| 函数功能 | 一个用户准备 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false  数据格式错误时，抛出异常 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteReadySuccess-1 | 用户可以准备 | 正常数据 | true | true | 通过 |
| testExecuteReadyFail-1 | 当用户不在房间内时要求准备，返回false | user不在房间内 | false | false | 通过 |
| testExecuteReadyFail-2 | 当用户不已经在准备状态时要求准备，返回false | user为一个已经准备的用户 | false | false | 通过 |

### 1.3.5支持用户取消准备

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean execute(JSONObject query) throws Exception  其中query必须包含：  “type” 值为“unready”  “user” 表示用户名 | | | | |
| 函数功能 | 一个用户取消准备 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false  数据格式错误时，抛出异常 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteUnreadySuccess-1 | 用户可以取消准备 | 正常数据 | true | true | 通过 |
| testExecuteUnreadyFail-1 | 当用户不在房间内时要求取消准备，返回false | user不在房间内 | false | false | 通过 |
| testExecuteUnreadyFail-2 | 当用户未准备时要求取消准备，返回false | user为一个未经准备的用户 | false | false | 通过 |

### 1.3.6支持用户发送消息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean execute(JSONObject query) throws Exception  其中query必须包含：  “type” 值为“message”  “user” 表示用户名  “message” 表示需要发送的消息 | | | | |
| 函数功能 | 一个用户在房间内说话 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false  数据格式错误时，抛出异常 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteMessageSuccess-1 | 用户可以在房间内发送消息 | 正常数据 | true | true | 通过 |
| testExecuteMessageFail-1 | 当用户不在房间内时要求发送消息，不予执行，返回false | user不在房间内 | false | false | 通过 |
| testExecuteMessageException-1 | 当query缺失message时，抛出异常 | query缺失message | 异常 | 异常 | 通过 |

### 1.3.6支持用户离开大厅

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | boolean logout(String user) throws Exception | | | | |
| 函数功能 | 一个用户离开大厅 | | | | |
| 测试原则 | 数据可操作时，返回true  数据不可操作但格式正确时，返回false | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testExecuteLogoutSuccess-1 | 用户可以离开大厅 | 正常数据，用户不在房间中 | true | true | 通过 |
| testExecuteLogoutSuccess-2 | 用户可以离开大厅 | 正常数据，用户在房间中 | true | true | 通过 |
| testExecuteLogout-1 | 当传入user不在大厅内时，返回false | user为一个不在大厅内的用户 | false | false | 通过 |

### 1.4支持查询大厅状态

由于服务器端需要向客户端发送大厅内的状态与事件，通信模块会调用大厅模块的函数来获取大厅内的状态与事件。接口函数主要有：

JSONObject getStatus() throws Exception

JSONObject getEvents(int now) throws Exception

JSONObject getDetails(String user) throws Exception

JSONObject getMessage(String user) throws Exception

下面将详细介绍这几个函数以及对应的测试过程

### 1.4.1 获取大厅状态

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | JSONObject getStatus() throws Exception | | | | |
| 函数功能 | 索要当前的大厅状态  返回函数应包含：  “users” JSONObject  “rooms” JSONArray  “count” int 表示当前的事件数量 | | | | |
| 测试原则 | 返回值包含上述内容 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testGetStatusSuccess-1 | 可以获得当前大厅状态 | - | 包含上述内容的字典 | 包含上述内容的字典 | 通过 |

### 1.4.2 获取大厅事件

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | JSONObject getEvents(int now) throws Exception | | | | |
| 函数功能 | 索要大厅中最近发生的事件（编号从now开始到最新为止）  返回函数应包含：  “events” JSONObject  “rooms” JSONArray  “count” int 表示当前的事件数量 | | | | |
| 测试原则 | 返回值包含上述内容 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testGetEventsSuccess-1 | 可以获得大厅事件 | 0 | 包含上述内容的字典 | 包含上述内容的字典 | 通过 |

### 1.4.3 获取用户资料

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | JSONObject getDetails(String user) throws Exception | | | | |
| 函数功能 | 索要大厅中一个用户的详细资料 | | | | |
| 测试原则 | 返回值包含上述内容 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testGetDetailsSuccess-1 | 可以得到正常的用户资料 | 正常用户 | 完整的用户资料 | 完整的用户资料 | 通过 |
| testGetDetailsException-1 | 当所给用户不在大厅时，抛出异常 | 不在大厅内的用户 | 异常 | 异常 | 通过 |

### 1.4.4 获取用户聊天内容

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | JSONObject getMessages(String user) throws Exception | | | | |
| 函数功能 | 索要与一个用户相关的聊天信息 | | | | |
| 测试原则 | 返回值包含上述内容 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testGetMessagesSuccess-1 | 可以得到正常的聊天资料 | 正常用户 | 完整的聊天记录 | 完整的聊天记录 | 通过 |
| testGetMessagesFail-1 | 当所给用户不在房间内时，返回空数组 | 不在房间内的用户 | 空数组 | 空数组 | 通过 |
| testGetMessagesFail-2 | 当所给用户不在大厅内时，返回空数组 | 不在大厅内的用户 | 空数组 | 空数组 | 通过 |

### 1.5支持与游戏模块的交互

当大厅模块检测到某一房间内的所有用户全部准备时，将进行如下流程：

1 大厅模块调用start函数以通知游戏模块准备一局游戏

2 当游戏模块准备好一局游戏时，调用setGameStartInfo以告知大厅模块

3 与客户端的通信模块会随时调用getGameStartInfo来获取游戏开始的信息

4 游戏结束后，游戏模块调用setGameFinishedInfo以设置游戏结束状态

为此，我们需要测试2、3、4三个函数。由于1不是大厅对外的接口，所以不必单元测试。

### 1.5.1 设置游戏开始信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | void setGameStartInfo(int id, JSONArray infos) | | | | |
| 函数功能 | 当游戏模块做好准备后，会调用此函数，设置关于游戏的一些信息。例如客户端应该连哪个地址等。 | | | | |
| 测试原则 | 调用该函数后，房间状态变为playing | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testSetGameStartInfo-1 | 测试该函数是否可以工作正常 | 1号房间 | 1号房间状态变为playing | 1号房间状态变为playing | 通过 |

### 1.5.2获取游戏开始信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | void getGameStartInfo(String user) | | | | |
| 函数功能 | 获取一个用户的开始游戏的信息。若未开始游戏则返回空。 | | | | |
| 测试原则 | 对于将要开始游戏的用户，返回正确的游戏信息；  对于其他用户，返回空。 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testGetGameStartInfo-1 | 测试输入正确的情况下是否可以正常工作 | 正常输入 | 游戏开始信息 | 游戏开始信息 | 通过 |
| testGetGameStartInfo-2 | 测试所在房间未开始游戏时是否返回空 | 在房间内的用户，但是未全部准备 | null | null | 通过 |
| testGetGameStartInfo-3 | 测试用户不在房间时是否返回空 | 不在房间内的用户 | null | null | 通过 |
| testGetGameStartInfo-4 | 测试用户不在大厅时是否返回空 | 不在大厅内的用户 | null | null | 通过 |

### 1.5.3 设置游戏结束信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 函数描述 | void setGameFinishedInfo(int id, JSONArray infos) | | | | |
| 函数功能 | 当游戏结束后，游戏模块调用此函数告知大厅模块游戏结果 | | | | |
| 测试原则 | 函数调用是否不出现异常  调用后房间状态是否变为WAITING  调用后用户的经验值等信息是否改变 | | | | |
| 测例名称 | 测试目的 | 输入/动作 | 期望响应 | 实际响应 | 状态 |
| testSetFinishedInfo | 测试该函数是否可以工作正常 | 正常输入 | 房间状态waiting；  用户经验值改变 | 房间状态waiting；  用户经验值改变 | 通过 |