

软工需求文档

钱桥，许建林，谭志鹏，黄鑫，张道维，任勇

December 21, 2013

Abstract

本文是清华大学软件工程大作业的需求文档，主要从：功能需求，系统需求，系统分工三个方面来分别叙述。

Contents

1	功能需求	2
1.1	用户身份验证	2
1.2	玩家数据	2
1.3	游戏大厅及房间	2
1.4	玩游戏	3
2	系统分工	4
2.1	各部分系统作用	4
2.2	工作流程	5
2.2.1	服务器启动、服务器之间建立联系	5
2.2.2	用户登陆	6
2.2.3	用户在大厅的行为	7
2.2.4	游戏开始	8
2.2.5	用户登出	9
3	系统需求	10
3.1	<客户端逻辑>	10
3.1.1	Sign up	10
3.1.2	Sign in	10
3.1.3	Hall	11
3.1.4	Game	11
3.1.5	Sign out	11
3.2	<登陆服务器>	12
3.2.1	Register	12
3.2.2	Login	12
3.2.3	Refresh User Data	12
3.3	<大厅服务器>	13
3.4	<游戏服务器>	14
3.5	客户端显示	15

1 功能需求

简介:

该部分从功能角度描述了 QQ 糖这款游戏的需求。功能需求共有五大部分，分别是用户身份验证、玩家数据、游戏大厅、玩游戏、好友系统。前四部分已基本确定，并将需求细化至可检测的几条。第五部分好友系统并不影响整体框架，且是附加功能，所以具体的功能、需求以及呈现方式将在工程后期再酌情添加。

此外，在前四部分中有一些附加功能，它们在作业中并没有要求（但可提高用户体验）。所以，这些部分的实现优先级较低。

1.1 用户身份验证

1. 用户可以在客户端中的注册界面进行注册，需要填写邮箱（帐号），昵称，密码以及确认密码。
 - (a) 附加功能：增加网页注册
 - (b) 附加功能：增加邮箱验证
2. 用户可以在客户端的登陆界面中进行登陆，登陆时用户可选择大厅。登陆成功则跳转至大厅界面，登陆失败则停留在登陆界面并告知原因。
 - (a) 附加功能：用户登陆后可看见各个大厅的人数以及容量。
3. 用户可以在任意界面退出游戏。

1.2 玩家数据

1. 游戏中有 4 项属性值：移动速度（V）、泡泡威力（P）、泡泡数量（N）、闪避率（D），每项属性值最低为 1，最高为 10。属性值越高该项能力越强，具体的数值关系将在后面给出。
2. 每个用户拥有自己的经验值（E）和等级（L），胜场数（S），负场输（F）。

附加功能： 用户拥有自己的金币（G）。
3.
 - (a) 附加功能：用户可花费金币更换角色。
 - (b) 附加功能：每升一级可以加一点属性值（永久），达到上限的属性不能增加。
 - (c) 附加功能：金币可以用来购买道道具，具体道具待定。
4. 用户的基础属性值为：(1, 1, 1, 1)。每局游戏开始时，该用户的各项能力取决于他的基础属性值。
5. 在大厅界面中给出合适的位置显示用户的数据

1.3 游戏大厅及房间

1. 游戏中有多个大厅供用户选择，用户登录时可以选择登入任何未满员的大厅。数据会在大厅之间同步，即用户始终只有一份数据，与进入的大厅无关。
2. 每个大厅中有 50 个房间，每个房间有 4 个座位。用户可选择任意空座坐下。成功进入房间（坐下）后跳转至房间界面。
 - (a) 附加功能：快捷加入功能，将为用户自动分配房间。
3. 在房间内，用户可以准备或取消准备，也可离开房间，跳转至大厅界面。
4. 在房间界面中需要以友好的方式显示每个用户的资料（昵称、头像、所有数据）。
5. 当房间内所有用户准备后开始游戏。
 - (a) 附加功能：房主可更换地图（瓶颈在于多个地图）。

- (b) **附加功能：**第一个加入空房间的人为房主，当房主退出房间时，选择座位编号最小的人作为房主。
- 6. 房间界面设有聊天板，供同房间的人聊天。
 - (a) **附加功能：**提供语音聊天。每个用户可以设置是否参加语音聊天。

1.4 玩游戏

1. 一局游戏最多有 4 个玩家，初始位置分别在地图四角的出生点。
2. 每个玩家可控制自己的人物在地图中上、下、左、右移动。控制方式为“点哪走哪”，具体地说，设人物当前位置为 A，玩家所按位置为 B，则人物向向量 AB 的水平、竖直分量之中较长的分量方向移动。移动速度为 $1.5 + 0.2 \times V$ 。
3. 每个玩家可通过点击“放置”按钮，控制自己的人物放置泡泡。场上的（该玩家放置的）泡泡数量不得超过 $N+1$ ，否则不能继续放置。
 - (a) **附加功能：**通过双击屏幕放置泡泡。
4. 游戏中地面分为可行走部分和不可行走部分，不可行走部分包括箱子和石头等。
5. 游戏中的箱子可以被炸开，由概率出现道具。
6. 玩家不可以穿过泡泡，但是一种情况下例外，即玩家放置泡泡的时候可以在泡泡上。
7. 玩家之间没有碰撞体积。
8. 泡泡会在放置 3 秒后自行爆炸，爆炸呈“十”字形，爆炸范围为 P（不包含泡泡所在格）。被其他泡泡炸到的泡泡会立即爆炸。
9. 若一个玩家被泡泡炸到，则进行闪避判定，闪避概率为 $5\% + D * 1\%$ 。若闪避不成功则立即死亡。
10. 吃到游戏道具可以增加对应的属性值 1 点。通过道具增加的属性值将在游戏结束后失效。
11. 游戏中每个玩家需要通过炸死其余玩家来获得胜利。若某一时刻只剩一名玩家存活，则该玩家获得胜利。
12. 若游戏开始三分钟后还有多名玩家存活，则存活的玩家获胜，死亡的玩家失败。
13. 游戏结束后，结算每个玩家的经验和金钱。胜利的玩家经验 +10，金币 +10；失败的玩家经验 +2；断线的玩家没有奖励。

2 系统分工

简介： 为了进一步明确项目的需求，我们必须进行系统划分，将每一个需求分配至对应的系统。例如，数据存储应该交给服务器端的系统做，而界面显示应该交给客户端的系统做。我们查询了一些游戏开发的教程，总结了一些经验，最终决定将整个项目分为五个系统。在本部分中，我们将简单介绍这五个系统的主要作用以及它们之间分工合作的模式与流程。

整个项目分为客户端与服务器端。客户端包含<客户端逻辑>与<客户端显示> 两个系统，它们在同一个工程之中，通过函数接口进行合作。服务器端包含<登陆服务器>、<大厅服务器>与<游戏服务器>三个系统，它们是不同的三个工程，通过网络通信进行合作。

2.1 各部分系统作用

- <登陆服务器>：负责存储所有用户数据，对用户的登陆进行验证，接受新用户的注册
- <大厅服务器>：负责维护大厅、房间及其内部所有的玩家状态、数据，并与他们通讯
- <游戏服务器>：负责维护一局游戏中各个玩家的数据，并与各个玩家进行通讯
- <客户端逻辑>：负责与服务器通讯，处理接受的数据，响应玩家的操作
- <客户端显示>：将数据以友好的方式呈献给用户

2.2 工作流程

2.2.1 服务器启动、服务器之间建立联系

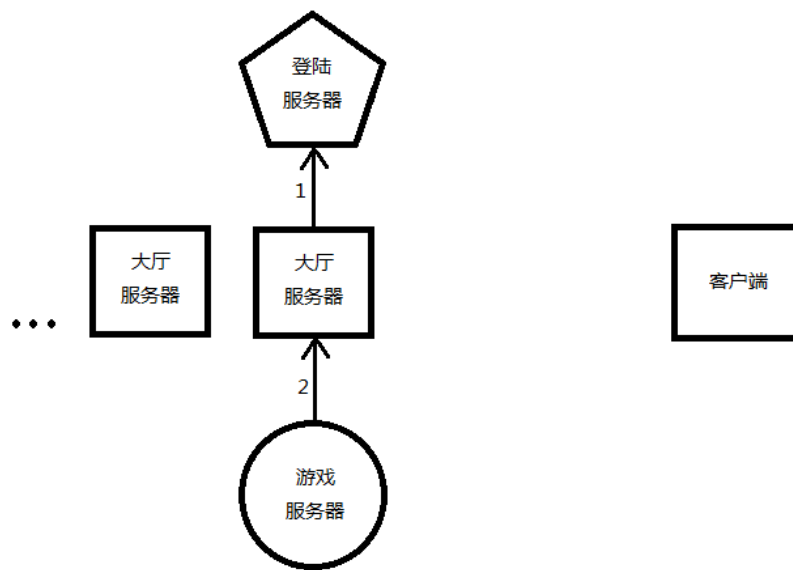


Figure 1: 服务器启动与建立联系

1. 大厅服务器与登陆服务器建立连接
2. 游戏服务器与大厅服务器建立连接

2.2.2 用户登陆

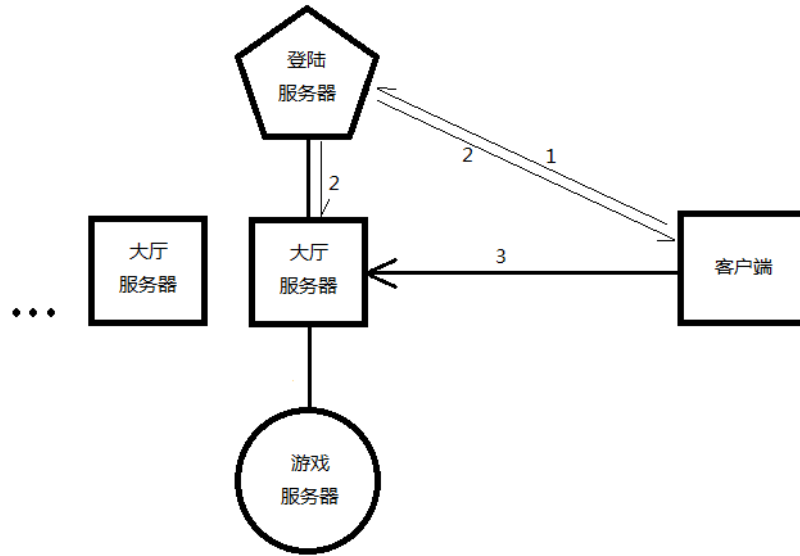


Figure 2: 用户登录

1. 客户端与登陆服务器建立连接，并发送用户名、密码以及希望加入的大厅。
2. 登陆服务器验证用户名密码，验证该大厅是否可以加入。若均满足，则向该大厅发送该客户端即将加入的信息，向该客户端发送大厅的 ip 地址和端口号，并生成一份一次性密码发送给大厅服务器与客户端。
3. 客户端与大厅服务器建立连接，并发送用户名和一次性密码。大厅服务器验证收到的一次性密码，若正确则接受该用户，即保持该连接。

2.2.3 用户在大厅的行为

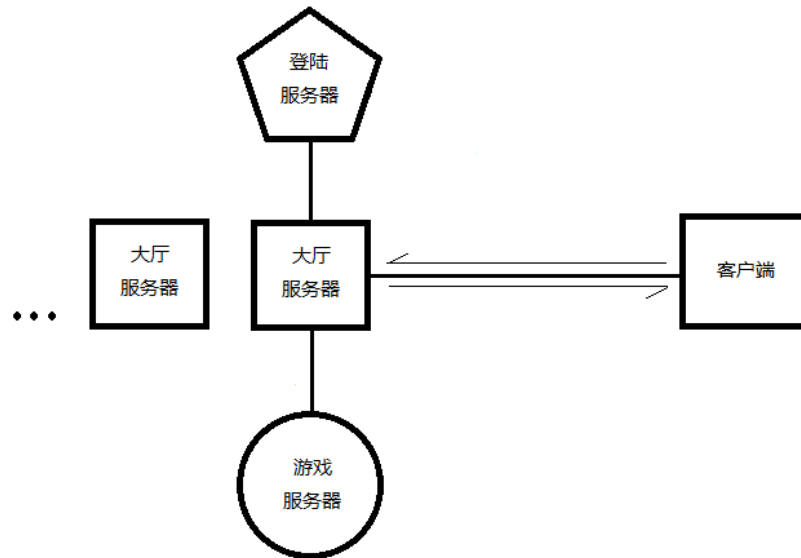


Figure 3: 用户在大厅的行为

1. 该用户进行准备、取消准备、进入房间、离开房间等行为，通过客户端与大厅服务器间的连接将上述行为发给大厅服务器
2. 其他用户有准备、取消准备、进入房间、离开房间等行为，通过大厅服务器与客户端间的连接将上述行为发给客户端

2.2.4 游戏开始

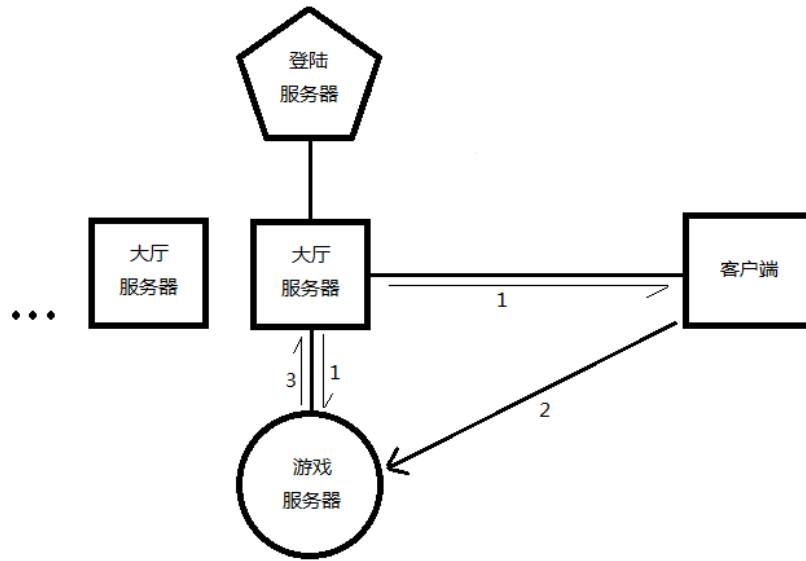


Figure 4: 游戏开始

1. 大厅服务器检测到一局游戏将要开始，将游戏服务器的 ip 和端口号发送给各个客户端，将参与该局游戏的用户资料发送给游戏服务器，并为每个用户生成一个一次性密码发送给游戏服务器和客户端。
2. 客户端向游戏服务器发起连接，发送用户名和一次性密码以验证身份。游戏服务器验证用户名与一次性密码，若正确则接受该用户，即保持该连接，并进行游戏的数据交互。游戏结束，游戏服务器与各个客户端断开连接，并向大厅服务器发送游戏结果。

2.2.5 用户登出

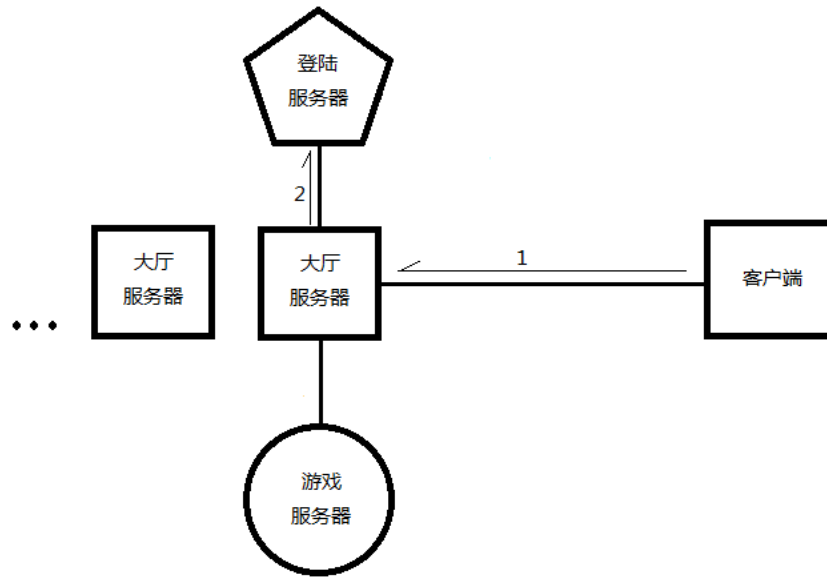


Figure 5: 用户登出

1. 客户端向大厅服务器发送登出消息，大厅服务器与客户端断开连接
2. 大厅服务器根据该用户从登陆到登出的所有事件以及登陆之前的用户数据计算出用户登出时的数据，把更新的数据发送给登陆服务器，登陆服务器在数据库中更新该用户的数据
3. 若客户端掉线（心跳包丢失），视为用户登出，自动进行步骤 2

3 系统需求

简介：该部分针对每个系统详细介绍了它的对外功能，并以 Use Case 图的形式进行展示。

3.1 <客户端逻辑>

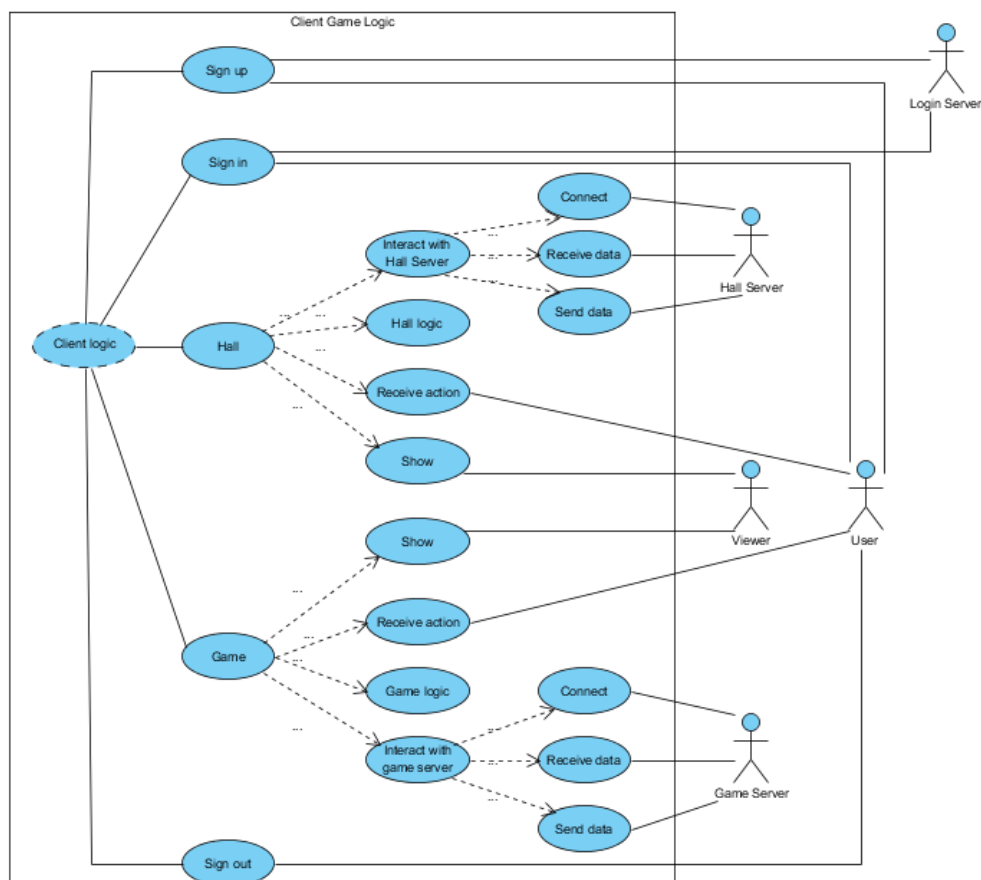


Figure 6: 客户端逻辑

3.1.1 Sign up

1. 响应 User 在显示界面的注册请求，为 Viewer 提供注册接口，接收用户名、密码等信息。
2. 与 Login Server 建立连接，并按照约定好的数据格式将用户名、密码等信息发送给服务器，具体的数据通信格式见 Wiki 页面 Client Register。若返回注册失败，或没有收到返回数据，为 Viewer 提供失败原因，并让 Viewer 显示注册失败。
3. 若返回注册成功，进入 Hall 模块。

3.1.2 Sign in

1. 响应 User 在显示界面的登录请求，为 Viewer 提供登录接口，接收用户名、密码等信息。
2. 与 Login Server 建立连接，并按照约定好的数据格式将用户名、密码等信息发送给服务器，具体的数据通信格式见 Wiki 页面 Client And Hall Server。若返回登录失败，或没有收到返回数据，为 Viewer 提供失败原因，并让 Viewer 显示登录失败。

3. 若返回登录成功，进入 Hall 模块。

3.1.3 Hall

1. connect 模块与 Hall Server 建立连接，并且发送自己的身份信息（登录时 Login Server 提供的一次性密码）。
2. Interact With Hall Server 模块持续监听与 Hall Server 的通信端口，随时接收来自 Hall Server 的信息，并由 Hall Logic 子模块进行相应解析。
3. Interact With Viewer 模块为 Viewer 提供相应接口，响应来自 Viewer 的用户请求（查看玩家信息，进入房间，游戏准备等），并由 Hall Logic 子模块进行相应解析。
4. Hall Logic 解析收到的信息，将 User 的行为（请求）通过 Interact With Hall Server 模块发送至 Hall Server，将修改后的数据通过 Interact With Viewer 模块传给 Viewer 并通知其重绘界面（若玩家在游戏中，则只更新数据，游戏结束后再重绘）
5. 当接收到 Hall Server 发来的游戏开始信息时，进入 Game 模块。

3.1.4 Game

1. Connect 模块与 Game Server 建立连接，并发送自己的身份信息（Hall Server 提供的一次性密码）。
2. Interact With Game Server 模块持续监听与 Game Server 的通信端口，将收到的信息交给 Game logic 子模块处理
3. Interact With Viewer 模块为 Viewer 提供相应的接口，响应用户在显示界面的游戏操作，并交由 Game logic 子模块处理。
4. Game logic 子模块处理收到的信息，将 User 的行为通过 Interact With Game Server 模块传给 Game Server，将修改后的数据通过 Interact With Viewer 模块传给 Viewer 并通知其重绘。
5. 收到来自 Game Server 的游戏结束的消息时，通过 Interact With Viewer 模块通知 Viewer 显示游戏结果，并回到 Hall 模块。

3.1.5 Sign out

1. 响应 User 在显示界面的登出请求，向 Login Server 发送登出消息。
2. 根据 Login Server 的反馈通知 Viewer 更新界面。

3.2 <登陆服务器>

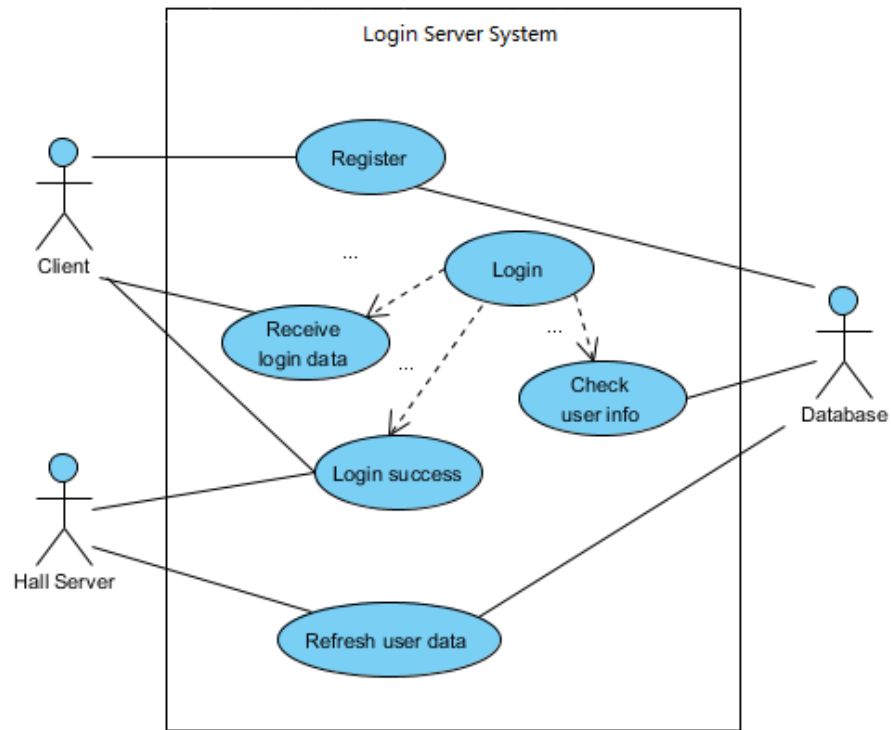


Figure 7: 登录服务器

3.2.1 Register

1. 接收 Client 的注册请求，检验合法后写入服务器。

3.2.2 Login

1. 接收 Client 的登陆请求，具体格式见 Wiki 中 LoginServer And HallServer。
2. 从 Database 中查找该用户的信息，检验登陆请求是否合法。
3. 对于合法的登陆，向 HallServer 发送该用户的信息以及一次性密码，向 Client 发送 Hall Server 的信息以及一次性密码。

3.2.3 Refresh User Data

1. 从 Hall Server 接收到更新用户个人信息的信息，写入 Database。

3.3 <大厅服务器>

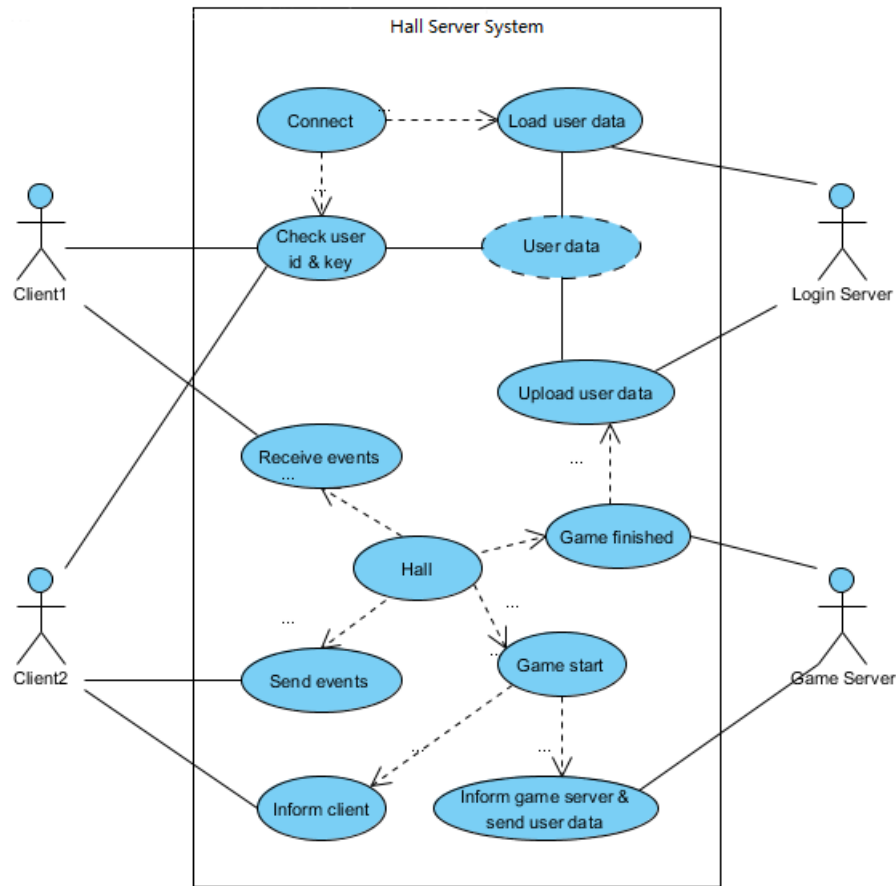


Figure 8: 大厅服务器

1. Connect

- (a) Load user data 模块负责接收即将登陆用户的信息，将用户数据存放于 User data 模块。
- (b) Check user id & key 负责接收来自 Client 的登陆请求并对其进行验证

2. Hall

- (a) Receive events 模块负责监听各个 Client 的操作请求，并交由 Hall 同一处理 Hall 通过 Send events 模块将大厅内发生的事件发送给各个 Client
- (b) 当有一局游戏将要开始时，通过 GameStart 模块将该信息通知相关的 Clients 并将这些用户的数据发送给 Game Server
- (c) 当一局游戏结束时，Game finished 模块会收到 Game Server 发来的结果。由 Upload user data 模块负责将用户数据更新，并发送给 Login Server。

3.4 <游戏服务器>

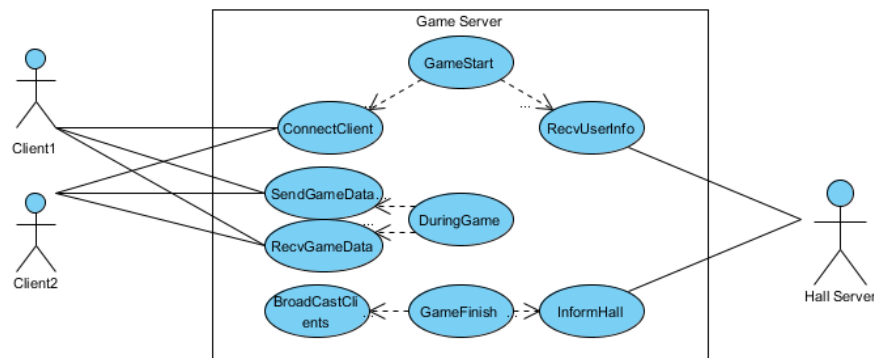


Figure 9: 游戏服务器

- 对于 GameStart, Game Server 负责接收一个大厅服务器的玩家的游戏请求
 - RecvUserInfo, 接收大厅服务器传传给的游戏信息: 每个用户的连接信息, 用户属性等。
 - ConnectClient, 与每一个请求的用户建立 tcp 通信连接。
- 对于 DuringGame, 游戏服务器对每个游戏中的用户建立一组 tcp 连接, 负责:
 - RecvGameData, 接收用户的游戏行为。
 - SendGameData, 每隔 0.1s 发送一次服务器当前的游戏数据, 与用户进行同步。
- 在 GameData 中, 对于用户控制的移动请求
 - 对于 RecvGameData, 用户发送自己的移动方向到服务器, 服务器更新每个用户的当前移动方向 (上下左右停), 并计算下次发送信息时用户更新的坐标。
 - 对于 SendGameData, 在向用户发送的当前游戏数据中, 移动部分包括每个用户的坐标信息以及速度方向。
- 在 SendGameData 中, 对于地图数据
 - 地图数据包括地图的样式, 地图方格中含有的道具, 每个人的出生位置。
 - 地图数据在与用户建立连接后, 游戏开始前, 发送给每一个用户。
 - 每个用户接收到地图数据后, 发送 ACK 信号给服务器, 服务器接收到所有 ACK 信号后准备开始游戏。
- 在 GameData 中, 对于泡泡 (炸弹) 请求
 - RecvGameData, 服务器随时接收用户的炸弹请求, 记录炸弹放置的位置坐标与时间。
 - SendGameData, 在炸弹爆炸时间 (3s) 到达后, 记录爆炸状态, 在当前给用户发送的游戏数据包中告知此次炸弹的爆炸。
- 在 GameLogic 中, 对于胜利
 - 服务器随时记录每个人的生存状态, 当生存人数为 1 或 0 时, 向每个客户端发送游戏结束的信息, 此时场上已存的泡泡将没有威力, 服务器不再更新数据。
 - 剩余人数为 1 时, 向剩余的用户发送游戏胜利的信息, 向其他用户发送游戏失败的信息。

- (c) 剩余人数为 0 时，向所有用户发送平局信息。
- 7. 在 GameFinish 中，对于游戏结束
 - (a) 在 InformHall 中，游戏结束后向大厅服务器发送结束的消息，使得大厅更新用户的数据（经验等）。
 - (b) 在 BroadcastClients 中，向每个用户发送结束的消息，使得用户从游戏状态变为大厅状态。

3.5 客户端显示

1. 登入界面：
 - (a) 图片素材需求：需要良好的登陆背景图片。
 - (b) 需要账号密码输入框，注册和登陆按钮。
2. 注册界面：
 - (a) 图片需求：背景图片
 - (b) 用户名，密码，确认密码，昵称等信息的填写
3. 大厅界面：
 - (a) 图片需求：背景图片，候选房间图片
 - (b) 需要动态显示可加入的房间以及房间状态（是否游戏中），房间人数，显示房间的地图设置等信息，点击房间可以进入
 - (c) 根据具体情况增加一些按钮来查看个人信息，好友信息，设置游戏属性（如背景音乐等）以及其他待扩展的内容（如商城，附近的朋友）
4. 个人信息，好友信息界面：
 - (a) 个人信息界面显示个人的属性信息
 - (b) 好友信息界面以列表形式显示好友信息
5. 房间界面
 - (a) 图片需求：角色图片，聊天栏背景，地图图片
 - (b) 显示房间中的每个角色的信息，以及他们所处的状态（准备与否），并且显示房主
 - (c) 第一个进入的角色为房主，后续加入的需要准备。房主可以按开始按钮开始游戏
 - (d) 房主可以选择地图，其他角色可以看到地图的改变
 - (e) 房间内有聊天区域，房间内部成员可以互相聊天

游戏界面

6.
 - (a) 图片需求：不同角色不同方向，不同姿态的图片，炸弹图片，炸弹爆炸图片，人物被击中图片，火花图片，地图元素图片，游戏结束图片（胜利，失败）
 - (b) 根据后台数据定时刷新各个角色所在的位置及各个泡泡状态，地图状态。
 - (c) 实现各个状态平滑过渡，加入人物行走，放炸弹，炸弹爆炸，人物被击中以及游戏结束等动画效果，优化游戏视觉效果。
 - (d) 适当位置显示倒计时，显示各个角色的状态
 - (e) 游戏结果显示胜利或者失败的效果
 - (f) 游戏结束后显示人物经验成长和奖励信息