

哈尔滨工业大学（深圳）

面向对象的软件构造实践 课程报告

学生班级：_____ 22 级 4 班

学生学号：_____ 220110430

学生姓名：_____ 吴梓滔

评阅教师：_____

报告成绩：_____

实验与创新实践教育中心制

2024 年 4 月

1 需求分析与任务分工

2.1 MainActivity 的按钮监听与跳转

2.2 跳转 GameActivity 并传递难度、音乐开关等数据

3.1 创建对应难度的游戏并设置界面显示为游戏界面

3.2 游戏功能和游戏内的飞行对象、得分等的绘制

4.1 GameActivity 用 handler 处理游戏子线程结束的反应

4.2 GameActivity 跳转到排行榜页面

4.3 排行榜的 activity 中，DAO 的加载，游戏数据存储进 DAO，并能保存虚拟机本地

4.4 排行榜在表格中正确显示

4.5 完善排行榜单机删除功能的逻辑，删除后排名仍正确显示

5.1 实现两个音乐播放管理类，提供音效开关的接口，可在 Basegame 内直接调用。

6.1 联机模式点击按键弹出 AlertDialog

6.2 客户端和服务端通讯，客户端处理服务器发送的信息执行游戏开始、结束和交换分数

2 网络通信方案

1 服务器的行为：

1.1 服务器初始化后，等待客户端连接。

1.2 持续监测连接成功的客户端数量，一旦达到 2 个，即通知两个客户端开始游戏。

1.3 接收客户端的消息，若客户端发来的消息为数字（分数信息），直接发往另一客户端。若客户端发来的消息为“end”，则标记该客户端游戏结束。同时检查客户端数量，若两个客户端都已游戏结束，那么通知两个客户端游戏结束。

1.4 此后可继续等待客户端连接。

2 客户端的行为：

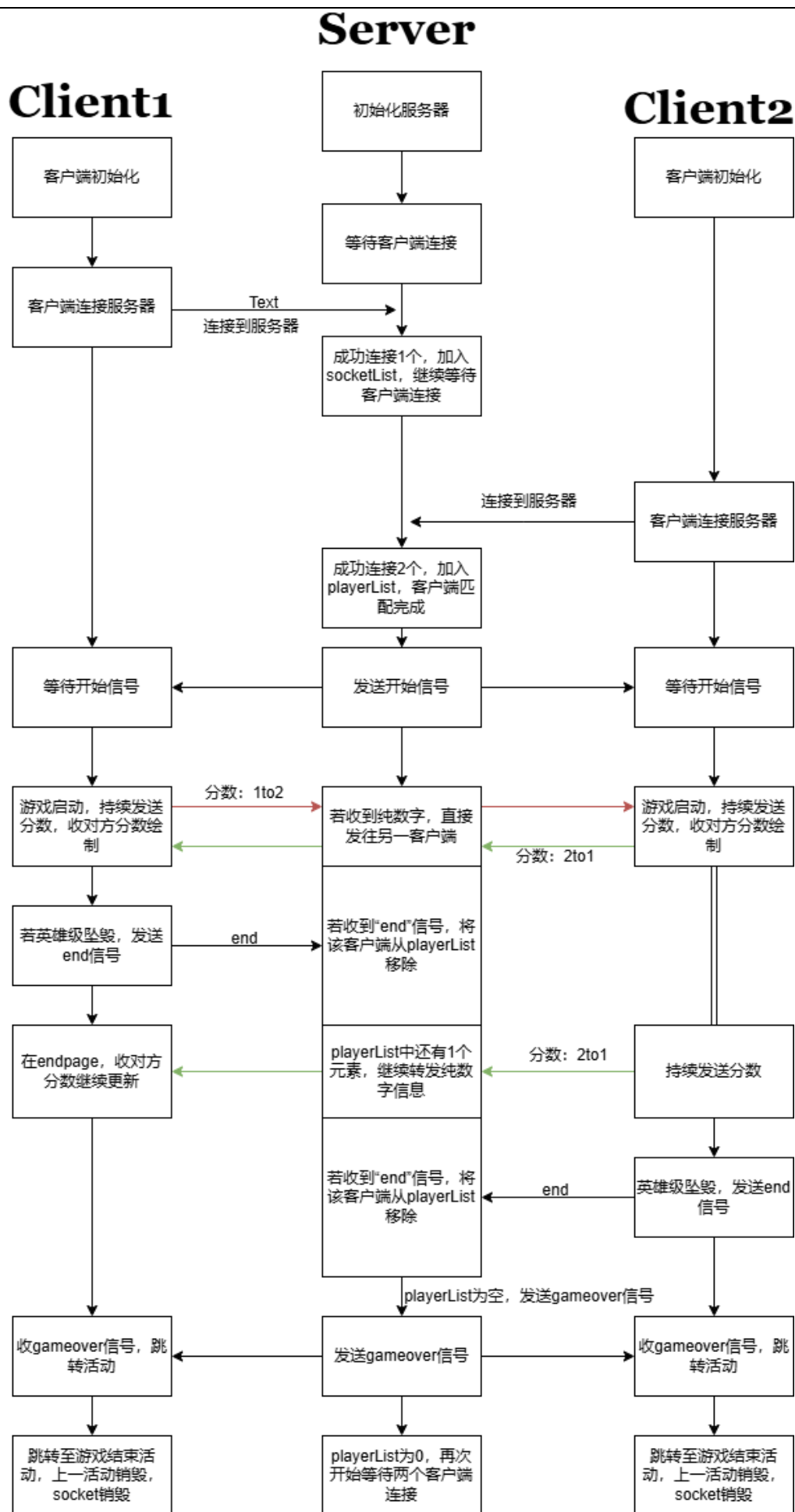
2.1 客户端创建即连接到服务器。

2.2 收到服务器开始游戏信号，立即开始游戏线程。

2.3 持续发送分数到服务器，绘制服务器发来的对方得分。

2.4 一旦游戏结束，即发送“end”到服务器。继续处理服务器发来的对方分数。

2.5 一旦收到服务器游戏结束通知，即跳转结束界面，关闭客户端连接。



3 调试分析

1 在保存排行榜的时候，单机删除某一条数据后，不能够正确地更新排行榜，有三条数据的时候，排名依次为 1，2，3，删除 2 后，剩下两条的排名为 1 和 3。为了正确更新，我经过研究后发现，适配器自动更新视图取决于它创建时绑定的那个数据对象，创建时绑定了一个 `List<map> maps`，后面其他数据结构更新，不会引发视图更新，而这个数据对象又不能更换成别的对象。最终解决是在 `AlertDialog` 的按键触发方法内，将 `maps` 清空并在那里重新装填，实现了排行榜的正确更新。

2 在联机对战的时候，为了实现死亡后先跳到比分界面更新对手分数，设置了一个 `endpage`，并且其中有两个 `TextView` 要在联机对战的 `Activity` 中获取并设置。最初无法进入游戏，后面检查 `logcat` 后，查 `stack overflow` 上面回答，因为获取 `TextView` 必须在 `setContentView` 之后，才能按照 `id` 获取此页面中的控件。

3 联机对战一方飞机死的时候，偶尔能成功，但更多时候卡死。经过查 `logcat`，发现要设置的对对手分数的 `TextView` 控件 `oppoScore` 为空。这是由于，`TextView oppoScore` 的创建和设置不在一个地方，并且都用游戏子线程中的 `gameOverFlag` 来判断是否执行，这导致有可能 `oppoScore` 需要设置的时候，`oppoScore` 还没有获取控件，此时为 `null`。为了保证 `oppoScore` 的设置语句一定在获取到该空间后面，在联机对战的 `Activity` 中增加了一个属性作为游戏结束的标志，共设置对手分数时使用，只有在切换页面后，才把这个标志设置为 `true`，成功解决该问题。

4 联机对战中，一方飞机死亡后直接卡死。这个问题表现在 `handler` 中的 `msg.what` 的值上，联机对战的 `Activity` 中，`handler` 实际上只需要用来处理 `ClientThread` 和主线程的交流，游戏线程和主线程没有交流，`ClientThread` 发往主线程的 `msg.what` 没有差别，所以逻辑上不需要做区分。但是如果不设置 `msg.what` 为 123 这样特殊的值（后来发现只要非 1 就行）就会卡死。最后发现，`handler` 收到的 `msg` 不止来自 `ClientThread` 线程。原来是因为游戏在创建的时候也要读入一个 `handler`，就把 `handler` 同时传入游戏线程和 `ClientThread`。而游戏线程在英雄机坠毁时会用 `handler` 发一个 `what` 为 1 的消息到主线程。如果 `handler` 在处理消息时不判断 `what` 不为 1，就会读到游戏线程发上来的消息，这个消息的 `obj` 为 `null`，无法处理而导致错误。通过这个 `bug`，我终于正确理解了 `handler` 的用法和 `Message` 的 `what` 属性的作用，修改了 `handler` 处理信息时的判断逻辑，功能就正常了。

4 系统核心功能运行结果与分析



图 2-1 游戏首页



图 2-2 单机难度选择



图 2-3 游戏界面



图 2-4 单机模式排行榜



图 2-5 排行榜记录删除



图 2-6 联机模式匹配



图 2-7 联机模式一方死亡后显示界面



图 2-8 联机模式游戏结束界面

游戏启动首页（图 2-1）可以选择音乐开关，两个按钮分别是开始单机游戏和开始联机模式。选择“开始游戏”按钮后进入单机模式的难度选择页面（图 2-2），任选一个后进入游戏界面开始游戏。在游戏界面内（图 2-3）可以显示玩家得分和 HP 值，当英雄级坠毁，则游戏结束，会进入排行榜页面（图 2-4），排行榜可以存储历次游戏记录，并且可以单击删除某条记录。单击某条记录，会弹出一个确认框（图 2-5），点击确定删除后，就会删除该条记录。在排行榜底端有“返回首页”按钮，点击可回到首页。在首页点击“联机对战”会开启联机匹配过程并弹窗提示（图 2-6），匹配成功后进入游戏，某方英雄机坠毁，另一方仍在游戏，那么会进入一个得分页面（图 2-7），此页面持续更新对手得分。双方游戏结束，会跳转结束统计页面（图 2-8），有一个返回按钮，点击可返回首页。

5 总结与建议

收获与体会：

了解了安卓开发的方法和过程，锻炼了调试代码和使用开发工具的能力，在调试过程中加深了对线程的理解。做实验的时候，感觉到安卓开发中做同一个功能的代码东一块西一块的，但是经过调试和理解，开始比较熟悉安卓平台开发的一个模式。Android Studio 开发的过程中遇到错误闪退或者卡死，出现异常，都是在 Logcat 里面，我最早不懂得使用，因为此前在 IDEA 上这些报错信息都是直接在终端会输出。后来还学习了自己设置日志输出来调试。

对课程的建议：

感觉教学设计的不好，实验 4 和实验 6 太难做，也不是实现难度大，但是做起来就是很困难，感觉要干什么要怎么做根本就没说清楚，课上讲完还是不知道要干什么。网络通信的实现最后还是自己到网上去学，验收的时候和老师交流发现可能自己摸索的不是规范的实现方法，但课程也没讲清楚要怎么做。因为上这个课前本来对安卓平台开发的一个模式就没有了解，做实验的过程基本上是为了实现相关功能就针对去学别人如何实现类似功能，最后还是没搞清楚安卓提供的一些东西是什么原理，糊里糊涂用着他就跑通了。安卓提供的功能像 adapter, handler 的原理和使用场景、使用案例能讲细一点就好了，或者能提供一些学习材料让人去系统地自学。