

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 计科 5 班

学生学号： 220110504

学生姓名： 李乐怡

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

（描述每次实验完成的主要功能，列出由自己完成的任务序号，如 1.1, 2.2 等，20 分）

实验一：1.1

创建名为“AircraftWar2024”的项目

实验二：2.2

设计单机游戏模式选择页面；

实现点击首页的“开始游戏”按钮后跳转到单机游戏模式选择页面，同时将音乐开关是否开启作为 Intent 参数进行传递。

实验三：3.1

移植导论代码到项目中，实现飞机大战的基本功能；

实现单机模式难度选择页面到游戏界面的跳转，同时将音乐开关是否开启作为 Intent 参数进行传递。

实验四：4.2

移植导论的文件存储 DAO 部分代码到安卓平台，文件存储在应用程序空间内；

设计开发排行榜页面；

在排行榜界面显示当前游戏难度和游戏数据（包括排名、用户名、成绩、时间）。

实验五：5.1

若音乐开关打开，游戏开始时播放背景音乐，boss 机出现时播放 boss 机音乐，背景音乐暂停；boss 机消失后背景音乐继续，boss 机音乐停止；游戏结束时停止背景音乐和 boss 机音乐的播放。

若音乐开关关闭，进行游戏时不播放任何背景音乐。

实验六：6.1

在游戏首页添加“联机对战”按钮，点击此按钮进入等待匹配对手状态，匹配成功则开始游戏，进入游戏主界面，游戏难度默认为普通模式；

游戏中每隔一段时间向服务器发送自己的得分；

将服务器传来的对方的分数显示在游戏页面；

一方玩家死亡后，已死亡玩家仍能实时更新显示对方的分数；

双方均死亡后，游戏结束，显示对战结果。

2 网络通信方案

（使用流程图或者伪代码方式详细描述网络通信方案，包括客户端和服务端端的连接建立和关闭过程、消息传递过程等，描述使用的数据结构、核心算法等，30 分）

流程步骤及其中每一步使用的具体数据结构、算法如下：

客户端 A	服务器端	客户端 B
	1. 创建服务器端套接字 serverSocket 并绑定到端口 9999 上	
2. 点击联机对战按钮，连接到服务器【→】： 创建子线程连接 Socket 服务器，用 10.0.2.2 和 9999 端口创建 Socket 对象，连接到 Socket 服务器等待匹配对手		
	3. 接受到 A 的 Socket 的连接请求，将该 socket 加入到 List<Socket> 类型的 sockets 中	
		4. 点击联机对战按钮，连接到服务器【←】： 创建子线程连接 Socket 服务器，用 10.0.2.2 和 9999 端口创建 Socket 对象，连接到 Socket 服务器等待匹配对手
	5. 接受到 B 的 Socket 的连接请求，将该 socket 加入到 List<Socket> 类型的 sockets 中，此时 sockets 中有两个元素，将“start”消息传给两个客户端	
6. 【←】接受到来自服务器端的“start”消息，显示游戏界面开始进行游戏来对战		6. 【→】接受到来自服务器端的“start”消息，显示游戏界面开始进行游戏来对战
7. 【→】若当前游戏还未结束，每隔 50ms，传递该客户端的当前成绩给服务器端		7. 【←】若当前游戏还未结束，每隔 50ms，传递该客户端的当前成绩给服务器端

	8. 接受 A、B 的当前成绩，分别传给 B、A 客户端进行通讯	
9. 【←】接受到来自服务器端的客户端 B 当前成绩的消息		9. 【→】接受到来自服务器端的客户端 A 当前成绩的消息
反复进行 7、9	反复进行 8	反复进行 7、9
10. 【→】当前游戏结束，传递“end”消息给服务器端		
	继续进行 8 11. 将 A 的 socket 从 sockets 中移除，此时 sockets 中只有一个元素	反复进行 7，不进行 9
12. 【←】接受到来自服务器端的客户端 B 当前成绩的消息		反复进行 7，不进行 9
	继续进行 8	反复进行 7，不进行 9
		13. 【←】当前游戏结束，传递“end”消息给服务器端
	14. 将 B 的 socket 从 sockets 中移除，此时 sockets 为空，给两个客户端传递“gameover”消息，通信结束，关闭两个套接字	
15. 【←】接受到来自服务器端的“gameover”消息		15. 【→】接受到来自服务器端的“gameover”消息
16. 关闭套接字 socket.close()		16. 关闭套接字 socket.close()
17. 跳转到对战结果页面，并将自己的成绩和对手的成绩传递到该 ResultActivity，进行显示		17. 跳转到对战结果页面，并将自己的成绩和对手的成绩传递到该 ResultActivity，进行显示

3 调试分析

（测试时出现问题的调试分析过程，模块设计及算法设计存在的问题及改进方法，20 分）

问题一：排行榜无法进行删除

存在问题：我所负责的 DAO 层删除操作的算法设计有问题：将文件中的数据依次读取出来，存到 List<Score>中，当读出来的 score 与需要删除的数据完全相同时，不将这一条 score 存入 List<Score>中。完成后将 List<Score>里的数据存入新的文件中，将旧的文件删除，将新的文件改名为旧文件的名字，过于复杂。

解决方法：List<Score>填入完成后的文件操作需要进行修改。将文件输出流 `FileOutputStream fop2= context.openFileOutput(file, MODE_PRIVATE)` 设为将文件内容清空再讲数据输出到文件中，以此来实现删除后排名等数据的更新。

问题二：两个客户端成功匹配后，均无法从首页页面跳转到游戏界面

存在问题：属于模块设计问题：在初次设计客户端时，将匹配成功后显示游戏界面的 `setContentView(game)` 字段放在了 MainActivity 的 `onClick()` 中，同时其余联机对战相关的客户端和服务器的连接、消息传递过程也都在 MainActivity 的 `onClick()` 中。但是 `setContentView(game)` 只有在 `onCreat()` 中才会生效、显示对应页面。

解决方法：MainActivity 的 `onClick()` 中若监听到被按下的按钮为联机对战按钮，则跳转到 OnlineActivity 中进行联机对战相关的所有操作（包括匹配成功后显示游戏界面的 `setContentView(game)`）

问题三：第一个玩家死后界面仍然可以动，分数也可以继续加；而在另一个玩家处对手分数始终不变，停留在对手死亡时的成绩。

存在问题：在之前的实验中对英雄机坠亡时利用 handler 来进行消息传递部分的代码放在了 GameActivity 里，这一部份是实现单机游戏结束后跳转到排行榜页面。

解决方法：把上述部分代码放到 BaseGame 里的 `postProcessAction()` 后处理方法中，判断若英雄机坠毁后，将是否绘制游戏页面的 `mlsDrawing` 置为 `false`，并且判断 `OnlineActivity.isOnline` 为 `false` 时再进行上述消息传递。在 BaseGame 的 `run()` 方法中，当 `mlsDrawing` 仍未 `true`、玩家还没死时，会调用 `action()` 方法和 `draw()` 方法，继续进行游戏、绘制更新视图，而当 `mlsDrawing` 变为 `false` 时，会判断当前是否还未两个玩家都结束游戏、即 `OnlineActivity.bothOnline` 为 `true`，则调用 `draw()`，停止进行游戏，只会更新

对手成绩的视图。

问题四：联机对战开始时英雄机不能显示，需要点击一次才会出现

存在问题：初次进行画面的绘制时，尤其是绘制英雄机，需要获得界面的宽度和高度，而只在 `GameActivity` 中才能获取宽高，我的框架是点击联机按钮后跳转到 `OnlineActivity`，在里面进行游戏画面绘制以及游戏，未跳转到 `GameActivity` 中。

解决方法：在 `OnlineActivity` 中添加 `getScreenHW()` 获取屏幕的宽高存到 `GameActivity.screenWidth` 和 `GameActivity.screenHeight` 中

4 系统核心功能运行结果与分析

【备注：由于截止日期前几天机房一直不开门，只能请同学用他们的电脑帮忙截图，但是同学的 gradle 配置出了点问题，有的界面可能状态栏不太一样。】

主界面：



单机游戏选择界面：



单机游戏界面：



单机游戏结束后排行榜界面：



联机对战界面及对战结束后结果界面：



5 总结与建议

（课程任务完成过程中的收获和体会，对课程设计的想法和建议，10 分）

收获：我学会了怎么利用 AS 设计简洁的界面，切实体会了 java 语言面向对象开发与安卓开发之间的关系。实验四的排行榜设计和显示、实验五的音乐播放，让我了解了后端与前端的协同工作关系。实验六：通过和同伴分别编写客户端和服务端端的代码，我了解并掌握了如何开启子线程进行网络通行，并使用 Handler 在子线程和主线程中进行通信，以此利用并发操作来提高速度；以及学会了如何利用 socket 套接字进行数据传输通信。

建议：无。