# 浙江大学计算机科学与技术学院

# Java程序设计课程报告

2020-2021学年秋冬学期

题目: 坦克	大战游戏的设计与开发
学号:	
学生姓名:	
所在专业:	
所在班级:	

# 目录

1	引言	1
	1.1 设计目的	. 1
	1.2 设计说明	. 1
2	总体设计	1
	2.1 功能模块设计	. 1
	2.2 流程图设计	. 2
3	详细设计	3
	3.1 服务端设计	. 3
	3.1.1 GUI设计	. 3
	3.1.2 数据库设计	. 3
	3.2 客户端设计	. 4
	3.2.1 坦克设计	. 5
	3.2.2 子弹设计	. 6
	3.2.3 网络连接设计	. 6
	3.2.4 消息协议设计	. 7
4	测试与运行	8
	4.1 程序测试	. 8
	4.2 程序运行	. 8
5	总结	11

## 1. 引言

本次开发的是一个多人坦克大战游戏,这是一个综合性的题目,完成题目后可以对Java语言中的各项功能有更好的理解和使用,加深对Java语言的掌握,提高自己的编程水平,为以后的工作打下一定的基础。

#### 1.1 设计目的

坦克大战是一个经典的游戏,本文使用Java语言编写一个多人联机版的坦克大战,具体功能如下:

- (1) 游戏分为服务端和客户端,需要先在服务器上启动服务端程序,然后玩家可以在客户端上 连接服务器参与游戏,连接到同一个服务器的玩家参与到同一局游戏
- (2) 服务端会记录用户的连接信息,包括ip,端口,以及连接时间
- (3) 玩家可以在客户端进行操作攻击其他玩家的坦克,如果玩家坦克的生命值降为0就会提示游戏结束,可以选择退出游戏

#### 1.2 设计说明

本程序采用Java程序设计语言,在IDEA IntelliJ下编辑,编译与调试,由单人独立开发完成。

## 2. 总体设计

#### 2.1 功能模块设计

本程序主要实现的功能有:

- (1) 服务端在服务器建立游戏服务并用数据库存储客户端连接信息
- (2) 客户端可以通过连接到服务器来加入一局游戏
- (3) 玩家可以通过键盘操作来控制坦克移动,发射子弹,攻击其他坦克
- (4) 坦克有一定量的生命值,被其他坦克的子弹击中后生命值会减少,生命值降为0则不能继续参与游戏,只能通过再次连接服务器来重新加入游戏
- (5) 多个玩家可以并发操作,游戏画面在多个客户端上同步

#### 总体功能图示:

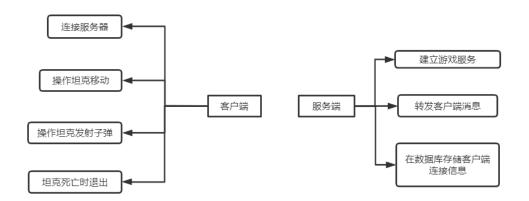


图 1: 总体功能

## 2.2 流程图设计

服务端流程图:

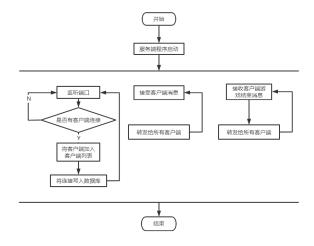


图 2: 服务端流程图

客户端流程图:

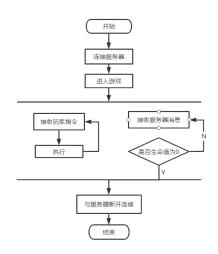


图 3: 客户端流程图

# 3. 详细设计

## 3.1 服务端设计

#### a. GUI设计

GUI类继承JFrame类,内含一个实现了Runnable的重画线程类

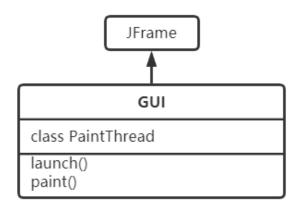


图 4: 服务端GUI类

其中,paint方法定义了图形界面每次刷新重画时的具体操作,launch方法会设置GUI的一些初始值,包括title,size,background等等,然后启动重画进程。

#### b. 数据库设计

数据库只用到了一张表

id ip port time

图 5: connections表

其中id是客户端的id, ip是客户端的ip, port是客户端的连接端口, time是与客户端建立连接的时间。

#### 3.2 客户端设计

客户端类图:

```
Client

public final int GAME_W = 800;
public final int GAME_H = 600;
private Tank mytank;
private final List<Tank> tanks = new ArrayList<>();
private final List<Bullet> bullets = new ArrayList<>();
private final NetClient nc;
private final ConnectDlg connectDlg;
private final GameOverDlg gameOverDlg;
private final ServerUnstartedDlg serverUnstartDlg;
public void gameOver()
public ServerUnstartedDlg getServerUnstartedDlg()
public List<Tank> getTanks()
public Tank getMyTank()
public NetClient get_nc()
```

图 6: 客户端类图

其中

- GAME\_W, GAME\_H是客户端GUI的宽和高
- mytank是客户端玩家自己的坦克
- tanks是其他所有的坦克
- bullets是游戏中所有的子弹
- nc是客户端与服务器的网络连接
- connectDlg, gameOverDlg, serverUnstartedDlg是用到的对话框(连接服务器,游戏结束确认,提示服务器未运行)
- gameOver方法用于处理坦克死亡游戏结束

客户端中的关键设计有: 坦克设计,子弹设计,网络连接设计,消息协议设计,下面会其中比较关键的部分进行解释(出于篇幅考虑,较平凡的细节会被略去)

#### a. 坦克设计

坦克在游戏中一共有8个方向:



图 7: 坦克设计

#### 坦克类图 (部分):

```
private int id;
private static final int vx = 5;
private static final int vy = 5;
private final String name;
private int x, y;
private boolean live = true;
private Direction mv_drc = Direction.STOP;
private Direction drc = Direction.D;
private Direction drc = Direction.D;
private Direction drc = Direction.D;
private int HP = 100;
private final HealthBar health_bar = new HealthBar();
private boolean btnL, btnU, btnR, btnD;
Client player;
private static final Image[] images;
private static final Map < Direction, Image > map = new HashMap <> 0;
private static int WIDTH = images[0].getWidth(null);
private static int HEIGHT = images[0].getHeight(null);
public Tank(String_name, int_x, int_y, String_name)
public Tank(String_name, int_x, int_y, Direction mv_d, Client p)
public void draw(Graphics g)
public void keyPressed(KeyEvent ke)
public void keyPressed(KeyEvent ke)
public void gfire()
public void got_hit(HitEvent hit)
```

图 8: 坦克类图

#### 其中

- id就是单纯的坦克id
- vx, vy是坦克沿X方向移动和沿Y方向移动的速度
- x, y是坦克的位置坐标, name是用户为自己心爱的tank起的名字
- live用来标识坦克是否存活, HP是坦克的当前生命值
- 四个btn用来标识四个方向键的活跃情况
- player是坦克所属的客户端
- mv\_drc是坦克的移动方向,drc是坦克的当前朝向(开火方向)

- images中存储了坦克8个方向的图片,WIDTH是坦克的宽,HEIGHT是坦克的高
- draw方法是坦克绘制的方法,会由客户端调用
- move方法用于坦克移动, orient方法用于移动后定向, fire方法用于发射子弹
- keyPressed和keyReleased两个方法分别用于响应键按下和键释放
- got\_hit方法用于处理坦克被子弹击中的事件

#### b. 子弹设计

子弹类图 (部分):

```
Bullet
private static final int vx = 10;
private static final int vy = 10;
public static final int DMG = 20;
private static int ID = 10;
private int id;
private int x, y;
private final int tank id;
private Client player;
private final Direction drc;
private boolean live = true;
public Bullet(int tank id, int x, int y, Direction d)
public Bullet(int id, int x, int y, Direction d, Client p)
public void draw(Graphics g)
public void move()
public boolean hit(Tank t)
```

图 9: 子弹类图

#### 其中

- vx, vv是子弹在X, Y方向的移动速度
- x, y是子弹的位置坐标
- 子弹只有移动方向drc而且不可变
- DMG是子弹的伤害
- draw方法用于子弹绘制
- hit方法接受一个坦克参数,用于处理子弹是否击中某个坦克的相关事务

#### c. 网络连接设计

NetClient类图(部分):

# private static final int BAOS\_SIZE = 128; private final Client player; private final Client player; private String SERVER\_IP; private int SERVER\_UDP\_PORT; private int TANK\_DEAD\_UDP\_PORT; private DatagramSocket ds = null; private final int BUF\_LEN = 1024; public class UDPThread implements Runnable public NetClient(Client p) public void connect(String ip, int port) private int getRandomUDPPort() public void send(Msg msg) public void sendDisconnectMsg()

图 10: NetClient类图

#### 其中

- BAOS\_SIZE是字节数组输出流的缓冲区大小
- player是连接所属的客户端
- BUF\_LEN是数据包缓冲区的大小
- connect方法用于于服务端建立连接
- send方法用于向服务端发送消息
- sendDisconnectMsg方法用于向服务器发送断开连接的消息
- UDPThread是用于与服务端维持交互的线程类

#### d. 消息协议设计

#### 消息接口:

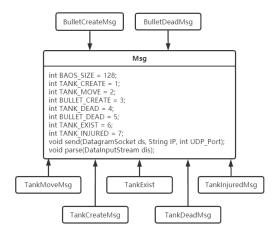


图 11: 消息接口

定义了7种不同的消息,分别对应:新坦克创建,坦克移动,新子弹产生,坦克阵亡,子弹被销毁,坦克已经存在(重复创建坦克时),坦克受伤。

继承Msg接口需要实现send和parse两个方法,用于向服务器发送消息和从服务器接收消息。

## 4. 测试与运行

#### 4.1 程序测试

在程序代码基本完成后,经过不断的调式与修改,最后测试本次设计的坦克大战游戏可以 正常运行,在基本功能与画面上与预期差距较小,没有出现严重的错误,但在一些细节方面仍 然存在着可以进一步完善或者改进的地方,比如为子弹击中坦克添加火光特效,改进刷新机制 让画面看起来更流畅,增加计分排名机制等等,鉴于时间紧迫,只能先保证基本的功能需求, 锦上添花的细节留给日后。

#### 4.2 程序运行

服务端运行界面:



图 12: 服务端运行界面

服务端的GUI界面会显示出服务器当前时间,当前连接的客户端,以及至今所有的连接。 服务端的GUI界面是可选的,可以通过修改源代码禁用GUI:

public static void main(String[] args){
 GUI gui = new GUI();
 gui.launch();
 start();

客户端运行界面:



图 13: 客户运行界面

客户端运行时会先弹出一个对话框提示连接到服务器,可以手动输入要连接的服务端IP。

点击connect后,连接成功,客户端界面上会显示一个坦克,而服务端界面上也会出现相应 的连接信息:



图 14: 客户端连接服务端

客户端可以通过按键实现移动,发射子弹:



图 15: 玩家按键操作坦克

再启动一个客户端并连接,可以看到两个客户端已经在同一局游戏中,可以看到对方,而 服务端上也更新了相应的信息:



图 16: 实例1

操纵一辆坦克向另一辆坦克射击,另一辆坦克被击中后生命值减少:



图 17: 坦克攻击其他坦克

继续射击指导对方坦克阵亡,然后阵亡的一方客户端弹出Game Over对话框,服务端显示 该客户端已断开连接:如果在服务端未启动的情况下强行连接,就会有服务端未启动的对话框

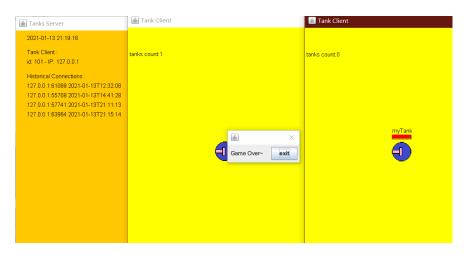


图 18: Game Over

#### 弹出:



图 19: 在服务端未启动的情况下强行连接

# 5. 总结

这个游戏主要的难点在于设计合适的消息协议,使每个客户端的操作对其他的客户端都是透明的,这样才能实现多人游戏,而服务端只起到转发消息和存储数据的作用,相当于一个中转站,而消息的及时转发使得各个客户端上的信息保持同步。

完成这个作业之后,收获还是不小的,不限于对网络编程技术的掌握,对Java图形编程的熟悉,对JDBC接口的了解,以及对多人游戏设计思路的初步思考,同时,由于在写程序的过程中出现了很多问题,自己收集错误信息,调试,查阅资料的能力也得到了提升,在网络上可以找到很多类似的项目,阅读他人的代码也让自己有了更多的想法。

总体来说,这次作业的体验还是很不错的,遗憾在于时间紧迫,还有很多可以做的更好的 地方,希望可以在之后的学习中做的更好。

# 参考文献

[1] [美]梁勇. Java语言程序设计(进阶篇)[M]. 机械工业出版社,2017.