

Developing a Java Game from Scratch

202220008 刘睿哲

1. 南京大学

E-mail: 1260898910@qq.com

摘要 本文主要介绍了我在《Java高级程序设计》课程中的大作业，主要设计了一个类似吃豆子的游戏，包含单人、双人联机游戏，游戏的保存和加载功能

关键词 Java, Pacman, 多线程, 联机, 保存, 加载

1 开发目标

本次大作业主要开发了一个类似吃豆子的游戏。在开始界面通过Enter、S、C键可分别进行单人游戏、双人联机的Server端和双人联机的Client端；在游戏中，可通过WASD或上下左右键进行移动；玩家需要吃完所有小怪，然后打到出现的boss，单人模式吃全部豆获胜，用时越短得分越高。双人模式下两人一起打boss。单人游戏过程中按Enter键可以进行保存。

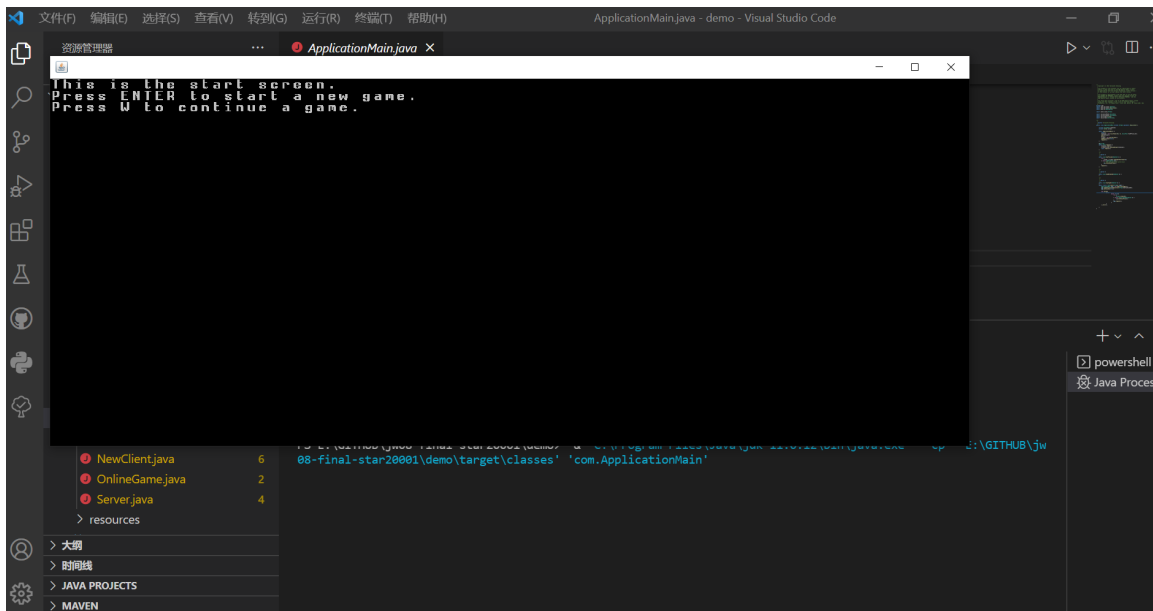


图 1 游戏开始界面
Figure 1 Start Screen

2 设计理念

UI基于AsciiPanel; Screen文件夹下的OneScreen主要负责游戏界面的显示,初始化各种对象和解析收到的消息; World文件夹下的Map主要负责解析和生成地图,统计地图中豆子的总数,World负责实现Player和Ghost的移动和由移动引发的各种事件,如碰撞,得分等; Server文件夹主要包含Server, ServerListen, Client, 负责网络通信和监听。

3 技术问题

3.1 并发控制

由于每个生物体都是一个线程,且两个生物体不能占据同一个tile,对同一生物体的两个攻击行为应该先后发生作用等要求,所以用synchronized关键字修饰World.setTile()函数,该函数主要负责实现Player和Boss的移动和由移动引发的各种事件,避免造成冲突。

3.2 网络通信

联机模式我设定是只有两个玩家。一开始的设想是写一个服务器,两个客户端,游戏的全部操作都是在服务器上完成,服务器将数据信息发给客户端,客户端将其输出显示出来。后来发现这样设计服务器传输的数据过大,且不好读取。在判断这些数据中有众多的重复数据产生。最后我决定把地图信息固定,两个客户端的地图信息完全一致。然后怪物的线程在服务端运行,并且每次刷新都将怪物的状态数据发送给客户端。而客户端将其自己的操作发给服务端服务端进行操作,然后将彼此结果发给客户端。通信方面一开始想要使用文件进行操作,即服务器把位置信息等都写进txt文件中,然后客户端对txt文件进行读取,最后总是有错误产生。中期只能退而求此次,改写成了打擂台模



图 2 单人游戏界面

Figure 2 Single Player Screen

式+socket通信模式。每次客户端通关后将其得分发送给服务端,服务端读取后和本地txt文件(记录排名)比较,最后判断玩家是否在前十名当中。这只是简单通信。最后实现了双人共同合作的游戏模式,打开applicationmain和onlinegame两个文件并且运行,即可同时作战了。

3.3 游戏保存/加载

进行单人游戏时候按下Enter键可以进行保存,将保存地图信息和游戏进度,重新加载后玩家将回到现有位置。设计原理是将游戏信息写入本地的txt文件中,每次加载时候读取txt文件。只能保存一份。

: 引用的标题

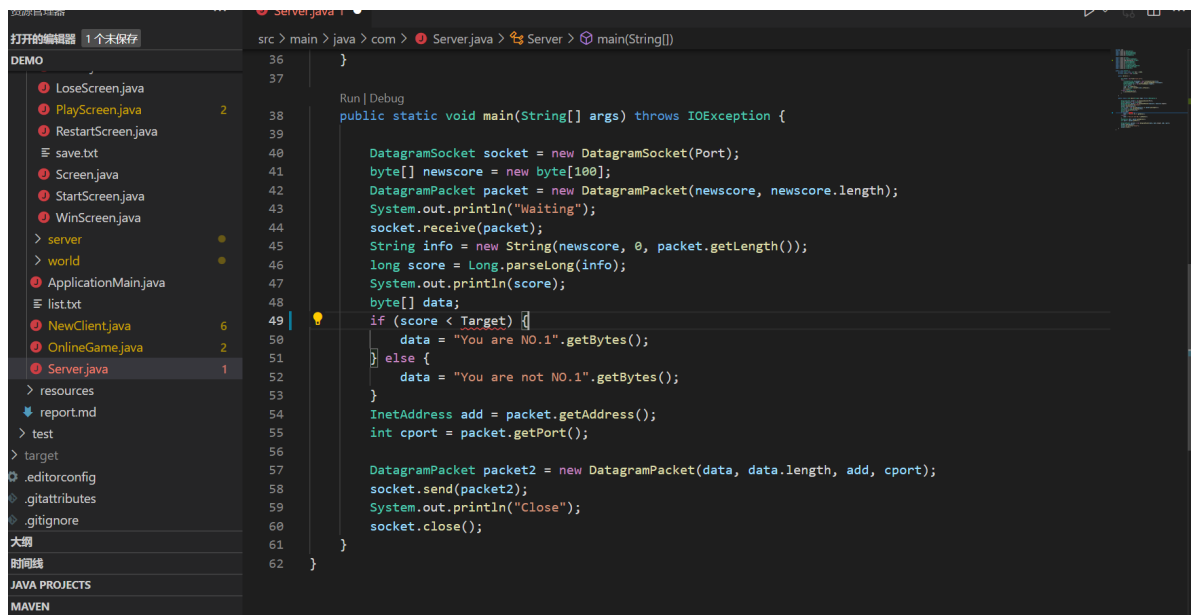


图 3 双人联机游戏界面
Figure 3 Two-Player Online Screen

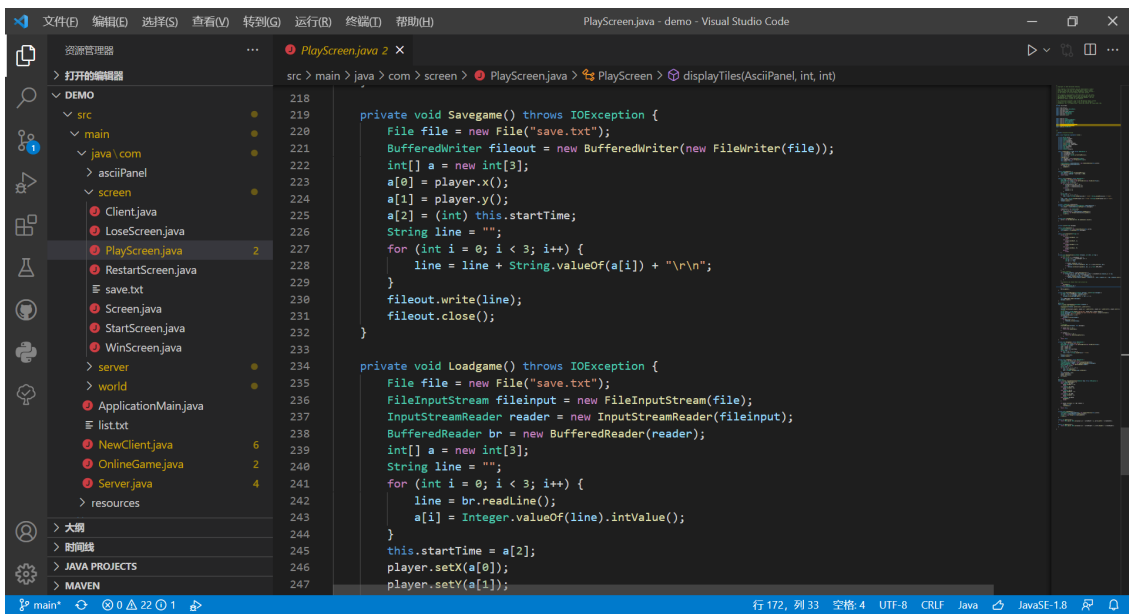


图 4 游戏加载机制
Figure 4 Playback

```
public void setplayer2() {
    CreatureFactory cf = new CreatureFactory(this.world);
    this.player2 = cf.newPlayer(this.message2);
}

public void moveplayer2(String s) {
    switch (s) {
        case "W":
            player2.moveBy(0, -1);
            break;
        case "S":
            player2.moveBy(0, 1);
            break;
        case "A":
            player2.moveBy(-1, 0);
            break;
        case "D":
            player2.moveBy(1, 0);
            break;
        default:
            break;
    }
}
```

图 5 通信雏形

Figure 5 Socket

4 工程问题

4.1 Maven

主要通过Maven进行项目构建和管理,通过Maven对项目进行管理非常方便好用,不用自己手动导入Jar包等。

5 技术问题

主要遇到的技术问题就是通信问题了。能够实现两个终端之间发送信息,但还是不知道应该怎样让两个客户端状态同步,可能同步的话需要另外写个简单的游戏会好点。还遇到过一个问题,就是在main里面不能用变量的问题,会一直报错,后来发现是函数调用的错误。目前关于通信部分,如图所示,我在每个客户端里面的playscreen修改,添加一个playerE,然后可以根据通信输入的数据在外部进行操控,操控其行动。然后每次repaint时候客户端会把怪物和玩家位置信息打印下来存在txt中,由服务端读取后进行统一。

```
public void wirtedown() throws IOException {
    File file = new File("saveai.txt");
    BufferedWriter fileout = new BufferedWriter(new FileWriter(file));
    int[][] a = new int[4][2];
    for (int i = 0; i < 4; i++) {
        if (creaturelist[i].hp() > 0) {
            a[i][0] = creaturelist[i].x();
            a[i][1] = creaturelist[i].y();
        } else {
            a[i][0] = -1;
            a[i][1] = -1;
        }
    }
    String line = "";
    for (int i = 0; i < 4; i++) {
        line = line + String.valueOf(a[i][0]) + "\r\n" + String.valueOf(a[i][1]) + "\r\n";
    }
    line = line + String.valueOf(player.x()) + "\r\n" + String.valueOf(player.y()) + "\r\n";
    fileout.write(line);
    fileout.close();
}
```

图 6 通信雏形
Figure 6 Socket2

6 课程感言

Java高级程序设计这门课很好很强大，我学会了很多，也发现了很多不足。比如，每次大作业添加新功能的时候，都会发现前面写的好烂，每次都要改好多。通过这门课不仅增强了我的代码能力，也提高了我的思维方式。最后不得不说拖延症真的害死人，能早点完成的还是要早点完成。

参考文献

1 Author A, Author B, Author C. Reference title. Journal, Year, Vol: Number or pages
2 Author A, Author B, Author C, et al. Reference title. In: Proceedings of Conference, Place, Year. Number or pages

Title

1

1. *Nanjing University*;
;

E-mail: 1260898910@qq.com

Abstract This article introduces my assignment in "Java Advanced Programming" course, which mainly designs a EatingDOUZI-like game, including single-player mode and two-player online mode, game saving/loading and other functions.

Keywords Java, Pacman, Multithreading, Online, save, load