中国科学：信息科学

SCIENTIA SINICA Informationis

论文

Developing a Java Game from Scratch

严思远[1]

1. 南京大学计算机科学与技术系, 南京 210046
E-mail: 191220137@smail.nju.edu.cn

收稿日期: 2021-xx-xx; 接受日期: 2021-xx-xx; 网络出版日期: 2022-xx-xx

国家自然科学基金 (批准号: 0000000, 0000000, 00000000)

摘要 本文主要介绍高级 Java 程序设计的课程大作业。该项目基于 AsciiPanel，实现了一个既支持本地双人对战，又支持网络四人对战的简易版泡泡堂游戏.

关键词 Java, Rougelike, 并发编程, 网络通信, 序列化存储

1 开发目标

1.1 游戏灵感：

项目灵感来源于小时候玩过的 Q 版泡泡堂。游戏中玩家可以控制角色上下左右移动并放置炸弹，炸弹会炸毁四周的墙壁以及炸死玩家，因此玩家的目标就是通过放置炸弹来破坏墙壁从而更快地到达目标地点，以及借助炸弹击败敌方玩家。原版游戏中还可以拾取各类道具来强化玩家，但是很遗憾本项目只实现了基本的移动和放置炸弹功能，有待完善。

1.2 游戏界面设计：

沿用 jw04 的 AsciiPanel 框架，所有图案均采用 AsciiPanel 中提供的。由于 AsciiPanel 的图案大小为 9×16，相对较小，因此为了保证游戏不会因为地图太大而导致多余操作过多（减少跑图时间），将地图 SIZE 大小设置为 20×20，即共由 20×20 的 AsciiPanel 图案组成游戏界面。游戏中不同的玩家控制相同图案但不同颜色的角色，玩家放置的炸弹具有相同的外观并对所有玩家均可造成伤害。

效果如1所示。

1.3 地图生成：

沿用自 jw04 的 MazeGenerator，但为了保证多人玩家进行游戏时玩家的出生点位都是可到达的，我对其生成机制做了一点小的修改，将玩家的出生点位加入了初始的 Stack 中，从而保证生成地图时从出生点位开始遍历并生成树状结构的地图。

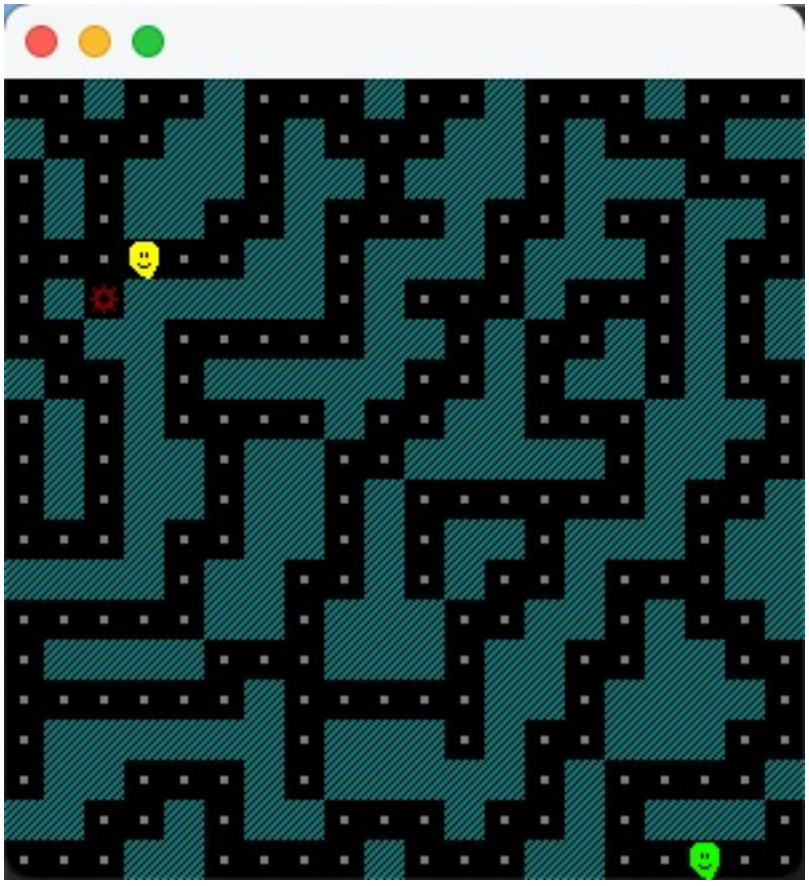

图 1 UI 展示

1.4 网络通信:

游戏主要采用帧同步，客户端从服务端接受状态信号，并根据该信号对游戏内容作出相应改变。客户端负责游戏内逻辑判断等内容，并将本地玩家的操作发送给服务端，而服务端只负责接受和发送各客户端的消息，同时对房主玩家的游戏内容做出改变。简而言之服务端相当于房主本地客户端与信号中转站的结合体。游戏本身并未要求多人同时开始，只要服务端在线（即房主在线），其他玩家即可随时加入并进行游戏。

1.5 游戏存档:

游戏内实现了快速存档/加载存档功能 (QuickLoad/Save)，通过对控制游戏内容的 World 类及涉及到的相关类实现序列化，从而达到保存游戏内容的目的，同时加载存档时只需要读入存档文件并以此初始化 World 对象即可。

2 设计理念

1. World 相关类的设计如2所示。

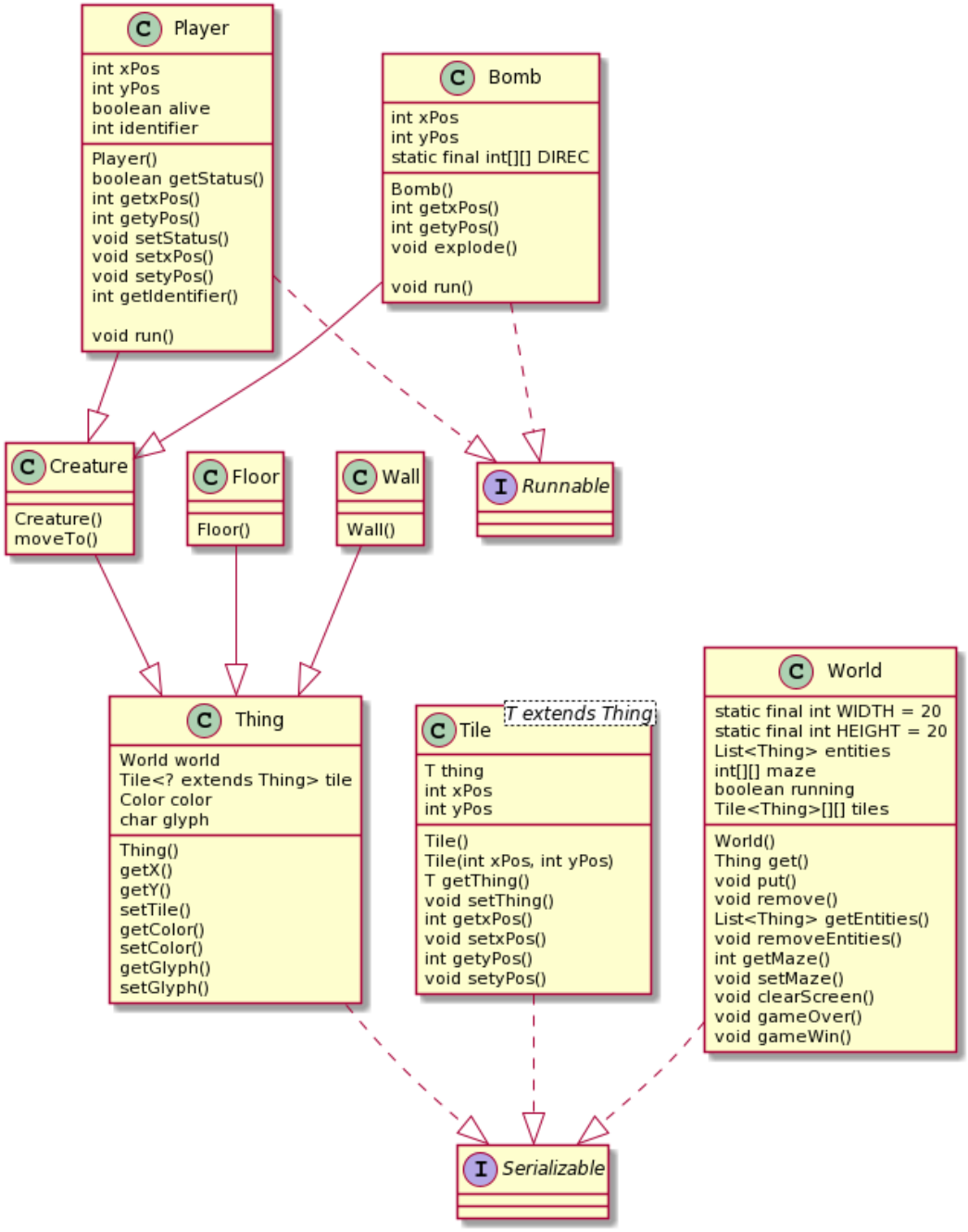

图 2 World 展示

2. UI 相关类的设计如3所示。

3. 网络通信部分采用 NIO Reactor 模式，相关类的设计如4所示。

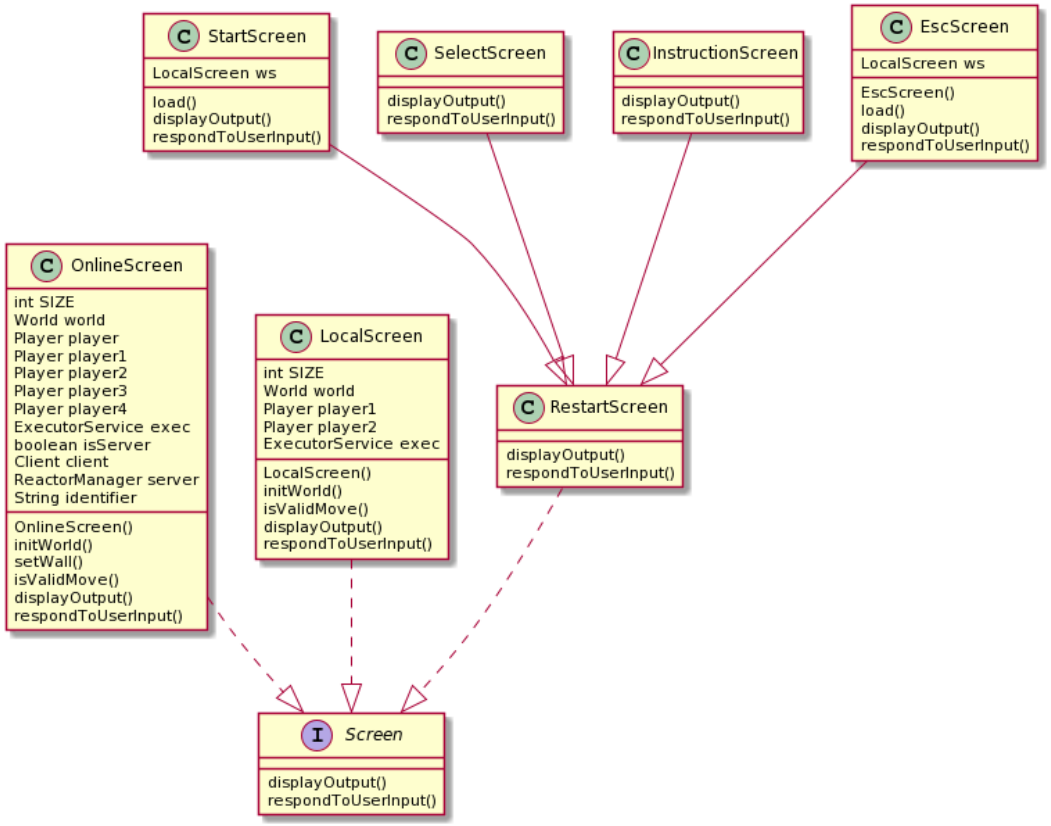

图 3 Screen 展示

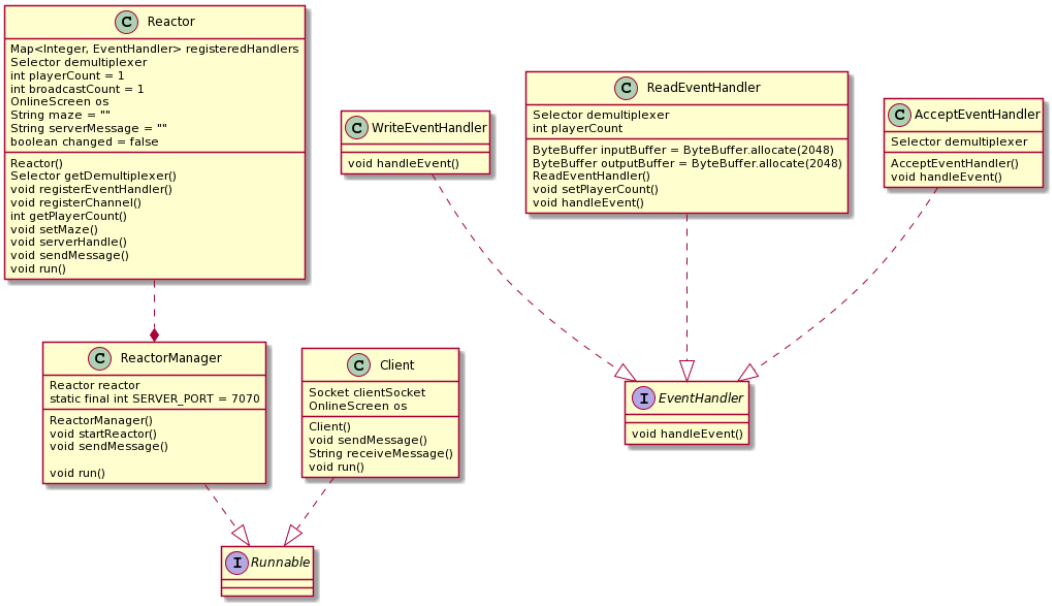

图 4 Reactor 展示

3 技术问题

3.1 并发控制

由于 World 中进行游戏时，两个不同的对象可能会同时移动到同一个位置，由此产生并发冲突，因此需要进行对资源管理的并发控制。项目中采用的方法如下：在移动前首先判断目的地是否可达、是否有别的玩家也准备向该地点移动，若有则停止移动让对方先走，否则申请该地点的锁，再次判断是否可达，然后再移动到该地点，释放相应的锁。

3.2 通信效率

在实现网络通信之前，我本来是想实现一个类似 Minecraft 服务器类似的结构，即服务端放置在云服务器中，客户端在玩家本地，当玩家进行操作时将操作内容发送到服务器，由服务器进行逻辑判断并广播同步。但在实际实现时，发现这样对服务端要求较高，需要实时运算并返回，一旦玩家数量、操作多起来，游戏状态的返回可能会不太及时，显著影响游戏体验。因此最终我决定采用帧同步，即客户端向服务端发送玩家操作、状态改变，服务端只负责转发这些消息，由客户端在本地处理并显示这些消息，这样降低了服务端的负担，并将其分摊到客户端上，既保证了消息转发的及时性，又不影响玩家游戏内容同步的体验。

3.3 输入输出

3.3.1 存档功能

存档功能里的输入输出主要是和文件进行交互：把游戏内容写入 save.data、从 save.data 中读入内容用来初始化 world 对象。项目中采用序列化方法，将控制游戏内容的 world 对象及其他相关对象 implements Serializable，然后在调用 save() 时用 ObjectOutputStream 写入 save.data，调用 load() 时用 ObjectInputStream 读入数据并根据对象的类名对应初始化。

3.3.2 网络功能

网络功能相关的输入输出主要是 client 和 server 之间的通信。一开始我采用 DataOutputStream 来发送信息，但是后来发现这种方法经常会导致数据分成几个部分发送（因为有多少读多少），增加了接收端的处理难度；于是我改用 BufferedWriter 发送消息，并在写完后调用 flush() 来保证数据能一次性全部发出，成功解决了问题。

4 工程问题

4.1 设计模式

采用了命令模式：Client 和 Server 接受和发送信息的行为分离，委托其他线程进行

4.2 工程方法

1. 采用单元测试，确保每个单元功能与预期一致。

2. 划分功能模块，每个模块实现的功能尽量保持独立清晰，便于大量模块集成测试

5 课程感言

上完高级 Java 程序设计后，最大的收获是认识了老曹和很多同学。不同于以往专业课的大多数老师，老曹的风趣幽默、平易近人、低 (xuan) 调 (fu) 给我留下了很深的印象，很少能见到这样愿意与同学们在各方面畅聊的老师，可谓亦师亦友；同学们在群里富有想象力、高技术力的展示也让人叹为观止，确实是印证了那句“不怕做不到，就怕想不到”。

课程内容方面，自然是学到了很多 Java 编程的开发技巧和思路。Java 和 C++ 有很多相似点，但在面向对象这方面 Java 有着独特之处。尽管不是第一次接触面向对象，但是用 Java 这门纯面向对象语言来进行实践，无疑给我带来了很多新的收获，对该思想的理解也更加深入。

最后，祝愿老曹身体健康，课程越开越好！