

Developing a Java Game from Scratch

林芳麒¹

1. 南京大学计算机科学与技术系，学号 191220057

E-mail: 2570178594@qq.com

摘要 本文介绍笔者的高级 Java 程序设计课程作业开发的过程，主要内容包含开发目标、设计理念、技术问题、工程问题及课程感言。

关键词 Java，面向对象，并发编程，网络对战

1 开发目标

1.1 游戏简介

该游戏是以 4399 小游戏泡泡堂为灵感设计的，游戏包含单人和多人模式。玩家通过躲避怪物巡逻或放置炸弹杀死怪物，到达终点取得胜利。

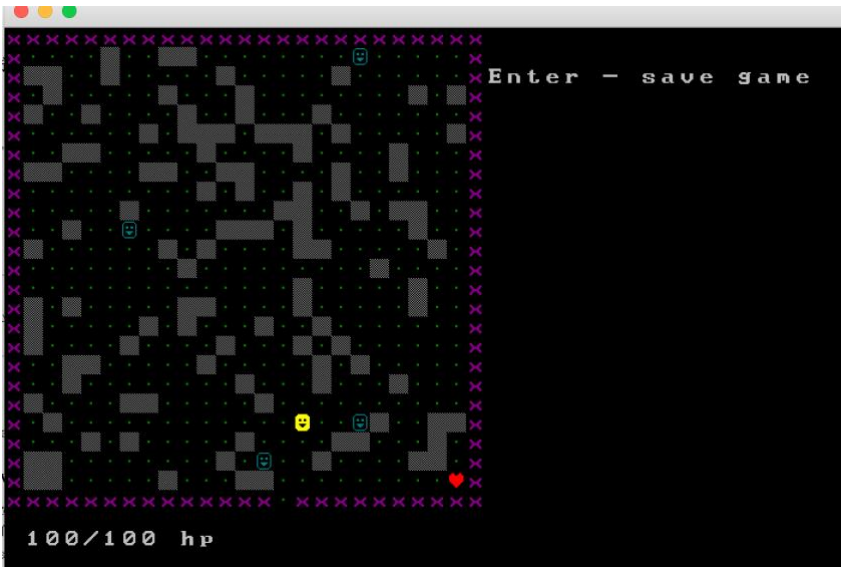


图 1 游戏界面

1.2 游戏操作

1.2.1 游戏主界面

游戏启动后首先进入主界面，玩家根据“”键可选择游戏模式，单人新游戏、继续之前

的游戏存档、多人模式。回车进入游戏。

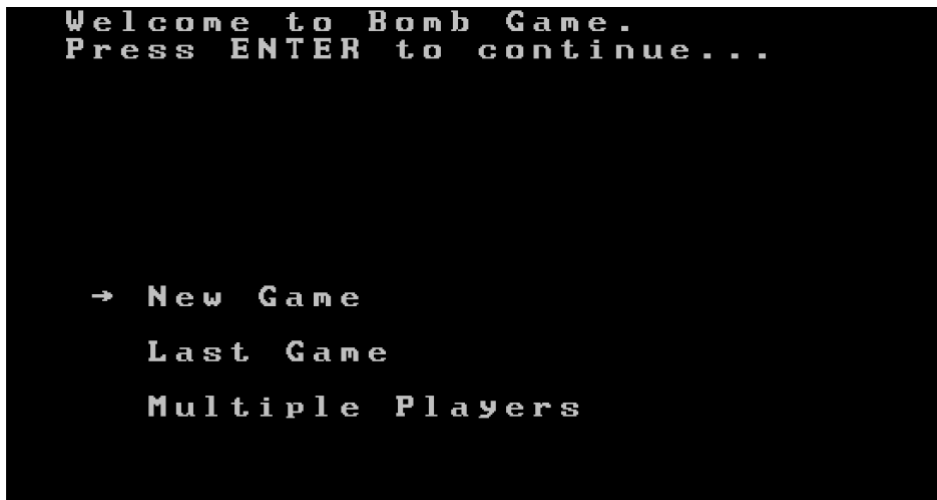


图 2 游戏主界面

1.2.2 玩家操作

移动：↑↓←→

放置炸弹：空格

存档：Enter 回车

注意：炸弹也会对玩家造成伤害，空格会在玩家所在位置放置炸弹，玩家需快速选择下一步操作离开炸弹范围。

1.2.3 怪兽属性

怪兽随机游走，速度为 1.4s 一个格子，一旦被炸弹炸到立即死亡。每攻击一次玩家，玩家 HP 值减 5。

1.2.4 单人模式

玩家 HP 值归零则游戏失败，玩家到达目标取得爱心即成功。

1.2.5 多人模式

需提前开启 Sever，等待 3 名玩家全进入游戏后才能正式开始游戏，一名玩家死亡即游戏失败，只要有一名玩家到达终点即可视为游戏成功。

2 设计理念

2.1 项目结构

项目的大体框架如下，不考虑绘图工具 AsciiPanel。该框架是在 jw04—Rougelike 分支框架上的拓展和修改。删去了 CreatureAI 类，增添一些多线程管理和网络对战相关类。

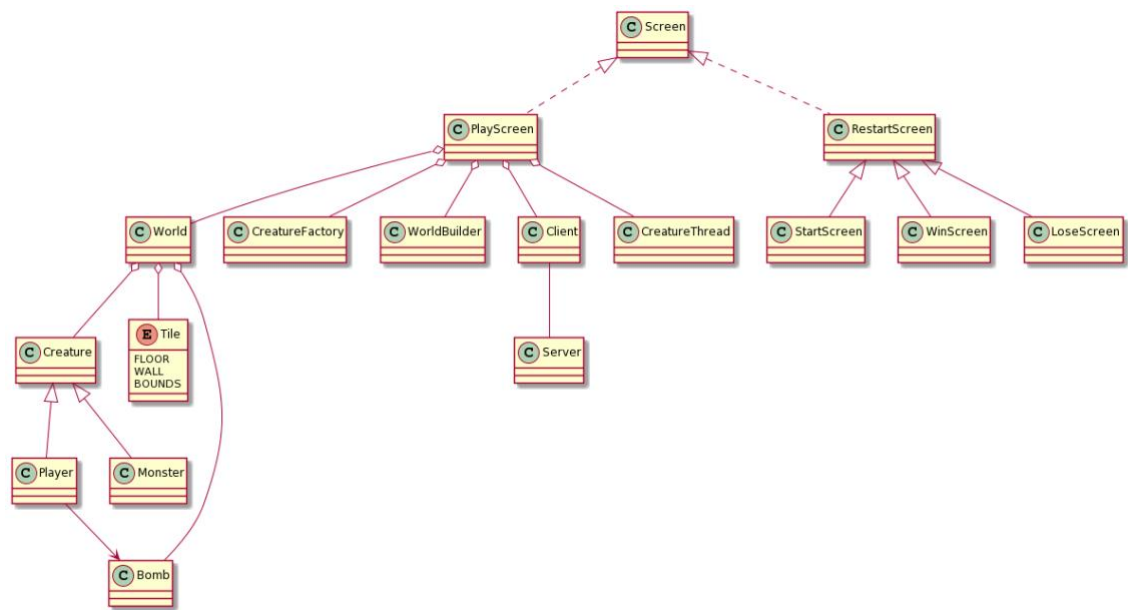


图 3 部分类关系图

2.2 主要类介绍

2.2.1 Creature

所有生物的父亲类，包含生物的基本属性（如 HP 值、移动速度、攻击值等）以及生物的基本行为。Player 和 Monster 都继承该类。

Player: 玩家，在父类基础上增添放置炸弹的操作。

Monster: 怪物，额外实现怪物随机游走和攻击玩家的行为。

2.2.2 Bomb

炸弹，实现 Runnable 接口，每个炸弹爆炸都是一个独立线程，具有 attack 生物的方法。

2.2.3 Tile

地图最小单元，每个 Tile 可能是 Wall、Floor、Bounds，终点 Tile 是 Goal。生物移动会被 Wall 或 Bounds 阻碍。

2.2.4 World

游戏世界主体，由 25 x 25 个 Tile 组成，包含所有怪物和玩家、炸弹。

2.2.5 PlayScreen

游戏主体窗口，负责游戏界面初始化、打印和响应用户操作。CreatureFactory 和 WorldBuild 分别用作由此初始化怪兽、玩家和地图。CreatureThread 用来启动怪兽线程。PlayScreen 也相当于一个客户端 Client 在运行。

3 技术解决

本课设技术问题主要关注在面向对象设计、并发编程、网络通信三方面上。

3.1 面向对象

模拟现实，一个游戏窗口对应一个游戏世界 World，世界地图 Tile 由 WorldBuilder 生成，世界中的每个角色 Creature 由 CreatureFactory 生成。不同角色具备不同属性，行为不同，如 Player 可以在世界里放置 Bomb。

面向对象的设计有助于后续程序的拓展性和复用性。例如多人模式时，游戏中出现三个玩家，玩家的操作相同，只需根据类的属性 ID 即可区分三个对象，三者行为独立于世界 World，不需要因为玩家的增多而大幅度修改游戏运行逻辑；还可以在地图 Tile 上增添多种道具元素，轻松增加游戏趣味性或改变游戏规则；但角色行为发生改变，只需修改角色部分代码，世界 World 不会受到影响等等。

3.2 并发编程

为了用户体验感，游戏逻辑处理和游戏界面打印为两个独立线程。

游戏中每个生物都实现了 Runnable 接口，在游戏资源加载完毕后，会启动 CreatureThread 将生物加入线程池运行。

Monster 线程的 run 方法主体是一个 while 循环，直到怪兽死亡。在 while 中，怪兽每执行一步操作（移动或攻击玩家）后即调用 TimeUnit.MILLISECONDS.sleep(怪兽移动间隔时间)，

线程进入阻塞状态，释放 CPU 资源，将机会让给其他线程执行。

需要考虑的并发问题是一个 Tile 只能同时被一个生物占据。所以，生物访问 Tile 和移动到 Tile 的两个行为应该锁起来，防止当 A 看到空 Tile，线程释放，B 也看到空 Tile，此时又切换到 A 线程，A 踏上了该 Tile，B 之后也踏上了该 Tile（因为它之前看到 Tile 是空的）。

游戏中作为独立线程的还有炸弹，因为炸弹有自己引爆时间和爆炸动画，独立于生物，没有游戏元素可以控制。

3.3 网络对战

采用 NIO Selector 核心组件管理多个网络链接。

多人模式下，要额外启动服务端 Server，作为中介，在 Clients 间广播信息。并且为了同步游戏，要保证每个玩家的开局情况相同，而每个游戏世界的地图和怪物是随机生成的，为了统一，在进入游戏前，Server 会向所有 Clients 发送相同的随机数种子。

当用户选择多人模式后，就会在 Client 于 Server 之间建立连接，并且将 Channel 注册到 selector 上。Client 于 Server 之间通过一种约定俗成信息格式进行沟通，比如玩家移动 Client 发送信息 “Action:1:39”,代表 1 号玩家向右移动一格。

Client 用户每执行一次操作就会向 Server 端发送信息，Server 再将信息广播给其他 Client。Client 读取信息后进行解析后再让自己的 player 执行操作以更新 PlayScreen。

需要注意的是，为了保证游戏的同步，怪兽线程在所有客户端与服务端建立连接后再启动，不是某个客户端一连上服务端就启动怪物，而是三个客户端都连接上服务器后再启动怪物线程。

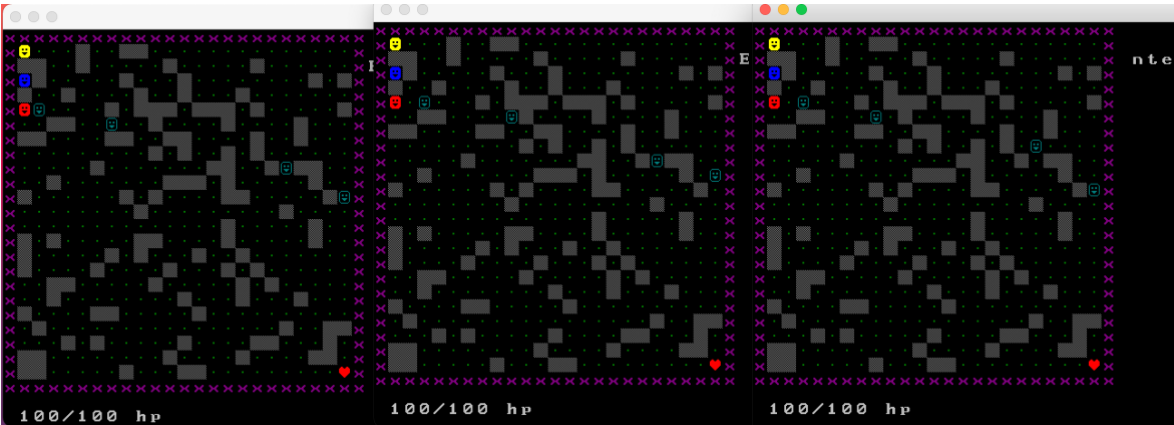


图 4 多人模式

4 工程管理

4.1 Maven

开发过程采用了自动构建工具 Maven，添加了 junit 测试依赖。

5 课程感言

第一次进行多线程编程和网络对战游戏的开发，面临很多挑战，也从这门课体会到真正的网络开发涉及到很多问题，还有很多东西需要学习。在逐步完善游戏的同时也对面向对象的好处有了更深体会，是难得一门作业和学习进度相适配的课程，在写代码过程中加深学习印象。

最想点赞的一点是，老师的鼓励式教学让整门课程的学习氛围很舒适，希望所有大学老师都能向老曹一样多贴近学生（。这样能让学生在课程学习过程中的参与感更强。