

# Developing a Java Game from Scratch

郑义<sup>1</sup>

1. 南京大学软件学院, 南京 210023

E-mail: 201250182@smail.nju.edu.cn

**摘要** 本文主要根据南京大学计算机科学与技术系 2021 秋季课程《Java 高级程序设计》的课程内容和作业来回顾大作业中的一些开发目标, 设计理念, 技术问题, 工程问题和课程感言。本课程的大作业是用 Java 开发出一款能够支持存档、网络对战等功能的 RougeLike 游戏, 通过 Java 的丰富功能和机制来实现一个完整的游戏。本游戏主要采用了 Java swing 作为 gui 的主要开发, 采用了 io 进行游戏进度的保存, 采用了 socket 和 nio 的 selector 机制来进行网络的通信, 采用了面向对象的设计理念来进行游戏的设计, 同时采用了 maven 来进行项目的构建

**关键词** Java, swing, nio, socket, maven, object-oriented

## 1 概述

《Java 高级程序设计》是一门在我看来十分优秀且收获颇多的课, 作为一名在大一已经接触过 Java 的同学, 在这门课上我感受到了 Java 技术的广泛性和高深性, 这门课程让我学会了很多我从来不知道也没有听说过的东西, 也让我再一次回过头去学习了之前学习到的一些知识, 在这门课里我觉得收获颇丰的几个知识点包括以下几点:

1. Java 高并发, 线程池的概念和锁机制
2. Java nio, 网络通信技术和 reactor 设计模式
3. 面向对象的设计方法
4. 等

本篇文章将会从本门课程的大作业中所使用的一些技术来简要介绍一下该项目开发过程中的一些东西。

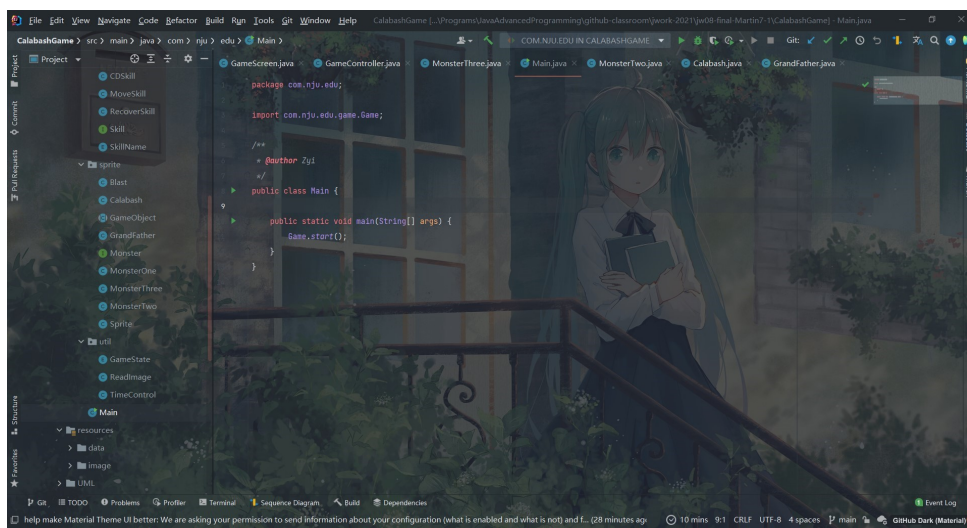


图 1 游戏开发概况

Figure 1 Caption

## 2 开发目标

《葫芦娃大战妖精》是一款采用 maven 进行项目构建, 百分百纯 Java 的一款 2d 桌面端射击游戏, 主要参考的游戏有《崩坏 2》,《飞机大战》等, 在该游戏中玩家将会扮演一名葫芦娃, 守护爷爷, 抵抗进攻敌人的火力, 该游戏中会出现的敌人主要有三种, 一种是蛤蟆精, 移动速度较慢但会定期发射进攻的子弹, 一种是蜈蚣精, 会在整个屏幕范围进行上下和左右的移动, 一种是蛇精, 会随着存活时间的增加而逐渐增加移动速度。同时, 葫芦娃也会定期被爷爷给予技能, 主要的技能有三种, 分别是加快移动速度, 减少射击的间隔和进行一定的回复。本游戏的获胜要求是能够在不死的情况下获得一定的分数, 随着游戏时间的增加, 怪物的出现频率会越来越高, 游戏难度也会逐步增大。

这款游戏的设计中主要是借鉴了之前很火的一些射击游戏, 联机模式下玩家们将齐心协力来击败敌人, 获得胜利。

## 3 设计理念

《葫芦娃大战妖精》中的设计理念主要采用了面向对象的编程范式, 同时将 gui 的设计和底层逻辑的实现进行分离。在设计过程中, 主要是分成了几个大模块来进行设计。

1. 首先是底层逻辑的实现, 包括主要人物葫芦娃, 妖精, 爷爷的移动逻辑、数值设计、移动模式、人物的设计机制等。
2. 然后是关于游戏核心玩法的设计, 由于是一个射击游戏, 核心玩法的设计主要是关于检测子弹和人物碰撞机制的设置, 以及关于子弹的数值设置和一些复杂一点的子弹触发机制的设置
3. 然后是关于 gui 的设置, 我用 swing 自己写了一个简单的游戏 gui。

下图是项目的代码结构:

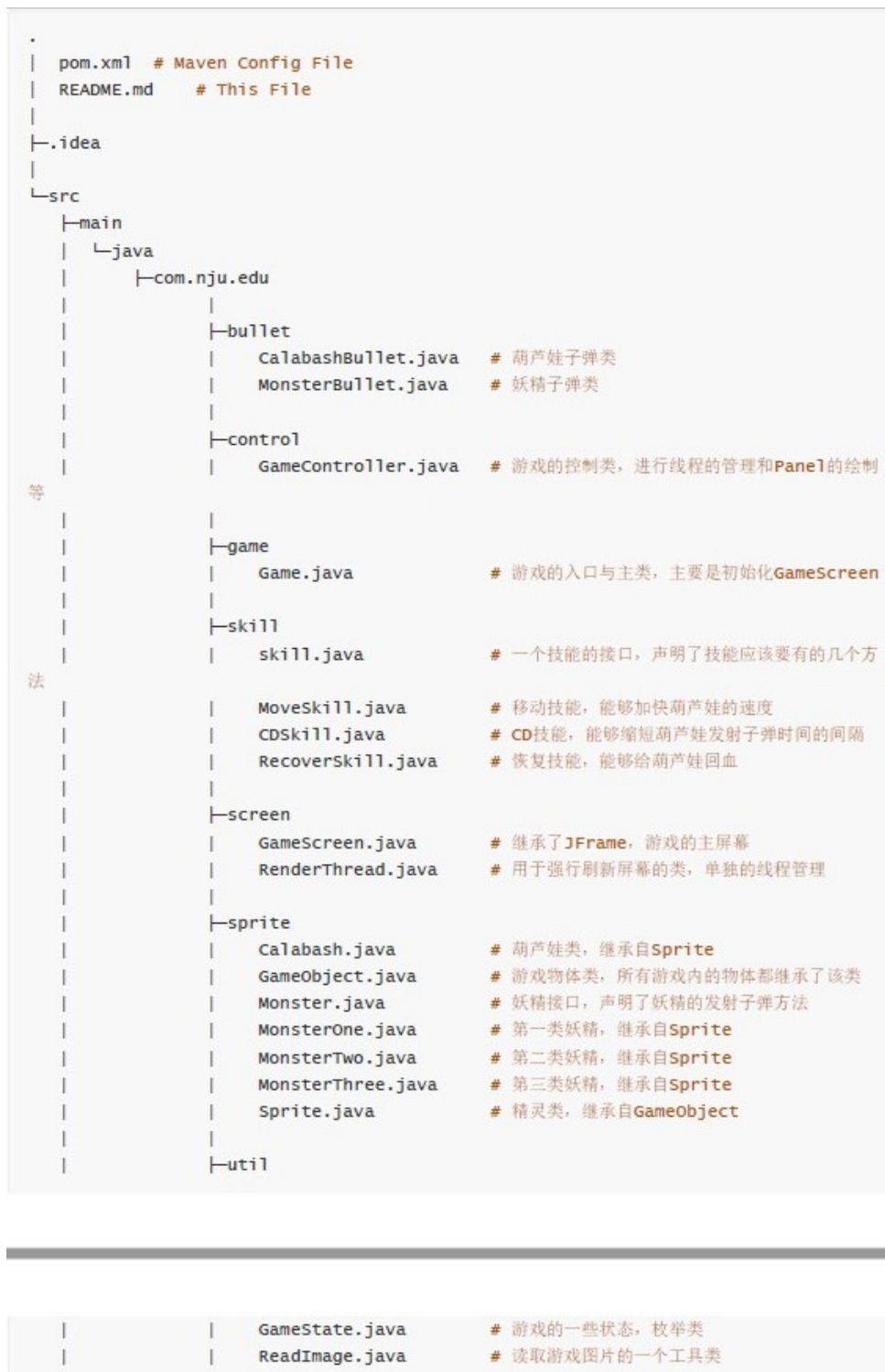


图 2 代码结构  
Figure 2 Caption

## 4 技术问题

在开发过程中,难免会遇到项目问题,尤其是在对各种技术不太熟练的同时,就需要查询文档和十分多的技术博客来查出问题所在,在此次项目的研发过程中,我主要遇到了以下几个问题:

1. 类的初始化顺序问题,类在被加载的时候是有一定的顺序,不熟悉类之间的加载顺序会导致执行过程中某一个类还没有被加载完就被其它类调用,就会出现 `initializationException` 的异常错误。Java 的类加载顺序是在调用某个类的构造函数最开始,就调用该类的父类的构造函数,进入父类之后首先会调用父类的静态字段 or static 语句段,然后是子类的静态字段 or static 语句段,然后是父类的普通变量和语句块被加载为默认值,然后是父类的构造方法被加载,然后是子类的普通变量和语句块被加载为默认值,最后是子类的构造方法被加载。在最初构建代码的时候,我对葫芦娃和爷爷类进行了单例模式的构建,同时对 `GameController` 进行了单例模式的构建,但引发了类初始异常的错误,后来发现是由于 `GameController` 的单例在被构建的时候会涉及到葫芦娃和爷爷与本类之间复杂的初始化关系(因为我在葫芦娃中也调用了 `GameController`),代码的高耦合度导致了类之间没办法很好的做到单一职责的作用,从而导致了异常的发生。
2. 类之间耦合度过高的问题,设计的不够完善导致了多个类之间的通信频繁,互相依赖度过高,且部分类中集成了太多的功能,没有很好的实现单一职责的面向对象设计准则,也导致在后续拓展的过程中需要花大功夫在相关类的一定程度的重构和添加相关功能上。这也是我在游戏设计之初没有对代码架构进行一个很好的设计导致的,在项目开始之初应该对类的功能与职责有一个很好的划分才能够实现很好的可拓展性和可修改性,高耦合度的代码带来的只有后期维护的痛苦。
3. 进度保存的 IO 问题,在进度保存的一开始设计中,由于我并没有游戏中赋予地图的概念(这也是初期设计的问题,后面发现想改已经太晚了),所以打算单纯的保存游戏人物的对象状态即可,于是我用了序列化来将当前在场上的对象都进行序列化保存到一个 `ser` 文件中,然后我发现这样做的一个问题就是每次在进度读取的时候我都需要强制转换来将文件中的对象转换成对应的对象,因为在序列化保存中,写入文件的其实只是一个 `Object`,但读出来的是各种类,这就导致了我必须将不同的类分开存储,然后就导致了我会有多个存取的 `FileWriter` 或者多个读取的 `FileReader`,一个是效率问题,另一个是同时有了这么多文件读写的代码,看起来其实是冗余且有很大一步优化空间的,这也是我在后面想要优化的一个内容。
4. 网络对战的 socket 问题,本次网络对战要求是用 `nio` 的 `selector` 机制来实现服务端和客户端的通信

综上,我发现许多问题的原因其实主要是因为前期的设计不合理而导致的,在一个项目启动之前,对其中逻辑和职责的分配其实是一个十分重要的内容,也是一个不可或缺的内容,好的设计和职责分配会在后面写代码的时候事半功倍,同时也能够增强代码的可读性和可修改性,使项目更容易维护和拓展。这也说明了其实我并没有很好理解面向对象中的一些基本准则。同时,有些库的使用并不是很熟练,对文档的内容还没有很好理解就开始使用,就导致了后续 bug 频繁的原因。

## 5 工程问题

工程问题在我看来和技术问题是同等重要的,好的项目构建和工程的管理能够让我们在制作项目的过程中事半功倍,本项目主要采用了 maven 来进行构建,在构建过程中还是有遇到许多工程问题的,像在上文所说的关于面向对象的设计,其实也算是工程问题,面向对象的编程范式本身就是为了解决工程中的很多问题而应运而生的,这次项目也让我深刻意识到了面向对象的重要性,后期开发过程中对一些细微之处的更改可能都会影响到其他的类,这是没有很好的设计每一个类的接口,导致了一个类中的实现可能会影响到其他的类。

在 maven 项目的构建中,也有遇到一些问题与困难,用 vscode 进行构建的时候,发现 vscode 使用 maven 时支持最高的 jdk 版本是 1.7,然后研究了一下 pom.xml 的写法,最后把版本定格在了 1.8,方便进行 lambda 表达式和函数式编程的 stream 的一些应用,包管理方面本项目主要是使用了 junit 和 jacoco 进行测试的管理,不过在这感叹一句 maven 对包管理提供了极大的便利,能够极大的缩短我们在进行第三方库的拓展上花费的时间。

同时在项目的设计中,我是想要用上一些设计模式的,比如单例模式来管理 player,对象池模式来管理出现的妖精和小怪,但由于上文技术问题中出现的问题,我最终放弃了这种想法,出现了这种问题的原因是因为我对这些设计模式掌握还不够到位,前期设计的时候并没有将这些考虑进去,而是写到一半的时候才发现这里好像可以用上这一种设计模式,这就导致了如果这时候要修改就会导致大范围的代码重构。

与此同时我也想感慨一下自己的代码能力还有很多的提升空间,以及第一次接触一个完成流程开发还不清楚一个项目的完整开发流程应该是什么样的,但是经过了这次项目的洗礼,我深刻明白了在项目启动之际对项目之中的功能和模块进行设计的好处,比如 UML 图的绘制以及不同类的职责分配其实都应该要花费极大心思去仔细设计,好的工程设计能够在开始写代码的时候省下很多的功夫。同时,我也深感自己对 Java 的很多机制其实都还不是很理解,看似会用了但其实没有明白背后的原理究竟是何,这也是让我在项目进展过程中感觉卡住的点,写一点遇到一个陌生的地方就得停下来去查文档和大范围观看一些技术博客才能够理解背后的原理,才能够真正的将它运用在我的项目之中。

## 6 课程感言

曹春老师的这门课是一门十分优秀的课,不仅结合了很多新鲜的技术和知识,也把 Java 的基础和很多机制的原理讲的十分的透彻,我认为在这门课上我收获颇丰,唯一就是自己的基础不够好导致了没办法很好的吸收每一个知识点,所以在作业中也没办法很好的将所有功能都实现的十分完美。

**致谢** 致谢.

感谢本课程的曹春老师,让我学会了十分多的知识,虽然可能我还没有很好的掌握他们,但至少他让我知道了 Java 这门课程的很多高级原理以及基础的使用,也让我明白了在接触到一项新技术的时候,我应该以一种什么样的方法来对待和学习他,同时他也让我在一学期当中开发出了一款

虽然可能不太符合他期望的游戏，但我还是感觉自己能够靠一人之力做出一款游戏是一种极大的自豪，同时也对我在一学期的课程学习中感到收获满满

## 参考文献

---

- 1 Chun Cao. Java Advanced Programming. Slides, 2021: all of pages

# Developing a Java Game from Scratch

Yi Zheng<sup>1</sup>

1. *Nanjing University Software Institute, Nanjing 210023, China*

E-mail: 201250182@smail.nju.edu.cn

**Abstract** This article mainly reviews some of the development goals, design concepts, technical issues, engineering issues and course reflections in the major assignments based on the course content and major assignments of the 2021 Fall Course "Java Advanced Programming" of the Department of Computer Science and Technology of Nanjing University. The major task of this course is to use Java to develop a RougeLike game that can support archives, online battles and other functions, and realize a complete game through the rich functions and mechanisms of Java. This game mainly uses Java swing as the main development of the gui, uses IO to save the game progress, uses the socket and nio selector mechanism for network communication, and uses object-oriented design concepts to design the game. At the same time Maven is used to build the project.

**Keywords** Java, swing, nio, socket, maven, object-oriented