

Developing a Java Game from Scratch

张杨阳¹

1. 南京大学, 南京210023

E-mail: 191220168@smail.nju.edu.cn

摘要 本项目是设计一个葫芦娃为原型的贪吃蛇类型游戏, 在设计过程中主要使用了面向对象, 多线程等技术, 采用文件写入读出来保存游戏进度, 使用NIO Selector来进行多人对战。同时使用maven技术来管理项目。

关键词 开发目标, 设计理念, 技术问题, 工程问题, 课程感言

1 开发目标

1.1 游戏介绍

本游戏是一个贪吃蛇游戏, 以葫芦娃为原型, 每吃一个豆子就成长并变长一格, 也就是多一个葫芦娃参加战斗, 葫芦娃在吃豆子自我强大的过程中逐渐强大, 从大娃到二娃一直到七娃交替出现。可以将吃一个豆子理解为击败一个怪物, 最后成长起来去打败蛇精。但是葫芦娃最后也长成为了蛇精的模样, 终究变成了自己最讨厌的人, 略带一丝讽刺意味。本游戏共有三个版本, 在第一个版本中共有两种游戏模式, 第一种是小图模式, 以通关为目的; 第二种是大图模式, 地图更大, 游戏目标是吃到更多的豆子, 采取积分制。在第二个版本中增加了存档功能, 可以在任意时刻存档保存游戏, 然后再读档从上一次游戏位置继续。第三个版本加入了多人对战系统, 可以多个玩家同时操控不同的贪吃蛇。

1.2 详细玩法讲解

1.2.1 版本1.0

第一个版本分为两个模式, 第一个是小图模式, 即只有一张4*4的地图, 以贪吃蛇填满整个屏幕为目的。如果贪吃蛇碰到四周的墙壁或者碰到自己的身体都会死亡, 因为地图过小并且以通关为目的, 小图模式的贪吃蛇是没有速度的, 方向键移动一下才会在蛇头的带领下移动一格。贪吃蛇吃到豆子会延长一个身体。



图 1 小图模式

大图模式拥有更加广阔的地图，相应的也添加了速度，当使用上下左右方向键时，不再是移动一格，而是贪吃蛇朝着上下左右的某一个方向一直移动，直到下一次键盘按键做出改变方向时才会改变方向。大图模式采用积分制的方式来记分，因为大图模式想要通关是非常非常非常困难的，很难将贪吃蛇布满整个屏幕。所以没吃下一个豆子贪吃蛇就会增长一个同时记一分。

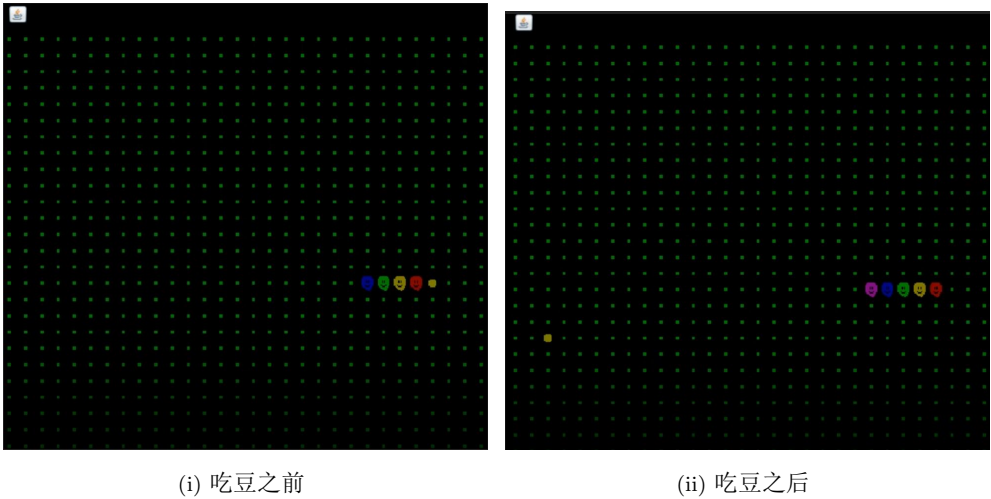


图 2 大图模式

1.2.2 版本2.0

第二个版本增加了回放功能，同时在开始界面增加了按钮设置，选择新的游戏(start new game!), 或者读档(start old game!)。在游戏的进行过程中，可以按下键盘上的s 键来保存游戏进度，放入log.txt中。下一次进行游戏时可以选择读档 (start old game!), 从上一次保存进度的地方进入游戏，恢复之前保存的游戏景象继续进行游戏。这一部分截图较难反应，主要依据视频。

1.2.3 版本3.0

第三个版本增加了多人对战功能，首先需要打开server服务器，启动服务器，然后再打开终端client，

每打开一个终端client就会加入一个新的玩家,最多支持三名玩家同时游戏。每名玩家可以控制自己颜色的贪吃蛇向不同的方向移动,移动情况和之前版本相同,按下对应键盘按键就会向某个方向一直移动。仅简单实现各个人物的移动,为保证在一个机器上的录制体验,没有涉及人物的死亡。但是在前两个版本当中存在win和lose的概念。

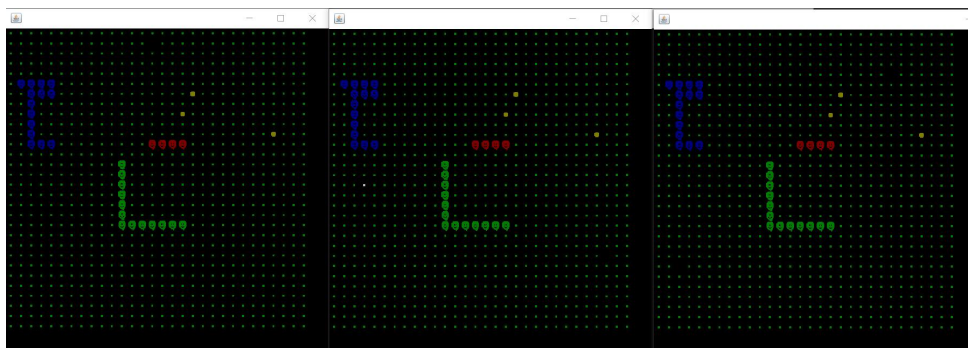


图 3 多人对战

1.2.4 优化方向

整个游戏是一个较为简单基础的部分,在游戏玩法上并没有很大的创新。首先可以增加图形化的葫芦娃,豆子和地面。现在仅使用基础图形,这个方面可以进行优化。其次在玩法上可以模仿市场上的贪吃蛇大作战等多人游戏,死亡之后尸体会形成豆子,这样可以大大加快游戏节奏。此外还可以创新玩法,比如没有豆子的存在,贪吃蛇的变长是依据时间的,在相同时间内每个玩家的贪吃蛇都会增加相同的长度,而且运动时间随着长度增加会不断加快,蛇头碰撞到别人或者自己都会死亡,死亡即淘汰。另外整体面向对象的结构也有一些紊乱,更加条理化,增加测试也是优化的方向。整体来说,还有很大的进步空间,预计在寒假中继续完成。

2 设计理念

2.1 代码整体布局

首先分为两个大的部分,一部分是服务器server,另一部分是客户端client,他们的整体布局是大致相同的,主要是main不同。client和server都采用maven进行管理。client部分下图展示整体布局(尝试使用urm-core自动构造类图,但没有成功,用文件关系来替代),主要是在jw04中Rougelike分支上进行的创作与修改。



图 4 代码整体布局

2.2 主要类介绍

1.ApplicationMain, 是main类, 在client中主要用于设计初始界面, 进行初始化, 并与服务器server建立连接, 同时对键盘操作予以回应; 在server中用于。2.Freshmain, 在client非多人对战时用来定时刷新屏幕, 在server中是向client不断发送信息。3.Freshmainclient, 用于多人对战时在client上从服务器接收信息并定时刷新屏幕将信息反馈到屏幕上。4.world用于承载其他的游戏内容, 是起到一个世界的作用, 里面包含了地图信息和人物信息, 即包含Tile和Creature。5.worldbuilder用于构建世界。6.Tile是一个枚举类型, 枚举了世界地面的类型, 是floor还是wall, 本项目主要使用floor。7.Creature是生物, 包含了贪吃蛇蛇头, 蛇身子以及能使贪吃蛇变大的豆子。8.CreatureFactory是构建creature的工厂, 用于构建不同类型的creature。9.PlayerAI是玩家主要控制的对象, 即player。通过方向键来控制playerai, 就可以让贪吃蛇进行各个方向的移动与生长。10.Screen是屏幕类, 一方面实现了对键盘的操作进行反馈, 另一方面把游戏的画面通过屏幕的形式显现出来。11.PlayerScreen是游戏界面, 包含了构造的世界和

各个人物以及对游戏过程中输入的信号进行传输。

2.3 设计的思路

整个项目采用面向对象的结构来设计,每一个对象实体设立一个类来表示,各个类之间存在继承关系和接口关系。对于一个client进行的单机游戏时,存在一个线程来不断的刷新屏幕,一个线程来保证贪吃蛇在一个方向上保持一定的速度前进。对于联机多人游戏来说,使用NIO Selector来进行管理,server接受到client的请求后,通过selector来对client进行读写操作。同时采用锁的控制,避免多个线程对同一个共享变量进行操作。

3 技术问题

3.1 面向对象

接触到java发现java对面向对象的需求很高,本项目采用面向对象的方法,但是很多时候构建新的对象在交互上存在较为麻烦的现象,所以本项目有的地方并没有严格的面向对象,这也是本项目可以精进的点之一。

3.2 并发编程

本项目采用并发编程,在单机游戏中,为了保证屏幕的实时更新,用一个线程专门来不断的刷新屏幕。为了能让贪吃蛇按一下方向键就往一个方向一直移动,用另一个线程来专门控制贪吃蛇一直移动。但这在jw06中遇到了问题,最后发现是从log中得到world的数据之后并没有启动让蛇移动的线程,导致方向键按下后贪吃蛇并不会移动。

3.3 文件读写

在游戏运行的过程中,按下s可以进行存档,将world转化为流存入log.txt当中。在读档游戏时,将log中存入的world拿出来构建一个新的屏幕进行游戏,此时之前world中的东西就转移到新的屏幕上来。

3.4 网络对战

采用NIO Selector来管理网络。首先要打开server,然后打开client后就可以与server建立连接并且将channel注册到selector上。所有游戏情景与动作的完成都由server来完成,client只需将键盘的方向键操作发送到server,然后server在playscreen上进行游戏模拟,再将屏幕信息发送到每一个client。这里要注意uid的一致性问题,以及类内容的一致性问题,不然在通信时会出现信息不对等的情况。

4 工程问题

采用自动构建工具maven进行工程管理。使用测试覆盖度报告生成插件jacoco进行自动生成单元测试覆盖度报告。(此部分完成较简略)

5 课程感言

java这门课让我学到了很多東西，对面向对象的理解更加深刻，第一次尝试并发编程，尝试网络通信来完成多人对战，重要的是第一次自己做了一个自己想做的游戏项目。这门课更多的是启发式学习，需要更多去看书，去查阅资料，去研究代码才能进步。作业每一步也与学习进度相匹配，不得不说这门课真的称得上是一门好课。因为各种各样的原因，这一次游戏项目的并不够完善，主要还是拖延症作祟，希望寒假里可以根据优化方向继续优化项目。同时，对于老师课上讲的内容，虽然在实验中有了很大的收获与理解，得到了一定的融会贯通，但还有一部分并没有能够理解或者说运用，在java学习的路上还是长路漫漫。曹老师鼓励式的教育也让我觉得课堂氛围以及课后讨论氛围很温暖。也谢谢老曹多次提醒我作业，拖了班级的后腿还是很愧疚的，辛苦曹老师了！