

Developing a Java Game from Scratch

闫佳琪¹

1. 南京大学 南京, 210023

E-mail: 191300069@smail.nju.edu.cn

摘要：本项目的目标是开发一个单人和多人联机的寻路小游戏。在设计游戏的过程中，贯穿游戏主题的是面向对象编程思想，利用了线程池多线程编程，NIO 网络编程等 java 中结构，实现了保存加载地图，maven 工程管理，对游戏进行图形化显示

关键词：java 面向对象 图形化 网络编程

1 开发目标

实现了一个在 50x106 大小的地图上，玩家通过寻找工具，击杀和躲避敌人，最终找到钥匙走入终点大门的可联机游戏。

1.1 故事简述

初始时，主角出生在 3x3 大小的 house 里，生命值为 3，不会受到攻击，主角的任务是走进终结之门。而可以走进门的前提是主角的背包里有钥匙（key）。为了获得钥匙，主角必须越过重重障碍：首先要在初始的家（house）下方的丛林中拿到小船（boat）并躲开小怪（goblin）的击杀，在这个过程中，主角可以反杀小怪也可以吃到小红心（energyFood）获取到额外的生命值，但如果不幸被小怪撞到生命值就会减一；拿到小船后，主角才可以越过丛林右面的小河（water），否则主角无法穿过小河，一直被困在丛林中。

越过小河后再走一段距离，主角就可以进入一个具有四重难关的小楼中，在每一层小楼中都有随机地雷（mine），主角需要越过这些地雷行走，否则生命值就会减一。在第一层楼中，主角必须拿到拐杖（Cane）才能走进第二关，所以主角必须躲过第一层楼中的所有地雷和四个小怪的攻击拿到拐杖。在第二层至第四层楼中，每一层都相似，有一个大怪兽（bigGoblin）和一些随机地雷。所以主角需要巧妙地避开地雷和大怪兽的流弹（marbles），被大怪兽的溜流弹不幸击中生命值就会减一，而避开大怪兽的流弹的最好方法就是击杀它。

当从第四层小楼走出来后，任务就变得轻松愉快了。只需要进入最上方的屋子中取得钥匙，可以顺便吃掉四个小红心增加生命值。最后，走入下方被四个小怪看守的长方形圣殿中进入终结之门就获得了游戏的胜利。

2 设计理念

2.1 程序主题框架

程序主体分为 3 个模块：asciiPanel，calabashbros 和 screen

2.1.1 asciiPanel

在这个模块中，程序进行绘图的规范设计

2.1.2 calabashbros

在这个模块中，可以丰富游戏中显示的对象，在这个游戏中实现的对象可分为两类：生物体和非生物体

2.1.2.1 非生物体

- 1. boat

小船在游戏地图左下角的丛林里随机出现，只有拥有小船的玩家才能渡过河流，可以被玩家拾取，不能被怪兽穿越

- 2. cane

拐杖在游戏地图右下角小怪守卫的第一层阁楼里出现，只有拥有拐杖的玩家才能渡过进入二至四楼，可以被玩家拾取，不能被怪兽穿越

- 3. carpet

地毯，玩家可以穿越，但是大怪和小怪都不能通过

- 4. energyFood

能量食物，只有玩家才可以拾取，拾取到的玩家生命值加一，不能被怪兽穿越

- 5. floor

地板，地图中的每一个 tile 都要有物体，在最开始绘制地图的时候，每一个 tile 都是一个地板，在放置物体时，物体取缔地板；当物体消失时，地板取缔物体，地板可以被生物体穿越

- **6. grass**

青草，属于 flint 子类，所有生物体都不能穿行也不能拾取

- **7. house**

玩家出生所在地，玩家可以穿越，怪兽不能穿越

- **8. key**

钥匙，玩家可以拾取，怪兽不能穿越，只有拥有钥匙的玩家才能走向终结之门

- **9. marbles**

流弹，具有方向，且一直向该方向移动，可以被玩家和大怪兽发射，颜色与发射的生命体颜色相同，遇到除地板外的任何物体都会停止，消失，遇到除发射者外的生物体，该生物体生命值减一

- **10. mine**

地雷，可以被玩家拾取，怪兽不能穿越，拾取的玩家生命值减一

- **11. portal**

终结之门，玩家进入其中，则游戏胜利，可以被玩家穿越，怪兽不能穿越

- **12. tree**

小树，属于 flint 子类，所有生物体都不能穿行也不能拾取

- **13. wall**

城墙，属于 flint 子类，所有生物体都不能穿行也不能拾取

- **14. water**

水流，用水流组成小河，除了拥有小船的玩家可以穿行，其他生物体都不能穿行也不能拾取

- **15. flint**

坚硬的物体，是所有不能被任何生物体穿行的物体的父类

2.1.2.2 生物体

生物体包括 calabash, goblin 和 bigGoblin 三种，其中 calabash 是可以被我们操控的主角，另外两个是主角寻找终结之门上的阻碍者。

• 1. calabash

主角 (calabash) 初始生命值为 3。当遭到怪物攻击时生命值减一，当生命值为 0 时，死亡，并从地图上消失，游戏失败并结束。

• 2. goblin

小怪 (goblin) 生命值仅为 1，不能增加，且不能发射流弹，只有在触碰到主角的时候才会攻击她。当遭到主角攻击时生命值减一，当生命值为 0 时，死亡，并从地图上消失。

• 3. bigGoblin

大怪兽 (bigGoblin) 生命值为 3，不能增加，可以发射流弹，且大怪兽会自动检索主角所处方位，根据主角所处方位来发射流弹。而且大怪兽只会在自己所在楼层移动，不会走出自己看守的地盘。当遭到主角攻击时生命值减一，当生命值为 0 时，死亡，并从地图上消失。

2.1.3 screen

初始时，随机创建丛林，河流；小船，地雷和丛林中的能量食物随意放置，增强游戏的随机性和体验感。

每一个生物体和流弹都是一个线程，玩家和怪兽的线程在游戏开始时就已经建立，流弹的线程在玩家有发射输入或大怪兽检测出玩家在其范围内时才会创建，而流弹的线程直至这个生物体死亡或流弹停止或游戏结束时，才会终止。

2.2 单人和多人游戏模式

2.2.1 单人模式

单人模式下，通过键盘操作唯一的玩家。输入 ↑ 代表希望玩家向上移动一格，输入 ↓ 代表希望玩家向下移动一格，输入 ← 代表希望玩家向左移动一格，输入 → 代表希望玩家向右移动一格；输入 e 代表希望玩家向上发射一枚流弹，输入 d 代表希望玩家向下发射一枚流弹，输入 s 代表希望玩家向左发射一枚流弹，输入 f 代表希望玩家向右发射一枚流弹。

当玩家的希望的移动方向的下一格为 Flint 的子类或者超出地图边界时，玩家的位置不会改变。

玩家的生命值和拥有的物品会在游戏界面的左上方显示：



如图所示，玩家的生命值在最前方显示，拾取的物品按照拾取次序排列其后。

2.2.2 多人模式

多人模式是在单人模式的基础上增加了一些功能：至多三个玩家可以同时上线，每个玩家颜色不同，拾取和穿越物体的规则和地图与单人模式相同。所以玩家要相互竞争只有一件的物品，比如只有拿到小船的玩家能过河，其余玩家要么被困在丛林要么被另外的玩家杀死。当玩家 A 用自己的流弹杀死了玩家 B 时，玩家 B 背包里的所有物品就归玩家 A 所有，玩家 B 死亡并从地图上消失。

2.3 保存和恢复地图

在单人模式下的玩家可以进行此组操作。

当玩家输入 a 时代表玩家进行保存地图操作，当玩家输入 b 时代表玩家进行恢复地图操作，存档位只有一个，所以每次保存地图时新的地图都会覆盖掉旧的地图。

3 技术问题

3.1 并行处理

在 WorldScreen.java 中利用 lock.lock() 和线程池为每一个生物体和流弹都创建了一个线程，当且仅当线程代表的物体死亡或者 all_offline 设置为 true 时，线程终结。为了更好地模拟真实形态，流弹每 50ms 移动一格，生物体最快每 500ms 移动一格。

当玩家按下 b 要求重开时，关闭线程池，将 all_offline 设置为 true，不再接收新的线程，等待现有的进程结束。待进程全部结束后，将 all_offline 设置为 false，创建新的线程池和地图界面，把保存的地图和生物体加载到地图上，然后玩家就可以看到存档时的界面了。

3.2 界面显示

界面显示用到的思想也是线程池，在线程池中加入一个绘制屏幕的线程，每隔 50ms 绘制一次屏幕。

3.3 网络通信

多人模式采取的是 NIO Selector 模式，为了实现通信和程序的简单化，我们设置了服务端主程序 Main6.java 和两个客户端程序 Client0.java 和 Client1.java。

在程序中，Main6.java 的代码是由单人模式的基础上增添了通讯功能的文件。当 Main6.java 接收到客户端程序发来的增加玩家请求时，会在本机的游戏中增加一个玩家线程；若接收到客户端程序发来的其对应玩家的执行操作请求则在服务端代码中执行这一步。

然后在线程中不断将整个 WorldScreen 用 byte 流形式广播出去：每一个 tile 的 glyph, r, g, b 都为 byte 存到输出流中，然后将左上方的玩家信息也读入其中，供客户端接收。

客户端实际上只负责发送信号和接收数据，真正用到的文件只有自己的主程序和用来绘制屏幕的 ClientScreen.java 文件。当客户端接收到服务端发送的瞬时地图时，用 ClientScreen.java 程序解码，识别出每一个 tile 中对应的 color 和 glyph，然后将其绘入自己的地图中显示在屏幕上。客户端无需知道每一个 tile 代表的具体物体和物体的各种属性，只需要知道形状和颜色，这也让我在这个项目中对于 CS 一种模式有了更深的理解。

4 工程问题

在本次作业中使用 maven 项目管理，在 pom.xml 中添加需要的依赖和插件。用 junit 作为单元插件。

由于对 maven 运行的不了解，在进行项目管理时遇到了诸多困难，比如 maven 在构建 jar 包时 java 目录下只会包含 java 文件，而原来的架构中 java 文件和 resource 文件都在同一个 java 目录下，没有单独分出来，应该把 resource 文件夹放在与 java 文件夹所在的同一文件夹下。

利用 mvn package 对项目进行打包

最后，在对应的文件夹下：

执行 `Java -cp .\project6-1.0.0.jar Main6` 可以执行主程序

执行 `Java -cp .\project6-1.0.0.jar Client0` 可以执行客户端 1 程序

执行 `Java -cp .\project6-1.0.0.jar Client1` 可以执行客户端 2 程序

执行 `mvn test` 可以执行测试

5 课程问题

在 Java 这门课学到了很多，以前只是在学习 c++ 时曾在书本上对面向对象编程有一个模糊的概念了解。在这一学期的学习结束中，不仅对 java 这门语言和面向对象编程有了更深的认识，也了解到了 java 中可以用到的各种强大的功能。Java 高级程序设计也进一步教会了我如何自主学习和规划自己的时间，拖延总是不对的，所以没有 ddl 是万万不能的，在 ddl 之前赶出来的作业只能勉强算是功能完整，但是离曾经设想过的完整的程序还是差得很远，在日后的学习过程中会努力地更好地管理自己，不在每一次作业中都做最晚提交的 ddl 人。