

# Developing a Java Game from Scratch

叶起厚<sup>1</sup>

1. 南京大学, 南京 210046

E-mail: 1746844479@qq.com

基金资助

**摘要** 说明自己是如何进行 java 游戏的开发的, 对游戏进行概括, 并对自己的设计理念、遇到的技术与工程问题、课程感言等进行描述

**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

本次作业分为 3 个部分, 分别为图形化移动显示和单元测试 (jw05)、存档读取 (jw06)、联机游戏 (jw07)。希望通过编写一个小游戏, 达到图形化显示和趣味性的目的, 同时也能够通过存档读取的设置和联机游戏的功能来进一步学习掌握有关 java 文件读写以及网络通信的知识, 并在构建项目时使用 maven 来进一步了解 java 自动化构建以及单元测试的使用。

### 1.1 游戏简介

#### 1.1.1 选择界面

当打开程序以后, 会进入一个主界面, 主界面上会显示当前可以运行的游戏模式。其中按下 enter 或数字键 1 可以新开单人游戏; 按下数字键 2 可以读取存储信息继续上次游戏; 按下数字键 3 会将当前程序作为服务器端; 按下数字键 4 可以进入玩家选择界面, 再按下数字键 1 或 2 可以选择控制的玩家加入到服务器端以进行多人游戏。(开启服务器以后会直接开始多人游戏的进行, 所以如果不想直接输掉的话, 应尽快选择控制的玩家进入多人游戏中)

#### 1.1.2 游戏画面

开始游戏以后会在窗口左侧显示世界画面, 其中也包括了各类创造物。金币会随机在地图上进行生成, 怪物会向玩家方向移动, 玩家的操作也会显示在地图上。其中画面的刷新频率为 60fps。

窗口右侧会显示当前控制玩家的血量和金钱数量, 同时也会显示操作玩法介绍, 以及剩余敌人的数量。

当游戏胜利或者结束以后都会进入到相应的结算界面中, 再按照之前主界面的操作进行操作即可重开游戏。

### 1.1.3 游戏基本操作

游戏是通过捡取地上随机生成的货币（每拾取一个增加 10 金币），并通过消耗货币进行射击（10 金币）、放炸弹（50 金币）和回血（20 金币）的操作。

每名玩家共有 10 滴血，敌人有 5 滴血，子弹造成伤害时会扣除 1 滴血，炸弹会扣除 5 滴血，回血一次回 1 滴血，玩家之间也可以进行互相攻击（队内伤害）。其中子弹可以击穿冰块，炸弹可以炸掉所有障碍（冰块和石块）。

当敌人被全部消灭时获胜，己方血量为 0 时输掉。

### 1.1.4 存档读取功能

其中存档时会当前地图的大小以及地图上的创造物（包括金币、玩家、敌人和方块）按行存储到一个文本文件中，该文本文件位于 saves 文件夹下，名字为 save.txt。

存储操作通过在游戏时按下 F5 键来进行存储，存储完以后会关闭当前运行界面，可以通过读取游戏继续之前的游戏。这样设置的目的是之前为了进行测试，便于检验是否能够正常存储和读取。存储功能只能在单人模式下使用。

读取操作通过在主界面输入 2 以后，按行读取分析之前存储的文本文件的数据来新建一个世界（只不过这个世界的显示与之前最后保存的世界一致）。之后的操作与新建游戏后的操作一致。

### 1.1.5 联机合作功能

联机功能固定为 2 个玩家，服务端会生成地图和创造物，玩家端可以连入服务端来对对应玩家进行操作。当控制玩家死亡后对应游戏端玩家游戏失败，当敌人全部被玩家消灭以后游戏胜利。

联网的玩家动作和敌人随机动作等都是在服务器端口进行，客户端主要是通过读取服务端传出的地图数据进行刷新（和读取存档类似）以及发送对应玩家的动作给服务器来让服务器进行判断并对对应玩家进行操作。地图刷新同步功能可以当作是文件读写操作的类似操作。

当结束游戏以后玩家端会出现与单人游戏时一样的结束画面，可以继续为玩家端进行单人游戏的重开，若想重新开始多人游戏，则需要关闭之前的服务端并重开一个服务端，之后就可以按照之前一样的操作开始多人游戏。

## 1.2 灵感来源

灵感主要来自多个游戏，分别为吃豆人、坦克大战、炸弹人等。通过将这几个游戏进行整合，实现了一个通过拾取金币（吃豆子获取积分），然后射击（坦克大战）或放置炸弹（炸弹人）的方式来击败敌人获取胜利的游戏。

## 2 设计理念

### 2.1 总体设计方式

代码的总体设计方式采用了面向对象的设计。

以之前 jw04 时的 AsciiPanel 为基础进行界面的开发，并以迷宫的代码为基础进行地图方块的生成，具体生成时先生成迷宫的数组，之后再通过随机计算对该迷宫进行改造，即原本迷宫的墙壁有可能会变成空地、冰块或石块，之后就可以把这些冰块和石块作为创造物添加到地图上（地图本身是一块空地）。

其中玩家、敌人都是固定数目随机生成到地图上（多人游戏玩家为固定位置），金币生成通过设置一个新的线程在地图上每隔一段时间随机生成，子弹、炸弹都是通过玩家生成。

### 2.1.1 画面刷新

通过设置一个线程每隔一段时间调用 `repaint` 函数对界面进行刷新。

### 2.1.2 图片设置

图片调用是通过 `asciiPanel` 包中的 `AsciiFont` 和 `AsciiPanel` 类来对对应的图片进行像素的读取和存储，然后在屏幕上显示时通过输入对应的图片的下标来打印出对应的图片。所以我使用了图片编辑软件（我使用的是 windows 系统自带的画图 3D 软件）来对给的图片中自己需要的位置的像素块进行修改绘制，以在界面上显示自己绘制的图片。

### 2.1.3 创造物运动方式

其中敌人、子弹、炸弹都是通过线程来进行对应的操作的。玩家的移动和相应的操作是通过键盘输入来进行操作。

### 2.1.4 界面的跳转

打开运行程序后会出现一个主界面 `StartScreen`，进行对应的键盘输入后会进入 `PlayScreen`、`LoadScreen`、`ServerScreen` 和 `SelectScreen`（`SelectScreen` 再进行键盘输入后会进入 `ClientScreen`），当游戏胜利或失败后会进入 `WinScreen` 或 `LoseScreen`（`ServerScreen` 无法进入，因为该界面实际上只是一个服务端，并没有窗口显示功能等），当单人模式时进行存档时会进入主界面 `StartScreen` 中。

其中 `StartScreen`、`SelectScreen`、`WinScreen`、`LoseScreen` 都是 `RestartScreen` 的子类，可以进行相应的选择来实现界面的返回跳转；而 `LoadScreen`、`ServerScreen`、`ClientScreen` 是 `PlayScreen` 的子类，可以进行世界界面的显示和对玩家的操作（只不过 `ServerScreen` 因为在自己的线程上启动了服务器，所以无法进一步进行画面的显示）。这所有的屏幕类都是 `Screen` 的子类，`Screen` 是一个接口，用以说明屏幕都可以对键盘输入进行反应，并且也可以显示相应的画面。

### 2.1.5 单人模式

通过对键盘输入的响应来对世界中的玩家进行操作，按下 `F5` 后可以进行保存并返回主界面显示。

其中保存时是先对地图大小进行存储，然后对创造物种类及其相应的位置和数值进行存储，并按行写入文件。读取文件时则是按行读取，并根据数据来创造地图并设置创造物，创造物设置好以后与通常游戏一致，即只是在开始时的世界创造有所差异。

### 2.1.6 多人模式

多人模式固定为 2 个玩家，其中服务端设置端口和创建世界来供客户端连接。

客户端的界面实现与读取操作大体一致，通过服务端每隔一段时间发送给各个客户端世界数据，客户端可以读取发送过来的世界数据来不断创建更新世界情况。客户端对各自的玩家进行操作（键盘输入时），会往 `channel` 里写入对应玩家的 ID 和相应的操作，服务端接收到数据以后会判断玩家并对对应玩家实行相应的操作。

## 2.2 设计好处

### 2.2.1 派生子类

可以沿用之前的界面或者是创造物，一是可以方便管理和操作（例如读取时创造物都可以看做是 Creature 这一父类，屏幕返回时都可以返回给 Screen 类）；二是可以继续使用父类的部分代码或对其进行相应的修改，便于方法的理解和更新，提升了开发效率。

### 2.2.2 图形化

使用之前任务所使用的 AsciiPanel 框架，只需要修改提供的图片即可完成自己的图形化设计。

### 2.2.3 多线程并发

多线程并发的设计可以让不同的工作互相之间没有干扰，而不会存在先后关系从而导致等待等情况。

### 2.2.4 网络设计

通过每次都由服务端来处理操作并将世界数据发送给客户端来显现界面，使客户端之间能够保持数据的一致性，且所有操作都是在同一个服务端上进行的，也不会产生相互之间的矛盾。

## 3 技术问题

### 3.1 通信数据传输问题

因为采用 selector 时，如果 channel 设置了 OP\_READ 模式的话则不会接收到 key 的 isWritable 的信号，所以写操作也放到了 isReadable 的操作中。而 isReadable 只有接收到对应客户端的信息后才会操作，所以需要客户端不断发送信息，但又不能是具体的操作信息（否则会产生混淆），所以我让客户端每隔一段时间发送一个字符串“update”，使服务端能够接收到信号并传送回来。

但这样子如果需要发送特定的操作时，可能会和之前定时发送的信息产生干扰导致读取到的信息错误。所以最后我让一个玩家能够拥有 2 个客户端，其中一个定时发送信息读取界面的客户端，另一个是发送操作信息的客户端，即一个专门用来读，另一个用来写。

### 3.2 通信效率

目前实现的通信是服务端每次都将全部的世界信息传入到客户端中，每次更新时都需要对数据进行全部写入和读取，而每次更新时只有部分数据会发生变化，所以应该只需要针对那几个创造物进行操作即可。因为如果要对特定的创造物进行操作的话，则需要对每个创造物都在本地（客户端）进行存储（如数组或列表等），或者是读取对应位置的创造物的下标和种类等并对其进行特定的操作。这样就需要对生成创造物、销毁创造物以及对对应创造物的操作进行判断。需要对原有的全部写入进行筛选和细分。（该部分未实现）

### 3.3 并发控制

一开始将各个敌人都各自设置了一个线程进行操作，但没过多长时间就产生了报错，上网搜索后说是列表的数据错误。对代码进行查看以后，发现代码中获取对应位置的创造物的操作是通过读取世界所含创造物列表来遍历判断，而每个敌人和玩家的移动可能会导致对应的创造物位置发生改变或

者消失（如捡钱操作），这样子就会对列表进行了修改，而此时有些线程可能正在对列表进行访问（敌人获取玩家位置等），这样就会导致临界区产生错误。

所以要消除这个问题就需要对临界区进行操作，一般的操作应该是通过设置加锁来对临界区进行保护。但我上网对产生的报错进行搜索后，发现可以通过将原本的数组列表 `ArrayList` 修改为 `Copy-On-WriteArrayList`，该列表不需要进行加锁操作并且写入也不会阻塞读取操作，只有写入和写入之间会需要同步等待，其实现方法是通过先进行一个拷贝操作，写入是对该副本进行操作，当写完之后才会对原列表进行替换，可以说是专门为并发临界区设计的数组列表。

### 3.4 输入输出

在文件写入读取时并未发生错误，但在读取存档生成世界以后，我发现剩余怪物的数量一直在当前数量和最大数量之间进行闪烁。后来发现是因为我的读取屏幕是新创游戏屏幕的子类，所以每次在构造函数进行初始化时都会在开头隐式调用父类的默认构造函数。而导致创造出了 2 个世界，使得怪物数量在 2 个世界中不断变更）

所以如果要修改这个问题，就需要强制取消对父类默认构造函数的调用。所以我就在父类中又写了一个带参的构造函数，该构造函数没有任何操作，然后在子类中显式调用该带参构造函数以消除对默认构造函数的调用，并在其后实现其自身的操作。

### 3.5 面向对象带来的好处

面向对象可以让我的屏幕、世界和每一个创造物等都视为一个个独立的个体，并通过对这些个体的具体操作来对不同个体实行不同操作。将各个元件布置在特定的位置，并通过操作来演绎特定的故事。面向对象的编程就像是在一个舞台上进行着个体之间的表演和交互，既使得编写时容易掌握，也可以让编写更富有乐趣性以及容易理解。

## 4 工程问题

### 4.1 设计方法

采用面向对象的设计方法，使得代码编写起来更加易于理解，并且可以使得代码的耦合性降低，使得修改时产生连带错误的可能性降低。

采用了继承的操作，子类对父类的继承和重载使生成的类更加丰富，且开发时有时只需要对特定的方法进行重载或新增方法，并可以沿用部分方法，提升了开发效率。

### 4.2 工程方法

使用了 `maven` 进行代码的自动编译、链接等操作。一是可以对代码进行归纳整理，查找起来更加方便；二是可以进行相应的测试，在运行前就发现对应的问题并加以修改；三是可以自动编译生成对应的项目，减少了人工操作；四是可以加载网络上的库，使不需要手动进行下载到对应目录中。使用 `maven` 对项目进行管理生成，提高了项目的管理能力，提升了开发的效率。

## 5 课程感言

一学期上完以后，在这门课中学到的不仅是 `java` 程序的编写，还有对于面向对象编程的理解。同

时有时候在写代码项目遇到困难的时候也可以通过在群里同学们的聊天获得相关的知识，有时也可以上网查找相关的问题来学得相关的知识和使用方法。

当然，在这门课当中（其实其他课也存在这种情况）学到的另一个重要的知识，就是不要犯拖延症。犯拖延症一是会在最后匆忙之中完成任务，质量不见得会有多好，同时也无法在刚学习完相关知识后即使进行巩固。每次课程的作业都是在学习完相应的知识后进行布置的，如果能够及时进行作业的话，可以对学到的知识进行加深巩固，有利于知识的掌握。

在课程上老师在讲课的过程中不时地穿插着相应代码的执行展示，也进一步加深了我对于知识的掌握，更加形象地理解了这样操作以后会产生什么样的结果。课程后使用编写游戏的作业形式也让作业的实现完成更加具有趣味性和成就感。

**致谢** 感谢曹老师一学期的辛勤教学，以及群内大家一起进行的讨论。

## Developing a Java Game from Scratch

Ye Qihou<sup>1</sup>

1. *Nanjing University, Nanjing 210046, China*

E-mail: 1746844479@qq.com

**Abstract** Tell others about how to develop a java game and summarize the game. Describe about the author's thoughts when developing and problems about technic and engineering, and then feelings about the course.

**Keywords** Target, Thoughts, Technical problem, Engineering problem, Feelings about the course