

# Developing a Java Game from Scratch

李岳<sup>1</sup>

E-mail: 1684555698@qq.com

收稿日期: 2021-12-31; 接受日期: 2021-12-31

**摘要** 关于用 Java 完成游戏的制作的感想以及课程建议

**关键词** Java, Game, Assignment

## 1 开发目标

我写的游戏主要用了 Libgdx 这个引擎，内容就是简单的操纵人物瞄准打怪。至于灵感来源，好像没有一个特定的来源，因为目前的游戏结构真的很简单。包括防御力攻击力等数值都是我瞎填的，平衡性还有待调整（关于为什么 Player 射速这么快，因为这样看起来很炫酷）。游戏只有 Slime 和 Player 两种生物，曾经设置的掉落物也写了一点后发现没时间了就删了。

寒假的时候我还想继续完善一下游戏，加入多种怪物，以及不同的武器，道具，职业。一开始写这个游戏的时候我是有这方面的想法的，所以游戏结构上似可能也很容易调整。

---

## 2 设计理念

### 2.1 jw05-游戏本体

我的游戏主要由 Entity, Screen, Worlds, Assets, MainGame 这五个类组成。

#### 2.1.1 Entity

Entity 包含游戏中所有实体，每一个 Entity 都要有速度，位置，状态和 id。此外游戏每一次 update 会根据 Entity 当前的速度来计算下一刻的位置。Entity 的子类有 Player, Slime, Bullet(这里据我了解 Minecraft 中所有非生物的物品包括射出的箭矢，丢出去的雪球之类的都被视为 Entity，于是借鉴了这种方式，把 Bullet 也看做 Entity，这样就能直接继承 Entity 的速度，位置等属性)。

#### 2.1.2 Screen

Screen 包含游戏的所有可能产生的窗口，具体来说有 MainMenuScreen(开始游戏时)，GameScreen(正在游戏时)，EscScreen(Esc 暂停游戏时)，WinningScreen(游戏胜利时)。每一个 Screen 会在达到特定要求后切换到对应的 Screen。最重要的是 GameScreen，在 GameScreen 里需要载入许多

#### 2.1.3 World

World 类用 ArrayList 存储了所有 Entity，每一次 GameScreen update 时，会 update 当前的 world，由 world 循环分布 update 存储的每一个 Entity。

#### 2.1.4 Assets

Assets 类用于加载所有的材质，这里用的 Libgdx 引擎的材质加载，包括动画播放（动画播放的实现就是在每个实体加一个 stateTime 的属性，每一个 Entity 更新时 stateTime 会加上这一帧所用的时间，这样根据 stateTime 来计算目前的动画应该播到哪一帧）Assets 类在游戏创建时就将所有需要的材质加载好，并存储为 static 成员变量，当每个 Entity 需要使用时只需要从 Assets 里调用即可。Assets 类被回收时会卸载掉所有载入的材质。

#### 2.1.5 MainGame

MainGame 类主要用于创建游戏以及实现网络通信部分。

## 2.2 jw06-地图保存

地图保存上我采用的方法是将 World 的状态 (包括 EntityList 的所有 Entity 的状态) 编码存储起来, 载入就是读取之前保存的存档然后把 World 恢复出来。

## 2.3 jw07-网络对战

网络通信部份我的实现方法是每个 Client 向 Server 发送自己的 Input 和状态, Server 收到 Client 的信息后在 Server 端计算下一帧 World 的状态, 然后广播给所有 Client, Client 收到 Server 发来的信息就解析并更新自己的 World。

---

## 3 技术问题

### 3.1 jw05-游戏本体

游戏本体上的问题。首先是地图怎么制作，这里我用的 Tiled 这个软件可以载入材质，用笔刷画地图，而且是以 tile 的形式保存的。每一个 tile 可以由其特定的标签，这些标签有不同的用处。在检测碰撞上，需要为每一个非 ground 的 tile 设置为 blocked 标签，当 Entity 移动的下一个位置和标签为 blocked 的 tile 重合时，会设置 entity 这个方向的速度为 0，并将其位置设为移动前的位置。

然后是 slime 的 ai，这个只是简单的写了如果 Player 在 Slime 视野范围内，Slime 就向着 Player 的方向移动，不在的话就随机游走，当 Slime 接触 Player 时会产生一个逆向的速度。

下一部分是 Player 的攻击，根据鼠标在屏幕点击的位置生成 Bullet，并把 owner 设置为这个 Player (Owner 不会被 bullet 打伤)。由于看起来很炫酷，我没有设置子弹的射出时间，但这样有个问题是 Slime 收到子弹伤害时会持续受到伤害很容易就死亡。于是我为每个 Entity 设置了无敌帧，当受到伤害时 Entity 的 State 设为 Invincible，经过 3 个 Update 后设置为 Weak，处于 Invincible 的状态是不会受到伤害的。

关于并发控制上，我没有把每个 Entity 都设置为一个线程 (jw05 早期我是这么做的)，但是每个生物一个线程明显不符合正常的游戏逻辑，一般的游戏是一个 update 和一个 render，一个主线程循环进行 update，每次 update 所有 Entity，随即进行 render，绘制出这一帧。由于 update 是顺序 update 每个 entity，并且所使用的容器也是 libgdx 带的支持并发的容器，所以不会产生“撞上”的问题。

### 3.2 jw06-存档

存档上的问题不多，只需要简单的保存当前状态然后载入即可。

### 3.3 jw07-网络

网络上我用的 NIO selector。Client 发自己的状态和输入，Server 广播 World 的状态。关于怎么提高通信的效率上，我的方法是尽量让 Client 和 Server 发送的数据少，Client 不需要自己的 World，只需要发送自己的输入就可以了，有了输入后本应该在本地计算的下一帧位置转到 Server 一同处理。

## 4 工程问题

### 4.1 jw05-游戏本体

设计方法上采用了工厂模式, 创建 Entity 时可以直接调用工厂的静态方法得到一个对象。

### 4.2 jw06-存档

存档上没有什么工程问题。(maybe)

### 4.3 jw07-网络

用 NIO selector, Client 端和 Server 端的接受都是用的 NIO selector (Client 接受 Server 的广播, Server 接受 Client 的信息)

---

## 5 课程感言

首先说一下我上这门课的感想吧。

在选这门课之前，我对 Java 的了解只停留在 Minecraft 的环境依赖上。抱着对 Minecraft 的兴趣（确信选了这门课，最后却意想不到的收获了很多。

这是上大学来我第一次接触的面向对象的课程，自己之前也只是用过 C with stl，也从来没有自己制作过一个有图形化的界面，可以跑起来的程序，感谢 Java！感谢老曹！

作为 OOP 的启蒙（甚至是 Network, Thread, File 等应用的启蒙），即使之后我不再深入学 Java 了，它的影响也算是颇深了。

然后是对课程的建议。

这种每个作业都又前后关联的作业形式真的很不错，希望老曹能保持下去。另外小作业有各种图形化的展示，大作业以开发一个游戏为目标，做起来也是很有意思，我曾多次给舍友看 Java 的作业并怂恿他们选这门课（

如果必须要提出一个需要改进的地方，就是自己写出来的代码明知道是屎山但不知道怎么改，也没有人帮忙看代码，如果能加同学之间互评互改环节会不会好一些？

希望这门课越办越好！