

## 第40课

# 发布方块传奇

把能玩的作品交给第一次使用它的人



发布的起点是另一个人能照着说明安装。整理版本、JAR、源码、配置与测试记录，演练更新和回退，再用五分钟说明作品的规则与已知限制。

默认继续第38、39课的OreRush。也可选已学过的任务、宠物或工具项目。本课的发布指交给测试伙伴的本地作品包，不要求公开上传。

**阅读方式：先猜结果，再运行核对；每次只改一个条件。**

环境：JDK 21、普通 Paper 1.21.11、同版本客户端。服务器插件由 Paper 加载；只有明确标注的规则测试使用 main。

## 开始前：先认清运行位置

前30课打好了普通 Java 程序的基础。这一阶段把代码接到 Paper 服务器：Minecraft 客户端负责进入世界，服务器加载插件，事件和命令把玩家动作交给你的方法。

本书固定使用 JDK 21 与 Paper 1.21.11，客户端版本也保持一致。下载页面上的最新版本可能已经变了，按版本号选择。本书采用普通 plugin.yml 与 Bukkit API；本阶段没有声称支持 Folia。

图框中的短例是局部片段，先读本节的位置说明，不要各自保存成插件。完整文件会明确给出文件名；同一文件分成多段时，按编号连接，包名、import 与外层类只出现一次。

- 预测：先写成功条件和预期消息。
- 运行：构建成功后，在专用本机测试服安装并重启。
- 核对：普通玩家、管理身份与控制台分别走对应路径。

场景插画帮助理解故事；图解解释真实规则，不作为 IDE 或游戏实测截图。做 Java 修改后重新构建、停服替换新 JAR，再启动；不要把旧 JAR 的表现当作新代码结果。

# 01 选一个能讲清楚的作品

阿方把一个JAR扔给同桌：“毕业啦！” 同桌连问版本、命令、配置和安装位置，阿方只好把庆祝烟花先收起来。发布是一份别人能照着使用的完整交付。

发布包把程序与使用说明放在一起：支持版本、安装方法、命令权限、配置、测试和已知限制。用干净测试服照着自己的说明安装一次，能发现熟悉项目的人最容易漏写的步骤。

## 选一个能讲清楚的作品

### 选择一个方向

小而完整的作品

### 写清边界

必须项与暂不实现项

### 解释已完成

展示真实测试证据

不要把选项当成成品：后面以已经写过的OreRush举例。任务、宠物和工具只是可选方向，必须自己实现并测试；选择更多方向不会自动加分。

## 核对表

| 方向   | 最小可交付玩法     | 暂缓的大工程  |
|------|-------------|---------|
| 小游戏  | 报名、计时、计分、结算 | 跨服联赛    |
| 任务系统 | 接任务、进度、一次领奖 | 复杂剧情编辑器 |
| 宠物伙伴 | 召唤、跟随、收回    | 完整战斗AI  |
| 实用工具 | 一个命令解决一个问题  | 万能管理面板  |

## 02 把必须完成和以后再做分开

给1.0.0写一张范围卡。卡上既写“会做什么”，也写“不会做什么”。这样同学问“重启后我的冠军呢”，你可以指出当前只保留本局内存数据，而不是临时假装冠军被苦力怕偷走。

### 把必须完成和以后再做分开

#### 玩法完成

报名、计时、计分、结算

#### 限制明确

不含历史榜与完整防作弊

#### 重启合同

本局清空，配置保留

完成定义：至少一次双人完整对局、十个游戏场景记录，以及24项规则检查通过。没有运行过的项目标“待测”，不能打勾装通关。

#### OreRush 1.0.0范围卡 • 位置见本节说明

必须：join、score、start、stop四个子命令  
 必须：铁矿3分、金矿5分、钻石10分，含深板岩版本  
 必须：世界与1到3600秒配置，管理员开始与停止  
 必须：自然结束显示成绩，取消或停用清理任务  
 不含：历史榜、实物奖励、自动矿场、完整反作弊  
 重启：本局成绩和报名清空，磁盘config.yml保留

- 普通玩家只能报名和查询；管理员才能开始或取消。
- 在指定世界的生存模式下，已报名玩家获准破坏目标矿石才计分。
- 在准备好的专用测试场中验证，不能把练习版本宣称为公开竞技服的防作弊系统。

## 03 把代码放到找得到的地方

第38课只有两个类，不必为了显得高级拆成二十个空盒子。GameRound管理数字与状态；OreRushPlugin连接Paper。先把职责讲清楚，等真实复杂度出现，再考虑command、listener和data等包。

### 把代码放到找得到的地方

#### 数字规则

GameRound独立测试

#### 服务器接线

OreRushPlugin连接Paper

#### 交付说明

源码、资源、Wrapper和记录

边界像围墙一样有用：GameRound不读取Player、不写文件、不操作方块。这样第39课的自动检查才能独立运行。别为了测试一个加分方法，先启动一整座服务器。

### 本项目的实际交付结构 · 位置见本节说明

```
OreRush/  
  build.gradle.kts  
  settings.gradle.kts  
  gradlew gradlew.bat gradle/wrapper/  
  src/main/java/de/teacherma/orerush/  
    GameRound.java  
    OreRushPlugin.java  
  src/main/resources/  
    plugin.yml  
    config.yml  
  checks/RoundChecks.java  
  README.md  
  CHANGELOG.md  
  TEST-RESULTS.md
```

## 04 给发布包一个准确版本

文件名、plugin.yml和说明书应说同一个版本。沿用第31课构建脚本，只核对下面字段。我们约定1.0.1表示小修复、1.1.0表示兼容的新功能；版本数字是约定，不是兼容性的魔法保证。

停下来想一想：更新后游戏表现仍像旧版，先改代码还是先检查安装包？怎样证明测试的是新构建？

### 给发布包一个准确版本

#### 构建版本

build.gradle.kts

#### 插件版本

plugin.yml

#### 交付版本

JAR文件名与说明一致

想通这一步：先核对插件版本、构建产物文件名和安装位置，再看启动日志。必要时比较文件哈希，避免在旧JAR上反复测试新源码。

#### 三个位置分别修改 · 位置见本节说明

```
// settings.gradle.kts
rootProject.name = "OreRush"
// build.gradle.kts
version = "1.0.0"
```

#### plugin.yml中的版本字段 · 位置见本节说明

```
version: 1.0.0
```

#### 项目根目录构建并查看内容 · 位置见本节说明

```
.\gradlew.bat clean build
jar tf build/libs/OreRush-1.0.0.jar
```

- JAR里应有plugin.yml、config.yml和两个主程序类的.class文件。
- 使用项目的Gradle Wrapper，保留wrapper目录和启动脚本。
- build成功只代表构建成功，不代表插件权限、世界条件或双人玩法已经测完。

# 05 配置不是成绩存档

配置说明游戏怎样玩，成绩说明游戏发生过什么。它们不是同一本账。当前插件保存config.yml，但报名与成绩只在内存里；停服后不会自动复活上一局，更不能从一个任务编号恢复倒计时。

### 配置不是成绩存档

#### 配置

说明游戏怎样玩

#### 本局数据

说明这一局发生什么

#### 未来历史

尚未实现就明确标注

想加历史榜怎么办：回到第29课的数据保存思路，明确格式版本、何时写入、读错怎么办。不要在每次破坏方块时把整份历史文件重写一次。

### 未来历史文件草案 尚未实现 · 位置见本节说明

```
schema-version: 1
# records可在未来保存比赛编号、玩家UUID和分数。
# 先设计校验、备份与失败处理，再编写持久化代码。
```

### 核对表

| 数据      | 当前保存位置       | 重启之后    |
|---------|--------------|---------|
| 世界与时长   | 磁盘config.yml | 再次读取并校验 |
| 本局报名与分数 | GameRound内存  | 清空      |
| 计时任务    | Paper运行时     | 取消，不恢复  |
| 历史成绩    | 当前没有实现       | 没有历史可读  |



马老师教编程  
www.teacher-ma.de

07 / 15

## 06 让说明书替你回答第一轮问题

README是作品的使用说明，不是给作者自己看的暗号。请一个没参与开发的同学照着安装；他停在哪一步，那一步通常就值得补充。下面是可改写的骨架，实际测试栏必须自己填写。

### 让说明书替你回答第一轮问题

#### 安装步骤

第一次使用者能照着做

#### 命令权限

玩家与管理员分清

#### 真实记录

已测、失败与待测分开

必须核对权限名：这里的blockacademy.orerush.admin要与第38课plugin.yml和代码一致。复制别人的说明时，最容易把别人的权限名也一块搬进来。

### README.md参考正文 · 位置见本节说明

# OreRush 1.0.0

用途：老师监督下的课堂挖矿计分赛。

环境：Java21、普通Paper1.21.11；不支持Folia。

安装：停服，将JAR放plugins，启动后检查日志。

配置：plugins/OreRush/config.yml；修改后停服再启动。

命令：/orerush join、score；管理员使用start、stop。

管理权限：blockacademy.orerush.admin，默认op。

规则：铁3、金5、钻石10分，均含深板岩版本。

条件：已报名、生存模式、指定世界、事件未被取消。

结果：自然结束广播本局分数；stop取消，不颁奖。

重启：清空报名与分数，不恢复本局；配置保留。

限制：无历史榜、无奖励、无矿区或矿石来源防作弊。

测试记录：写自己的日期、环境、通过项与未通过项。

反馈：向任课老师提交复现步骤和已脱敏的日志。

## 07 发布记录写变化 不写魔法

“更好更快更强”听起来像附魔广告，却没告诉玩家改了哪里。发布记录列出新增、修复和已知限制；测试记录另外保存实际证据。尚未做的改进放到计划区，不要伪装成已完成。

### 发布记录写变化 不写魔法

#### CHANGELOG

只写已实现的变化

#### TEST-RESULTS

留下本次真实证据

#### 下一版计划

未完成项不冒充发布功能

诚实记录也属于工程能力：本教材配套代码已通过编译和24项规则检查，但你修改后的版本需要重新测试。游戏内场景不能用“书上写了”代替真实执行。

#### CHANGELOG.md写法示例 • 位置见本节说明

# 1.0.0

新增：报名、分数查询、管理员开始和停止。

新增：六种矿石材料计分，含三种深板岩矿石。

新增：可配置世界和比赛时长。

限制：比赛不跨重启保存，不提供实物奖励。

#### TEST-RESULTS.md记录格式 • 位置见本节说明

日期与环境：按本次真实测试填写

用例编号 | 预期结果 | 实际结果 | 通过/失败/待测

失败项补充：复现步骤、日志位置、修复后的复测

构建版本：与最终交付的JAR保持一致

# 08 用干净测试服验收安装

自己的老服务器可能藏着旧配置、别的权限插件和上一次留下的JAR。请老师准备一个单独的本地测试目录，从新安装走一遍；需要测试保护交互时，再明确加入保护插件。测试环境也要能解释。

## 用干净测试服验收安装

### 干净安装

没有旧JAR或旧配置干扰

### 正常对局

两名玩家完整走一局

### 边界回归

24项规则与十个游戏场景

在专用本机测试服验收：不要为交作业开放公网端口、上传完整服务器或复制真实玩家资料。服务器许可与运行条件由老师确认，本书不替你接受协议。

## 核对表

| 检查点  | 看到什么才算过               |
|------|-----------------------|
| 环境   | Java和Paper版本符合说明      |
| 插件加载 | 无启动异常，命令已注册           |
| 首次配置 | 生成正确的数据目录与默认文件        |
| 普通玩家 | 不能使用start或stop        |
| 正常玩法 | 两人报名、计分、结束可复现         |
| 自动检查 | 重新运行并得到PASS 24 checks |
| 边界检查 | 第39课十个场景逐项记录          |
| 重启   | 旧局清空，配置仍在             |

## 09 更新失败要知道怎样退回去

更新像换铁路道岔，先让火车停稳。停服并等待退出，备份当前JAR与对应插件数据目录，再替换为一个新版本。若启动失败，先保留日志，停服后恢复相匹配的旧文件，不能在运行中乱换。

### 更新失败要知道怎样退回去



清楚标注作品身份：这是学习者制作的第三方练习插件，不要宣称获得Minecraft或Paper官方认证。发布到公开平台是下一项独立决定，不是本课自动完成的动作。

#### 本地交付包建议 · 位置见本节说明

```

release/
  OreRush-1.0.0.jar
  README.md
  CHANGELOG.md
  TEST-RESULTS.md
source/ # 源码、构建文件、wrapper、检查程序
    
```

#### PowerShell核对传输文件 · 位置见本节说明

```
Get-FileHash .\OreRush-1.0.0.jar -Algorithm SHA256
```

- SHA256一致可用于核对文件副本，不等于插件安全认证。
- 备份与回退要实际演练；未来若改变数据格式，更要验证旧程序能否读取。
- 不打包整个服务器、账户信息或私人日志；公开使用素材前核对来源与许可。

# 10 五分钟演示 先让同学玩到

演示不必把每一行import念成咒语。先让观众知道目标，再展示一局和一个被正确拒绝的操作。可以在演示前把配置调短，并说明是演示设置；发生意外时展示证据，比假装无事发生更专业。

## 五分钟演示 先让同学玩到

### 先体验

让观众看到一局

### 再讲规则

解释一个正确拒绝

### 最后证据

展示一次真实修复

准备可复现的备用方案：保存已测试版本和简短操作卡。若现场服务器不能用，可以解释日志与测试结果，但应明确这次没有完成现场游戏演示。

- 为什么用UUID作键，而不是只用玩家名字？
- 为什么先进入ENDING，再广播成绩？
- 重启保留什么、清除什么？如何证明没有第二个计时任务？

## 核对表

| 时间   | 演示内容                 |
|------|----------------------|
| 第1分钟 | 用一句话说明玩法与适用环境        |
| 第2分钟 | 两人报名，老师开始短局          |
| 第3分钟 | 挖矿查分，展示自然结算          |
| 第4分钟 | 重复start或未报名挖矿被挡住     |
| 第5分钟 | 解释GameRound职责与一次真实修复 |

# 11 毕业评分看证据 也看解释

评分不是数代码行数：复制一万行而说不清状态，不如亲手做出一个可靠的小插件。用同一张表检查作品，请试用伙伴给出能复现的反馈。建议具体到动作，不要只说“它有点怪”。

### 毕业评分看证据 也看解释

#### 运行与编程

能用且职责清楚

#### 体验与测试

反馈准确且可复现

#### 表达与改进

解释限制并选一条反馈

反馈不是命令清单：收集三条意见，选择最影响体验的一条。说明采纳或暂缓的理由，不用为了讨好每个人把插件改成会烤面包的末影龙。

### 把模糊反馈变为行动 • 位置见本节说明

模糊：为什么总是0分？  
具体：没有join就挖钻石，查询得到未报名提示。  
判断：属于规则拒绝；可以在说明里更早提醒先报名。  
下一步：先改提示，再请同学重新走一遍流程。

### 核对表

| 维度与分值    | 观察证据                 |
|----------|----------------------|
| 运行 30分   | 能安装、正常玩法完成、失败不破坏状态   |
| 编程 25分   | 职责清楚，正确使用条件、集合、方法与事件 |
| 玩家友好 20分 | 命令反馈、权限、配置和说明容易理解    |
| 测试 15分   | 正常与边界测试有真实记录，修复后复测   |
| 表达 10分   | 能解释关键设计、已知限制与下一步     |



马老师编程  
www.teacher-ma.de

13 / 15

# 12 拓展挑战与自己的验收记录

留下源代码、实际运行结果和一条问题记录。先用自己的话解释，再运行验证；结果与预测不同的地方，往往最值得讨论。

## 拓展任务与通关复盘

### 木镐

完成本地毕业包

### 铁镐

只选一个改进并验收

### 钻石镐

用三条反馈制定下一版

下一站：40节课的主线到这里完成。挑一个具体问题，独立经历“设计、编码、测试、说明”的下一轮冒险。  
官方延伸阅读：[docs.papermc.io/paper/dev/project-setup/](https://docs.papermc.io/paper/dev/project-setup/)

- 木镐：提交一个最小可玩作品、独立JAR、源码、说明书、真实测试记录，并做五分钟演示。
- 铁镐：从配置、权限反馈或历史记录中只选一个改进，补齐正常、边界和失败测试。
- 钻石镐：邀请三名同学试用，选一条反馈制定1.1.0计划；实现并验收后才把它列入已发布功能。
- 验收要求：完成一次5分钟演示和同桌独立安装；提交版本清单、已测场景、未测项目和可执行的回退步骤。

## 核对表

| 我能独立完成         | 已掌握                      | 再练一次                     |
|----------------|--------------------------|--------------------------|
| 能交付别人可以安装的作品   | <input type="checkbox"/> | <input type="checkbox"/> |
| 能解释核心规则与已知限制   | <input type="checkbox"/> | <input type="checkbox"/> |
| 会用反馈和测试推动下一次改进 | <input type="checkbox"/> | <input type="checkbox"/> |

# 毕业作品的下一轮冒险

第40课的终点是一份别人能安装、能核对、能给出反馈的作品。挑出一个真实问题：可以是更清楚的拒绝提示、更稳定的状态清理，或一次有合同的数据保存。把它写成下一版的一个验收条件。

## 下一版计划

| 问题   | 先定义结果       | 再留下证据       |
|------|-------------|-------------|
| 改进体验 | 新读者在哪一步不再卡住 | 让另一位试用者重走流程 |
| 修复缺陷 | 同一输入按合同处理   | 保留修复前后回归结果  |
| 增加功能 | 边界和失败策略写清楚  | 实现后才列入已发布变化 |

技术参考：本课使用 Paper 1.21.11 与 Java 21。  
版本 API： <https://jd.papermc.io/paper/1.21.11/>  
项目建立与发布结构： <https://docs.papermc.io/paper/dev/project-setup/>