Guide contents

Install and verify the toolchain

Set up a Java project

Add the first game object

Package and maintain

Sources / References

Documentation: In progress **Quick start: In progress**

Offline copy from ModDocs. Source links remain clickable.

Slay the Spire modding

For the original Slay the Spire, the established Java route uses **ModTheSpire** to load mod JARs and **BaseMod** for game-specific hooks and a development console. These projects target the first game; do not assume their APIs apply to a sequel. Read ModTheSpire's wiki and the BaseMod repository alongside this guide.

> **Quick start:** confirm ModTheSpire and BaseMod launch the game, create a Java 8 mod project using their current example, build a JAR, and verify that it appears in the loader before adding cards or characters.

Install and verify the toolchain

ModTheSpire's installation instructions place `ModTheSpire.jar` beside the game's `desktop-1.0.jar`, with mod JARs in the installation's `mods/` folder. Use the version of the launcher compatible with your installed game. BaseMod's README documents its own dependency on ModTheSpire and a Java 8 development path. Start once with just the launcher and BaseMod enabled. If that fails, stop before diagnosing a new card's Java code.

Use a separate game/profile for testing. Keep a copy of the original game JAR and your save. ModTheSpire can replace base files or patch code, but its wiki warns that wholesale file replacement conflicts with other mods; prefer its patch mechanism for targeted behavioral changes.

Set up a Java project

BaseMod's repository contains a `testmod` and Maven build configuration. Copy a compatible example project or follow the BaseMod wiki for project structure. Reference your installed game's JAR only for development; do not bundle it into a published mod. The README describes building with Maven and copying the resulting JAR to ModTheSpire's `mods/` directory. Use a unique package name and mod ID.

Before adding content, load a mod that only logs a startup message. Confirm the ModTheSpire launcher lists the JAR, BaseMod loads, and the message appears in the game log. If the JAR is absent, inspect its internal mod metadata and packaging; if it appears but fails, inspect Java version, missing dependencies and the first exception.

Add the first game object

BaseMod exposes hooks for cards, relics, potions, events and characters; the hook index documents the subscription points. Pick a single card or relic and follow a current example. A usable card needs a unique ID, text/localization, image, behavior, and registration through the appropriate BaseMod callback. Test it in a run, then test a save/reload and its interactions with upgrades or copies if relevant. A card class that compiles but never enters the game's pool is not complete content.

Use the BaseMod console reference for testing rather than relying on random runs to find a new object. Its README also notes that invalid console IDs can crash the game and certain commands can make a starting-room save unloadable, so use a disposable test run.

Package and maintain

Ship your own JAR and original assets, list ModTheSpire/BaseMod as dependencies, and state the game and Java versions you tested. Reinstall the package from scratch to catch accidentally referenced local files. Re-test after launcher or game updates; binary patches and hook signatures are version-sensitive.

This guide is **unfinished** beyond the loader and project path. It does not yet provide a complete, version-tested card implementation, art/localization files and clean-package test.

Sources / References

External project and documentation links cited in this guide. Check the upstream release and version before following version-sensitive steps.

- ModTheSpire's wiki
- BaseMod repository
- BaseMod wiki
- hook index
- BaseMod console reference