

The Lock On Mod - One-Page App Summary

Source basis: README.md, build.gradle, src/main/java, src/main/resources/mcmod.info

What it is

Zelda Targeting is a Minecraft Forge mod for Minecraft 1.12.2 that adds Zelda-style lock-on combat with camera steering, HUD overlays, and sound feedback. Repo metadata identifies version 1.3.0 and a client-focused architecture registered through Forge mod lifecycle hooks.

Who it is for

Primary persona: Minecraft 1.12.2 players and modpack users who want action-game style target locking and configurable combat HUD/audio behavior.

What it does

- Adds lock-on and target cycling keybinds (default R to toggle, Q/E to cycle).
- Finds valid targets using range, angle, line-of-sight, and entity-type filters.
- Supports target priority modes: nearest, health, threat, and angle.
- Tracks a locked target and steers camera with smoothing/preset options and optional auto third-person.
- Renders reticles and HUD stats (health, distance, damage prediction, hits-to-kill, vulnerabilities).
- Plays themed lock/switch/lethal/lost sounds with per-event volume/pitch plus cooldown protection.
- Shows animated floating damage numbers and uses observed damage events for post-hit accuracy.

How it works (repo-evidenced architecture)

- Entry + lifecycle: ZeldaTargetingMod delegates to CommonProxy/ClientProxy; config loads during preInit.
- Input/tick flow: TargetingManager listens on Forge event bus, handles key input, and updates lock state each client tick.
- Target pipeline: EntityDetector gathers candidates; TargetSelector ranks them; TargetTracker validates distance/dimension over time.
- Presentation + combat signals: TargetRenderer and DamageNumbersRenderer draw overlays; DamageEventListener caches LivingDamageEvent values; TargetingSounds handles audio cues.
- Config/UI path: GuiFactory opens GuiTargetingConfig, and TargetingConfig persists settings to zeldatargeting.cfg.
- External backend services or network API dependencies: Not found in repo.

How to run (minimal)

- Install prerequisites from repo docs: Java 8+, Minecraft 1.12.2, Forge 14.23.5.2859+.
- From repo root, generate IDE run configs: gradlew genIntelliJRuns or gradlew genEclipseRuns (README.txt).
- Import the Gradle project in your IDE and launch the generated client run configuration (build.gradle defines runs.client with workingDirectory run/).
- Single command run instruction in README (for example, explicit gradlew runClient step): Not found in repo.