Minegicka Legacy — Dossier de projet (Forge 1.20.1, Windows 11)

> **But** : Porter le mod *Minegicka3* (Minecraft 1.7.10) vers **Forge 1.20.1** en environnement **Windows 11**, avec une boîte à outils de refactor (OpenRewrite, Error Prone/Refaster, Semgrep) et un process reproductible. Ce document sert d'index pour les IA et pour les devs.

---

1) Contexte & objectifs

- **Legacy (1.7.10)** : repo d'origine *Minegicka3* (lecture/audit, éventuellement compilation via RFG).
- **Cible (1.20.1)** : nouveau workspace Forge pour le port, avec génération de données (Datagen) et APIs modernes (DeferredRegister, Capabilities, netcode moderne, modèles JSON, etc.).
- **Langage** : Java 17 (requis pour MC 1.20.x).

---

2) Emplacements des projets

- **Projet de port (Forge 1.20.1)** : `C:\Users\ulyss\mods\minegicka_port_1_20_1`
- **Legacy (lecture)** : `C:\Users\ulyss\dev\Minegicka3`
- **Legacy (compilable avec RFG)** : `C:\Users\ulyss\dev\minegicka3-legacy-rfg`

Ces chemins sont la référence pour toutes les commandes/CI.

---

3) Outils & installations (Windows 11)

3.1 Système & gestionnaires

- **winget** pour installations graphiques/CLI.
- **Windows Terminal / PowerShell** pour toutes les commandes.

3.2 SDK & utilitaires

- **JDK 17** (Temurin) — requis par Forge 1.20.x
- **Git**
- **ripgrep (rg)**, **fd** — recherche ultra-rapide dans le code/ressources
- **jq**, **yq**, **Taplo CLI** — JSON/YAML/TOML (mods.toml, configs)
- **Microsoft PowerToys → PowerRename** — renommage massif d'assets
- **Python 3.12** + **Semgrep** (ou via **pipx**)
- **(Optionnel) WSL + Ubuntu** pour **Comby** (recherches/remplacements structurés cross-lang)

3.3 IDE & assistants IA

- **IntelliJ IDEA Community** + plugin **Minecraft Development**
- **Assistants IA** (optionnels) : GitHub Copilot Chat, Sourcegraph Cody, JetBrains AI Assistant

4) Workspace Forge 1.20.1 (projet cible)

4.1 Création

1. Télécharger le **MDK 1.20.1** (Forge) → dézipper dans `C:\Users\ulyss\mods\minegicka_port_1_20_1`.
2. Générer les runs : `./gradlew genIntellijRuns`

4.2 Fichier `gradle.properties` (référence)

```
org.gradle.jvmargs=-Xmx3G
org.gradle.daemon=true

# Environnement
minecraft_version=1.20.1
minecraft_version_range=[1.20.1,1.21)
forge_version=47.4.8
forge_version_range=[47,)
loader_version_range=[47,)
mapping_channel=official
mapping_version=1.20.1

# Mod
mod_id=minegicka
mod_name=Minegicka Legacy
mod_license=MIT
mod_version=0.1.0
mod_group_id=com.alco.minegickalegacy
mod_authors=alco
mod_description=MAGIC !.
```

4.3 `mods.toml` (placeholders à garder)

```
modLoader="javafml"
loaderVersion="${loader_version_range}"
license="${mod_license}"

[[mods]]
modId="${mod_id}"
version="${mod_version}"
displayName="${mod_name}"
description='''${mod_description}'''

[[dependencies.${mod_id}]]
modId="forge"
mandatory=true
versionRange="${forge_version_range}"
ordering="NONE"
```

```
side="BOTH"

[[dependencies.${mod_id}]]
modId="minecraft"
mandatory=true
versionRange="${minecraft_version_range}"
ordering="NONE"
side="BOTH"
```

4.4 Classe principale (exemple minimal correct)

Fichier : `src/main/java/com/alco/minegickalegacy/MinegickaMod.java`

```
package com.alco.minegickalegacy;

import net.minecraftforge.fml.common.Mod;

@Mod(MinegickaMod.MODID)
public class MinegickaMod {
    public static final String MODID = "minegicka";

    public MinegickaMod() {
        // init (events, registries, etc.)
    }
}
```

> **Règle** : nom de fichier = nom de la classe publique. `@Mod("minegicka")` doit matcher `mod_id`.

4.5 Tâches Gradle utiles

- Lancer le client dev : `./gradlew runClient`
- Lancer la **DataGen** : `./gradlew runData`
- Nettoyer/compilation : `./gradlew clean build`

---

5) Outils de refactor dans le projet Forge

5.1 OpenRewrite

`build.gradle` (extraits) :

```
plugins {
    id "java"
    id "org.openrewrite.rewrite" version "7.16.0" // ajustable
}
```

```
repositories { mavenCentral() }

dependencies {
    rewrite(platform("org.openrewrite.recipe:rewrite-recipe-bom:2.15.0"))
    rewrite("org.openrewrite.recipe:rewrite-migrate-java")
}

rewrite {
    activeRecipe("minegicka.port.v1")
}
```

`rewrite.yml` (racine) :

```
type: specs.openrewrite.org/v1beta/recipe
name: minegicka.port.v1
recipeList:
  - org.openrewrite.java.ChangePackage:
      oldPackageName: "cpw.mods.fml"
      newPackageName: "net.minecraftforge.fml"
      recursive: true
  - org.openrewrite.java.ChangeType:
      oldFullyQualifiedTypeName: "net.minecraft.util.ResourceLocation"
      newFullyQualifiedTypeName: "net.minecraft.resources.ResourceLocation"
  - org.openrewrite.java.migrate.JavaUtilObjects:
      minimumJavaVersion: 17
```

Commandes : - Appliquer les recettes : `./gradlew rewriteRun`

5.2 Error Prone (Refaster prêt pour plus tard)

`build.gradle` (extraits) :

```
plugins {
    id "net.ltgt.errorprone" version "4.0.1"
}

dependencies {
    errorprone("com.google.errorprone:error_prone_core:2.26.1")
}

tasks.withType(JavaCompile).configureEach {
    options.errorprone.enabled = true
    // Pour appliquer des patchs Refaster quand on en aura :
    // options.errorprone.errorproneArgs += ["-XepPatchChecks:Refaster", "-
XepPatchLocation:IN_PLACE"]
}
```

Commandes : `./gradlew clean compileJava`

5.3 Semgrep

Arborescence : `semgrep-rules/java/port.yml`

```
rules:
  - id: minegicka-detect-ieep
    languages: [java]
    severity: WARNING
    message: "IEEP détecté : migrer vers Capabilities"
    patterns:
      - pattern: class $C implements
net.minecraft.entity.player.IExtendedEntityProperties { ... }
  - id: minegicka-detect-simple-network
    languages: [java]
    severity: WARNING
    message: "SimpleNetworkWrapper détecté : migrer vers netcode moderne"
    patterns:
      - pattern:
net.minecraftforge.fml.common.network.simpleimpl.SimpleNetworkWrapper $X
= ...
```

Commandes : `semgrep --config .\semgrep-rules\java\port.yml`

---

6) Audit du projet legacy (Minegicka3)

6.1 Lecture simple

Repo : `C:\Users\ulyss\dev\Minegicka3` - Greps utiles : - `rg -n "cpw\.mods\.fml|GameRegistry|IExtendedEntityProperties|SimpleNetworkWrapper|IIcon|ItemRenderer|WorldGen|TileEntitySpecialRenderer" src` - `fd -H -t f src\main\resources` - Sortie attendue : liste d'APIs legacy à migrer vers : DeferredRegister / Capabilities / netcode moderne / Datagen + modèles JSON.

6.2 Audit compilable (RFG)

Repo : `C:\Users\ulyss\dev\minegicka3-legacy-rfg` - `settings.gradle` → `pluginManagement` avec Maven GTNH - `build.gradle` → plugin `com.gtnewhorizons.retrofuturagradle` (ex: `1.4.6`) - Commandes : `./gradlew wrapper`, `./gradlew tasks`, `./gradlew build`

---

7) Processus de travail recommandé

1. **Configurer** le workspace Forge (Étapes 4.1–4.4).
2. **Brancher** OpenRewrite + Error Prone + Semgrep (Étape 5).

3. **Copier** quelques classes du legacy → projet 1.20.1.
4. **Exécuter** `rewriteRun` → corriger imports/types mécaniques.
5. **Compiler** (`compileJava`) → traiter les erreurs restantes par refactor ciblé.
6. **Écrire** des recettes supplémentaires (OpenRewrite) et **règles Refaster** pour les swaps 1:1 stables.
7. **DataGen** (`runData`) pour générer JSON (recipes/loot/tags/models) et limiter l'édition manuelle.
8. **Tester** avec `runClient`.

---

8) Consistances & pièges fréquents

- **MODID** : `@Mod("minegicka")` = `mod_id=minegicka` = `mods.toml modId = ${mod_id}`.
- **Group/Package** : `mod_group_id=com.alco.minegickalegacy` ↔ `package com.alco.minegickalegacy;` (chemin des sources).
- **Licence** : aligner `mod_license` et `license` (ex: MIT partout).
- **Placeholders** : préférer `${...}` dans `mods.toml` pour éviter les divergences (ranges Forge/Minecraft/loader, nom, version, description).
- **Nom du fichier** = **nom de la classe publique** (ex: `MinegickaMod.java`).
- `ResourceLocation` : préférer `ResourceLocation.tryParse` (évite les warnings/removals).

---

9) Commandes récap (Windows)

```
# Forge 1.20.1
cd C:\Users\ulyss\mods\minegicka_port_1_20_1
.\gradlew.bat genIntellijRuns
.\gradlew.bat runClient
.\gradlew.bat runData
.\gradlew.bat rewriteRun
.\gradlew.bat clean compileJava

# Semgrep
semgrep --config .\semgrep-rules\java\port.yml

# Legacy (RFG)
cd C:\Users\ulyss\dev\minegicka3-legacy-rfg
.\gradlew.bat wrapper
.\gradlew.bat tasks
.\gradlew.bat build
```

---

10) Point d'entrée "mémoire IA" (copier/coller)

- **OS** : Windows 11 (FR), terminal PowerShell.
- **Cible** : Forge **1.20.1** (Java 17, FML 47.x), MDK installé.
- **Projet** : `C:\Users\ulyss\mods\minegicka_port_1_20_1`.

- **Legacy** : `C:\Users\ulyss\dev\Minegicka3` (lecture), `C:\Users\ulyss\dev\minegicka3-legacy-rfg` (build via RFG).
- **mod_id** = `minegicka`, **group** = `com.alco.minegickalegacy`, **main class** = `com.alco.minegickalegacy.MinegickaMod`.
- **Gradle props** : versions/ranges Forge/Minecraft en placeholders, licence **MIT**.
- **Refactor** : OpenRewrite (`rewrite.yml` → `minegicka.port.v1`), Error Prone (Refaster prêt), Semgrep (détection IEEP/SNW).
- **Data** : utiliser **runData** pour générer JSON.
- **Règle** : toujours aligner MODID / group / package / mods.toml.

Fin — Ce document doit être tenu à jour au fil du port. Ajouter ici :

- Recettes OpenRewrite spécifiques (GameRegistry → DeferredRegister, IEEP → Capabilities, SNW → netcode moderne).
- Décisions de licence/dépendances, matrices de compatibilité, et TODOs.

---

11) IntelliJ — runs disponibles & quand les utiliser

- **runClient** : lance le **client** (avec serveur intégré). À utiliser pour tout ce qui est **rendu/UI/input**, tests rapides d'items/blocs, overlays HUD, keybinds. Astuce : F3+T pour recharger les assets.
- **runServer** : lance un **serveur dédié** (sans client). À utiliser pour la **logique serveur**, worldgen, netcode, perfs serveur et détection d'appels client côté serveur. Connecte un client pour tester la synchro.
- **runData** : exécute la **DataGen** (ne lance pas le jeu). Génère JSON (recipes/loot/blockstates/models/tags) dans `src/generated/resources`. Workflow : `runData` → commit des JSON → `runClient`.
- **runGameTestServer** : exécute les **GameTests** (tests automatisés de gameplay), produit un rapport et s'arrête. Idéal pour CI.

  Mémoire (si nécessaire) : **Run > Edit Configurations… > VM options** → `-Xmx3G`.

---

12) Setup double IDE (IntelliJ = ancre Forge, VS Code = cockpit IA)

- **IntelliJ IDEA + Minecraft Development** : runs, debug, refactors sûrs, SSR.
- **VS Code + extension(s) OpenAI** : pilotage IA (génération/édition guidée dans le repo), sans toucher aux runs.

Cohabitation propre

1. **Un seul run** (client/serveur) à la fois.
2. **Gradle daemon activé** (`org.gradle.daemon=true`).
3. **Style/formatage unifié** via EditorConfig/Spotless.

VS Code (facultatif) — tâches Gradle prêtes

Crée `.vscode/tasks.json` :

```
{
  "version": "2.0.0",
  "tasks": [
    { "label": "gradle: runClient",    "type": "shell", "command": ".
\gradlew.bat runClient" },
    { "label": "gradle: runData",      "type": "shell", "command": ".
\gradlew.bat runData" },
    { "label": "gradle: rewriteRun",   "type": "shell", "command": ".
\gradlew.bat rewriteRun" },
    { "label": "gradle: gameTests",    "type": "shell", "command": ".
\gradlew.bat runGameTestServer" }
  ]
}
```

---

13) EditorConfig & formatage (pour IA + humains)

Place **.editorconfig** à la racine (sans extension) :

```
root = true

[*]
charset = utf-8
end_of_line = lf
insert_final_newline = true
trim_trailing_whitespace = true

[*.java]
indent_style = space
indent_size = 4
```

Optionnel : **Spotless** dans `build.gradle` pour enforcement automatique :

```
plugins { id "com.diffplug.spotless" version "6.25.0" }
spotless {
  java { googleJavaFormat(); target 'src/**/*.java' }
}
```

Commandes : `./gradlew spotlessApply` puis `spotlessCheck` en CI.

---

14) Rappels de cohérence

- **MODID** : `@Mod("minegicka")` = `mod_id=minegicka` = `${mod_id}` dans `mods.toml`.

- **Group/Package** : `mod_group_id=com.alco.minegickalegacy` ↔ `package com.alco.minegickalegacy;` (et chemin des sources).
- **Licence** : même valeur dans `gradle.properties` et `mods.toml` (ex. **MIT**).
- **Placeholders** : utiliser `${forge_version_range}`, `${minecraft_version_range}`, etc., dans `mods.toml`.
- **DataGen** : inclure `src/generated/resources` dans les resources (MDK récent : déjà fait).

---

15) Nouveau récap commandes (inclut GameTests)

```
# Projet Forge 1.20.1
cd C:\Users\ulyss\mods\minegicka_port_1_20_1
.\gradlew.bat genIntellijRuns
.\gradlew.bat runClient
.\gradlew.bat runServer
.\gradlew.bat runData
.\gradlew.bat runGameTestServer
.\gradlew.bat rewriteRun
.\gradlew.bat clean compileJava

# Semgrep
semgrep --config .\semgrep-rules\java\port.yml

# Legacy (RFG)
cd C:\Users\ulyss\dev\minegicka3-legacy-rfg
.\gradlew.bat wrapper
.\gradlew.bat tasks
.\gradlew.bat build
```