

# Minegicka Legacy → Forge 1.20.1 — Étape 3

## « Lecture froide »

Ce document résume **tout ce qui a été fait** pendant la mise en place de l'audit « legacy », ainsi que **les infos essentielles** à fournir à une IA pour poursuivre efficacement. Dernière mäj : **19 sept. 2025** (Europe/Paris).

---

### 1) Objectif de l'étape 3

Produire une **carte des migrations** à partir d'une **lecture froide** + (option) d'un **audit compilable RFG** pour obtenir des erreurs concrètes.

Surfaces à cartographier → cibles 1.20.1 : - **Registries** (GameRegistry.) → *DeferredRegister* - *IEEP* (*IExtendedEntityProperties*) → *Capabilities* - *Netcode* (*SimpleNetworkWrapper*) → *SimpleChannel* / réseau moderne - *Rendu* (*IIcon*, *ItemRenderer*, *TESR*) → *Modèles JSON* + *BlockEntityRenderer* si besoin - *Worldgen* (*IWorldGenerator*/*WorldGen*) → *Features/Placements* + *Biome Modifiers* - *Assets* → *Datagen* (*models/loot/recipes*) + *JSON\**

Livrables attendus : fichiers de **grep** par thème, un **rfg-build.log** (si audit compilable), et un **migration-map.md**.

---

### 2) État de l'environnement (confirmé)

- **OS** : Windows 11 x64.
- **Java runtime pour outils récents** : **Temurin 17.0.16+8** (OK).
- **Gradle (système)** : **9.1.0** installé manuellement dans `C:\Gradle\gradle-9.1.0` (OK).
- **⚠ Ne pas** utiliser `gradle` (9.x) dans le projet Forge ; utiliser **uniquement le wrapper**.
- **Projet Forge 1.20.1 (cible)** : wrapper **Gradle 8.8** opérationnel (`.\gradlew.bat tasks` OK).
- **Projet RFG (audit compilable 1.7.10)** : wrapper **Gradle 8.8** opérationnel, tâches RFG visibles, `build` OK « à vide ».
- **Toolchain Java 8** : **Temurin 8u462** installée.
- `gradle.properties` présent ; auto-détection JDK activée. (NB : si un chemin figé est indiqué, vérifier qu'il correspond bien à la version installée.)

Chemins clés : - **LEGACY** : `C:\Users\ulyss\dev\Minegicka3` - **RFG** : `C:\Users\ulyss\dev\minegicka3-legacy-rfg` - **Forge 1.20.1** : `C:\Users\ulyss\mods\minegicka_port_1_20_1`

Bonnes pratiques PowerShell : - Ne **jamais** coller le prompt `PS C:\>` avec les commandes. - Exécuter `./gradlew.bat` **dans** le dossier qui contient le wrapper.

---

### 3) Configuration RFG utilisée (finale)

**settings.gradle** (sans BOM) :

```
pluginManagement {
    repositories {
        maven {
            name = "GTNH Maven"
            url = "https://nexus.gtnewhorizons.com/repository/public/"
            content {
                includeGroup("com.gtnewhorizons")
                includeGroup("com.gtnewhorizons.retrofuturagradle")
            }
        }
        gradlePluginPortal()
        mavenCentral()
    }
}

// Optionnel (recommandé si on veut auto-provision des toolchains) :
plugins {
    id "org.gradle.toolchains.foojay-resolver-convention" version "0.7.0"
}

rootProject.name = "minegicka3-legacy-rfg"
```

**build.gradle** (DSL plugins, sans BOM) :

```
plugins {
    id "java"
    id "com.gtnewhorizons.retrofuturagradle" version "1.4.6"
}

repositories {
    maven { url = "https://maven.minecraftforge.net" }
    maven { url = "https://nexus.gtnewhorizons.com/repository/public/" }
    mavenCentral()
}

tasks.withType(JavaCompile).configureEach {
    options.encoding = "UTF-8"
}
```

**gradle.properties** (si nécessaire pour toolchain locale) :

```
org.gradle.java.installations.auto-detect=true
# Adapter ce chemin si on fige une installation précise :
```

```
# org.gradle.java.installations.paths=C:\\Program Files\\Eclipse Adoptium\\  
\\jdk-8.0.462.0-hotspot
```

Wrapper : - Généré avec **Gradle 8.8** (C:\\Gradle\\gradle-8.8\\bin\\gradle.bat wrapper). -  
Toujours appeler ./gradlew.bat ... ensuite.

## 4) Ce qui a été fait (chronologie synthétique)

1. **Java 17** et **Gradle 9.1** installés (usage système).
2. Projet **Forge 1.20.1** : wrapper 8.8 détecté, `tasks` OK. **Consigne** : n'utiliser que `./gradlew.bat` (8.8), pas `gradle` (9.x).
3. Création du projet **RFG** :
4. Ajout de `settings.gradle` (Nexus GTNH) et `build.gradle` (plugin RFG 1.4.6).
5. Correction d'un **BOM/ZWSP** au début du fichier (erreur `?pluginManagement`).
6. Correction du **repo plugin** (passage à `nexus.gtnewhorizons.com`).
7. Génération du **wrapper 8.8** via Gradle 8.8.
8. **Résolution du plugin RFG** OK, `tasks` OK.
9. Toolchain **Java 8** :
10. Option Foojay (plugin) ajoutée **et/ou** installation locale **Temurin 8u462**.
11. `tasks` / `build` fonctionnent ; RFG télécharge et prépare l'environnement 1.7.10.
12. **Tentative de copie des sources legacy** : chemin standard `src\\main\\java` introuvable → prévu un script pour **détecter** et **copier**/lier automatiquement (voir §6).

## 5) Prochaines actions (pour clore l'étape 3)

1) **Brancher les sources legacy** dans le projet RFG (copie ou symlink), via le script §6. 2) Lancer `./gradlew.bat clean build -x test` dans `minegicka3-legacy-rfg` et **capturer** `rfg-build.log`. 3) Lancer les **greps** (`rg / fd`) sur le repo legacy et déposer les fichiers dans `audit-legacy`. 4) Compléter **migration-map.md** (modèle §7) à partir des fichiers `*.txt` et des erreurs de compilation.

**Definition of Done (Étape 3)** : - `audit-legacy\\registries.txt`, `ieep.txt`, `netcode.txt`, `renders.txt`, `worldgen.txt`, `resources.txt` - `audit-legacy\\rfg-build.log` - `audit-legacy\\migration-map.md` rempli

## 6) Script PowerShell — détection & copie des sources legacy

À exécuter depuis **RFG** : `C:\\Users\\ulyss\\dev\\minegicka3-legacy-rfg`

```
$LEGACY = "C:\\Users\\ulyss\\dev\\Minegicka3"  
$RFG     = "C:\\Users\\ulyss\\dev\\minegicka3-legacy-rfg"  
$DST_J   = Join-Path $RFG "src\\main\\java"  
$DST_R   = Join-Path $RFG "src\\main\\resources"  
New-Item -ItemType Directory -Force -Path $DST_J, $DST_R | Out-
```

```

Null

$candidatesJava = @(
    "src\main\java","src\minecraft","src\java","src"
) | ForEach-Object { Join-Path $LEGACY $_ }
$javaSrc = $candidatesJava | Where-Object { Test-Path $_ } |
Select-Object -First 1
if (-not $javaSrc) {
    $javaSrc = (Get-ChildItem -Path $LEGACY -Recurse -Directory -
ErrorAction SilentlyContinue |
                Where-Object { (Get-ChildItem -Path $_.FullName -
Filter *.java -Recurse -ErrorAction SilentlyContinue | Measure-
Object).Count -gt 50 } |
                Select-Object -First 1).FullName
}
if (-not $javaSrc) { Write-Error "Impossible de trouver les
sources Java dans $LEGACY"; return }

$candidatesRes = @(
    "src\main\resources","src\resources","resources","src\minecraft\resources"
) | ForEach-Object { Join-Path $LEGACY $_ }
$resSrc = $candidatesRes | Where-Object { Test-Path $_ } | Select-
Object -First 1

robocopy $javaSrc $DST_J /E /MT
if ($resSrc) { robocopy $resSrc $DST_R /E /MT }

$AUDIT = Join-Path $LEGACY "audit-legacy"
New-Item -ItemType Directory -Force -Path $AUDIT | Out-Null
Set-Location $RFG
./gradlew.bat clean build -x test *>&1 | Tee-Object "$AUDIT\rfg-
build.log"

```

## 7) Modèle — Carte des migrations (audit-legacy/migration-map.md)

# Carte des migrations Minegicka3 → Forge 1.20.1

Surface legacy	Indices (fichier)	Fichiers clés
Cible 1.20.1	Actions de migration	
-----	-----	-----
Registries (GameRegistry.*)	registries.txt	...
DeferredRegister	Créer DeferredRegister pour	
items/blocs		
Données joueur (IEEP)	iEEP.txt	...
Capabilities	Capability + storage +	
attach		

Réseau (SimpleNetworkWrapper)	netcode.txt	...	
Netcode Forge moderne (SimpleChannel)	Channel, packets,		
handlers			
Rendu (IIcon/ItemRenderer/TESR)	renders.txt	...	
Modèles JSON + BlockEntityRenderer (si BE)	Datagen models + BER si		
nécessaire			
Worldgen (IWorldGenerator/WorldGen)	worldgen.txt	...	
Worldgen moderne + Biome Modifiers	Features/Placements + data		
JSON			
Assets (textures/lang/etc.)	resources.txt	...	
Datagen + modèles JSON	runData, générer models/loot/		
recipes			

## 8) Greps — cheat-sheet (lecture froide)

À exécuter depuis n'importe où (nécessite `rg` et `fd`).

```
$LEGACY = "C:\Users\ulyss\dev\Minegicka3"
$OUT = Join-Path $LEGACY "audit-legacy"
New-Item -ItemType Directory -Force -Path $OUT | Out-Null

rg -n --no-heading -S -g "src/**" `
    "cpw\.mods\.fml|GameRegistry|IExtendedEntityProperties|
    SimpleNetworkWrapper|IIcon|IconRegister|ItemRenderer|WorldGen|
    TileEntitySpecialRenderer" `
    $LEGACY > "$OUT\01_greps_core.txt"

rg -n -S -g "src/**" "GameRegistry" $LEGACY >
"$OUT\registries.txt"
rg -n -S -g "src/**" "IExtendedEntityProperties" $LEGACY >
"$OUT\ieep.txt"
rg -n -S -g "src/**" "SimpleNetworkWrapper" $LEGACY >
"$OUT\netcode.txt"
rg -n -S -g "src/**" "IIcon|IconRegister|ItemRenderer|
TileEntitySpecialRenderer" $LEGACY > "$OUT\renders.txt"
rg -n -S -g "src/**" "WorldGen|IWorldGenerator" $LEGACY >
"$OUT\worldgen.txt"

# Ressources
fd -H -t f ".*" "$LEGACY\src\main\resources" >
"$OUT\resources.txt"
```

## 9) Pièges rencontrés & correctifs

- **Prompt collé** ( `PS C:\...>` dans la commande) → erreurs bizarres → **taper uniquement** la commande.

- **Wrapper absent** → générer avec Gradle 8.8 (`gradle.bat wrapper`) puis utiliser **uniquement** `./gradlew.bat`.
- **ForgeGradle vs Gradle 9.x** → ne pas appeler `gradle` (9.x) dans le projet Forge ; rester sur wrapper 8.8.
- **BOM/ZWSP** en tête de fichiers Gradle → réécrire les fichiers **sans BOM**.
- **Repo plugin RFG** mal configuré → utiliser **Nexus GTNH** dans `pluginManagement`.
- **Toolchain JDK 8** manquante → plugin **Foojay** ou installation **Temurin 8** + `gradle.properties`.

## 10) Ce qu'il faut fournir à une IA pour continuer

Copie/colle ce bloc dans le prompt de l'IA pour reprendre immédiatement.

- **Contexte** : Migration d'un mod 1.7.10 (Minegicka3) → Forge 1.20.1. On est à l'**étape 3** (audit legacy) : produire la **carte des migrations**.
- **Chemins** : `LEGACY=C:\Users\ulyss\dev\Minegicka3`, `RFG=C:\Users\ulyss\dev\minegicka3-legacy-rfg`, `FORGE=C:\Users\ulyss\mods\minegicka_port_1_20_1`.
- **Versions/outils** : Java 17 (outils), Gradle système 9.1 (ne pas utiliser sur Forge), wrappers **Gradle 8.8** (Forge & RFG), **RFG 1.4.6**, JDK 8 Temurin installé.
- **RFG config** : `settings.gradle` avec `pluginManagement` pointant **Nexus GTNH** + (option) plugin **Foojay**. `build.gradle` avec `plugins { java; com.gtnewhorizons.retrofuturagradle 1.4.6 }`.
- **État** : RFG **build OK à vide**. Brancher/copier les sources legacy peut être nécessaire (structure non standard). Voir script §6.
- **Tâches à exécuter** : 1) Copier/symlink sources legacy → RFG, lancer `./gradlew.bat clean build -x test`, récolter `rfg-build.log`. 2) Lancer les **greps** (cmds §8) pour `registries.txt`, `ieep.txt`, `netcode.txt`, `renders.txt`, `worldgen.txt`, `resources.txt`. 3) Remplir **migration-map.md** (modèle §7) et proposer **priorités** d'attaque.
- **Contraintes** : ne **pas** upgrader le wrapper Gradle Forge en 9.x ; ne **pas** coller le prompt PowerShell ; conserver fichiers Gradle **sans BOM**.
- **Livrables** : tous les fichiers `audit-legacy/*` + `migration-map.md` rempli.

## 11) Références rapides (exécution)

- Forge 1.20.1 :

```
cd C:\Users\ulyss\mods\minegicka_port_1_20_1
.\gradlew.bat --version # doit afficher Gradle 8.8
.\gradlew.bat tasks
```

- RFG (audit compilable) :

```
cd C:\Users\ulyss\dev\minegicka3-legacy-rfg
.\gradlew.bat tasks
.\gradlew.bat clean build -x test
```

---

## Fin ♦

Ce document sert de **brief IA** et de **checklist** opérable pour finaliser l'étape 3 et amorcer l'étape migration/refactor. N'hésite pas à l'enrichir avec les **premiers extraits** `rfg-build.log` + greps pour générer la carte des migrations priorisée.