

# Minegicka — Résumé d'avancement (au 20/09/2025)

## 1) DataGen & Assets (rappel synthétique)

- Providers couvrant tous les blocs/items : MinegickaBlockstateProvider, MinegickaItemModelProvider, MinegickaLootTableProvider.
- Item models → textures vanilla (au départ) ; LangProvider avec addSafe ; runData -PdisableErrorprone OK.
- Assets legacy présents pour GUI/entités ; helpers MinegickaTextures ; placeholders utilisés côté items/blocs (au départ).

## 2) Réseau, HUD & Événements (rappel synthétique)

- MinegickaNetwork + SpellTriggerPacket ; ServerSpellManager diffuse start/stop et cache le sort actif.
- ClientSpellManager met à jour overlay/toasts ; ManaOverlayRenderer affiche mana, combo, sort courant.
- PlayerManaEvents stop les sorts sur logout/respawn/mort/changement de dimension ; JSON inchangés sauf 2 nouvelles clés de langue.

## 3) Statut gameplay (rappel synthétique)

- Les logiques lourdes (CoreClient/CoreServer, Spell complet, GUIs, renderers) restauraient encore à porter initialement ; packets surtout textuels.

## 4) Spell Lifecycle & Server Management (NOUVEAU)

- Ajout des enums SpellStopReason/SpellForm et des contrats de cycle de vie (SpellExecutor, SpellContext, SpellExecutors, SpellInstance).
- Les casts actifs traquent durée, tick updates et raisons d'arrêt tout en réutilisant les effets MagickDefinition (start/intervals).
- Refonte de ServerSpellManager : possède les SpellInstance, registres par niveau, diffuse start/stop, helpers de teardown (manuel/niveau/joueur).
- Flux gameplay : SpellCasting.cast délègue à ServerSpellManager.startSpell ; combos vides → stopSpell (SpellCastRequestPacket).
- Événements & ticks : hooks sur ticks serveur/joueur pour garder l'état des sorts en phase ; capability events stoppent sur logout/mort/changement de dimension.
- Validation : ./gradlew compileJava OK ; avertissements ErrorProne existants (ordinals d'enum, enqueueWork futures, lookups FML dépréciés).
- À suivre (optionnel) : étendre les executors (drain mana, spawns d'entités), tests/validations in-game pour sorts répétés/interruption, adresser les warnings ErrorProne.

## 5) Exécuteurs, attributs & consommation (NOUVEAU)

- Utilitaires partagés (SpellAttributes, SpellContext) : lecture des stats de staff, taille de combinaison, drain de mana via capability avec feedback joueur.

- Sélection d'exécuteurs refaite (SpellExecutors) : variantes soutenues qui re■déclenchent les effets selon timers, durées à la legacy, arrêt si sustain cost échoue.
- Effets mis à jour (MagickEffects) pour réutiliser le helper staff commun et éviter la duplication de logique de stats.
- Tests : ./gradlew compileJava OK ; ajuster les multiplicateurs de sustain quand les entités de sorts seront portées.

## 6) Icônes d'items & pipeline textures (NOUVEAU)

- Remplacement des fallbacks (fer/vanilla) par des modèles dédiés Minegicka : chaque entrée de registre pointe vers une icône texturée du pack.
- 38 textures stylisées créées (sphères colorées, tiges teintées, essences gemmifiées, staffs/chapeaux) en ressources/assets/minegicka/textures/item/.
- Script tools/generate\_textures.py (Pillow) pour générer/mettre à jour les icônes selon des règles de couleur (Values/Element).
- runData régénéré : chaque item model référence les nouvelles textures (src/generated/.../models/item/\*).
- Next steps : si couleurs/items évoluent → relancer tools/generate\_textures.py puis ./gradlew runData ; lancer le client pour prévisualiser.

Document mis à jour automatiquement le 20/09/2025.