Wooly dungeons development outline

From chatgpt and me:

help me develop this mod idea, I will also be using gemini as another developer all three of will be a team got it?

got it — team of three (you + me + gemini). here's a tight, build-ready plan that turns your "wooly dungeons" idea (Vault Hunters × CTM × Aether-style portals) into a shippable MVP, with concrete tasks we can divide up.

elevator pitch

Wooly Dungeons: 16 color-keyed portals lead to procedurally assembled puzzle/gauntlet dungeons. Beat a dungeon to earn that color's **Wool Block** for the **Monument** (CTM tradition), plus milestone bonuses (gold → emerald → diamond → netherite). Each portal run has a visible countdown; when time's up, the portal seals and the run ends. Inspirations: CTM/Vechs' Super Hostile (16 wools on a Victory Monument) and Vault Hunters' timed vaults, plus Aether-style custom portals. (superhostile.fandom.com)

target stack & templates

- **Multi-loader**: start with the **MultiLoader Template** (common code + Fabric + NeoForge subprojects). It keeps us future-proof and lets pack makers choose their loader. (GitHub)
- **Dimensions & portals (Fabric first)**: use Fabric's dimension & **Custom Portal API** pattern to define color-keyed portals with **wool frames** (block tag). We can mirror on NeoForge after MVP. (Fabric Wiki)
- **Dungeons**: build with Minecraft's **Jigsaw** structure system (data-driven rooms stitched at runtime). This gives "Zelda-like" modular puzzles without heavy codegen. (Fabric Wiki)

core systems (MVP scope)

1. **Portals (16 colors)**

   - Frame = any rectangle of blocks in `#wool_portal_frames` (i.e., all 16 wool colors). Activation item = Shears on the frame or a “Wool Catalyst.” Each color links to its own sub-dimension theme. (Aether used glowstone+water; we’ll echo the “non-obsidian” vibe with wool.) (aether.fandom.com)

2. **Timed runs**

   - Global per-portal **timer** displayed on a bossbar/overlay; default MVP = **45 min** (you proposed 3h; we can tune — Vault Hunters uses 25m as a baseline, so players will recognize the mechanic). Warnings at T-60, T-15, T-5, T-1. On timeout, auto-extract players and **seal** the portal until a cooldown expires. (Vaults are timed; bossbar/scoreboard timers are standard.) (vault-hunters.fandom.com)

3. **Dungeons via Jigsaws**

   - Each color = **theme + pool** of rooms (puzzle, combat, parkour, key/lock). We’ll ship 10–12 rooms per color in MVP and let the jigsaw pick 6–8 per run (with entrance/core/endpiece pools). (Fabric Wiki)

4. **Rewards & CTM Monument**

   - Beating the end room gives the **Color Wool Block** for the Monument (CTM staple). Collecting every 4 colors grants a **bonus token** (Gold → Emerald → Diamond → Netherite). Tokens double as **catalysts** in recipes and can be **recipe remainders** (not consumed) using the platform’s remainder/container system. (superhostile.fandom.com)

5. **Rerun rules & portal sealing**

   - After you finish or timeout, that color’s portal **closes** server-wide for a set cooldown (e.g., 2 real-time hours) and reopens with a new layout next time. Vault-style “no dawdling” tension is preserved. (vault-hunters.fandom.com)

division of work (initial 2-week sprint)

You

- Pick game version (suggest **1.21** Fabric first, then NeoForge) and repo license (MIT).
- Define visual direction per color (biome palette, light level, puzzle vibe).

- Sketch Monument location & footprint (e.g., 7×7, 5×3 grid).

Me (GPT-5 Thinking)

- Generate the **multiloader repo** scaffold (common/fabric/neoforge), gradle wired.
- Implement **Custom Portal** for wool frames (Fabric) with datapack tags + activation item.
- Add **scoreboard+bossbar timer** helper utilities & sealing logic (extract → set cooldown → message). (Fabric Wiki)

Gemini

- Build **jigsaw structure sets** for 1–2 colors (say, Lime & Magenta): entrance/core/turn/end pools + processors, test in a flat single-biome dimension.
- Author 10 rooms per color (Nudge: keep connectors at consistent Y and use marker blocks for lock/key). (Fabric Wiki)

implementation notes (concrete)

- **Portal**
    - Tag all wools: `data/woolydungeons/tags/blocks/wool_portal_frames.json`.
    - Fabric side: register a **portal** that accepts any blockstate in that tag and encodes the **Color** property to pick the target dimension (e.g., `woolydungeons:lime_realm`). Use Kyrptonaught's API pattern for frame detection + ignition hook. (Fabric Wiki)
- **Dimension & worldgen**
    - For each color, define a **single-biome dimension** with fog/sky tint, then register a **structure_set** listing the color's dungeon start with low spacing (one per 512×512). Jigsaw pools live under `data/woolydungeons/structures/...`. (Fabric Wiki)
- **Timer**

  - Create `objective wool_timer dummy`. Tick function decrements per player **in that dimension**; mirror on a **bossbar**. On zero: teleport home, set a `cooldown` scoreboard for that color, and swap the portal block to a “Sealed Wool Gate.” (Bossbar/scoreboard patterns are well-trodden.) (Reddit)

- **Monument & tokens**

  - Monument = block entity that validates placed **Color Wool Block** items (there are 16). After every 4, emit **Bonus Token**. Use **recipe remainder** (`recipeRemainder()` on Fabric or Forge `getRemainingItems`) so tokens act as catalysts, not consumed. (Fabric Wiki)

content & balance guide

- Start with **4 colors** (Lime, Magenta, Cyan, Orange) for MVP; push to 16 later (Minecraft has exactly **sixteen wool colors**). (Minecraft Wiki)

- Room archetypes per color:

  - **Puzzle** (levers, redstone paths, light sensors), **Mob trial**, **Parkour**, **Key/Lock** (collect “spool” items), **Boss** (mini-arena).

- Expected run time: 15–25 minutes per dungeon in MVP (recognizable from Vaults), with your optional **3-hour “hardcore expedition”** variant for servers that want long sessions. (vault-hunters.fandom.com)

milestones

1. **M0 (day 2):** Repo boots + one working wool portal to a dev dimension; bossbar timer counting down in that dimension. (Fabric Wiki)

2. **M1 (day 5):** Lime realm jigsaw dungeon with 8 rooms, end chest drops **Lime Wool Block**, portal seals on timeout/finish. (Fabric Wiki)

3. **M2 (day 9):** Monument block entity tracks progress; grants **Gold Token** at 4 colors (remainder recipe working). (Fabric Wiki)

4. **M3 (day 14):** 4 colors playable; configs for timers/cooldowns; alpha playtest.

nice-to-haves after MVP

- Variant objectives (scavenger rooms, monoliths, mini-boss) — familiar to Vault Hunters players. (vault-hunters.fandom.com)
- Mapmaker hooks (datapack-only room packs).
- Cosmetic reliquaries for completing a color set.

---

if you want, I can spit out the **multiloader repo skeleton**, the Fabric **wool portal** registration, and a **starter jigsaw pool** (Lime) next. just say the word and I'll generate the code + data pack scaffolding so you and Gemini can drop rooms in and run.

this is from another friend who will be acting as project manager with a chatgpt assistant as well, here is their generated mvp. help me get this set up with eclipse ide for enterprise edition eclipse-jee-2025-09-R-macosx-cocoa-aarch64. can you generate the project scaffold for a basic git repo so I can clone it and get started? you tell me how to get started with the generated starter code and whatever resources I will need to get besides eclipse

---

ChatGPT said:

perfect — Forge-only it is. here's the MVP story map and then Story #1 with exact, do-this-next tasks. i'm keeping the first feature ultra-simple: a teleport item that sends you to a stub Lime dimension (defined via datapack JSON) and back. portals & jigsaw dungeons come after.
MVP stories (Forge)

  Forge workspace & skeleton

    Set up MDK, run client/server, hello-world mod id compiles.
    docs.minecraftforge.net

  Stub Lime dimension (datapack JSON)

    Add a single-biome test dimension via resources datapack.

Minecraft Wiki+1

Teleport item (first feature)

Right-click to jump to Lime (or back to Overworld). Uses ServerPlayer teleport API.
docs.minecraftforge.net

Run timer HUD

Bossbar/scoreboard countdown visible while in Lime; on timeout, auto-extract.
Minecraft Wiki+1

Simple jigsaw dungeon

1 entrance → 1–2 rooms → endpiece via jigsaw pools (datapack).
Gist

Reward drop + basic Monument block

Clear end → grant Lime Wool block; Monument tracks progress.

Portal block & frame detection (upgrade)

Replace “teleport item” flow with a Lime wool frame + ignition to open a portal.

Story #1 — “Hello Lime”: Forge project + Lime dimension + teleport item

Goal: build runs in dev, a /give-able item teleports you to a lime-tinted test dimension and back.
Tasks (exact steps)

Generate Forge mod project

Use the official “Getting Started with Forge” guide for your target (e.g., 1.20.1/1.20.6).

Import Gradle project, set modid = woolydungeons.

Verify runClient starts and the mod shows in Mods List.
docs.minecraftforge.net

Wire registries

Create a DeferredRegister<Item> and DeferredRegister<CreativeModeTab> for woolydungeons.

Add a creative tab “Wooly Dungeons (Dev)”.

Add the teleport item

Register LIME_KEYSTONE (rarity UNCOMMON).

In use()/useOn(), server-side only:

If player.level().dimension() != LIME_REALM_KEY, get ServerLevel dest = server.getLevel(LIME_REALM_KEY) and call ((ServerPlayer)player).teleportTo(dest, x, y, z, yaw, pitch).

Else, teleport to Overworld spawn (server.overworld() and similar call).

Keep a safe target (e.g., BlockPos(0, 120, 0) + descend to first solid block) to avoid void falls.

(API surface lives in Forge/Mojang server classes you already have after MDK setup.)
docs.minecraftforge.net

Define the Lime dimension (datapack JSON)

Under src/main/resources/data/woolydungeons/dimension/, add lime_realm.json.

Make it a simple single-biome world (flat/noise), with custom fog/sky tint if you want.

If needed, also define dimension_type/ and a basic biome/ json.

This leverages vanilla data packs for worldgen in Forge.
Minecraft Wiki+1

Resource keys & constants

Create ResourceKey<Level> LIME_REALM_KEY = ResourceKey.create(Registries.DIMENSION, new ResourceLocation("woolydungeons","lime_realm"));

Use this key in the teleport item.

Dev test

/give @p woolydungeons:lime_keystone

Right-click in Overworld → arrive in Lime realm at safe Y.

Use again → return to Overworld spawn.

Confirm no crashes on dimension load.

(Optional) tiny HUD proof

Create a bossbar named woolydungeons:run when entering Lime; remove on exit. (Real countdown comes in Story #2.)
Minecraft Wiki

Docs & checklist

Add DEVELOPING.md: run configs, how to add another dimension via JSON, how the teleport item routes you between dimensions.

Why this order?

Forge workspace first (unblocks everything).
docs.minecraftforge.net

Datapack-defined dimension next (modern MC expects worldgen via data packs).
Minecraft Wiki

Teleport item before portals to prove the core loop ("go to color realm, come back") with minimal code. Portals, frames, and sealing come later.

want me to drop in ready-to-paste snippets next (registry boilerplate, a minimal lime_realm.json, and the use() body for the keystone)?