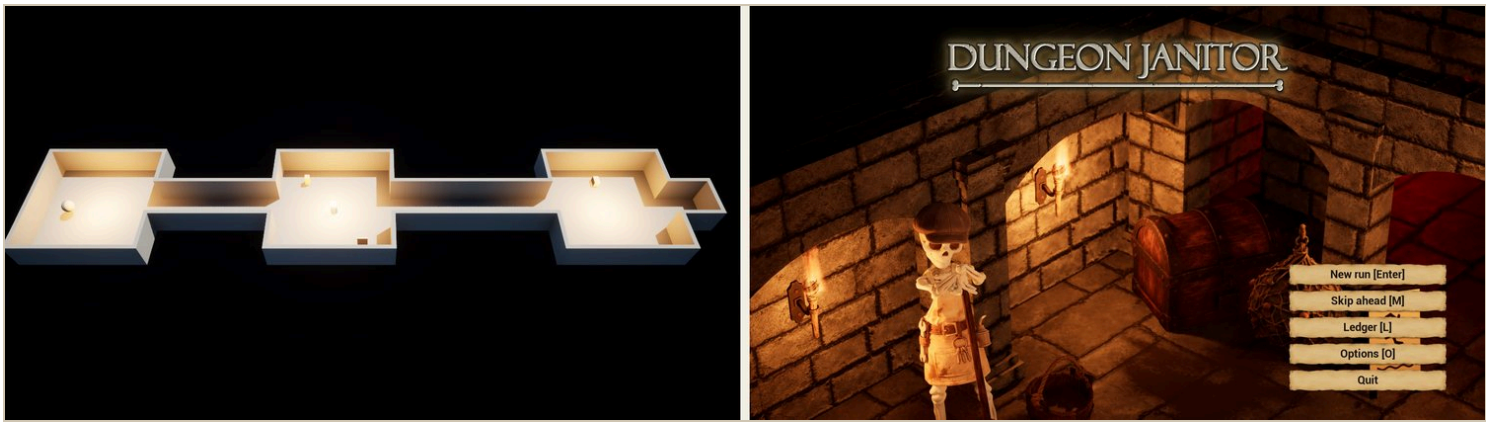
Dungeon Janitor: how three people directed agents to build it, and the 28 things that broke

Glass Box dev log for the 2026 Epic AI MegaJam, Unreal Engine 5.9 track, from TeamDADS: SirPick (team owner), Tallboy, Mignightxrush. Gameplay video, 31 seconds.



Left: the dungeon in the first hours of Nov 12. Right: the title screen at midday on Nov 15; its stone lettering was replaced that afternoon (section 4, item 8)

In short

- **The game.** Dungeon Janitor: a skeleton janitor keeps a wizard's dungeon running while adventurer parties raid it. Unreal Engine 5.9, C++, Windows.
- **Who made it.** Three people. Nobody typed the game's code by hand: each person directed agents (Claude Code and Codex CLI), and the agents wrote, built, tested and captured.
- **How it was directed.** One orchestrator session per person plans and checks evidence. Builder agents and Codex write code from written briefs (147 kept). The rule: a change lands only after a build, a scripted run of the real game (433 scenario files) and a look at the frames. We broke that rule more than once, and section 4 says where.
- **What it cost.** 44 packaged builds and about 460 commits in four days, one whole game cut on day two, and the 28 failures in section 4. Nine of them had passed every check we had and were caught only when a person looked or listened (items 2, 5, 6, 8, 9, 10, 21, 22 and 28).
- **One of the 28.** An agent making voice lines changed how it typed to go faster, and about 70 recordings in a row said "Tastes like feet. Works, though." Its own check passed every one. A person heard it.
- **What we would tell another team.** A check tests what you thought to ask, so put a person in front of real frames early. Name the engine feature before writing your own. An agent that stops to ask while you sleep has cost you the night.

1. The game and the theme

You are a skeleton janitor who tried to rob a wizard, was caught, and was cursed to keep the wizard's dungeon clean until you have 100 Souls. Adventurer parties raid it every round to break its Heart. Between rounds you place traps and monster dens. During a raid you run the corridors yourself: re-arm spent traps by hand, mop the floor so raiders slip, rattle to pull them off the Heart, sweep up the remains. You win a run at 100 Souls and lose when the Heart breaks.

The feel SirPick asked for: dungeon management, the defences you set before each wave in a lane-defence game, a tabletop role-playing party kicking in the door, and a dash of the small nuisance who ruins bigger people's plans.

The theme was "Director's Cut". We took it the way the kickoff stream put it: build what you never had the time or the crew for, and be its director. Each of us directed a crew of agents, and the work was deciding what goes in and what is cut. So the entry is a game three people could not have hand-built in four days: two phases of play, raiders who weigh risk, twelve kinds of Mimic, a new dungeon every run. We cut as well: a whole second game, Swift, came off `main` on day two, and the Saturday feature freeze went on day three.

Our first answer had been literal: a dungeon "cut from a game", with "Director's Notes" on parchment and an ending card that marked each note KEPT or CUT. On Nov 12 SirPick played that build and asked whether it did anything that could not have been done before. The honest answer was no: the theme sat in the dressing, not the mechanics. That premise stayed in the text until the last day, when he replaced it with the story above. Inside the game the theme is lighter now than in that first version.

2. How the team worked

Three people, each directing their own agents on their own machine. Version control is Diversion, not git. It syncs without push or pull and hard-locks binary Unreal assets to whoever saved them. Each person, and each long-running agent session, had its own branch and workspace (a separate checkout). `main` is the entry. One rule file, `AGENTS.md`, is read by every agent.

WHO	WHAT THEY DID	AGENTS AND TOOLS, FROM THE LOGS
SirPick	Team owner. The bulk of the dungeon game, from the first hour: the level, the traps, the monsters and Mimics, the janitor, the game's logic and interface, and fitting everyone's work together. The generated 3D art for creatures, props and traps, and the text-to-animation clip set. Every build and package.	Claude Code (a Fable 5.2 architect with Opus 5.5 workstream agents on night one, Fable 5.2 builders from Nov 12, Opus 5.5 for everything once the Fable allowance ran out); Codex CLI for code and every generated image; Meshy; Tripo Studio; the Kimodo plugin; Blender; Python for sound.

WHO	WHAT THEY DID	AGENTS AND TOOLS, FROM THE LOGS
Mignightxrush	The raiders' AI mind (their decision code) and the procedural dungeon generator. Also the Living Dungeon slice on day one, whose spawner and Soul ideas went into the dungeon, the first five hero models, the Spike Pit and Net Trap, room dressing, and the raiders' thoughts and voices.	Claude Code: one scheduled run with nobody at the keyboard, then live sessions on his own branch, with separate reviewer agents for frames; Tripo Studio and MiniMax Audio driven by Claude in Chrome.
Tallboy	The 39 adventurer models and 58 weapon meshes (Nov 14 to 15), the class attack clips and the janitor's mop clips, attack and torch effects, blood stains, the Options screen, the title badge and key art, the trailer, and the NVIDIA DLSS work with a packaged frame-rate check. Before that: Swift, a flying prototype (Nov 12 to 13), and the move of `main` to the dungeon.	Claude Code on Fable 5.2 with forked sub-agents and separate reviewer agents; Tripo Studio; ChatGPT image generation; Blender; for Swift, Sorceress (an AI asset service).

3. How the agents were directed

Three people, three ways. SirPick ran an architect session with builder agents under it (below). Tallboy ran one main session that forked sub-agents and kept separate reviewer agents who saw only captured frames; where the reviewers and his own eye disagreed, his eye won (section 4, item 6). Mignightxrush began with a run nobody attended and then worked in live sessions on his own branch.

Architect and builders. On SirPick's side one session is the architect (later called the orchestrator). It plans, checks evidence, commits and merges, and writes no game code. Under it sit builder agents, one per workstream: traps and the janitor, raider AI, core and UI, art and audio, later creatures.

Who wrote the code changed twice. On night one, Claude Opus agents, one per workstream, wrote briefs (the written task for a coding tool) and Codex CLI wrote all the code from them. The 147 brief files are in the repository. From mid-morning on Nov 12, at SirPick's request, Claude Fable agents wrote code directly, to compare: fewer round trips, and reports that listed their own failures. On the evening of Nov 13 SirPick ran out of Fable usage. Each builder wrote a handoff with "five things a successor would get wrong", and Opus agents took over, with Codex writing the larger changes.

Scenarios as evidence. Nobody sat at the editor on night one. A scenario is a JSON file that drives the real game: place this, wait, press that, check these asserts, save these screenshots. The runner is a small module inside the game, switched on by a command-line flag and inert otherwise. It carries out each step in the running game and writes a result file and PNG frames. The same file runs in the editor build, with no rendering at all, or inside a packaged exe. A change lands only after a build, a scenario run and a look at the frames. That is the rule; items 4 and 8 in section 4 are times we did not keep it.

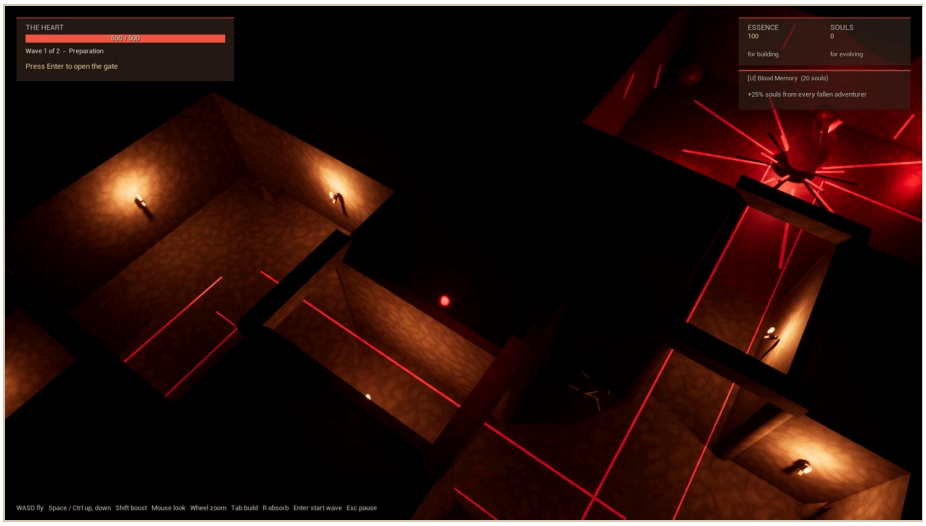
Lanes. Several agents shared one PC and one C++ module. A lane (first called a turn) is a small text file that says which agent may run Unreal right now, with a queue file beside it. By Nov 13 there were three: editor, headless (a test run with no rendering) and GPU (anything that renders, packaging included). Agents queue first come, first

served, a turn is capped at 20 minutes, and lanes are taken in one fixed order. Code waits in a staging folder until its agent's build turn, so a half-written file cannot break everyone's compile.

The builder does not grade its own work. Anything visual is reviewed by a separate agent or a teammate, from real screenshots. On Swift, reviewer agents scored captured frames out of 10, round after round. Unreviewed frames are marked "builder's own frames" in the log.

Spending, removals and posting stay human. Agents spend credits only against an approved list, and never delete, submit or post.

One run with nobody there. Mignightxrush's day-one slice was built by a scheduled Claude Code task while he slept. Its prompt was written on Nov 11, is declared in `DECLARATIONS.md` and is in the repository word for word (`docs/living-dungeon/RUN_PROMPT.md`). Its first rule is a time gate: if the clock is before midnight Eastern on Nov 12, "create NOTHING (no branch, no files, no assets) and stop". Its second is the rule SirPick's side had to learn the hard way on two nights (section 4, item 1): "Nobody is watching, so never block on a question: make the smallest safe choice, record it, and keep going." The run went from 12:01 AM to about 1:15 AM Eastern: seven compiles, four content rebuilds, about a dozen self-test runs and two passes by a reviewer agent, which listed 13 problems and then 7. It ended with a playable slice and 57 of 57 checks passing.



What the unattended run had made by 1:15 AM on Nov 12: the Living Dungeon slice. It was not picked as the entry; its spawner and Soul ideas went into the dungeon

How SirPick's side was wired:

```
SirPick (person)      plays each package, sends notes, approves spending and merges
  |
orchestrator session  plans, writes briefs, runs builds and scenarios, reads frames,
  |                   commits and merges; writes no game code
  |-- builder agents      one per workstream: janitor and traps, raider AI,
  |                       core and UI, art and audio, creatures
  |-- Codex CLI           code from written briefs; every generated image
  |-- reviewer agents     see only frames and captures, never the builder's notes
  `-- lanes               editor, headless, GPU: one agent runs Unreal at a time
```

The editor lane's two files on the last afternoon, with three sessions waiting behind the orchestrator (ARCH):

```
.editor-turn         ARCH | 2026-11-15T13:32:54 | C:\new\DADS\MegaJam-SP-Dungeon
.editor-turn.queue   TRAILER | 2026-11-15T13:33:53
                     CUTS | 2026-11-15T13:33:53
                     LOOK | 2026-11-15T13:33:55
```

A brief, as it was sent to Codex on Nov 12 (the whole file, `devlog/sirpick-dungeon/briefs/w1_l2c_ring_color.md`, less its heading):

```
Tiny change. Work only in the file listed. Touch no file outside this list. Do not build,
do not launch Unreal, do not run `dv` or `git`.

Engine feature: the existing UMaterialInstanceDynamic parameters on the trap ring
(no new systems).

Why: rendered in game (devlog/2026-11-12_SirPick_L2_traps_spent.png, intensity 2), the
spent ring (1.0, 0.35, 0.0) reads yellow under tonemapping and bloom, close to the
torch-lit floor. At intensity 4 both colours bloom toward white; at 1 they are clean but
a little dim. Spent must read clearly orange, armed clearly cyan.

File you edit (and nothing else): staging/W1/Source/Traps/SirPickTrapBase.h

Change
- SpentColor default: FLinearColor(1.0f, 0.16f, 0.0f).
- RingIntensity default: 1.5f.
Nothing else. Do not touch ArmedColor, the arc colour, the ink rim, or any logic.

Done means: the file differs from before only in those two default values.
Final message: the two lines before and after.
```

Most briefs are longer, but they have the same parts: the files it may touch, the engine feature to use, why (with the frame that showed the problem), the change, and what "done" means.

A scenario, the kind of file that stands in for a person at the keyboard. This one checked the title screen and the janitor's end-of-round remark on Nov 15:

```
{ "op": "settle", "min_seconds": 6, "min_frames": 60 },
{ "op": "console", "command": "spd.phase Title" },
{ "op": "wait", "seconds": 5.0 },
{ "op": "screenshot", "name": "v25_1_title_live", "show_ui": true },
{ "op": "console", "command": "spd.title.start", "label": "a new run on a generated dungeon" },
{ "op": "wait", "seconds": 14 },
{ "op": "assert", "check": "phase", "equals": "Build", "label": "the run began" },
{ "op": "console", "command": "spd.raid.start", "label": "a round with nothing built" },
{ "op": "wait_until", "check": "phase", "equals": "Intermission", "timeout": 200 },
{ "op": "screenshot", "name": "v25_4_report", "show_ui": true },
{ "op": "console", "command": "spd.card.confirm" },
{ "op": "wait", "seconds": 3.0 },
{ "op": "screenshot", "name": "v25_5_round_end_thought", "show_ui": true }
```

And what a run hands back: the result file of the janitor's own scenario on Nov 15, shortened.

```
{ "scenario": "w1_player", "result": "pass", "duration_s": 81.77,
  "asserts_passed": 46, "asserts_failed": 0,
```

```
  "steps": [ { "i": 0, "op": "settle", "label": "settle shaders", "ok": true,
               "detail": "seconds=6.000 frames=3402 shader_jobs=0" },
             ... 197 more steps ] }
```

A model prompt, from Mignightxrush's Spike Pit on Nov 14 (Tripo text-to-3D; the full list is in `ASSETS.md`): "a square pitfall trap in a worn stone dungeon floor with sharp upright wooden stakes and iron spikes in the pit, hand-painted fantasy game prop, one single object, no character, no text". It came out as a ring of stakes, which became the spikes.

---

4. What broke, and what we learned

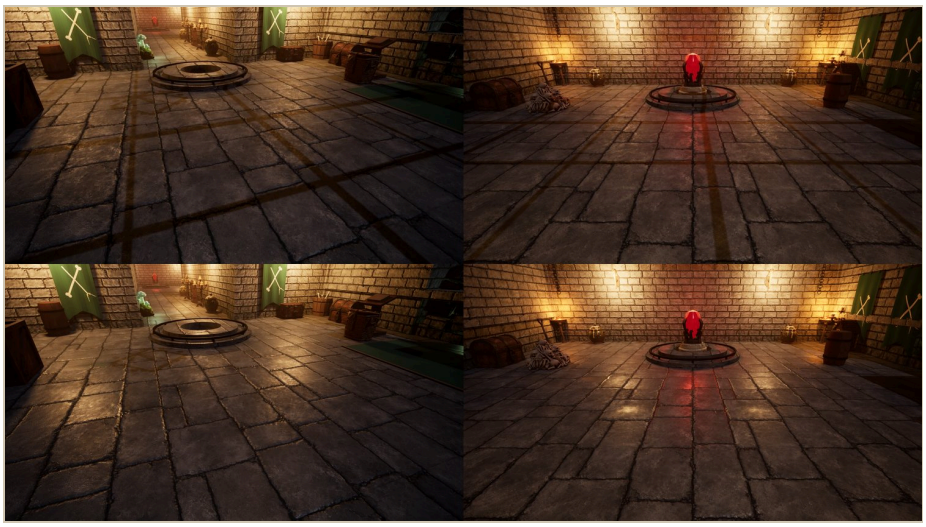
Twenty-eight failures: eleven in full, sixteen in a table, and one from the last evening's playtest.

1. The agent stopped to ask while SirPick slept, twice (nights of Nov 12 to 13 and Nov 14 to 15). On the first night the architect stopped the overnight work at 2:10 AM and waited for answers. About three and a half hours of machine time were lost. The same night it spent 330 Tripo credits remaking six creatures the game already had, when SirPick had asked for variants. Rule since: no spending without a named list he approved. The rule about questions was written down too, and it still happened again. On the last night SirPick asked what the agent could work on overnight. It answered with a list of seven items, three decisions and "reply go", then did nothing for six hours. His words at 7 AM: "You asked a question while I was asleep. We lost that time." Rule since: a late question like that is the go-ahead. Start with your own defaults and report what you chose.

2. Every raider glided (Nov 13). A package had passed its smoke test 16 of 16, and SirPick saw every raider slide along the floor without walking. The template animation blueprint plays the walk only under acceleration, and AI path following sets velocity directly. One engine flag: 0 of 20 walking before, 20 of 20 after. A check tests what you thought to ask.

3. A game that could not be lost (Nov 13). In three scripted 30-round runs every raider died in a Pit and the Heart was never hit. The one "loss" was the test janitor falling out of the world. Then a headless run said a slip cost a raider almost nothing, while a rendered run measured 3.9 to 4.5 seconds lost. Every balance run had been headless.

4. "Fixed", said without looking (Nov 13). The architect told SirPick a misaligned placement ghost was the lid's pivot (measured: it was not) and that ranged raiders never hit the Heart (they do). Brown floor bands were blamed on a decal, then on the build grid. They were shadows of the ceiling beams. Rule: never call something fixed on a log line you have not read or a frame you have not opened.



Top row: the brown bands, Nov 13. Bottom row: after

5. Raiders who used the wrong motion (Nov 15). The new adventurers went in with their weapons in their hands and nobody checked what the hands did. Every class attacked with one shared clip, an overhead blow. So the archer, bow in hand, looked as if she threw something, and what flew was a plain sphere; the wizard chopped the air to cast; the rogue chopped with a dagger; the cleric healed with an attack swing. Tallboy saw the archer in a playtest. Each class now has a clip posed from its own frames (a bow draw and loose, a cast, a stab, a heal) and the archer shoots an arrow. The cleric still swung an empty hand, because her mace is worn at her hip; an hour before the package Tallboy had her call down a shaft of gold light instead, and left her rules alone. Still wrong when this was written: the wizard casts with his staff on his back, and the warrior's potion clip raised his shield over his face and was taken out. Lesson: a model holding the right weapon tells you nothing about the motion. Look at each class doing each thing.

6. Two reviewers said yes and Tallboy said no (Nov 15). The first mop fix placed the mop from the janitor's hands in code. Two independent reviewer agents passed it for the jam build from stills and frame sequences. Tallboy watched the same thing as a clip and rejected it in one line: the broom still whipped about in the swing. Reviewers reading frames miss what a person sees in motion. He asked whether Kimodo (the text-to-animation plugin) should regenerate the clips with a fist; it cannot (its skeleton has no finger joints), so the clips were edited instead, and that version he accepted, apart from one flick at the sweep's loop seam that he also caught by eye.

7. A puddle nobody had ever seen (Nov 15). Asked to make the mop puddle look like real water, the agent built new materials, captured them, and saw nothing. The old puddle showed nothing either: its decal box was 6 cm deep and did not reach the floor, so the puddle had been invisible in the game all along while its checks, which test slipping and timing, passed. The same mistake had been made that morning with the blood decals (3 cm).

8. The title that failed twice (Nov 15). The stone title was approved as a picture and then came out wrong in the game. SirPick: "Dog, that title is a fail", and after the first fix, "Still a fail". The agent had sized the widget from the texture's own size, and a texture imported in the same run reports 32 by 32 until it has finished loading. The size is now written in the code. Both wrong versions had passed the build and the scripted checks; neither check looked at the frame. On the last afternoon Tallboy replaced the stone lettering with a badge he made himself (section 5).

9. A faster way to type that typed nothing, and 1,900 credits (Nov 15). The raider voices were made one line at a time in MiniMax's web page, the agent typing each line into the box through Claude in Chrome. Halfway through, to go faster, it swapped the typing for a script that set the box's text directly. The box showed each new line, but the page kept the old one, so about 70 generations in a row all said "Tastes like feet. Works, though." Each one cost credits, and each file was a little different, so the agent's check (count the downloads, compare the files) passed every time. Mignightxrush caught it by ear: "you just wasted so many credits". About 1,900 free credits were gone. The bad clips were deleted, the page reloaded and typing put back, and from then on every line was checked in the site's own History, text and voice, before it was downloaded. Lesson: on a site that spends credits, keep the method that worked; if you change it, prove one output where the site records it before running more.

10. Raiders who could not hit the Heart, from a pulse (Nov 15). In his 4 PM playtest Tallboy saw raiders reach the Heart, never land a blow and run. The Heart pulses every 1.5 seconds and hurts whoever stands at it, and every pulse played the raider's 1.5-second flinch, which cancels his swing. So a raider at the Heart flinched forever. The damage stayed as it was; he now flinches only at the 1st, 4th, 9th and 16th pulse and shrugs off the ones between. It was found by playing, not by a check.

11. Sixteen weapon looks, and a reviewer who said no (Nov 15). Each bow and staff fires its own look (frost, leaf, ember, bone and so on). A capture of all sixteen went to an independent reviewer, who said no: only the frost bow read; five bows were a few specks and the steel bow showed nothing; two staffs fired white bolts with the wrong light, because the light and the cast motes still used the wizard's class colour; the bone staff was a row of flat discs. All were tuned in one pass (bigger impacts with a tinted flash, a streak in the bow's colour, the weapon's colour on the light), but with the team's 3 PM package target close it was not sent back for a second review, and the log says so.

The other sixteen:

WHAT HAPPENED	WHAT WE CHANGED OR LEARNED
12. Packages that looked fine and were not (Nov 12, 13). A trial package built and zipped without an error and held none of our content: the config never cooked the dungeon's folder. Later packages crashed at map load on a stale cook.	The package script handles both and runs a short scenario inside the packaged exe before it calls a build good.
13. Raiders who could not reach the Heart (Nov 12, 14). The Heart's collision cut a hole in the navigation mesh, and our own "move" and "hit" range checks disagreed and flipped thousands of times in one tick.	After three rounds of patching our own distance checks, raiders and creatures moved to the engine's own reach test. Name the engine feature first; after three rounds on one thing, stop and look at the whole.
14. Two of SirPick's words, misread (Nov 13). "Threat level" was built as the round number; he meant a score per raider. "Trap damage on or off" became a table of balance runs; he meant re-arming traps mid-round.	Restate the person's definition before designing from it.
15. Lanes that jammed (Nov 12, 13, 15). Three agents each lost 40 to 65 minutes polling for the one build lane. Later, agents took lanes in opposite orders and deadlocked, twice. On the last afternoon six sessions queued for one lane, and a merge took 25 minutes, most of it waiting.	A queue, then one enforced order for taking lanes. The fix itself broke the package script once.

WHAT HAPPENED	WHAT WE CHANGED OR LEARNED
16. Version control surprises (Nov 13). A teammate's commit arrived mid-build. A commit-all tracked a generated build folder and broke other workspaces for half an hour. A merge to `main` was refused: 1,933 assets were locked to the teammate who had saved them.	Commit named files, not everything. A locked asset reaches `main` only through its lock holder.
17. The grey floor that reached main (Nov 14). Mignightxrush's script for the Spike Pit's floor-hole material ran twice and the floor stopped compiling on any fresh load. The builder's own shader cache hid it, and package v9 had untextured flagstones. The builder's first answer, "shaders still compiling", was wrong.	A material that compiles in the session that built it has not been tested. Check it from a clean cache.
18. Two package near-misses (Nov 14). A package nearly cooked over a build SirPick already had. Another died at launch with no log: the dungeon generator read the command line before the engine had set it, which is harmless in the editor and fatal in a packaged build.	The script refuses a folder that exists. A debugger on the packaged exe found the second; no scenario could have.
19. Merged to main before anyone played it (Nov 14). The changes passed their builds and scenarios and the agent merged at once. SirPick: "you merged it to main already? thats bold, i figured it would have been built from mrpickering first to test" (`mrpickering` is his test branch). Nothing broke that time.	Package from the branch, let a person play it, merge to `main` only on his word.
20. A new trap that broke an old one, and a fix that teleported raiders (Nov 14). Importing the Tripo tripwire for the Spike Pit overwrote the asset the Arrow Wall already used for its wire. The same day a fix let the pit reach 35 cm into the next tile, and Mignightxrush wrote back: "the AI was on another tile and teleported to the other trap".	The pit's wire became its own asset and the original was put back. The reach was taken out: each trap now counts only a raider whose body is over it.
21. Copied from the hand-made level without asking why (Nov 14). The generator took the hand-made level's rules for walls. On a generated map they left posts poking into every doorway, and low walls that looked broken in first person: "not the same height". The torches had no flames at all: the loader looked in the wrong folder and quietly used a fallback that drew grey sparks.	Each was found by Mignightxrush from a frame, not by a check.
22. Torch flames, three tries (Nov 15). Several needles of flame, then a wider flame that a reviewer agent passed and Tallboy called no good, then a new shader with smoke and a moving light.	The cause was found only from a close-up: the flame was additive and could not be seen against the wall its own light had washed out.
23. Two people in the same files (Nov 15). Tallboy's agent and SirPick's both changed the UI and the janitor; one pull needed four files repaired and one commit was refused seconds after the other landed. It happened a third time at 3:12 PM ET over the dev log itself.	Save a patch of every changed file before each pull. That made the repairs ten-minute jobs.
24. A key already taken, and a package killed by its own orchestrator (Nov 15). "Hide the controls" went on F1, which is the engine's wireframe view in a Development build. The same afternoon the orchestrator started a second job while its own package was cooking, and the first try at package v22 was lost.	The key moved to J. One Unreal job at a time, even when both are yours.

WHAT HAPPENED	WHAT WE CHANGED OR LEARNED
25. Approval that was not approval (Nov 15). SirPick wrote "The main menu looks good to start". The agent read it as finished and stopped work on the title scene.	He meant a good start: "No, I meant keep going."
26. Dressing that drew nothing, and lights that shone through a closed wing (Nov 15). The first cobwebs were thinner than a pixel from the game camera and every one was dropped. The grime stains were invisible for three tries. Then the reviewer agent sent the dressing back: closed wings, which must be pure black, glowed warm.	Hiding a light actor does not switch its light off, and the code had placed each light by its bounds, which are its whole falloff sphere. A rug the reviewer called a placeholder was taken out, and the chains now hang from hooks.
27. The first speech bubbles failed review (Nov 15). Drawn in the 3D world, they went under the HUD and off the screen, nothing said who was talking, lines replaced each other in a third of a second, and four lines echoed catchphrases from other works.	They moved to a screen layer with the speaker's name and a minimum time on screen, and the four lines were rewritten. Five review rounds in all.

And one from the last evening:

28. A boss under the floor (Nov 15). In the evening playtest a boss survived the Spike Pit, Mignightxrush re-armed it, and the boss fell through the map. The pit had been built for raiders who die in it: it held whoever fell in on the spikes every frame until they were gone. A survivor was never let go, and re-arming shut the floor over him. Every Spike Pit test had passed, because every test raider died in it. Fixed (survivors climb out; a re-arm puts anyone left down there out first) and caught by a new test with two boss Warriors. The same playtest found raiders searching a chest a teammate had already searched: the shared "searched" list was only read by raiders exploring, not by one who saw loot glinting.

Also logged: Codex once added its own dev log entry, outside its brief.

---

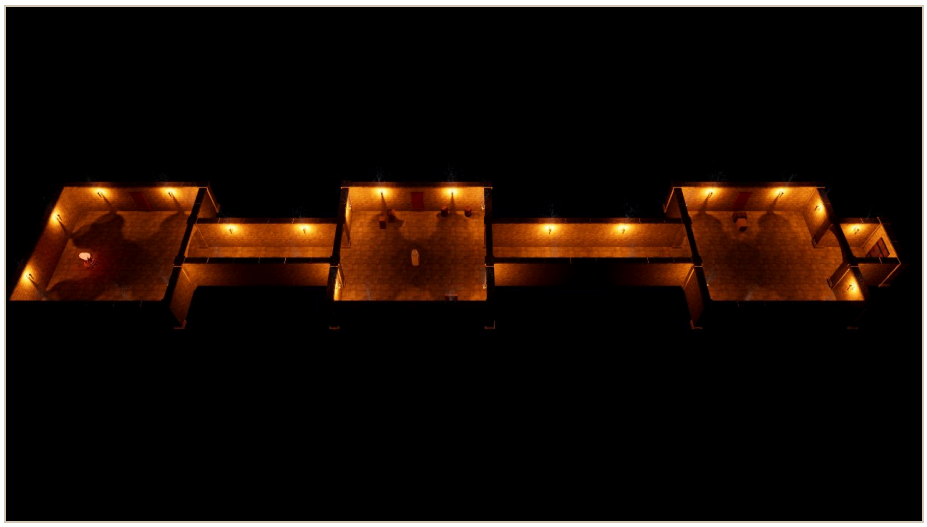
5. Timeline

Nov 11, before the jam. Joined the jam, installed UE 5.9, set up Unreal MCP, wrote `AGENTS.md`, moved to Diversion. No game content. Pre-jam tools and prompts are in `DECLARATIONS.md`.

Nov 12: three slices to compare

The jam opened at midnight Eastern (ET). We built three thin games to compare: Tallboy's Swift (fly a swift over a village at dusk and never land), SirPick's "Cut Content" dungeon, and Mignightxrush's Living Dungeon, built by an unattended scheduled agent run.

- Dungeon: foundation at 1:10 AM ET, every must-have by 4:20, two packages by 8:17.
- Swift: its reviewer agent scored the look 3 of 10 in round one and 6 by morning.
- About 10:30 AM ET SirPick plays the dungeon ("pretty fun") and sets version 2: raiders die and their remains stay, monster spawners, Souls kept between runs, more traps with upgrades, the Mimic.
- Afternoon: Mignightxrush's hero models go in. The first import stood in a T-pose and shrank to nothing in every clip, until a Blender re-export fixed the rig's scale.



First lighting pass, Nov 12: one amber hue, sent back because it would hide the orange "trap spent" ring

Nov 13: one game is cut

- Overnight: the Fallen (raise a dead raider's spirit as a guard), a plan view, a first-person view, Tripo creature models.
- Morning: renamed Dungeon Janitor. A run becomes about 30 rounds, won at 100 Souls.
- Text-to-animation works with UE 5.9's Kimodo plugin, run locally: 13 clips, then 91 retargeted onto our characters. Hours were lost first, because the architect treated a plugin already set up on this PC as new.
- Afternoon: the team picks the dungeon as the entry. Swift comes off `main` in an ordinary commit and stays whole on branch `swift`, with its own log in `devlog/swift/`. No history was removed.
- The hero mind (the raiders' decision code, started by Mignightxrush) merges in from a second agent session: each raider gets a threat score, and the range of scores allowed rises each round.
- Evening: SirPick runs out of Fable usage, and asks that any empty prop raiders search can become a Mimic.



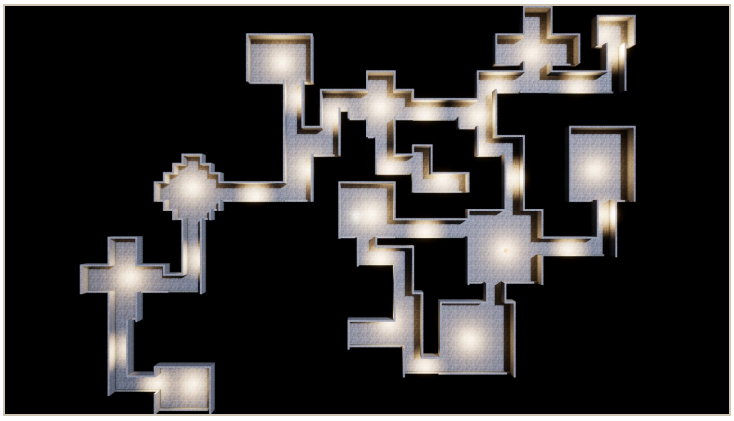
Mimic test sheet, Nov 13: seven of the kinds, each in a wide shot and at the bite

Nov 14: mimics and a generated dungeon

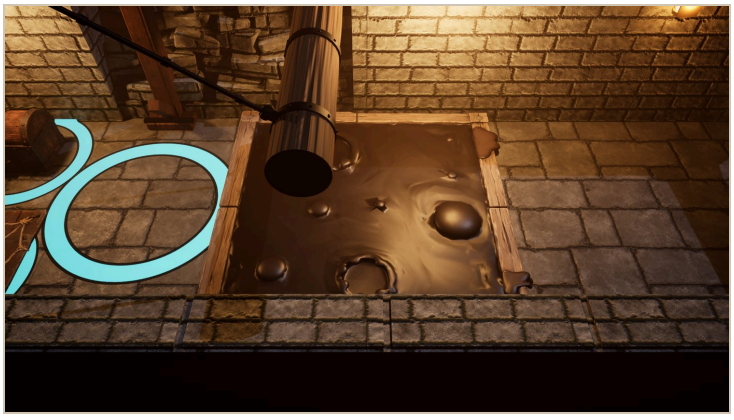
- The Mimic changes from a trap into a creature that bursts out of its prop, bites, chases and hops home to re-arm. Twelve kinds, all modelled.
- Mignightxrush, each step from a note he sent while he played. The build phase first: the doorway raiders come in by is kept clear, every trap and monster card states its damage, and an upgrade is picked from two paths instead of browsed.
- Two new traps, modelled in Tripo through Claude in Chrome (four models, about 180 credits): a Spike Pit whose trapdoors drop open into a real shaft with stakes at the bottom, and a Net Trap that snatches a raider into the air. The old Pit went.
- Then a procedural dungeon generator, written in one session (3,000 of 3,000 seeds valid): a new dungeon every run, wings closed by rubble that open as the rounds go on, the Heart moved deeper as they open, and three map sizes of about 16, 29 and 44 rooms. Rooms got furniture by theme from eight more Tripo models.
- About three in ten crates and vases hold a find. Raiders search or smash them, and the player can place crates, vases and chests as bait. On a generated map raiders go room by room, search what they pass, and cross a room off once it has no Heart.
- The Saturday night feature freeze in our schedule did not hold. SirPick moved the cutoff to Sunday afternoon.



The Mimic as a creature, out of its chest and after a raider, Nov 14



A generated dungeon from seed 7, before dressing, Nov 14



The first Tripo tar patch read as milk chocolate, Nov 14. It was darkened to wet black

Nov 15: deadline day

Tallboy:

- 39 new adventurers on the five existing class skeletons, so existing animation still plays, plus 58 weapon meshes. A raider who loots a container may come out with better gear.
- From his playtest lists: the janitor holds his mop, hits and deaths leave blood the mop wipes away, spells and arrows take their look from the weapon, the torches got a new flame, and an Options screen, a turning camera and a bone-hand cursor went in. Nine sub-agents each drafted one piece in a scratch folder without building, because his playtest held the game's binaries. The main session applied, built and captured each, and separate reviewer agents judged the frames.
- DLSS: NVIDIA has no plugin for UE 5.9, so the 5.8 one was compiled against 5.9 after a script added casts to 162 logging calls the newer compiler refused. His agent then packaged `main` and measured 116 frames a second at 1080p on an RTX 4070 laptop (108 in a busy fight). DLAA was about 1 ms a frame cheaper than the engine's own upscaler.
- The cleric, who had been swinging an empty hand, got her own smite: a cast and a thin shaft of gold light. Her rules were left alone that close to a package.



The new adventurers in the game, with the janitor in front, Nov 15

Late on the last afternoon Tallboy called the script-drawn stone lettering on the title cheap and made a badge himself in ChatGPT image generation, from two images: an earlier red enamel plaque of his, and a screenshot of our menu's parchment title as a type reference. It went on the title as it came, and from it he made a banner, key art and the game's icon. It is red, which the team had kept for the Heart alone; that was his call. The prompt, as he wrote it:

> Use the TWO images for two specific purposes. The standalone horizontal red enamel plaque is the base asset to edit. The screenshot of a dungeon menu with a tan parchment label is strictly a TYPOGRAPHY reference. Redesign ONLY the lettering on the standalone dark oxblood red enamel brass-rim badge. Copy the typographic feel of the supplied parchment label as closely as possible: actual old print-shop letterpress typography, chunky condensed antique slab serif with blunt, slightly bracketed blocks, a prominent curved descender-free hook on the J, substantial vintage slab feet, compact open counters, believable consistent type design, flat matte warm ivory stamping ink, a little broken ink and pinhole wear like the shown parchment title. JANITOR must be large and stout in the lower line, and DUNGEON smaller letterspaced in the upper line with the same family of strong antique

```
serifs. Set it very deliberately on two clean baselines, consistent kerning and a readable silhouette; distress only the ink coverage in subtle patches, not the basic letter shapes. No outline, bevel, extrusion, glow, gradients on the type, no modern sans. Correct spelling exactly "DUNGEON JANITOR", no extra text. The original sign is dark red enamel with brass rim and a small centered ivory skull in a newsboy cap at the crown. Keep these, keep the large clean string mop and stained wooden pail layered BEHIND the lettering, and keep the two subtle dark burgundy drips below the lower-left rim. Keep the color and layout of the red badge, NO parchment surface, no scene, and no key icon. Export one isolated wide transparent logo asset with full alpha around the shape and its drips, with safe margins, ready to use as an in-game menu PNG.
```



The title badge Tallboy made in ChatGPT image generation, Nov 15



Key art from the same badge, Nov 15

SirPick, working from his notes while he played each build (packages v22 to v44 in one day):

- The title screen became a small 3D set, the janitor leaning on his mop against a torch-lit wall.
- The janitor thinks out loud: 184 short lines for 23 moments (a trap goes off, the Heart is hit, a quiet round), on a paper scrap over his head. Codex wrote that module from a written brief.
- The shop moved to a stone slab with a picture for every item, hints became one-at-a-time lessons that go away when the player does the thing, and the round-end report was rebuilt after his two words on it: "looks awful".

Mignightxrush:

- For SirPick, a chart of the raiders' decision loop with how often each question is asked in a real round. It showed the potion question asked by every raider 43 to 119 times a second, even with no potion carried. It is now asked four times a second and only by a raider holding one.
- Generated rooms are dressed by theme (cobwebs, grime, bones in a crypt, and shackles and a slumped skeleton made in Tripo), crates hold supplies as well as treasure, and from round two raiders search room by room.
- From his own playtest: each round began with a stall of more than a second while the first raider of each class loaded Tallboy's new models. 149 assets now load in the background as the level starts.
- He asked for raiders who talk, "super funny but real at same time". They speak when something happens to them, and a thought bubble gives the reason behind a decision: "25% left. Nobody's paying me enough for this." Then real voices from MiniMax text-to-speech: 309 short clips recorded and 307 used, one library voice per class (section 4, item 9 is what that cost). That evening he found two more faults by playing (item 28).



A raider's thought over his name tag, with the round's HUD, Nov 15



A boss who survived the Spike Pit, hopping out over the far rim, Nov 15

---

6. Tools and models

Unreal Engine 5.9 with Unreal MCP and Geometry Script. Claude Code (Claude Fable 5.2, Claude Opus 5.5) for planning, briefs, verification and, from mid-morning on Nov 12, game code. Codex CLI (OpenAI, gpt-6-astra) for code from briefs and its built-in image generation. Meshy image-to-3D for 26 meshes from our own concept images. Tripo Studio, driven by Claude in Chrome or by hand, for creatures, props, Mimics, traps, heroes and auto-rigs. ChatGPT image generation for the adventurers' concept sheets and the title badge. The UE 5.9 Kimodo plugin (NVIDIA Kimodo motion model, Meta Llama 3 text encoder), run locally, for animation clips. Python scripts (numpy, scipy, Pillow) for sound effects and music, synthesised with no samples, and for texture maps. MiniMax Audio text-to-speech for the raiders' 307 voice lines, in five library voices, none cloned from a real person. Blender 5.1 and 5.3 beta, headless, for fitting models. NVIDIA's DLSS 4.5 plugin, compiled against 5.9. Adobe Mixamo: packs for posing in Blender, and two of its animations in the game (below).

Content we did not make: the UE5 template mannequins and their locomotion, the Roboto engine font, Niagara template effects, NVIDIA's DLSS plugin, one creature sound from We Love Indies, two Adobe Mixamo animations (Slipping and Dizzy Idle, on the five raiders, added in the last hours), and two system fonts used only to draw three pictures. The full list with licences is `ASSETS.md`.

---

7. Numbers

WHAT	NUMBER
Scenario files	433
Codex briefs kept	147
Codex sessions in the jam's folders on SirPick's PC, Nov 12 to 15	255
Packaged builds	44 by 8:11 PM ET on Nov 15: the first, then v2 to v44

WHAT	NUMBER
Commits	About 460 on the repository's one counter for every branch (the test branch was at dv.commit.460 and `main` at 461 at 8:11 PM ET on Nov 15)
Meshy credits	780: two batches of 13 models
Tripo credits	SirPick's account: 1,790 through Nov 14, 330 of them the mistake in item 1. Mignightxrush's: 540 on Nov 14 and 135 on Nov 15. Tallboy's: 170 for a sword and shield; the credits for the 39 adventurers were not recorded
MiniMax credits, Mignightxrush's account (free credits)	About 7,600 characters in 309 clips, 307 of them used; about 1,900 more wasted (item 9)
Frame rate, packaged	116 fps at 1920 x 1080 with everything on, 108 in a busy fight (RTX 4070 laptop, `main` at dv.commit.374)
Swift's first night	Sorceress 563 credits, Tripo 660 credits
Zipped build size	0.61 GB (Nov 12), 0.85 GB (v5), 0.87 GB (v10), 1.12 GB (v26), 1.19 GB (v44: 1,191,132,575 bytes). Our own limit was 1 GB; SirPick chose not to cut content, so the upload uses butler, itch's command-line uploader

Not proven: most Nov 14 frames of the generated dungeon are the builder's own, balance numbers come from scripted runs, and DLSS was never tried on a card that is not NVIDIA's.

---

8. Where it ended

At 8:11 PM Eastern on Nov 15, 48 minutes before the deadline, package v44 came out of the cook with all three people's work in it: 1,191,132,575 bytes zipped, no cook errors, its smoke scenario passing inside the packaged exe. An illustrated opening was drawn and is in the build switched off, so a player never sees it. Not in it at all: a last try at putting the theme into play, with each party's strongest raider as its "Director" calling "Cut!" when the party retreats. An agent measured first: a party retreated in 2 of 230 raids. SirPick's call was no game change, and the code sits uncompiled in a staging folder. The last change into the build was Tallboy's: a slip and a dizzy animation for the raiders, taken from Adobe Mixamo and credited in `ASSETS.md` and on the itch page. One recorded voice line, a joke borrowed from an internet meme that an agent had flagged as wrong for a Teen-rated entry, was in the game for a few hours on the last day; SirPick cut it before submission. Our own schedule had the final package at 3 PM Eastern. It slipped by about four hours.

The last hours went on balance. A first outside playtest that afternoon found the raiders too weak by round 8. Its log showed the party shrinking from 4 to 2 or 3 between rounds 1 and 10 while the defence stayed at nine goblins, a slime and five traps: the Heart was never hit, and the goblins landed 331 bites against 67 hits taken. Package v34 gave raiders gear and skills by level, a floor on party size, and 6% health and 4% damage a round. SirPick played it: "repairing should cost mana. right now there is no downside to death. everything is still too easy". v36 raised that to 15% health and 8% damage a round up to round 15, with a fourth raider from round 5. v37 makes repairs in a raid cost Mana: a quarter of a trap's price to re-arm it, half a den's price to revive it. SirPick then played v38, and v39 was built from his notes: Mana upkeep between rounds for spent traps and hurt den creatures, and a five-tile

no-build zone at the raiders' door, because two Goblin Dens beside it beat every party until round 5. His log also showed an archer shooting a spider dead in five arrows from outside its web while it stood idle; it now fights back. Scripted checks on those last changes pass. They went in during the final two hours, too late for another outside playtest.

Four rules we would hand to another team:

- **Put a person in front of the moving game early and often.** Nine of our failures passed every check, and reviewer agents reading stills passed a mop a person rejected at once.
- **Name the engine feature before you write your own.** After three rounds of fixes on one thing, stop and look at the whole.
- **Write down what an agent does when nobody answers.** Smallest safe choice, record it, keep going.
- **Keep spending, deleting and posting with a person.** The 330 credits and the 1,900 credits in section 4 were both spent by agents doing what they thought was asked.

About this page: an agent compiled it on Nov 15 from our running log, the workstream logs, `ASSETS.md` and the credit ledger. A second agent, shown only the page, read it cold as a judge would, twice, and this version answers its lists of faults. File names in `this type` are in our Diversion repository.