

PORTOFOLIO

PANDORA'S SNAKE — GAMESEED 2026, SLAFURRY STUDIOS

TEAM PROJECT (11 members) — Project Lead & Technical Designer

Pandora's Snake is an absurd comedy action-destruction game about Pandora, a reckless witch streamer, and Jormungandr, the legendary World Serpent, equipped with a machine gun strapped to his back. This was built for the GameSeed 2026 game jam.

My biggest challenge was merging four very different gameplay systems — a wanted-level mechanic, a roguelike structure, procedural map generation, and Snake-like movement into one coherent, scalable architecture, without the team blocking on each other's code. I restructured team responsibilities across four different disciplines to avoid single points of failure, and made the tough call to cut scope from four levels down to one polished level when we hit the four-week deadline.

The unified architecture let an 11-person team iterate in parallel instead of blocking on each other's work, and cutting scope early meant we shipped something genuinely polished, the game placed Top 61 out of 500+ submissions.

Turns out prioritizing depth over breadth is harder emotionally than it is technically. I had to convince the team and myself, that one great level beats four mediocre ones. This jam taught me that architecture decisions made in the first few days directly determine how painful (or painless) scope cuts are later in the project.



Demo: <https://youtu.be/OZleQY8Gmlg>

Download: <https://lordzaini.itch.io/pandoras-snake>

Github: <https://github.com/Slafurry-Studios/Pandora-Snake>



Demo: https://youtu.be/_cmhbfIWUXk

Download: <https://slafurrystudios.itch.io/an-unfinished-game>

Github: <https://github.com/Slafurry-Studios/An-Unfinished-Game>

AN UNFINISHED GAME — GAME TODAY 2026, SLAFURRY STUDIOS

TEAM PROJECT (5 members) — Project Lead & Game Programmer

An Unfinished Game is a 2D narrative platformer about a game studio founder who can't let go of his perfectionist ideals — to the point where those ideals turn against him.

Around the midpoint of development, I realized the original plan with 20 rooms and telling the full story wasn't going to be finished in time. I had to restructure the narrative itself, not just cut content, so the story still landed emotionally in 10 rooms instead of 20.

Alongside that, I built development automation (reminder bots, a Google Drive retrieval bot, a deployment bot) that reduced repetitive manual work across the team by roughly 40%, and introduced Semantic Versioning so bug tracking and releases stayed organized as we iterated quickly.

The narrative restructuring meant the game shipped with a complete emotional arc despite losing half its planned content, and the automation tooling freed the team to spend more time on the actual game instead of manual busywork.

Cutting a story in half without breaking it forced me to identify which scenes actually carried the emotional weight versus which were just nice to have. It's a skill I didn't expect a programmer role to teach me recognizing what's essential versus decorative applies to code architecture just as much as it applies to narrative.



DIRECTIVE — BULLET HELL JAM, SLAFURRY STUDIOS

TEAM PROJECT (8 members) — Project Lead & Game Programmer

Directive is a top-down 2D bullet hell shooter built for Bullet Hell Jam. The goal was simple, engaging gameplay backed by serious technical optimization, the kind of game that has to stay smooth even when hundreds of bullets are on screen at once.

I led four programmers working in parallel, which meant designing a system architecture upfront that let each person build independently without merge conflicts or overlapping responsibilities. On the technical side, I optimized bullet collision using Boxcast2D instead of a more expensive default approach, implemented a sprite atlas to cut down draw calls, and built a controller-based AI system that separated "brain" (decision-making) from "action" (execution) — which made it much easier to design new enemy behaviors quickly under jam time pressure.

These optimizations took the game from running at roughly 15 FPS to a stable 60 FPS under heavy bullet loads, the difference between a game that felt broken and one that felt genuinely playable.

Performance problems in games are rarely about "writing faster code" they're about picking the right data structures and algorithms for the specific bottleneck. Boxcast2D wasn't a magic fix; it was the right tool for this specific collision problem. This project taught me to actually profile before optimizing, instead of guessing.

Demo: <https://youtu.be/FFWvz2SXUBI>

Download: <https://lordzaini.itch.io/directive>

Github: <https://github.com/Slafurry-Studios/Directive>

OMNI-GEAR PROTOCOL — GAMEDEV.JS JAM, SLAFURRY STUDIOS

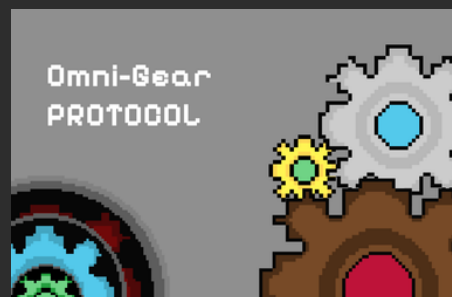
TEAM PROJECT (8 members) — Project Lead & Game Programmer

Omni-Gear Protocol is a physics platformer where you play a robot shell that's nearly useless on its own, its ability to move and interact with the world depends entirely on three specialized "Cogs" you carry and swap between. Built for Gamedev.js Jam, and released open-source with full documentation.

Because the team was spread across three countries and working asynchronously, I focused less on hands-on coding and more on making sure the team could work without me being a bottleneck: writing clear, intuitive documentation for our level designer so they could build puzzles without waiting on programmer availability, reviewing every pull request on GitHub with actionable feedback, and keeping the team aligned through regular progress and testing videos across time zones.

The puzzle-mobility mechanic scaled cleanly across many levels without needing constant programmer intervention, and the open-source documentation meant the project remained usable and understandable after the jam ended, not just a pile of jam-code nobody could touch again.

Leading a distributed, multinational team taught me that documentation isn't a "nice to have", it's the actual infrastructure that lets asynchronous teams function at all. When you can't just walk over and explain something, writing it down clearly is the leadership.



Demo: <https://youtu.be/NinhfBNIOqo>

Download: <https://lordzaini.itch.io/omni-gear-protocol>

Github: <https://github.com/Slafurry-Studios/Omni-Gear-Protocol>
