

Tova 6-Week Roadmap

Prepared February 3, 2026

Repository Snapshot

- Vite + Three.js single-page client (index.html + src/main.js).
- Current world: environment (sky/fog/lights), procedural terrain with hill and town flat, ocean plane, castle and town primitives.
- Player: pointer-lock first-person, WASD movement, chat commands for fly/walk, terrain alignment via getHeightAt raycasts.
- Postprocessing: EffectComposer with bloom and tone mapping; performance tuned (no AA, capped pixel ratio).
- Godot backup project provides voxel interaction reference but is not wired into the web app.

Architecture Constraints

- Terrain origin and flattened town area are hard-coded; castle/town depend on these coordinates.
- Terrain.getHeightAt uses per-query raycasts; scaling actor count could be expensive.
- Pointer lock is required for core controls; mobile support will need an alternative input scheme.
- No test or lint scripts exist in package.json; quality checks are manual today.
- Only dependency is three; adding features may require new libraries (noise, UI, assets).

Milestones

- Milestone 1 (Weeks 1-2): Stabilize core world systems and add reproducible world configuration.
- Milestone 2 (Weeks 3-4): Add interaction layer and exploration objectives.
- Milestone 3 (Weeks 5-6): Content pass, polish, and release readiness.

Dependencies

- Seeded terrain may require a noise library or a custom deterministic noise utility.
- Interaction/UI work needs a shared state model plus a raycast targeting system.
- New assets (models, textures, audio) need a pipeline and clear licensing.
- Performance profiling tools (Stats.js or custom HUD) are needed for tuning.

Risks

- Increased raycasts can reduce frame rate; may need a heightmap or cached sampling.
- Bloom and water transmission can be expensive on integrated GPUs.
- Pointer lock permissions can block onboarding or mobile usage.
- Tight coupling to terrain coordinates makes large-scale world changes risky without refactors.

Week-by-Week Plan

Week 1 Goal | Deliverables

1 | Baseline and planning | Document invariants, centralize world constants, add dev HUD (FPS + toggles).

2 | Deterministic terrain + environment | Seeded terrain generation, expose water/fog/lighting params, add spawn points.

3 | Interaction framework | Raycast target system, hover highlight, on-screen prompt, interactable interface.

4 | Exploration loop | Points of interest for castle/town, simple objectives list, lightweight progress UI.

5 | Content + performance pass | Improve town layout/roads/props, optimize height queries or LOD, tune shadows.

6 | Release readiness | Quality settings, help screen, build + README updates, bug triage.