Sen proceduralny

Sen proceduralny

Strona referencyjna. Pliki: `core/dream/` (`dream_generator.gd`, `dream_slot.gd`, `dream_palette.gd`, `dream_palette_entry.gd`, `dream_anchor.gd`, `dream_spawn_point.gd`), `content/dream/` (palety i makiety), `content/rooms/room_d.tscn`. Plan w `projekt.md`, sekcje 3.1 i 3.2. Opowieść i klocki: odcinek „Sen, który za każdym razem jest inny”.

Dwie warstwy

- **Warstwa stała** (ręczna, wspólna dla jawy i snu): droga główna, punkty stałe (np. namiot, ognisko), obiekty fabularne. We śnie wygląda gorzej (`RealmLook`, `RealmAware`), ale jej układ się nie zmienia.
- **Warstwa losowa** (tylko sen i koszmar): odnogi od drogi, woda, roślinność, obozy ze spawnami. Buduje ją `DreamGenerator` przy każdym wejściu w sen i usuwa po obudzeniu.

Ziarno (powtarzalna losowość)

Element	Opis
`GameState.dream_seed`	Ziarno bieżącego snu (`get_dream_seed()`, `set_dream_seed()`). 0 = jawa. Zapisywane w `world.dream_seed` (`SAVE_VERSION = 6`; starsze zapisy dostają 0).
Losowanie	`RealmManager.enter_dream()` losuje nowe ziarno (różne od 0) przy każdym wejściu w sen albo koszmar. `return_to_waking()` je zeruje. `switch_dream_kind()` (sen ↔ koszmar) zachowuje ziarno.
`RealmManager.reseed_dream(seed)`	Nowe ziarno bez zmiany rzeczywistości (0 = losowe), sygnał `dream_reseeded`; tylko we śnie.
Ziarno slotu	Hash tekstu `"<dream_seed>
Kolejność	Sloty są przetwarzane po `slot_id`, alfabetycznie.

Skutek: to samo ziarno daje ten sam sen, także po zapisie i wczytaniu oraz po wyjściu z pokoju i powrocie w tym samym śnie. Zmiana `slot_id`, palety albo położenia elementów warstwy stałej zmienia układ.

DreamSlot (Marker3D)

Pole	Opis
`slot_id`	Stały identyfikator w pokoju, np. `vegetation_north`.
`slot_type`	`BRANCH` (odnoga), `WATER`, `VEGETATION`, `CAMP`.
strefa	Dziecko `Area3D` z `CollisionShape3D` (pudełko albo kula); bez niego koło o promieniu `radius`.
`min_items`, `max_items`	Ile elementów (dla `CAMP`: obozów). W koszmarze razy `nightmare_density` generatora.
`min_distance`	Minimalna odległość między elementami slotu.
`dream_palette`, `nightmare_palette`	Palety snu i koszmaru (puste `nightmare_palette` = paleta snu).
Odnoga	`branch_length_min/max`, `branch_step`, `branch_turn_degrees`.
Obóz	`spawn_table_id`: nadpisuje `table_id` w `DreamSpawnPoint` obozów z tego slotu.

DreamPalette i DreamPaletteEntry

Paleta to lista wpisów. Wpis ma pola: `scene`, `weight` (waga losowania), `min_count` i `max_count` (na slot; 0 = bez limitu), `radius` (promień zajętości) i `tags`. Odnoga bierze wpisy z tagiem `path` (kawałki ścieżki) i `camp` (obóz na końcu).

DreamGenerator (węzeł w pokoju)

Pole	Domyślnie	Opis
`road_path`	(brak)	Droga główna (`Path3D`).
`road_clearance`	2.5 m	Wolny pas wokół osi drogi.
`auto_protect_radius`	1.5 m	Wolny promień wokół spawnów, ognisk i `Interactable` pokoju (bez tagu od autora).

Pole	Domyślnie	Opis
`story_clearance`	1.5 m	Wolny promień wokół obiektów z grupy `story_critical`.
`max_generated`	200	Najwięcej elementów w pokoju (po przekroczeniu ostrzeżenie).
`nightmare_density`	1.5	Mnożnik liczby elementów w koszmarze.
`room_id_override`	(puste)	Id pokoju do ziarna; puste = bieżący pokój.

- Nasłuchuje `realm_changed`, `dream_reseeded`, `room_changed` i `state_replaced`. We śnie buduje węzeł `Generated` (grupa `dream_generated`), na jawie go usuwa.
- Reguły: nic bliżej drogi niż `road_clearance`, w promieniu `DreamAnchor` ani przy `story_critical`; elementy nie nachodzą na siebie (promienie), w slocie trzymają `min_distance`. Elementy statyczne pokoju są chronione automatycznie, bez tagu od autora: spawny (`spawn_points`), ogniska (`meditation_points`) i wszystkie `Interactable` (portale, drzwi, przedmioty), w promieniu `auto_protect_radius` (domyślnie 1.5 m).
- **Odnoga**: zaczyna się w slocie przy drodze (pierwszy kawałek tuż za wolnym pasem), idzie od drogi z losowymi skrętami i kończy ślepym zaułkiem z obozem. Na przeszkodzie (punkt stały, inny element) kończy się wcześniej.
- **Obóz**: makieta z `DreamSpawnPoint` (`table_id`). Prawdziwych przeciwników podłączy zadanie 009.
- Slot bez palety albo z pustą paletą: ostrzeżenie (`push_warning`), slot jest pomijany.
- Zduplikowany `slot_id` w pokoju: ostrzeżenie (`push_warning`), bo dwa sloty z tym samym id dostają to samo ziarno (`duplicate_slot_ids()`).
- Odczyt: `get_layout()` (slot, typ, scena, pozycja, promień), `counts_by_type()`, `set_slots_visible()`.

DreamAnchor i story_critical

- `DreamAnchor` (Marker3D, grupa `dream_anchor`, `radius`): punkt stały warstwy stałej, np. pod namiotem albo ogniskiem.
- Grupa `story_critical`: obiekty fabularne, których generator nie może zasłonić.

Pokój testowy „droga” (`room_d`)

Droga z zachodu na wschód (`Path3D`), namiot i ognisko z `DreamAnchor`, kamień w grupie `story_critical`, sloty `branch_north`, `camp_far`, `vegetation_north` (strefa-pudełko), `vegetation_south` (strefa-kula), `water_east`. Wejście: portal w pokoju A.

Menu deweloperskie

Sekcja „Sen proceduralny” (F9): bieżące ziarno, „Nowe ziarno (przeładuj sen)”, „Ustaw ziarno”, liczniki elementów według typu, „Pokaż strefy slotów”.

Testy

`tests/test_dream_generator.gd`, `tests/test_dev_menu.gd` (sekcja snu), `tests/test_state.gd` (zapis v6).

Opublikowano z repozytorium Horrorek (commit 87717ad) skryptem publikuj-bookstack.ps1.

Revision #1
Created 2026-10-28 22:54:00 UTC by Tomasz Konieczny
Updated 2026-10-28 22:54:00 UTC by Tomasz Konieczny