Medytacja i sen

Medytacja i sen

Strona referencyjna. Pliki: `core/state/game_state.gd`, `core/realm/`, `core/meditation/`, `core/ui/fade_overlay.tscn`, `content/realm/`. Opowieść i klocki: odcinek „Ognisko, sen i druga wersja tego samego pokoju”.

GameState: czas i rzeczywistość

`GameState` jest jedynym źródłem prawdy o stanie świata. Pozostałe systemy go odzwierciedlają.

Metoda	Działanie
`advance_time(hours)`	Przesuwa zegar (hours > 0). Po 24:00 zawija godzinę i zwiększa dzień, także o kilka dni naraz.
`get_hour() -> float`, `get_day() -> int`	Godzina 0–24 (z ułamkiem) i numer dnia (od 1).
`set_realm(realm)`, `get_realm()`	Rzeczywistość: `GameState.REALM_WAKING` (`waking`) albo `GameState.REALM_DREAM` (`dream`). Inne wartości są ignorowane.
`set_last_point(id, room_id)`, `get_last_point()`, `get_last_point_room()`	Ostatni punkt medytacji i jego pokój: tam wraca gracz ze snu, także gdy śpiąc przeszedł do innego pokoju.

- Sygnały: `hour_changed(hour)`, `day_changed(day)`, `state_replaced` (po `reset()` i `deserialize()`).
- Zapis: `serialize()` ma sekcję `world` (`hour`, `day`, `realm`, `last_meditation_point`) i od wersji 3 sekcję `story` (akt, zakres questu, użycia koszmaru; zob. „Koszmar i skok”). Pokój ogniska (`last_meditation_room`) i bieżący pokój (`room_id`) są w sekcji `world` od wersji 4–5 (zob. „Lokacje”). Od wersji 6 zapis ma też ziarno snu (`world.dream_seed`, zob. „Sen proceduralny”). Od wersji 7 także statystyki (zob. „Statystyki, leczenie i awans”). Obecnie `SAVE_VERSION = 7`. Starsze zapisy wczytują się z wartościami domyślnymi: 8:00, dzień 1, jawa, akt 1.

RealmManager (autoload)

Przejścia między jawą a snem z zaciemnieniem ekranu (`FadeOverlay`).

Element	Opis
`enum Realm { WAKING, DREAM, NIGHTMARE }`	Rzeczywistość w sygnałach. `get_realm_id()` zwraca odpowiadający StringName. Koszmar opisuje strona „Koszmar i skok”.
`enter_dream(dream_id, kind = Realm.DREAM) -> bool`	Zaciemnienie, zapis `dream` w GameState, flaga `dream_<id>_visited`, sygnały `realm_changed` i `dream_entered`, rozjaśnienie. `false`, gdy gracz już śpi albo trwa przejście.
`return_to_waking() -> bool`	Jak wyżej, plus: jeśli ognisko z `last_meditation_point` stoi w innym pokoju (`last_meditation_room`) niż bieżący, ładuje ten pokój (`LocationManager`), a potem ustawia gracza (grupa `player`) na `ReturnMarker` ogniska.
`transition_seconds`	Czas zaciemnienia i rozjaśnienia (domyślnie 0.6 s; w testach 0).
`is_transitioning()`	Czy trwa przejście.
Sygnały	`realm_changing(to)`, `realm_changed(realm)`, `dream_entered(dream_id)`, `nightmare_entered(dream_id)`.

Po wczytaniu zapisu (`state_replaced`) RealmManager przyjmuje rzeczywistość z GameState bez animacji.

RealmAware

Komponent `Node3D`. Pole `realms: Array[StringName]`, np. `[&"dream"]`.

- Węzeł jest aktywny tylko w rzeczywistościach z listy. Poza nimi: `visible = false`, `process_mode = DISABLED` oraz wyłączone kolizje siebie i potomków (CollisionObject3D: warstwy 0, Area3D: także `monitoring`/`monitorable`, CSGShape3D: `use_collision`). Oryginalne wartości wracają po włączeniu.
- Węzły bez `RealmAware` są wspólne dla obu rzeczywistości.
- Nie zagnieżdżaj `RealmAware` w `RealmAware`.
- Ten sam mechanizm obsłuży przeciwników w dwóch formach.

RealmLook i RealmEnvironment

- `RealmLook` (zasób `.tres`): `background_color`, `ambient_color`, `ambient_energy`, `light_energy`, `fog_enabled`, `fog_density`, `fog_light_color`.

- `RealmEnvironment` (Node): `waking_look`, `dream_look`, `world_environment_path`, `light_path`. Przy starcie i przy `realm_changed` stosuje właściwy wygląd.
- Każda lokacja ma własne zasoby w `content/realm/` (np. `basic_waking.tres`, `basic_dream.tres`, `basic_nightmare.tres`, wspólne dla pokoi roboczych A, B, C).

MeditationPoint (ognisko)

Podklasa `Interactable` w grupie `meditation_points`, z dzieckiem `ReturnMarker` (Marker3D).

Pole lub metoda	Opis
`point_id`	Stały identyfikator punktu, np. `meditation_room_a`.
`dream_id`	Sen otwierany z tego punktu (flaga `dream_<id>_visited`).
`dream_available`	Czy z punktu w ogóle da się zasnąć.
`required_flag`	Warunek fabularny; puste = brak warunku.
`rest_rules`	Co daje „Odpocznij” (`MeditationRules`: część zdrowia i poczytalności do przywrócenia). Puste = w pełni. Zob. „Statystyki, leczenie i awans”.
`can_dream() -> bool`	`dream_available` i prawdziwa flaga `required_flag`.
`open_menu() -> MeditationMenu`	Otwiera menu (interakcja robi to sama, zapisując też `last_meditation_point`).
`menu_opened(menu)`	Sygnał dla systemów, które dopisują pozycje menu.

add_option: nowe pozycje menu medytacji

```
menu.add_option(id, label_key, is_available, on_selected, extra)
```

- `is_available`: Callable bez argumentów zwracający bool (false = pozycja wyszarzona).
- `on_selected`: Callable bez argumentów.
- `extra`: `realms` (gdzie pozycja jest widoczna), `locked_label_key`, `on_adjust` (strzałki lewo/prawo), `label_suffix`, `order` (domyślnie 100, „Wróć” ma 1000).

Szczegóły menu (pauza, mysz, sterowanie): strona „Menu: pauza i medytacja”.

Testy

`tests/test_state.gd`, `tests/test_realm.gd`, `tests/test_meditation.gd`, `tests/test_main_scene.gd`, `tests/test_realm_rooms.gd`.

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

Revision #4
Created 2026-10-27 23:24:46 UTC by Tomasz Konieczny
Updated 2026-10-28 22:53:55 UTC by Tomasz Konieczny