Lokacje: scena główna, pokoje i portale

Lokacje: scena główna, pokoje i portale

Strona referencyjna. Pliki: `content/main.tscn`, `core/game/game_root.gd`, `core/world/` (`location_manager.gd`, `room_info.gd`, `room_registry.gd`, `spawn_point.gd`, `portal.gd`, `portal.tscn`), `content/world/rooms.tres`, `content/rooms/room_a.tscn`, `room_b.tscn`, `room_c.tscn`.

Scena główna

`content/main.tscn` to scena startowa gry (`run/main_scene`). Jej korzeń (`GameRoot`) trzyma elementy trwałe, które nie znikają przy zmianie pokoju:

Węzeł	Rola
`Player`	Gracz z czujnikiem interakcji.
`MainCamera3D/PhantomCameraHost`, `PlayerCamera`, `CameraControl`	Kamera za plecami sterowana myszą i padem.
`InteractionPrompt`, `PauseMenu`	Podpowiedź i menu pauzy.
`FadeOverlay`	Wspólne zaciemnienie (grupa `fade_overlay`), używane przez `LocationManager` i `RealmManager`.
`RoomHost`	Węzeł, pod którym ładowany jest bieżący pokój.

Pola `GameRoot`: `room_registry` (rejestr pokoi), `start_room` i `start_spawn` (początek nowej gry), `room_host` i `player`. Przy starcie `GameRoot` ładuje pokój z zapisu (`GameState.get_room()`), a gdy go nie ma, pokój startowy.

Pokój

Pokój to scena **bez gracza i kamery**. Ma własne:

- WorldEnvironment, światło i `RealmEnvironment` (wygląd jawy, snu i koszmaru),
- geometrię, obiekty, ogniska i obiekty `RealmAware`,
- spawny (`SpawnPoint`) i portale (`Portal`).

Rzeczywistość i ogniska między pokojami

- Rzeczywistość (jawa, sen, koszmar) nie zmienia się przy przejściu przez portal: nowy pokój przy starcie sam ustawia `RealmAware` i `RealmEnvironment` według `RealmManager`. Śpiący gracz może więc przejść do innego pokoju i zobaczy go w wersji snu (albo koszmaru).
- Ognisko (`MeditationPoint`) przy interakcji zapisuje w `GameState` swój punkt i pokój (`set_last_point(id, room_id)`).
- `RealmManager.return_to_waking()`: po zaciemnieniu ustawia jawę, a jeśli ognisko stoi w innym pokoju niż bieżący, ładuje ten pokój (`LocationManager.load_room_now`) i stawia gracza na `ReturnMarker` ogniska.
- Wczytanie zapisu we śnie: `RealmManager` przyjmuje sen z `GameState`, a `SaveManager` ładuje zapisany pokój i stawia gracza tam, gdzie stał (zob. strona „Zapis gry”). Przy starcie sceny głównej z samym stanem (bez pliku zapisu) gracz staje na pierwszym spawnie.

Rejestr pokoi

- `RoomInfo` (zasób): `id` (np. `room_a`), `scene`, `name_key` (np. `ROOM_A_NAME`).
- `RoomRegistry` (zasób): lista `rooms`, metoda `find(id)`.
- `content/world/rooms.tres`: rejestr gry. Nowy pokój = nowy wpis tutaj.

LocationManager (autoload)

Element	Opis
`setup(room_host, registry, player)`	Podłącza scenę główną (woła `GameRoot`).

Element	Opis
`go_to(room_id, spawn_id = "") -> bool`	Zaciemnienie, wymiana pokoju, gracz na spawnie, rozjaśnienie. `false` dla nieznanego pokoju albo w trakcie przejścia.
`load_room_now(room_id, spawn_id = "") -> bool`	Wymiana bez zaciemnienia (start gry, wczytanie zapisu).
`get_current_room()`, `get_room_node()`, `get_room_info(id)`, `is_transitioning()`	Odczyt stanu.
`transition_seconds`	Czas zaciemnienia i rozjaśnienia (domyślnie 0.4 s; w testach 0).
Sygnały	`room_changing(from, to)`, `room_changed(room_id)`.

- Bieżący pokój zapisuje się w `GameState` (`set_room`/`get_room`, pole `world.room_id`), a pokój ostatniego ogniska w `world.last_meditation_room` (obecnie `SAVE_VERSION = 7`). Starsze zapisy mają puste pola: gra startuje w pokoju startowym, a powrót ze snu zostaje w bieżącym pokoju.
- `has_world()`: czy scena główna wywołała `setup()` (bez niej pokoi nie da się ładować).
- Stary pokój jest usuwany z drzewa od razu, więc grupy (spawny, ogniska) nie zawierają jego węzłów.

SpawnPoint

`Marker3D` ze skryptem `spawn_point.gd`, pole `spawn_id` (np. `from_b`). Gracz staje na pozycji markera i patrzy wzdłuż jego osi −Z. Bez podanego `spawn_id` (albo przy nieznanym) gracz staje na pierwszym spawnie pokoju.

Portal

Podklasa `Interactable` (`core/world/portal.tscn`):

Pole	Opis
`target_room`, `target_spawn`	Dokąd prowadzi przejście.
`required_flag`	Flaga, która musi być prawdziwa; puste = zawsze otwarte.
`locked_prompt_key`	Podpowiedź przy zablokowanym przejściu (domyślnie `PROMPT_PASSAGE_LOCKED`).

Podpowiedź otwartego portalu to `PROMPT_GO` („Przejdź”). Zablokowany portal emituje `denied(actor)`, a gracz zostaje w pokoju.

Trwałość obiektów

- `Pickup.pickup_id`: zebranie zapisuje flagę `collected_<pickup_id>`, a przy ponownym wejściu do pokoju przedmiot sam znika. Bez `pickup_id` przedmiot odradza się za każdym razem.
- `Door.open_flag`: drzwi z ustawioną flagą startują otwarte, także po powrocie do pokoju.

Quest testowy (pokoje robocze)

- **A**: klucz (`pickup_room_a_key_cellar`), ognisko A, portal do B; wersja snu (ściany snu, monolit) i koszmaru (oko), stół tylko na jawie.
- **B**: drzwi na klucz (otwór 1.2 × 2.2 m), za nimi przedsionek z portalem do C (`required_flag = door_cellar_open`), portal do A.
- **C (piwnica)**: ognisko B (sen wymaga `door_cellar_open`), portal do B.
- **D (droga)**: droga główna, namiot, ognisko D, kamień `story_critical`, sloty snu (odnoga z obozem, obóz, roślinność, woda); portal do A (pokój A ma portal do D). Zob. strona „Sen proceduralny”.

Wszystkie trzy pokoje używają wspólnych zasobów wyglądu `content/realm/basic_waking.tres`, `basic_dream.tres`, `basic_nightmare.tres`. Docelowo każda lokacja dostanie własne.

Testy

`tests/test_location.gd`, `tests/test_quest_rooms.gd`, `tests/test_door_passage.gd` (drzwi w pokoju B), `tests/test_main_scene.gd` (scena główna, wersja snu pokoju A, ogniska), `tests/test_realm_rooms.gd` (sen i koszmar przez portale, obudzenie w pokoju ogniska, zapis we śnie).

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

Revision #2
Created 2026-10-28 00:48:14 UTC by Tomasz Konieczny
Updated 2026-10-28 22:53:58 UTC by Tomasz Konieczny