Odcinek: Menu, które widzi tylko twórca

Odcinek: Menu, które widzi tylko twórca

> Szkic do BookStacka. Faza 1. Poziomy: **[A]** klikane w edytorze, **[B]** gotowy skrypt do wklejenia, **[C]** skrypt z komentarzami dydaktycznymi. Pliki w repozytorium: `core/dev/dev_menu.tscn`, `core/dev/dev_menu.gd`, `core/dev/dev_actions.gd`, `content/main.tscn`, test `tests/test_dev_menu.gd`.

Opowieść

Żeby obejrzeć koszmar w piwnicy, trzeba było dotąd: podnieść klucz w pokoju A, przejść do B, otworzyć drzwi, zejść do piwnicy, zasnąć przy ognisku i liczyć na 5% szansy. Za każdym razem.

Teraz wystarczy nacisnąć **F9**. Gra staje, pojawia się kursor i panel z przyciskami. „Koszmar” i ekran gaśnie, a po chwili jesteśmy w koszmarze w tym pokoju, w którym staliśmy. „Pokój: room_c” i jesteśmy w piwnicy. „door_cellar_open”: przełącznik i drzwi są otwarte, choć nikt ich nie otwierał. „Następne zaśnięcie = koszmar”: ognisko przy najbliższym „Zaśnij” na pewno da koszmar, a gra nie odnotuje, że koszmar w tym queście już był.

Gracz tego menu nigdy nie zobaczy. W wersji wydanej węzeł menu usuwa się sam w pierwszej klatce, jakby go nigdy nie było.

Pojęcie: narzędzie deweloperskie a stan gry

> **Dobre narzędzie deweloperskie nie ma własnych sztuczek. Zmienia stan gry tymi samymi drogami, którymi zmienia go gra.**
>
> - Przycisk „Sen” nie przestawia ręcznie widoczności ścian. Woła `RealmManager.enter_dream()`, czyli to samo, co ognisko. Dlatego to, co widzimy, jest prawdziwym snem, a nie podróbką.
> - Narzędzie może **pominąć warunki** (losowanie koszmaru, flagę fabularną przy ognisku), ale nie **mechanikę** (zaciemnienie, sygnały, zapis w `GameState`).
> - Treść (które flagi, które przedmioty) nie jest wpisana w kod menu. Ustawia ją autor w scenie głównej, w polach `known_flags` i `give_items`.
> - Wydanie jest czyste: `OS.is_debug_build()` mówi, czy to wersja deweloperska. Jeśli nie, menu znika.
>
> Zasada kciuka: **skrót przez warunki tak, skrót przez mechanikę nie.**

Klocek 1 [A]: własna flaga i przedmiot w menu

1. Otwórz `content/main.tscn` i zaznacz węzeł **DevMenu**.
2. W polu **Known Flags** dopisz nazwę flagi, np. `lever_cellar_pulled`. W menu pojawi się przełącznik.
3. W polu **Give Items** dodaj zasób przedmiotu z `content/items/`. W menu pojawi się przycisk „Daj: …”.
4. Uruchom grę (F5) i naciśnij F9.

Klocek 2 [B]: nowa sekcja menu

Dowolny skrypt sceny może dopisać sekcję. Sekcja „Sen proceduralny” ma już zarezerwowane miejsce w kolejności (zadanie 006):

```
func _ready() -> void:
	var dev_menu: Node = get_tree().current_scene.get_node_or_null("DevMenu")
	if dev_menu == null:
		return  # wydanie: menu się usunęło
	dev_menu.add_section(&"procedural_dream", "Sen proceduralny", _build_dream_section)


func _build_dream_section(box: VBoxContainer) -> void:
	var button: Button = Button.new()
	button.text = "Nowe ziarno"
	button.pressed.connect(func() -> void: print("przeładowanie snu"))
	box.add_child(button)
```

Klocek 3 [C]: jak menu znika z wydania

Wersja z komentarzami dydaktycznymi. Pełny kod jest w `res://core/dev/dev_menu.gd`.

```
## Sprawdzenie, czy to build debug. Statyczne pole, żeby testy mogły je podmienić.
static var debug_build_check: Callable = Callable(OS, &"is_debug_build")

## Włącza menu także w wydaniu (tylko testy).
@export var force_enabled: bool = false


func _ready() -> void:
	# 1. Wersja wydana (release): is_debug_build() zwraca false.
	#    Wtedy węzeł usuwa się sam i nie zostawia w grze żadnego śladu:
	#    ani klawisza F9, ani panelu, ani przycisków.
	if not (force_enabled or bool(debug_build_check.call())):
		queue_free()
		return
	# 2. Menu ma działać także wtedy, gdy samo wstrzymuje grę (pauza).
	process_mode = Node.PROCESS_MODE_ALWAYS
	# 3. Panel jest schowany do pierwszego F9.
	_root.visible = false
	_title.text = TEXTS[&"title"]
```

```
□_add_default_sections()
```

Do przemyślenia: dlaczego sprawdzenie builda jest w polu `debug_build_check`, a nie bezpośrednio jako `OS.is_debug_build()`? Jak bez tego sprawdzić w teście, że w wydaniu menu znika?

Sprawdź, że działa

Z katalogu głównego projektu, w PowerShellu:

```
.\tools\run-tests.ps1
```

Oczekiwane: w tabeli każdy wiersz ma `Wynik = OK`, w tym `test_dev_menu`. W edytorze: F5, potem F9.

Ciekawostka z kulis

Przełączenie „Sen → Koszmar” wymagało nowej funkcji w `RealmManager`. Bez niej menu musiałoby najpierw obudzić gracza, a obudzenie przenosi go do ostatniego ogniska, czasem do innego pokoju. Narzędzie, które przy okazji teleportuje, myliłoby więcej, niż pomagało. Stąd `switch_dream_kind()`: zmienia odmianę snu z zaciemnieniem, nie ruszając gracza.

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

Revision #2
Created 2026-10-28 00:48:07 UTC by Tomasz Konieczny
Updated 2026-10-28 22:53:48 UTC by Tomasz Konieczny