

GODOT ENGINE I GDSCRIPT

po swojsku

CZĘŚĆ 2

OD PUSTEGO PROJEKTU
DO PIERWSZEGO PROGRAMU

```
extends Node  
func _ready():  
    print("Witaj w świecie gier!")  
var state = {"score": 0,  
             "lives": 3}  
func _process(delta):  
    state.score += delta * 10  
    if state.score > 100:  
        print("Nowy świat czeka!")  
        get_tree().change_scene_to_file(  
            "res://World.tscn")
```



DARIUSZ GOŁĘBIOWSKI

GODOT ENGINE I GDSCRIPT POSWOJSKU Część 2: Od pustego projektu do pierwszego programu

**PORADNIK POWSTAŁ NA BAZIE
DOŚWIADCZENIA OSOBISTEGO AUTORA
Z PRZYJACIELSKIM WSPARCIEM AI**

NOTA WYDAWCY

UWAGA!

PORADNIK JEST KONTYNUACJĄ CZĘŚCI 1:

Wprowadzenie do świata gier i pierwsze skrypty. Dla lepszego zrozumienia zawartości tej książki, Autor i Wydawnictwo zalecają rozpocząć naukę GODOT Engine od Części 1.

Wszelkie prawa do zawartości tej książki są zastrzeżone. Nieautoryzowane przez autora i/lub wydawnictwo poswojsku.pl rozpowszechnianie całości lub dowolnego fragmentu niniejszej publikacji w jakiegokolwiek postaci jest zabronione. Wykonanie kopii jakiegokolwiek z dostępnych metod (między innymi: elektroniczną, kserograficzną, fotograficzną) spowoduje naruszenie praw autorskich niniejszego dzieła.

OD AUTORA:

Pamiętaj proszę- uszanuj zaangażowanie oraz godziny pracy, które spędziłem nad napisaniem oraz opracowaniem książki:

GODOT ENGINE I GDSCRIPT POSWOJSKU Część 2: Od pustego projektu do pierwszego programu - używaj tylko leganie kupiony ebook.

Wydawnictwo poswojsku.pl:

1. dołożyło wszelkich starań, by zawarte w tej książce informacje były kompletne, poprawne oraz rzetelne,
2. nie ponosi żadnej odpowiedzialności za ewentualne szkody, które mogą wyniknąć z wykorzystania informacji zawartych w tej książce.

Wydawnictwo poswojsku.pl – kontakt:

Strona firmowa: www.poswojsku.pl

e-mail: bok@poswojsku.pl

ul. Paprocka 86, 98–220 Zduńska Wola

ISBN: 978-83-68360-50-9

Copyright © poswojsku.pl 2026

Autor: Gołębiowski Dariusz

W publikacji wykorzystano fonty:

Lato udostępniany na licencji SIL Open Font License,

Roboto, udostępniane na licencji Apache License, Version 2.0.

Informacja o wykorzystaniu sztucznej inteligencji

Podczas przygotowywania publikacji wykorzystano narzędzia AI jako wsparcie w opracowywaniu koncepcji, porządkowaniu materiału, redagowaniu wybranych fragmentów oraz przygotowywaniu części elementów graficznych. Całość została zweryfikowana, uzupełniona i zatwierdzona przez autora, który ostatecznie wykonał: dobór, układ i treść publikacji.

Wybrane ilustracje mają charakter syntetyczny i zostały przygotowane lub zmodyfikowane z wykorzystaniem narzędzi generatywnej sztucznej inteligencji.

Od Wydawnictwa Cyfrowego poswojsku.pl

KILKA SŁÓW O FORMACH ŻEŃSKICH I MĘSKICH

Wspaniała Czytelniczko - Drogi Czytelniku

Jak nowoczesnemu wydawnictwu, zależy nam, aby nasz publikacje były przyjazne dla wszystkich osób. Jednocześnie staramy się, aby nasi autorzy pisali w sposób prosty, naturalny i wygodny w czytaniu.

Dlatego w książce czasami używane są formy męskie, a czasami żeńskie. Wszystkie odnoszą się do każdej osoby czytającej tę publikację, niezależnie od płci, wieku, pochodzenia, stanowiska czy sposobu, w jaki sama siebie określa.

Nie stosujemy konsekwentnie zapisów typu „czytelnik/ czytelniczka”, „nauczyciel/ nauczycielka” itp., ponieważ utrudniają one płynność czytania.

Jeżeli ktoś zauważy, że

- liczba końcówek męskich nie zgadza się z liczbą końcówek żeńskich,
- istnieją inne końcówki, które nie zostały użyte,

z góry przepraszamy za nas i naszych Autorów. Nie jest to działanie zamierzone – po prostu poswojsku jest wydawcą książek ;), a nie firmą księgową, dbającą o zgodność liczby końcówek językowych.

Wszystkim

Wspaniałym Czytelniczkom oraz

Drogim Czytelnikom

życzymy przyjemnej lektury tej książki.

Spis treści

NOTA WYDAWCY.....	2
Wprowadzenie.....	14
Od poznawania kodu do tworzenia zasad.....	15
Kod ma być zrozumiały.....	16
Od warunku do działającej mini-gry.....	18
Błędy są częścią tej podróży.....	19
Nie bój się eksperymentować.....	20
Zaczynamy.....	21
Moduł 3 – Instrukcje warunkowe i pętle w akcji	22
1. Logika programu: „jeżeli wydarzy się to, wykonaj tamto”.....	24

2. Instrukcja if.....	27
3. Kilka możliwych ścieżek.....	29
4. Łączenie warunków.....	32
Operator and.....	33
Operator or.....	33
Operator not.....	33
5. Warunki zagnieżdżone.....	35
6. Pętla a warunek.....	38
7. Pętla for.....	40
8. Pętla while.....	43
9. Pętla nieskończona.....	45
10. break – przerwanie pętli.....	48
11. continue – pominięcie bieżącej iteracji.....	49
12. Pętla a funkcja wywoływana w każdej klatce.....	51
13. Czym jest delta?.....	54

14. Ruch CharacterBody2D.....	57
15. Przygotowanie akcji sterowania.....	60
16. Odliczanie czasu za pomocą delta.....	63
17. Instrukcja match.....	66
18. Praktyka poswojsku: „Klikaj, zanim czas minie!”.....	69
Utworzenie sceny.....	70
Skrypt gry.....	71
Połączenie sygnałów.....	73
19. Jak działa mini-gra?.....	75
20. Zadania dodatkowe.....	78
Zadanie 1. Punkty specjalne.....	79
Zadanie 2. Zmiana tekstu przy końcówce czasu.....	79
Zadanie 3. Poziomy wyniku.....	79
Zadanie 4. Skrócenie czasu gry.....	80
21. Najczęstsze błędy.....	81
Pętla while nie ma warunku zakończenia.....	82
Użycie pętli zamiast _process().....	82

Niepotrzebne mnożenie velocity przez delta.....	82
Brak delta przy samodzielnej zmianie pozycji.....	83
Licznik pokazuje wartość ujemną.....	83
Licznik za wcześnie pokazuje zero.....	83
Kod znajduje się poza funkcją.....	84
Niewłaściwe wcięcia.....	84
Zbyt wiele zagnieżdżonych warunków.....	84
22. Co już potrafisz?.....	85
Moduł 4 – Funkcje, sygnały i zdarzenia.....	88
1. Funkcje – małe zadania programu.....	91
2. Dlaczego dzielimy kod na funkcje?.....	94
3. Czy każdy fragment kodu powinien stać się funkcją?.....	97
4. Parametry funkcji.....	99
5. Wartości domyślne parametrów.....	101
6. Funkcje zwracające wynik.....	103
7. Funkcja czy metoda?.....	106

8. Czym jest zdarzenie?.....	108
9. Czym jest sygnał?.....	111
10. Sygnał wbudowany przycisku.....	114
11. Połączenie sygnału za pomocą kodu.....	117
12. Sprawdzenie i usunięcie połączenia.....	120
13. Tworzenie własnego sygnału.....	122
14. Projekt praktyczny: statek kosmiczny przekazujący punkty.....	124
15. Przygotowanie sceny statki kosmicznej..	126
16. Skrypt statku kosmicznego.....	129
17. Połączenie sygnału body_entered.....	133
18. Przygotowanie gracza.....	136
19. Przygotowanie sceny głównej.....	139
20. Skrypt sceny głównej.....	140

21. Połączenie własnego sygnału statku kosmicznego.....	142
22. Połączenie własnego sygnału za pomocą kodu.....	144
23. Funkcja czy sygnał?.....	146
24. Jeden sygnał i kilku odbiorców.....	147
25. Oczekiwanie na sygnał za pomocą await	148
26. Zadania dodatkowe.....	151
Zadanie 1. Statek kosmiczny o większej wartości.....	151
Zadanie 2. Kilka stateków kosmicznych.....	151
Zadanie 3. Komunikat o dużym wyniku.....	152
Zadanie 4. Sygnał zmiany wyniku.....	152
Zadanie 5. Dźwięk zebrania.....	152
27. Najczęstsze błędy.....	153
Błędna nazwa zmiennej.....	154
Sygnał nie został połączony.....	154

Ten sam sygnał został połączony dwukrotnie.....	154
Sygnał został wyemitowany przed utworzeniem połączenia	155
Niezgodna liczba parametrów.....	155
Gracz nie należy do właściwej grupy.....	156
Statek kosmiczny nie wykrywa gracza.....	156
Statek kosmiczny nalicza punkty kilka razy.....	157
Statek kosmiczny zna zbyt wiele elementów gry.....	157
28. Co już potrafisz?.....	159
Podsumowanie - Twój nowy warsztat.....	162
PODZIĘKOWANIA.....	163
Poznaj inne książki poswojsku.pl.....	164
Szkolenia i Webinary.....	167
Zostańmy w kontakcie!.....	169
Prawa autorskie.....	171
Nazwy handlowe i znaki towarowe.....	172

Wprowadzenie

Pusty projekt ma w sobie coś tajemniczego ;).

Na ekranie nie ma jeszcze bohaterki, przeciwników ani nawet punktów do zdobycia. Nic się nie porusza, a historia – nie istnieje. Nic nie reaguje. Nie istnieją zasady, zwycięstwa ani porażki. Jest jedynie przestrzeń czekająca na pierwszą decyzję twórcy. To brzmi dumnie:

TWÓRCA, czyli TY :).

A potem pojawia się kod.

Jakaś zmienne, warunki, funkcje, a nawet sygnały wysłane przez obiekt.

Z każdą kolejną instrukcją pusty projekt zaczyna się zmieniać. Najpierw nieśmiało reaguje na kliknięcie. Później odlicza czas, porusza postacią, przyznaje punkty i rozpoznaje zdarzenia. W końcu przestaje być pustym projektem. Nawet jeżeli ma błędy, to i tak:

staje się programem.

I właśnie temu poświęcona jest druga część książki „**Godot Engine i GDScript poswojsku**”.

Od poznawania kodu do tworzenia zasad

W pierwszej części rozpoczęliśmy podróż po świecie Godot Engine i GDScriptu. Poznaliśmy środowisko, sceny, węzły, skrypty oraz podstawowe elementy języka. Był to niezbędny fundament.

Teraz pora na następny krok.

Nie będziemy już tylko wydawać programowi pojedynczych poleceń. Nauczymy go wybierać właściwą drogę, powtarzać działania, reagować na zdarzenia i przekazywać informacje pomiędzy elementami gry.

To właśnie w tym momencie programowanie zaczyna przypominać budowanie świata.

Każdy warunek staje się jego prawem:

- *jeżeli gracz dotknie statku – zrób coś, np. przyznaj punkty;*
- *jeżeli czas dobiegnie końca – zrób coś, np. zakończ rozgrywkę;*
- *jeżeli wynik przekroczy określoną wartość – wyświetl komunikat.*

Pętle pozwalają powtarzać działania.

Funkcje porządkują kod i nadają mu czytelną strukturę.

Sygnały umożliwiają obiektom komunikowanie się bez tworzenia pomiędzy nimi niepotrzebnych zależności.

To już nie jest zbiór przypadkowych instrukcji.

To system. Twój system!

Kod ma być zrozumiały

Programowanie bywa przedstawiane jako dziedzina dostępna wyłącznie dla osób obdarzonych niezwykłym talentem matematycznym albo tajemną wiedzą techniczną.

Nie zgadzam się z tym.

Kod jest sposobem opisywania zasad. Trzeba poznać jego słowa, składnię i logikę, ale najważniejsze pozostaje rozumienie tego, **co chcemy osiągnąć**.

Dlatego nie będziemy uczyć się GDScriptu poprzez bezrefleksyjne przepisywanie kolejnych fragmentów kodu. Każde nowe zagadnienie rozłożymy na mniejsze części. Sprawdzimy, jak działa, po co zostało stworzone i kiedy warto je wykorzystać.

Nie chodzi o to, aby zapamiętać każdą instrukcję.

Chodzi o to, aby potrafić zadać właściwe pytania:

- Co powinno wydarzyć się w programie?
- Jaki warunek musi zostać spełniony?
- Które działanie należy powtórzyć?
- Czy ten fragment kodu warto wydzielić do funkcji?
- Który obiekt powinien poinformować pozostałe o zdarzeniu?
- Dlaczego program zachowuje się inaczej, niż planowaliśmy?

Umiejętność odpowiadania na takie pytania jest znacznie cenniejsza niż znajomość dziesiątek poleceń na pamięć.

Od warunku do działającej mini-gry

Zacznijmy od instrukcji warunkowych. Dzięki nim program przestanie wykonywać wszystkie polecenia bez zastanowienia i zacznie podejmować decyzje.

Następnie zajmiemy się pętlami. Poznamy ich możliwości, ale również związane z nimi zagrożenia. Jedna źle przygotowana pętla może przecież sprawić, że gra przestanie odpowiadać.

Wyjaśnimy także, czym jest Δt i dlaczego ma aż tak ogromne znaczenie dla płynnego ruchu oraz odmierzania czasu.

Przygotujemy sterowanie obiektem `CharacterBody2D`, akcje wejściowe i licznik niezależny od liczby klatek wyświetlanych przez komputer.

Zdobytą wiedzę wykorzystamy w praktyce, tworząc mini-grę „**Klikaj, zanim czas minie!**”. Nie będzie to przykład istniejący wyłącznie po to, aby pokazać kilka linijek kodu. Powstanie działający projekt z zasadami, punktami, upływającym czasem i możliwością dalszej rozbudowy.

Później przejdziemy do funkcji, zdarzeń i sygnałów.
Zbudujemy scenę ze statkiem kosmicznym, który będzie przekazywał graczowi punkty. Nauczymy obiekty współpracować, ale jednocześnie zadbamy o to, aby każdy z nich miał jasno określone zadanie.

Krok po kroku pusty projekt zacznie żyć.

Błędy są częścią tej podróży

Podczas pracy coś czasami nie zadziała.

Sygnał nie zostanie połączony.

Nazwa zmiennej okaże się nieprawidłowa.

Wcięcie znajdzie się w niewłaściwym miejscu.

Licznik pokaże wartość ujemną, a statek kosmiczny naliczy punkty kilka razy.

To normalne.

Błąd wcale nie oznacza, że Ty nie nadajesz się do programowania.

Oznacza jedynie, że program wykonał otrzymane instrukcje inaczej, niż oczekiwałeś.

Dlatego w książce znajdziesz omówienia częstych problemów oraz ich możliwych przyczyn. Nauczysz się nie tylko tworzyć kod, ale również obserwować jego działanie, szukać źródła problemu i świadomie wprowadzać poprawki.

Programista nie jest osobą, która nigdy się nie myli.

Programista jest osobą, która potrafi znaleźć drogę od błędu do rozwiązania.

Nie bój się eksperymentować

Przykłady przedstawione w książce są punktem wyjścia, a nie granicą Twoich możliwości.

Zmieniaj wartości. Dodawaj własne warunki. Przetwarzaj obiekty. Rozbudowuj funkcje. Sprawdzaj, co się wydarzy, gdy połączysz jeden sygnał z kilkoma odbiorcami albo zmodyfikujesz zasady przyznawania punktów.

Jeżeli projekt przestanie działać — spróbuj ustalić dlaczego.

Jeżeli zadziała inaczej, niż planowałeś — zastanów się, czy przypadkiem nie odkryłeś nowej mechaniki.

Właśnie w ten sposób rozwija się prawdziwe zrozumienie programowania: nie podczas biernego czytania, lecz wtedy, gdy samodzielnie zadajesz pytania i sprawdzasz odpowiedzi w działającym projekcie.

Zaczynamy

Przed nami warunki, pętle, funkcje, sygnały, zdarzenia, ruch i odmierzanie czasu. Będziemy pisać kod, budować sceny, łączyć ich elementy i poprawiać błędy.

Nie musisz wiedzieć wszystkiego od razu.

Wystarczy, że wykonasz kolejny krok.

Otwórz Godot Engine. Utwórz projekt. Dodaj pierwszą scenę i napisz pierwszą instrukcję.

Pusty projekt już na Ciebie czeka.

Nadajmy mu zasady.

Wprawmy go w ruch.

Napiszmy pierwszy prawdziwy program.

Praktycznie. Zrozumiale. Poswojsku.

Moduł 3 – Instrukcje warunkowe i pętle w akcji

```
func _check_magic_impulse() -> void:
    if not Input.is_action_just_pressed(&"ui_accept"):
        return
    if magic_impulses_remaining <= 0:
        $UI/Message.text = "Moc Astilusa musi się odrodzić." if _is_polish() else "Astilus's power must recover."
        return
    var player := $Astilus as HorizontalLiftReturnPlayer
    if player == null or not player.start_magic_cast(MAGIC_CAST_DURATION):
        $UI/Message.text = "Astilus musi się zatrzymać, aby użyć mocy." if _is_polish() else "Astilus must stop to use his power."
        return
    magic_impulses_remaining -= 1
    magic_effect_origin = player.global_position
    magic_effect_remaining = MAGIC_EFFECT_DURATION
    var repelled_harpies: int = _repel_nearby_harpies(magic_effect_origin)
    var destroyed_rocks: int = _destroy_nearby_rocks(magic_effect_origin)
    $UI/Message.text = _magic_result_message(repelled_harpies, destroyed_rocks)
    _update_ui()
    queue_redraw()
```

Cele modułu:

- Nauczyć się kontrolować przebieg programu,
- Podejmować decyzje,
- Powtarzać działania,
- Reagować na upływ czasu,
- Sterować stanem prostej gry.

Po ukończeniu tego modułu będziesz potrafić:

- budować warunki za pomocą `if`, `elif` i `else`;
- łączyć warunki operatorami logicznymi;
- korzystać z pętli `for` i `while`;
- przerywać pętlę albo pomijać jej wybrane wykonania;
- odróżniać pętlę od funkcji wywoływanej w każdej klatce;
- rozumieć podstawową rolę parametru `delta`;
- sterować prostym ruchem postaci;
- przygotować licznik czasu;
- wykorzystać instrukcję `match`;
- zbudować mini-gry opartą na czasie i punktach.

1. Logika programu: „jeżeli wydarzy się to, wykonaj tamto”

```
if stageFlag == 0:  
    >|    var stage = 0  
if stageFlag == 0:  
    >|    var stage = 0
```

Silnik Godot Engine wykonuje instrukcje zgodnie z przygotowanymi (w kodzie) przez programistkę zasadami.

W grach i aplikacjach wiele takich zasad ma postać warunków:

- jeżeli gracz ma właściwy klucz, otwórz określone drzwi;
- jeżeli punkty życia spadną do zera, zakończ grę;
- jeżeli czas się skończy, zablokuj możliwość zdobywania punktów;
- jeżeli użytkownik poda poprawne hasło, pozwól mu przejść dalej;
- jeżeli przeciwnik zobaczy gracza, rozpocznij pościg;
- jeżeli postać stoi na podłożu, pozwól jej skoczyć.

Warunek odpowiada na pytanie, którego wynikiem jest wartość logiczna:

true

albo:

false

Przykładowo:

health <= 0

oznacza pytanie:

**Czy wartość zmiennej health jest mniejsza
lub równa zero?**

***Jeżeli odpowiedź brzmi „tak”,
wynikiem warunku jest true.***

2. Instrukcja if

Najprostsza instrukcja warunkowa wygląda następująco:

if warunek:

 instrukcja

Czyli jeżeli **warunek** da logiczną wartość *TRUE*, wówczas wykonywana jest **instrukcja**

Przykład:

```
func check_health(health: int) -> void:
```

```
    if health <= 0:
```

```
        print("Koniec gry!")
```

Komunikat zostanie wyświetlony tylko wtedy, gdy wartość `health` będzie mniejsza lub równa zero.

Kod należący do warunku musi być prawidłowo wcięty, czyli ustawiony od lewej krawędzi, a pozwól iż przypomnę, że 4 spacje w przypadku takich języków kodowania jak Python czy używany w Godot Engine GDScript, niekoniecznie równy jest spacji. Warto o tym pamiętać, bo inaczej możesz się spotkać z sytuacją, w której wizualnie – wszystko wygląda o.k., ale program „wyrzuca” błąd. Poniżej – zasy wcięcia kodu dla omawianej instrukcji if:

```
if health <= 0:
```

```
    print("Ta instrukcja należy do warunku.")
```

```
print("Ta instrukcja zostanie wykonana niezależnie od  
warunku.")
```

Wcięcia nie służą wyłącznie poprawieniu wyglądu kodu. W GDScript są częścią jego składni.

3. Kilka możliwych ścieżek

```
func _update_interaction_hint() -> void:
    var polish: bool = _is_polish()
    var nearest := _nearest_container()
    if nearest != null:
        $UI/Interaction.text = "SPACJA – zbadaj skrytkę"
        if polish else "SPACE – examine cache"
    elif opened_containers >= CACHE_ORDER.size():
        $UI/Interaction.text = "ARSENAL ZABEZPIECZONY –
        IDŹ DO WINDY" if polish else "ARSENAL SECURED –
        REACH THE LIFT"
    else:
        $UI/Interaction.text = "SZUKAJ KOLEJNEGO ZNAKU
        MERLINA" if polish else "FIND THE NEXT SIGN OF
        MERLIN"
```

Jeżeli program powinien wybrać jedną z kilku możliwości, możemy użyć

if,
elif,
else:

```
func show_health_status(health: int) -> void:
```

```
    if health <= 0:
```

```
        print("Koniec gry!")
```

```
    elif health < 20:
```

```
        print("Uważaj, masz mało punktów życia!")
```

```
    elif health < 60:
```

```
        print("Postać jest ranna.")
```

```
    else:
```

```
        print("Postać jest w dobrej formie.")
```

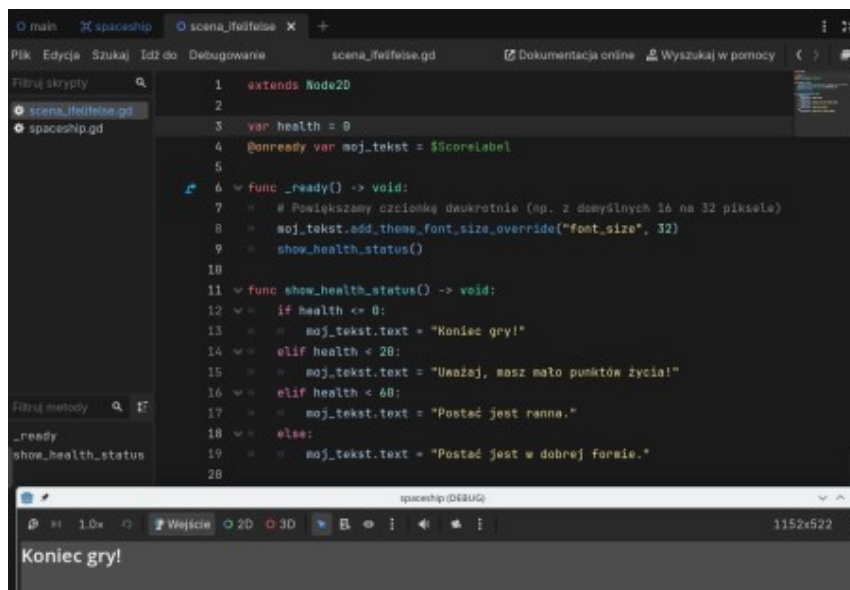
Godot sprawdza warunki od góry.

Jeżeli pierwszy warunek jest prawdziwy, wykonuje należący do niego blok i pomija pozostałe części tej konstrukcji.

Dla wartości:

```
health = 0
```

prawdziwe są zarówno warunki `health <= 0`, jak i `health < 20`.



```
1 extends Node2D
2
3 var health = 0
4 @onready var moj_tekst = $ScoreLabel
5
6 func _ready() -> void:
7     # Powiększony czcionkę dwukrotnie (np. z domyślnych 16 na 32 piksele)
8     moj_tekst.add_theme_font_size_override("font_size", 32)
9     show_health_status()
10
11 func show_health_status() -> void:
12     if health <= 0:
13         moj_tekst.text = "Koniec gry!"
14     elif health < 20:
15         moj_tekst.text = "Uważaj, masz mało punktów życia!"
16     elif health < 40:
17         moj_tekst.text = "Postać jest ranna."
18     else:
19         moj_tekst.text = "Postać jest w dobrej formie."
20
```

Zostanie jednak wykonany tylko pierwszy pasujący blok:

`print("Koniec gry!")`

Dlatego kolejność warunków ma znaczenie. Zwykle zaczynamy od przypadków najbardziej szczegółowych albo najważniejszych.

4. Łączenie warunków

Operatory:

and

or

not

Za pomocą operatorów and, or i not możemy budować bardziej rozbudowane warunki.

Operator and

Obie części warunku muszą być prawdziwe:

```
func can_open_door(has_key: bool, door_is_locked: bool) -> bool:
```

```
    return has_key and door_is_locked
```

Operator or

Wystarczy, że przynajmniej jedna część warunku będzie prawdziwa:

```
func can_enter(has_ticket: bool, is_admin: bool) -> bool:
```

```
    return has_ticket or is_admin
```

Operator not

Operator not odwraca wartość logiczną:

```
func show_door_status(door_is_open: bool) -> void:
```

```
    if not door_is_open:
```

```
        print("Drzwi są zamknięte.")
```

Jeżeli `door_is_open` ma wartość `false`, wyrażenie:
`not door_is_open`
będzie miało wartość `true`.

Witaj w gronie twórców. Do dzieła!

**Serdecznie dziękujemy za poświęcenie
Twojego czasu na zapoznanie się z
początkiem drugiej części poradnika dla
osób szukających wiedzy w obszarze gier, a
w szczególności ich tworzenia w Silnku
Godot Engine i języku kodowania GDScript.
Zapraszamy także do nabycia pełnej wersji
tego ebooka. Więcej informacji znajdziesz
na stronie wydawnictwa**

www.poswojsku.pl

PODZIĘKOWANIA

Serdecznie dziękuję za zapoznanie się z zawartością mojego poradnika. Mam nadzieję, że przyniósł on Tobie Droga Czytelniczko/ Wspaniały Czytelniku - dużo wiedzy oraz zrozumienia odnośnie sposobów efektywnego wykorzystania narzędzi związanych z AI. Zapraszam Ciebie także do kolejnych części poradnika „AI w edukacji”.

W poradniku wykorzystano:

- własne materiały graficzne,
- prace AI,
- cliparty z programu LibreOffice na licencji CCO.

DZIĘKUJĘ ZA UWAGĘ

Autor poradnika: DARIUSZ GOŁĘBIEWSKI

Poznaj inne książki poswojsku.pl

Jeśli spodobał się Tobie ten poradnik i chcesz dalej rozwijać swoją wiedzę o cyberbezpieczeństwie, technologii i świadomym korzystaniu z internetu, oto moje inne książki, które mogą w tym pomóc. Każda z nich powstała z myślą o osobach, które szukają praktycznych wskazówek, konkretnych przykładów i języka zrozumiałego dla każdego.

Twoje bezpieczeństwo w świecie cyber i sztucznej inteligencji – Część 1: Wprowadzenie

Kompleksowy wstęp do tematyki ochrony danych, prywatności i bezpiecznego korzystania z sieci. To książka dla tych, którzy chcą szybko zrozumieć, na czym polegają najważniejsze zagrożenia i jak zacząć się przed nimi skutecznie bronić – bez skomplikowanego żargonu. Znajdziesz tu przykłady z życia i proste instrukcje krok po kroku.

Twoje bezpieczeństwo w świecie cyber i sztucznej inteligencji – Część 2: Cyberhigiena

Praktyczny przewodnik po codziennych nawykach, które realnie zwiększają Twoje bezpieczeństwo. Dowiesz się, jak tworzyć silne hasła, chronić urządzenia, wykrywać próby

oszustw i przygotować swoją rodzinę na zagrożenia cyfrowe. To pozycja obowiązkowa, jeśli chcesz, żeby cyberhigiena była naturalną częścią Twojego życia.

Twoje bezpieczeństwo w świecie cyber i sztucznej inteligencji – Część 3: Dziecko i Ty

Poradnik stworzony specjalnie dla rodziców i opiekunów, którzy chcą wprowadzać dzieci w cyfrowy świat z rozsądkiem i spokojem. Znajdziesz tu nie tylko zasady i rekomendacje, ale też gotowe sposoby rozmowy o bezpieczeństwie i budowania zaufania. Ta książka pomoże Ci chronić najmłodszych bez straszenia i nadmiernej kontroli.

AI w edukacji – Część 1: Praktyczny poradnik nie tylko dla nauczycieli

Sztuczna inteligencja to nie tylko moda, ale i realne narzędzie, które może usprawnić naukę i pracę. W tej książce pokazuję, jak korzystać z AI w prosty sposób – od generowania treści, przez wspieranie kreatywności, po automatyzację codziennych zadań. Idealna dla edukatorek/edukatorów i osób, które chcą poznać podstawy nowoczesnych technologii.

AI w edukacji – Część 2: Praktyczne pomysły na kreatywną edukację

Kontynuacja pierwszej części – pełna inspiracji, scenariuszy zajęć i ćwiczeń. Dowiesz się, jak prowadzić warsztaty i lekcje, które łączą AI z rozwojem kompetencji cyfrowych, logicznego myślenia i twórczego podejścia do nauki. Świetna pozycja dla wszystkich, którzy szukają konkretnych narzędzi i gotowych rozwiązań.

Stwórz Grę Mobilną

Praktyczny przewodnik dla osób, które marzą o stworzeniu własnej gry na smartfon. Od absolutnych podstaw programowania w JavaScript i React Native, przez projektowanie rozgrywki, aż po publikację gry. Jeśli chcesz uczyć się kodowania w sposób ciekawy i namacalny, to ta książka będzie Twoim drogowskazem.

Saga CyberJestestwa

Powieść science fiction dla tych, którzy chcą oderwać się od codzienności i zanurzyć w refleksyjnej historii o sensie istnienia, wolności i relacjach w obliczu zmian. To opowieść o ludziach i technologii, o wyborach i konsekwencjach – dla

miłośniczek/miłośników literatury, którzy cenią głębsze przesłanie i oryginalny klimat.

Wszystkie moje ebooki możesz nabyć na:
stronie wydawnictwa cyfrowego **poswojsku.pl**

Szkolenia i Webinary

Jeśli chcesz pogłębić wiedzę, zdobyć praktyczne umiejętności i od razu wprowadzić je w życie – zapraszam Cię na moje szkolenia i webinary. Każde z nich zostało przygotowane tak, aby w przystępny sposób przekazać konkretne rozwiązania i pomóc Ci działać od zaraz.

Bezpieczni w sieci – Jak chronić siebie i rodzinę przed cyberzagrożeniami

Szkolenie, w którym krok po kroku omawiam zagrożenia najczęściej dotyczące rodziny – od fałszywych wiadomości i wyłudzeń danych, po ochronę urządzeń

domowych i zabezpieczanie dzieci w internecie. Idealne dla rodziców, opiekunów i osób, które chcą działać świadomie.

Cyberbezpieczeństwo dla małych organizacji i firm

Praktyczny warsztat dla właścicieli firm, fundacji i urzędów, którzy chcą nauczyć się chronić dane pracowników i klientów bez kosztownych wdrożeń. Omawiam darmowe narzędzia, procedury bezpieczeństwa i sposoby budowania kultury cyberhigieny w zespole.

AI w życiu codziennym – od podstaw

Webinar pokazujący, jak sztuczna inteligencja może ułatwić pracę, naukę i organizację codziennych spraw. Dowiesz się, jak korzystać z AI do tworzenia treści, automatyzacji zadań i rozwijania nowych umiejętności – nawet jeśli nie masz doświadczenia technicznego.

Szyfrowanie danych – dyski, pliki, poczta

Szkolenie wprowadzające w świat szyfrowania, pokazujące krok po kroku, jak zabezpieczyć swoje dane prywatne i firmowe za pomocą bezpłatnych narzędzi. Idealne dla każdego, kto chce uniknąć utraty poufnych informacji.

Cyfrowe bezpieczeństwo dziecka – jak mądrze wspierać młodych użytkowników internetu

Spotkanie dla rodziców i nauczycieli, którzy chcą dowiedzieć się, jak rozmawiać z dziećmi o zagrożeniach online, jak ustawiać kontrolę rodzicielską i jak budować zaufanie w cyfrowym świecie.

Aktualne terminy szkoleń i webinarów znajdziesz na

stronie: poswojsku.pl

a webinarów: www.poswojsku.com.pl

Zapraszam również do kontaktu – chętnie pomogę dobrać szkolenie odpowiednie dla Twoich potrzeb.

Zostańmy w kontakcie!

Jeśli chcesz być na bieżąco z nowymi książkami, szkoleniami i inspiracjami o cyberbezpieczeństwie, technologii i AI – zapraszam Cię do obserwowania moich profili. Dzięki temu nie przegapisz premier, promocji i wartościowych materiałów które tworzę: często, prosto, przystępnie i z humorem.

- ♦ Strona internetowa 📌 poswojsku.pl
- ♦ Facebook 📌 facebook.com/poswojsku
- ♦ YouTube 📌 youtube.com/@poswojsku
- ♦ LinkedIn 📌 linkedin.com/in/golebiowski-dariusz
- ♦ Instagram 📌 instagram.com/poswojsku
- ♦ Threads 📌 threads.com/@poswojsku
- ♦ TikTok 📌 tiktok.com/@astilus
- ♦ Amazon Author Page 📌
amazon.com/author/dariuszgolebiowski
- ♦ Goodreads 📌 goodreads.com/dariuszgolebiowski

**Proszę, dołącz do mnie – razem budujmy
bezpieczniejszy i bardziej świadomy świat cyfrowy!**

Więcej informacji - www.poswojsku.pl

Prawa autorskie

Copyright © 2026 Dariusz Gołębiowski

Wszelkie prawa zastrzeżone.

Żadna część niniejszej publikacji nie może być kopiowana, rozpowszechniana, udostępniana ani wykorzystywana w inny sposób bez uprzedniej zgody uprawnionego, z wyjątkiem przypadków dozwolonych przez obowiązujące przepisy prawa, w szczególności w ramach dozwolonego cytatu.

Nazwy narzędzi, modeli i usług sztucznej inteligencji zostały wymienione wyłącznie w celach informacyjnych, edukacyjnych, porównawczych i ilustracyjnych.

Publikacja ma charakter niezależny. Nie jest oficjalnym materiałem żadnego z producentów ani dostawców opisanych narzędzi i nie stanowi przejawu ich współpracy, rekomendacji, zatwierdzenia, patronatu ani sponsoringu.

Nazwy handlowe i znaki towarowe

Wszystkie wymienione w publikacji nazwy firm, organizacji, produktów, modeli, usług, platform oraz powiązane z nimi logotypy mogą być znakami towarowymi, zastrzeżonymi znakami towarowymi albo nazwami handlowymi należącymi do odpowiednich właścicieli.

W szczególności:

- **Godot, Godot Engine** oraz powiązane z nimi logotypy są znakami towarowymi należącymi do **Godot Foundation**.
- **GDScript** jest nazwą języka programowania stworzonego i rozwijanego w ramach otwartoźródłowego projektu Godot Engine.

Pozostałe nazwy firm, organizacji, produktów, modeli i usług wymienione w publikacji mogą stanowić znaki towarowe, zastrzeżone znaki towarowe albo nazwy handlowe należące do odpowiednich właścicieli.

Ich użycie w niniejszym poradniku służy wyłącznie identyfikacji i omówieniu opisywanych rozwiązań.

Autor i wydawca nie roszczą sobie żadnych praw do cudzych znaków, nazw, logotypów ani elementów identyfikacji wizualnej.

Informacje dotyczące narzędzi i usług przedstawiają stan wiedzy na dzień opracowania publikacji. Nazwy, właściciele, funkcjonalności, warunki korzystania oraz dostępność poszczególnych usług mogą ulec zmianie.