



MOJA GRA

Czyli projekt z podstaw programowania



18 STYCZNIA 2025

WIKTORIA ZAJĄC
44961



Wstęp

Tworzenie mojej pierwszej w życiu gry, było dla mnie niezwykłym i trudnym wyzwaniem. Trudności jakie napotkałam podczas tworzenie, starałam się analizować i poprawiać. Największym wyzwaniem według mnie stanowiło stworzenie sterowania w 3D. Dodatkowo postanowiłam wybrać inny silnik niż Unreal Engine. Jest to silnik Godot Engine, w którym programowałam językiem GD Script (podobnym do Pythona). Dzięki temu pomysłowi oraz umiejętnościom, myślę, że ta gra jak na mnie – osobę początkującą, która próbowała swoich sił w tworzeniu gry – jest grą nieco zaawansowaną. Miło mi jest teraz prezentować mój czasochłonny progres, w który włożyłam nie tylko programowanie, ale i zaplanowanie gatunku gry. Postanowiłam w tej grze połączyć zwiedzanie wraz z rozwiązywaniem łamigłówek, oraz moim celem było przygotowanie postaci motyli w programie Blender, wraz z animacjami. Oto ukazuję, jak krok po kroku stworzyłam grę od zera:

1. Pobranie Godota (wersji najnowszej)

Postanowiłam, że na dobry początek zaprogramuję i stworzę grę w Godot Engine 4, ponieważ ma przyjazny interfejs i język programowania dla mnie intuicyjny i skuteczny jak na mój poziom początkującego dewelopera.

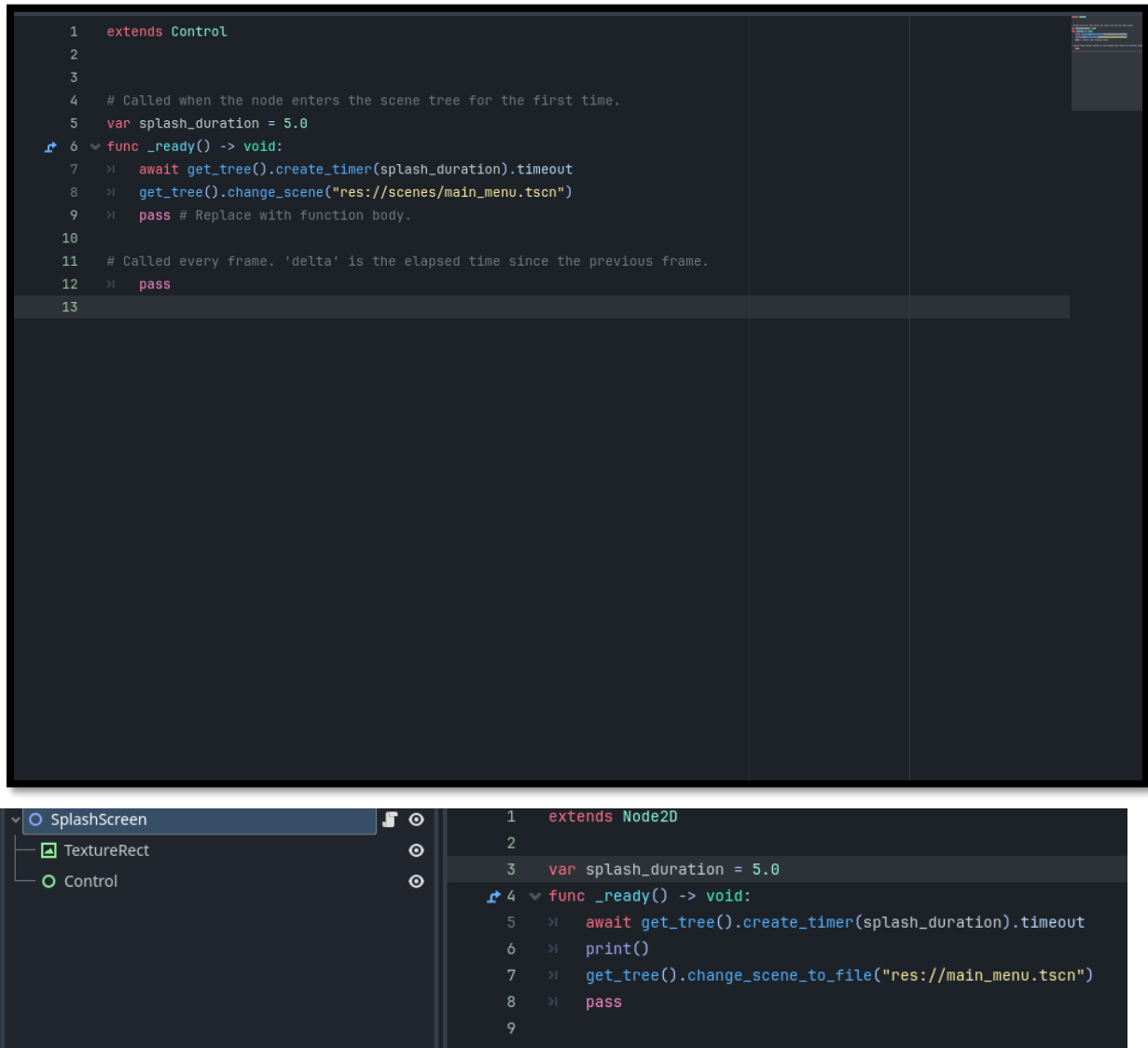
2. Stworzenie nowego pliku w Godot

Nazwałam ten plik nazwą gry, jaką jest: Misty Butterfly Adventure. Przygotowuję także zasoby z Blendera, strony internetowej z modelami 3D oraz własne zdjęcia (w tym przypadku do puzzli – swój rysunek cyfrowy) Puzzle tworzę dzieląc tę fotografię na części w programie, które później zapisuję jako plik zip, z częściami puzzli. Motyle tworzę w blenderze – osobno każdy gatunek, w tym przypadku Aglais Io, Morpho oraz Urania Ripheus. Używam gotowych tekstur ze wzorami poszczególnych motyli. Następnie robię dla każdego osobne animacje poprzez shape keys oraz action editor. Dzięki temu moje animacje są zarówno idle, chodzenia jak i latania, a nawet ataku. Dodatkowo wszystkie te animacje zawierają się w jednej postaci motyla. Natomiast ze strony z modelami 3D – wybieram różne dodatki do mojej sceny gry, aby była urozmaicona.

3. Stworzenie splash screena

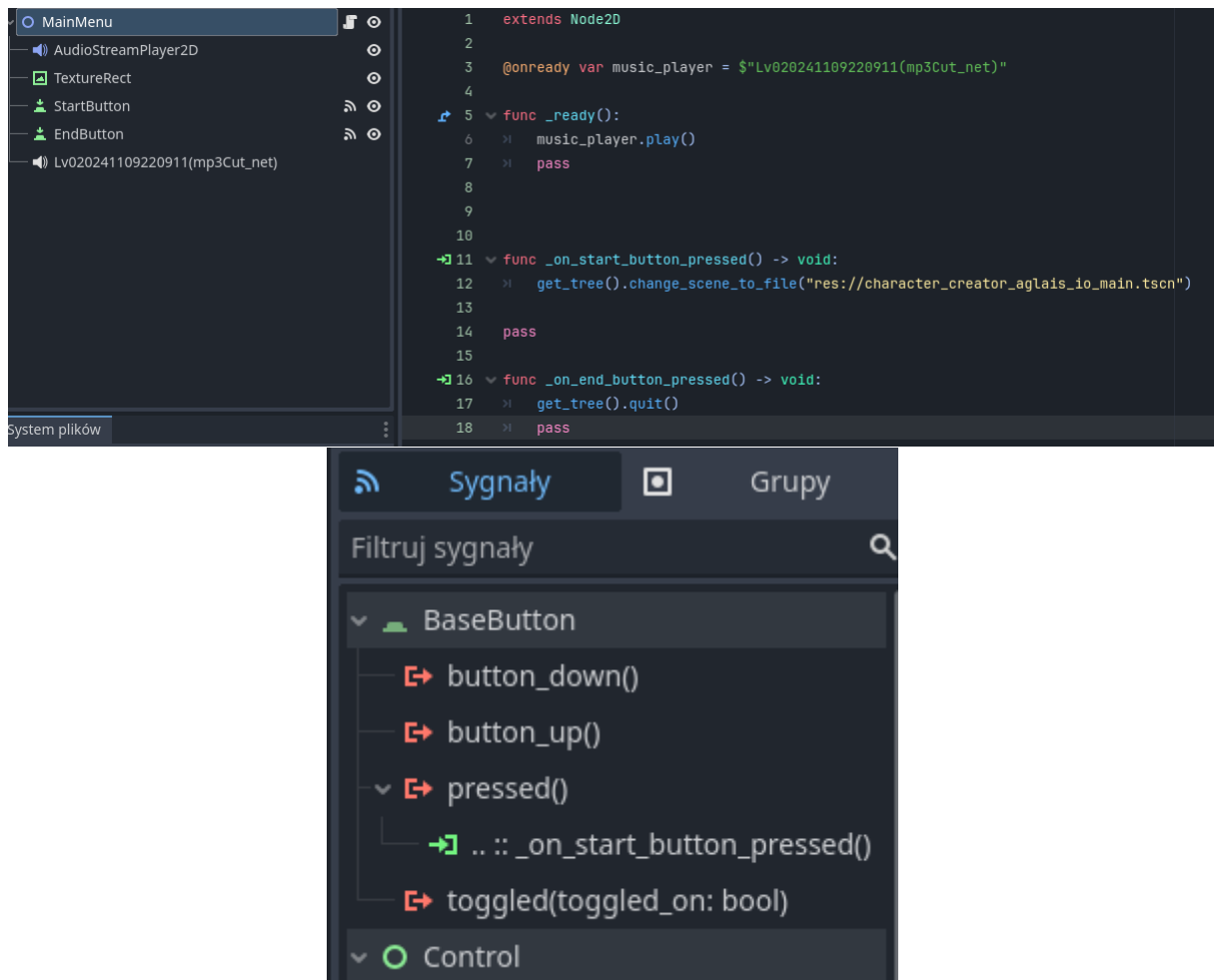
Tworzę nową scenę w Godocie (jest to scena 2D). Nazywa się typem Node2D i dołączam do niego korzenie, które dziedziczą po Node. Te korzenie to texturerec – czyli tło splash screena oraz control. Do splash screena dodaję skrypt, aby ustawić czas (okres) pojawiania się splash screena na początku gry (w tym przypadku jest to 5 sekund). Następnie dodaję funkcję, która odpowiada za koniec (upływ)

tego czasu pojawienia się splash screena i następnie przechodząc do pliku sceny menu, który następnie stworzyłam. Na końcu sprawdzam, czy mój splash screen działa.



4. Stworzenie głównego menu

Dodając nową scenę 2D w Godocie – myślę nad zaimportowaniem zdjęcia przyciągającego uwagę (jest to zdjęcie motyla Rusałki Pawik). Następnie dodaję przyciski (Start Button i End Button) odpowiadające rozpoczęciu i wyjściu z gry. Dodaję także muzykę, aby stał się klimat tej gry. W skrypcie Main Menu: uruchamiam poprzez `@onready` – muzykę i podaję jej źródło, które zaimportowałam do Godota. Następnie wywołuję funkcję `ready`, która służy jako odtwarzacz muzyki. Następnie do podanych przycisków dodaję funkcję zarówno przejścia do następnej sceny (jaką jest kreator postaci, wybór motyla) oraz wyjście z gry. Co ważne węzły przycisków muszą być `connected` – połączone, a nie `rozłączone`, aby mogły działać. Musi być ustawiony dla nich sygnał.



5. Tworzenie kreatora postaci

Dodaję tym razem do sceny 3D i 2D, obiekty, przyciski. W scenie 2D:

*Tworzę przycisk back to menu (oczywiście z sygnałem i skryptem)

*Przycisk Begin adventure! (przejścia do świata konkretnym, wybranym motylem)

*Możliwość wpisania nazwy użytkownika, która nawet po nie wpisaniu też może wprowadzić do gry

W scenie 3D zaś:

*Importuję własną scenę z Blendera, aby stanowiło urok tej gry (rośliny, rzeźby, skały)

*Importuję model wody i jej animację aby nadać trochę życia

*Ustawiam kamerę i światło w kierunku motyla, który importuję na skałę (złudzenie siedzenia motyla na skałe)

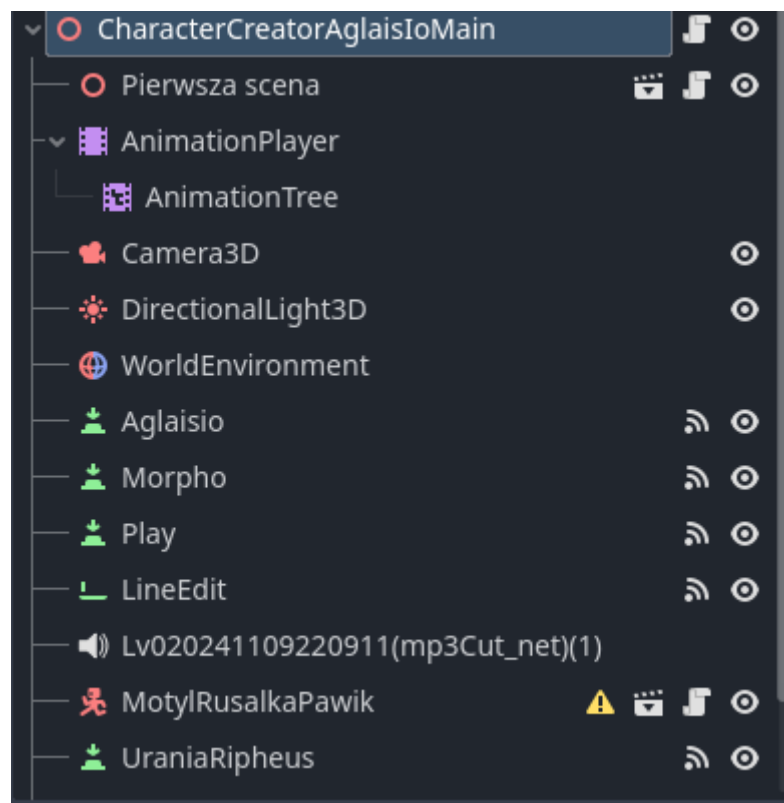
*Uwaga: dla każdego motyla robię osobną scenę z kreatora po kliknięciu przycisku, któregoś z tych pięknych stworzeń

* Przypisuję wkrótce skrypt do całej sceny kreatora (osobno Aglais io, Morpho i Urania Ripheus) i dodaję zaimportowany dźwięk (śpiew ptaków, cykanie świerszczy) – ustawiam go na loop – aby mógł w kółko w nieskończoność być włączony. Do każdego przycisku przypisuję scenę z motylem (mają być

różne), a po tym jak wybiorę np. motyla Aglais io i wejdę do gry, to świat pojawi mi się właśnie z tym

```
1 extends Node3D
2
3 @onready var music_player = $*Lv020241109220911(mp3Cut_net)(1)
4
5
6
7 func _ready():
8     >| music_player.play()
9     >|
10
11
12 func _on_aglaisio_pressed() -> void:
13     >| get_tree().change_scene_to_file("res://character_creator_aglais_io_main.tscn")
14     >|
15     >| pass
16
17
18 func _on_morpho_pressed() -> void:
19     >| get_tree().change_scene_to_file("res://morpho.tscn")
20     >| pass
21
22
23
24 func _on_play_pressed() -> void:
25     >| get_tree().change_scene_to_file("res://demo/ŚwiatRusalkaPawik.tscn")
26     >| pass
27
28
29 func _on_uranio_ripheus_pressed() -> void:
30     >| get_tree().change_scene_to_file("res://uranio_ripheus.tscn")
31     >| pass
32
33
34 func _on_line_edit_renamed() -> void:
35     >| pass
36
37
38 func _on_button_pressed() -> void:
39     >| get_tree().change_scene_to_file("res://main_menu.tscn")
40     >| pass |
41
```

motylem.



6. Stworzenie otwartego świata (osobno dla Aglais Io, Morpho i Urania Ripheus)

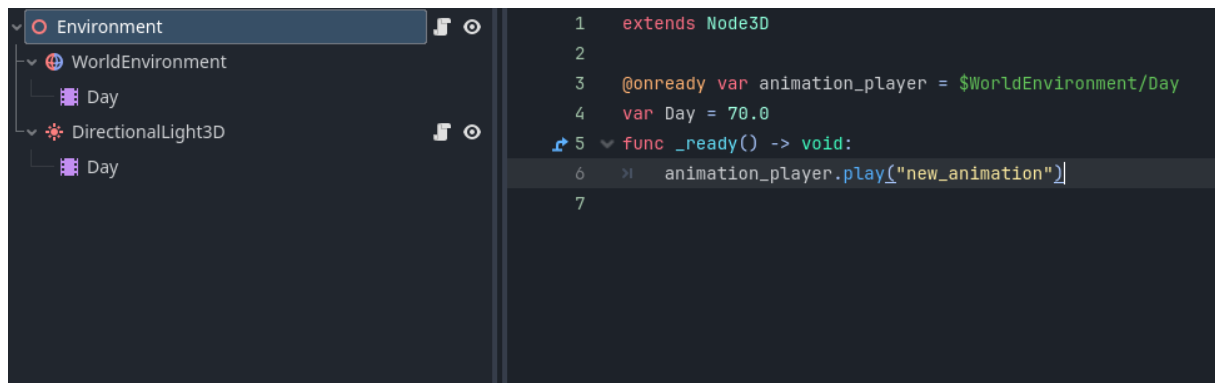
Wkrótce nadszedł moment najtrudniejszy do stworzenia, a zarazem najbardziej istotny. Na podstawie opcji terrain – dodaję teren gry z możliwością kolizji, aby mogły się stykać ze sobą, oddziaływać. Dodaję też obiekty (latającą wyspę z możliwością kolizji, skały, drzewa, wodę, rośliny). Do głównego skryptu sceny dodaję muzykę oraz funkcję do niej. Dodaję też możliwość kliknięcia na przycisk ESC oraz przejścia prawym przyciskiem myszy do łamigłówki jaką będzie układanie puzzli.

W user interface dodaję instrukcję obsługi gry, a w scenie 2D dodaję napisy, w jaki sposób przejść do łamigłówki oraz jak zwiedzać otwarty świat. Dodałam także opcję renderowania serwera, aby jeszcze dla dociekliwych mogła być widoczna siatka poszczególnych brył.

Wkrótce dodaję postać motyla, który był najbardziej czasochłonnym procesem w stworzeniu sterowania. Dodałam także możliwość latania, czyli grawitację na -9.8, animację, funkcję do animacji (kiedy daną animację ma wykorzystać), szybkość poruszania, atak (poruszanie skrzydłami przy kliknięciu lewym przyciskiem myszy), a także ruch kamery wraz z postacią.

Do środowiska Environement – dodaję cykl nocy i dnia na podstawie animacji cyklicznej słońca

```
1  @tool
2  extends Node
3  @onready var music_player = $Video250114122628
4  func _ready() -> void:
5      music_player.play() # Rozpoczyna odtwarzanie
6
7
8      if not Engine.is_editor_hint() and has_node("UI"):
9          $UI.player = $AnimacjewjednejpostaciRusalkaPawik
10
11
12  func _on_sketchfab_scene_ready() -> void:
13      pass
14
15  func _input(event):|
16      if event is InputEventMouseButton:
17          if event.button_index == MOUSE_BUTTON_RIGHT and event.pressed:
18              get_tree().change_scene_to_file("res://Match.tscn")
19
```



7. Stworzenie łamigłówki (puzzle)

Każdy element puzzle zapisuję w osobnym node2D jako korzeń każdego z nich. Następnie każdy z tych elementów ustawiam na Vector2, aby mogły oddziaływać ze sobą w scenie 2D. Następnie w każdym Node dodaję skrypt, czyli dany, osobny element jest przesuwany czy nie. Ustawiam także odpowiednią pozycję puzzle (gdzie powinien się znajdować). Do gry dodaję też przycisk end brain game, aby powrócić do świata.

```
1 extends Sprite2D
2
3 var correct_position: Vector2 = Vector2.ZERO |
4 func set_correct_position(target_position: Vector2) -> void:
5     correct_position = target_position
6
```

Zakończenie

Tworzenie gry w godot engine było dla mnie ogromnym wyzwaniem. Dodatkowo moim marzeniem było nawet stworzyć chociaż najprostszą na świecie grę. Sprawilo mi to ogromną przyjemność, ale i walkę z przeciwnościami, które wprowadzały mnie do bezradności. Na szczęście wybrnęłam z tego, a gra jest już gotowa do wypróbowania.