

PEMROGRAMAN GAME

OLEH: FERDI SAPUTRA
NIM : 225520211012

PENDAHULUAN

Dalam dunia pengembangan game modern, kemampuan untuk membuat game sederhana menjadi landasan penting sebelum melangkah ke proyek yang lebih kompleks. Melalui proyek ini, saya berusaha membuat sebuah game platformer 2D sederhana menggunakan Godot Engine secara mandiri. Fokus utama adalah memahami mekanisme dasar pembuatan karakter, tilemap, navigasi scene, dan elemen antarmuka pengguna (UI).

Proyek ini dirancang untuk melatih keterampilan teknis dalam menggunakan node-node penting di Godot seperti CharacterBody2D, TileMap, Control, Button, serta mengimplementasikan script sederhana dengan GDScript. Selain itu, tujuan dari proyek ini adalah mengembangkan pemahaman mendalam tentang struktur scene, animasi karakter, serta pembuatan menu navigasi yang fungsional.

Dengan mengandalkan eksperimen dan eksplorasi langsung terhadap fitur-fitur Godot, proyek ini menunjukkan bagaimana game sederhana dapat dikembangkan dari tahap awal hingga siap dijalankan, tanpa bergantung pada tutorial eksternal.

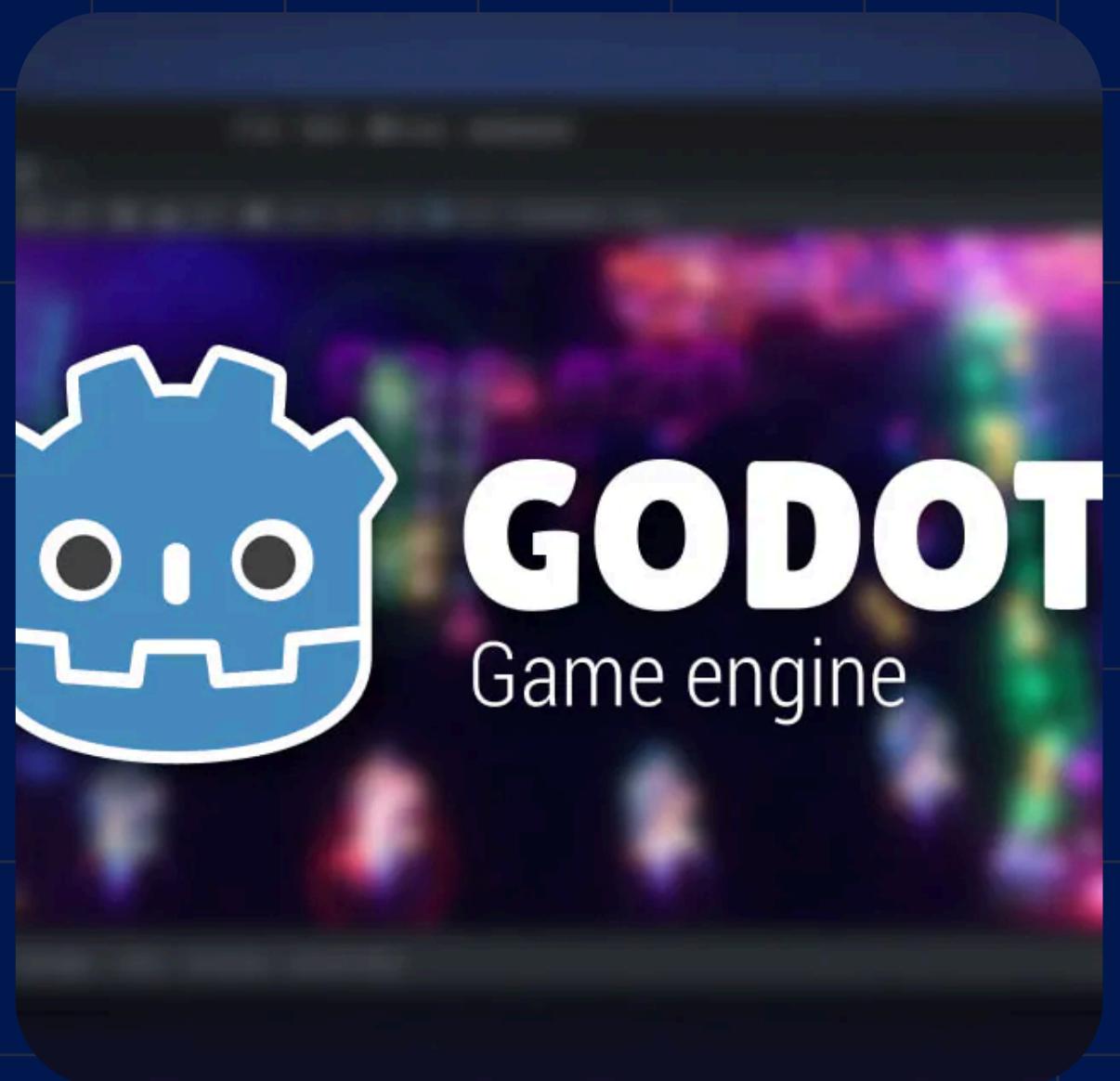
TUJUAN

- **Membuat game 2D platformer sederhana berbasis Godot Engine.**
- **Memahami konsep pembuatan scene, node, dan navigasi antar scene.**
- **Mengembangkan skill scripting dasar menggunakan GDScript.**
- **Membuat antarmuka pengguna (UI) sederhana untuk navigasi game.**
- **Meningkatkan kemampuan desain level dasar dengan TileMap.**

3. ALAT DAN BAHAN

- Software: Godot Engine versi terbaru.
- Asset: Gambar karakter dan tile sederhana untuk environment.

⋮ ⋮ ⋮



4. METODE PENGERJAAN

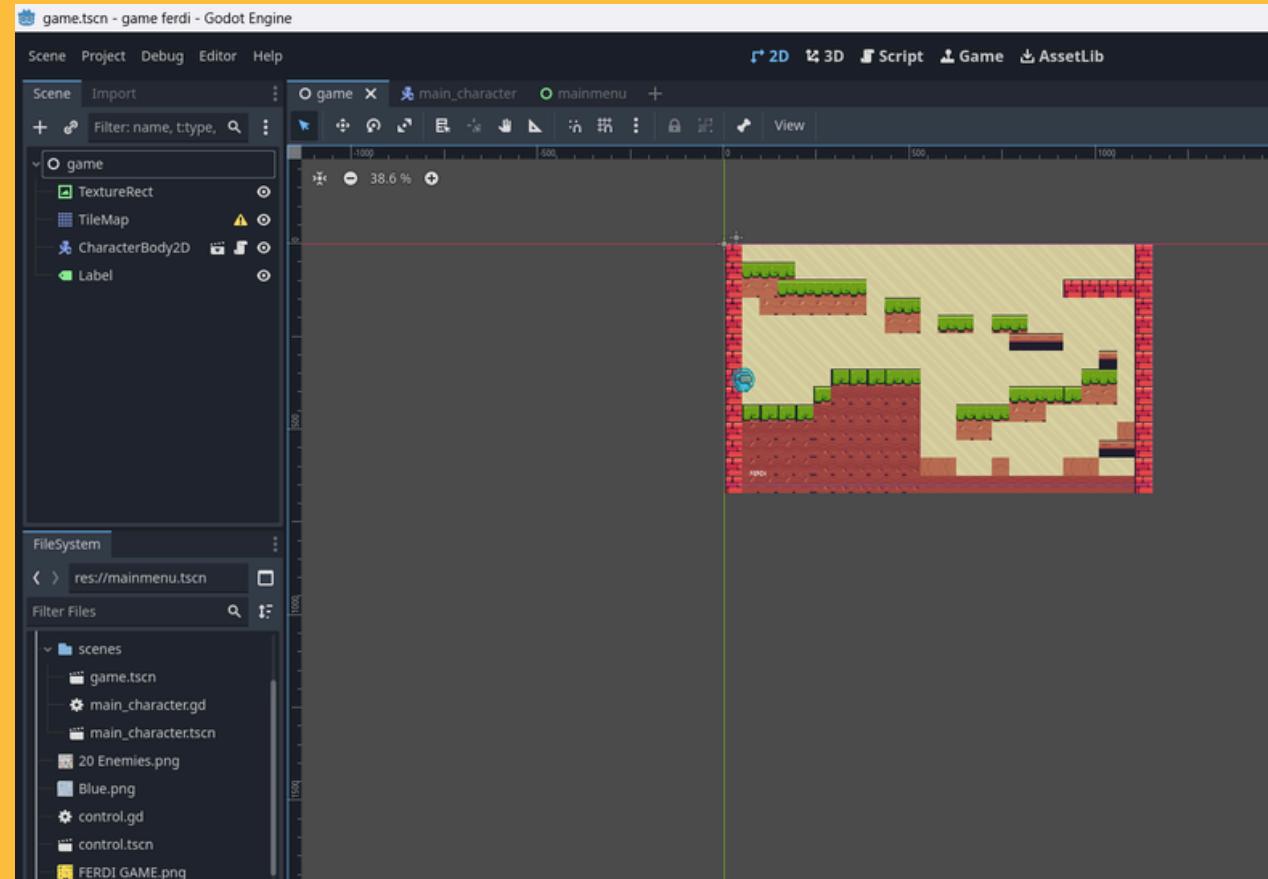
4.1 PEMBUATAN STRUKTUR PROJECT

PERTAMA, DIBUAT FOLDER PROJECT DENGAN STRUKTUR RAPI:

SCENES/ → BERISI SCENE UTAMA DAN SUBSCENE SEPERTI KARAKTER.

ASSETS/ → BERISI GAMBAR KARAKTER, TILESET, BACKGROUND.

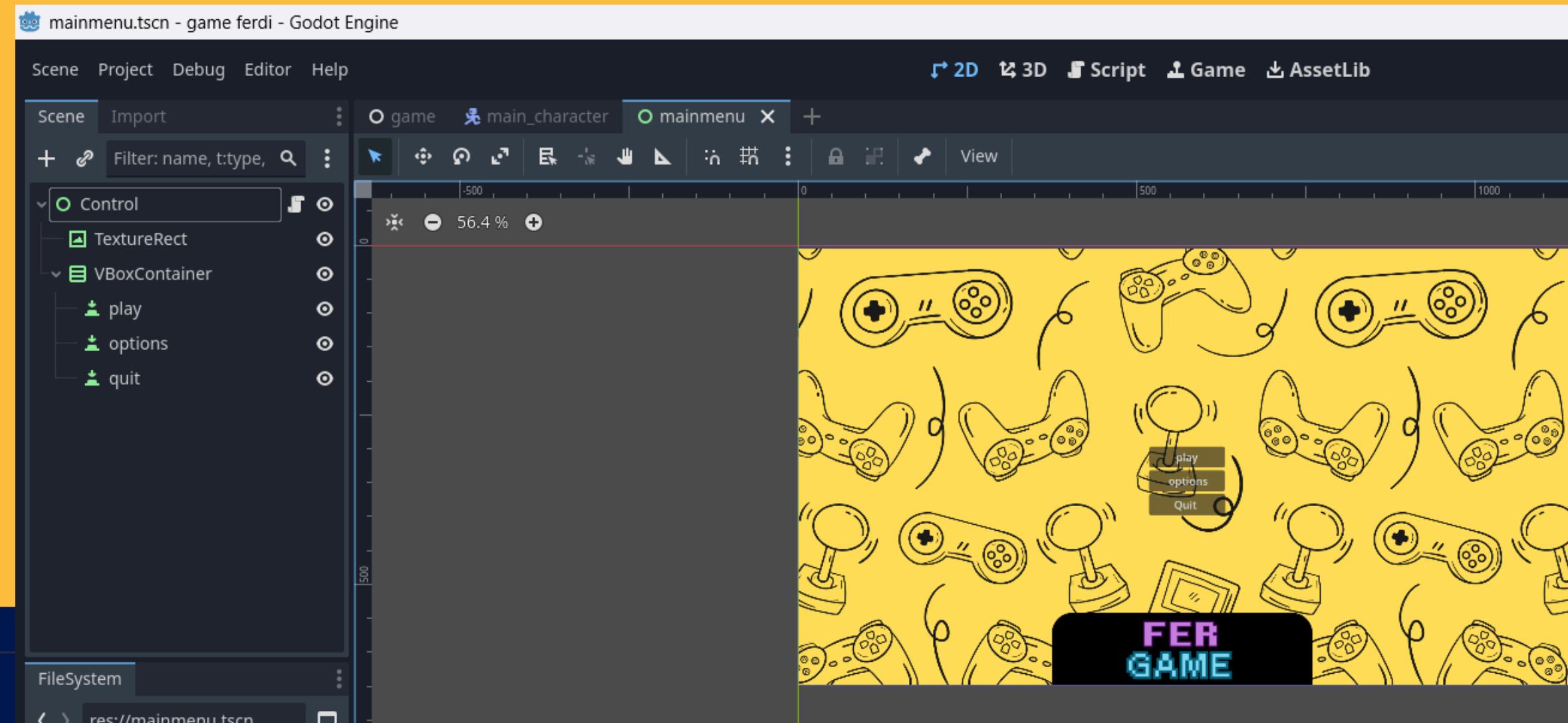
SCRIPTS/ → TEMPAT SCRIPT-SCRIPT .GD.



4. METODE PENGERJAAN

4.2 PEMBUATAN MENU UTAMA

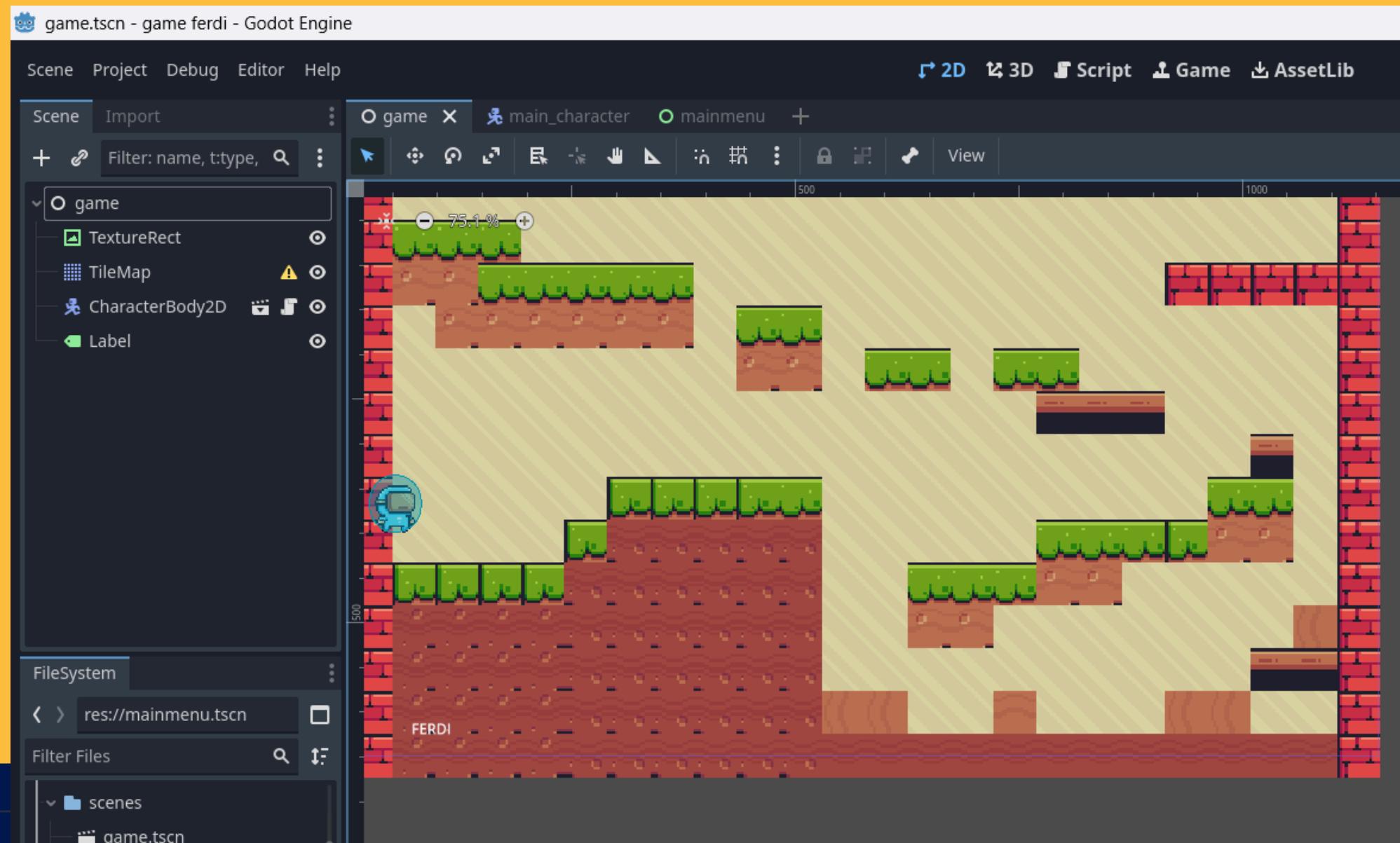
- MENAMBAHKAN NODE CONTROL SEBAGAI DASAR MENU.
- MEMBUAT TOMBOL PLAY, OPTIONS, DAN QUIT MENGGUNAKAN NODE BUTTON DALAM VBOXCONTAINER.
- MENYUSUN POSISI TOMBOL AGAR RAPI DAN PROPORSIONAL.
- MENAMBAHKAN BACKGROUND MENGGUNAKAN NODE TEXTURERECT DAN MENEMPATKANNYA DI BAWAH TOMBOL (MENGATUR URUTAN NODE AGAR BACKGROUND BERADA DI BELAKANG).



4. METODE PENGERJAAN

4.3 PEMBUATAN SCENE GAME

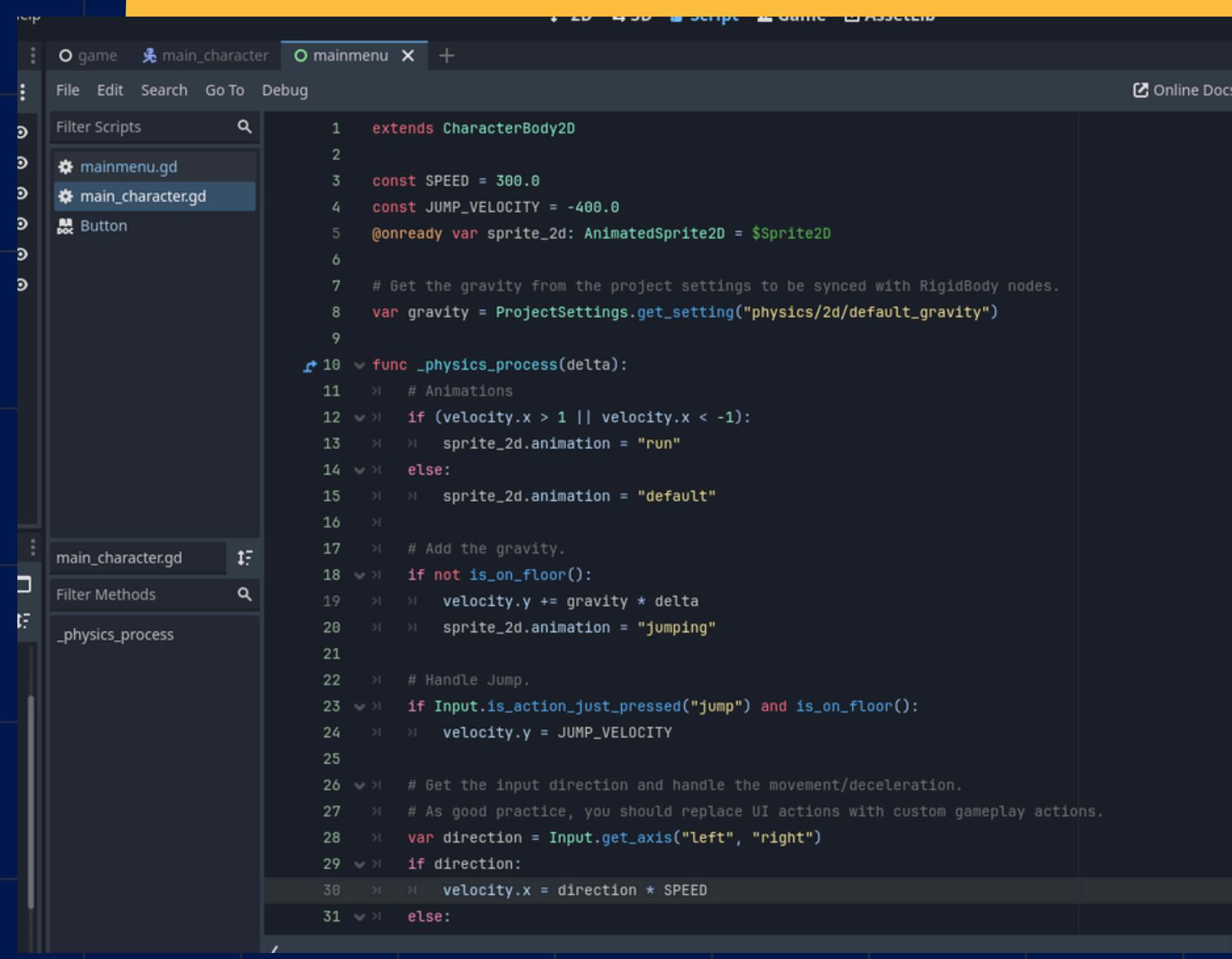
- MEMBUAT NODE NODE2D SEBAGAI ROOT UNTUK SCENE GAMEPLAY.
- MENAMBAHKAN TILEMAP UNTUK LANTAI/ARENA BERJALAN KARAKTER.
- MENAMBAHKAN KARAKTER PLAYER SEBAGAI CHARACTERBODY2D, DILENGKAPI SPRITE, COLLISION, DAN ANIMASI.



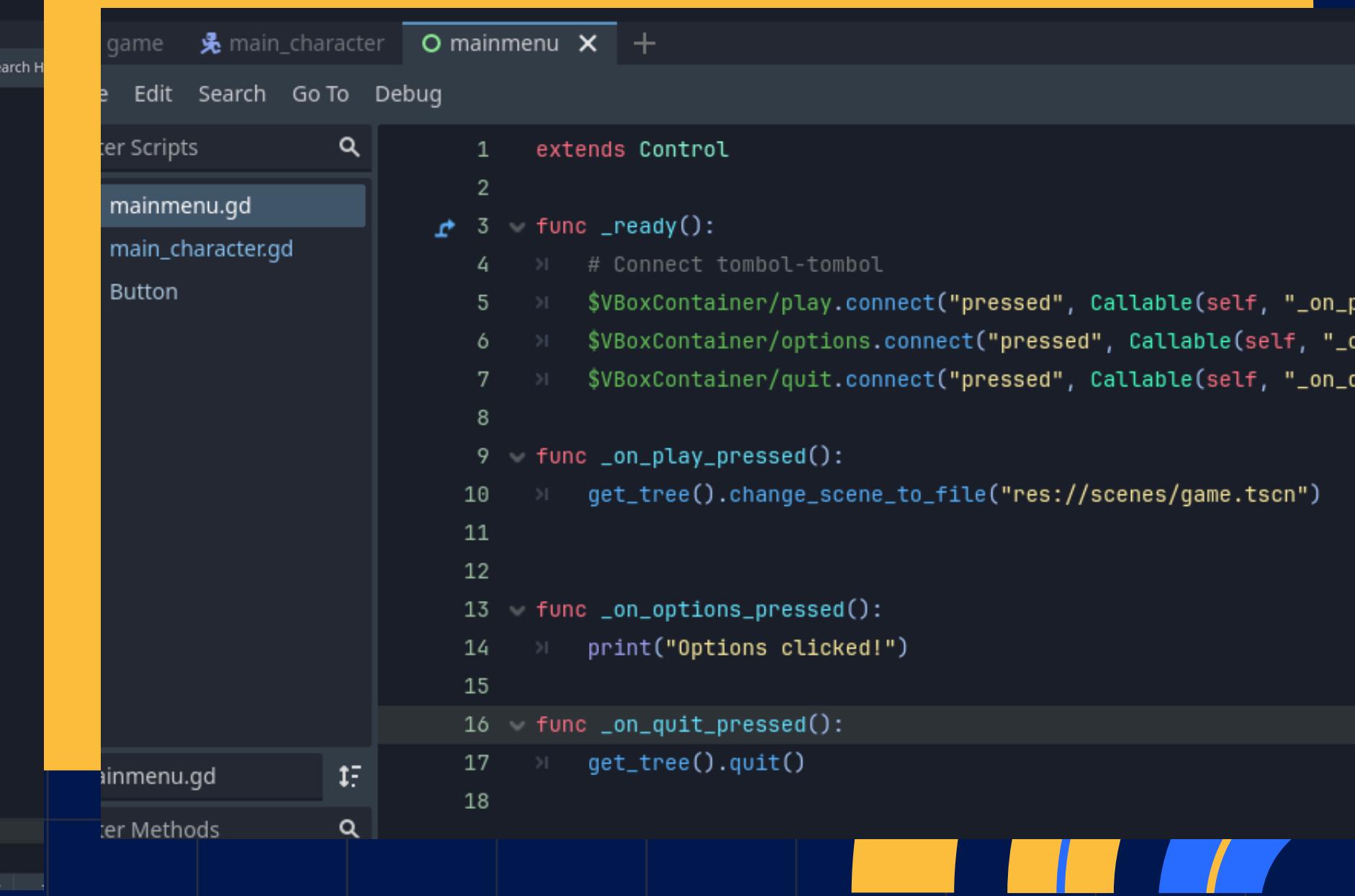
4. METODE PENGERAJAAN

4.4 SCRIPTING DAN NAVIGASI

- MENGHUBUNGKAN TOMBOL PLAY AGAR KETIKA DIKLICK, BERPINDAH KE SCENE GAMEPLAY (NODE2D BUKAN LANGSUNG KE KARAKTER).
- MENAMBAHKAN SCRIPT NAVIGASI ANTAR SCENE DI MAINMENU.GD.
- MEMBUAT SCRIPT KONTROL DASAR KARAKTER (BERGERAK KANAN, KIRI, LOMPAT).



```
1  extends CharacterBody2D
2
3  const SPEED = 300.0
4  const JUMP_VELOCITY = -400.0
5  @onready var sprite_2d: AnimatedSprite2D = $Sprite2D
6
7  # Get the gravity from the project settings to be synced with RigidBody nodes.
8  var gravity = ProjectSettings.get_setting("physics/2d/default_gravity")
9
10 func _physics_process(delta):
11     # Animations
12     if (velocity.x > 1 || velocity.x < -1):
13         sprite_2d.animation = "run"
14     else:
15         sprite_2d.animation = "default"
16
17     # Add the gravity.
18     if not is_on_floor():
19         velocity.y += gravity * delta
20         sprite_2d.animation = "jumping"
21
22     # Handle Jump.
23     if Input.is_action_just_pressed("jump") and is_on_floor():
24         velocity.y = JUMP_VELOCITY
25
26     # Get the input direction and handle the movement/deceleration.
27     # As good practice, you should replace UI actions with custom gameplay actions.
28     var direction = Input.get_axis("left", "right")
29     if direction:
30         velocity.x = direction * SPEED
31     else:
```

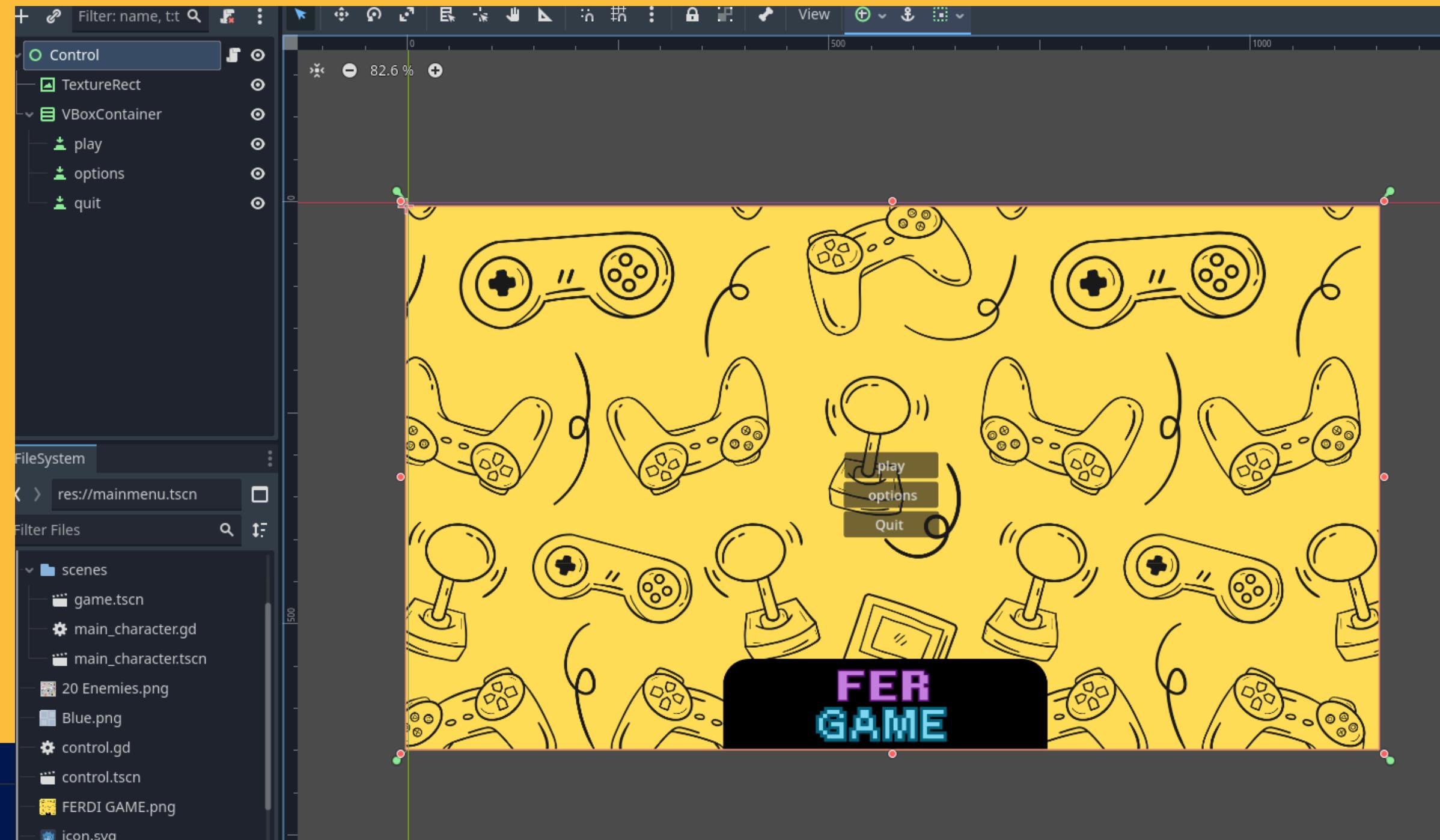


```
1  extends Control
2
3  func _ready():
4      # Connect tombol-tombol
5      $VBoxContainer/play.connect("pressed", Callable(self, "_on_play_pressed"))
6      $VBoxContainer/options.connect("pressed", Callable(self, "_on_options_pressed"))
7      $VBoxContainer/quit.connect("pressed", Callable(self, "_on_quit_pressed"))
8
9  func _on_play_pressed():
10     get_tree().change_scene_to_file("res://scenes/game.tscn")
11
12
13  func _on_options_pressed():
14     print("Options clicked!")
15
16  func _on_quit_pressed():
17     get_tree().quit()
18
```

4. METODE PENGERJAAN

4.5 PENYESUAIAN DAN ESTETIKA

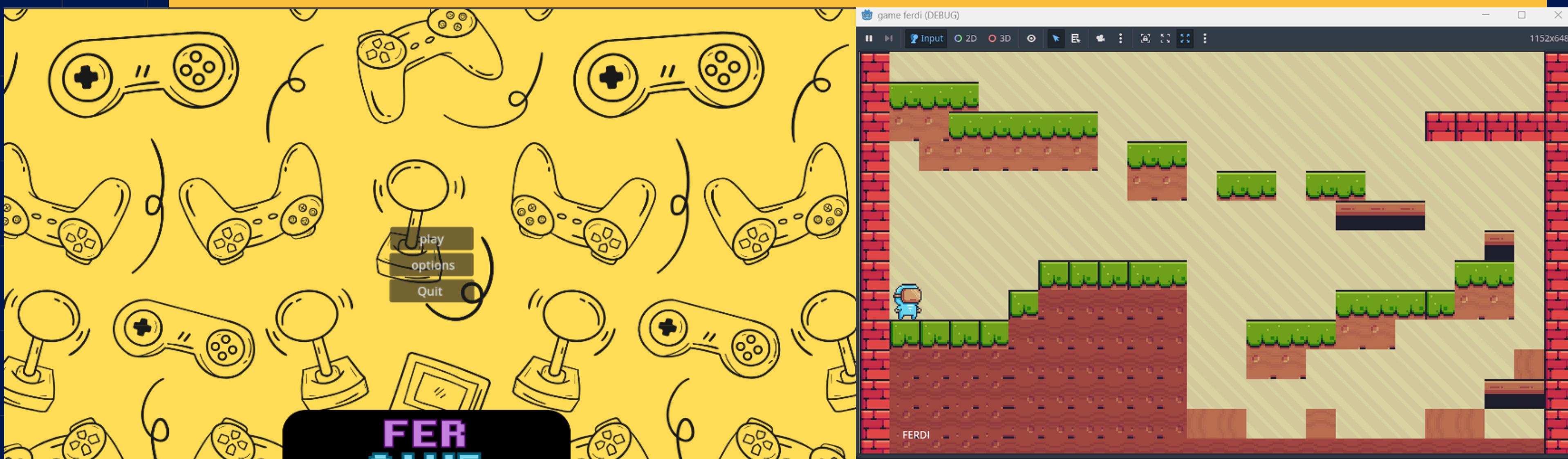
- MENAMBAHKAN BACKGROUND DI MENU DAN GAME SCENE.
- MENGATUR SCALE BUTTON AGAR LEBIH BESAR DAN LEBIH MUDAH DIKLICK.
- MENGATUR POSISI KAMERA DAN VIEWPORT AGAR TAMPILAN NYAMAN.



5. HASIL

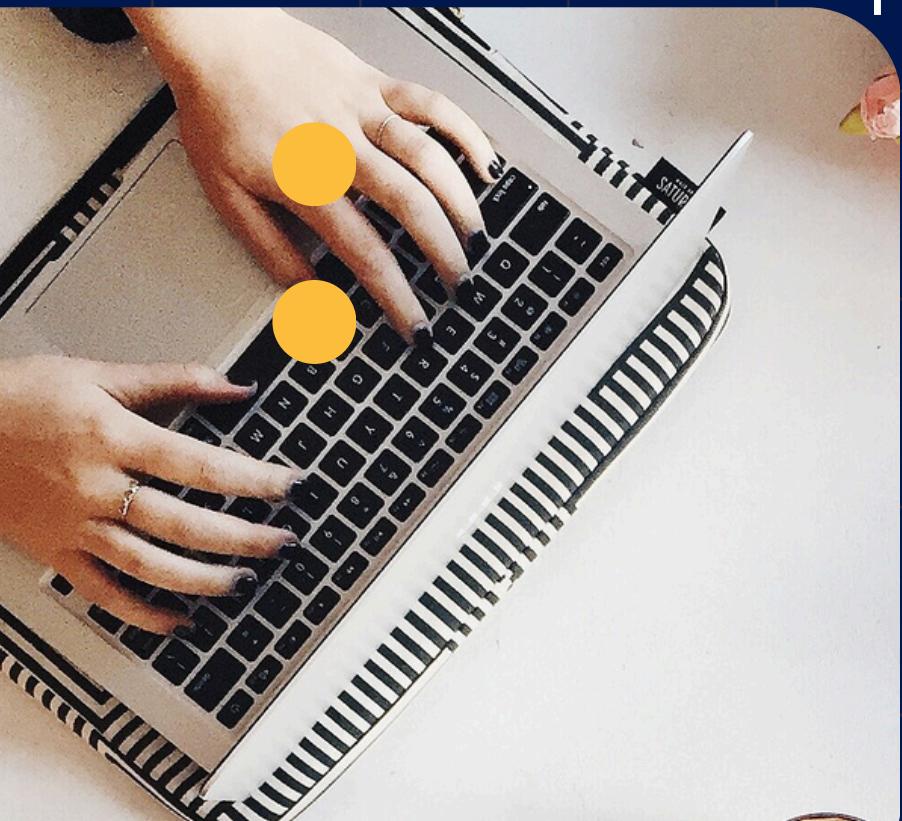
HASIL

- MENU UTAMA DAPAT TAMPIL DENGAN BACKGROUND DAN TOMBOL NAVIGASI.
- TOMBOL PLAY BERHASIL MENGARAHKAN KE SCENE GAMEPLAY YANG LENGKAP DENGAN KARAKTER DAN TILEMAP.
- KARAKTER DAPAT BERJALAN DAN MELOMPAT MENGGUNAKAN KONTROL KEYBOARD.
- SEMUA FUNGSI TOMBOL BERJALAN TANPA ERROR.
- TAMPILAN GAME LEBIH RAPI BERKAT BACKGROUND DAN PENGATURAN POSISI LAYER YANG TEPAT.



KESIMPULAN

Melalui pembuatan proyek game platformer 2D sederhana ini, saya memperoleh pemahaman mendalam tentang pengelolaan scene, scripting dasar, serta navigasi antar scene menggunakan Godot. Dengan mengandalkan eksplorasi mandiri, saya juga belajar cara mengatasi berbagai tantangan teknis yang muncul selama pengembangan. Pengalaman ini menjadi pondasi penting untuk mengembangkan proyek game yang lebih kompleks di masa depan.



**TERIMA
KASIH**