

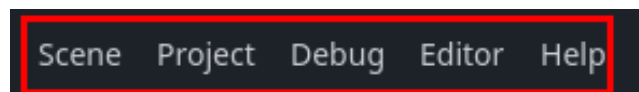
COACHING PROGRAM Day 1

Godot

Bagian 0: Apa itu Godot Engine

Godot Engine adalah game engine platform, gratis dan open source dirancang untuk membuat game 2D dan 3D yang menargetkan platform PC, mobile, dan web. Godot Engine mendukung beberapa bahasa pemrograman, termasuk C#, dan GDScript. Tetapi kebanyakan para developer menggunakan GDScript. GDScript adalah bahasa skrip yang dirancang khusus untuk Godot Engine. Ini mudah dipelajari dan digunakan, bahkan oleh pemula.

Bagian 1: Pengenalan UI



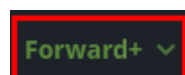
Menu : Bagian ini berfungsi sebagai tempat untuk pengaturan proyek secara keseluruhan. kita dapat menyimpan scene, mengonfigurasi opsi debug, dan membuka dokumentasi, serta beberapa fitur lainnya yang disediakan dalam menu ini.



Tab Layer: Tombol ini memungkinkan kita beralih antara konten layar tengah ke layar utama yang berbeda.



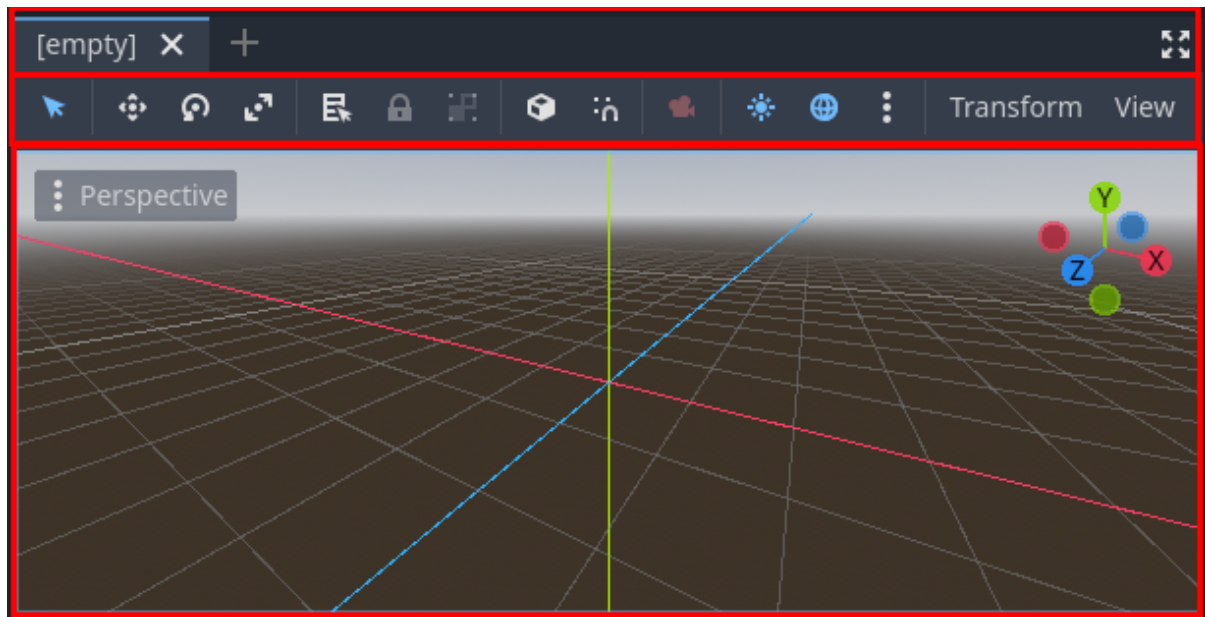
Play Button : Tombol-tombol ini memberi kita kemampuan untuk menjalankan dan menghentikan proyek secara keseluruhan atau adegan tertentu, mempermudah proses uji coba.



Dropdown Renderer: Bagian ini memungkinkan kita mengganti renderer yang digunakan untuk menampilkan grafis. kita dapat memilih antara Forward+, yang merupakan renderer penuh fitur di bagian atas, hingga mode Kinerja yang lebih ringan.

COACHING PROGRAM Day 1

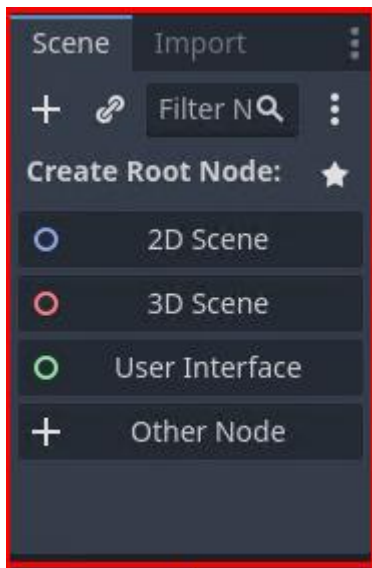
Godot



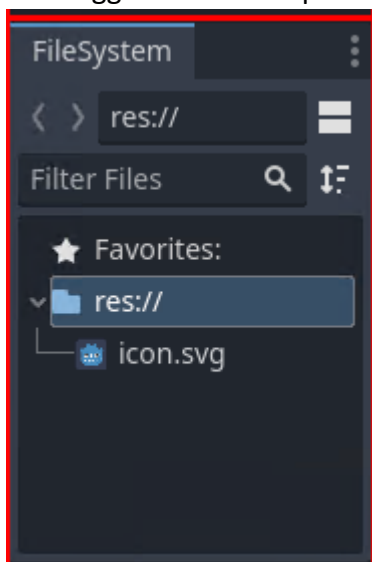
1. **Scene Tab:** Bagian ini memberi kita kemampuan untuk berpindah antara adegan atau membuat yang baru dengan menggunakan tombol tambah. Secara default, adegan yang baru dan kosong akan aktif. Tombol maksimalkan di ujung berfungsi untuk mengisi jendela aplikasi dengan tampilan layar tengah.
2. **Toolbar:** Ini adalah kumpulan tombol, toggle, dan menu yang berlaku untuk tampilan. Di sini, kita dapat beralih antara berbagai fungsi, seperti memilih, memindahkan, memutar, dan mengubah ukuran node yang dipilih. Tombol-tombol pertama dalam toolbar memungkinkan Anda melakukan fungsi-fungsi tersebut.
3. **Viewport:** Ini adalah pandangan utama dari adegan dalam kebanyakan kasus. Pada mode 2D, tampilan ini merupakan bidang datar, pada mode 3D, tampilan ini menggambarkan lingkungan tiga dimensi, dan dalam mode Skrip, tampilan ini berfungsi sebagai lingkungan pengembangan terpadu (IDE).

COACHING PROGRAM Day 1

Godot



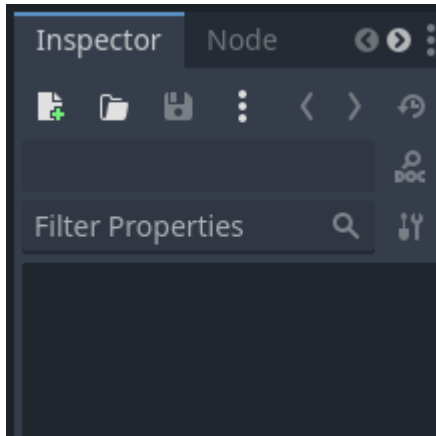
Dock Scene pada bagian ini menunjukkan hierarki node-node dalam adegan yang sedang aktif. Ketika kita membuka adegan yang kosong seperti yang kita lakukan sekarang, itu menampilkan tombol-tombol untuk memilih jenis node akar. Untuk penjelasan apa itu NODE itu nanti dijelaskan. Untuk saat ini, anggap saja mereka sebagai objek-objek yang digunakan dalam permainan kita, mulai dari sprite dan mesh 3D hingga tombol dan pemutar audio.



Dock FileSystem pada bagian ini menunjukkan semua file dalam proyek Anda, seperti gambar, skrip, dan model 3D. Akar/Root dari folder proyek adalah **res://**, yang merupakan singkatan dari resources. Secara default, Godot dilengkapi dengan satu gambar: logo Godot.

COACHING PROGRAM Day 1

Godot

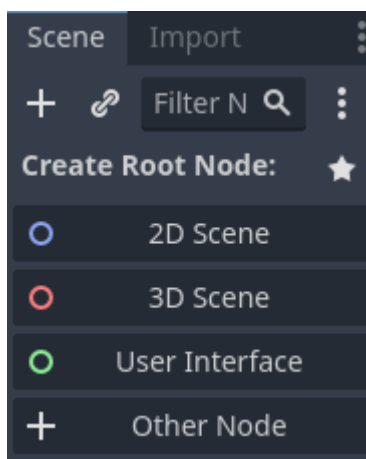


Dock di sebelah kanan menyimpan Inspector, yang memungkinkan kita melihat dan mengedit properti dari node atau sumber daya yang dipilih. Selain Inspector, ada dua tab lagi: Node dan History. Tab Node berisi gambaran umum tentang sinyal dan grup node yang dipilih, sementara tab History menunjukkan daftar tindakan terakhir yang Anda lakukan dalam proyek, seperti memindahkan node atau menyesuaikan pengaturan, sebagai contoh [kasih contoh].



Panel bagian bawah terdapat jendela output, konsol debug, mixer audio, editor animasi, dan editor shader. Editor yang bersifat kontekstual seperti editor tilemap juga akan muncul di sini ketika diperlukan.

Bagian 3: Membuat Node dan Scene



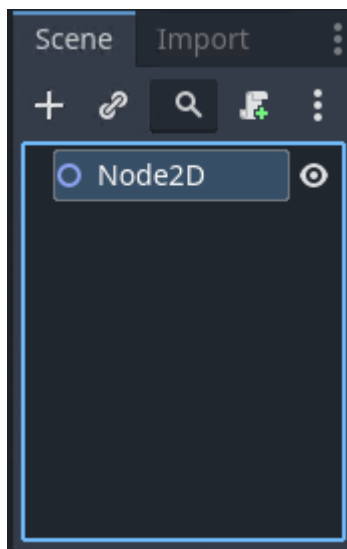
Node adalah blok bangunan terkecil dalam Godot dan bisa memiliki berbagai macam variasi. Sebuah node dapat menampilkan sprite, menggambar garis, memutar audio, menampilkan checkbox, melakukan permintaan web, dan banyak lagi. kita bahkan dapat membuat node kustom sendiri!

COACHING PROGRAM Day 1

Godot

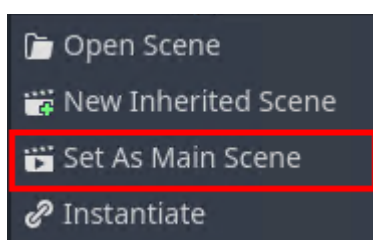
Apa yang kita buat di sini adalah sebuah node Node2D, sebuah objek 2D yang tidak melakukan apa pun secara langsung, tetapi memberikan posisi, rotasi, dan skala. kita akan menggunakan node ini sebagai induk untuk menambahkan node lain sebagai anak-anaknya.

Untuk membuat tujuannya lebih jelas, ubah nama Node2D menjadi GameRoot. kita dapat melakukannya dengan mengklik node lagi saat dipilih, menekan F2, atau mengklik kanan node dan memilih Rename dalam menu konteks. Sekarang ketik nama "GameRoot" dan tekan Enter untuk mengonfirmasi.



[tinggall save, terus ingetin kalau di saranke untuk membuat folder sesuai dengan jenis file yang di buat (misal : untuk file scan di simpan di folder scene, untuk gambar di simpan pada folder gambar dst.)]

Di Godot, sebuah scene adalah hierarkis dari node-node yang berfungsi sebagai blok bangunan untuk membuat game dan aplikasi. Adegan dapat digunakan sebagai blueprint, memungkinkan kita membuat instance dari mereka dalam scene lain. Sebagai contoh, kita mungkin memiliki scene "world" yang mencakup sebuah instansi dari adegan "avatar pemain" dan scene "antarmuka pengguna", yang menampilkan informasi seperti jumlah permata yang dikumpulkan oleh pemain.



COACHING PROGRAM Day 1

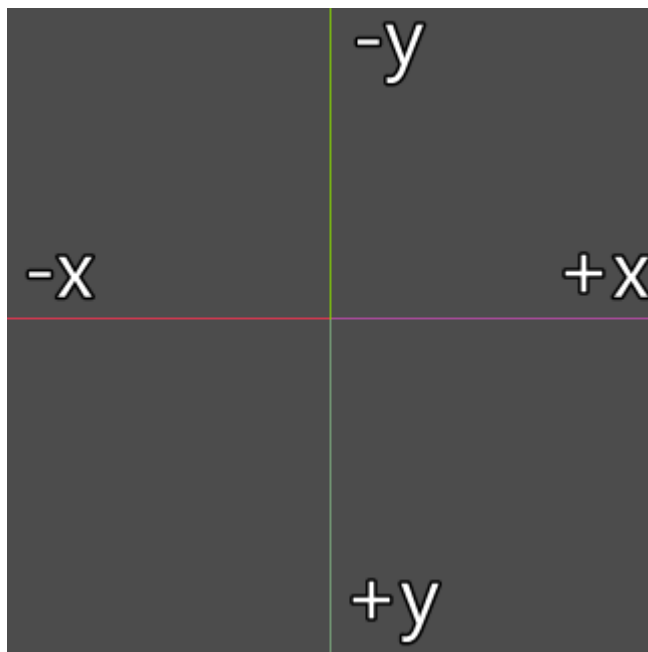
Godot

Setiap proyek di Godot memiliki scene utama, yang merupakan scene pertama yang dimuat saat memulai permainan. Dalam kebanyakan game, ini adalah layar judul, intro, atau kredit.. Untuk mengaturnya seperti itu, klik kanan pada file adegan game.tscn dan pilih Set As Main Scene dalam menu konteks.

[test menambah object, buka scene tab baru -> pilih 2D -> masukkan gambar icon godot (berikan sedikit penjelasan)]

[ini penjelasannya]

Garis berwarna pada tampilan viewport layar 2D disebut sumbu x dan y, dengan garis horizontal dari kiri ke kanan sebagai sumbu x dan garis vertikal dari atas ke bawah sebagai sumbu y. Setiap posisi x di sebelah kiri pusat adalah negatif, sementara setiap posisi x di sebelah kanan pusat adalah positif. Begitu juga dengan sumbu y: setiap posisi y di atas pusat adalah negatif, sementara setiap posisi y di bawah pusat adalah positif.



Setiap posisi 2D dipetakan pada sumbu-sumbu ini. Node sprite sebaiknya berada di tengah viewport agar lebih mudah menginstansiasi adegan baru ini dan menempatkannya nanti, sehingga posisinya harus (X:0, Y:0). Meskipun kita dapat menempatkan ikon dengan mata dan berharap yang terbaik, cara yang lebih sesuai untuk memanipulasi posisi node adalah dengan menggunakan Inspector.

[posisi icon disarankan di Tengah, jika sudah tinggal save, test jalankan, fungsi icon ini sebagai child yang di mana yg di buat pada scene ini akan digunakan pada scene lain]

COACHING PROGRAM Day 1

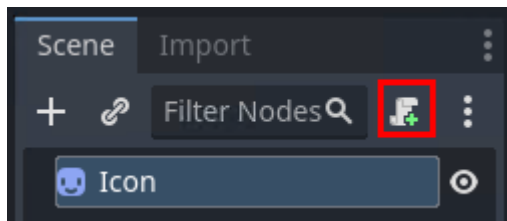
Godot

Bagian 3: Scripting

Pemrograman menggunakan script merupakan keterampilan dasar ketika bekerja dengan Game Engine apa pun, termasuk Godot. Script adalah kumpulan perintah yang memberi tahu Engine apa yang harus dilakukan dan dapat memperluas fungsionalitasnya. Di Godot, Script dapat dilampirkan pada node untuk menambahkan fitur dan perilaku baru ke dalamnya.

Sebagai contoh, sebuah node Sprite 2D pada dasarnya hanya dapat menampilkan tekstur, tetapi dengan Script, object tersebut dapat mengikuti kursor, melompat, dsb. Script seperti ini dapat digunakan untuk memanggil API web, menghitung rumus, dan melakukan tindakan lain yang diperlukan untuk membuat game atau aplikasi.

Untuk Godot sendiri itu mendukung dua bahasa pemrograman utama untuk skrip: GDScript dan C#.



untuk memulai, buka scene ikon. Sekarang pilih node Icon melalui dock Scene dan klik tombol skrip di kanan atas dock Scene, yang terlihat seperti kertas dengan tanda plus berwarna hijau.

Secara default, akan dibuat GDScript baru yang dibangun berdasarkan jenis node yang dipilih, yaitu Sprite 2D. Path tempat script akan ditempatkan harus diubah, jadi klik tombol folder di sebelah properti Path. Pada jendela yang terbuka, naik satu direktori dengan panah ke atas di kiri atas dan buat folder baru dengan nama "scripts" di direktori root proyek.

Bagian extends Sprite 2D di bagian atas membuat skrip ini dapat mengakses semua properti node Sprite 2D, seperti tekstur dan transformasinya.

func _ready() adalah suatu fungsi yang dipanggil oleh mesin sekali saat awal dari masa hidup node.

Fungsi _process dipanggil setiap frame oleh mesin.

pass tidak melakukan apa-apa di sini kecuali keluar dari fungsi. Mereka berfungsi sebagai tempat penampung untuk kode Anda sendiri mungkin dalam Bahasa lain ini seperti return.

[Bikin di _ready()]

COACHING PROGRAM Day 1

Godot

```
print("Hello, I'm ", name)
```

[selanjutnya bikin di _process]

```
rotate(delta)
```

[tambahan menggunakan IF]

```
extends Sprite2D

# Variable to control rotation direction
var rotateClockwise = true

# Called when the node enters the scene tree for the first time.
func _ready():
    print("Hello, I'm ", name)

# Called every frame. 'delta' is the elapsed time since the previous frame.
func _process(delta):
    # Check if rotation direction should be changed based on input
    if Input.is_action_pressed("ui_right"):
        rotateClockwise = true
    elif Input.is_action_pressed("ui_left"):
        rotateClockwise = false

    # Rotate based on the direction
    if rotateClockwise:
        rotate(delta)
    else:
        rotate(-delta)

    # Example of using an if statement to change rotation speed
    if Input.is_action_pressed("ui_up"):
        rotate(delta * 2) # Double rotation speed if 'ui_up' is pressed
```

[penjelasan]

1. extends Sprite2D: Mendeklarasikan bahwa skrip ini adalah turunan dari kelas Sprite2D. Ini berarti skrip ini akan terkait dengan node Sprite2D.

COACHING PROGRAM Day 1

Godot

2. `var rotateClockwise = true`: Membuat variabel `rotateClockwise` yang digunakan untuk mengontrol arah rotasi. Nilai awal diatur sebagai `true`, yang berarti rotasi searah jarum jam.
3. `func _ready()`:: Fungsi ini dipanggil sekali saat node memasuki pohon adegan untuk pertama kalinya. Di sini, pesan "Hello, I'm " diikuti dengan nama node dicetak ke konsol.
4. `func _process(delta)`:: Fungsi ini dipanggil setiap frame. `delta` adalah waktu yang berlalu sejak frame sebelumnya.
 - a. Pernyataan `if` untuk Mengubah Arah Rotasi:
 - i. Pernyataan `if` dan `elif` digunakan untuk memeriksa apakah tombol panah kanan atau kiri ditekan.
 - ii. Jika panah kanan ditekan, `rotateClockwise` diatur sebagai `true` (searah jarum jam).
 - iii. Jika panah kiri ditekan, `rotateClockwise` diatur sebagai `false` (berlawanan jarum jam).
 - b. Rotasi Berdasarkan Arah:
 - i. Pernyataan `if` digunakan untuk memeriksa nilai `rotateClockwise`. Jika `true`, node dirotasi searah jarum jam, jika `false`, dirotasi berlawanan jarum jam.
 - c. Contoh Pernyataan `if` untuk Mengubah Kecepatan Rotasi:
 - i. Pernyataan `if` digunakan untuk memeriksa apakah tombol panah atas ditekan.
 - ii. Jika ya, kecepatan rotasi dua kali lipat dengan mengalikan `delta` dengan 2.

COACHING PROGRAM Day 1

Godot

```

    extends Sprite2D

    # Variable to control rotation direction
    var rotateClockwise = true

    # Called when the node enters the scene tree for the first time.
    func _ready():
        print("Hello, I'm ", name)

    # Called every frame. 'delta' is the elapsed time since the previous frame.
    func _process(delta):
        # Check if rotation direction should be changed based on input
        if Input.is_action_pressed("ui_right"):
            rotateClockwise = true
        elif Input.is_action_pressed("ui_left"):
            rotateClockwise = false

        # Rotate based on the direction
        if rotateClockwise:
            rotate(delta)
        else:
            rotate(-delta)

    # Example of using a while loop to print numbers
    var i = 0
    while i < 5:
        print("Loop iteration:", i)
        i += 1

```