**Mecânicas**

Ações que influenciam na gameplay. Formas que o jogador tem de interagir com o jogo.

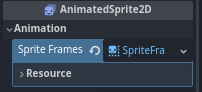
10 mecânicas ruins não carregam um jogo! O que importa é ela ser boa.

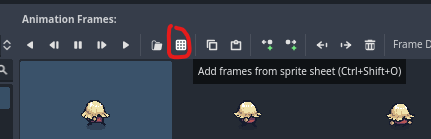
**Criando nosso primeiro inimigo:**

Iremos escolher como primeiro inimigo o PawnRed

Iremos fazer com que nossos inimigos tenha apenas 1 animação! Então iremos criar um nova cena para o inimigo, nela iremos criar um node de character, colocar um colission shape e colocaremos um novo node! **O Animated Sprite2d**

Nele iremos criar um novo Sprite Frames

E irar aparecer uma aba tipo a da animation passada. Ela é para fazer animações mais simples!



Iremos nessa opção, escolher a pagina do arquivo e configurar e selecionar na ordem

E pronto, nosso 1ª inimigo terá sido criado.

**Fazendo com que o inimigo ande até X direção:**

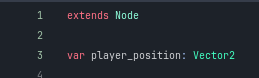
IA de “plástico” não importa muito se é algo superrealista e etc, o importante é que a pessoa goste!

Criando Script do inimigo.

Para o inimigo vim na nossa direção precisamos calcular a posição do player – posição do inimigo.

**Mas como pegar a posição do player?** Usando Autoload (singletons)

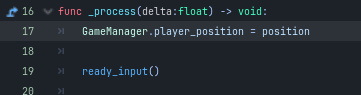
É um Script global, criado no res que chamamos de game\_manager. Como queremos apenas a posição do jogador podemos criar um var player\_position

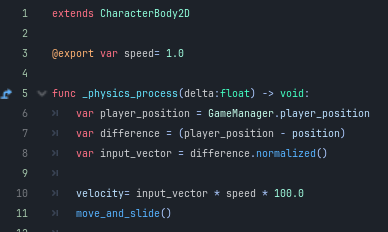


Geralmente quando queremos anexar um script a gente jogaria no node! Mas esse não. A gente vai em Project Settings> autoload> pasta e procurar o script e add.

Isso significa que o script vai sempre carregar primeiro que tudo.

E para os scripts lerem isso é só colocar: GameManager.player\_position = position



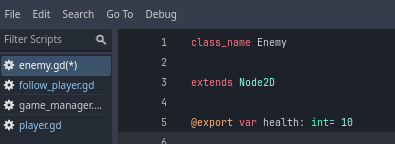


Configuramos também no script do inimigo a função de girar (igual do player)



**Vamos criar o script do inimigo: Para criar sistema de dano/vida.**

Na pasta enemies criaremos um novo script chamado enemy.



1ª Iremos colocar ponto de vida do nosso inimigo.

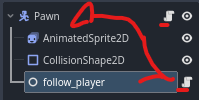
Cada node só pode ter 1 script anexado a ele. Então como posso fazer com que muitos scripts atuem no mesmo node? Composição.

Criamos um novo node dentro do que a gente quer sendo um “Node” e anexar o script nele. Após isso configuramos os scripts para ver se não há erro.

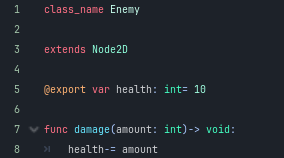
No script Follow Player mudamos:



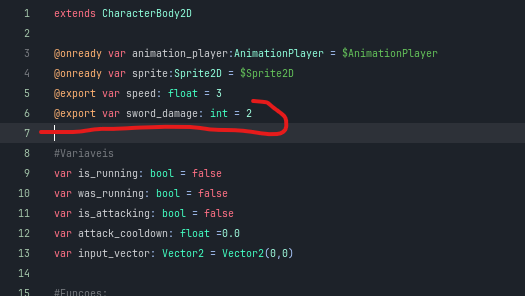
Pois agora o script esta anexado no “filho” e a gente quer ter acesso ao “Pai”.



**Vamos mexer no script enemy agora.**

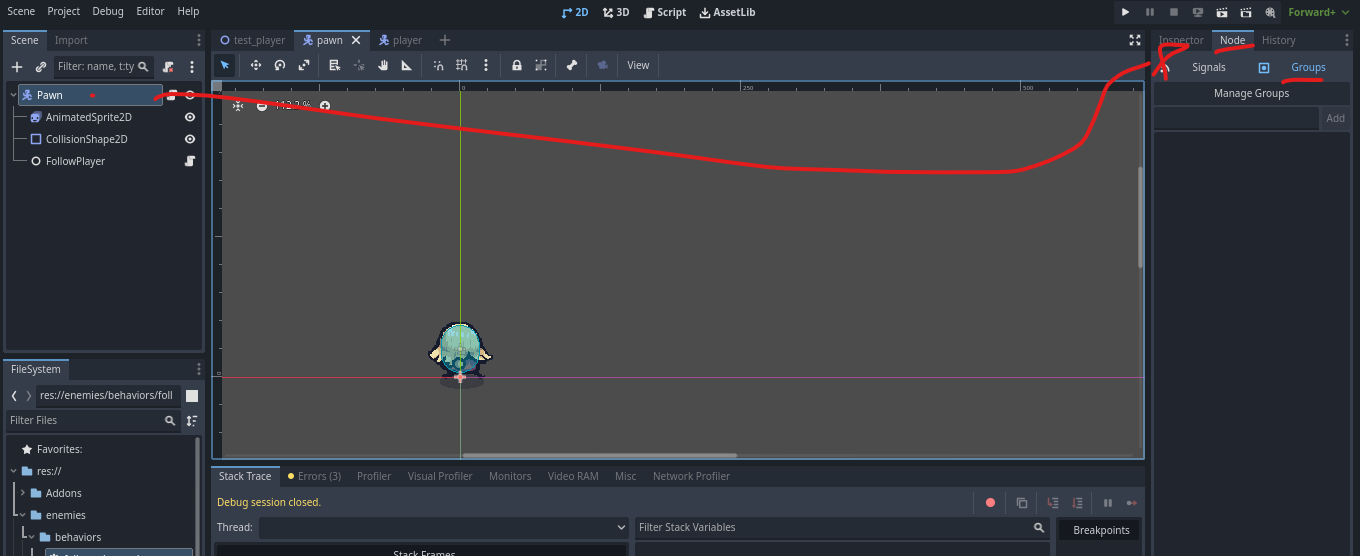


Se temos um “dano” que o inimigo vai tomar, então quanto de dano a espada vai da? Assim mexeremos no script do jogador para fazer isso.



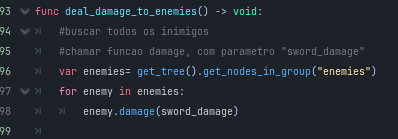
E na função attack iremos configurar o dano no inimigo.

Mas como iremos fazer com que uma função de outro script seja acessada por ai? Se a gente for no Pawn e em Node tem algo chamado “Grupos”

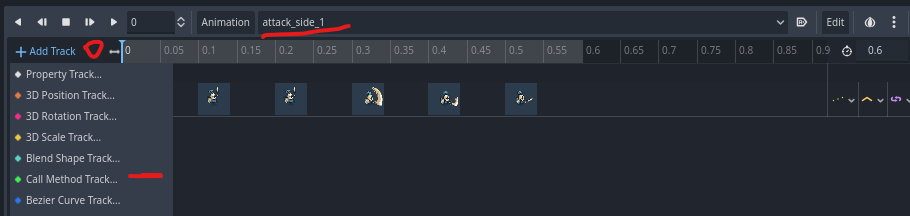


Colocamos o Pawn no grupo enemies e depois volto no script do player

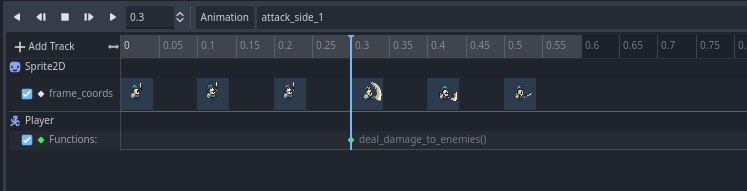
Primeiro precisamos pegar a “arvore” usando get\_tree e pegar todos os nodes do grupo enimes usando: get\_tree().get\_nodes\_in\_group("enemies")



E onde vamos chamar essa função? Na nossa AnimationPlayer

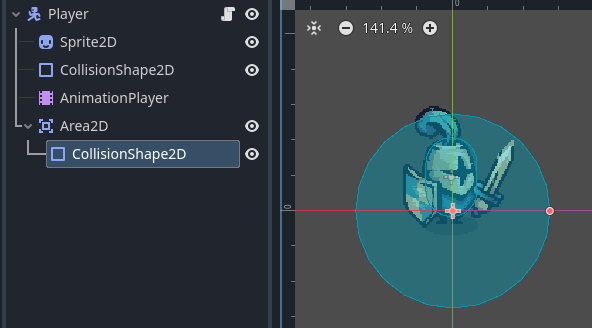


Escolhe aonde vai chamar a função e seleciona a função.

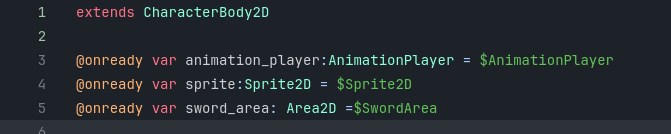


**Ferir inimigos próximos**:

Criar um node que verifica a área de dano. Na cena player vamos criar um node chamado: Area2d e nele criaremos um collisionshape2d e colocar um shape



Renomeamos para **SwordArea.** Agora configurando no Scritp:





Mas eu queria bater apenas no inimigo que tiver na minha frente.

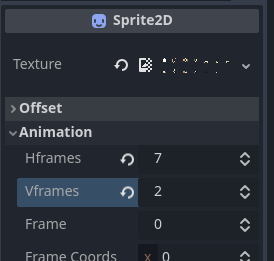
**Dot Product**: Função Matematica no qual obtem resultado baseado na comparação de vetores.



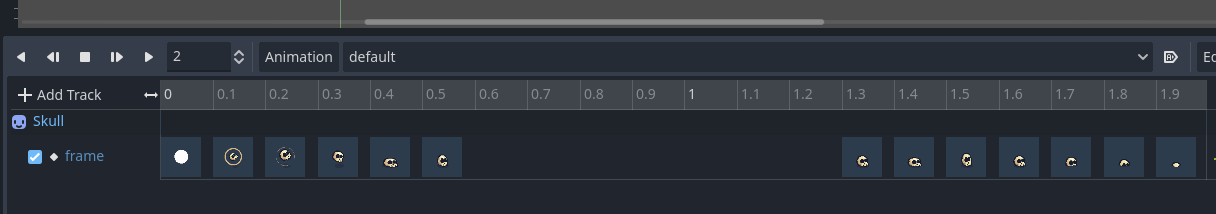
E agora? Tem que fazer o inimigo receber o dano ou “morrer”

No pacote tem uma caveira. Iremos usar ela

Para isso iremos criar uma nova cena chama skull com um sprite2d, move a textura para o textura e coloca em Animation.

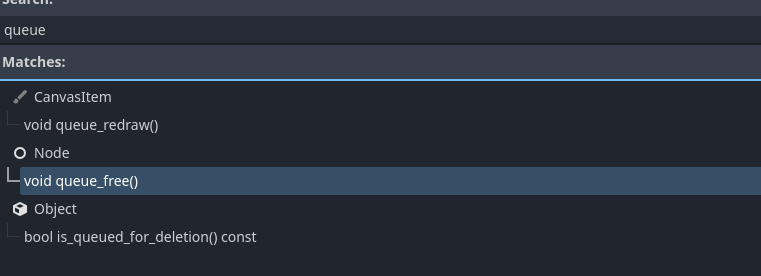


No sprite2d iremos criar um Animationplayer



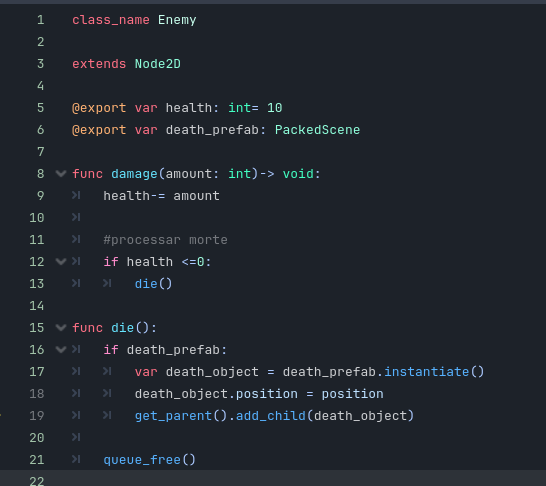
Agora iremos criar uma faixa de função, add track e colocar uma call method track.

Insere uma key e chamaremos a queue free

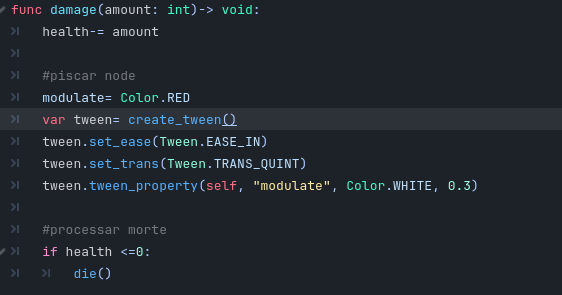


Ela marca a função para ser removido do jogo (destruindo do jogo)

Agora no script do inimigo:

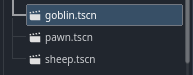


E para mudar a cor do objeto ao tomar dano?



**Criando mais inimigos no jogo**:

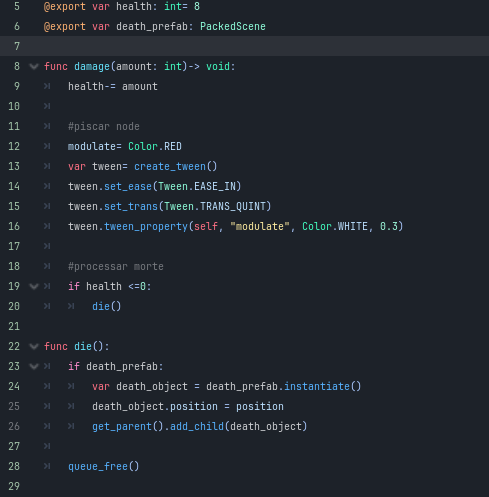
Eu vou na cena do inimigo que criei e duplico ela, e troco o animatedsprite para o que eu quero (apagando e selecionando o novo). Nisso posso mudar a vida, velocidade e etc do inimigo.



Criei 3 inimigos, o goblin, peão e a ovelha.

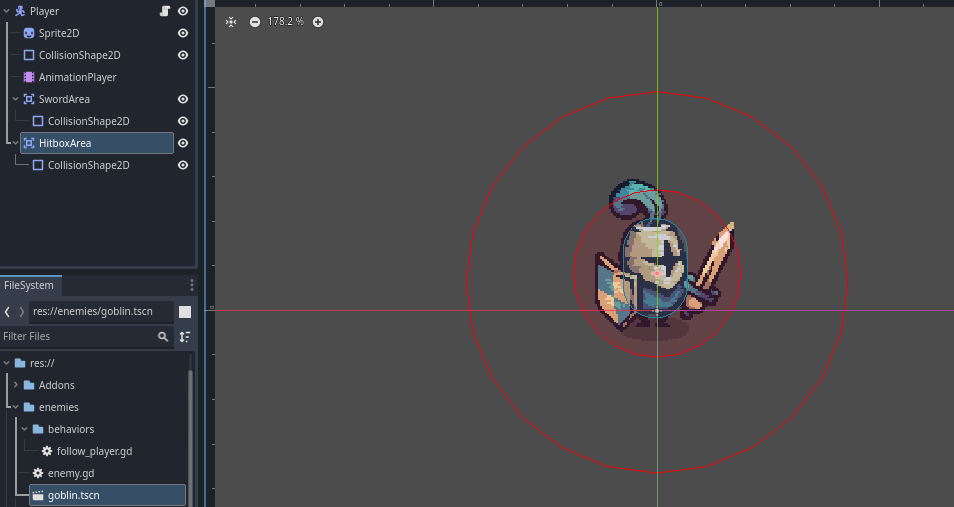
**Vida para o jogador:**

Vamos mexer no script do inimigo e copiar tudo para o player



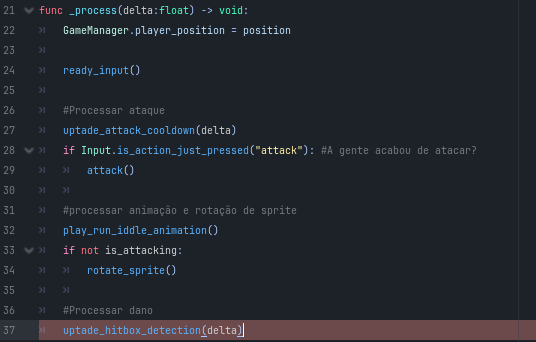
Coloquei esse mesmo código para o player.

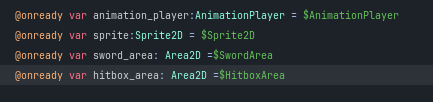
Só que tem um porem... A função para aplicar o dano. Para isso vamos criar um node no personagem chamado Area2d que precisa de um collisionshape2d



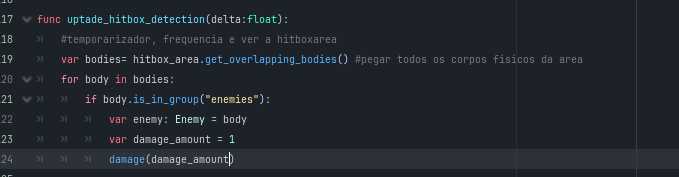
Mexendo no script do player agora

Na função process iremos criar uma função para processar o dano.

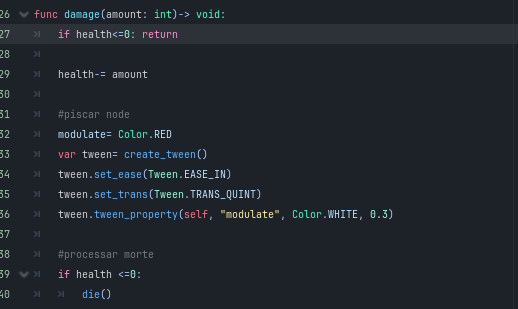




Criamos uma variável para saber o hitbox

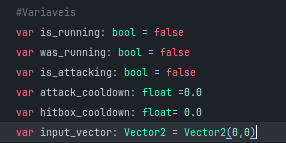


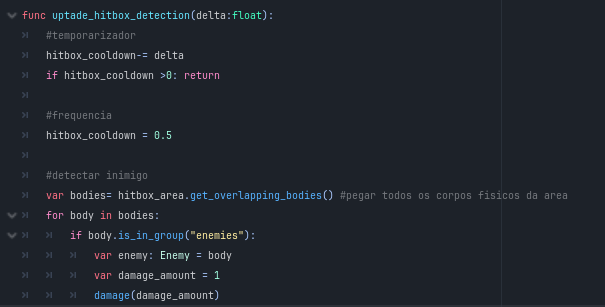
E na função damage colocamos para caso o personagem tenha morrido ele retornar.



Para evitar possíveis erros.

Para definir o temporarizador, criamos uma variável de hitbox cooldown





Agora criar uma “morte” para o jogador

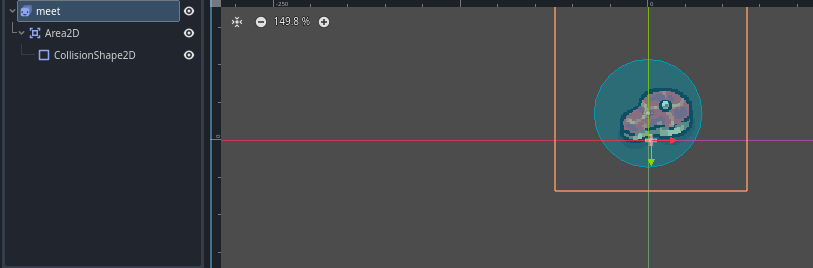
Na morte que a gente já tem o “skull” iremos duplicar ele e aumentamos o tamanho.

E no death do player arrastamos a cena que criamos.

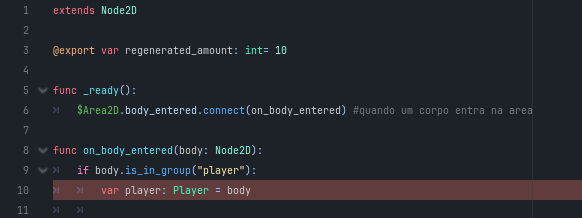
**Regeneração de vida:**

Vamos criar um objeto que “recupera” a vida.

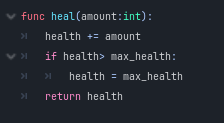
Criamos uma nova cena com o tipo de animatedsprite2d e criamos um novo spriteframe, e selecionamos o objeto escolhido (meet) e nele criamos um area2d + colissionshape pois não queremos colisão física.



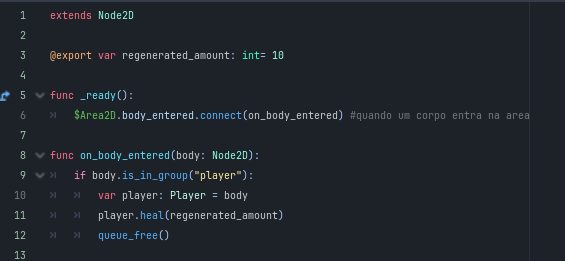
Criando script para ela para segurar a quantidade de vida a ser regenerado.



No player vamos criar a função de regenerar vida

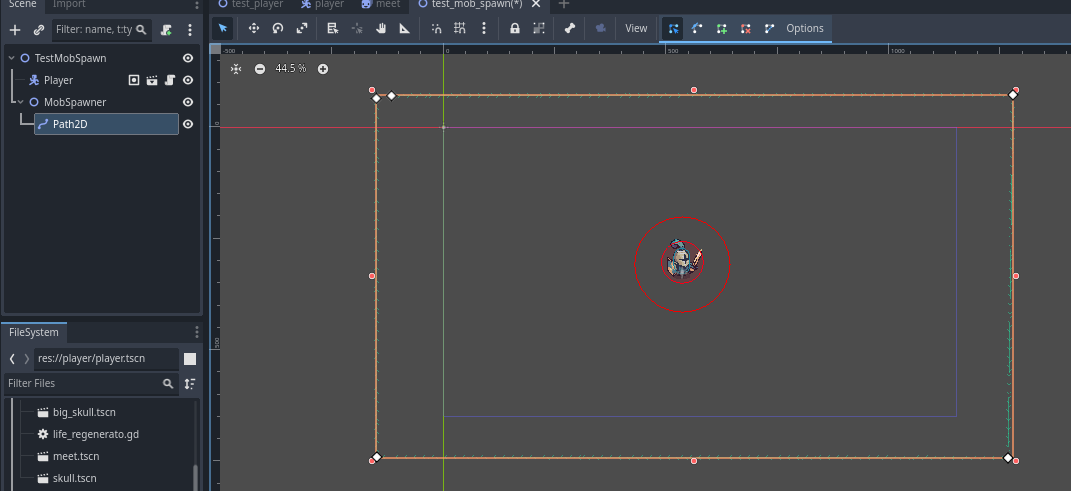


E agora no script da vida é só chamar essa função:



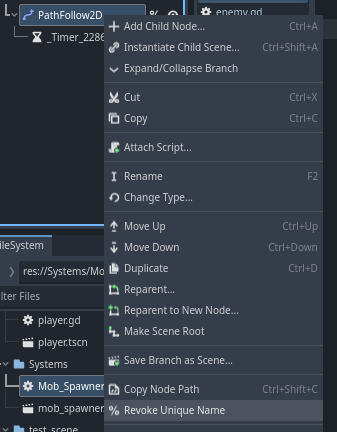
**Criação programática de criaturas:**

Criei uma cena de teste, e nela coloquei meu personagem e um node2d que chamei de mobspawner e movi para exatamente onde o player estar e criei um path2d (que é um caminho para seguir) e com ele desenharemos de onde os monstros irão nascer.

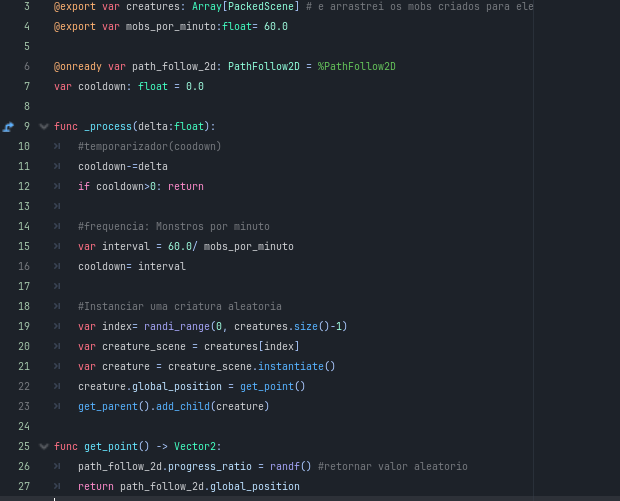


Dentro do path2d criaremos um PathFollow2D = simplesmente uma forma de andar nesse caminho que criamos no path2d.

Assim agora iremos criar um script no mobspawner



Marca para Unique Name, para ter acesso ao mob\_spawner em todo lugar.



**Um toque de magia para nosso jogo:**