



Godot: plataformaPixel

Prof. Thiago Felski Pereira, MSc.

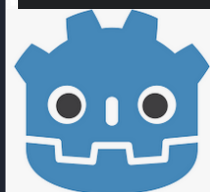
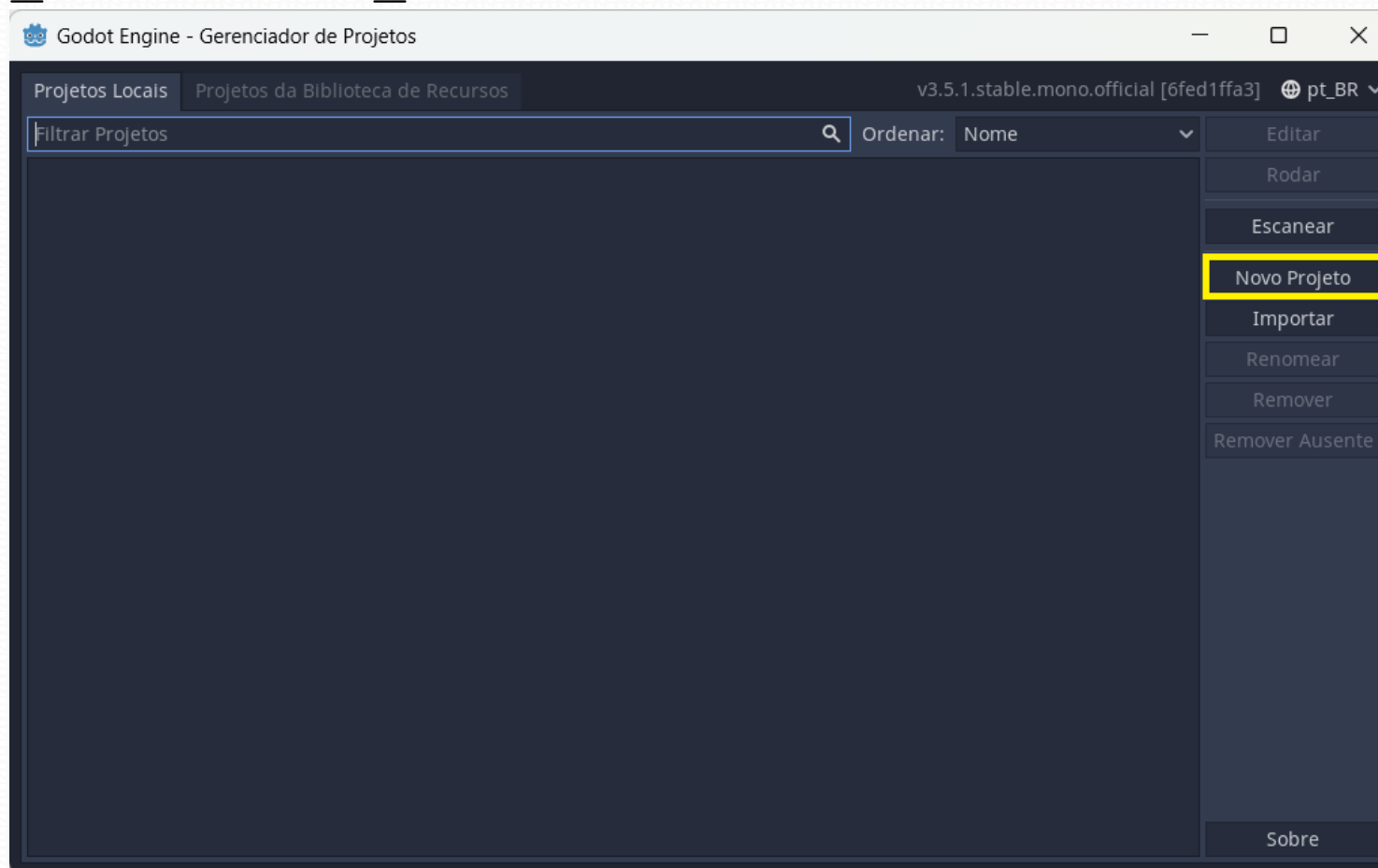
Visão Geral

- Esse será nosso primeiro jogo de plataforma no GODOT
 - Por isso, eu estiver “rápido demais”, não deixe de me informar.
- Com esse jogo iremos
 - Estruturar um projeto de jogo simples
 - Mover o personagem do jogador pelas plataformas
 - Pulando ou caindo delas
- Bibliografia
 - <https://www.kenney.nl/assets/pixel-platformer>
 - <https://youtu.be/f3WGFwCduY0>
 - <https://youtu.be/-9wDi3Y08GM>



Godot: plataformaPixel

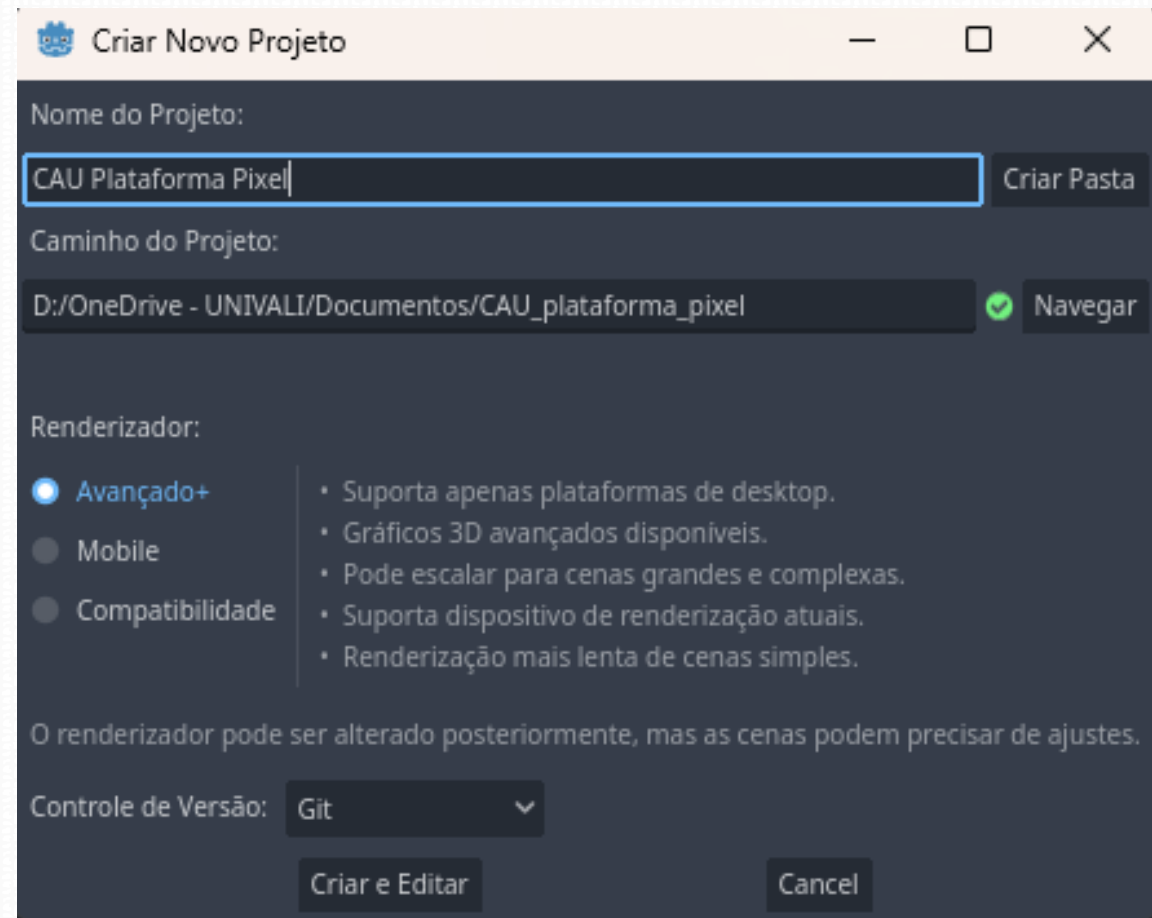
- Clique em **Novo Projeto**
- Nomeie a pasta como: CAU_plataforma_pixel



3

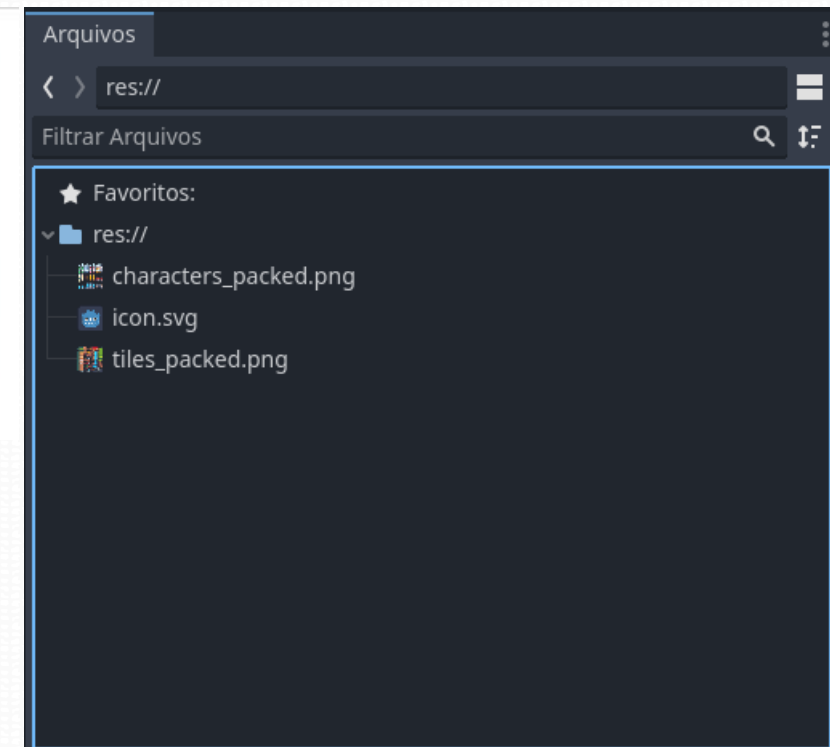
Godot: plataformaPixel

- Clique em **Novo Projeto**
- Nomeie o projeto e crie uma pasta com o mesmo nome:
 - CAU_plataforma_pixel



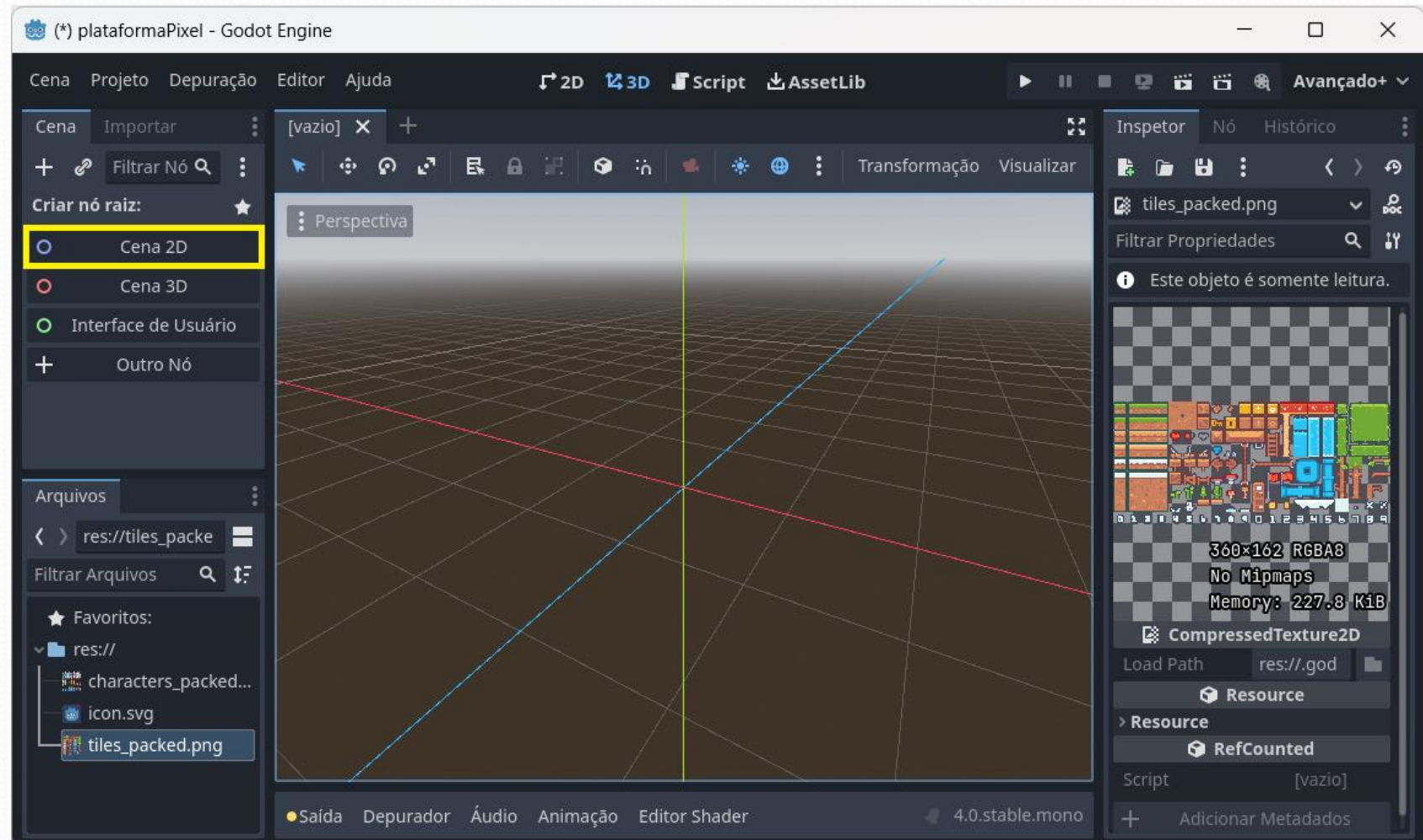
Godot: plataformaPixel

- Como o foco deste tutorial não é criar a arte do jogo, nós iremos apenas baixar uma arte grátis da internet.
 - Para isso, basta fazer o download do pixelplatformer na página do Kenney e arrastar o **character_packed** e **tiles_packed** para nosso projeto.



Godot: plataformaPixel

- Vamos criar uma Cena 2D que será nosso Mundo, ou apenas uma fase em um jogo maior.

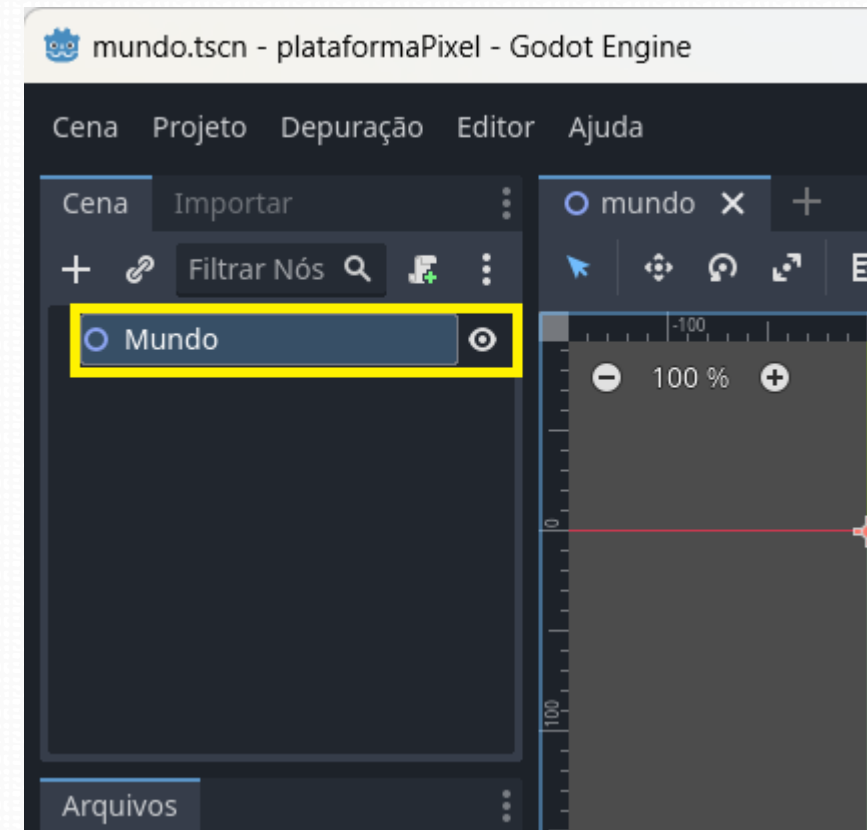
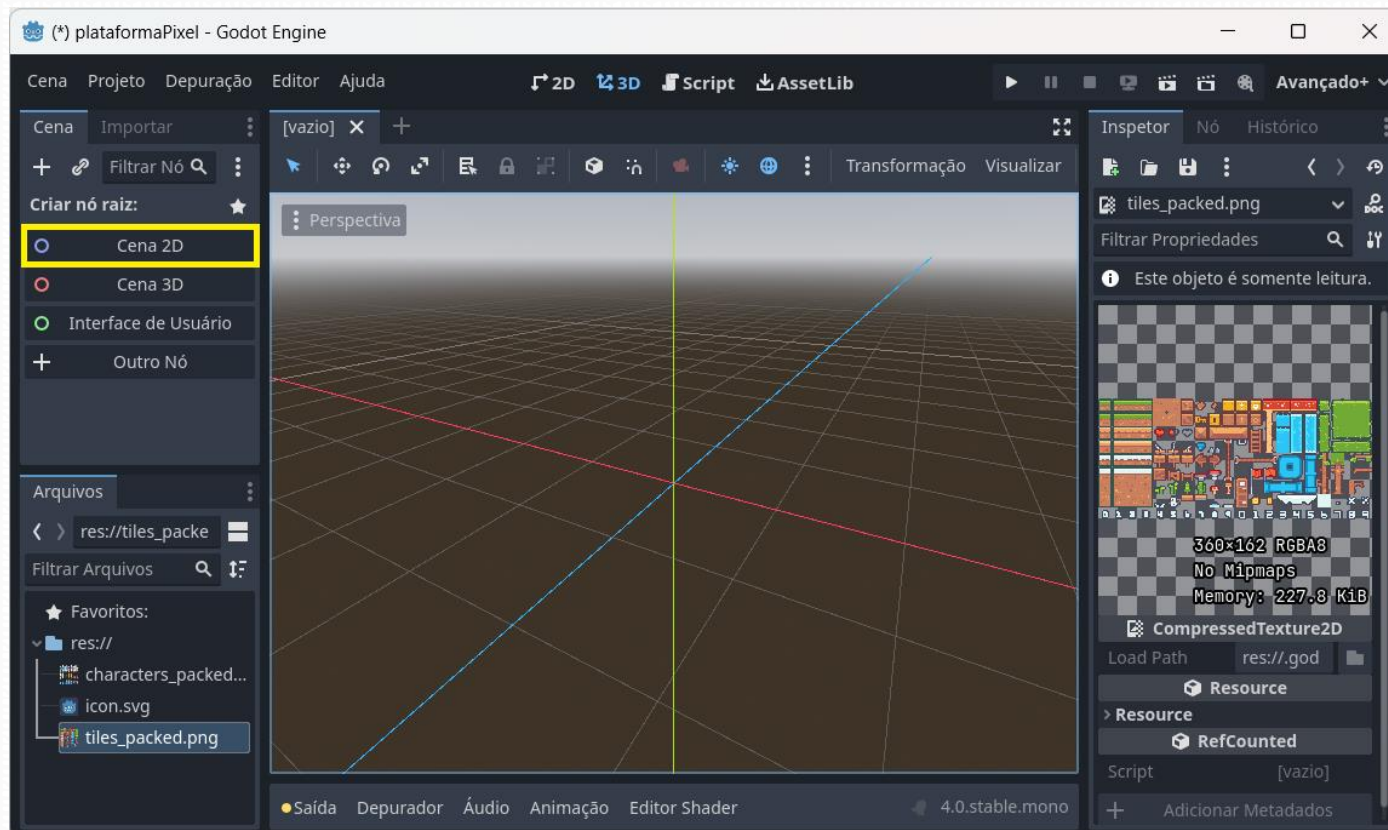


6



Godot: plataformaPixel

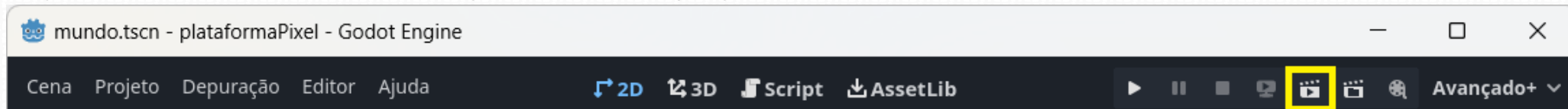
- Vamos criar uma **Cena 2D** que será nosso Mundo, ou apenas uma fase em um jogo maior. Após criada renomeie o **Node2D** criado para **Mundo** e salve a cena.



7

Godot: plataformaPixel

- Apesar de não ter muito para exibir, já podemos executar essa cena



- Apenas para teste arraste o tiles_packed.png para mundo e execute novamente a cena.
 - Note que a imagem aparece na cena.
 - Note também que em mundo aparece um subitem chamado TilesPacked do tipo Sprite2D

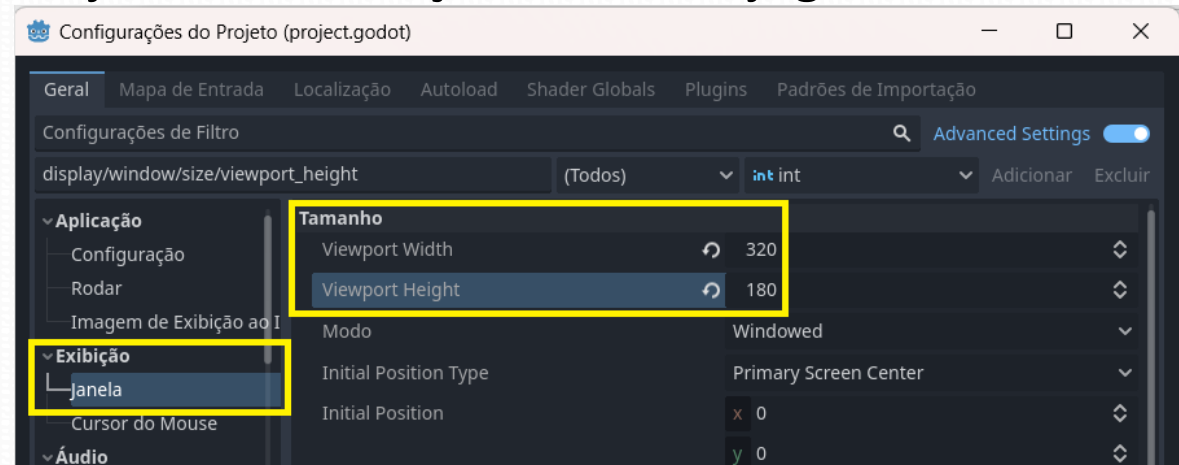


8

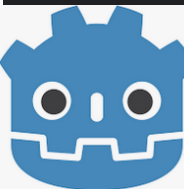
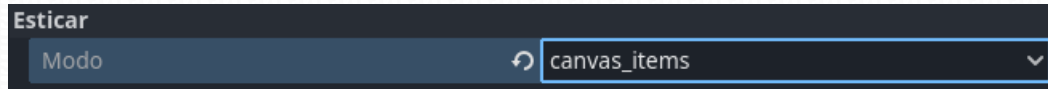
Godot: plataformaPixel

- Vamos usar essa imagem de exemplo para ajustar a resolução do nosso jogo.

- Projeto -> Configurações do Projeto...
- Encontre o menu
 - Exibição -> Janela
 - Defina o tamanho da janela para 320x180

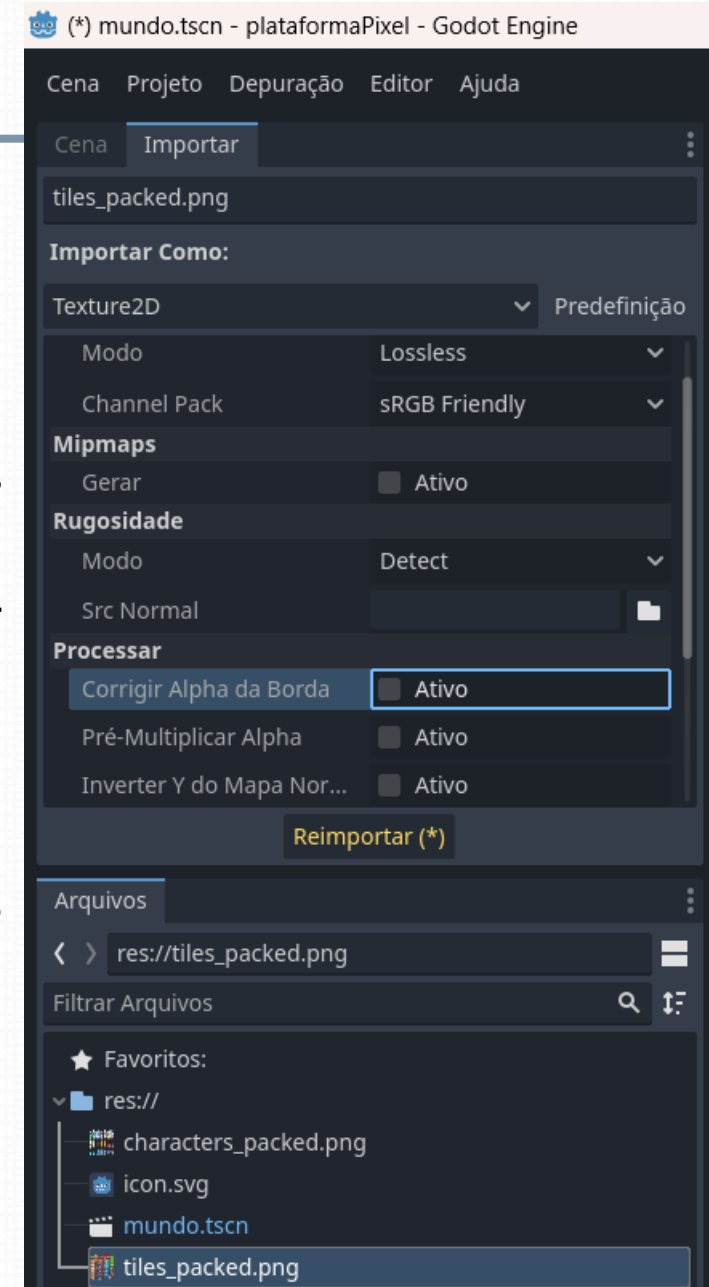


- Com isso ajustamos a resolução no jogo, mas não está bom para trabalharmos com ele, para isso ajuste também
 - Windows ... Overwrite para 960x540, ou seja 3x o tamanho original
 - Essa opção só aparecerá se você habilitar advanced settings
 - Vá no menu Esticar e mude o modo para canvas_items



Godot: plataformaPixel

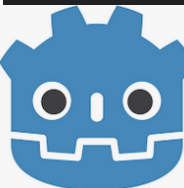
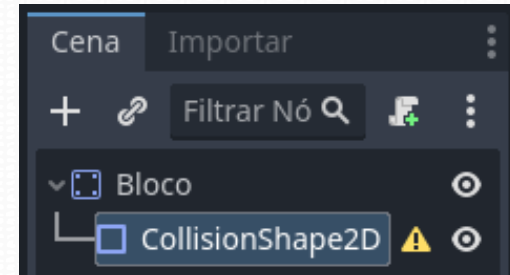
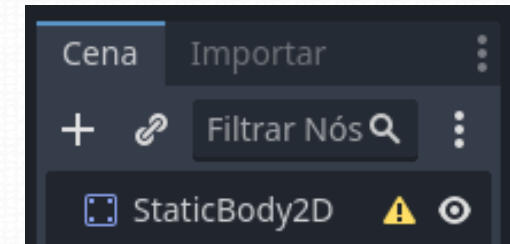
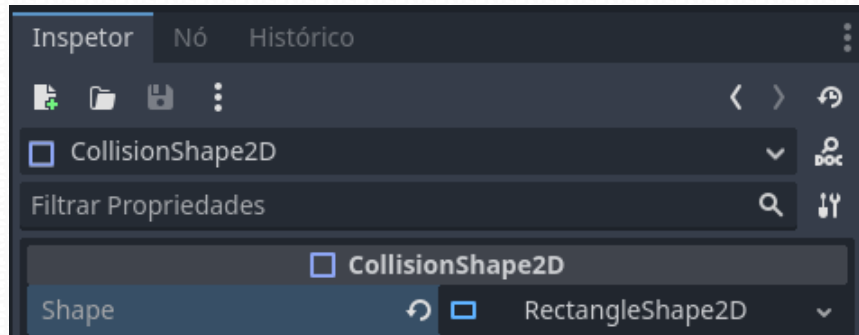
- Note que nossa imagem ficou embaçada, isso não acontece por uma perda de resolução, mas porque o Godot aplica um filtro de “arredondamento de bordas” (Blur)
 - Como isso retira a identidade dos nossos pixels vamos desabilitar esse efeito
 - Selecione a imagem -> Inspetor -> Canvas Item -> texture filter e procure o filtro que ajusta melhor a imagem
 - Execute e veja se nota a diferença
- Agora você já pode deletar o Sprite2D (TilesPacked) de Mundo, pois era só para teste mesmo.



10

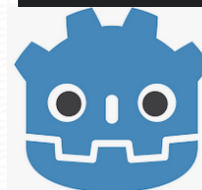
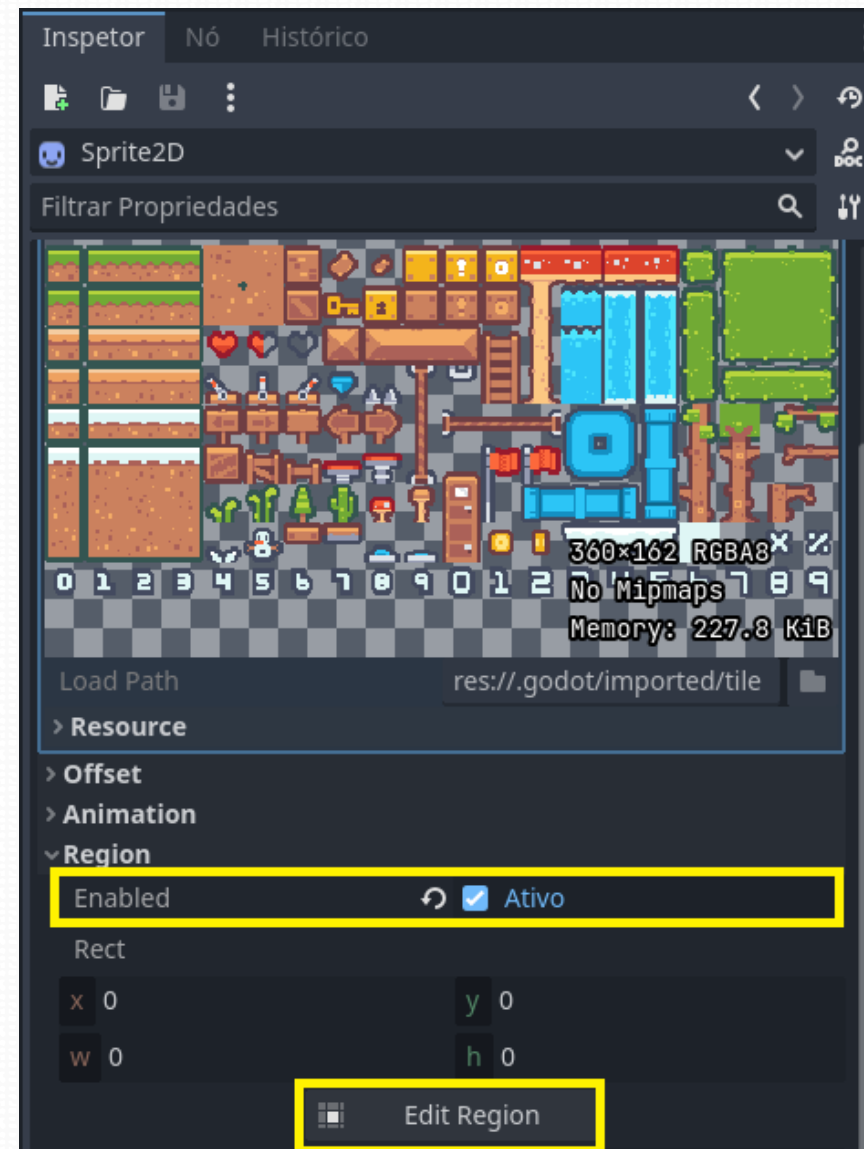
Cena: Bloco

- Crie uma nova cena.
- Instancie um Nó: StaticBody2D.
 - Note que há um alerta indicando que ele não tem forma definida
 - Renomeie o StaticBody2D para Bloco.
 - Adicione um CollisionShape2D como nó filho do Bloco.
 - No inspetor defina o shape como RectangleShape2D.



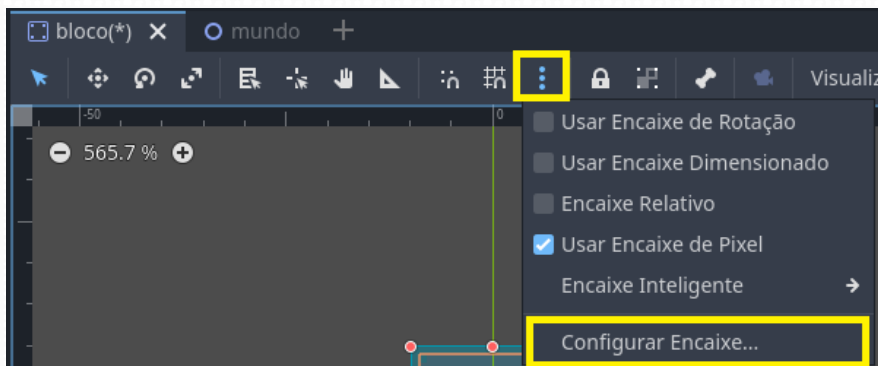
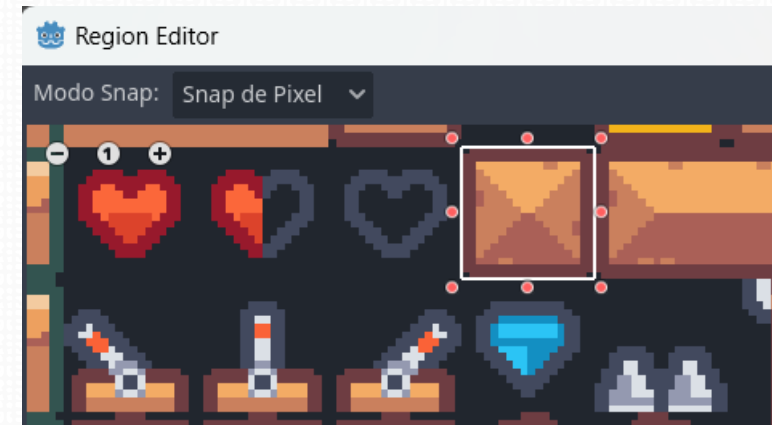
Cena: Bloco

- Adicione uma textura ao seu Bloco.
 - Crie um nó filho para o bloco do tipo **Sprite2D**.
 - Por mais estranho que pareça, arraste o `tiles_packed.png` para o campo texture do **Sprite2D**.
- Definindo uma região específica da textura para seu bloco
 - Inspetor -> Region -> Enabled -> (marque a opção: **Ativo**).
 - Clique em **Edit Region**.



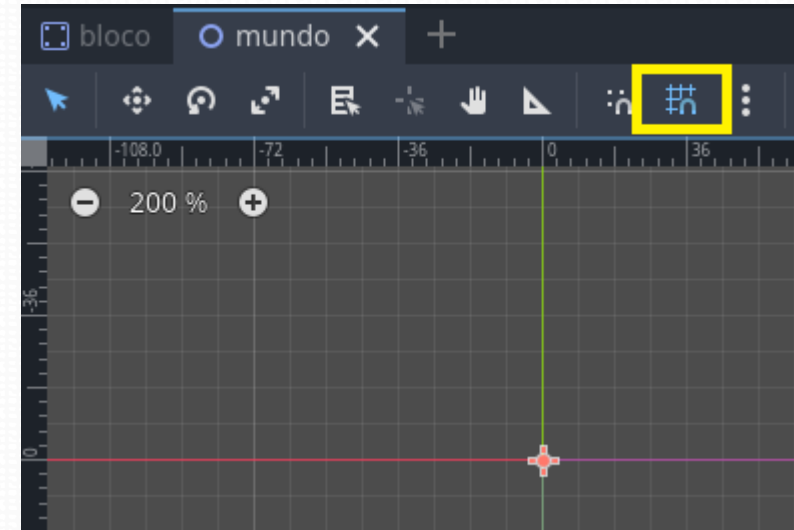
Cena: Bloco

- Na opção **Modo Snap**: escolha **Snap de Pixel**
 - Dê zoom na imagem para facilitar a visualização.
 - Clique e arraste para selecionar a região desejada.
 - Feche o Editor de Região.
- Mova o Sprite2D para cima do CollisionShape2D na hierarquia.
- Os *Assets do Kenney* são todos 9x9, então configure o encaixe para esse tamanho.
 - Ajuste o **ColisionShape2D** para cobrir todo bloco.



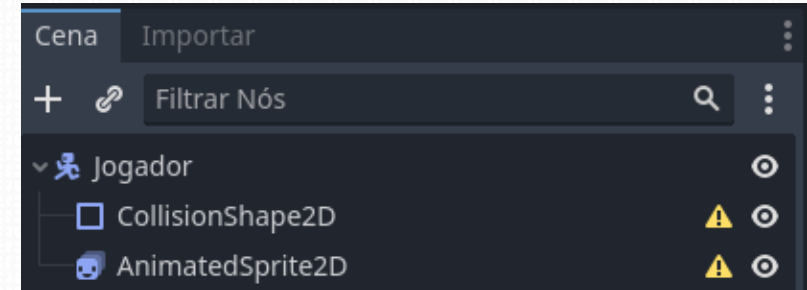
Cena: Mundo

- Configure o encaixe na cena **mundo** também para 9x9.
- Arraste um **bloco** para seu **mundo**.
- Também é interessante ativar a visualização das grades para orientar nosso desenvolvimento.
- Instancie alguns blocos como cena filha do nosso mundo.
 - **Instantiate child Scene.**
 - CTRL+D com um bloco selecionado acelera o processo.
- Execute o jogo para testar.

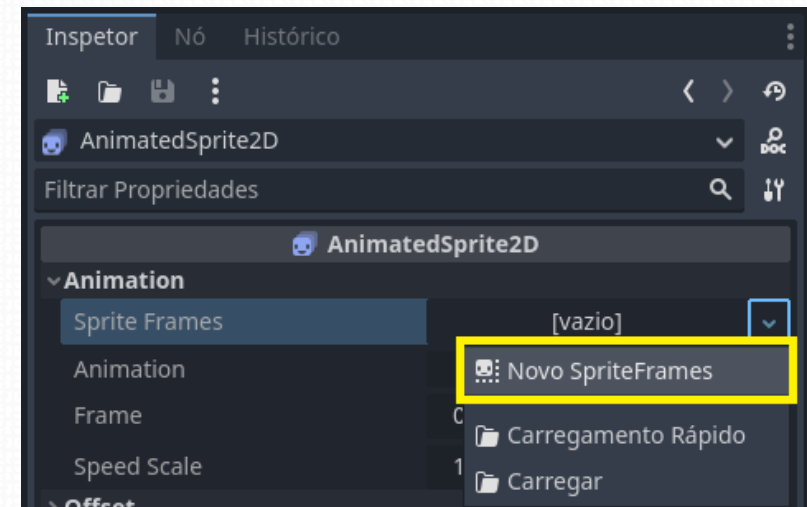


Cena: Jogador

- Nova cena
 - Adicione um CharacterBody2D e renomeie para Jogador
 - Adicione um CollisionShape2D
 - Adicione um AnimatedSprite2D



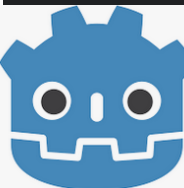
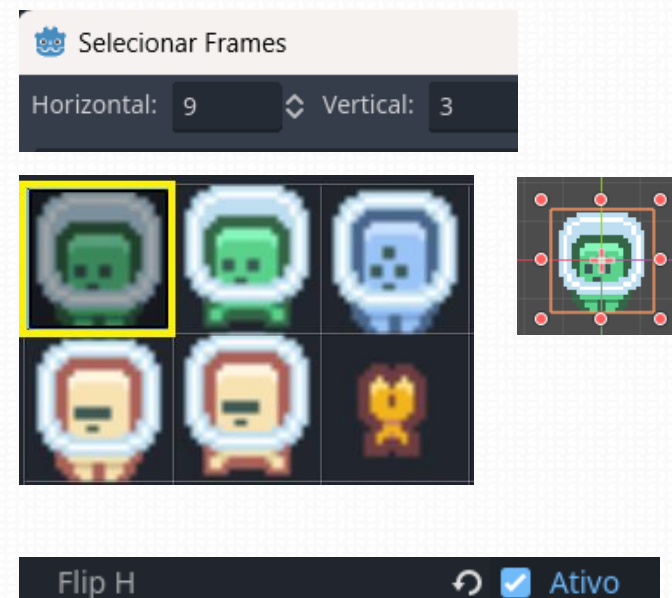
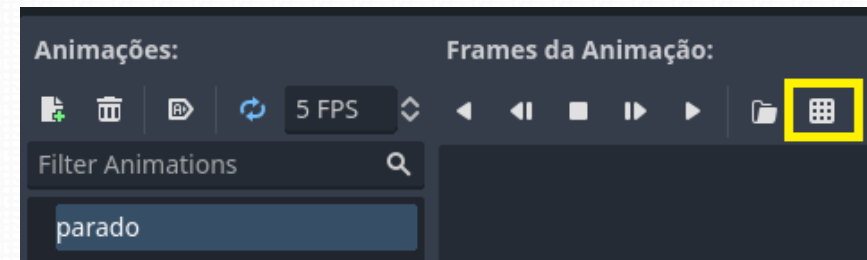
- Vamos adicionar frames ao nosso AnimatedSprite2D
 - Inspector -> Animation -> Sprite Frames -> Novo Sprite Frames
 - Clique no SpriteFrames que você acabou de adicionar para abrir o menu na parte de baixo do editor



Cena: Jogador

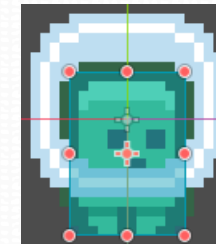
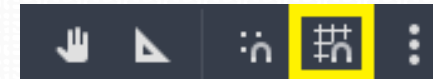
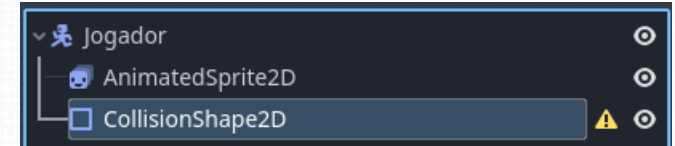
- No menu que abriu na parte inferior: **Frames de Animação**

- Renomeie a animação default para parado.
- Clique no botão de grade para adicionar uma imagem.
- Selecione: `characters_packed.png`
- Ao abrir ajuste as dimensões para 9x3.
- Escolha o primeiro frame.
 - É possível dar zoom na imagem para ajudar na seleção.
- Apesar de aparecer embaçado na seleção de frames, o frame fica bom, pois foi ajustado o filtro.
 - Inspector -> CanvasItem -> Texture -> Filter -> Nearest
- Ajuste também para o personagem olhar pra direita.
 - Inspector -> AnimatedSprite2D -> Offset -> Flip H

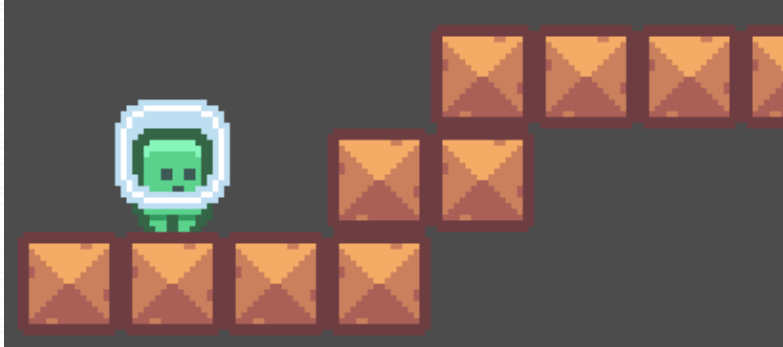


Cena: Jogador

- Ajuste a colisão do personagem.
 - Coloque o **CollisionShape2D** abaixo do **AnimatedSprite2D** na hierarquia.
 - Adicione um **RectangleShape2D** ao **Shape**.
 - Para conseguir ajustar a colisão corretamente é necessário desabilitar o: **Usar Encaixe de Grade**.
 - Ajuste a colisão para cobrir apenas o personagem.
 - Cuide para os pés ficarem bem alinhados com a colisão.
 - Salve a cena Jogador.



Cena: Mundo

- Volte na cena Mundo.
 - Adicione o Personagem ao Mundo como uma Child Scene.
 - Botão direito -> Instantiate Child Scene -> jogador.tscn
 - Posicione o jogador na cena.
 - Desabilite o Encaixe de Grade.
 - Posicione o Personagem.
 - Habilite o Encaixe de Grade.
- 
- The screenshot shows a green alien character with a white helmet standing on a platform made of brown blocks. The blocks are arranged in a stepped pattern, with the character standing on the leftmost block. The background is dark gray.
- Teste o Mundo e veja que o personagem aparece na cena.
 - Observe que ainda não adicionamos gravidade e colisão, portanto os blocos e personagens vão permanecer exatamente onde foram deixados.



Cena: Jogador (Script)

- Adicione um Script ao jogador.
 - Lembre de definir o script como C#.
 - Ao criar você vai notar que já tem um modelo com bastante código incluído, mas não vamos usá-lo agora.
 - Delete o código até ficar assim:

```
1  using Godot;
2  using System;
3
4  public partial class jogador : CharacterBody2D
5  {
6
7  }
```



Cena: Jogador (Script)

- Função chamada quando o objeto entra na cena
 _Ready()
 - Iremos escrever um texto quando o jogador for criado na cena.
 - Útil para testar o código escrito.
 - Salve o código e faça um **Build** do Jogador.
- Vá para cena Mundo e execute a cena.
 - Note que como o jogador foi incluído na cena Mundo ele disparou a função _Ready()

```
using Godot;
using System;

1 reference
public partial class jogador : CharacterBody2D
{
    1 reference
    public override void _Ready(){
        GD.Print("O personagem surgiu no mundo!");
    }
}
```



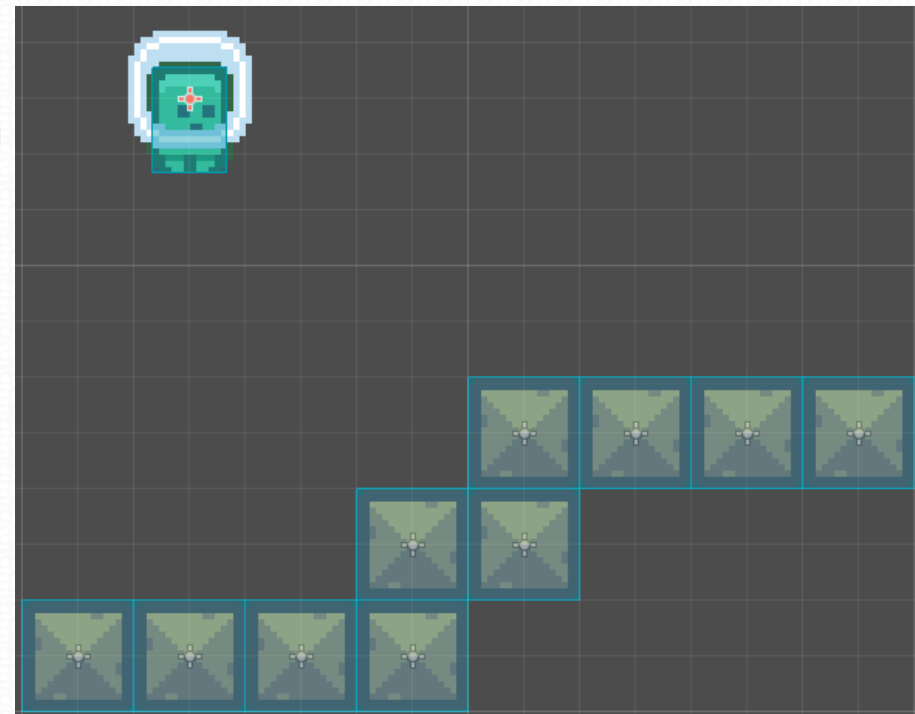
20



Cena: Jogador (Script)

- A função `_Process()` tenta executar cada frame o mais rápido que puder, já `_PhysicsProcess()` irá executar em tempos (delta) regulares, sendo mais indicados para o que precisamos.
- Ajuste o Jogador no Mundo para ficar em queda livre.
- Teste seu código.

```
private float gravidade = (float)0.05;  
1 reference  
public override void _PhysicsProcess(double delta)  
{  
    Velocity = Velocity + new Vector2(0,gravidade);  
  
    MoveAndCollide(Velocity);  
}
```



Cena: Jogador (Script)

- DESAFIO
 - Veja se consegue mover o personagem para esquerda e para direita.



(22)



UNIVALI



(23)



UNIVALI