



Godot: TileMapLayer

Prof. Thiago Felski Pereira, MSc.

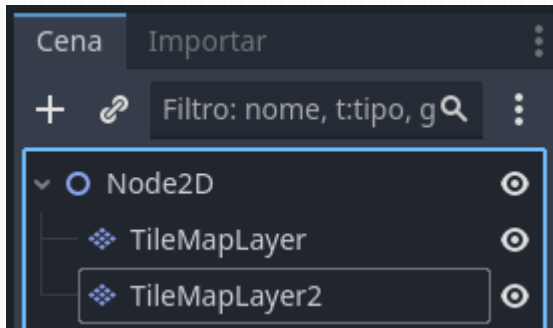
Visão Geral

- Desenhar mapas para jogos de plataforma seria uma tarefa um tanto frustrante para o artista se não tivéssemos recursos que facilitam a produção artística.
 - E não estamos falando de recursos como `ctrl+c` / `ctrl+d` para duplicar um ou mais blocos e posicioná-los individualmente.
- Um mapa de blocos é uma grade de blocos usada para criar o layout de um jogo que traz vários benefícios.
 - Primeiro, eles tornam possível desenhar o layout “pintando” os blocos em uma grade
 - Segundo, eles permitem níveis muito maiores porque são otimizados para desenhar um grande número de peças.
 - Finalmente, você pode adicionar formas de colisão, oclusão e navegação aos blocos, adicionando maior funcionalidade ao TileMap.
- Para entender melhor o conceito vamos fazer na prática.

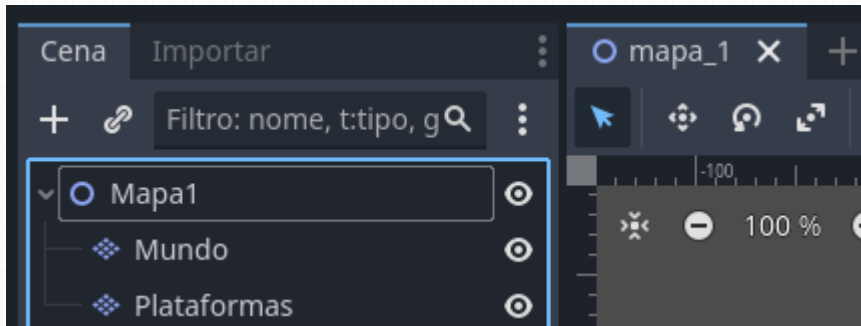


Cena: Mapa1

- Foi criada uma cena Mapa1 conforme estrutura apresentada, a seguir.

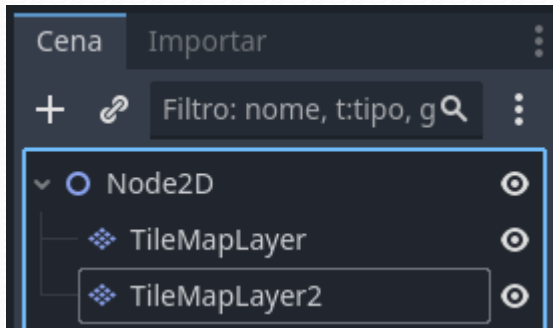


- Os nomes então foram ajustados para facilitar a leitura:

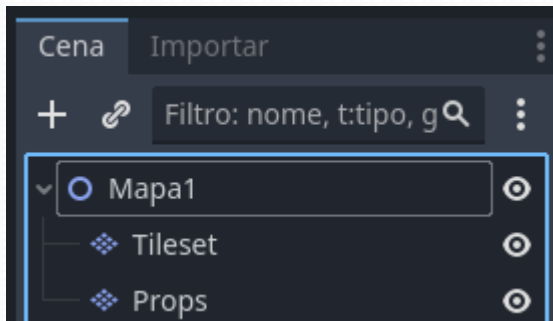


Cena: Mapa1

- Foi criada uma cena Mapa1 conforme estrutura apresentada, a seguir.

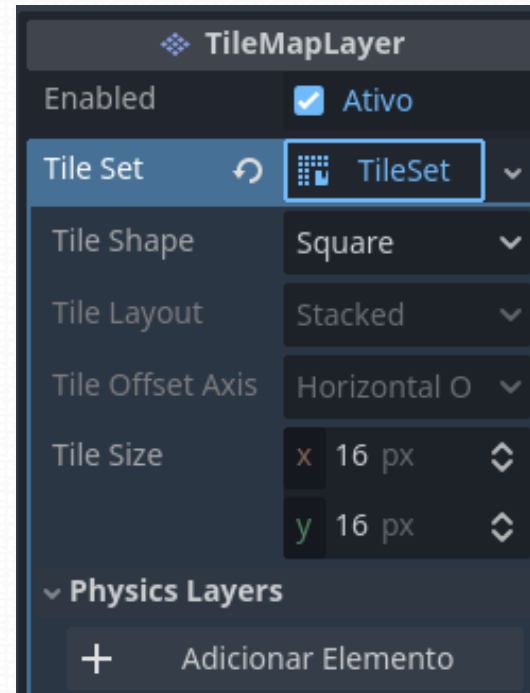


- Os nomes então foram ajustados para facilitar a leitura.
 - Salvamos a cena com o nome Mapa1.



Cena: Mapa1 -> Tileset

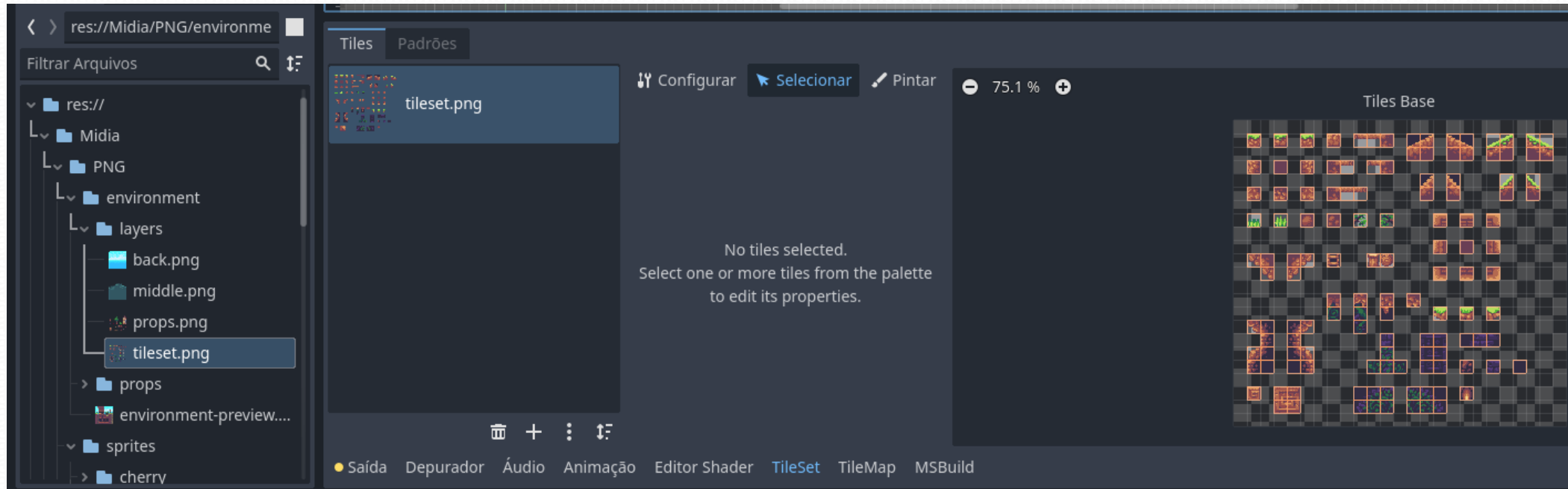
- Configuramos o Tileset no Inspetor.
 - Primeiro adicionamos um novo Tileset.
- Em seguida adicionamos uma camada de física.



5

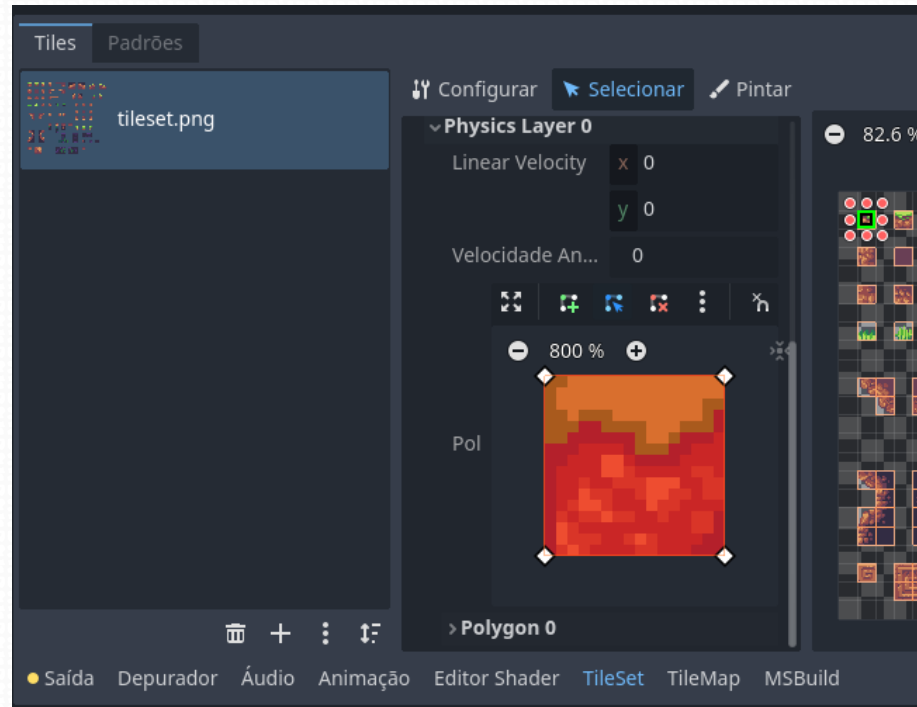
Cena: Mapa1 -> Tileset

- Arraste tileset.png para o quadro de Tiles na guia inferior da ferramenta.
 - Deixe que a própria ferramenta faça as definições automaticamente.



Cena: Mapa1 -> Tileset

- Selecione **um ou mais** Blocos do TileSet e defina a área de colisão em:
 - Física -> Physics Layer 0 -> Polygons -> (aperte F para selecionar toda área)



- A imagem mostra o processo sendo feito com apenas um Bloco, mas é possível selecionar vários de uma vez para acelerar o processo.

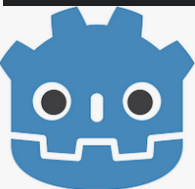


Atividade

- Desenhe a cena Mapa1
 - Selecione um ou mais blocos no TileMap da cena Mapa1.
 - Para desenhar vários blocos livremente: basta manter o botão esquerdo do mapa apertado.
 - Para desenhar uma linha de blocos: basta manter o botão esquerdo do mapa e o shift para formar uma linha.
 - Para desenhar quadrados de blocos use ctrl+shift.
 - Para apagar algo que não gostou, basta utilizar a opção de borracha no TileSet.
 - Explore as ferramentas de desenho do TileMap.

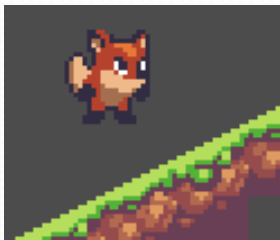


- Requisitos
 - O mapa deve ser maior do que uma tela de visão e deve ser completo e não apenas uma quantidade de elementos dispersos na tela.

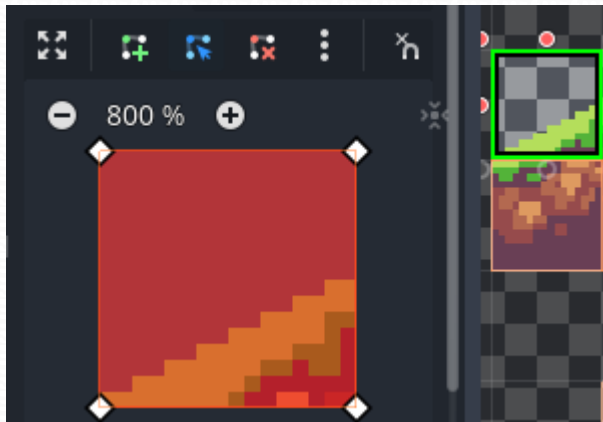


Cena: Mapa1 -> Tileset (Rampas)

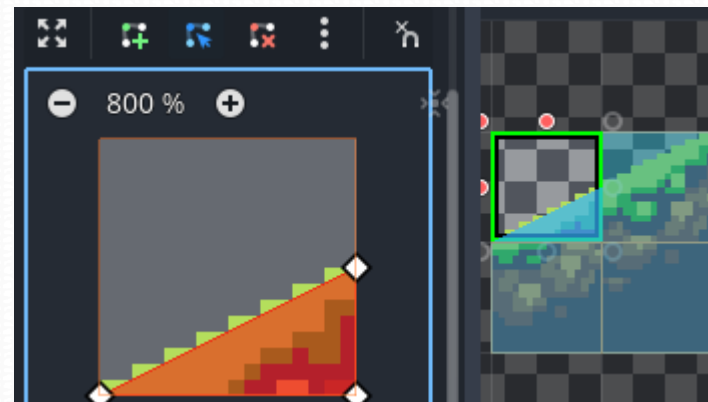
- **Vamos resolver o problema das rampas.**
- Observe que o personagem fica flutuando em cima da rampa.



- Isso ocorre porque nossa colisão não está ajustada para o formato da imagem.
 - Para ajustar, iremos utilizar o menu acima da área de colisão para ajustar sua forma.
 - Antes



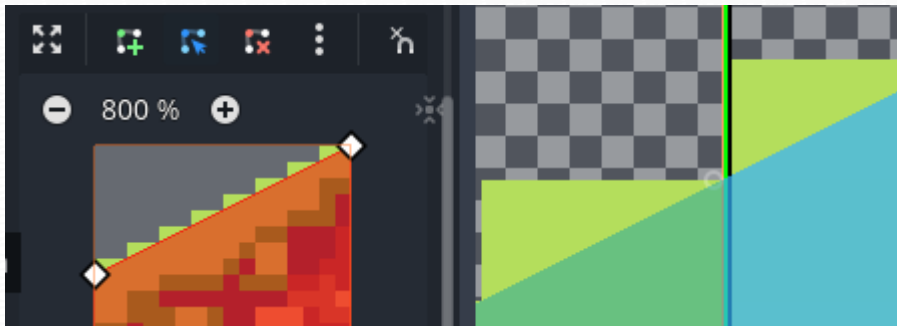
Depois



9

Cena: Mapa1 -> Tileset (Rampas)

- **Vamos resolver o problema das rampas.**
- No processo de desenhar tem 2 problemas que podem acontecer.
 - **Primeiro:** o personagem ainda não sobe a rampa, pois ele bate nas divisórias.
 - Resolve-se isso colocando bastante zoom para verificar as conexões e fazer ajustes.



- **Segundo:** o personagem desce a rampa como se estivesse descendo degraus de uma escada, pois ele cai cada vez que desconecta do pixel da rampa.
 - Resolve-se isso ajustando a propriedade **Floor** no **Inspetor** do personagem.



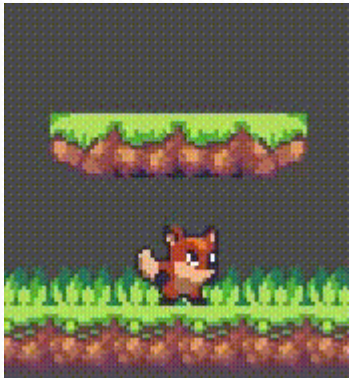
Cena: Mapa1 -> Tileset (Rampas)

- **Vamos resolver o problema das rampas.**
- No processo de desenhar tem 2 problemas que podem acontecer.
 - **Segundo:** o personagem desce a rampa como se estivesse descendo degraus de uma escada, pois ele cai cada vez que desconecta do pixel da rampa.
 - Resolve-se isso ajustando a propriedade **Floor** no **Inspetor** do personagem.
 - **Max Angle:** indica a inclinação máxima que o personagem consegue subir sem pular.
 - **Snap Length:** indica a quantidade de pixels que o personagem vai deslizar antes de cair. No meu exemplo, ajustar essa propriedade para 3 pixels deu o resultado desejado.

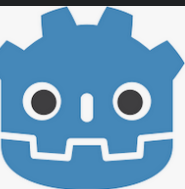


Cena: Mapa1 -> Tileset (atravessando)

- Em jogos de plataforma é comum existirem plataformas que permitem atravessá-las de baixo para cima.



- Para que isso seja possível, basta alterar o TileSet dos blocos que deseja habilitar essa funcionalidade.
 - TileSet -> Selecione o bloco desejado -> Física -> Polygon 0 -> Marque a opção De uma forma.

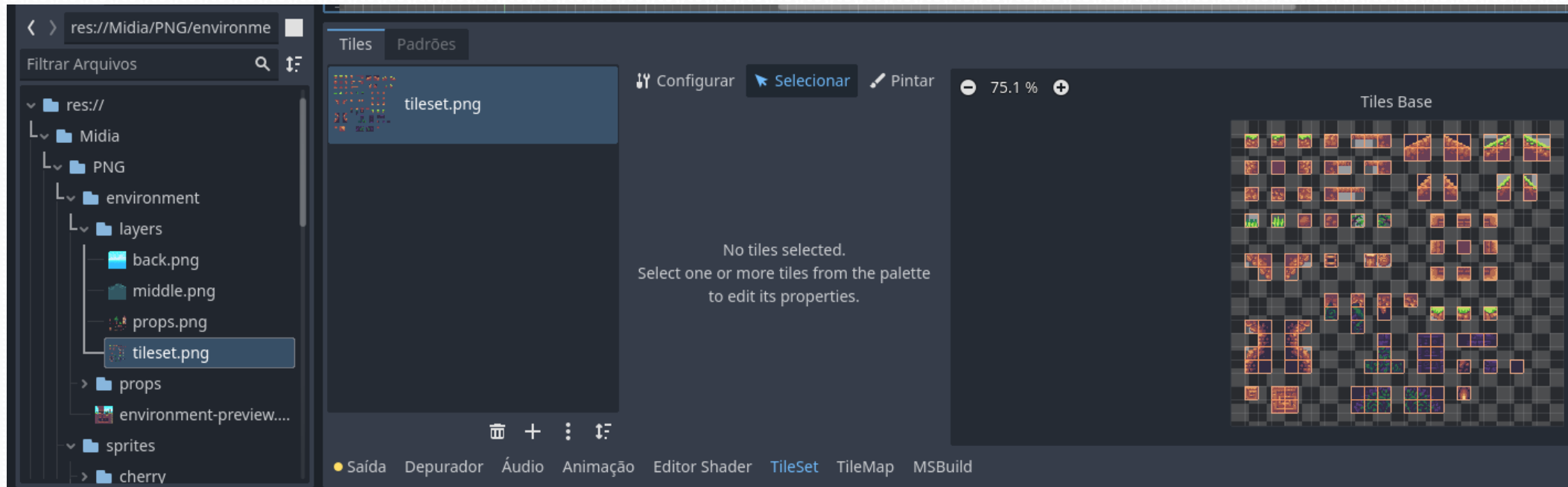


Cena: Mapa1 -> Tiles (animados)

- Tiles animados são parecidos com Tiles normais, a diferença está no momento de adicionar o tileset. Usaremos o seguinte tile para demonstrar a funcionalidade.

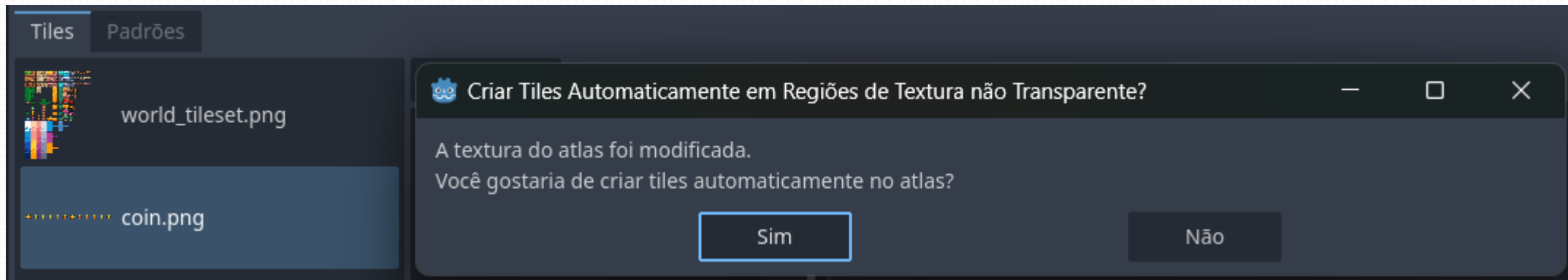


- No mesmo quadro que você já tem o Tileset, adicione o tileset de moedas.



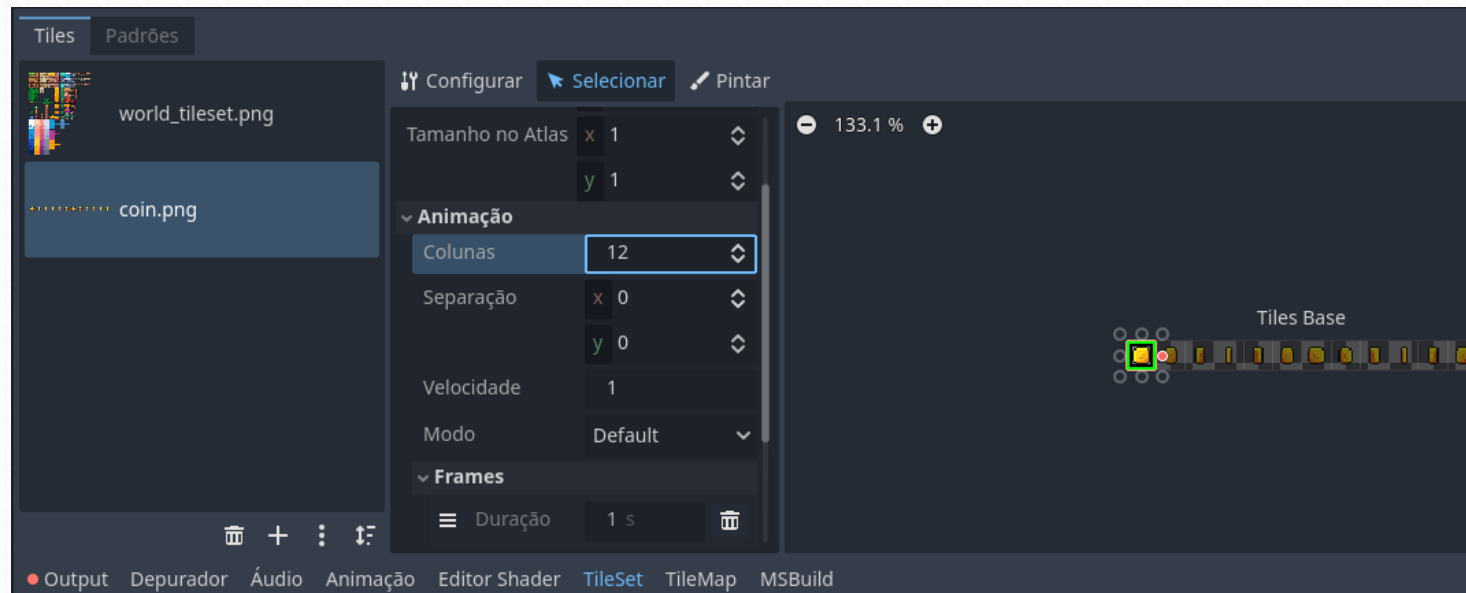
Cena: Mapa1 -> Tiles (animados)

- Mas no momento que ele oferecer para criar os tiles, selecione **NÃO**.
 - Isso permitirá a seleção manual dos tiles, necessário para animação.



Cena: Mapa1 -> Tiles (animados)

- Selecione o primeiro frame da moeda.

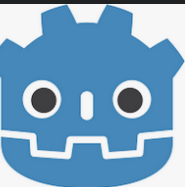
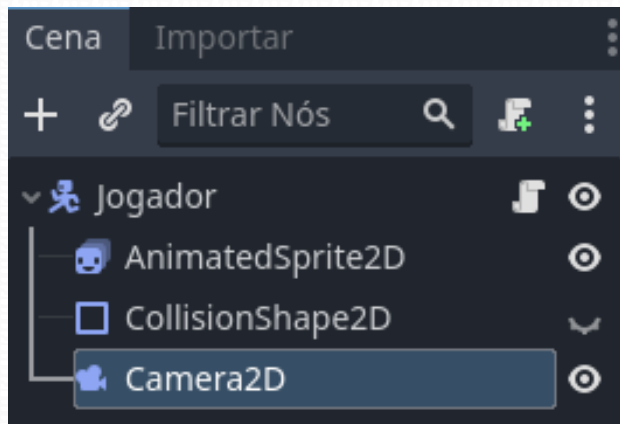


- Ajuste o número de colunas para 12, pois esse é o número de frames dessa moeda.
- Depois disso adicione Frames até chegar em 12.
 - Note que você pode definir a duração de cada frame.



Cena: Jogador

- Agora que nossos mapas terão um tamanho maior do que nos nossos testes iniciais, iremos adicionar uma Camera2D ao Jogador.
 - Adicione um Nó filho ao Jogador do Tipo Camera2D.



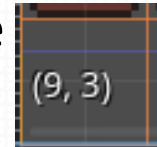
Cena: Area2D + TileMapLayer

- Removendo um Tile quando o Jogador colidir com ele.
- **01:** Coloque uma Area2D com os sinais de colisão ativos na área que o jogador vai colidir.

- Coloquei nas posições das moedas.



- **02:** Com o TileMapLayer selecionado deixe o mouse sobre a área do Tile que deseja remover.



- **03:** Use [Export] para mandar a coordenada e o TileMapLayer para o código da Area2D, além disso, inclua o tipo moeda.

```
[Export(PropertyHint.Enum, "vida,dano,morte,moeda")] public string tipo = "dano";  
4 references  
[Export] TileMapLayer mapa;  
4 references  
[Export] Vector2I coordenada;
```

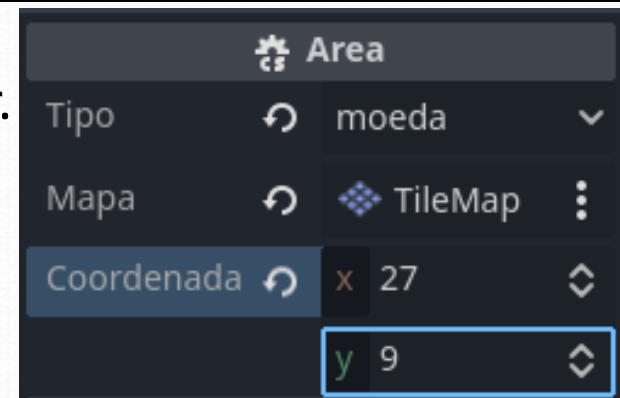


Cena: Area2D + TileMapLayer

- Removendo um Tile quando o Jogador colidir com ele.
- **04:** Programe uma condição que quando o jogador colidir com a área de moeda. Tanto a área quando a moeda deixarão de existir, lembre-se de criar uma função para somar essa moeda ao jogador.

```
if (tipo == "moeda") {  
    ((Jogador)corpo).Moeda(1);  
    QueueFree(); //deleta a Area2D  
    mapa.EraseCell(coordenada); //deleta o Tile da Moeda  
}
```

- 05: Lembre-se de colocar os valores no Inspetor.



Atividade

- **ATIVIDADE1** – Crie um Tile que deixa de existir um tempo após o jogador pisar em cima.
 - Utilize timer para esperar antes de deletar o Tile.
 - Ajuste a Area um pouco acima do tile, senão o jogador pode pisar no Tile sem entrar na área.
 - Lembre-se de controlar o tempo no `_Process` da `Area2D`.
- **ATIVIDADE 2** – Faça o tile reaparecer um tempo depois.
 - Utilize outro timer que inicia quando o primeiro finaliza.
 - Utilize o comando: `SetCell(Vector2I Coord_jogo, int Fonte, Vector2I coord_atlas, int Alternativa)`
 - Exemplo da imagem:

```
mapa.SetCell(coordenada, 0, new Vector2I(1,6), 0);
```

