Godot: SunnyLand: TileMap avançado

Prof. Thiago Felski Pereira, MSc.

Visão Geral

- No Platforma2D nós vimos os aspectos básicos do desenvolvimento de um jogo de Plataforma.
- Agora iremos repassar aspectos de um jogo de plataforma2D tentando aplicar conceitos mais avançados que exigirão:
  - Maior conhecimento dos recursos do GODOT.
  - Mais habilidades de programação.

- Bibliografia
  - https://ansimuz.itch.io/sunny-land-pixel-game-art
  - https://godotengine.org/download/windows/
    - Esse tutorial foi feito na versão 4.1 do Godot (C#)

UNIVALI

Visão Geral

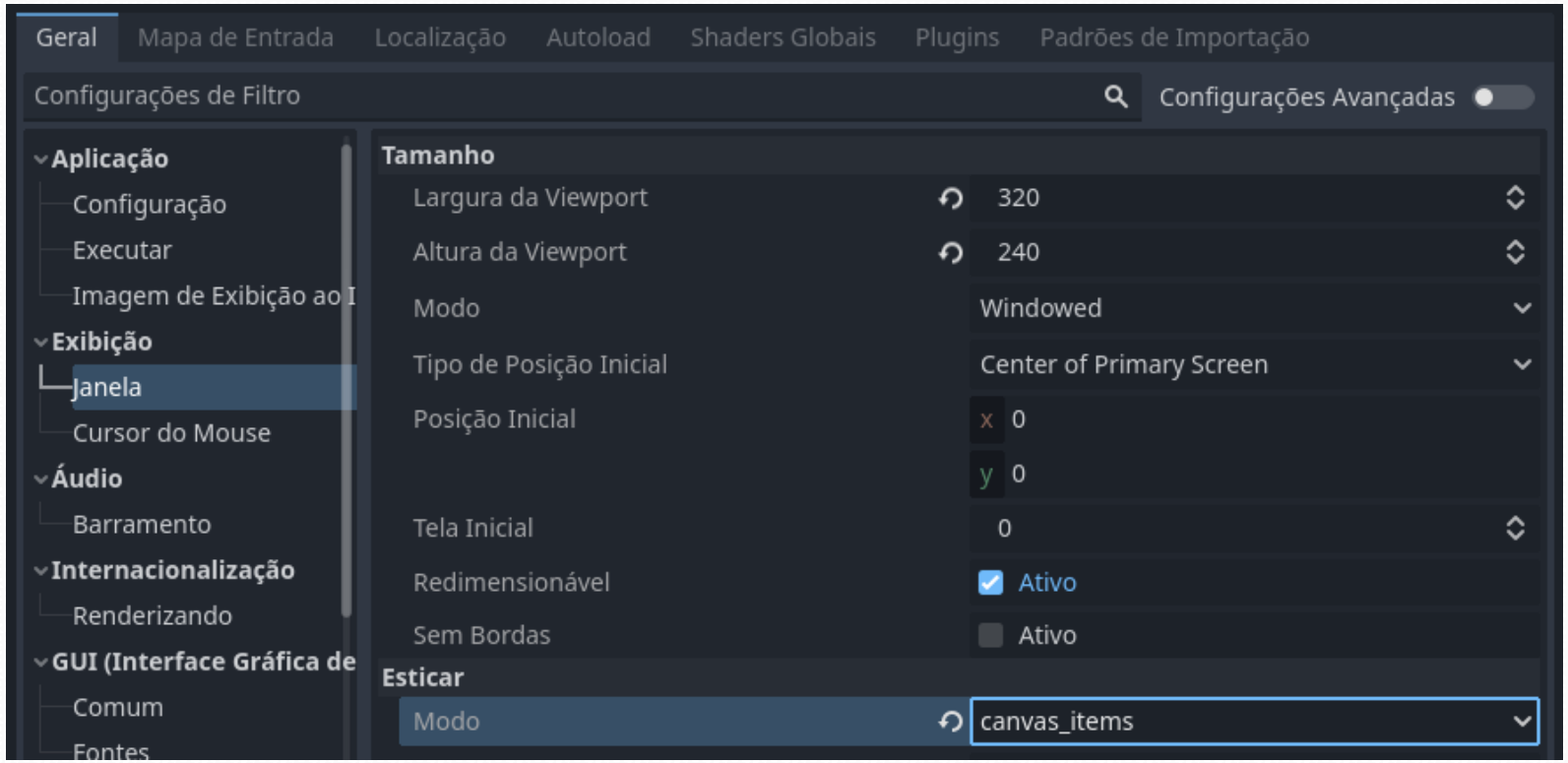
- Mapa de tijolos (`TileMap`)
  - Revisão do conteúdo.
  - Conteúdo avançado com `TileMap`.

- Bibliografia
  - https://ansimuz.itch.io/sunny-land-pixel-game-art
  - https://godotengine.org/download/windows/
    - Esse tutorial foi feito na versão 4.1 do Godot (C#)

UNIVALI

Configurações Iniciais

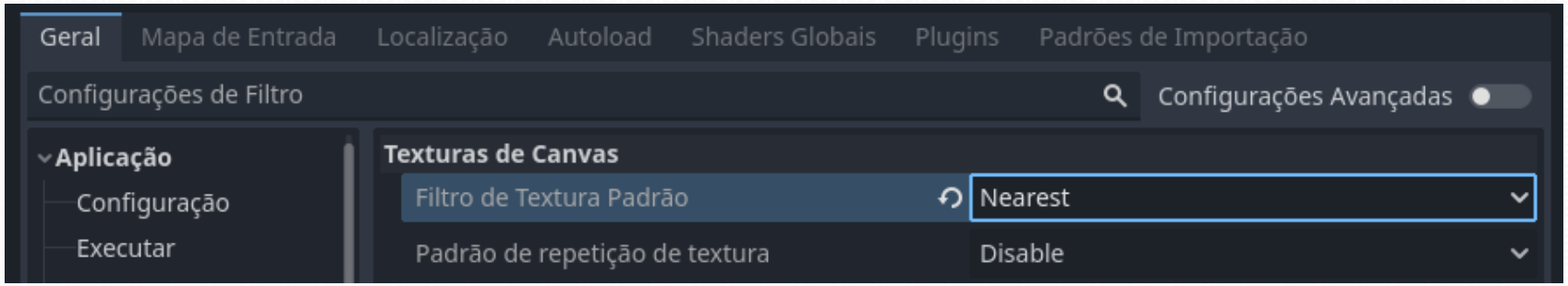
- Projeto -> Configurações do Projeto...
  - Geral -> Exibição -> Janela



UNIVALI

Configurações Iniciais

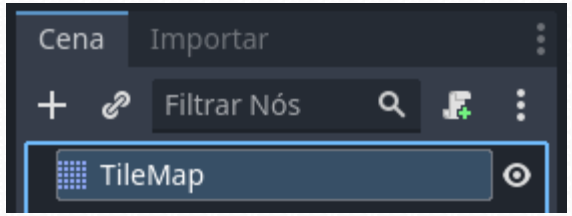
- `Projeto -> Configurações do Projeto...`
  - `Geral -> Renderização -> Textura`



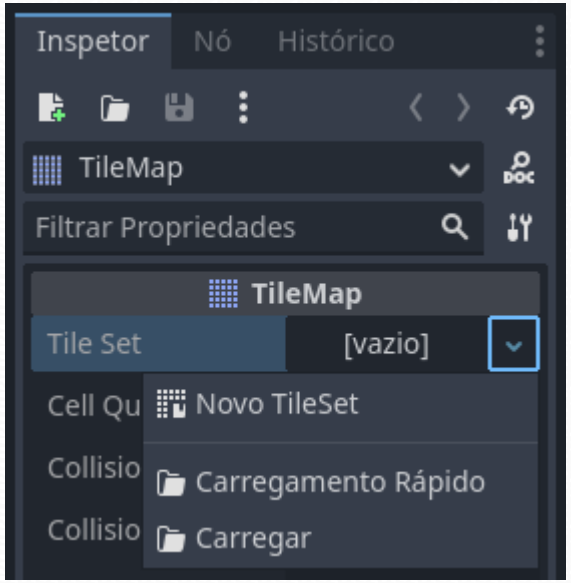
UNIVALI

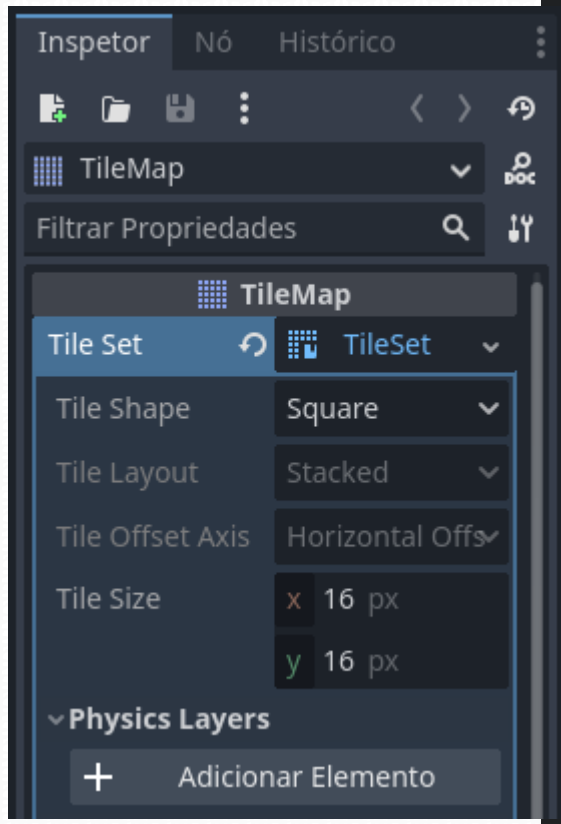
Tile Map

- Crie uma cena `TileMap`
  - Essa cena irá permitir que criemos mapas de forma mais dinâmica do que adicionar física a cada estrutura que se deseja inserir.



- Vá ao `Inspetor`:
  - Adicione um `Novo TileSet` ao seu `TileMap`.
  - Adicione uma camada de física ao seu `TileSet` (`+ Adicionar Elemento`)

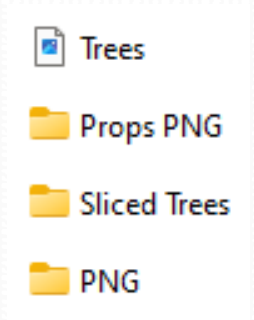


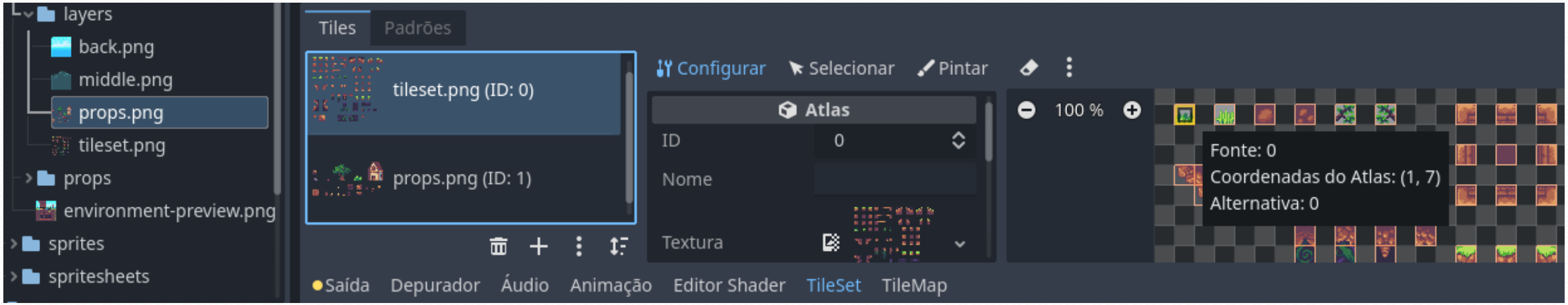


UNIVALI

Tile Map

- Adicione imagens baixadas do Sunny Land a pasta do seu projeto:
  - https://ansimuz.itch.io/sunny-land-pixel-game-art
- É recomendável deixar os arquivos baixados em uma única pasta.
  - Exemplo: Midia.
- O `TileSet` mostra algumas informações importantes que requerem nossa atenção.
  - Cada Tile possui um ID único, sendo tileset.png (ID: 0) e props.png (ID: 1).
  - Além disso o primeiro tile do ID: 0 possui coordenadas (1, 7) no Atlas e cada Tile possuirá sua própria coordenada no atlas (consulte deixando o mouse sobre o Tile)

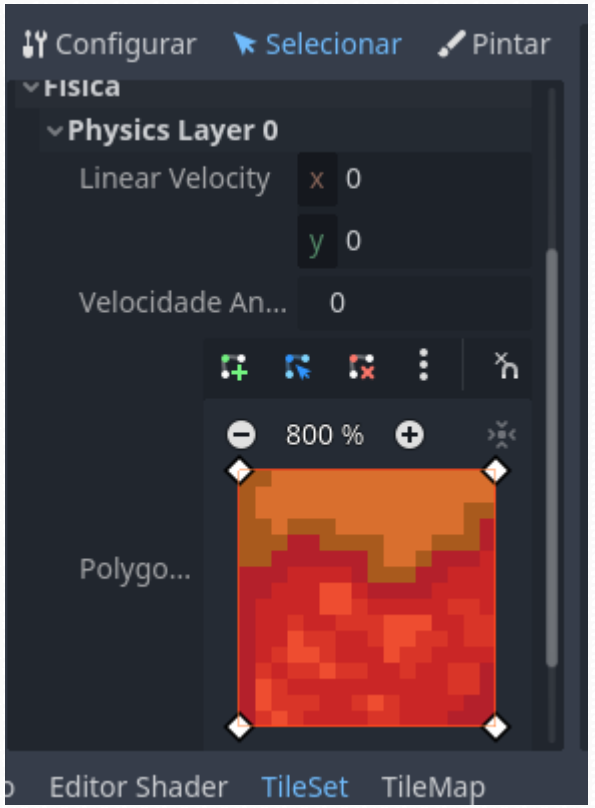




UNIVALI

Tile Map

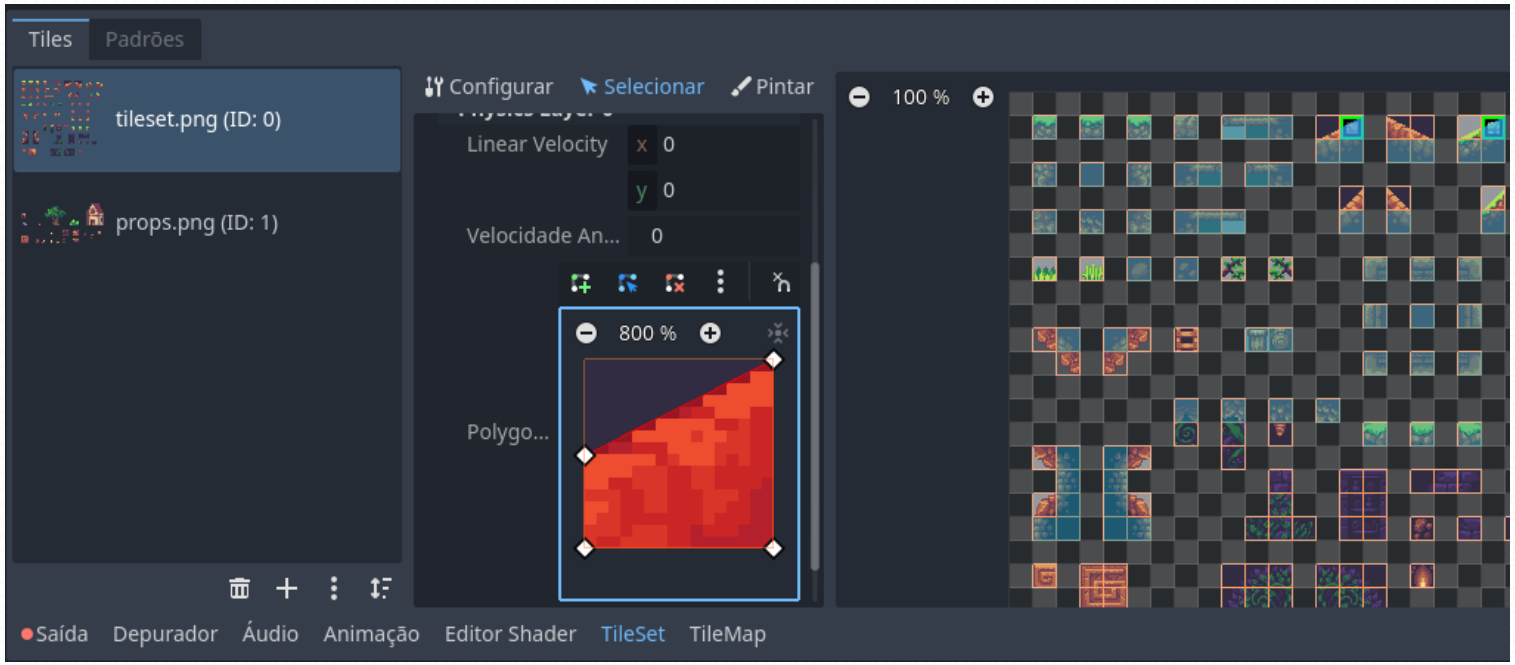
- Selecione um ou mais Tiles no seu atlas para editar suas configurações.
  - Nesse exemplo foram adicionadas colisões aos blocos onde a colisão ocupa o bloco inteiro, para isso bastou ir no `Physics Layer 0` e apertar F (*fill*, preencher).



UNIVALI

Tile Map

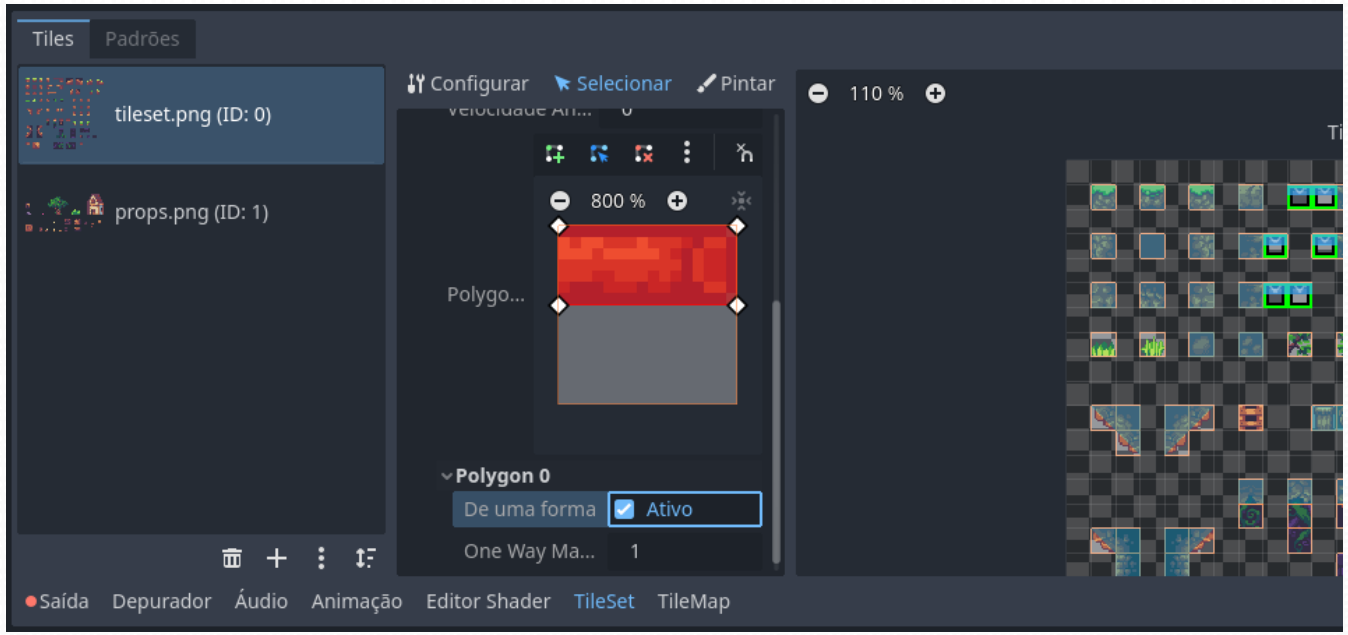
- Selecione um ou mais Tiles no seu atlas para editar suas configurações.
  - Nesse outro exemplo foram adicionadas colisões aos blocos onde a colisão ocupa o bloco inteiro, para isso bastou ir no `Physics Layer 0` e apertar F (*fill*, preencher). E depois disso os pontos foram ajustados para colisão na forma de rampa.
  - Note que é possível adicionar e remover pontos para ajustar as colisões a sua necessidade.



UNIVALI

Tile Map

- Selecione um ou mais Tiles no seu atlas para editar suas configurações.
  - Já nesse exemplo, além de ter editado a área de colisão para os blocos indicados na imagem. Também foi marcado o `Polygon 0 -> De uma forma` como ativo.
    - Isso torna possível atravessar o bloco debaixo para cima, mas colida de cima para baixo.
    - Permitindo o jogador ficar em cima do bloco, mas não colidindo com ele quando pular debaixo.

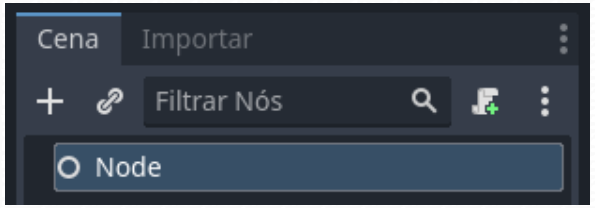


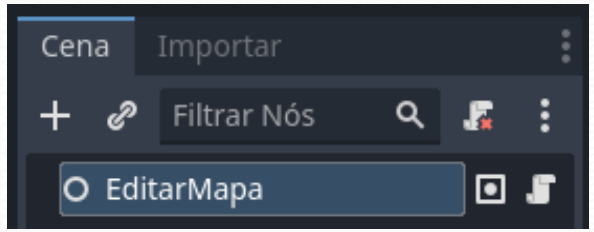
UNIVALI

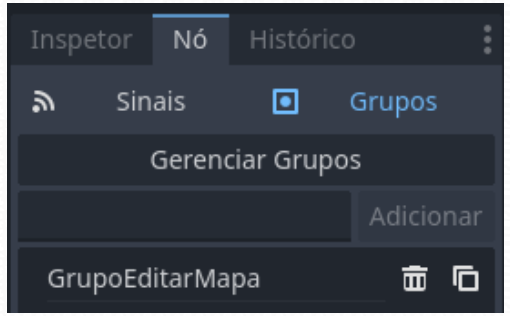
Cena: Editar TileMap em código

- Um `TileMap` pode ser modificado via script.
- Isso permite que algumas interações fiquem mais fáceis de serem implementadas.
  - Remover Blocos de Coordenadas específicas.
  - Adicionar Blocos específicos em coordenadas específicas.
- Com isso podemos criar condições para pontes serem içadas ou abaixadas entre outras interações.

- Primeiro crie a seguinte estrutura.
  - Crie um Nodo simples.
  - Renomeie o Nodo.
  - Adicione um script a ele.
  - Crie um grupo para chamarmos os scripts.







UNIVALI

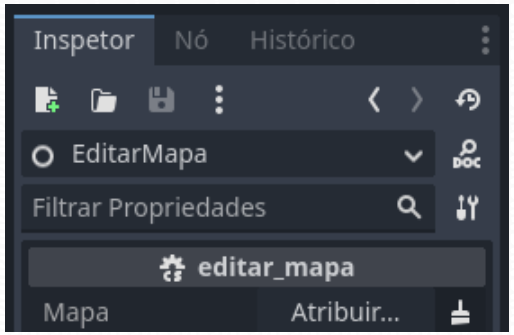
Cena: EditarMapa (Script)

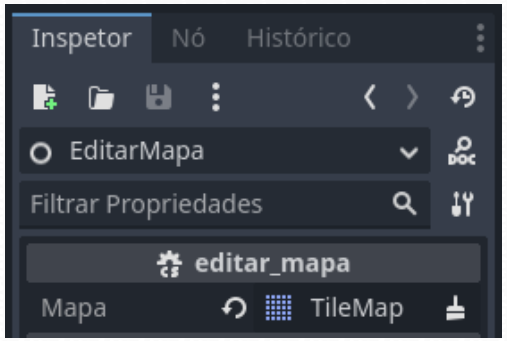
- **Primeiro passo:** para as funções do Script poderem editar o mapa é conhece-lo.

- Para conhecer o mapa vamos incluir uma diretiva `[Export]` que nos permitirá ligar, pela interface o mapa que queremos editar com script.

```
public partial class editar_mapa : Node {
    4 references
    [Export] private TileMap mapa;
}
```

- **Segundo passo**: ao incluir o script na cena e construir a solução (`build`) a variável `[Export]` vai ficar visível no Inspetor, permitindo que arrastemos o mapa alvo.





UNIVALI

Cena: EditarMapa (Script)

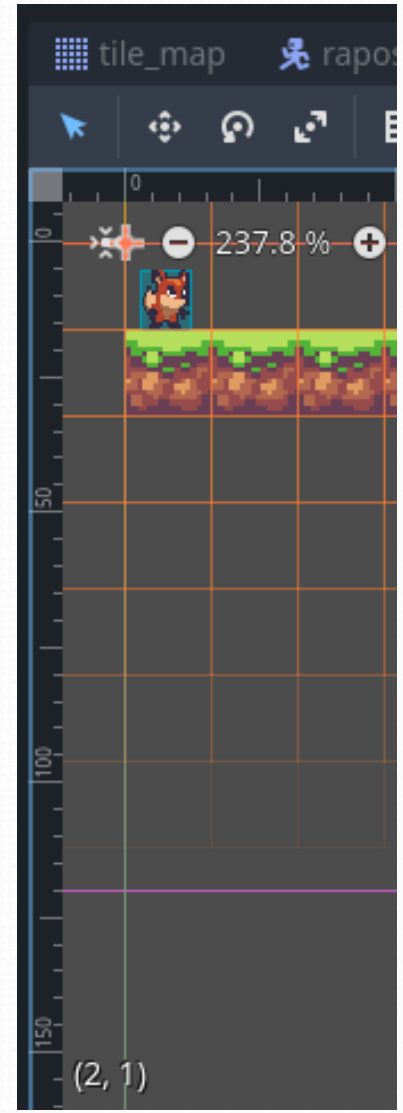
- **Terceiro passo:** Remoção de células no TileMap.

- Comando `EraseCell(tileID, tileCoord);` para remover a célula de um `TileMap` de uma coordenada específica.
  - `tileID`: ID do tile que fica visível na aba Tiles.
  - `tileCoord`: coordenada do TileMap que terá a célula removida.

```
public partial class editar_mapa : Node {
    5 references
    [Export] private TileMap mapa;
    1 reference
    public void RemoverCelulaMapa (int tileID, Vector2I tileCoord) {
        //apaga uma célula cujo ID é tileID e a coordenada é coord
        mapa.EraseCell(tileID, tileCoord);
    }
}
```

Cena: EditarMapa (Remover Celula)

- Exemplo de remoção de Células com a função
  - `EraseCell(tileID, coord)`
  - Quando o jogador clicar em um botão, a célula da imagem será removida.
  - Ao posicionar o mouse na coordenada onde se encontra a célula a ser removida, observasse que sua coordenada é (2,1).
    - O chão, 2 quadrados a frente da Raposa.
- A função que apagaria a célula ficaria com os seguintes parâmetros:
  - `EraseCell(0, new Vector2I(2,1));`
- Mas para facilitar chamar no script da Raposa usaremos a função
  - `RemoverCelulaMapa(tileID, coord);`



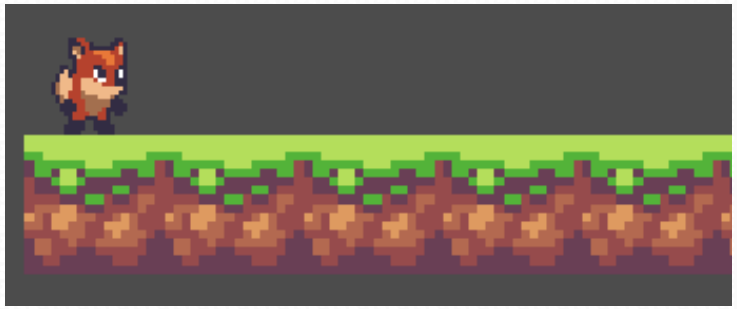

```
GetTree().CallGroup("GrupoEditarMapa", "RemoverCelulaMapa",0, new Vector2I(2,1));
```

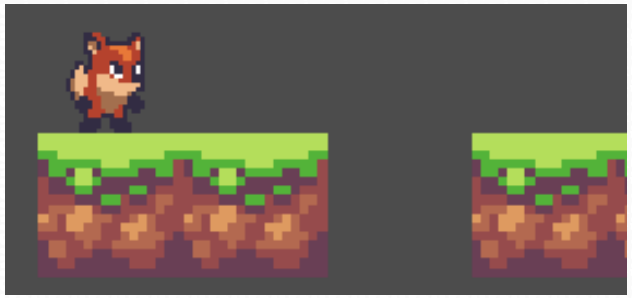

UNIVALI

Cena: EditarMapa (Remover Celulas)

- No script da Raposa foi incluído um código que ao apertar o botão ativar a célula da coordenada (2,1) será removida.

```
public override void _PhysicsProcess(double delta)
{
    if (Input.IsActionJustPressed("ativar")){
        GD.Print("Removeu célula");
        GetTree().CallGroup("GrupoEditarMapa", "RemoverCelulaMapa",0, new Vector2I(2,1));
    }
```





UNIVALI

Cena: EditarMapa (Remover Celulas)

- **Desafio aula**: Faça uma ponte (várias células) desaparecerem quando a raposa chegar em um ponto específico do mapa.
  - Dica: quando o jogador entrar em uma Area2D ele chamará uma função que irá remover os blocos de parede.

- **Desafio para casa**: Crie uma função nova no script EditarMapa que utilizará laços de repetição para remover uma quantidade de linhas, a partir de um ponto inicial, passado por parâmetro pelo usuário.
  - Dica1: `RemoverLinhaMapa(tileID, coord, tamanho);`
  - Dica2: Utilize laço de repetição nessa função para apagar uma quantidade de células definida pelo tamanho.
  - Dica3: a coordenada pode ser separada da seguinte forma:
    - `new Vector2I (coord.X, coord.Y)`

UNIVALI

Cena: EditarMapa (Script)

- **Terceiro passo:** Adição de células no TileMap.

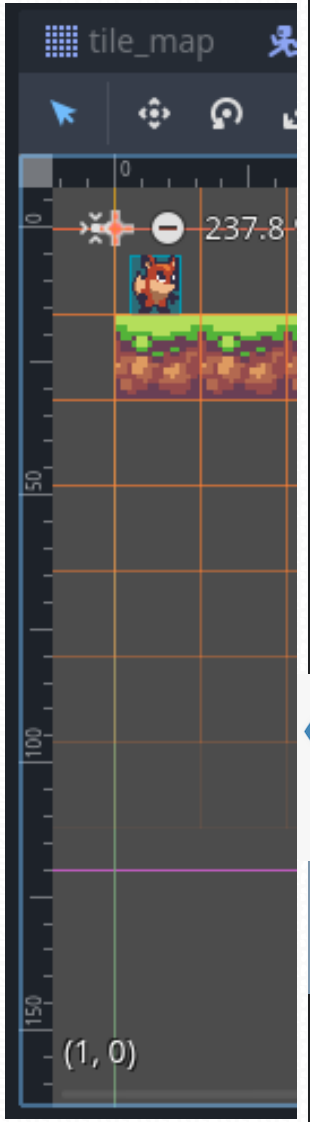
- Comando `SetCell(tileID, tileCoord, atlasID, atlasCoord);` para adicionar a célula de um `TileMap` de uma coordenada específica.
  - `tileID`: ID do tile que fica visível na aba Tiles.
  - `tileCoord`: coordenada do TileMap que terá a célula adicionado.
  - `atlasID`: ID do atlas que contém o tile.
  - `atlasCoord`: coordenada do Atlas que contém o tile a ser adicionado.

```
public partial class editar_mapa : Node {
    5 references
    [Export] private TileMap mapa;
    1 reference
    public void AdicionarCelulaMapa (int tileID,  Vector2I tileCoord, int atlasID, Vector2I atlasCoord) {
        mapa.SetCell(tileID, tileCoord, atlasID, atlasCoord);
    }
}
```

UNIVALI

Cena: EditarMapa (Adicionar Células)

- Exemplo de adição de Células com a função
  - `SetCell(tileID, tileCoord, atlasID, atlasCoord)`
  - Será utilizado `tileID` e `tileCoord` para escolher a coordenada do tile em que a célula deverá aparecer.
  - Será utilizado o `atlasID` e `atlasCoord` para escolher a célula do atlas que queremos que apareça na coordenada selecionada.
  - Ao posicionar o mouse na coordenada onde deseja-se adicionar a célula, observasse que sua coordenada inicial é (1,0).
    - O quadrado logo a frente da Raposa.
  - Além disso é necessário escolher o quadrado que irá aparecer naquela posição: (10,10).



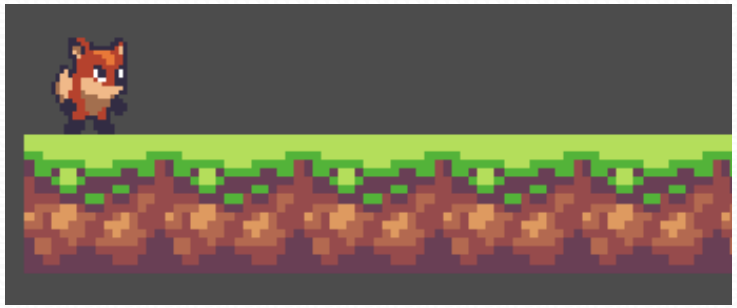
UNIVALI

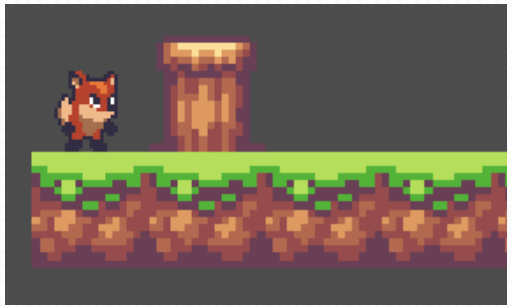
Cena: EditarMapa (Adicionar Células)

```
GetTree().CallGroup("GrupoEditarMapa", "AdicionarCelulaMapa",0, new Vector2I(1,0), new Vector2I(10,10));
```

- No script da Raposa foi incluído um código que ao apertar o botão ativar a célula (10,10), escolhida do atlasID: 0 será adicionada à coordenada (1,0) do tileID: 0.

```
public override void _PhysicsProcess(double delta)
{
    if (Input.IsActionJustPressed("ativar")){
        GD.Print("Adicionou célula");
        GetTree().CallGroup("GrupoEditarMapa", "AdicionarCelulaMapa",0, new Vector2I(1,0), new Vector2I(10,10));
    }
```





UNIVALI

Cena: EditarMapa (Adicionar Celulas)

- **Desafio aula**: Faça uma parede (várias células) aparecerem quando a raposa chegar em um ponto específico do mapa, impedindo-a de voltar o caminho.
  - Dica: quando o jogador entrar em uma Area2D ele chamará uma função que irá adicionar os blocos de parede.

- **Desafio para casa**: Crie uma função nova no script EditarMapa que utilizará laços de repetição para adicionar uma quantidade de linhas, a partir de um ponto inicial, passado por parâmetro pelo usuário.
  - Dica1: `AdicionarColunaMapa(tileID, tileCoord, AtlasID, AtlasCoord, tamanho);`
  - Dica2: Utilize laço de repetição nessa função para adicionar uma quantidade de células definida pelo tamanho.
  - Dica3: a coordenada pode ser separada da seguinte forma:
    - `new Vector2I (coord.X, coord.Y)`

UNIVALI

Tile Map (Posição do jogador no TileMap)

- Convertendo a posição do jogador em uma coordenada do TileMap.
  - Útil para permitir interações do jogador com o mapa a sua volta, por exemplo:
    - Remover uma célula a frente.
    - Adicionar um objeto onde está.
    - Alterar um objeto onde está.
    - ...

- CONTINUAR
- ENSINAR A SABER QUAL O TILE QUE ESTÁ EM ALGUM LUGAR.
  - Desafiar a cavar somente quando o tile onde está for de terra.