



# Godot: plataformaPixel: Interações

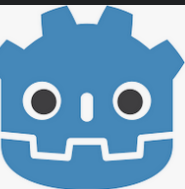
---

Prof. Thiago Felski Pereira, MSc.

# Visão Geral

---

- Nessa continuação iremos melhorar interaja deforma ativa com o mundo.
  - Coletaremos itens no mapa.
  - Ativaremos itens no mapa.
  - Teremos perigos estáticos e móveis.
- Bibliografia
  - <https://www.kenney.nl/assets/pixel-platformer>



[ 2 ]

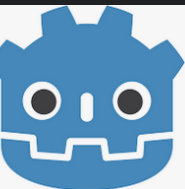
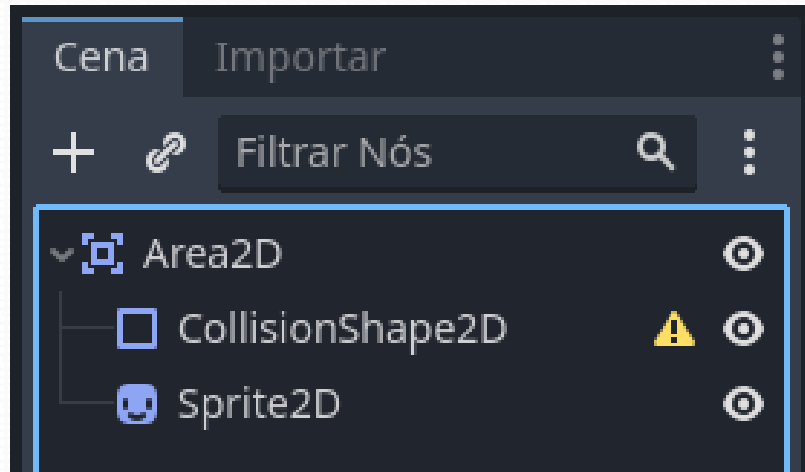


UNIVALI

# Cena: Espinho

---

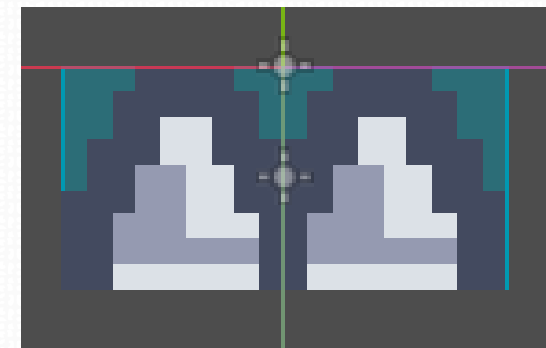
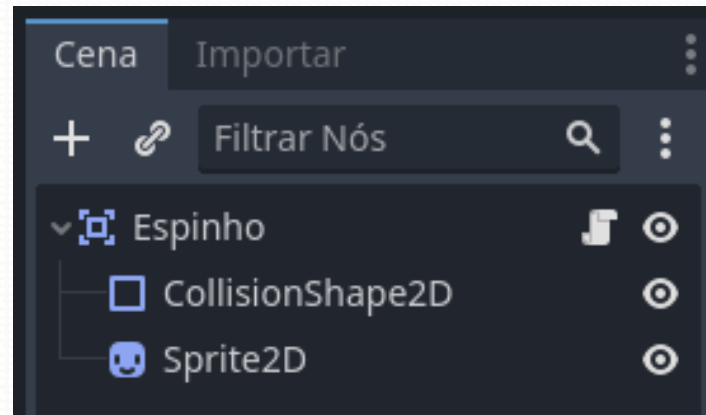
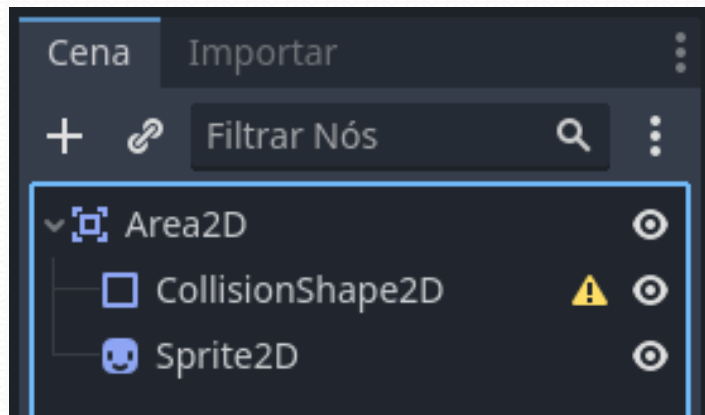
- Vamos criar alguns espinhos que poderão machucar o personagem.
  - Tornando assim nosso jogo mais desafiador.
- Primeiro crie a seguinte estrutura.
  - Area2D é um tipo de nó que permitirá detectar quando alguém colidiu com ele, iremos usar muito esse bloco.



# Cena: Espinho

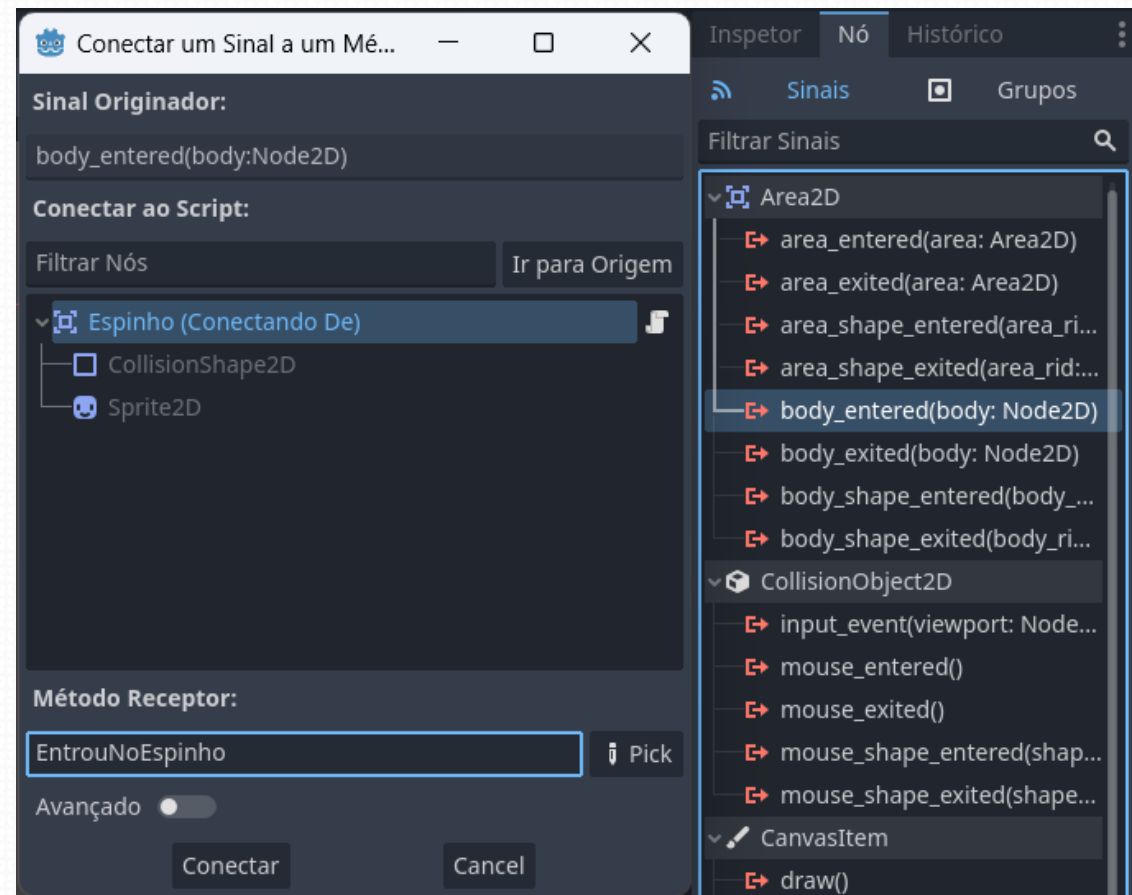
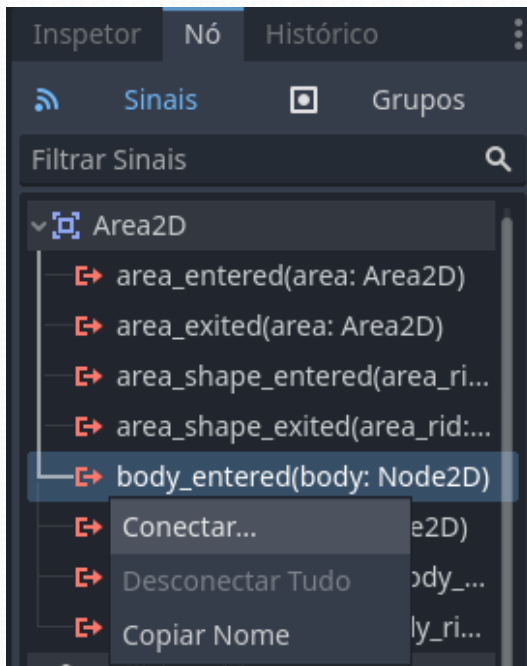
---

- Vamos criar alguns espinhos que poderão machucar o personagem.
  - Tornando assim nosso jogo mais desafiador.
- Primeiro crie a seguinte estrutura.
  - Area2D é um tipo de nó que permitirá detectar quando alguém colidiu com ele, iremos usar muito esse bloco.
  - Renomeie a Area2D para Espinho e adicione um script a ela.
  - Além disso coloque a imagem de um espinho e defina sua área de colisão.



# Cena: Espinho

- Vamos criar alguns espinhos que poderão machucar o personagem.
  - Tornando assim nosso jogo mais desafiador.
- Criando um **senal** de alerta para colisão.
  - Com a cena principal selecionada ...



# Cena: Espinho

---

- Vamos criar alguns espinhos que poderão machucar o personagem.
  - Tornando assim nosso jogo mais desafiador.
- Criando um script no Espinho para tratar a colisão.
  - O script irá detectar se quem colidiu com o Espinho foi o jogador, nesse caso iremos chamar a função espetado no próprio script do jogador

```
public void EntrouNoEspinho(Node2D body) {  
    if (body is Jogador) {  
        ((Jogador)body).Espetado();  
    }  
}
```



# Cena: Jogador

---

- Vamos criar alguns espinhos que poderão machucar o personagem.
  - Tornando assim nosso jogo mais desafiador.
- No script do Jogador iremos criar a função Espetado
  - Essa função irá retornar o jogador ao ponto de salvamento e retirar uma vida do jogador.
  - Precisaremos descobrir como criar esses pontos de salvamento e como definir e trabalhar com a vida.

```
6 references
private int vida;
8 references
private Vector2 pontoDeSalvamento;
```

```
public void Espetado() {
    GlobalPosition = pontoDeSalvamento;
    vida = vida - 1;
    GD.Print("Retornou ao ponto salvo");
}
```

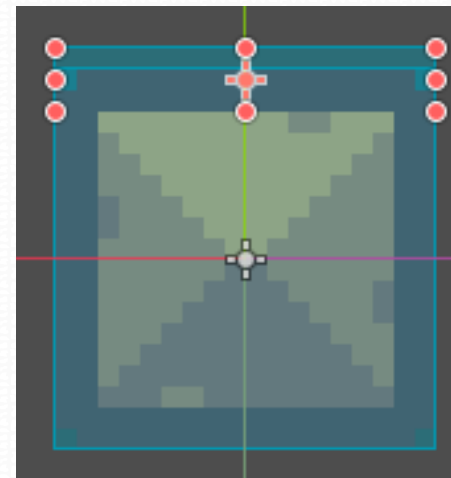
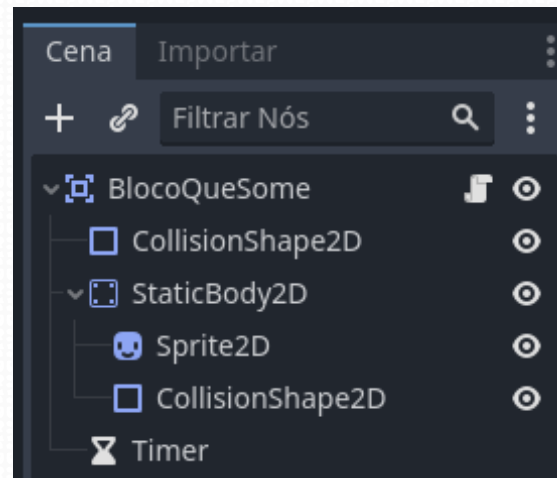
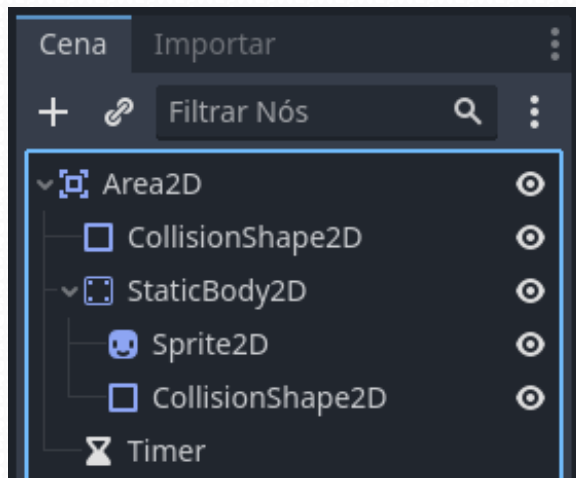


[ 7 ]

# Cena: BlocoQueSome

---

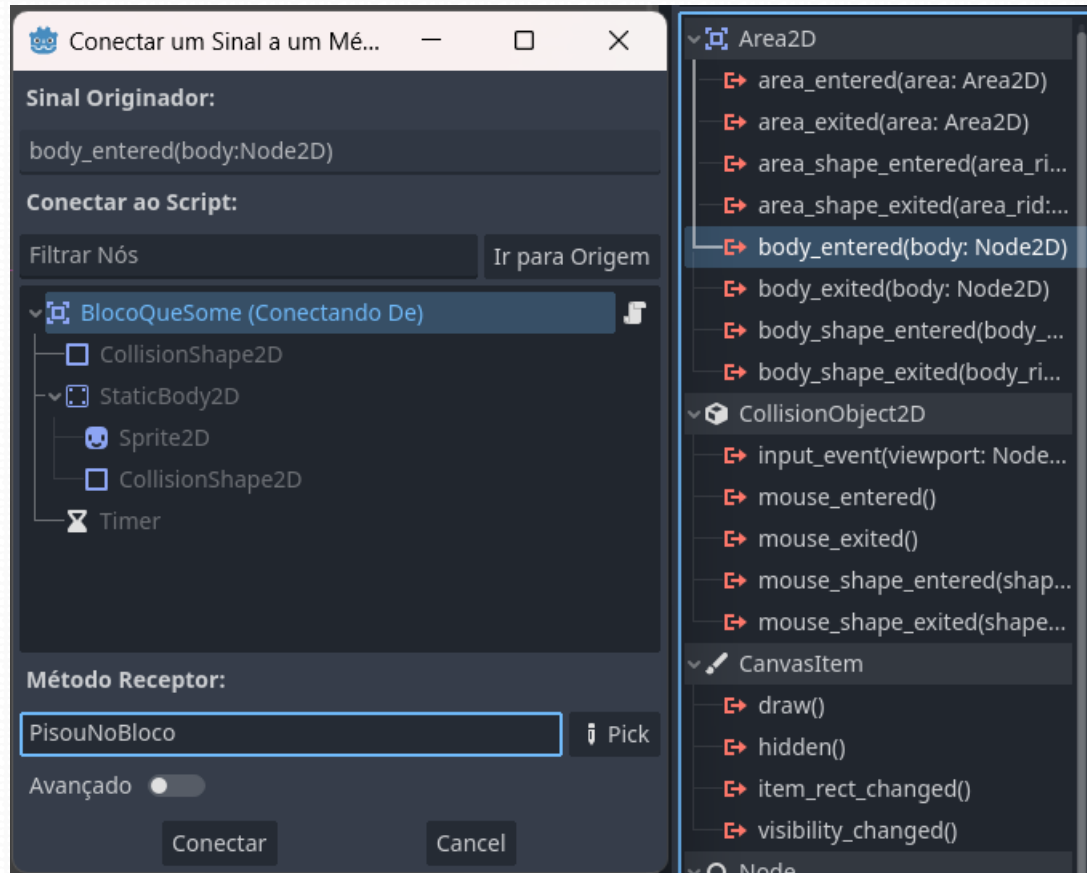
- Vamos criar alguns bloco que somem com o tempo.
- Primeiro crie a seguinte estrutura.
  - Renomeie a Area2D para BlocoQueSome e adicione um script a ela.
  - Note que a foi colocada uma área de colisão para detectarmos que o jogador entrou nela.
  - Também foi incluído um timer para definir o tempo que o bloco existirá depois de pisado.





# Cena: BlocoQueSome

- Vamos preparar para detectar a colisão desse bloco com o Jogador.



# Cena: BlocoQueSome

---

- Vamos programar o script do BlocoQueSome

```
public partial class BlocoQueSome : Area2D {  
    7 references  
    private Timer tempo;  
    6 references  
    private bool pisou=false;  
    4 references  
    public override void _Ready() {  
        //tempo recebe o nó Timer  
        tempo = GetNode<Timer>("Timer");  
    }  
    3 references  
    public override void _Process(double delta) {  
        if (tempo.TimeLeft < 0.1f && pisou) {  
            //Remove o bloco do jogo  
            QueueFree();  
        }  
    }  
}
```

```
    public void PisouNoBloco(Node2D body) {  
        if (body is Jogador) {  
            //Inicia uma Contagem Regressiva de 5 segundos  
            //E indica que já pisou  
            tempo.Start(5);  
            pisou=true;  
        }  
    }  
} //Fim da classe BlocoQueSome
```

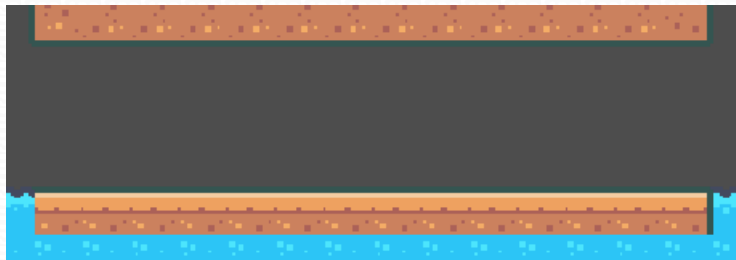


[ 10 ]

# Enunciado Prática Final

---

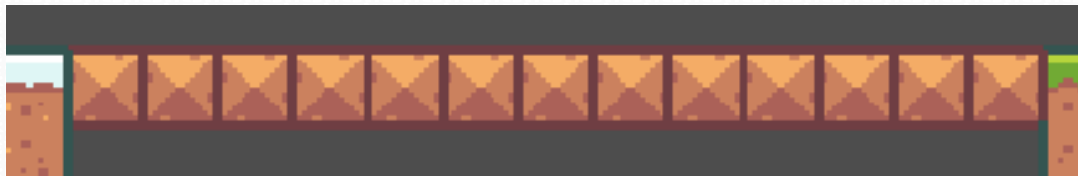
- Você está recebendo um código com algumas coisas prontas e deve fazer modificações nele para atender os requisitos do projeto. Temos um total de X questões.
- Questão 1: Espinhos
  - Adicione espinhos no chão e no teto da área indicada na figura.
  - Lembre-se: Não deverá ser impossível passar por essa área.



# Enunciado Prática Final

---

- Você está recebendo um código com algumas coisas prontas e deve fazer modificações nele para atender os requisitos do projeto. Temos um total de 4 questões.
- Questão 2: Blocos que somem
  - Substitua os blocos do tilemap por blocos que somem.
  - Tente ajustar o tempo de existência do bloco para 2 segundos.



12

# Enunciado Prática Final

---

- Você está recebendo um código com algumas coisas prontas e deve fazer modificações nele para atender os requisitos do projeto. Temos um total de 4 questões.
- Questão 3: Diamantes
  - Insira 3 diamantes coletáveis espalhados no mapa.
  - Esses diamantes deverão sumir quando o jogador colidir com eles.

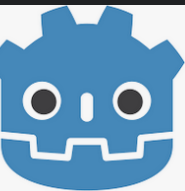


13

# Enunciado Prática Final

---

- Você está recebendo um código com algumas coisas prontas e deve fazer modificações nele para atender os requisitos do projeto. Temos um total de 4 questões.
- Questão 4: Crie um ou mais pontos de salvamento pelo mapa.
  - Quando o jogador colidir com o ponto de salvamento ele deverá registrar sua posição atual.
  - Para isso chame uma função no jogador que define o ponto de salvamento como a posição atual do jogador (`GlobalPosition`)



# Enunciado Prática Final

---

- Você está recebendo um código com algumas coisas prontas e deve fazer modificações nele para atender os requisitos do projeto. Temos um total de X questões.
- Questão 5: Desafio Livre
  - Implemente uma funcionalidade extra a sua escolha no seu jogo.
  - Sugestões de funcionalidade:
    - Inimigos ou espinhos móveis;
    - finalizar o jogo ou trocar de mapa quando pegar todos os diamantes;
    - finalizar o jogo quando acabarem as vidas; e
    - passagens secretas.



15



( 16 )



UNIVALI