

Gerenciando Cenas

Prof. Thiago Felski Pereira, MSc.

Visão Geral

- O jogo está começando a tomar forma.
- Temos:
 - Um objeto **Jogador** que anda em todas as direções em grade.
 - Como o movimento de um personagem em um tabuleiro.
 - Objetos **Parede** usaremos para desenhar nosso cenário de fato.
 - O movimento do Jogador é bloqueado por elas.
 - Objetos **Caixa** podem ser empurrados pelo Jogador.
 - Na aula passada já estudamos todas as interações desses objetos.
 - As caixas podem ser empurradas em todas as direções, mas não para dentro de paredes ou outras caixas.



Visão Geral

- Precisamos agora de um cenário real e um objetivo para nosso jogo tomar forma.
 - Iremos desenhar nosso mapa completo.
 - Iremos inserir um objeto objetivo e fazer o jogador vencer se conseguir levar a caixa naquele lugar.

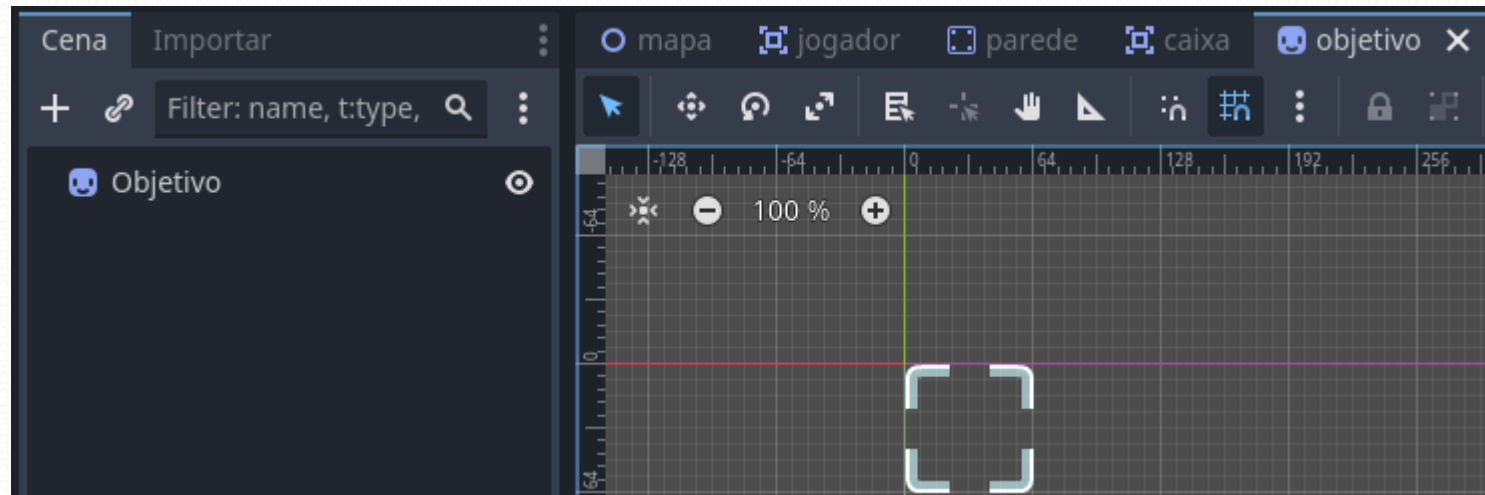


[3]



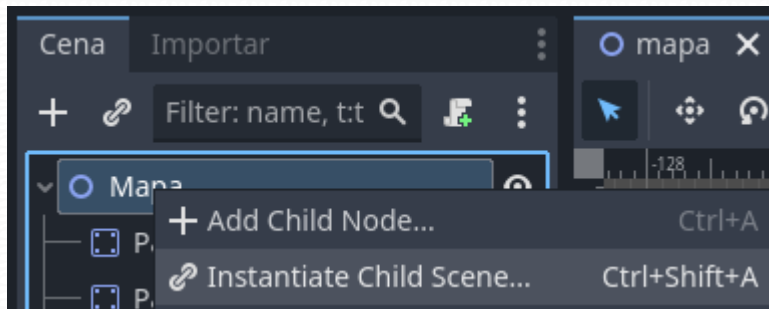
Cena Objetivo (detectar colisão)

- Vamos começar criando a cena **Objetivo** do nosso Jogo.
 - Essa é a cena mais simples que já criamos, pois é apenas um Sprite2D:
 - Crie uma nova cena.
 - Menu -> Cena -> Nova Cena
 - Adicione um `Sprite2D` e renomeie para **Objetivo**.
 - Insira a imagem que preparamos para nosso objetivo e redimensione-a.
 - Resultado esperado:



Cena Mapa

- Insira o **Objetivo** no seu **Mapa**.
 - Relembrando: passos para inserir uma cena dentro de outra cena:
 - Vá para cena que você deseja colocar a outra cena.
 - Clique no nó principal com o botão direito.
 - Selecione a opção `Instantiate Child Scene...`

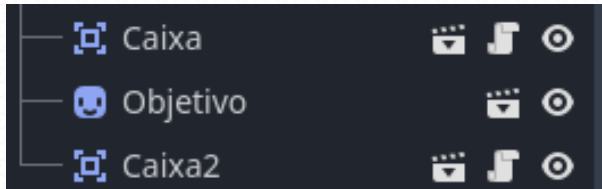


- Se tudo deu certo temos um objetivo no nosso mapa, só precisaremos configurá-lo.

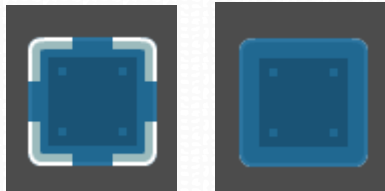


Cena Mapa (teste)

- Relembrando a ordem das cenas.
 - **Faça o seguinte experimento:** coloque a cena Objetivo entre as cenas de duas Caixas e teste.



- Note que quando movimentar as Caixas para o Objetivo uma vai ficar à frente e a outra a atrás.

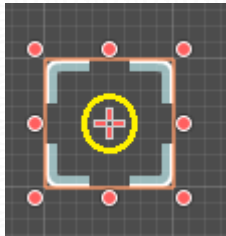


- Isso acontece porque as cenas são desenhadas na ordem que aparecem na hierarquia, assim:
 - No primeiro caso a caixa é desenhada e depois o objetivo é desenhado por cima e no segundo caso o objetivo é desenhado e a caixa é desenhada por cima.

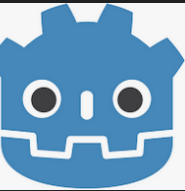
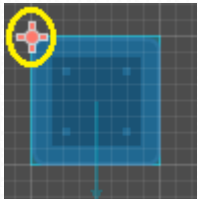


Cena Mapa (teste2)

- Descobrindo a coordenada que a **Caixa** deve ficar para vencermos o jogo.
 - Quando selecionamos o Objetivo na cena do Mapa podemos verificar no Inspetor a coordenada que ele se encontra.
 - Inspetor -> Node2D -> Transform -> Position
 - Mas note que o ponto indicado para coordenada é dado pela mira que circulei em amarelo para ajudar a identificar.



- Note que na **Caixa** nosso ponto não está centralizado como no **Objetivo**.



[7]

Cena Mapa (teste2)

- Por isso a maneira mais eficiente, nesse momento, de testar a posição que a **Caixa** deve ficar para atingir o **Objetivo** é movimentar manualmente a caixa para a posição desejada.



- A coordenada da Caixa nesse momento é a posição real que a caixa deverá ficar para vencermos o jogo.



Cena Jogador

- Vamos editar o Script do **Jogador** para escrever uma mensagem de vitória se o Jogador empurrar a caixa para o Objetivo.
- Já temos no Script um trecho de código que verifica se o jogador conseguiu empurrar a caixa.

```
Area2D colisor = ray.GetCollider() as Area2D;  
if ((bool)colisor.Call("Empurrar", direcao, area_quadrado)) {  
    Position += direcao * area_quadrado;  
    //Testa se empurrou a caixa para um objetivo  
}
```

- A Caixa empurrada ficará uma posição à frente de onde o jogador estará após se movimentar.

```
Vector2 destino_caixa = Position + direcao * area_quadrado;
```

- Usaremos essa variável que acabamos de criar para testar se a Caixa chegou no Objetivo.



Cena Jogador

- Vamos editar o `Script` do **Jogador** para escrever uma mensagem de vitória se o Jogador empurrar a caixa para o Objetivo.
 - Fazendo o teste de a caixa chegou na coordenada.
 - Substitua as ?? pelas coordenadas x e y que você identificou como destino da Caixa.

```
if(destino_caixa == new Vector2(??,??)){  
|   GD.Print("Vitória");  
}
```

- Teste seu código.



(10)



UNIVALI

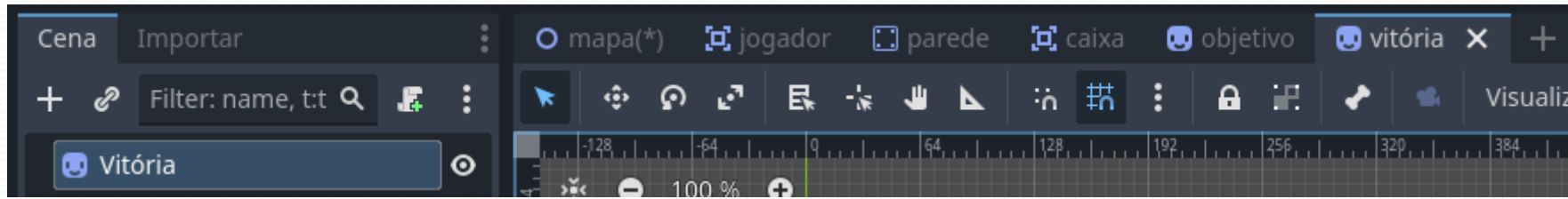
Gerenciando Cenas

- Como finalizar seu jogo.
 - No Godot, temos o comando `GetTree().Quit();` que quando executado irá finalizar o seu jogo.
- Já o comando `GetTree().ChangeSceneToFile("caminho do arquivo da cena");`
 - Irá mudar para a cena indicada pelo caminho escolhido entre aspas.
 - **Mas como encontrar esse caminho?**

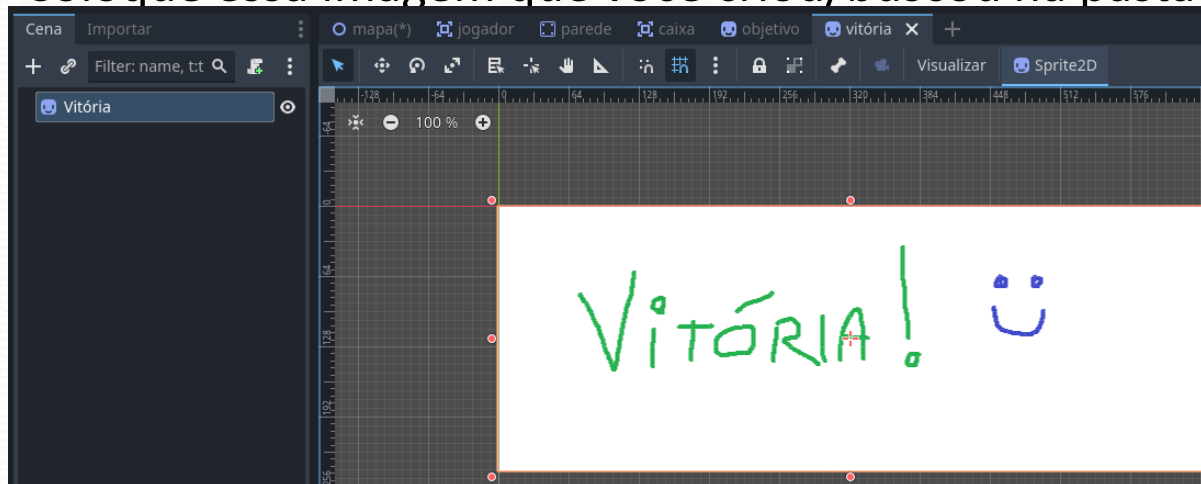


Gerenciando Cenas

- Trocando de cena.
 - Crie uma cena chamada Vitória para abrir no momento que o jogador vencer o jogo.

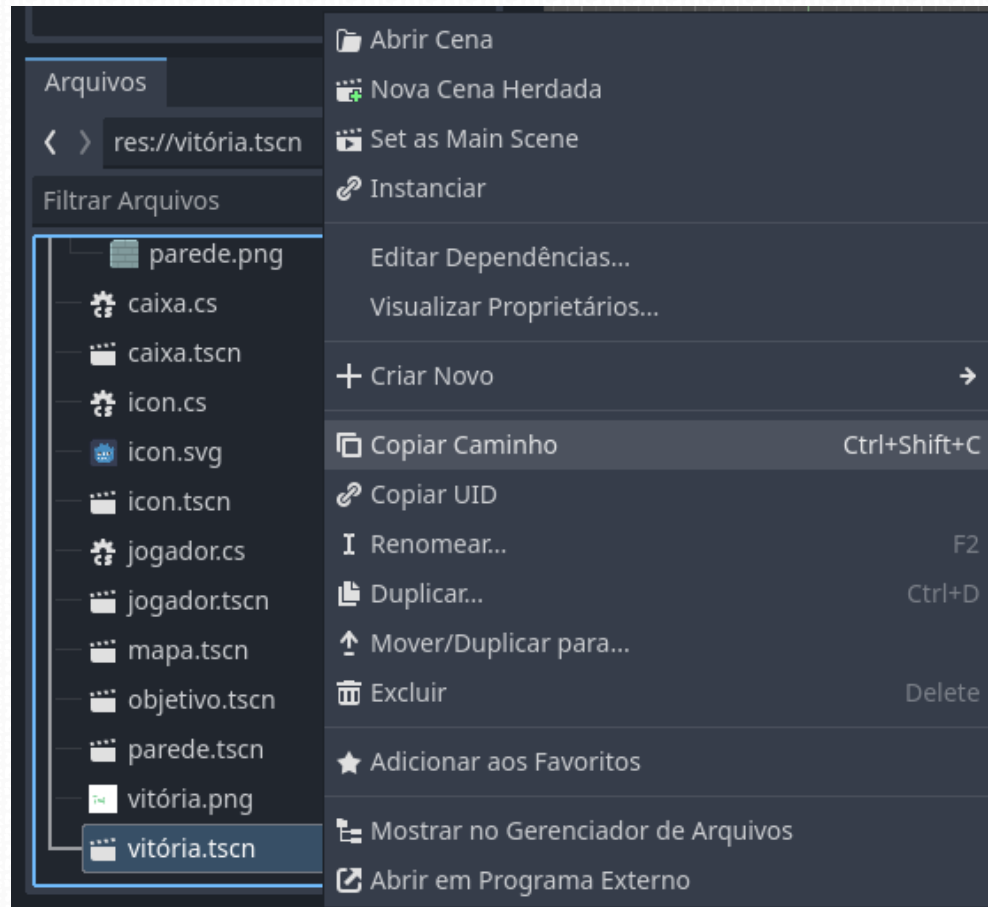


- Crie uma imagem que represente vitória no paint ou outra ferramenta de desenho, alternativamente, pode-se pegar uma imagem pronta na internet.
- Coloque essa imagem que você criou/buscou na pasta do projeto e arraste para o Vitória.



Gerenciando Cenas

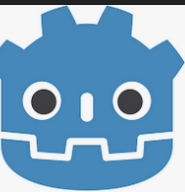
- Trocando de cena.
 - Encontre a cena **vitória.tscn** e copie o caminho.



Gerenciando Cenas

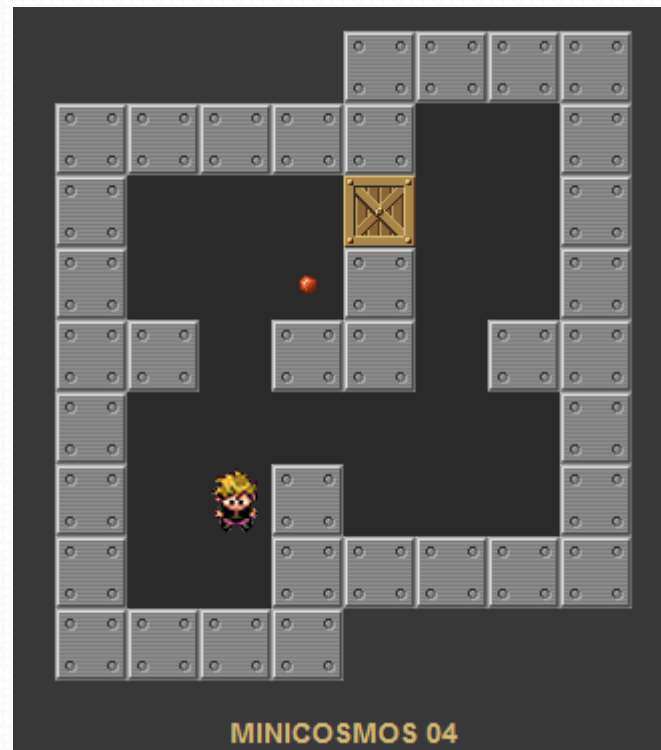
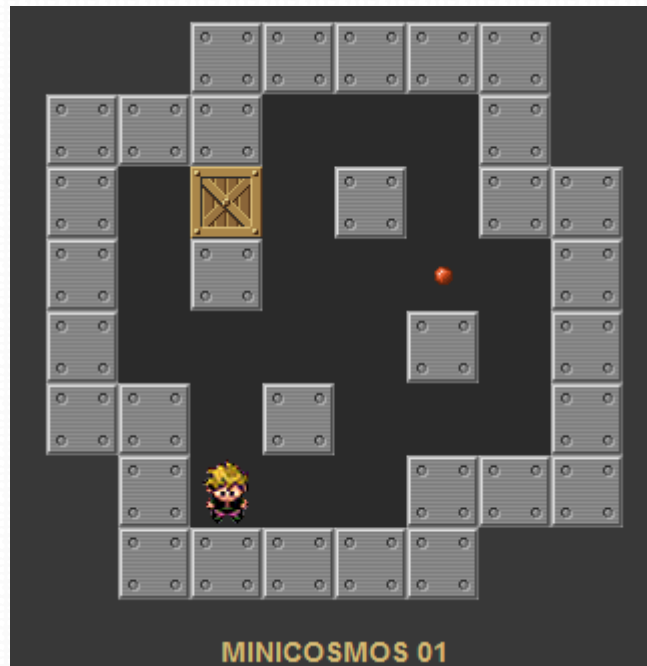
- Trocando de cena.
 - Coloque o caminho na função de troca de cena.
- Use isso no lugar de apenas imprimir vitória no console.
- **Teste seu programa.**

```
GetTree().ChangeSceneToFile("res://vitória.tscn");
```



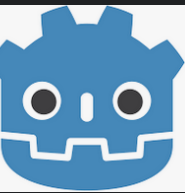
Atividade

- Crie 2 mapas:
 - Comece no mapa MINICOSMOS01 e mude para o mapa MINICOSMOS04 quando vencer.



Atividade: Desafio

- Crie 2 mapas:
 - Após o término do mapa MINICOSMOS04 mostrar a cena de vitória.



16



UNIVALI

Obrigado pela atenção!
