Curso Técnico

de Programação de Jogos Digitais

Aula 34

índice

[1. Movimento Flip 3](#_Toc170112854)

[2. Salto 4](#_Toc170112855)

[3. Referencias 7](#_Toc170112856)

# Movimento Flip

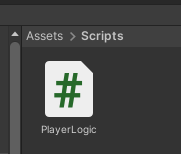
No contexto de um jogo, o movimento flip geralmente se refere a uma ação em que um personagem ou objeto vira ou gira rapidamente, geralmente em torno de um eixo, alterando sua orientação. Existem diferentes tipos de "flip" que podem ser implementados em jogos:

Flip Horizontal (Eixo Y): O personagem ou objeto é invertido horizontalmente, como um espelhamento ao longo do eixo vertical. Isso é comumente usado em jogos 2D de plataforma, onde um personagem vira de frente para o lado oposto quando muda de direção.

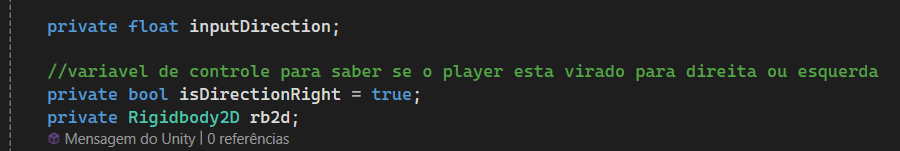
Flip Vertical (Eixo X): O personagem ou objeto é invertido verticalmente, como um espelhamento ao longo do eixo horizontal. Esse tipo de flip é menos comum, mas pode ser usado para criar efeitos específicos, como uma animação de salto mortal.

Flip 3D (Eixo Z ou outros): Em jogos 3D, o flip pode envolver uma rotação completa ao redor de qualquer eixo (X, Y, ou Z). Isso pode ser usado para acrobacias, piruetas ou mudanças de direção.

Para adicionar ao personagem o flip, a ação de estar sempre de frente para onde se anda vamos ao script criado devemos modificar o script.

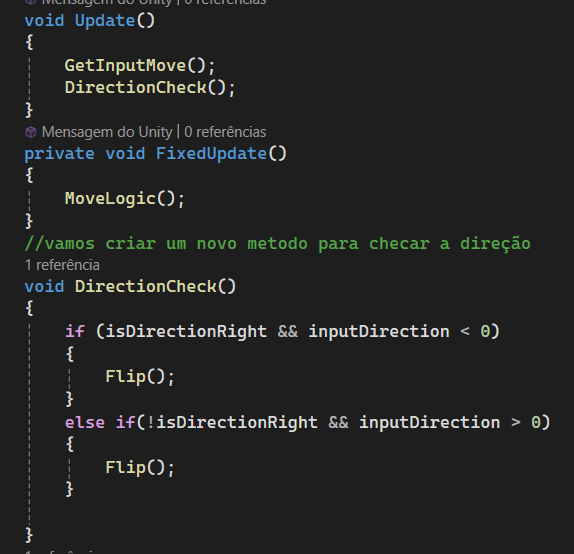


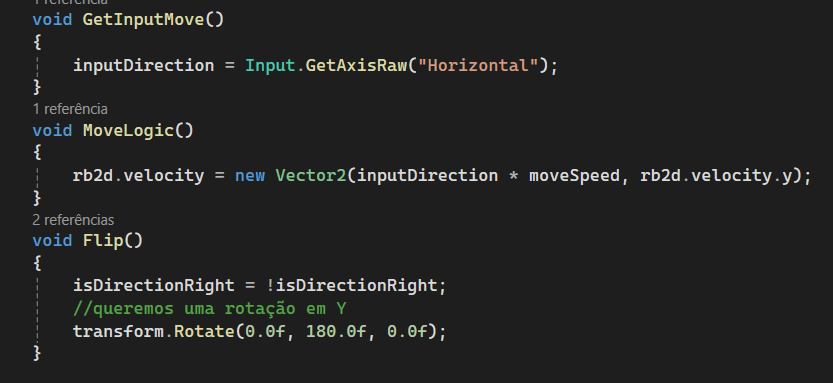
Vamos gerar algumas modificações, adicionando uma variável que ira comparar para que lado o botão direcional esta sendo pressionado.



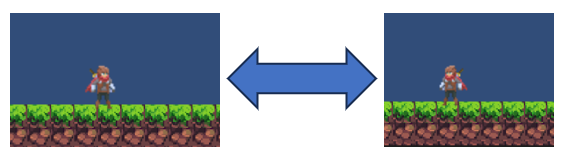
Então dentro do FixedUpdate gerar uma função que fará uso desta variável e irá fazer a comparação sobre o movimento do personagem.

Então dentro do Update devemos chamar esta função criada.



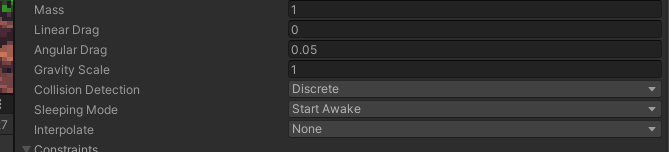


Após as modificações podemos voltar para Unity e ver como o personagem já faz o flip

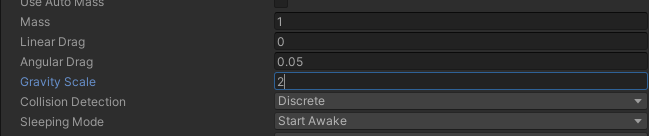


# Salto

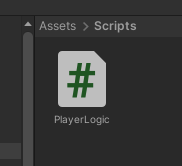
Agora vamos gerar o movimento de pulo para o personagem, então devemos trabalhar com a gravidade. Para criar a mecânica de pulo selecione o objeto player/rigidbody2d e então gravity Scale



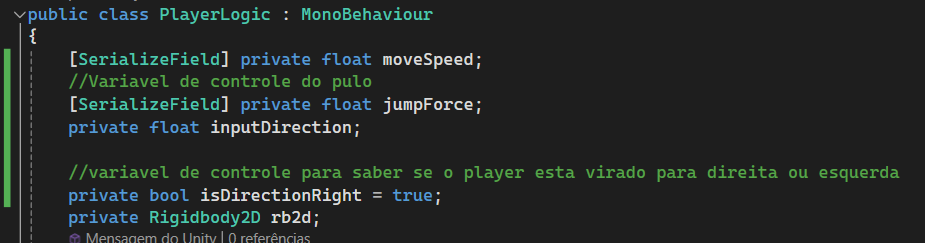
Modifique o valor para 2



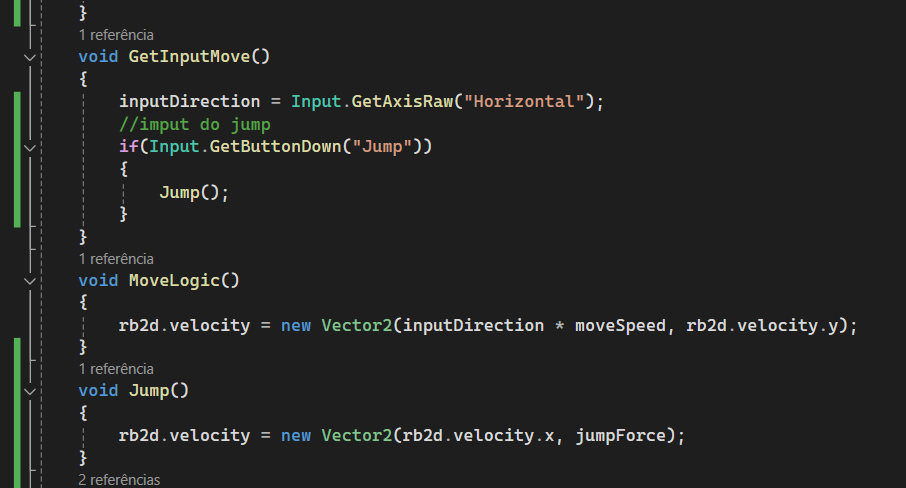
Agora retorne para o script



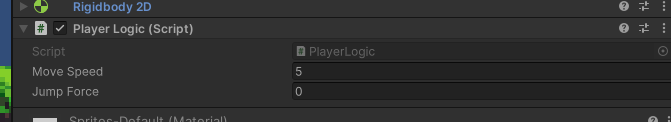
Adicione a variável jumpForce que irá controlar a altura do pulo do personagem. Poderíamos criar esse valor fixo mas deixando ele através de uma variável podemos usar isso futuramente como uma habilidade de nosso personagem.



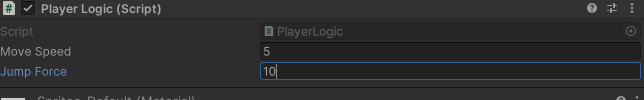
Adicione então uma função GetInputMove que ligara o salto ao toque no botão do teclado.



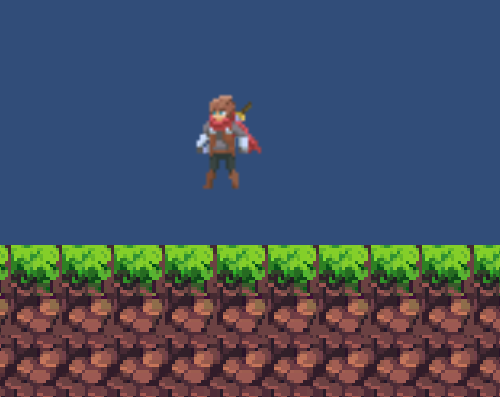
Volte para Unity e veja a nova variável criada



Apontamos um novo valor para a força do pulo.



E então testamos



Os códigos usados em nossas aulas estão disponíveis em nosso repositório no GitHub.



https://github.com/rildexter/pjd/tree/main/djm

# Referencias

1. Unity Technologies. (2020). **Unity User Manual.** Unity Technologies.
2. [C#, 2020] **Visual C# Developer Center**, Microsoft Docs.

Microsoft Corporation. (2020).

1. Geig, M. (2018). **Unity 2018 Game Development in 24 Hours**, Sams Teach Yourself. Sams Publishing.
2. Hocking, J. (2015). **Unity in Action: Multiplatform Game Development in C#.** Manning Publications.