Curso Técnico

de Programação de Jogos Digitais

Aula 43

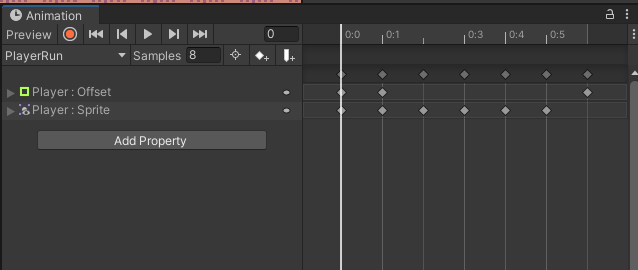
índice

[1. Animação de dano 3](#_Toc170133911)

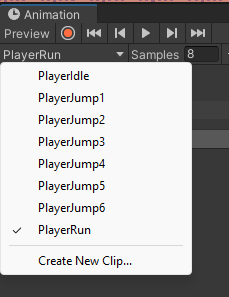
[2. Referencias 8](#_Toc170133912)

# Animação de dano

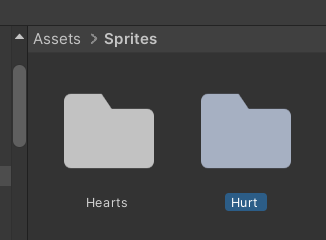
Para gerar a animação que deverá ocorrer quando o personagem tocar o inimigo, devemos inicialmente clicar no objeto player e então ir até a janela de Animation.



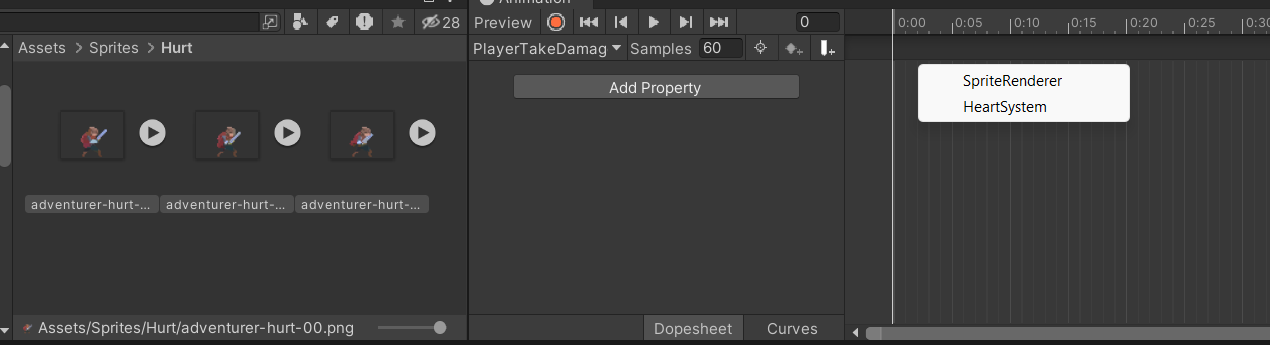
Crie então um novo clipe chamado PlayerTakeDamageAnim

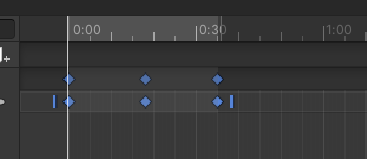


Salve na pasta animator. Procure agora a pasta hurt dentro de sprites, nela teremos as animações do player sendo ferido.



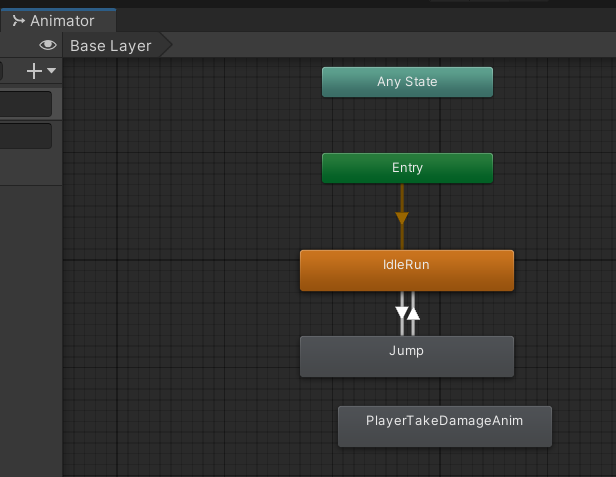
Arraste as imagens para linha de tempo e escolha Sprite Renderer

  
Arraste a linha para a animação ficar com 0.3 segundo

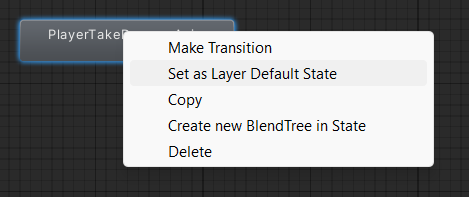
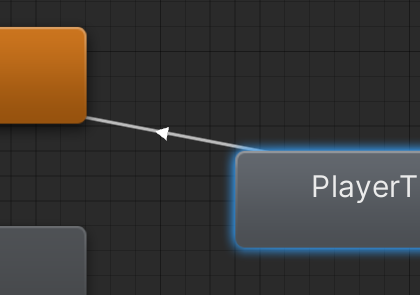


Faça o teste e vamos criar a lógica para a animação ser acionada

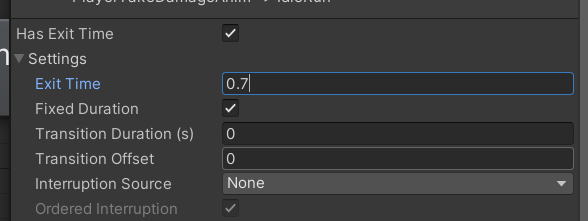
Siga para a aba animator em Windows/Animation/animator



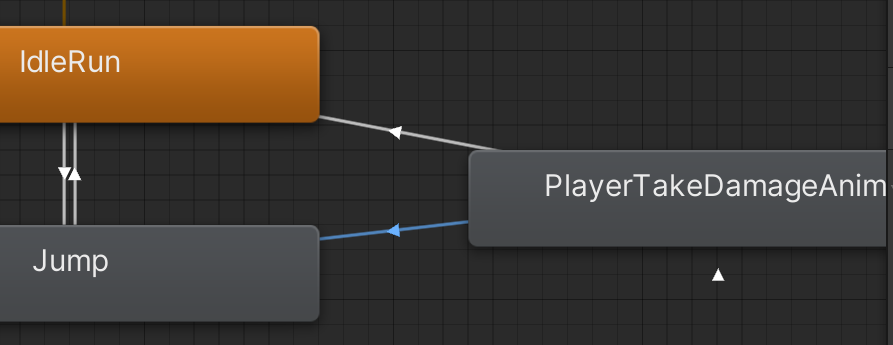
Clique com botão direito do mouse e escolha make transition

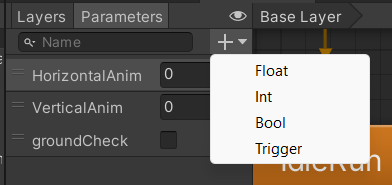
Ele não precisará de condicionais mas somente em settings mude o transition duration para zero e exit time para 0.7

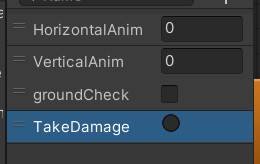


Faça o mesmo ligando uma transição ao jump

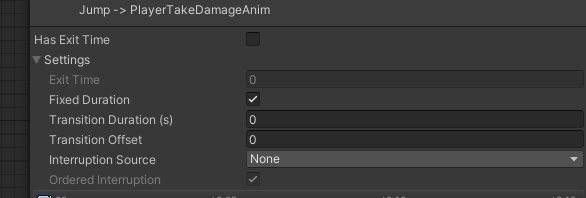


Para gerar o retorno as estas ações devemos criar um novo parâmetro do tipo trigger

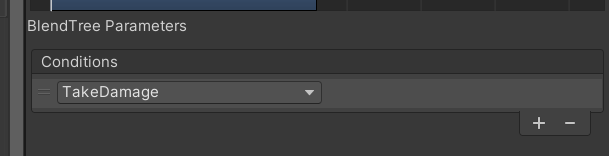




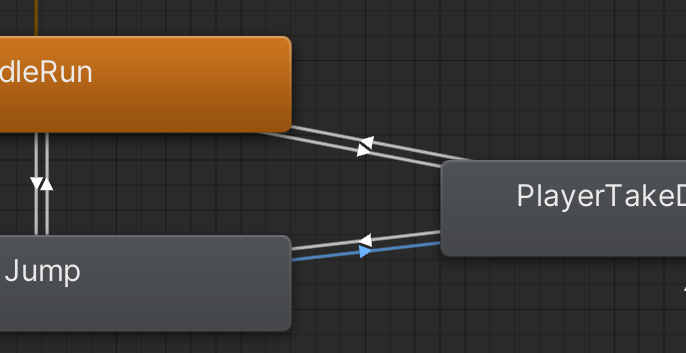
Crie a nova transição e zere os valores para exit time, transition duration desmarque has exit time.



Gere uma condição chamando o takedamage.



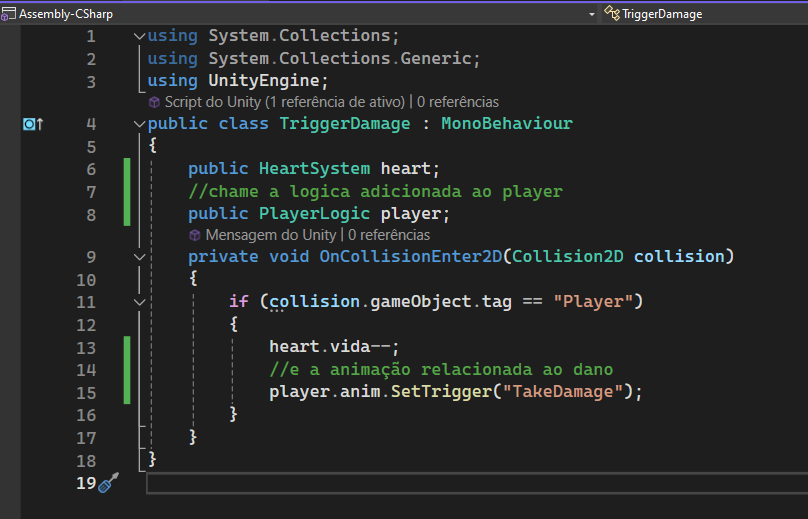
Faça o mesmo para o jump e salve.



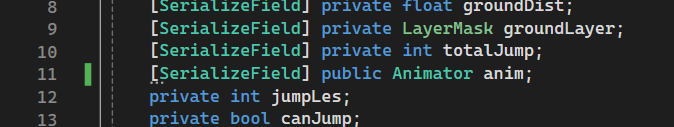


Cada transição criada possui uma seta mostrando como os efeito ali gerados reagem. Podemos criar quantas ligações sejam necessárias mas sempre se mantenha atento a lógica de comunicação entre elas.

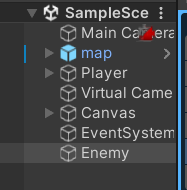
Abra o script TriggerDamage e faça as modificações a seguir.



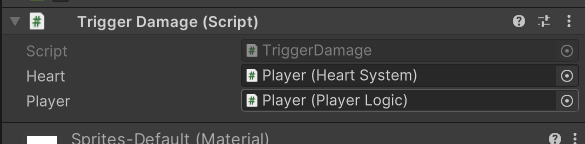
Teremos de deixar a anim publica no script PlayerLogic



Retorne para Unity e selecione o Enemy.



Arraste o objeto player para player no script TriggerDamage no Enemy



Execute o projeto e faça o teste da nova animação.

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.