Curso Técnico

de Programação de Jogos Digitais

Aula 44

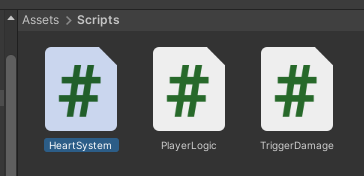
índice

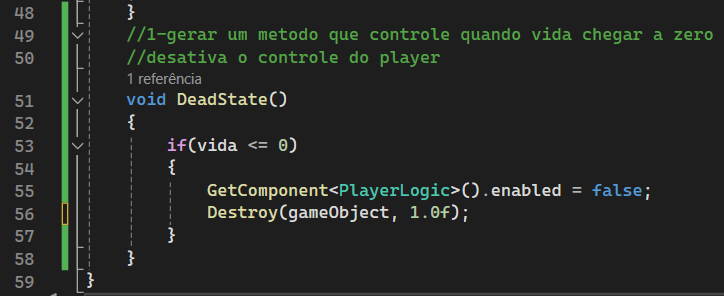
[1. Dano mortal 3](#_Toc170134851)

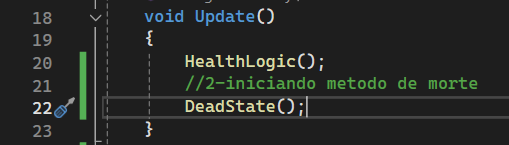
[2. Referencias 8](#_Toc170134852)

# Dano mortal

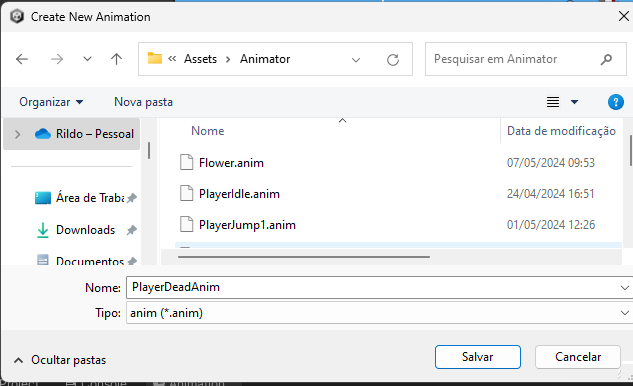
Quando o personagem tem sua energia finalizada, ele deverá ter uma animação mostrando que ele perdeu o jogo. Para gerar uma resposta a dano fatal devemos ir até ao script HeartSystem



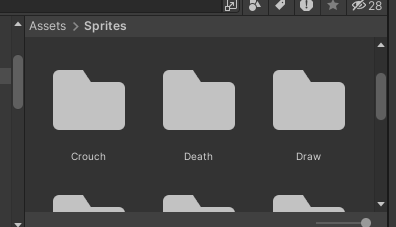




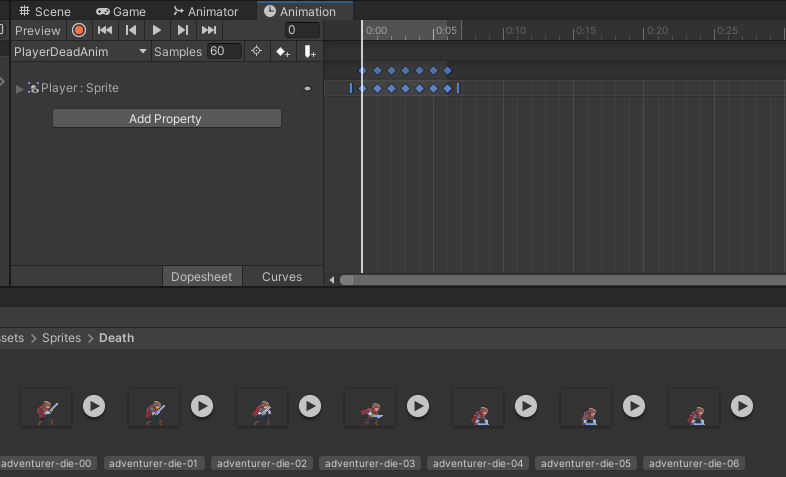
Salve e retorne para Unity e faça o teste. Vamos então criar a animação de morte. Selecione o player e vá na aba Animation. Crie um novo clip chamado PlayerDeadAnim



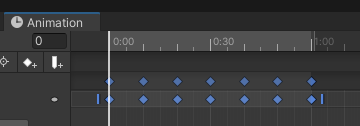
Devemos abrir a pasta de sprites e busque pelas imagens de morte



Arraste todas as imagens para linha de samples do animation



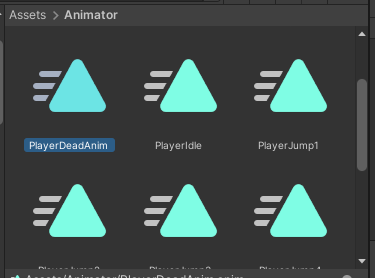
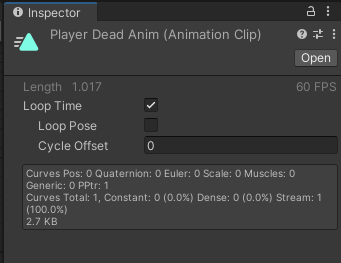
Escolha sprite renderer e então deixe toda animação com 1 segundo.



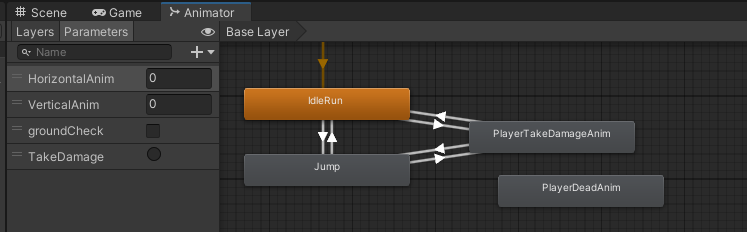
Faça um teste e veja se animação de morte funciona para o Animation.



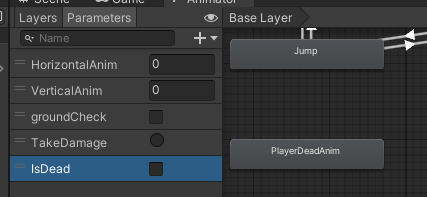
Abra a pasta assets/Animation. Clique na animação de PlayerDeadAnim e no inspector retire o loop.

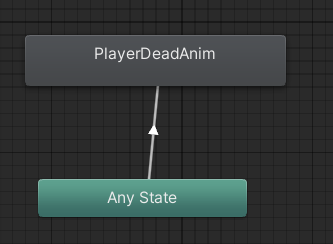
Para gerar a lógica de chamar a animação vá em animator.



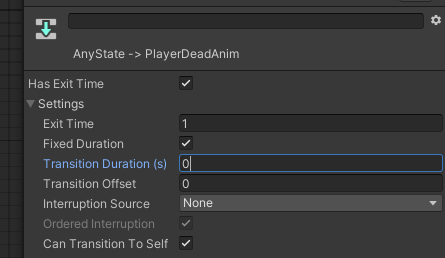
Crie um novo parâmetro tipo trigger chamado IsDead



Crie uma transição de Any State para o PlayerDeadAnim



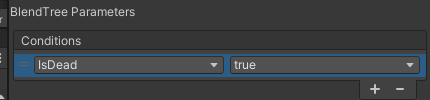
Faça as seguintes configurações para essa transição



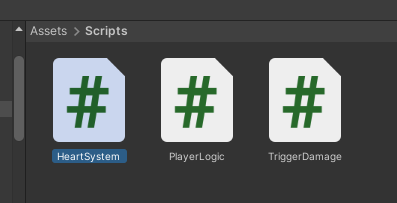
Ative o has exit Time;

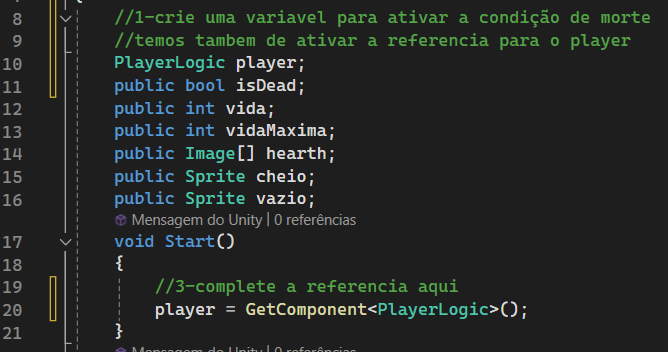
Exit Time coloque o mesmo valor de tempo da animação (Como escolhemos 1 no tempo da animação aqui também ficara como 1 mas caso tenha de dar o desconto pela animação isso poderá chegar até 1.5)

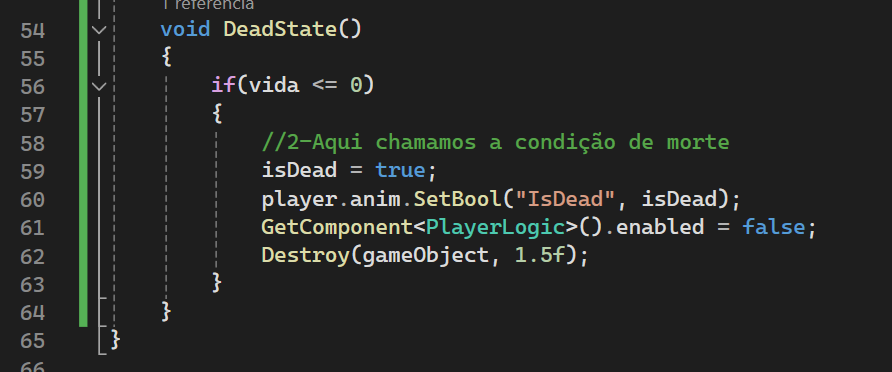
Para o Transition duration o valor deverá ser zero. Adicione uma condição IsDead True



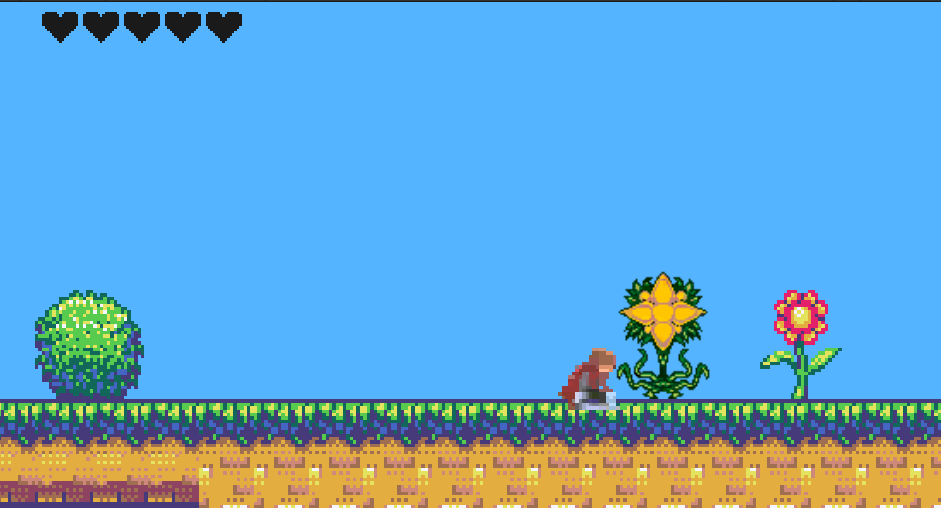
Então devemos ir ao script HeartSystem e ativar essa condição.







Salve e retorne para Unity.





São os detalhes que fazem seu jogo ser querido por seu publico, por isso até mesmo algo simples como uma animação de dano deve ser bem feita.

Por falar em pequenos detalhes ainda podemos adicionar um efeito de empurrar nosso herói ao perigo de dano, então faremos isso em nosso próximo encontro.

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.