Curso Técnico

de Programação de Jogos Digitais

Aula 41

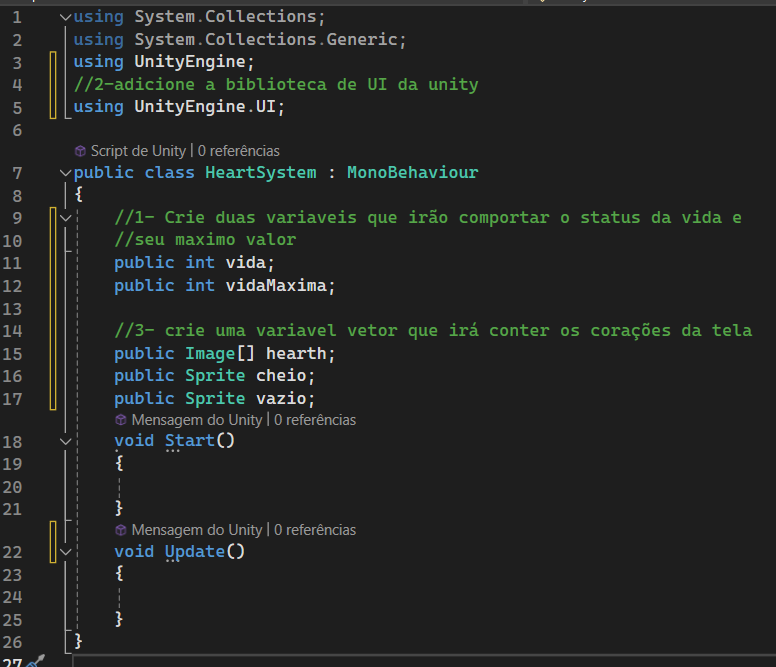
índice

[1. Logica para o sistema de vida 3](#_Toc170128023)

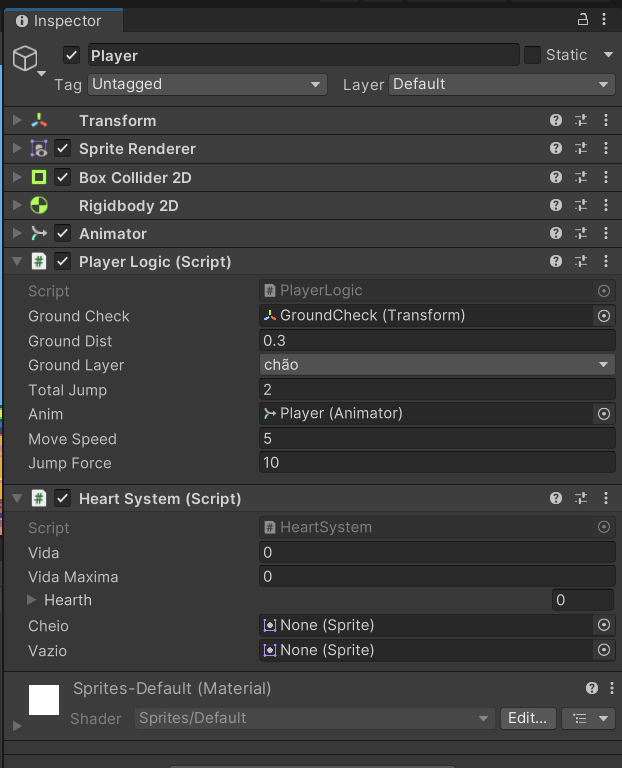
[2. Referencias 8](#_Toc170128024)

# Logica para o sistema de vida

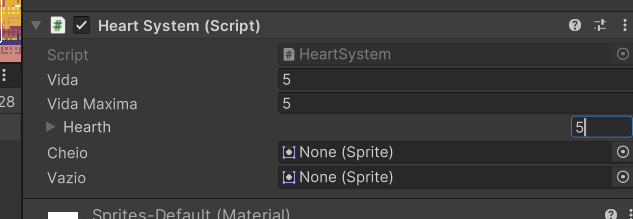
Para a lógica dos corações crie um novo script chamado HeartSystem na pasta de Scripts. Adicione o código a seguir.



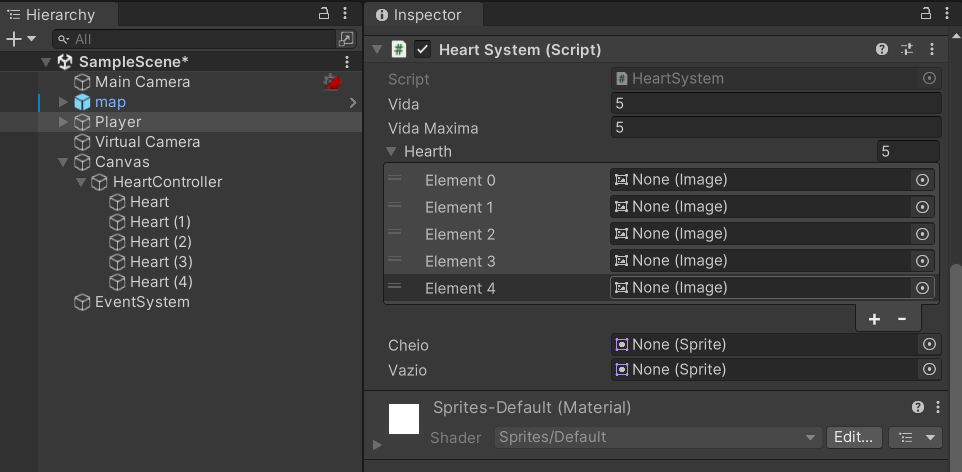
Salve e retorne para Unity e escolha o player e arraste o novo script para o inspector

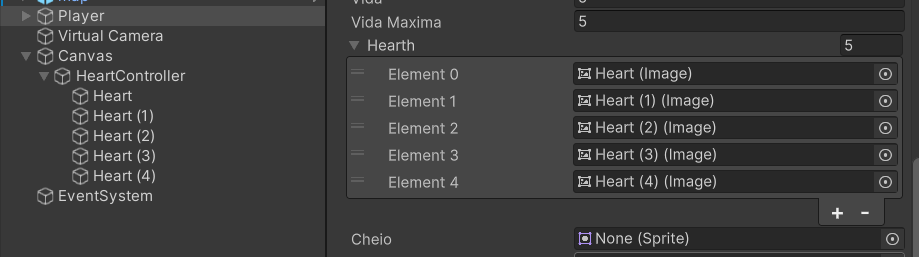


Adicione os valores para a vida do player onde vida e vida máxima são valores que podemos mudar conforme nosso gosto, mas em Hearth adicione o número exato de corações que deixou em tela.



Arraste os corações para dentro dos elementos criados

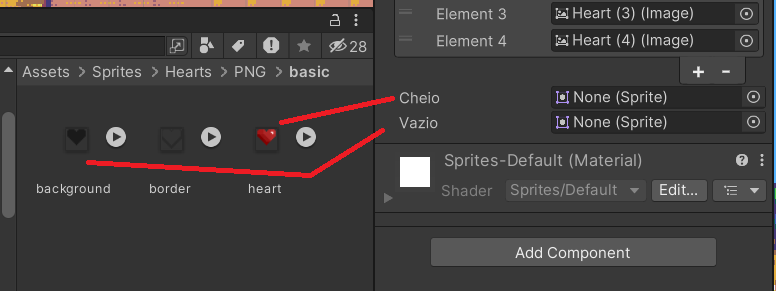


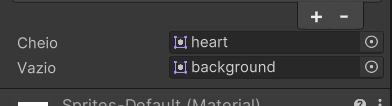




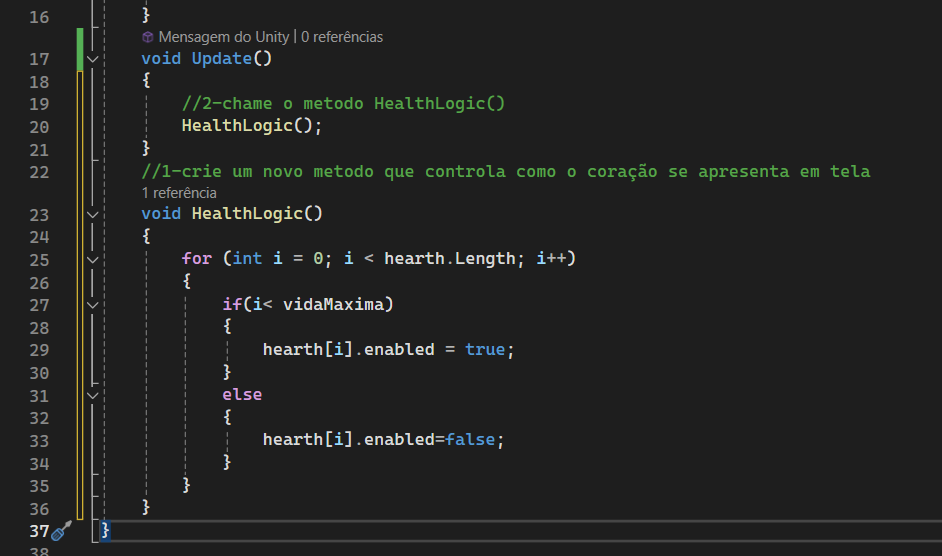
Muito cuidado na ordem de cada objeto com seu elemento. Eles devem seguir a ordem de criação para que eles funcionem corretamente em tela.

Agora podemos mostrar para Unity as imagens de coração cheios e vazios. Abra sua pasta de sprites de corações e arraste para cheio a imagem de coração cheio e em vazio a imagem do coração vazio.





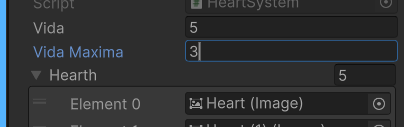
Retorne para o script HeartSystem e adicione a seguinte lógica de funcionamento.



Salve e retorne para Unity. Vamos fazer um teste

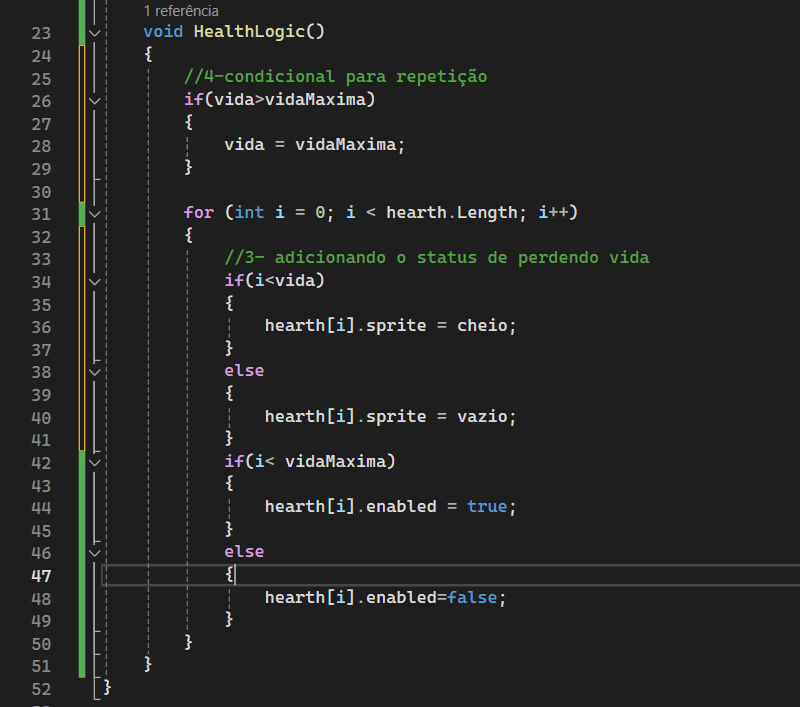


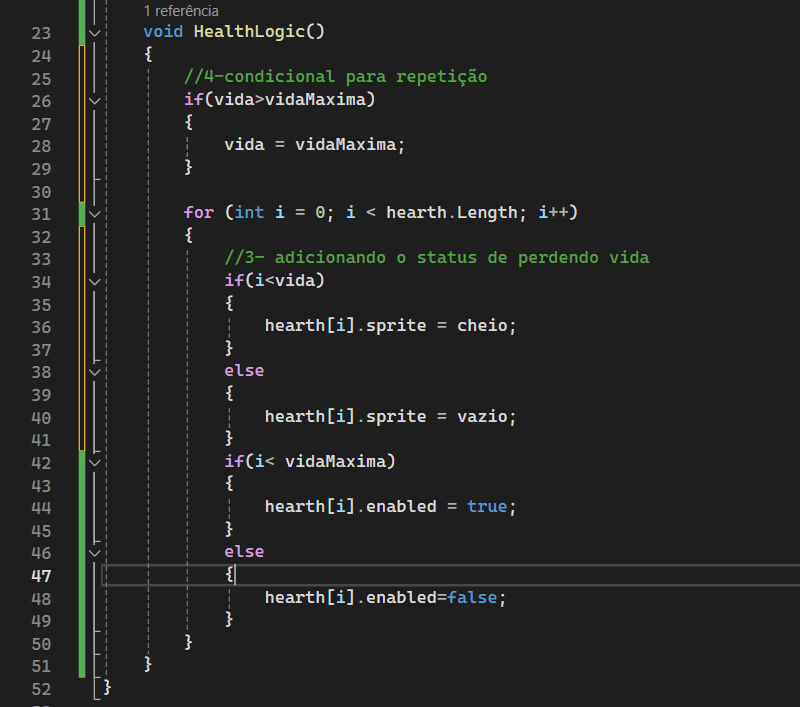
Lembrem-se se caso mude a vida máxima do jogador então estará mudando isso em tela de jogo.



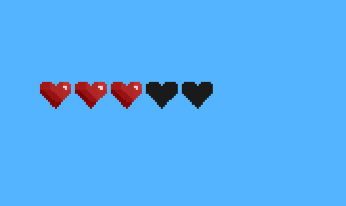
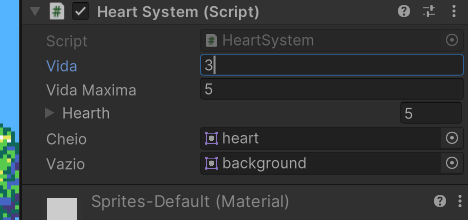


Mas ainda devemos adicionar uma mecânica que mostre que a vida está sendo perdida, então retorne ao script HeartSystem.





Após modificações salve e retorne para Unity. Então se a vida do personagem for diminuída isso acontecerá em tela.



E caso queiramos adicionar mais vida será criado corações vazios



Com isso falta gerar a lógica que une o status do player ao efeito dos corações a mudarem de cor.

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.