Curso Técnico

de Programação de Jogos Digitais

Aula 36

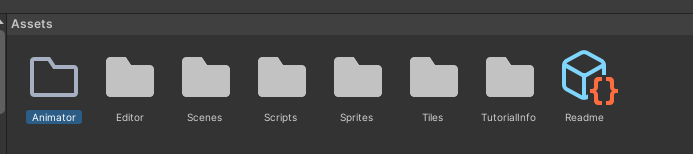
índice

[1. Fluidez no jogo 3](#_Toc170121601)

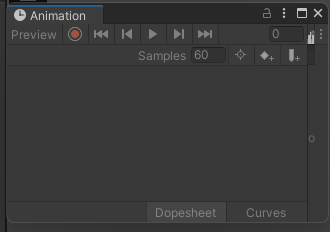
[2. Referencias 9](#_Toc170121602)

# Fluidez no jogo

Vamos trabalhar com alguns elementos de nosso jogo, fazendo que o movimento em tela fique mais fluido. Vamos criar uma pasta para nossas animações. Dentro de Assets crie animator.



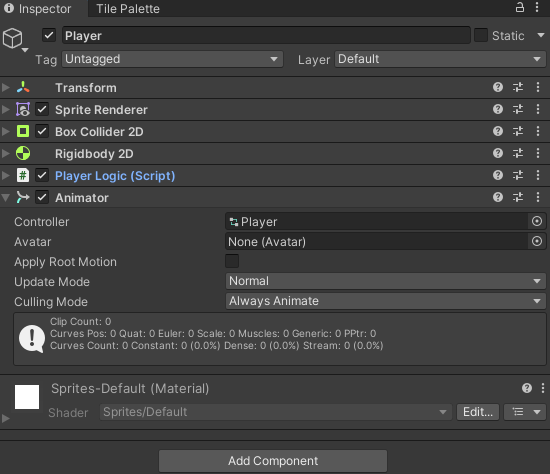
Vamos criar uma janela para nossas animações, seguindo para a barra de ferramentas Windows/Animation/animation



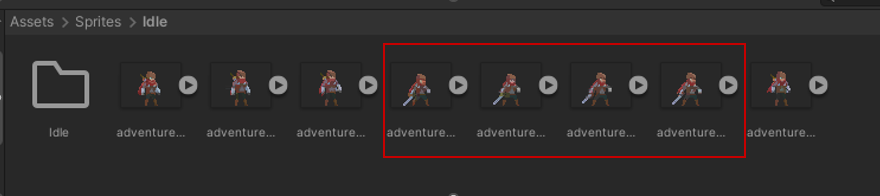
|  |  |
| --- | --- |
| Selecione o player na hierarquia. | Clique em create na janela de animator |
|  |  |

Crie uma nova animação com o nome PlayerIdle. Salve na pasta de animator dentro da pasta assets.

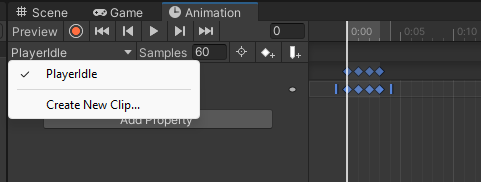
O objeto player possui agora o animator como componente



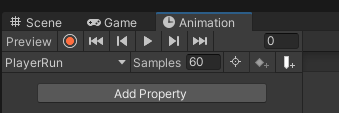
Vamos começar a selecionar o player estático então siga até a pasta Idle e selecione somente as imagens com o personagem parado com a arma



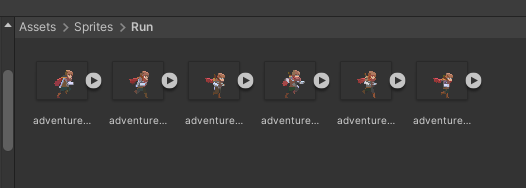
Arraste elas para a animação



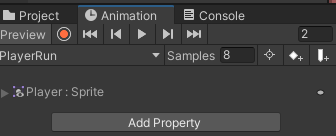
Vamos então criar um novo clipe em Create new Clip com o nome PlayerRun e salvar na pasta Animator dentro da pasta assets. Então podemos iniciar a animação, na janela Animation.



Agora selecionamos as imagens do personagem a correr

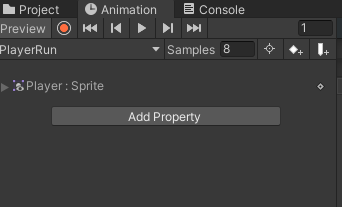


Mude a animação de samples de 60 para 8, fazendo que a imagem não fique tão rápida.

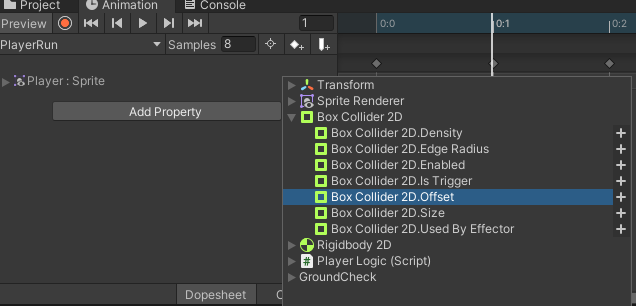


|  |  |
| --- | --- |
| Pressione o play para visualizar a velocidade. | O rigidbody2d não consegue acompanhar o movimento então devemos adicionarmos essa propriedade. |
|  |  |

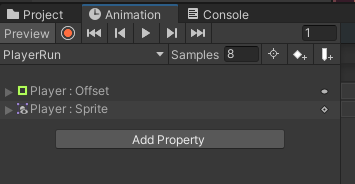
Clique em add property no animator.



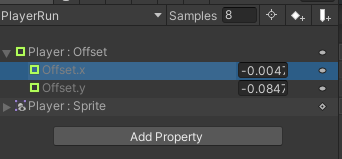
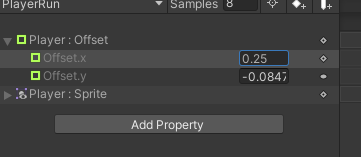
Selecione o boxcollider2d em offset



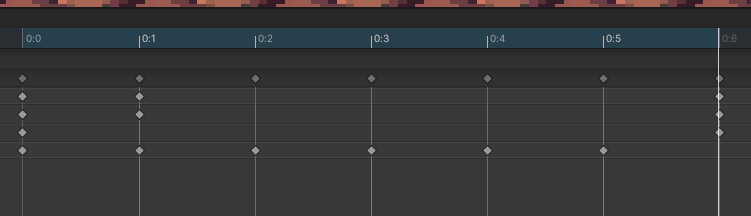
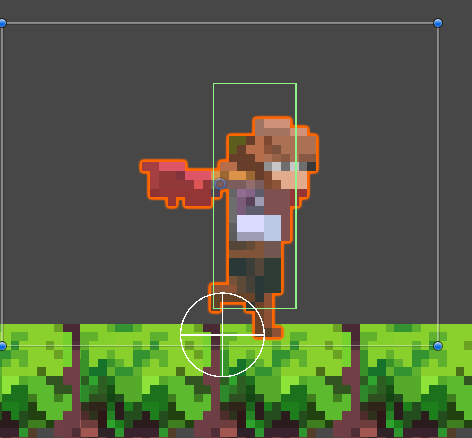
Alguns sprites serão adicionados a linha do tempo



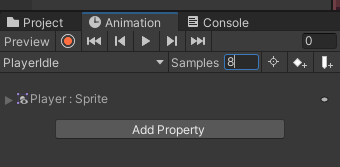
Modifique o eixo x no offset até ficar na posição correta na imagem

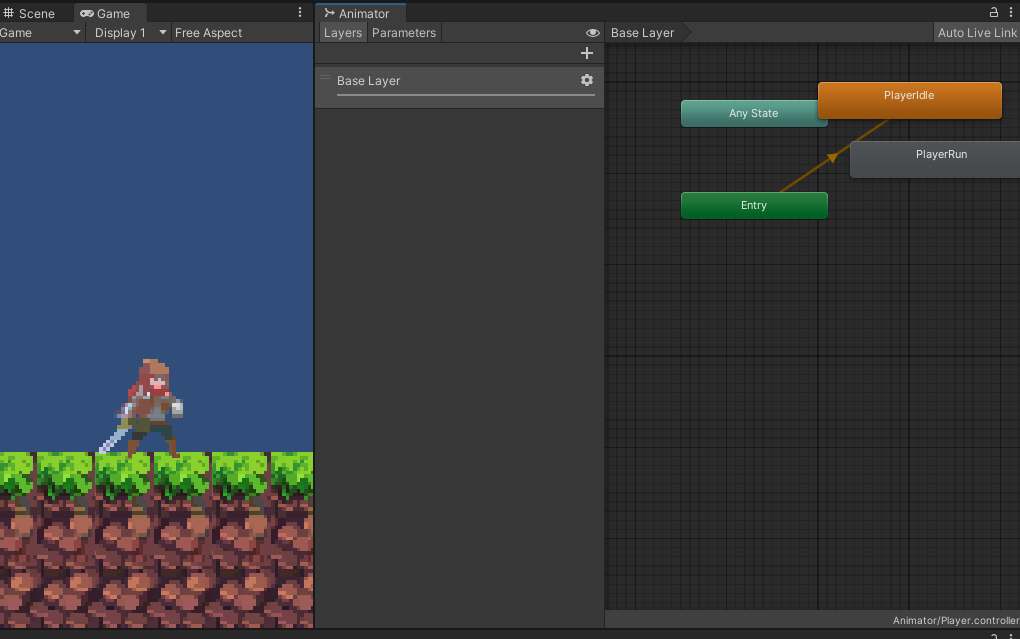
Faça o mesmo no ultimo frame colocando o mesmo valor para x

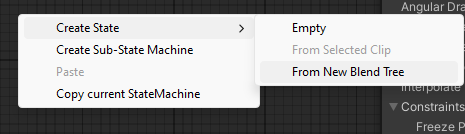
Vá para PlayerIdle e mude o samples para 8 .

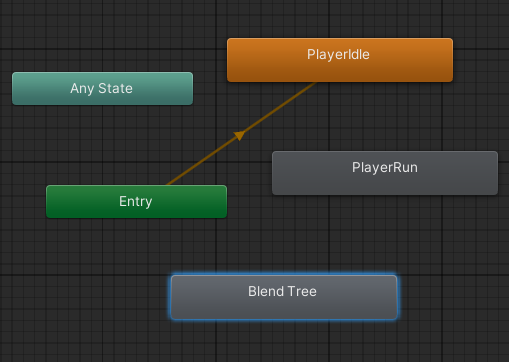


Temos então o personagem com a animação parada e em movimento. Devemos criar agora a lógica para gerar a transição. Vamos para Windows/Animation/animator

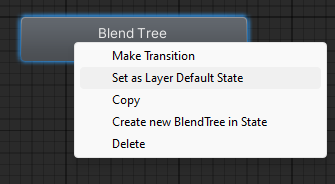


Clique com botão direito e escolha create State/ from New Blend Tree

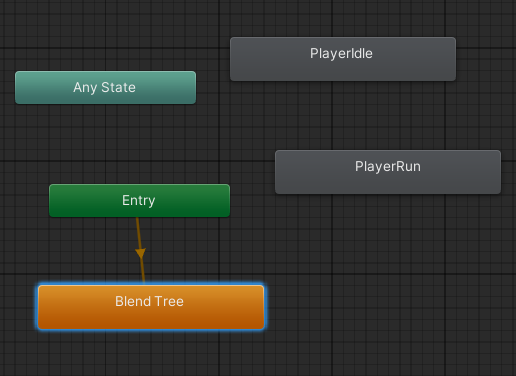
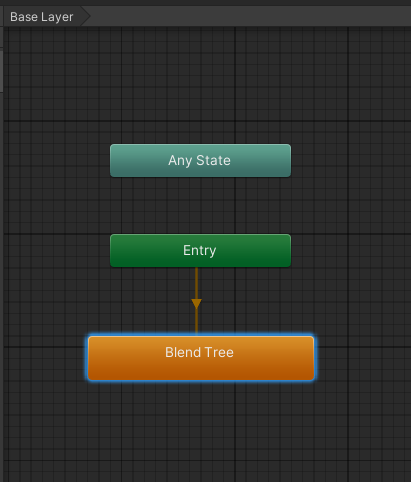




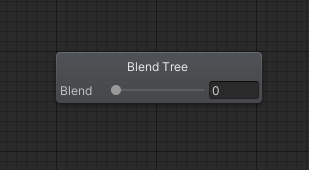
Botão direito emcima do Blend Tree e escolha set as layer defalt state



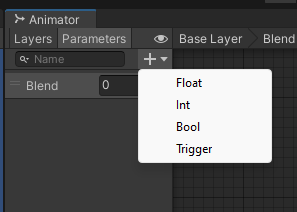
Isso irá fazer que esta seja a primeira animação que seu jogo terá ao iniciar. Podemos deletar a PlayerIdle e PlayerRun

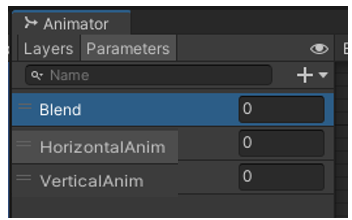
Clique duas vezes na blend Tree



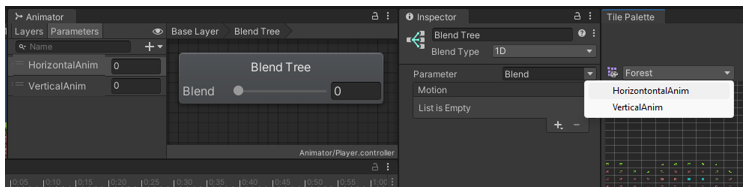
Vamos adicionar alguns parâmetros para que ela funcione.



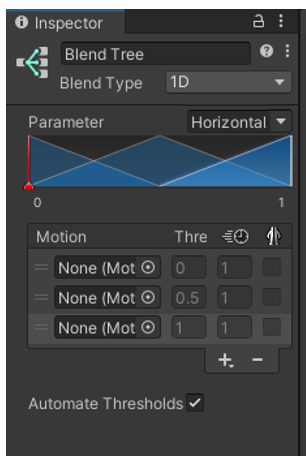
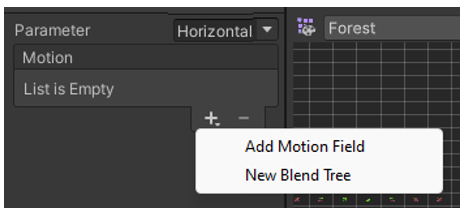
Crie dois parâmetros do tipo float com nome HorizontalAnim e VerticalAnim



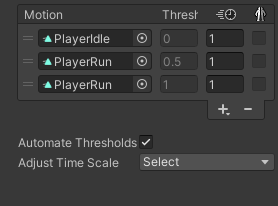
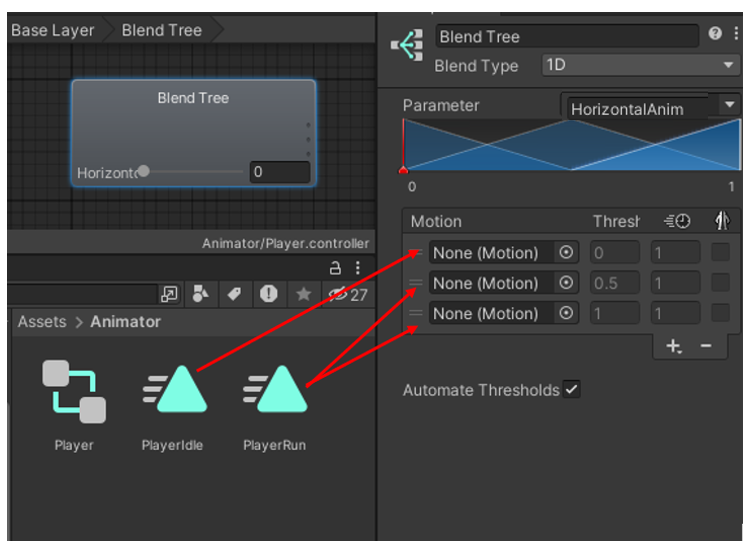
Delete o parâmetro blend e selecione a blend tree para mudar o parâmetro de Blend para HorizontalAnim



Agora vá em motion e adicione 03 motion Field. Ele ficará como a segunda imagem.



Devemos ir até a pasta animator e arrastar o playerIdle para o primeiro parâmetro e PlayerRun para os outros dois espaços



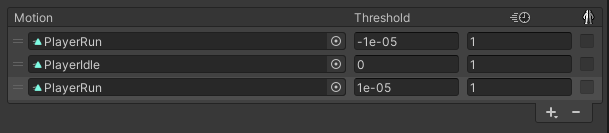
Desmarque o automate Thresholds para modificarmos alguns valores



PlayerIdle: 0

PlayerRun: 0.00001

PlayerRun: - 0.00001



Com estas configurações fazem que o player sem movimento fique com a imagem parada e se mover mudara para a corrida.

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.