Curso Técnico

de Programação de Jogos Digitais

Aula 12

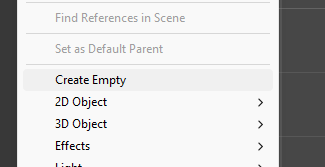
índice

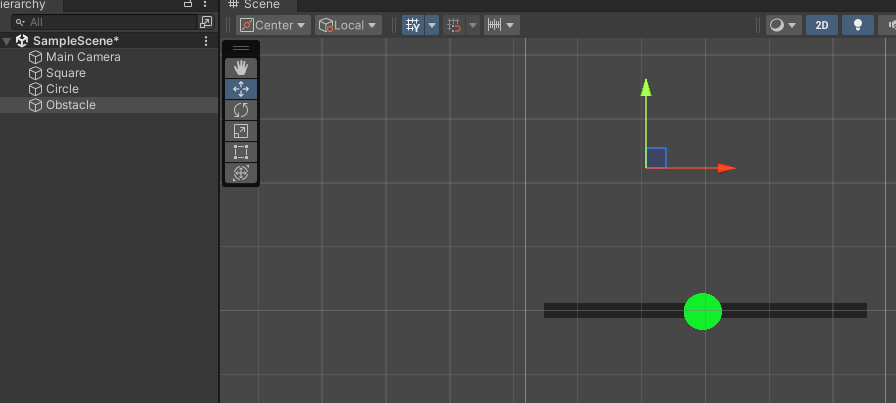
[1. Movimento de objetos 3](#_Toc169508335)

[2. Referencias 8](#_Toc169508336)

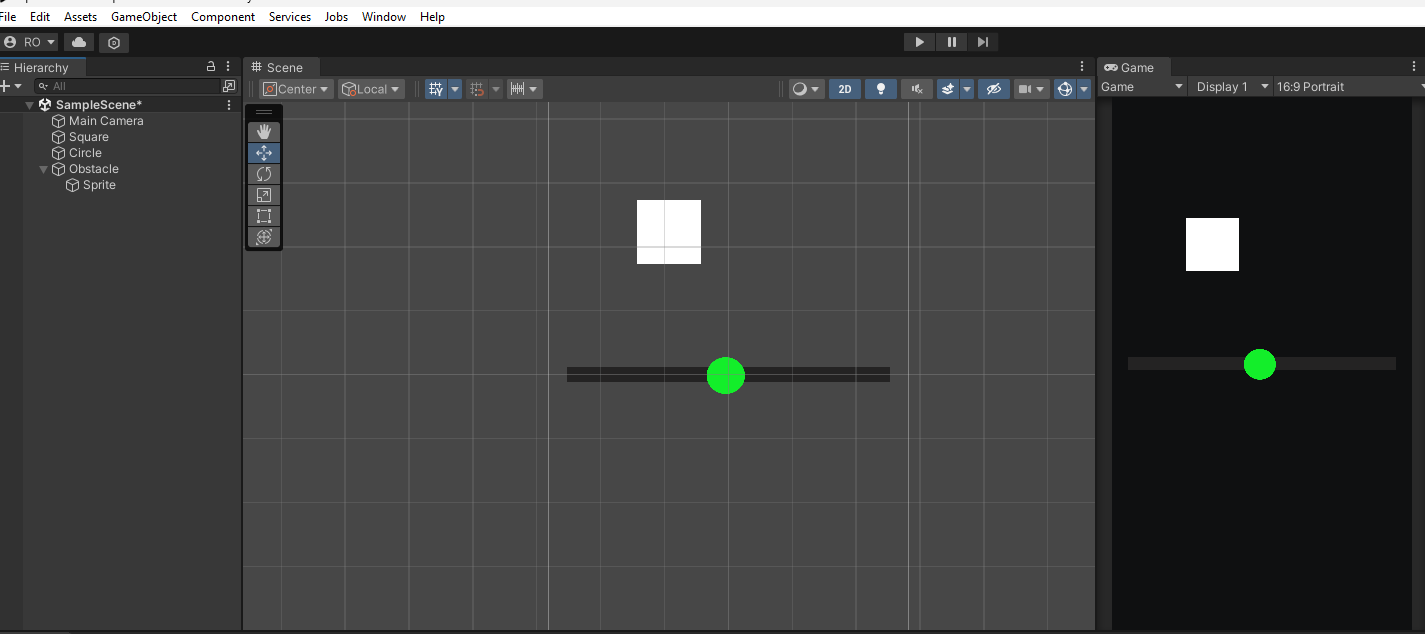
# Movimento de objetos

Agora vamos gerar os objetos que caem do alto, então devemos criar um objeto vazio. Clique botão direito na hierarquia e escolha create empty. Renomeie para Obstacle.

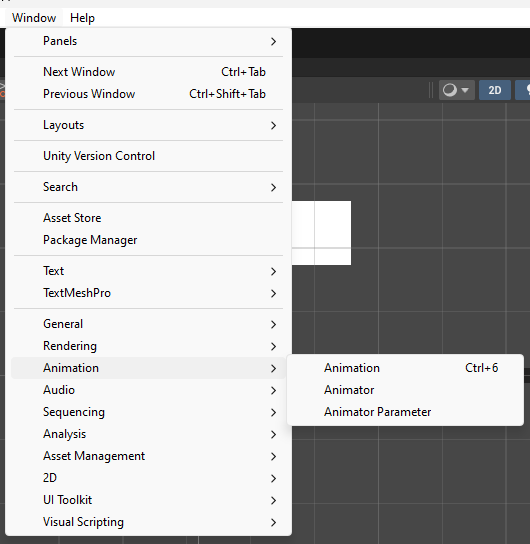




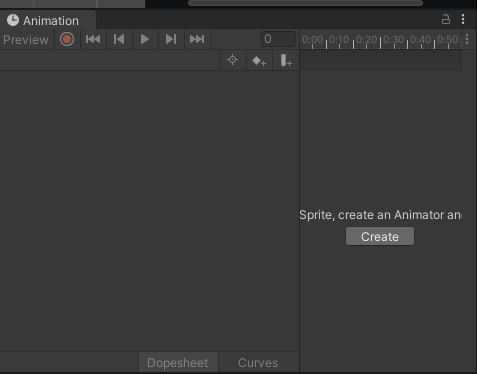
Então dentro do objeto vazio clique com botão direito e escolha square.



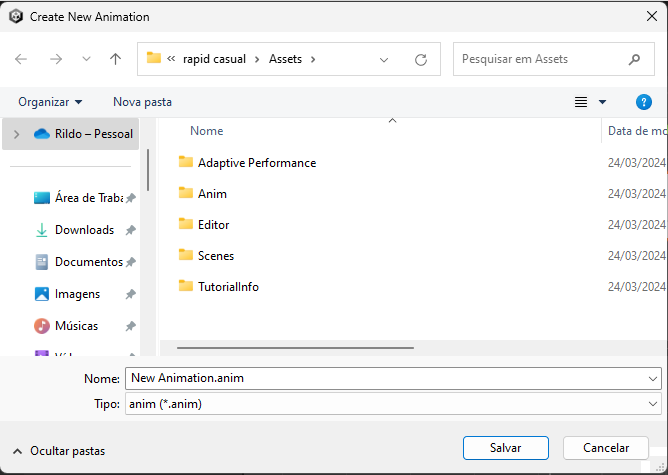
Agora vamos fazer uma animação de rotação em nosso quadrado. Com o quadrado selecionado, procure na barra de ferramentas por Windows e busque por Animation/Animation ou simplesmente aperte Ctrl+f6



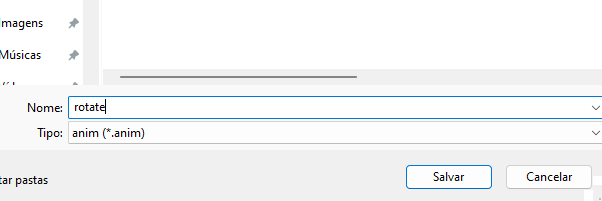
Será criado uma janela de animação.



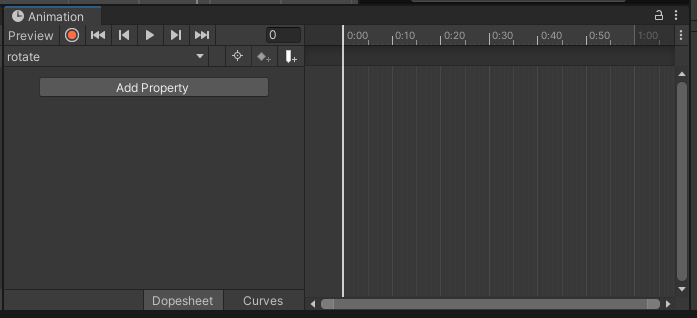
Crie uma pasta Anim dentro da pasta assets. Então com o objeto selecionado clique em create e selecione a pasta assets/Anim para salvar sua animação.



Por motivo de organização criei uma nova pasta Anim para manter nossas animações, mas isso não é obrigatório. No arquivo vamos chama-lo de rotate.

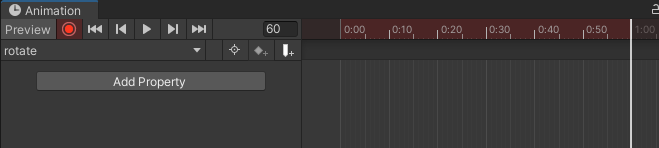


Agora na aba de animação temos uma linha de tempo (timeline)

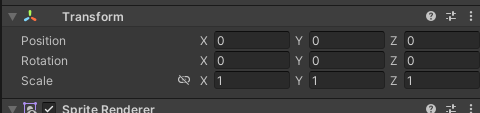


Clique no botão vermelho para iniciar a gravação.

Mude para o frame 60 que será o final da animação.

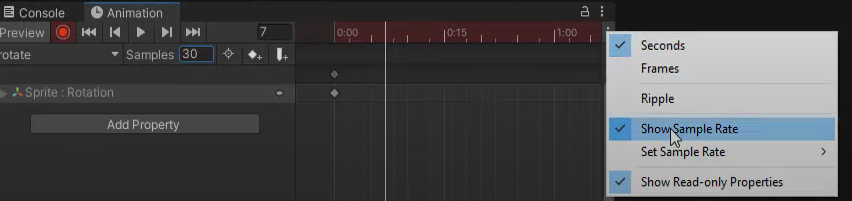


Faça o quadrado rotacionar 180 graus no eixo Z

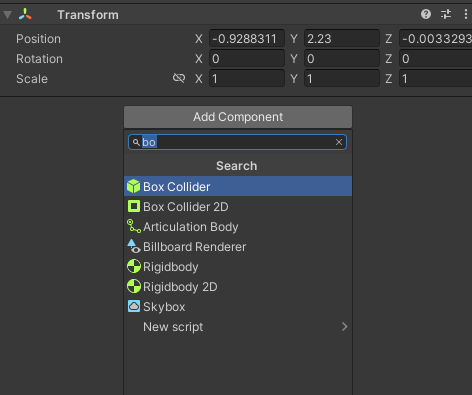




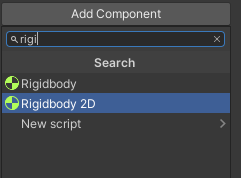
Podemos mudar a velocidade da animação, mudando a quantidade de samples.



Podemos então incluir a gravidade para nosso projeto.Selecione o objeto e vá em add componente



Procure por rigidbody 2d



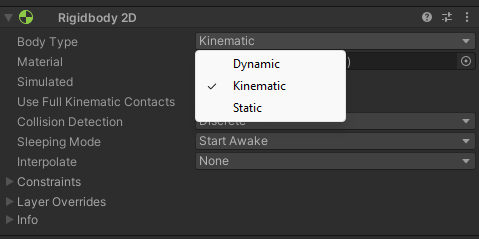
Quando executar o projeto o objeto deverá ter movimento de queda.





Todo movimento de gravidade segue a posição de tela do celular através de seu giroscópio.

Precisamos do rigid body para nossas colisões. Para o momento vamos retirar a gravidade deixando a opção kinematic acionada.



Nosso projeto possui movimento, mas temos de criar algumas outras modificações que ainda serão adicionadas com scripts usando linguagem C Sharp.

# 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.