Curso Técnico

de Programação de Jogos Digitais

Aula 25

índice

[1. Scripts Usados na Unity para Jogos 3](#_Toc169677781)

[2. Estrutura de um Script na Unity 3](#_Toc169677782)

[3. Código para logica do player 5](#_Toc169677783)

[4. Referencias 8](#_Toc169677784)

# Scripts Usados na Unity para Jogos

Os scripts são componentes essenciais no desenvolvimento de jogos na Unity. Eles são responsáveis por definir a lógica e o comportamento dos elementos do jogo, desde a movimentação de personagens até a interação com objetos e o controle do fluxo de jogo. Na Unity, os scripts são escritos principalmente em C, uma linguagem de programação poderosa e versátil.



Os scripts permitem que os desenvolvedores criem interações complexas e dinâmicas em seus jogos. Ao contrário de elementos estáticos, os scripts adicionam vida e funcionalidade aos componentes do jogo.

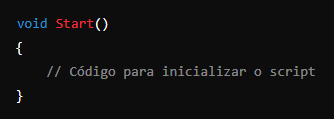
Os scripts são utilizados para:

* Controlar a Lógica do Jogo: Determinam como os elementos do jogo interagem uns com os outros e com o jogador. Por exemplo, scripts podem ser usados para gerenciar a lógica de pontuação, condições de vitória ou derrota, e progressão de níveis.
* Manipular a Física: Permitem que objetos respondam de maneira realista às forças físicas, como gravidade, colisões e impulsos. A Unity oferece uma poderosa engine de física que pode ser controlada através de scripts para criar interações físicas realistas.
* Gerenciar Animações: Controlam quando e como as animações dos personagens e objetos são reproduzidas. Através de scripts, é possível sincronizar movimentos, transições e eventos de animação com precisão.
* Interagir com o Usuário: Capturam e respondem às entradas do jogador, como toques na tela, cliques do mouse ou pressionamentos de teclas. Isso é essencial para criar uma experiência de jogo interativa e responsiva.

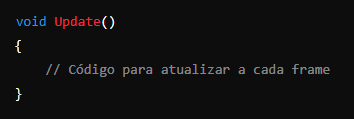
# Estrutura de um Script na Unity

Um script típico na Unity segue uma estrutura básica e implementa funções específicas que a Unity chama em momentos definidos do ciclo de vida de um jogo. Os métodos mais comuns são:

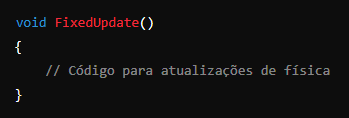
- `Start`: Chamado uma vez quando o script é inicializado. Usado para configurar estados iniciais.



- `Update`: Chamado uma vez por frame. Utilizado para atualizar lógicas que precisam ser verificadas constantemente, como movimentação ou verificação de condições.



- `FixedUpdate`: Chamado em intervalos fixos e é utilizado para atualizações relacionadas à física.

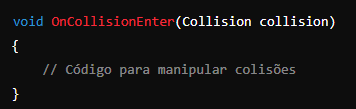




Qual a diferença de uso do Update e do FixedUpdate em um script na Unity?

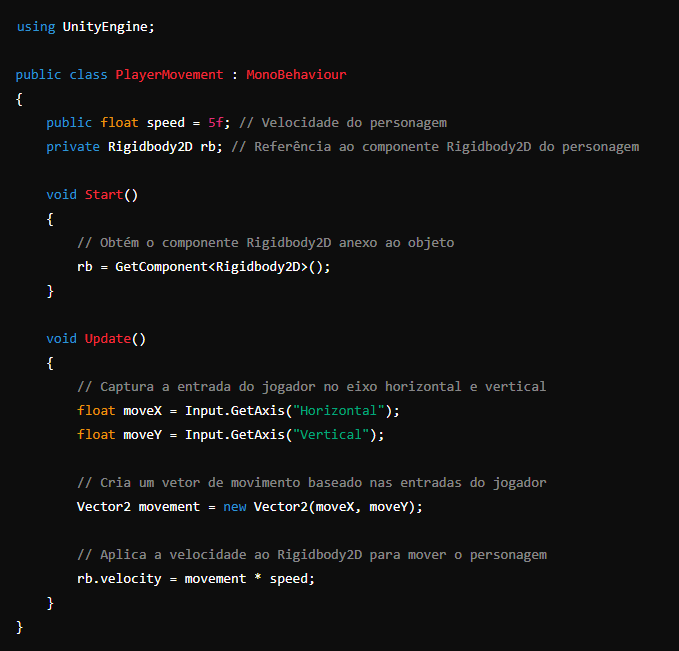
O `Update` é chamado uma vez por frame e é ideal para lógicas dependentes do frame rate, enquanto `FixedUpdate` é chamado em intervalos de tempo fixos e é usado para manipulações de física consistentes.

- `OnCollisionEnter`: Chamado quando o objeto com um collider anexo colide com outro objeto.



Exemplo de Script: Movimentação de Personagem.

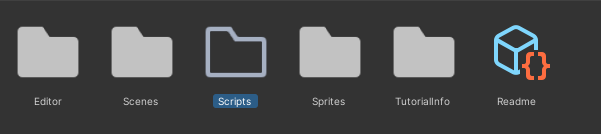
Aqui está um exemplo simples de um script que controla a movimentação de um personagem em um jogo 2D:



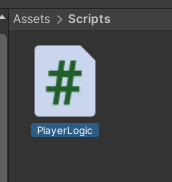
Este script captura as entradas do jogador através das teclas de seta ou WASD e move o personagem de acordo com essas entradas, aplicando velocidade ao Rigidbody2D.

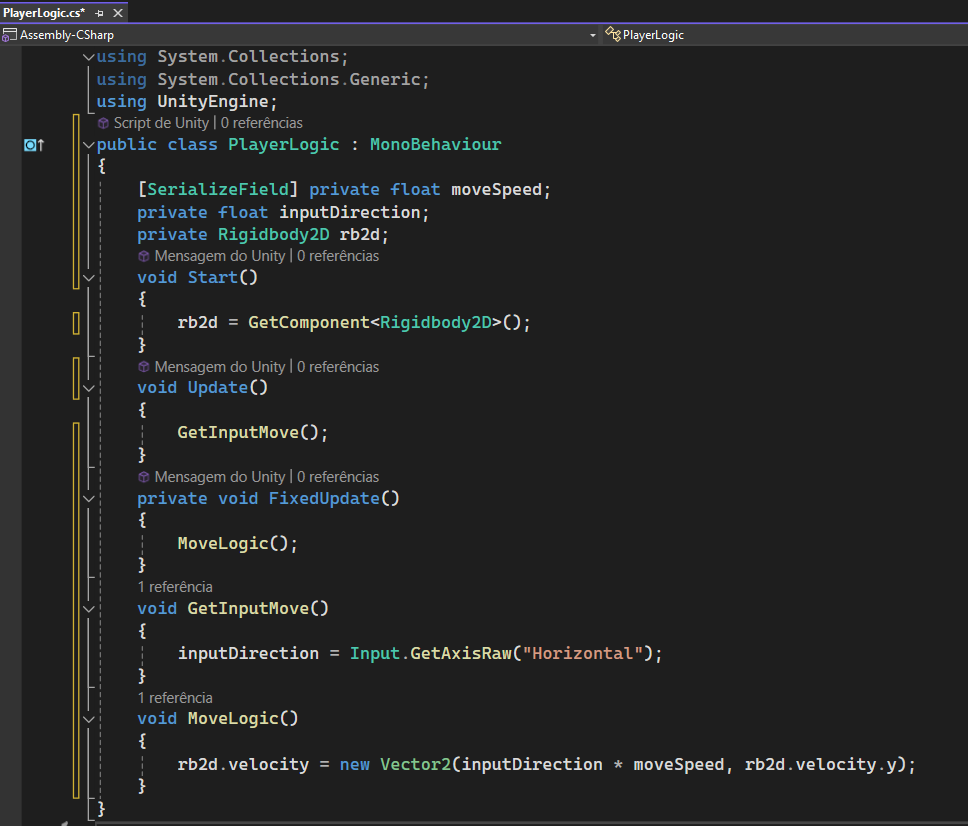
# Código para logica do player

Abra a pasta de scripts.

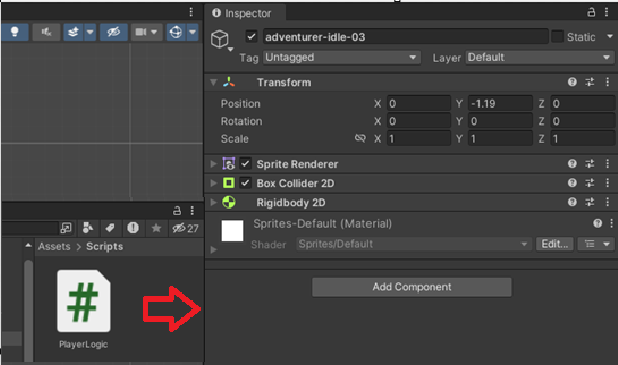


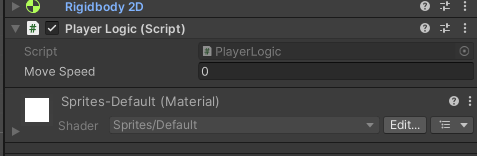
Então devemos criar nosso primeiro script para o player e adicionar o código a seguir.



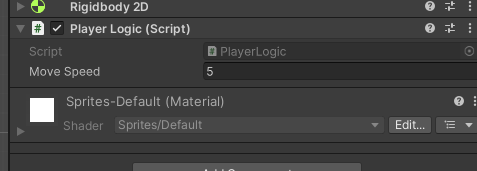


Vamos arrastar o script para o player.





Aumente a velocidade para gerar o movimento.



Então vamos testar o movimento.



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.