Curso Técnico

de Programação de Jogos Digitais

Aula 17

índice

[1. O que são Objetos Prefab na Unity? 3](#_Toc169590912)

[2. Prefabricados 4](#_Toc169590913)

[3. Referencias 8](#_Toc169590914)

# O que são Objetos Prefab na Unity?

Um Prefab é essencialmente uma "plantilha" de um objeto de jogo. Ele armazena uma coleção de componentes e configurações que podem ser instanciados repetidamente em cenas diferentes ou várias vezes dentro da mesma cena. Uma vez criado, o Prefab pode ser atualizado, e essas atualizações podem ser refletidas em todas as suas instâncias.

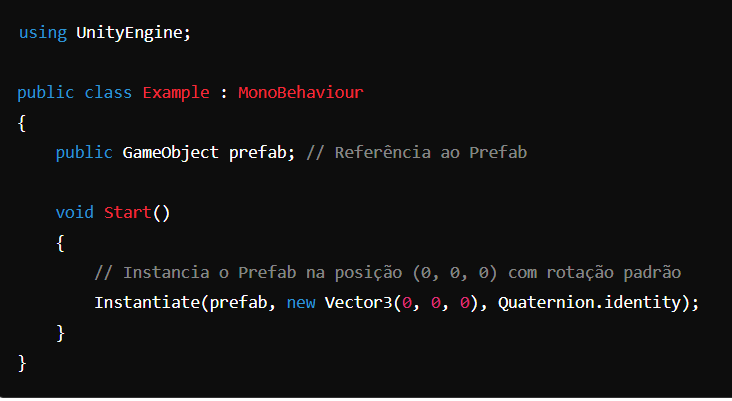
Para criar um Prefab, você deve:

1. Criar um Objeto de Jogo: Monte o objeto de jogo na cena com todos os componentes necessários (Transform, Rigidbody, Scripts, etc.).

2. Arrastar para a Pasta de Prefabs: Arraste o objeto de jogo da hierarquia para a pasta de Prefabs no Project Window. Isso cria um arquivo de Prefab que pode ser reutilizado.

Exemplo de Código

Abaixo, um exemplo de como instanciar um Prefab em um script C#:



Vantagens dos Prefabs

1. Reutilização: Prefabs permitem que você reutilize objetos de jogo complexos sem precisar recriá-los ou configurar cada componente manualmente.

2. Consistência: Manter a consistência entre várias instâncias de um objeto de jogo é mais fácil. Atualizações feitas no Prefab são refletidas em todas as suas instâncias.

3. Desempenho: Instanciar Prefabs em tempo de execução é mais eficiente do que criar novos objetos de jogo do zero.

4. Organização: Facilita a organização do projeto, mantendo objetos complexos e suas configurações centralizadas.



Prefabs são uma ferramenta na Unity que permite a criação e gerenciamento eficiente de objetos de jogo complexos.

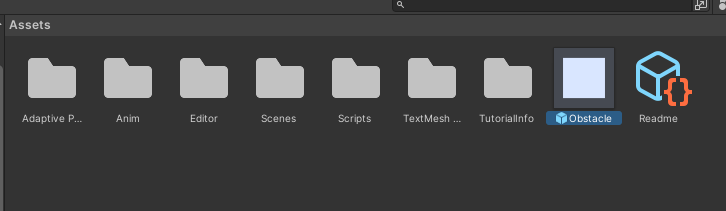
Prefabs são utilizados em diversas situações, como:

* Criação de Inimigos: Gerar múltiplos inimigos idênticos em diferentes partes de uma cena.
* Instanciação Dinâmica: Criar objetos durante o jogo, como projéteis disparados pelo jogador ou itens coletáveis.
* Gerenciamento de Recursos: Facilitar a gestão de recursos, como árvores em um jogo de mundo aberto.

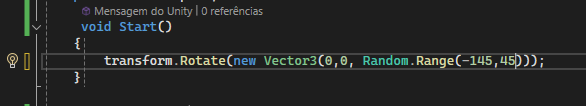
# Prefabricados

Como o objeto obstacle será sempre usado para gerar vários objetos é interessante transformá-lo em um modelo Pré fabricado ou prefab.

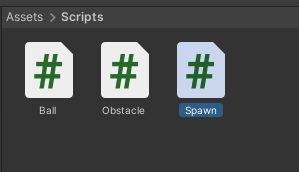
Para isso basta arrastar o elemento para dentro da pasta assets.

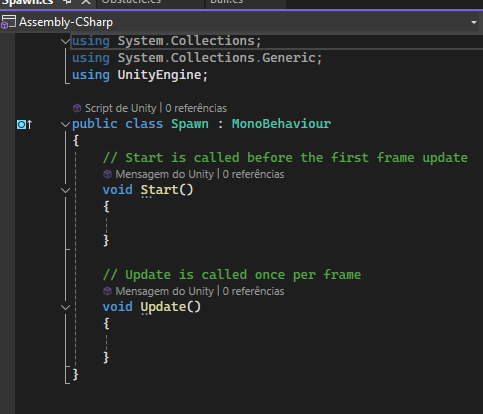


Vamos fazer que o obstacle sempre vá para baixo em direção a bolinha



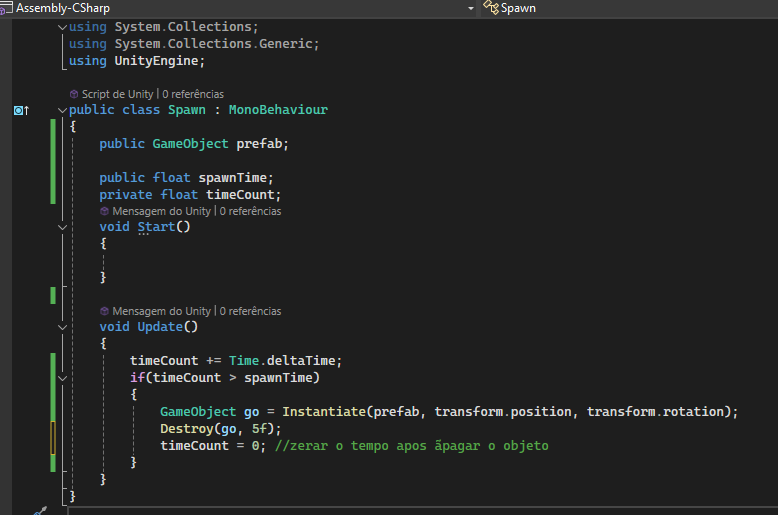
Temos o modelo de nosso objeto então vamos gerar agora um criador de objetos repetidos .Então devemos criar um novo script, o Spawn.

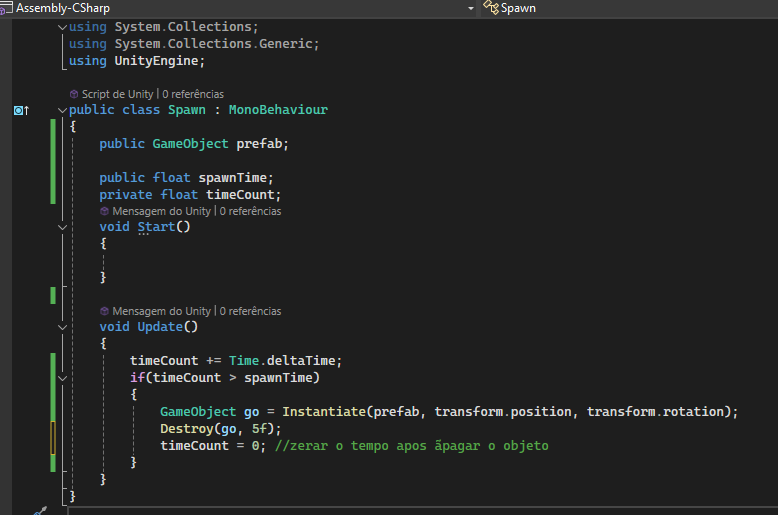




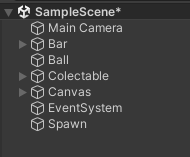
Vamos então no script do Spawn gerar uma regra de sempre criar nossos obstáculos conforme o tempo passa e também que eles sejam destruídos após algum tempo.

Crie uma variável para controlar o tempo de criação de cada objeto repetido ( spawnTime) e um contador de tempo (timeCount).

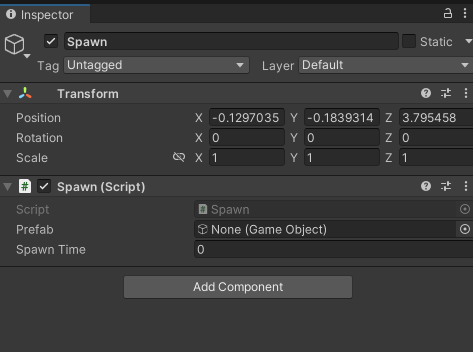


Então no update gere a logica de criação do objeto e tempo máximo de duração em tela.

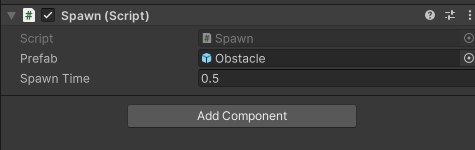
Crie na Unity um objeto vazio para Spawn



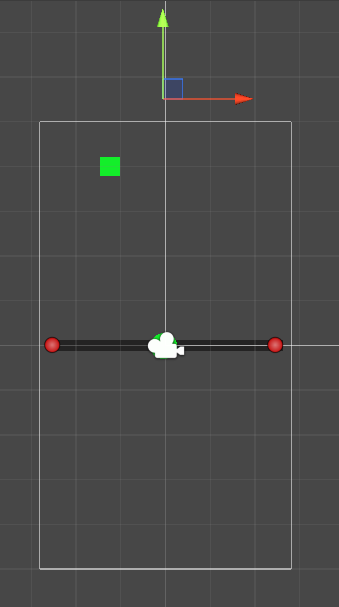
Arraste o script do Spawn para o Spawn



Vá a pasta assets e selecione o gameobjet do obstacle para o prefab do Spawn e adicione a velocidade



Arraste o objeto Spawn para fora da câmera, fazendo que os novos objetos sejam gerados acima da visão da tela do jogo.



Faça mudança de coordenadas como ver necessário , aprendendo assim como modificar o script.

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.