

MOVIMENTO HORIZONTAL E FÍSICA BÁSICA

PROF. ME. HÉLIO ESPERIDIÃO

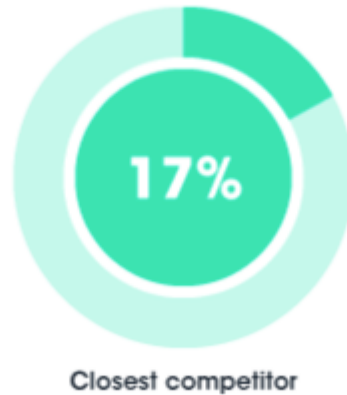
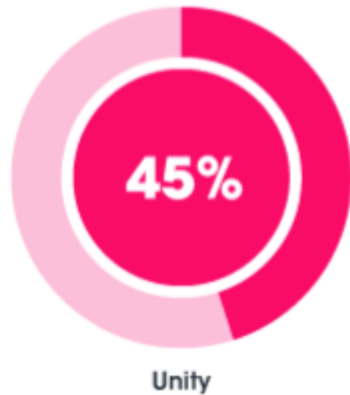


Unity

Fatia de mercado mundial para o desenvolvimento de jogos.

34% dos 1.000 maiores jogos mobile gratuitos são feitos com Unity.

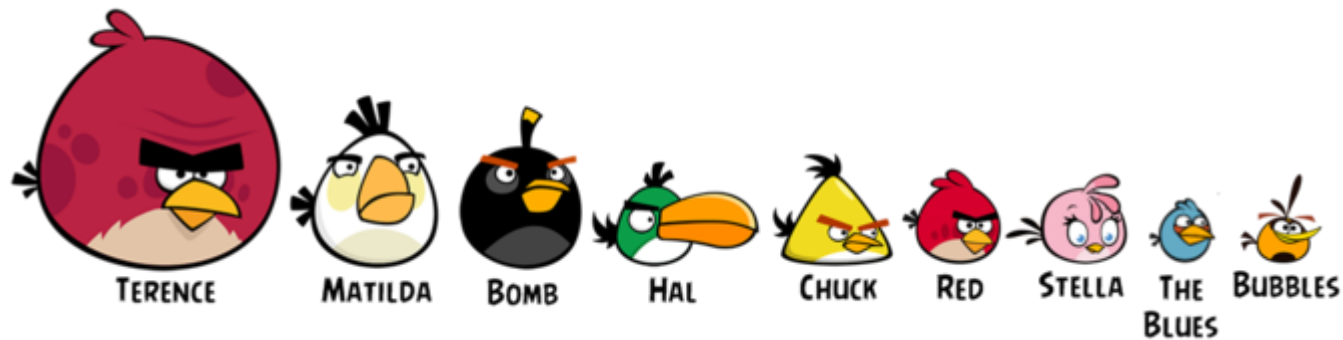
Global game engine market share



Popularidade do Unity

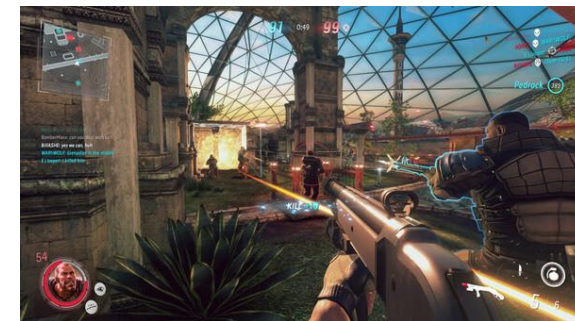
Essa enorme popularidade faz jus à sua capacidade: a game engine permite criar jogos em 2D ou 3D com os mais diversos estilos de gráficos e mecânicas e para diferentes plataformas.

Muitos jogos famosos, como Angry Birds 2, Bad Piggies, Roller Coaster Tycoon World e até o Pokemon GO foram criados com ela



Mais Jogos criados com Unity

Assassin's Creed: Identity
Temple Run Trilogy
Deus Ex: The Fall
Ballistic Overkill
Knights of Pen and Paper



Game Engine

Game engine (ou, em português, motor de jogo), consiste em um conjunto de ferramentas capazes de facilitar o desenvolvimento de um jogo.

Geralmente possuem recursos para criação de funções gráficas, física aos objetos, trilhas sonoras, entre outras ações.

Por que é o mais usado?

Facilidade de aprendizado

Muita documentação e fórum de discussões na internet

Cria/portar games para várias plataformas

Versão gratuita

Crie jogos do Sistema Operacional que preferir

Jogos em 2D e 3D

Assets

Asset significa: Ativo.

Em tecnologia os "assets" são os recursos do seu projeto. O seu "banco" de bibliotecas.

No caso de jogos pode ser entendido como o conjunto de imagens, gráficos, sons e recursos que são utilizados para o desenvolvimento do jogo.



Onde encontrar?

Os assets podem ser construídos pelo próprio desenvolvedor ou encontrados na web nas seguintes formas de distribuição: Gratuito ou pago.

Sites com conteúdo gratuito:

- <https://www.gameart2d.com/freebies.html>
- <http://kenney.nl/assets>
- <http://unluckystudio.com/category/freegameart/>
- <https://itch.io/game-assets/free>
- <http://spritedatabase.net/>
- <https://www.spritters-resource.com/>
- <https://craftpix.net/>
- <https://opengameart.org/>

Unity

Executa o jogo

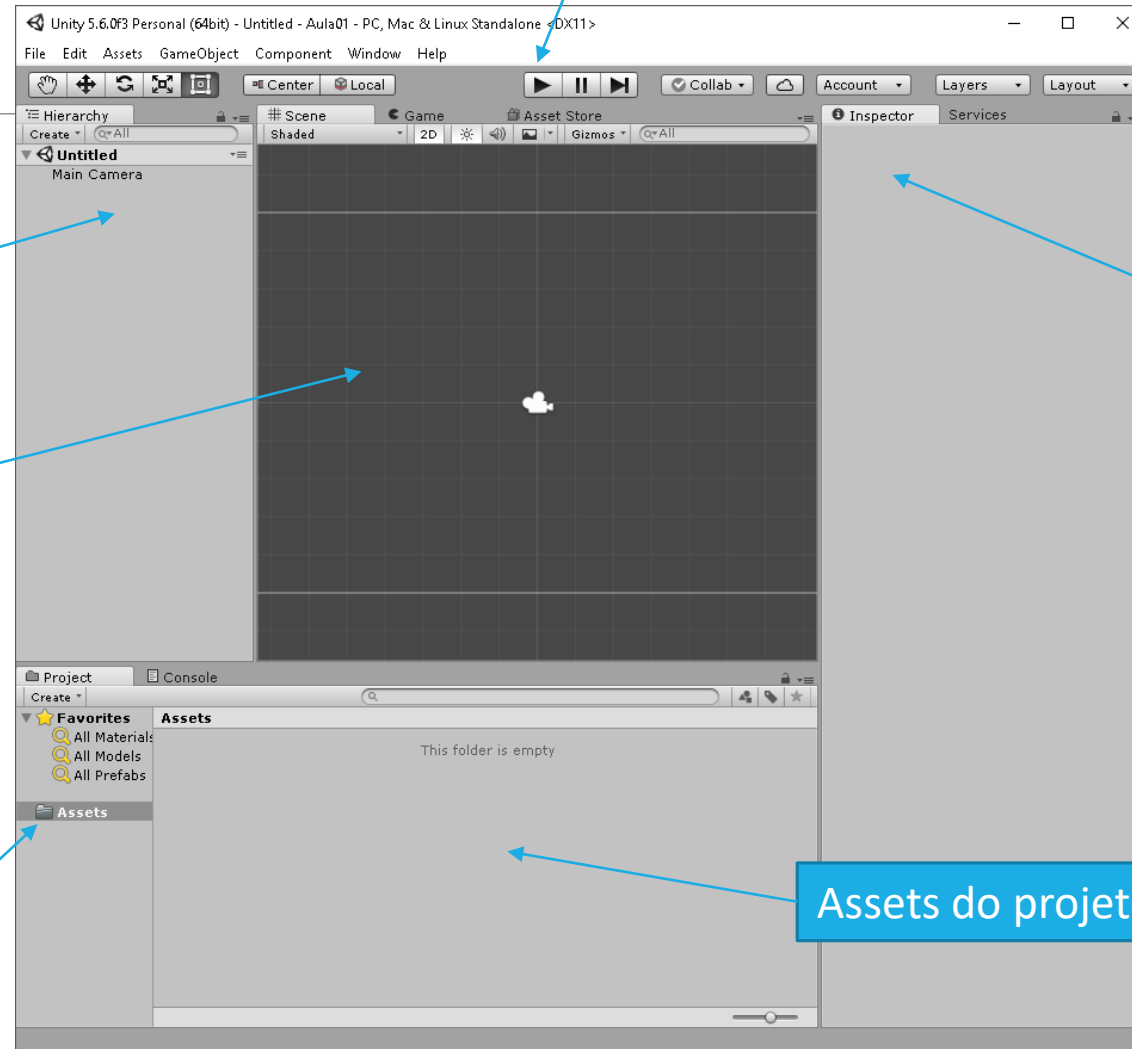
Lista de elementos de cena

Cenário

Propriedades de um elemento de cena

Pastas do projeto

Assets do projeto



Acrescente seus Assets

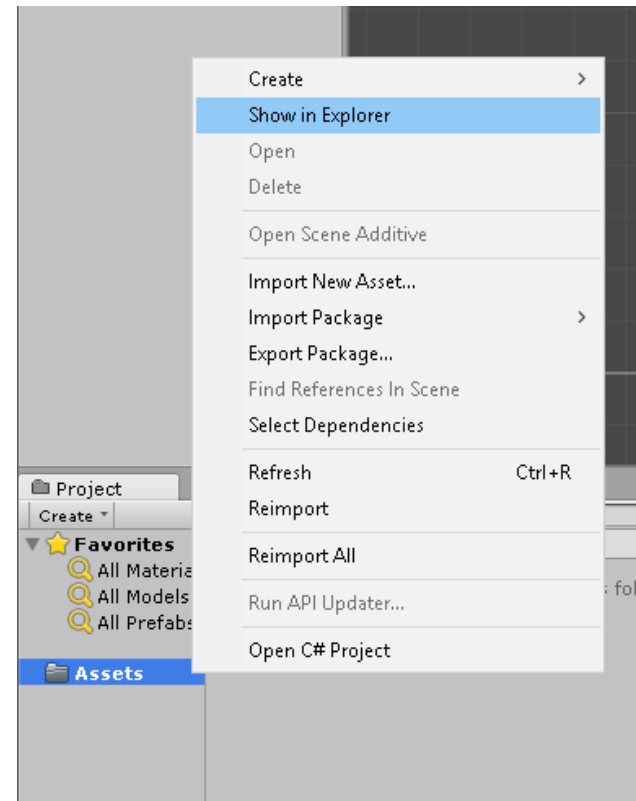
Clique o botão direito do mouse sobre a pasta Assets.

Após abrir o menu escolha a opção

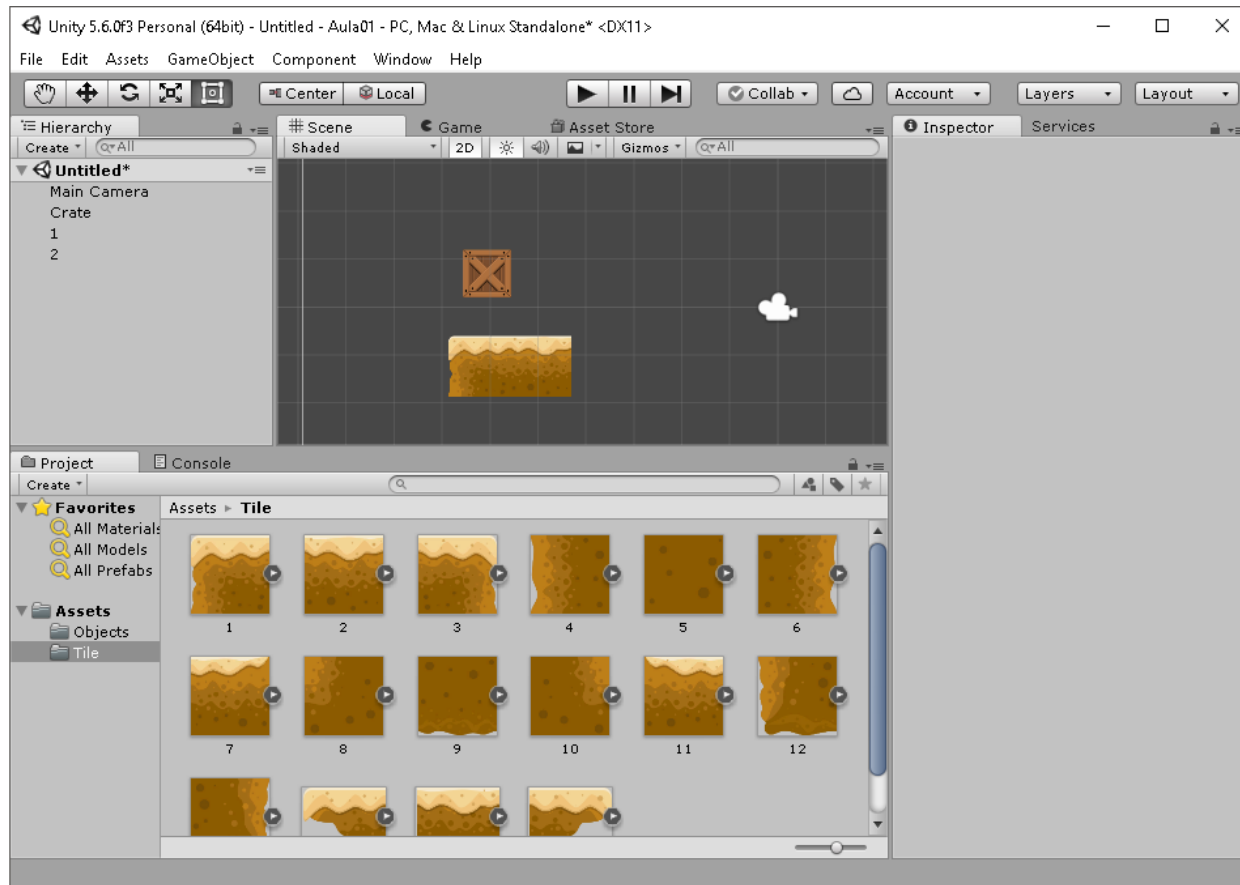
“Show in Explore”

O unity irá abrir uma pasta e você deve colocar todos os seus assets dentro.

Caso possua muitos assets crie mais pastas dentro da pasta Assets para manter a organização



Posicione os Assets no seu cenário



Conceitos sobre física

Física em jogos se refere a uma simulação controlada pela própria engine.

No Unity, existem duas engines de física disponíveis, uma 2D, baseada na Box2D, e outra em 3D, chamada PhysX.

Corpos rígidos

O principal tipo de física usado em jogos é a física de corpos rígidos, que considera que objetos não deformam ou alteram seu volume e material quando afetados por interações como colisões.

RigidBody2D

- Componentes que marcam um objeto como parte da simulação física. Configura propriedades como massa, velocidade e arrasto, e permite o controle de como esse objeto se movimenta e gira em resposta a forças.
- O componente também guarda métodos para a aplicação de forças sobre o objeto.

Collider

Componentes que marcam o volume físico de um objeto, assim como o material físico (com atrito e elasticidade) que define parte de seu comportamento.

Existem vários tipos de colisores, cada um com formato diferente (por exemplo, BoxCollider, SphereCollider).

Também mantem informações sobre o tipo de interação que tem com outros objetos (colisão ou sobreposição). Esse componente tem métodos e eventos que utilizamos para criar lógica para interações físicas, como OnCollisionEnter e OnCollisionExit. Para que um script possa utilizar essas funções-evento, o objeto a que está atrelado tem que ter um Collider.

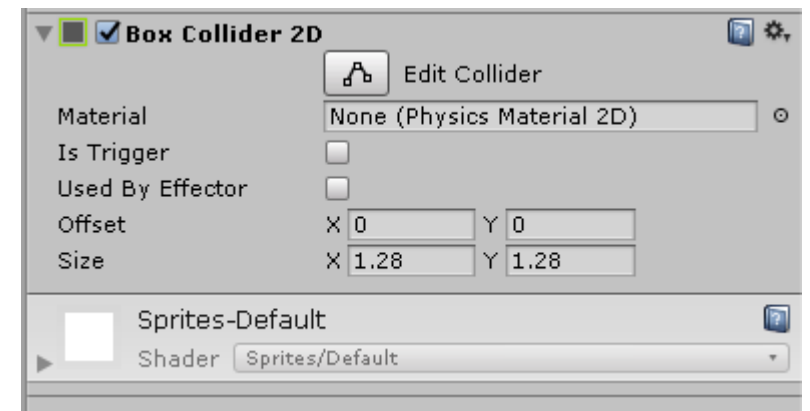
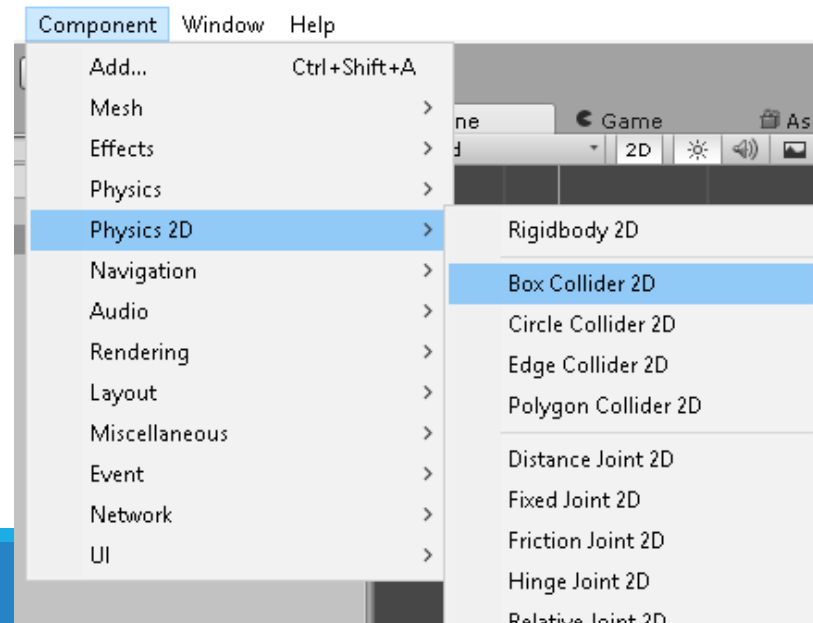
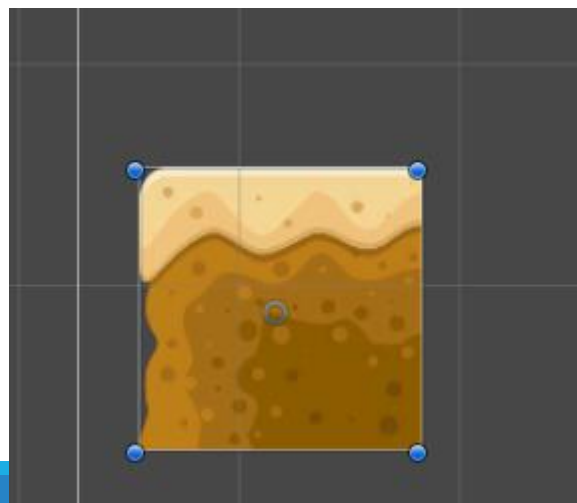
Colisão de elementos

Acrescente o chão do cenário.

Clique sobre o componente.

Vá ao menu Component >> Physics2D>>Box Collider 2D

Faça para todos os elementos onde pode ocorrer colisão.



RigidBody 2D

Acrescento o caixote

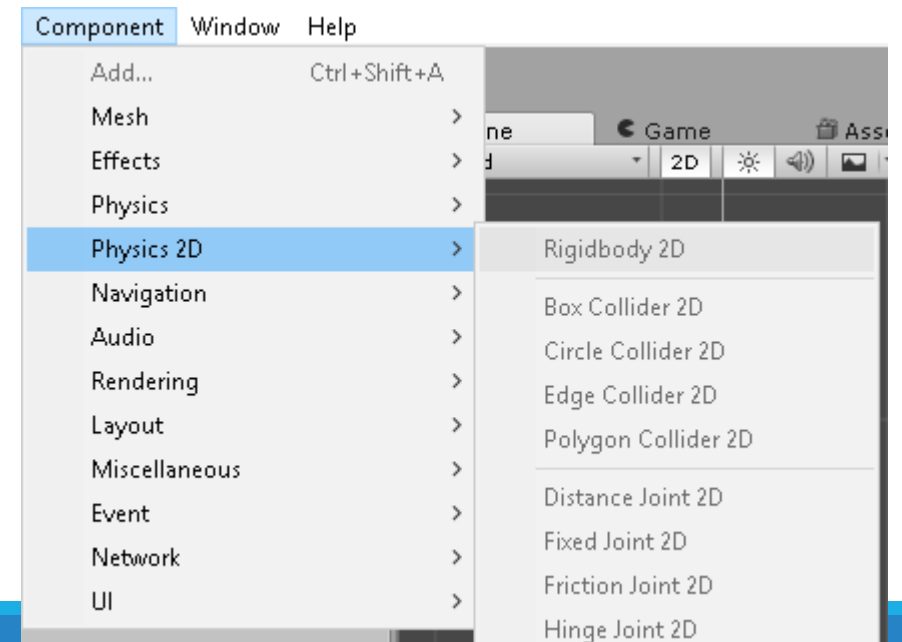
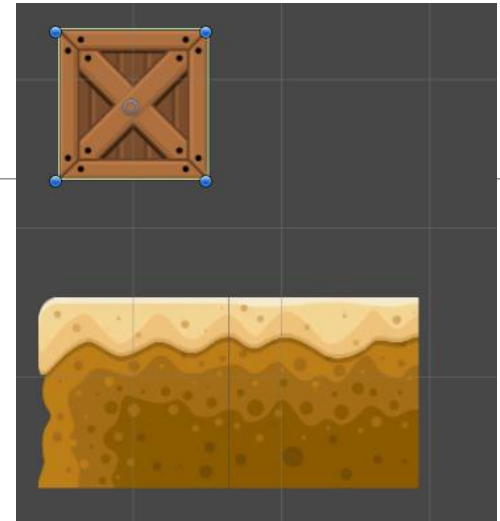
Defina o como caixa de colisão 2D

- Vá ao menu Component >> Physics2D>>Box Collider 2D

Acrescente Física ao de corpo Rígido

- Vá ao menu menu Component >> Physics2D>>RigidBody 2D

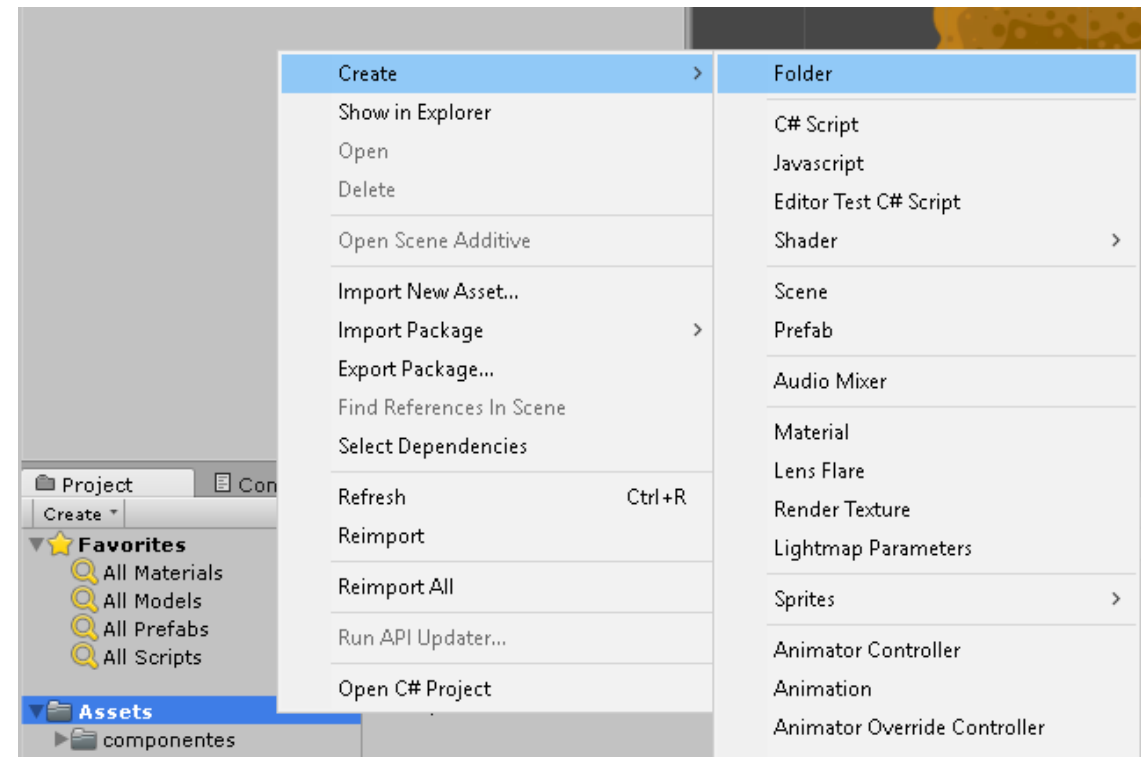
Como a caixa é um corpo rígido ao rodar o jogo ela vai cair se se colidir com o chão que por sinal é uma caixa de colisão.



Movimentação

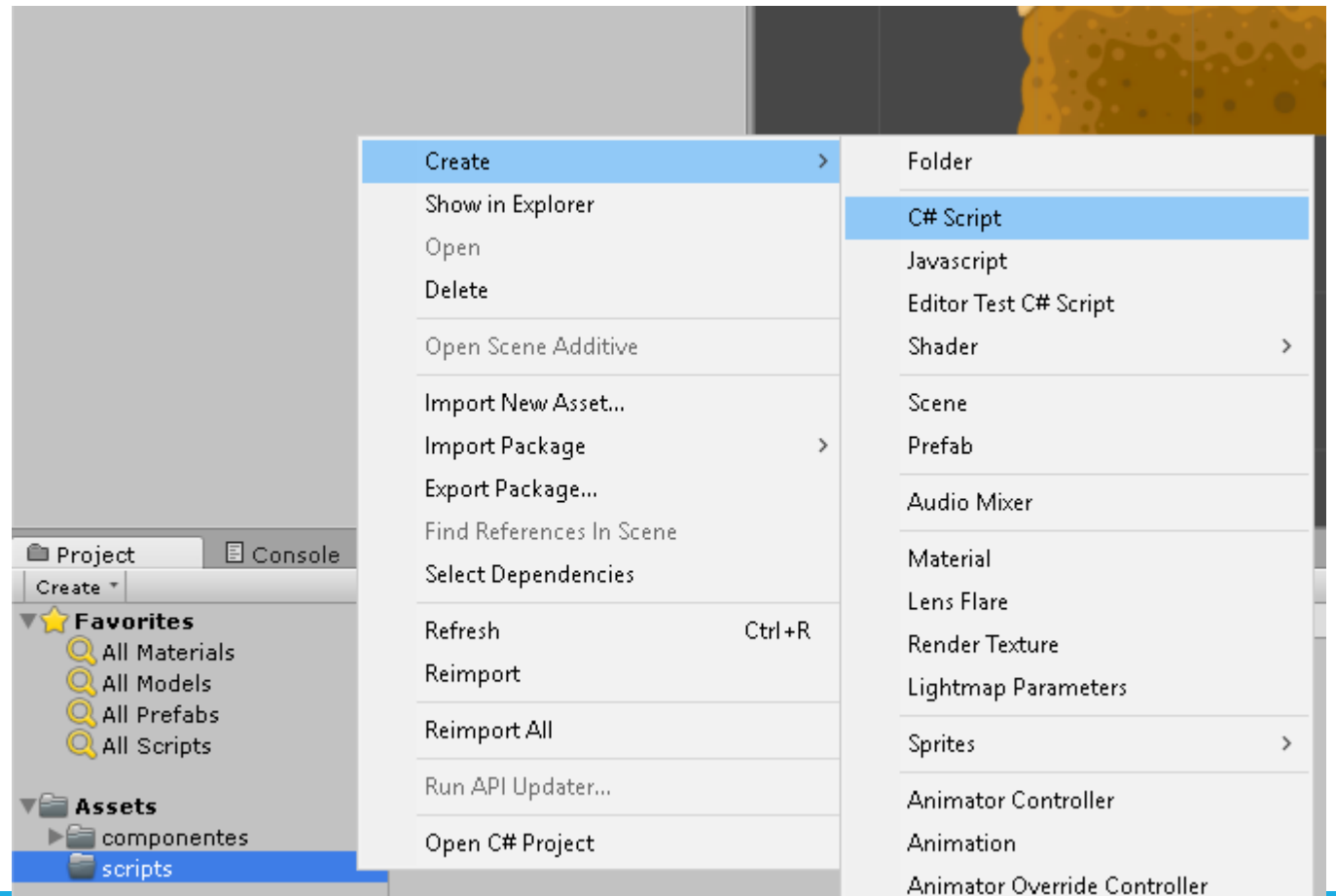
Para movimentação primeiramente crie uma pasta para organizar seus códigos.

Crie uma pasta chamada scripts



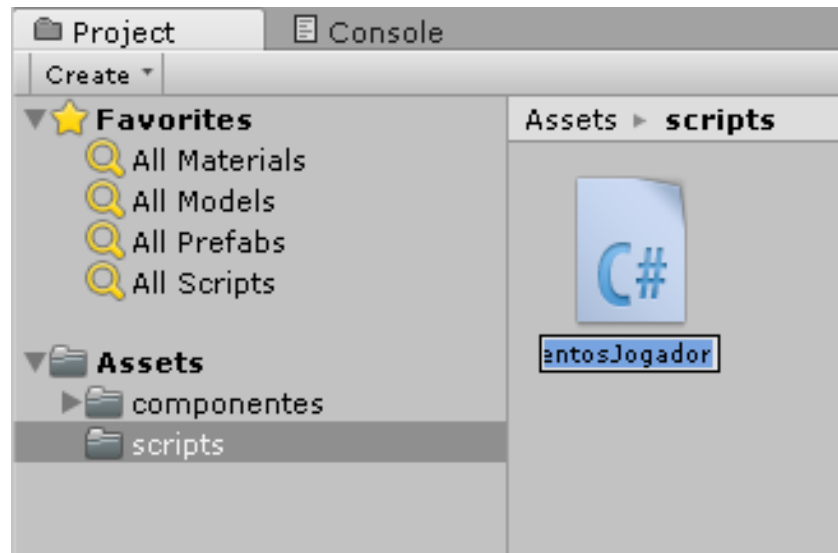
Script

Clique com o botão direito sobre a pasta scripts e escolha a opção: Create >> C# Script



Renomeie

Mude o nome do arquivo criado para : movimentosJogador e efetue dois cliques para editar o script



Entenda o Script

Nome do Arquivo

Características de comportamento individual.

```
using UnityEngine;
using System.Collections;

public class MovimentoJogador : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }

}
```

Executa somente quando
O jogo é iniciado

Executa a todo momento

Atributos do Personagem

Os atributos neste momento do curso podem ser considerados variáveis que devem ser utilizadas para representar características do personagem.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Personagem : MonoBehaviour {

    //A propriedade personagemRigidbody2D faz
    //referência ao Rigidbody2D que foi adicionado ao personagem
    public Rigidbody2D personagemRigidbody2D;
    public float direcaoHorizontal=0;
    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}
```

Inicialização do Rigidbody2D

```
public class Personagem : MonoBehaviour {  
  
    //A propriedade personagemRigidbody2D faz  
    //referência ao Rigidbody2D que foi adicionado ao personagem  
    public Rigidbody2D personagemRigidbody2D;  
    public float direcaoHorizontal=0;  
    // Use this for initialization  
    void Start () {  
        //Inicializa o Rigidbody2D que foi acrescentado ao personagem.  
        this.personagemRigidbody2D = gameObject.GetComponent<Rigidbody2D>();  
    }  
  
    // Update is called once per frame  
    void Update () {  
    }  
}
```

Criar movimento horizontal

Determina a intensidade do movimento

Recebe até 1 para direita e até -1 para esquerda. Quando parado Recebe 0.

```
// Update is called once per frame
void Update () {
    this.direcaoHorizontal = Input.GetAxis("Horizontal");
    float intensidade = 5.0f;
    float x = this.direcaoHorizontal * intensidade;
    float y = personagemRigidbody2D.velocity.y;
    Vector2 movimento = new Vector2(x,y);
    this.personagemRigidbody2D.velocity = movimento;
}
```

Determina o módulo, sentido e direção do movimento

Rotacionar e tombar.

Para impedir que o personagem rotacione em 360 ou cair no chão de costas. Acrescente as seguintes linhas no método start

```
// Use this for initialization
void Start () {
    //Inicializa o Rigidbody2D que foi acrescentado ao personagem.
    this.personagemRigidbody2D = gameObject.GetComponent<Rigidbody2D>();
    this.personagemRigidbody2D.freezeRotation = true;
}
```



Impede movimento rotacional

Código completo

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Personagem : MonoBehaviour {

    //A propriedade personagemRigidbody2D faz
    //referência ao Rigidbody2D que foi adicionado ao personagem
    public Rigidbody2D personagemRigidbody2D;
    public float direcaoHorizontal=0;
    // Use this for initialization
    void Start () {
        //Inicializa o Rigidbody2D que foi acrescentado ao personagem.
        this.personagemRigidbody2D = gameObject.GetComponent<Rigidbody2D>();
        this.personagemRigidbody2D.freezeRotation = true;
    }

    // Update is called once per frame
    void Update () {
        this.direcaoHorizontal = Input.GetAxis("Horizontal");
        float intensidade = 5.0f;
        float x = this.direcaoHorizontal * intensidade;
        float y = personagemRigidbody2D.velocity.y;
        Vector2 movimento = new Vector2(x,y);
        this.personagemRigidbody2D.velocity = movimento;
    }
}
```

Melhorando o Código

```
public class Personagem : MonoBehaviour {  
    //A propriedade personagemRigidbody2D faz  
    //referência ao Rigidbody2D que foi adicionado ao personagem  
    public Rigidbody2D personagemRigidbody2D;  
    public float direcaoHorizontal=0;  
    // Use this for initialization  
    void Start () {  
        //Inicializa o Rigidbody2D que foi acrescentado ao personagem.  
        this.personagemRigidbody2D = gameObject.GetComponent<Rigidbody2D>();  
        this.personagemRigidbody2D.freezeRotation = true;  
    }  
    // Update is called once per frame  
    void Update () {  
        movimentoHorizontal(5.0f);  
    }  
    void movimentoHorizontal(float intensidade)  
    {  
        this.direcaoHorizontal = Input.GetAxis("Horizontal");  
        float x = this.direcaoHorizontal * intensidade;  
        float y = personagemRigidbody2D.velocity.y;  
        Vector2 movimento = new Vector2(x, y);  
        this.personagemRigidbody2D.velocity = movimento;  
    }  
}
```

Atributos do componente
Que recebe o script

Método que executa
quanto o jogo é iniciado

Método que executa quando
ocorre troca de frame

Método que executa
para movimentar
O personagem

Método recebe
intensidade do
movimento