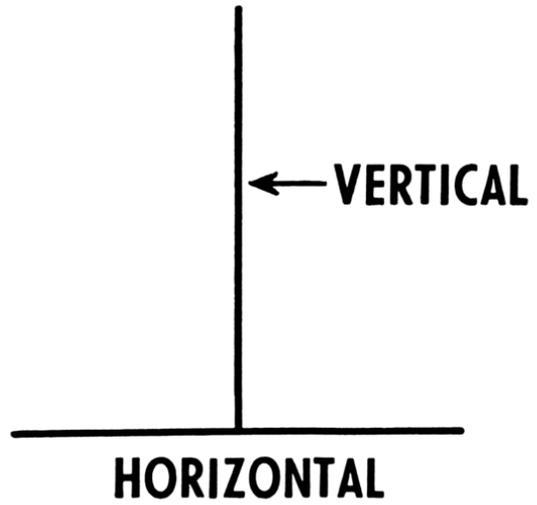


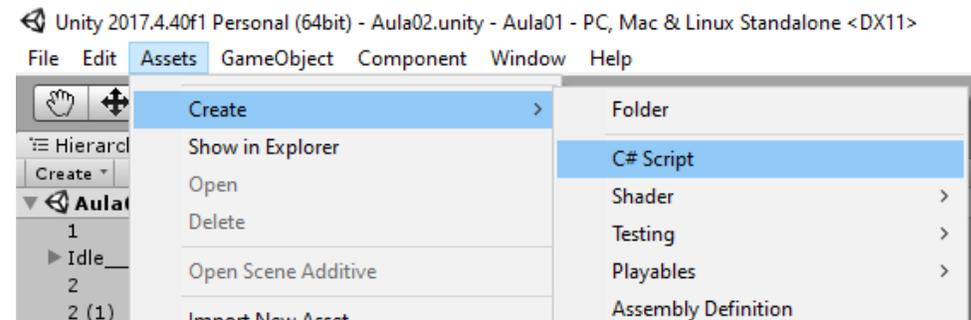
Movimento Vertical

Prof. Me. Hélio Esperidião

Movimento Vertical



Crie um Script



- Será criado um arquivo na sessão de assets do Unity.
- Nomeie o arquivo com as mesmas regras de declaração de variáveis
 - Mais a regra da primeira letra em máscara.
 - Efetue dois cliques no arquivo gerado que o editor de código abrirá o código

Gravity Scale

É uma propriedade que multiplica a gravidade de um corpo rígido por um valor.

O que acontece se a escala de gravidade for igual a 1?

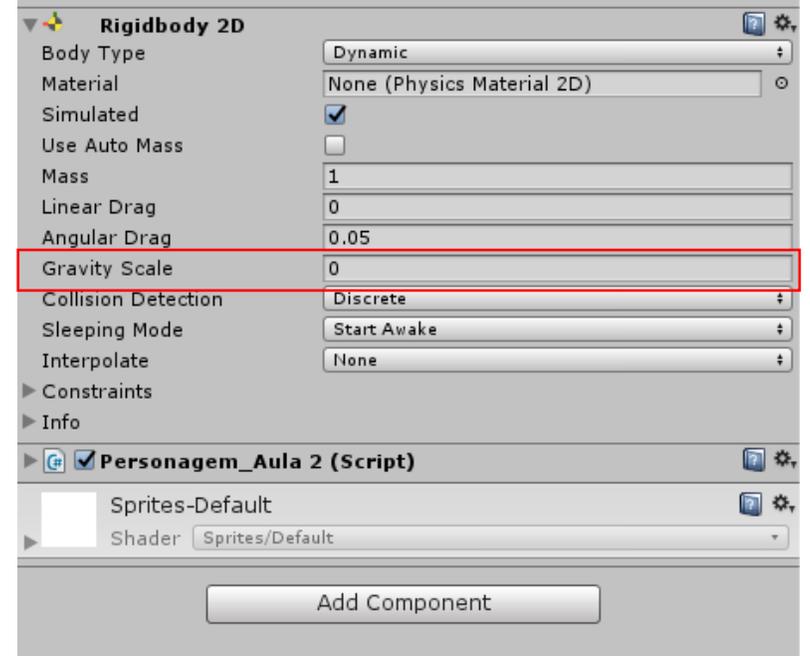
O que acontece se a escala de gravidade for igual a 0?

O que acontece se a escala de gravidade for igual a -1?

O que acontece se a escala de gravidade for igual a 3?

Escala de gravidade

- Configure o personagem como **colisor**.
- Configure o personagem como corpo **rígido**.
- Clique no personagem e procure a propriedade Gravity Scale.
 - Mude o *Gravity Scale* para zero



Atributos do Personagem

```
using System.Collections;
using UnityEngine;
public class Personagem : MonoBehaviour {

    // Variáveis para controlar a velocidade do personagem
    float VelocidadeX; // Velocidade no eixo X (horizontal)
    float VelocidadeY; // Velocidade no eixo Y (vertical)
    float VelocidadeHorizontalMaxima; // Velocidade máxima que o personagem pode atingir no eixo X
    float DirecaoHorizontal; // Direção do movimento horizontal (-1 para esquerda, 1 para direita, 0 para parado)

    float VelocidadeVerticalMaxima; // Velocidade máxima que o personagem pode atingir no eixo Y
    float DirecaoVertical; // Direção do movimento vertical (-1 para baixo, 1 para cima, 0 para parado)

    Vector2 VetorVelocidadePersonagem; // Vetor que armazena a velocidade do personagem em X e Y

    // Referência ao componente Rigidbody2D do personagem
    Rigidbody2D CorpoRigidoPersonagem;

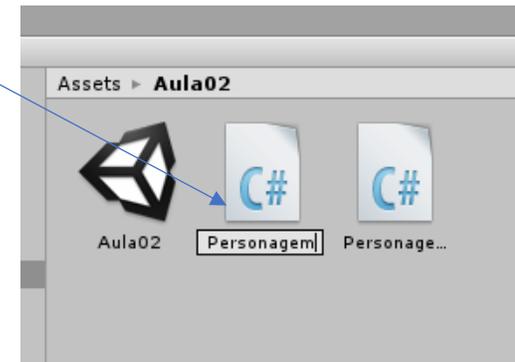
    void Start () {

    }

    void Update () {

    }

}
```



```

public class Personagem : MonoBehaviour {
    // Variáveis para controlar a velocidade do personagem
    float VelocidadeX; // Velocidade no eixo X (horizontal)
    float VelocidadeY; // Velocidade no eixo Y (vertical)
    float VelocidadeHorizontalMaxima; // Velocidade máxima que o personagem pode atingir no eixo X
    float DirecaoHorizontal; //(-1 para esquerda, 1 para direita, 0 para parado)
    float VelocidadeVerticalMaxima; // Velocidade máxima que o personagem pode atingir no eixo Y
    float DirecaoVertical; // (-1 para baixo, 1 para cima, 0 para parado)
    Vector2 VetorVelocidadePersonagem; // Vetor que armazena a velocidade do personagem em X e Y
    Rigidbody2D CorpoRigidoPersonagem; // Referência ao componente Rigidbody2D do personagem
    void Start () {
        // Obtém o componente Rigidbody2D do objeto ao qual este script está anexado
        CorpoRigidoPersonagem = GetComponent<Rigidbody2D> ();

        CorpoRigidoPersonagem.gravityScale = 0f; // (0 significa que não há gravidade)
        CorpoRigidoPersonagem.freezeRotation = true;

        VelocidadeX = 0f; // Começa parado no eixo X
        VelocidadeY = 0f; // Começa parado no eixo Y

        VelocidadeHorizontalMaxima = 5f; // Define a velocidade máxima no eixo X
        DirecaoHorizontal = 0f; // Começa sem direção definida

        VelocidadeVerticalMaxima = 5f; // Define a velocidade máxima no eixo Y
        DirecaoVertical = 0f; // Começa sem direção definida

        // Cria um vetor de velocidade inicial para o personagem
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);
    }
}

```

void Start ()

- Observe que a propriedade **gravityScale** é modificada para 0, isso impede que o personagem sofra ação da gravidade.
- Isso permite que ele se movimente para cima ou para baixo.

void MovimentoVertical()

```
void MovimentoVertical(){  
    // Obtém a direção vertical a partir do input do jogador (teclado ou controle)  
    // Input.GetAxis("Vertical") retorna um valor entre -1 e 1:  
    // -1 para baixo, 1 para cima, 0 se nenhuma tecla estiver sendo pressionada  
    DirecaoVertical = Input.GetAxis ("Vertical");  
  
    // Mantém a velocidade atual no eixo X (para não interferir no movimento horizontal)  
    VelocidadeX = CorpoRigidoPersonagem.velocity.x;  
  
    // Calcula a velocidade no eixo Y multiplicando a direção pela velocidade máxima  
    VelocidadeY = VelocidadeVerticalMaxima * DirecaoVertical;  
  
    // Cria um novo vetor de velocidade com os valores calculados  
    VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);  
  
    // Aplica o vetor de velocidade ao Rigidbody2D para mover o personagem  
    CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;  
}
```

```

using UnityEngine;
public class PersonagemAula2 : MonoBehaviour {
    float VelocidadeX; float VelocidadeY; float VelocidadeHorizontalMaxima;
    float DirecaoHorizontal; float VelocidadeVerticalMaxima;
    float DirecaoVertical; Vector2 VetorVelocidadePersonagem;
    Rigidbody2D CorpoRigidoPersonagem;
    void Start () {
        CorpoRigidoPersonagem = GetComponent<Rigidbody2D> ();
        CorpoRigidoPersonagem.gravityScale = 0f;
        CorpoRigidoPersonagem.freezeRotation = true;
        VelocidadeX = 0f;
        VelocidadeY = 0f;
        VelocidadeHorizontalMaxima = 5f;
        DirecaoHorizontal = 0f;
        VelocidadeVerticalMaxima = 5f;
        DirecaoVertical = 0f;
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);
    }
    void Update () {
        MovimentoHorizontal ();
        MovimentoVertical ();
    }
    void MovimentoVertical(){
        DirecaoVertical = Input.GetAxis ("Vertical");
        VelocidadeX = CorpoRigidoPersonagem.velocity.x;
        VelocidadeY = VelocidadeVerticalMaxima * DirecaoVertical;
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);
        CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;
    }
    void MovimentoHorizontal(){
        DirecaoHorizontal = Input.GetAxis ("Horizontal");
        VelocidadeX = VelocidadeHorizontalMaxima * DirecaoHorizontal;
        VelocidadeY = CorpoRigidoPersonagem.velocity.y;
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);
        CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;
    }
}

```

Código completo

Dentro do update antes de cada frame é chamado o “bloco”

MovimentoHorizontal() e o “bloco”
MovimentoVertical()