

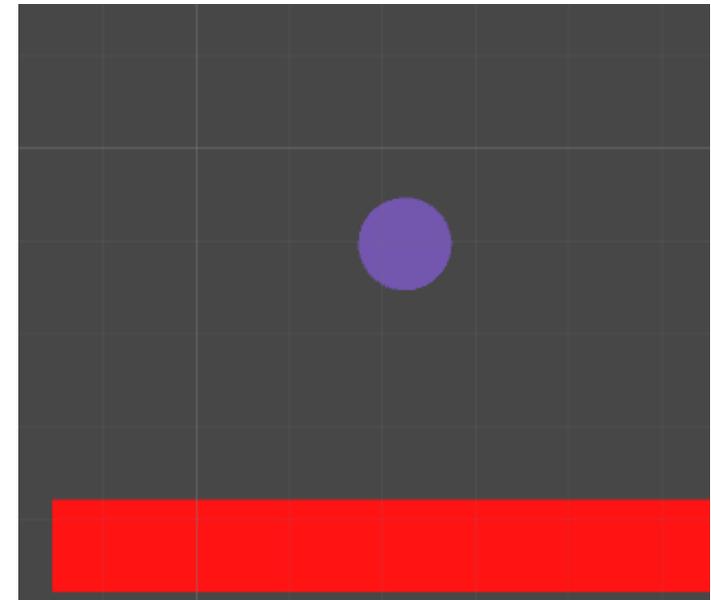


# Pulo Simples – Unity3d

Prof. Me. Hélio Esperidião

# Configuração inicial

- **Circulo:**
  - Component >> Physic2D>>CircleCollider 2D
  - Component >> Physic2D>>RigidBody 2D
- **Chao:**
  - Component >> Physic2D>>Box Collider 2D



# Variáveis e inicialização

```
public class PersonagemAula03 : MonoBehaviour {  
    bool ApertouBotaoPular; // para saber se o botão de pular foi apertado  
    float VelocidadePuloSimples; // variável que guarda a velocidade de pulo 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;  
    float VelocidadeVerticalMaxima; //Variável que armazena a velocidade Maxima Vertical do personagem  
    float DirecaoVertical; // Variável que armazena a direção vertical do personamagem  
    Vector2 VetorVelocidadePersonagem; // Vetor que armazena a velocidade do personagem em X e Y  
    Rigidbody2D CorpoRigidoPersonagem; // Referência ao componente Rigidbody2D do personagem  
    void Start () { // Método chamado uma vez no início do jogo  
        ApertouBotaoPular = false; // inicia a variável com false;  
        VelocidadePuloSimples = 10; // determina o valor da velocidade do Pulo;  
        CorpoRigidoPersonagem = GetComponent<Rigidbody2D> (); // Obtém Rigidbody2D  
        CorpoRigidoPersonagem.gravityScale = 3.0f; // Define a escala da gravidade para o Rigidbody2D  
        CorpoRigidoPersonagem.freezeRotation = true; // Congela a rotação do Rigidbody2D  
        VelocidadeX = 0f; // Começa parado no eixo X  
        VelocidadeY = 0f; // Começa parado no eixo Y  
        VelocidadeHorizontalMaxima = 5.0f; // Define a velocidade máxima no eixo X  
        DirecaoHorizontal = 0f; // Começa sem direção definida  
        VelocidadeVerticalMaxima = 5.0f; // Define a velocidade maxima no eixo y  
  
        VelocidadeVerDirecaoHorizontalMaxima = 5.0f; // Define a velocidade maxima no eixo y  
        DirecaoVertical = 0f; // Define a direção do movimento na vertical  
        // Cria um vetor de velocidade inicial para o personagem  
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);  
        // Aplica o vetor de velocidade ao Rigidbody2D para mover o personagem  
        CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;  
    }  
}
```

```
void Update () {  
    MovimentoHorizontal ();  
    MovimentoPuloSimples();  
}
```

# Método MovimentoPuloSimples()

```
void MovimentoPuloSimples(){  
    // Verifica se o botão de pular foi pressionado.  
    // "Jump" é o nome do botão configurado no Input Manager do Unity.  
    ApertouBotaoPular = Input.GetButtonDown ("Jump");  
  
    // Verifica se o botão de pular foi pressionado.  
    if (ApertouBotaoPular == true) {  
        // Armazena a velocidade atual no eixo X do personagem.  
        // CorpoRigidoPersonagem.velocity.x retorna a velocidade horizontal do Rigidbody2D.  
        VelocidadeX = CorpoRigidoPersonagem.velocity.x;  
  
        // Define a velocidade no eixo Y para o valor de VelocidadePuloSimples.  
        // VelocidadePuloSimples é uma variável que define a força do pulo.  
        VelocidadeY = VelocidadePuloSimples;  
  
        // Cria um novo vetor de velocidade com a velocidade X atual e a nova velocidade Y.  
        // Isso mantém a velocidade horizontal e aplica a força do pulo no eixo Y.  
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);  
  
        // Aplica o novo vetor de velocidade ao Rigidbody2D do personagem.  
        // Isso faz o personagem pular, mantendo sua velocidade horizontal.  
        CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;  
    }  
}
```

Qual o  
problema?

Qual o problema de lógica que  
pode ser visualizado no código  
apresentado?

# Problemas com o código

Observe que não há limites de quantas vezes o jogador pode pressionar o botão de espaço.

Se o jogador pressionar muitas vezes ele terá a impressão de que está voando.

Precisamos lidar com esse comportamento por meio de programação.

# Variáveis e inicialização

```
public class PersonagemAula03 : MonoBehaviour {
    bool ApertouBotaoPular; // para saber se o botão de pular foi apertado
    float VelocidadePuloSimples; // variavel que guarda a velocidade de pulo do personagem
    Collider2D Colisor2dPersonagem; //variável que armazena o colisor do personagem
    bool EstaTocandoAlgumColisor; //Variável que verificar se o personagem está tocando um colisor
    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; //Variável que armazena a velocidade Maxima Vertical do personagem
    float DirecaoVertical; // Variável que armazena a direção vertical do personamagem
    Vector2 VetorVelocidadePersonagem; // Vetor que armazena a velocidade do personagem em X e Y
    Rigidbody2D CorpoRigidoPersonagem; // Referência ao componente Rigidbody2D do personagem
    // Método chamado uma vez no início do jogo
    void Start () {
        ApertouBotaoPular = false; // inicia a variavel com false;
        EstaTocandoAlgumColisor = false; // inicializa a variavel como falso;
        VelocidadePuloSimples = 10; // determina o valor da velocidade do Pulo;
        CorpoRigidoPersonagem = GetComponent<Rigidbody2D> (); // Obtém o componente Rigidbody2D
        Colisor2dPersonagem = GetComponent<Collider2D> (); // Obtpem o componente Collider2D do objeto que possui o script
        CorpoRigidoPersonagem.gravityScale = 3.0f; // Define a escala da gravidade para o Rigidbody2D
        // Congela a rotação do Rigidbody2D para evitar que o personagem gire ao colidir com algo
        CorpoRigidoPersonagem.freezeRotation = true;
        // Inicializa as variáveis de velocidade
        VelocidadeX = 0f; // Começa parado no eixo X
        VelocidadeY = 0f; // Começa parado no eixo Y
        VelocidadeHorizontalMaxima = 5.0f; // Define a velocidade máxima no eixo X
        DirecaoHorizontal = 0f; // Começa sem direção definida
        VelocidadeVerticalMaxima = 5.0f; // Define a velocidade maxima no eixo y
        DirecaoVertical = 0f; // Define a direção do movimento na vertical
        // Cria um vetor de velocidade inicial para o personagem
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);
        // Aplica o vetor de velocidade ao Rigidbody2D para mover o personagem
        CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;
    }
}
```

# Update ()

```
void Update () {  
    MovimentoHorizontal ();  
    MovimentoPuloUnico();  
    //MovimentoVertical ();  
    //MovimentoPuloSimples();  
}
```

# Modificação do código

```
void MovimentoPuloUnico(){
    // Verifica se o botão de pular foi pressionado.
    // "Jump" é o nome do botão configurado no Input Manager do Unity.
    ApertouBotaoPular = Input.GetButtonDown ("Jump");
    // Verifica se o personagem está tocando algum colisor.
    // Colisor2dPersonagem é uma referência ao Collider2D do personagem.
    // IsTouchingLayers() retorna true se o colisor estiver tocando qualquer camada.
    EstaTocandoAlgumColisor = Colisor2dPersonagem.IsTouchingLayers ();
    // Verifica se o botão de pular foi pressionado e se o personagem está tocando algum colisor.
    if (ApertouBotaoPular == true && EstaTocandoAlgumColisor == true) {
        // Armazena a velocidade atual no eixo X do personagem.
        VelocidadeX = CorpoRigidoPersonagem.velocity.x;
        // Define a velocidade no eixo Y para o valor de VelocidadePuloSimples.
        // VelocidadePuloSimples é provavelmente uma variável que define a força do pulo.
        VelocidadeY = VelocidadePuloSimples;
        // Cria um novo vetor de velocidade com a velocidade X atual e a nova velocidade Y.
        VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);
        // Aplica o novo vetor de velocidade ao Rigidbody2D do personagem.
        // Isso faz o personagem pular.
        CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;
    }
}
```

# Código completo

- <https://youtu.be/6oluCL5yMBY>
- <https://github.com/helioesperidiao/>
- <https://github.com/helioesperidiao/Desenvolvimento-de-jogos/>
- <https://github.com/helioesperidiao/Desenvolvimento-de-jogos/blob/main/Assets/Bimestre1/Aula03/PersonagemAula03.cs>