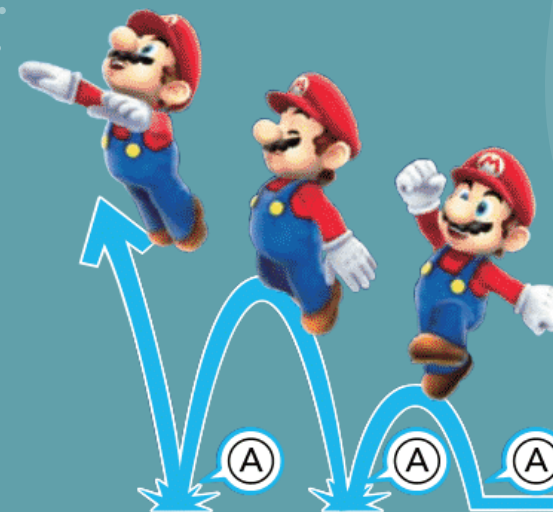


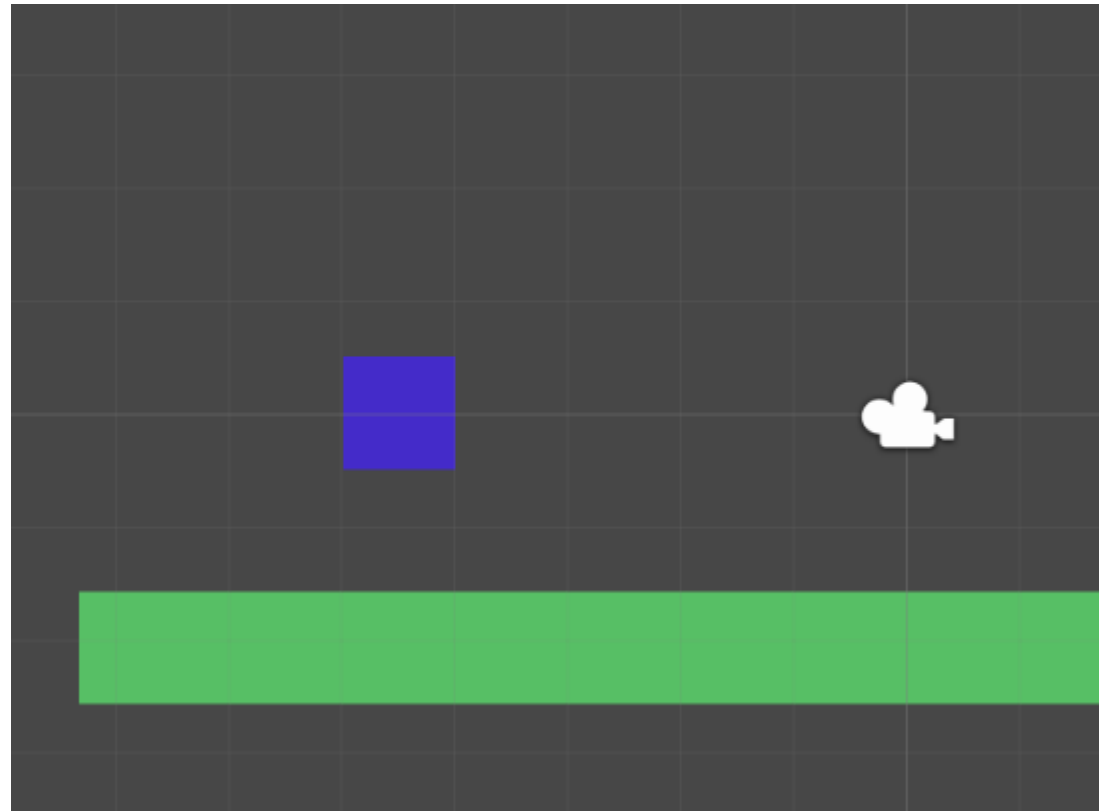
Double jump – Pulo Duplo

Prof. Me. Hélio Esperidião



Configuração Inicial

- Quadrado (Personagem):
 - Corpo Rígido
 - Colisor
- Chão
 - Colisor
 - Tag: chao



Exemplo 01: Pulo Duplo

Variáveis/atributos

```
public class Temp : MonoBehaviour {  
    int ContadorPulos;  
    bool PegouChave01;  
  
    SpriteRenderer Renderer;  
    string TagObjetoTocado;  
  
    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  
    // 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; //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 () {
```

```
    ContadorPulos = 0;  
    PegouChave01 = false;
```

```
    Renderer = GetComponent<SpriteRenderer> ();
```

```
    TagObjetoTocado = "";
```

```
    ApertouBotaoPular = false; // inicia a variavel com false;  
    EstaTocandoAlgumColisor = false; // inicializa a variavel como falso;
```

```
    VelocidadePuloSimples = 10.0f; // determina o valor da velocidade do Pulo;
```

```
    // Obtém o componente Rigidbody2D do objeto ao qual este script está anexado  
    CorpoRigidoPersonagem = GetComponent<Rigidbody2D> ();
```

```
    Colisor2dPersonagem = GetComponent<Collider2D> ();
```

```
    // Define a escala da gravidade para o Rigidbody2D (1 é o valor padrão)  
    CorpoRigidoPersonagem.gravityScale = 3.0f;
```

```
    // 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;
```

```
}
```

start

Update ()

// Método chamado uma vez por frame (quadro)

```
void Update () {
```

```
    MovimentoHorizontalFlip ();
```

```
    MovimentoPuloDuplo ();
```

```
    //MovimentoPuloUnico();
```

```
    //MovimentoHorizontal ();
```

```
    //MovimentoVertical ();
```

```
    //MovimentoPuloSimples();
```

```
}
```

PuloDuplo()

```
void MovimentoPuloDuplo(){  
    // 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 && ContadorPulos < 2 ) {  
        ContadorPulos = ContadorPulos+1;  
        // 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;  
    }  
}
```

Se Entrou no if
é somando + 1 em ContadorPulos

Por que contador de pulos recebe zero?

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;  
  
    if (TagObjetoTocado == "chao") {  
        ContadorPulos = 0;  
    }  
    //..  
}
```

Qual a diferença do anterior?

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;  
  
    ContadorPulos = 0;  
    //..  
}
```

Qual a diferença?

```
void OnCollisionEnter2D(Collision2D objetoTocado){  
    string tagTocada = objetoTocado.gameObject.tag;  
    ContadorPulos = 0;  
}
```

Neste exemplo o ContadorPulos é zerado toda vez que o OnCollisionEnter2D() é disparado, ou seja se o personagem tocar em qualquer colisor o contador de pulos é zerado.

Contains

//Contains verifica se um texto armazenado em uma string contém uma determinado texto.

//no exemplo abaixo verifica se a variavel TagObjetoTocado contem o texto "chao"

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;  
    if (TagObjetoTocado.Contains ("chao")) {  
        ContadorPulos = 0;  
    }  
}
```

Conteúdos de TagObjetoTocado que entram no if:
"arvore chao"
"pedra chao"
"chao bonus1"

Exemplo 2 – Múltiplos Pulos

- Vamos imaginar que inicialmente o jogador não é capaz de efetuar pulos.
- Ao coletar um o item “puloSimples” é habilitado o pulo simples.
- Ao coletar o item “puloDuplo” é habilidade o pulo duplo.

Atributos

```
public class Personagem02x03PuloMultiplo : MonoBehaviour {  
  
    int TotalPulos; //determina o limite de pulos do personagem  
    int ContadorPulos; // Conta quantos pulos o personagem pode efetuar  
  
    SpriteRenderer Rerender; //Contem a imagem do asset, utilizado para fazer o flip  
  
    string TagObjetoTocado;  
  
    bool PegouChave01;  
  
    bool ApertouBotaoPular; // para saber se o botão de pular foi apertado  
    float VelocidadePuloSimples; // varivael que guarda a velocidade de pulo do personagem  
    Collider2D Colisor2dPersonagem; //variável que armazena o colisor do personagem  
    bool EstaTocandoAlgunColisor; //Variável que verificar se o personagem está tocando um colisor  
  
    // 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; //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  
    // Referência ao componente Rigidbody2D do personagem  
    Rigidbody2D CorpoRigidoPersonagem; body2D CorpoRigidoPersonagem;
```

```

void Start () {
    //Inicialmente o personagem não pode pular pq o total de pulos é igual a zero.
    TotalPulos = 0; //determina o limite de pulos do personagem
    ContadorPulos = 0; // Conta quantos pulos o personagem pode efetuar

    Render = GetComponent<SpriteRenderer> ();

    PegouChave01 = false;
    TagObjetoTocado = "";

    ApertouBotaoPular = false; // inicia a variavel com false;
    EstaTocandoAlgoColisor = false; // inicializa a variavel como falso;

    VelocidadePuloSimples = 10.0f; // determina o valor da velocidade do Pulo;
    // Obtém o componente Rigidbody2D do objeto ao qual este script está anexado
    CorpoRigidoPersonagem = GetComponent<Rigidbody2D> ();

    Colisor2dPersonagem = GetComponent<Collider2D> ();

    // Define a escala da gravidade para o Rigidbody2D (1 é o valor padrão)
    CorpoRigidoPersonagem.gravityScale = 3.0f;

    // 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;
}

```

Start()

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

update

```
void MovimentoPuloMultiplo(){
    // 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 e se o personagem está tocando algum colisor.
    if (ApertouBotaoPular == true && ContadorPulos < TotalPulos) {

        ContadorPulos = ContadorPulos + 1; //ContadorPulos++;

        // 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;
    }
}
```

Para que seja realizado o pulo é necessário que o personagem tenha tocado no objeto

“bonusPuloUnico” ou no objeto “bonusPuloDuplo”

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;  
  
    if (TagObjetoTocado == "bonusPuloUnico") {  
        TotalPulos = 1;  
        Destroy (objetoTocado.gameObject);  
    }  
    if (TagObjetoTocado == "bonusPuloDuplo") {  
        TotalPulos = 2;  
        Destroy (objetoTocado.gameObject);  
    }  
}
```

Exemplo 3 – Remover Atritos das Paredes

- Neste exemplo será impedido que o personagem seja capaz de escalar paredes.
- Será criado um material que não possui atrito e caso o personagem toque na tag “semAtrito” o elemento tocado terá seu material modificado para atrito zero.

Atributos

```
public class Personagem02x03SemAtrito : MonoBehaviour {
    PhysicsMaterial2D MaterialSemAtrito;

    int TotalPulos; //determina o limite de pulos do personagem
    int ContadorPulos; // Conta quantos pulos o personagem pode efetuar

    SpriteRenderer Rerender; //Contem a imagem do asset, utilizado para fazer o flip

    string TagObjetoTocado;

    bool PegouChave01;

    bool ApertouBotaoPular; // para saber se o botão de pular foi apertado
    float VelocidadePuloSimples; // varivael 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

    // 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; //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
    // Referência ao componente Rigidbody2D do personagem
    Rigidbody2D CorpoRigidoPersonagem;
```

// Método chamado uma vez no início do jogo

```
void Start () {  
    // Cria um novo material e armazena na variavel MaterialSemAtrito  
    MaterialSemAtrito = new PhysicsMaterial2D();  
    MaterialSemAtrito.friction = 0f; // esse material possui atrito zero;  
    //Inicialmente o personagem não pode pular pq o total de pulos é igual a zero.  
    TotalPulos = 0; //determina o limite de pulos do personagem  
    ContadorPulos = 0; // Conta quantos pulos o personagem pode efetuar  
    Rerender = GetComponent<SpriteRenderer> (); // incializa a variável  
    PegouChave01 = false;  
    TagObjetoTocado = "";  
    ApertouBotaoPular = false; // inicia a variavel com false;  
    EstaTocandoAlgumColisor = false; // inicializa a variavel como falso;  
    VelocidadePuloSimples = 10.0f; // determina o valor da velocidade do Pulo;  
    // Obtém o componente Rigidbody2D do objeto ao qual este script está anexado  
    CorpoRigidoPersonagem = GetComponent<Rigidbody2D> ();  
    Colisor2dPersonagem = GetComponent<Collider2D> ();  
    // Define a escala da gravidade para o Rigidbody2D (1 é o valor padrão)  
    CorpoRigidoPersonagem.gravityScale = 3.0f;  
    CorpoRigidoPersonagem.freezeRotation = true; // Congela a rotação do Rigidbody2D  
    // 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;  
}
```

Start()

Caso o personagem toque em um colisor que contenha a tag “semAtrito” o elemento tocado recebe um material que não possui atrito. Ps. A configuração do Material foi realizada no **Start()**

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;  
  
    if (TagObjetoTocado.Contains("semAtrito")) {  
        print ("Tocou TAG: semAtrito");  
        objetoTocado.collider.sharedMaterial = MaterialSemAtrito;  
    }  
}
```