

INPUTS

PROF. ME. HÉLIO ESPERIDIÃO

```
public class Personagem : MonoBehaviour {
    // recebe corpo rígido do personagem (Componentes>Física 2d > Corpo Rígido)
    public Rigidbody2D personagemRigidbody2D;
    // Recebe zero quando está parado até -1 quando move para esquerda
    //e até +1 quando move para direita.
    public float direcaoHorizontal=0;
    // Conta as caixas que foram coletadas
    public int contadorCaixas = 0;
    // determina a intensidade do pulo
    public float intensidadePulo = 5f;
    // determina a intensidade de caminhar.
    public float intensidadeMovimentoHorizontal = 5f;
    //recebe o SpriteRenderer do elemento de jogo ou do personagem
    public SpriteRenderer personagemSpriteRenderer;
    //Conta quantos pulos foram realizados.
    public int contadorPulos=0;
```

Entradas do mouse.

Criaremos o método EntradasMouse para tratar a interação com o mouse.

```
void Update() {  
    //chama o método para movimentação horizontal;  
    MovimentoHorizontal();  
    //chama o método para movimento do pulo  
    //MovimentoPulo();  
    // Movimento();  
    EntradasMouse();  
}
```

```
void EntradasMouse()
{
    if (Input.GetMouseButtonDown(1))
    {
        print("Apertou o botão direito");
    }
    if (Input.GetMouseButtonDown(0))
    {
        print("Apertou o botão esquerdo");
    }
}
```

Retorna Verdadeiro se o usuário clicou com o botão direito do mouse.

Retorna Verdadeiro se o usuário clicou com o botão esquerdo do mouse.

EXEMPLO PRÁTICO: PULAR COM O MOUSE.

```
void EntradasMouse() {  
    if (Input.GetMouseButtonDown(1))  
    {  
        print("Apertou o botão direito");  
    }  
    if (Input.GetMouseButtonDown(0))  
    {  
        float x = personagemRigidbody2D.velocity.x;  
        //"variável global com a intensidade do pulo"  
        float y = this.intensidadePulo;  
        //verifica se o contador de pulos é menor que dois.  
        if (contadorPulos < 2) {  
            // Crie um vetor de velocidade com componentes x,y do plano.  
            Vector2 movimento = new Vector2(x, y);  
            //Adiciona o vetor de movimento ao corpo rígido do personagem.  
            this.personagemRigidbody2D.velocity = movimento;  
            //Conta mais um na variável que conta os pulos.  
            this.contadorPulos++;  
        }  
    }  
}
```

Retorna Verdadeiro se o usuário clicou com o botão esquerdo do mouse.

Executa o código necessário para o pulo

Utilização das teclas do Teclado

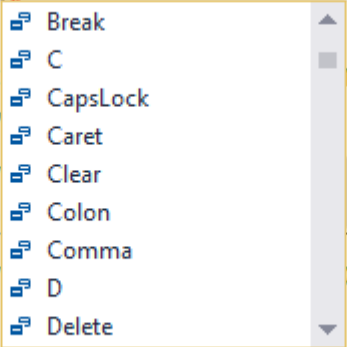
```
void MovimentoPulo(){  
    //recupera a velocidade em x no momento do pulo  
    float x = personagemRigidbody2D.velocity.x;  
    //"variável global com a intensidade do pulo"  
    float y = this.intensidadePulo;  
    //verifica se a tecla de espaço foi pressionada  
    //e se o contador de pulos é menor que dois.  
    if (Input.GetKeyDown(KeyCode.K) && contadorPulos <2)  
    {  
        // Crie um vetor de velocidade com componentes x,y do plano.  
        Vector2 movimento = new Vector2(x, y);  
        //Adiciona o vetor de movimento ao corpo rígido do personagem.  
        this.personagemRigidbody2D.velocity = movimento;  
        //Conta mais um na variável que conta os pulos.  
        this.contadorPulos++;  
    }  
}
```

O exemplo apresenta a execução do pulo quando o usuário pressionar a tecla K no teclado

KeyCode possui a codificação de todas as teclas do teclado, com ele também é possível configurar gamepads e configurações mais avançadas do mouse.

Veja as opções quando se está usando o keyCode. Verifique que é possível utilizar a tecla C, D, Delete, etc.

```
if (Input.GetKeyDown(KeyCode.) && contadorPulos < 2)
{
    // Crie um vetor de veloc
    Vector2 movimento = new V
    //Adiciona o vetor de mov
    this.personagemRigidbody2
    //Conta mais um na variáv
    this.contadorPulos++;
}
```



- Break
- C
- CapsLock
- Caret
- Clear
- Colon
- Comma
- D
- Delete

KeyCode.K = 107
'k' key.

Verifique todas as opções disponíveis para o **KeyCode**

<https://docs.unity3d.com/ScriptReference/KeyCode.html>

Question	Question mark '?' key.
At	At key '@'.
LeftBracket	Left square bracket key '['.
Backslash	Backslash key '\'.
RightBracket	Right square bracket key ']'. Right square bracket key ']'.
Caret	Caret key '^'.
Underscore	Underscore '_' key.
BackQuote	Back quote key '`'.
A	'a' key.
B	'b' key.
C	'c' key.
D	'd' key.
E	'e' key.
F	'f' key.
G	'g' key.
H	'h' key.
I	'i' key.
J	'j' key.
K	'k' key.
L	'l' key.
M	'm' key.
N	'n' key.
O	'o' key.