



# ENTRADAS DE DADOS EM JOGOS

Prof. Me. Hélio Esperidião



# Entrada de dados

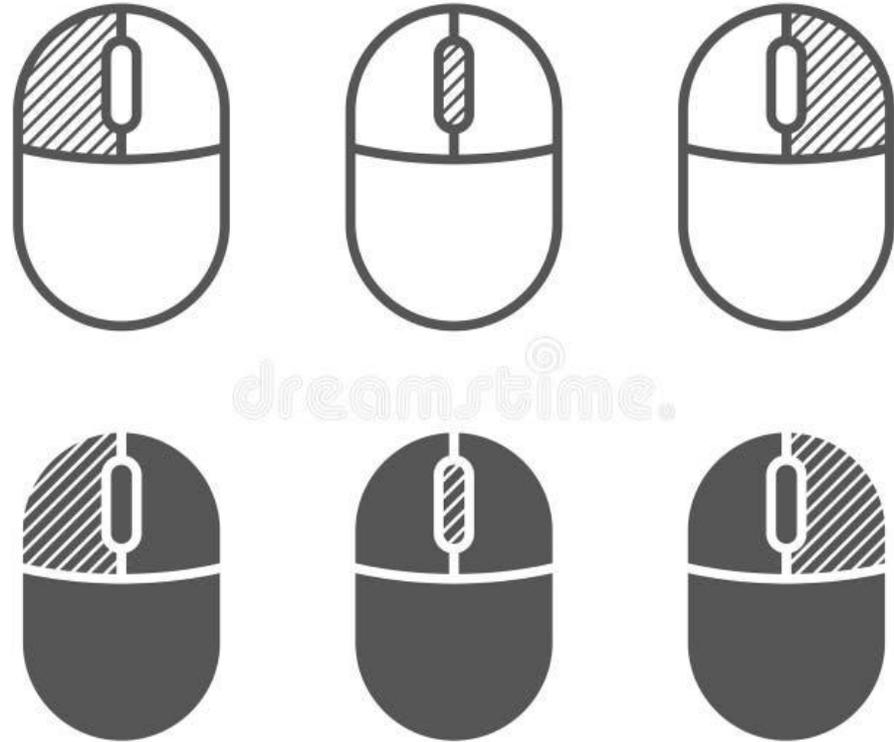
- Para que o usuário interaja com os jogos, este precisa ler dados por meio de periféricos como teclado, mouse, controles e etc,



# input

- O input no Unity é responsável por todas as entradas de dados:
- Teclado
- Mouse
- Controles
- Touch

Input -  
mouse



# mouse

```
void TesteMouse(){
    if (Input.GetMouseButtonDown (0) == true) {
        print ("Apertou o botão esquerdo");
    }
    if (Input.GetMouseButtonDown (1) == true) {
        print ("Apertou o botão direito");
    }
    if (Input.GetMouseButtonDown (2) == true) {
        print ("Apertou o botão do meio");
    }
}
```

# Pular com botão do meio do mouse

## void PuloDuplo()

```
void PuloDuplo(){
    bool apertou = Input.GetMouseButtonDown (2);
    // esse bloco só é executado se for apertado o botão de Jump
    // e se o personagem ContadorPulos < 2
    if (apertou == true && ContadorPulos < TotalPulos ) {
        ContadorPulos = ContadorPulos+1;
        Vx = CorpoRigido.velocity.x;
        Vy = VelocidadePular;
        Vector2 pulo = new Vector2 (Vx, Vy);
        CorpoRigido.velocity = pulo;
    }
}
```

# Movimentar com o click do mouse

## void MovimentoHorizontalFlip()

```
void MovimentoHorizontalFlip(){
    if (Input.GetMouseButtonDown (0) == true) {
        DirecaoHorizontal = -1;
    }
    if (Input.GetMouseButtonDown (1) == true) {
        DirecaoHorizontal = 1;
    }
    Vx = VelocidadeAndar * DirecaoHorizontal;
    Vy = CorpoRigido.velocity.y;

    Vector2 andar = new Vector2 (Vx, Vy);
    CorpoRigido.velocity = andar;
    if (DirecaoHorizontal < 0) {
        Renderer.flipX = true;
    }else if(DirecaoHorizontal > 0){
        Renderer.flipX = false;
    }
}
```

# OnTriggerEnter2D/ OnCollisionEnter2D

```
void OnTriggerEnter2D(Collider2D objetoTriggerTocado){  
    string tagTocadaTrigger = objetoTriggerTocado.gameObject.tag;  
}  
void OnCollisionEnter2D(Collision2D objetoTocado)  
{  
    ContadorPulos = 0;  
    string tagObjetoTocado = objetoTocado.gameObject.tag;  
}
```

# Diferença entre `KeyDown`, `GetKey` e `KeyUp`

Método	Quando retorna true	Uso típico
<code>Input.GetKeyDown</code>	No frame em que a tecla foi pressionada	Ações únicas, como pular
<code>Input.GetKey</code>	Enquanto a tecla estiver sendo mantida	Sustentar movimento, correr, etc.
<code>Input.GetKeyUp</code>	No frame em que a tecla foi solta	Soltar um poder, encerrar ação
<code>Input.GetButtonDown</code>	Retorna true <b>somente no frame em que o botão é pressionado</b>	Comum em jogos de plataforma, tiro ou interação com objetos.

# `Input.GetKeyDown(KeyCode);`

- `GetKeyDown` é um método da classe `Input` no Unity, usado para detectar quando uma tecla específica é pressionada no exato quadro (frame) em que o evento ocorre.
- É comumente usado em jogos para capturar ações do jogador, como pular, atirar ou interagir com objetos.

# Sintaxe

```
if (Input.GetKeyDown(KeyCode.Space)){  
    print("A tecla Espaço foi pressionada!");  
}
```

# Outras variações de uso do input

## use o nome das teclas

```
void PuloDuplo(){  
    bool apertou = Input.GetKeyDown(KeyCode.Space);  
    if (apertou == true && ContadorPulos < TotalPulos ) {  
        ContadorPulos = ContadorPulos+1;  
        Vx = CorpoRigido.velocity.x;  
        Vy = VelocidadePular;  
        Vector2 pulo = new Vector2 (Vx, Vy);  
        CorpoRigido.velocity = pulo;  
    }  
}
```

# Pular com o botão control esquerdo

```
void PuloDuplo(){
    bool apertou = Input.GetKeyDown(KeyCode.LeftControl);
    if (apertou == true && ContadorPulos < TotalPulos ) {
        ContadorPulos = ContadorPulos+1;
        Vx = CorpoRigido.velocity.x;
        Vy = VelocidadePular;
        Vector2 pulo = new Vector2 (Vx, Vy);
        CorpoRigido.velocity = pulo;
    }
}
```

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

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

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

# GetKey()

- **GetKey** é um método da classe Input no Unity usado para detectar se uma tecla está sendo mantida pressionada.
- Diferente de **GetKeyDown**, que só detecta o instante em que a tecla foi pressionada, **GetKey** retorna true em todos os frames enquanto a tecla estiver pressionada.

# Sintaxe

```
if (Input.GetKey(KeyCode.W))  
{  
    print("A tecla W está sendo mantida pressionada.");  
}
```

# Player 1

```
void MovimentoHorizontalPlayer1(){  
  
    if (Input.GetKey (KeyCode.RightArrow)) {  
        DirecaoHorizontal = 1;  
    } else if (Input.GetKey (KeyCode.LeftArrow)) {  
        DirecaoHorizontal = -1;  
    } else {  
        DirecaoHorizontal = 0;  
    }  
  
    VelocidadeX = VelocidadeHorizontalMaxima * DirecaoHorizontal;  
  
    VelocidadeY = CorpoRigidoPersonagem.velocity.y;  
  
    VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);  
  
    CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;  
}
```

# Player 2

```
void MovimentoHorizontalPlayer2(){  
  
    if (Input.GetKey (KeyCode.D)) {  
        DirecaoHorizontal = 1;  
    } else if (Input.GetKey (KeyCode.A)) {  
        DirecaoHorizontal = -1;  
    } else {  
        DirecaoHorizontal = 0;  
    }  
    VelocidadeX = VelocidadeHorizontalMaxima * DirecaoHorizontal;  
  
    VelocidadeY = CorpoRigidoPersonagem.velocity.y;  
  
    VetorVelocidadePersonagem = new Vector2 (VelocidadeX, VelocidadeY);  
  
    CorpoRigidoPersonagem.velocity = VetorVelocidadePersonagem;  
}
```

Utilize o GetKey para saber se estão segurando alguma tecla.

```
void PuloDuplo(){
    //utilize GetKey, para saber se está pressionando
    bool apertou1 = Input.GetKey(KeyCode.LeftControl);
    bool apertou2 = Input.GetMouseButtonDown (0);
    if (apertou1 == true ) {
        if (apertou2 == true) {
            if(ContadorPulos < TotalPulos){
                ContadorPulos = ContadorPulos + 1;
                Vx = CorpoRigido.velocity.x;
                Vy = VelocidadePular;
                Vector2 pulo = new Vector2 (Vx, Vy);
                CorpoRigido.velocity = pulo;
            }
        }
    }
}
```

# Outro exemplo de pulo.

`Input.GetKeyDown(KeyCode.Joystick1Button3);` => Botão Y do Controle do Xbox

```
void Pular(){
    ApertouJump = Input.GetButtonDown("Jump") || Input.GetKeyDown(KeyCode.Joystick0Button3);
    if (ApertouJump == true && ContadorPulos < TotalMaximoPulos){
        Vx = CorpoRigido.velocity.x;
        Vy = VelocidadePular * 0.5f;
        VetorVelocidade = new Vector2(Vx, Vy);
        CorpoRigido.velocity = VetorVelocidade;
        ContadorPulos = ContadorPulos + 1;
    }
}
```

# Mapa de botões do xbox

<b>Botão no Xbox</b>	<b>Nome no Unity (Input.GetButton / Input.GetKey)</b>	<b>Eixo / Observações</b>
<b>A</b> (Botão inferior)	joystick button 0	Ação principal, pular, confirmar
<b>B</b> (direita)	joystick button 1	Cancelar, voltar
<b>X</b> (esquerda)	joystick button 2	Usado para interagir
<b>Y</b> (cima)	joystick button 3	Alternar modo, menu, etc.
<b>LB</b> (Left Bumper)	joystick button 4	Ombro esquerdo
<b>RB</b> (Right Bumper)	joystick button 5	Ombro direito
<b>Back / Select</b>	joystick button 6	Menu secundário
<b>Start</b>	joystick button 7	Menu principal
<b>Left Stick (Click)</b>	joystick button 8	Clique do analógico esquerdo
<b>Right Stick (Click)</b>	joystick button 9	Clique do analógico direito
<b>Xbox / Guide Button</b>	Não detectável via Input Manager	Limitado por sistema operacional

# Eixos dos Analógicos e Triggers

<b>Movimento</b>	<b>Nome no Unity (Input.GetAxis)</b>	<b>Observações</b>
Left Stick Horizontal	Horizontal ou X axis	Esquerda (-1) ↔ Direita (+1)
Left Stick Vertical	Vertical ou Y axis	Baixo (-1) ↕ Cima (+1)
Right Stick Horizontal	3rd axis	Varia por plataforma
Right Stick Vertical	4th axis	Pode precisar ajuste nas configurações
LT (Left Trigger)	9th axis (em alguns casos)	Valores entre 0 e 1
RT (Right Trigger)	10th axis	Idem acima
D-Pad Horizontal	6th axis	Esquerda (-1) ↔ Direita (+1)
D-Pad Vertical	7th axis	Baixo (-1) ↕ Cima (+1)

# Exemplo de uso com GetKey ou GetButton

```
if (Input.GetKey("joystick button 0")) {  
    Debug.Log("Botão A pressionado");  
}
```

```
if (Input.GetAxis("Horizontal") > 0.1f) {  
    Debug.Log("Movendo para a direita");  
}
```

# Input.GetButtonDown("nomeDoBotao")

- Input.GetButtonDown(string buttonName) é um método da classe Input do Unity que verifica se um botão foi pressionado naquele frame específico.
- Ele retorna **true** somente uma vez, no frame exato em que o botão é pressionado.
- Nos frames seguintes, mesmo que o botão ainda esteja pressionado, ele retorna **false**

# Sintaxe

```
if (Input.GetButtonDown("Jump")) {  
    print("O botão de pulo foi pressionado!");  
}
```

# GetButtonDown — botão pressionado uma única vez

```
void PuloDuplo() {  
    bool apertou = Input.GetButtonDown("Fire3"); // "Fire3" usa Left Control  
    por padrão  
    if (apertou && ContadorPulos < TotalPulos) {  
        ContadorPulos += 1;  
        Vx = CorpoRigido.velocity.x;  
        Vy = VelocidadePular;  
        Vector2 pulo = new Vector2(Vx, Vy);  
        CorpoRigido.velocity = pulo;  
    }  
}
```

# GetButton — botão sendo pressionado continuamente

```
if (Input.GetButton("Fire1"))  
{  
    Debug.Log("Botão Fire1 está sendo segurado");  
}
```

# Botões padrão do Unity (Input Manager)

<b>Nome no Input Manager</b>	<b>Tipo</b>	<b>Teclas/Controles Padrão</b>	<b>Descrição</b>
<b>Fire1</b>	Botão	Ctrl esquerdo / Mouse 0	Ação principal (atirar, atacar, etc.)
<b>Fire2</b>	Botão	Alt esquerdo / Mouse 1	Ação secundária
<b>Fire3</b>	Botão	Ctrl direito / Mouse 2	Ação terciária
<b>Jump</b>	Botão	Barra de espaço (Space)	Pulo (Jump)
<b>Submit</b>	Botão	Enter / Joystick button 0	Confirmação (menus)
<b>Cancel</b>	Botão	Esc / Joystick button 1	Cancelar / Voltar

# Comparativo entre métodos de entrada no Unity

Quadro (Frame)	Pressionado?	GetButtonDown()	GetButton()	GetButtonUp()	GetKeyDown()	GetKey()	GetKeyUp()
Frame 1	NÃO	false	false	false	false	false	false
Frame 2	SIM	true	true	false	true	true	false
Frame 3	SIM (MANTIDO)	false	true	false	false	true	false
Frame 4	SOLTOU	false	false	true	false	false	true

# Guia definitivo

- Use `GetButtonDown/ Input.GetKeyDown` para ações que devem acontecer uma vez no início do toque, como pular, atirar um tiro, iniciar uma ação.
- Use `GetButton/GetKey` para ações contínuas, como correr, mover o personagem enquanto segura a tecla.
- Use `GetButtonUp` para detectar quando o jogador para de pressionar um botão, útil para liberar controles, parar ataques ou finalizar ações.

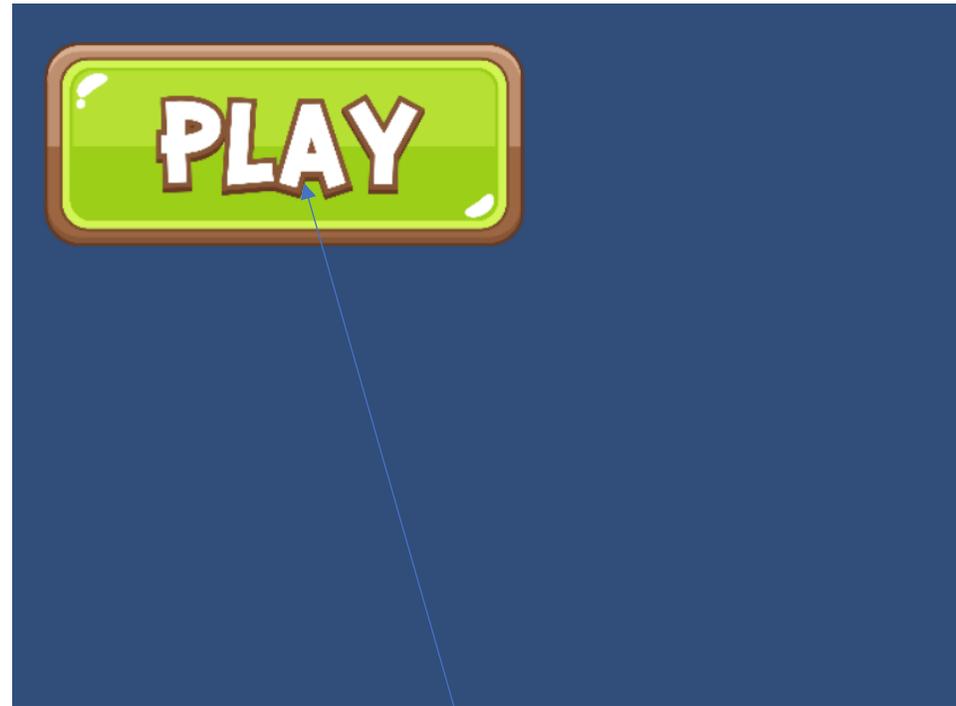
# Mais sobre o mouse.

- É possível detectar click e outros eventos sobre os assets.
- É necessário criar um script exclusivo para o asset ou grupo de assets que deseja detectar eventos.
- Posicione o código a seguir nos assets.
  - Ps. É um script diferente do script do personagem...

```

using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.SceneManagement;
public class ClickAsset : MonoBehaviour{
    //executado quando acontece um click no asset.
void OnMouseDown() {
    string tagObjeto = gameObject.tag;
    print("GameObject clicado: " + tagObjeto);
    if (tagObjeto == "botao_start") {
        SceneManager.LoadScene("fase01");
    }
}
//executado quando click é solto no asset.
void OnMouseUp() {
    string tagObjeto = gameObject.tag;
    print("Mouse solto sobre o GameObject: " + tagObjeto);
}
//executado quando mouse entra no asset.
void OnMouseEnter() {
    string tagObjeto = gameObject.tag;
    print("Mouse entrou no GameObject: " + tagObjeto);
}
//executado quando mouse sai no asset.
void OnMouseExit() {
    string tagObjeto = gameObject.tag;
    print("Mouse saiu do GameObject: " + tagObjeto);
}
//executado quando mouse está sobre o asset.
void OnMouseOver() {
    string tagObjeto = gameObject.tag;
    print("Mouse está sobre o GameObject: " + tagObjeto);
}
}

```



Esse script é inserido no botão  
 Configure o botão com a tag: "botao\_start"  
 Configure o botão com colisor

Quais outras mecânicas poderiam ser criadas  
 com base nesse script?