



# 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

# Atributos

```
float Vx;      //velocidade em x
float Vy;      //velocidade em y
float VelocidadeAndar;
float VelocidadePular;
float ContadorPulos;
float TotalPulos;
float DirecaoHorizontal;
Rigidbody2D CorpoRigido;
SpriteRenderer Renderer;
```

```
void Start ()
```

```
void Start () {
```

```
    TotalPulos = 2;
```

```
    VelocidadeAndar = 5;
```

```
    VelocidadePular = 5;
```

```
    CorpoRigido = GetComponent<Rigidbody2D> ();
```

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

```
    CorpoRigido.freezeRotation = true;
```

```
}
```

```
void Update ()
```

```
void Update () {  
    MovimentoHorizontalFlip ();  
    PuloDuplo();  
    EntradasMouse();  
}
```

# mouse

```
void EntradasMouse(){
    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;  
}
```

# 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 ']'.
<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.

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.Joystick1Button5);` => Botão Y do Controle do Xbox

```
void Pular(){
    ApertouJump = Input.GetButtonDown("Jump") || Input.GetKeyDown(KeyCode.Joystick1Button5);
    if (ApertouJump == true && ContadorPulos < TotalMaximoPulos){
        Vx = CorpoRigido.velocity.x;
        Vy = VelocidadePular * 0.5f;
        VetorVelocidade = new Vector2(Vx, Vy);
        CorpoRigido.velocity = VetorVelocidade;
        ContadorPulos = ContadorPulos + 1;
    }
}
```

# Posição do mouse na cena.

```
void EntradasMouse(){  
    //Recupera a posição (x,y) do mouse  
    //o centro da camera é a posicao (0,0)  
    Vector3 posicaoMouse = Camera.main.ScreenToWorldPoint (Input.mousePosition);  
    float x = posicaoMouse.x;  
    float y = posicaoMouse.y;  
    print (x + " - " + y);  
}
```

# O que está acontecendo?

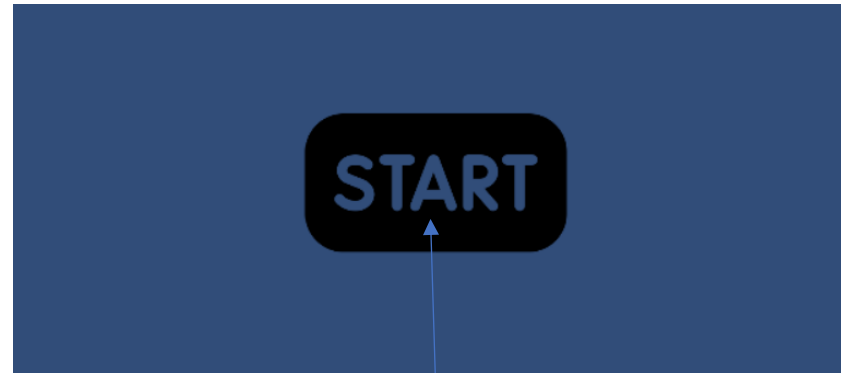
## MovimentoHorizontalFlip()

```
void MovimentoHorizontalFlip(){
    Vector3 posicaoMouse = Camera.main.ScreenToWorldPoint (Input.mousePosition);
    if (posicaoMouse.x < 0) {
        DirecaoHorizontal = -1;
    }
    if (posicaoMouse.x > 0) {
        DirecaoHorizontal = 1;
    }
    // Gera uma nova velocidade em x
    Vx = VelocidadeAndar * DirecaoHorizontal;
    //Recupera a velocidade em y que o personagem já possui
    Vy = CorpoRigido.velocity.y;
    //Cria um vetor de velocidade com os valores da velocidade em x e da velocidade em y (vx e vy)
    Vector2 andar = new Vector2 (Vx, Vy);
    // O vetor de velocidade é adicionado a velocidade do corpo rígido do personagem
    CorpoRigido.velocity = andar;
    if (DirecaoHorizontal < 0) {
        Renderer.flipX = true;
    }else if(DirecaoHorizontal > 0){
        Renderer.flipX = false;
    }
}
```

# 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"

Quais outras mecânicas poderiam ser criadas  
com base nesse script?