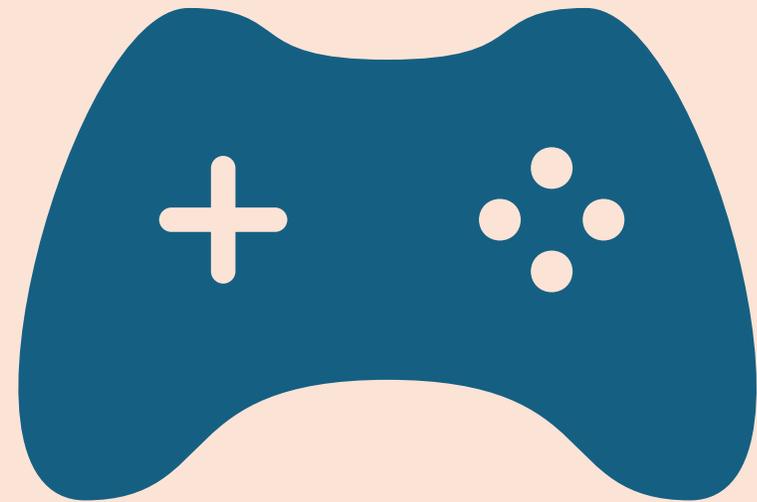


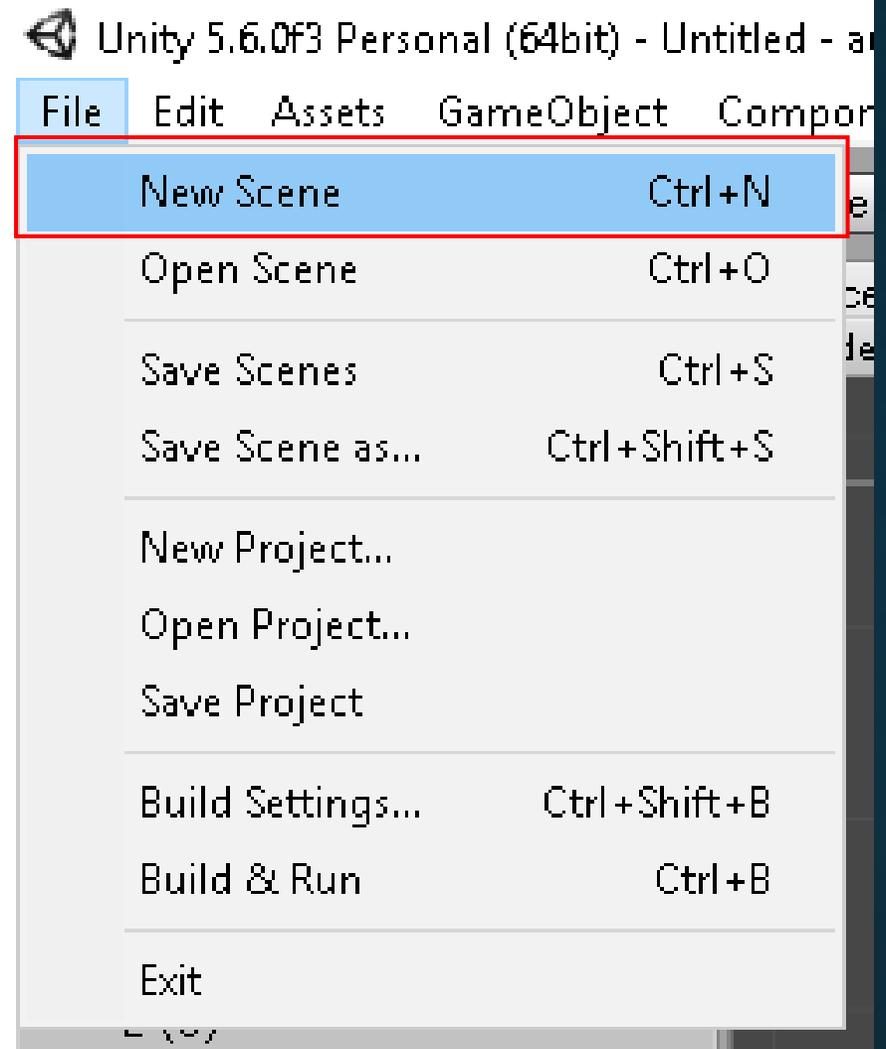
# NOVA FASE / GAME OVER

Prof. Me. Hélio Esperidião



# CRIAR UMA NOVA FASE OU TELA DE GAME OVER

- Considerando que já existe uma cena salva com a primeira fase.
  - A cena que existe foi salva com o nome “fase01”
- Crie uma nova cena (“fase”)
  - Menu >> File >> New Scene



# Nova cena

- Crie um novo cenário e acrescente um asset de game over.
  - É possível criar uma nova fase inteira, não somente uma cena de gameover.
- Vá ao menu file e salve a nova cena como “fase02”
  - Menu >> File >> Save Scene
    - Não modifique a pasta onde será salvo o arquivo, apenas de um nome



# Cenas

- Veja que os dois cenários salvos aparecem nos assets como o nome salvo (“fase01”, “fase02”)



- Abra o arquivo(“fase01”) da primeira cena.
- Vá ao menu File >> Build Settings.
- File >> Build Settings
- **A primeira cena da lista é a cena onde o jogo começa.**

Unity 5.6.0f3 Personal (64bit) - fase02.un

File Edit Assets GameObject Comp

- New Scene Ctrl+N
- Open Scene Ctrl+O
- Save Scenes Ctrl+S
- Save Scene as... Ctrl+Shift+S
- New Project...
- Open Project...
- Save Project
- Build Settings... Ctrl+Shift+B**
- Build & Run Ctrl+B
- Exit

Build Settings

Scenes In Build

- fase01 0

Platform

- PC, Mac & Linux Standalone
- iOS
- tvOS
- Android
- Tizen
- Xbox One
- PS Vita

PC, Mac & Linux Standalone

Target Platform Windows

Architecture x86

Copy PDB files

Development Build

Autoconnect Profiler

Script Debugging

Learn about Unity Cloud Build

Switch Platform Player Settings... Build Build And Run

Lista de cenas

Clique na opção Add openScenes

Verifique que o cenário aberto Será adicionado a lista.

- Abra a segunda fase por meio de dois cliques no arquivo "fase02"
- Vá ao menu file e escolha a opção Build Settings.

Unity 5.6.0f3 Personal (64bit) - fase02.un

File Edit Assets GameObject Comp

New Scene Ctrl+N

Open Scene Ctrl+O

Save Scenes Ctrl+S

Save Scene as... Ctrl+Shift+S

New Project...

Open Project...

Save Project

**Build Settings... Ctrl+Shift+B**

Build & Run Ctrl+B

Exit

Build Settings

Scenes In Build

<input checked="" type="checkbox"/> fase01	0
<input checked="" type="checkbox"/> fase02	1

Lista de cenas

Add Open Scenes

Platform

PC, Mac & Linux Standalone

iOS

tvOS

Android

Tizen

Xbox One

PS Vita

PC, Mac & Linux Standalone

Target Platform Windows

Architecture x86

Copy PDB files

Development Build

Autoconnect Profiler

Script Debugging

Learn about Unity Cloud Build

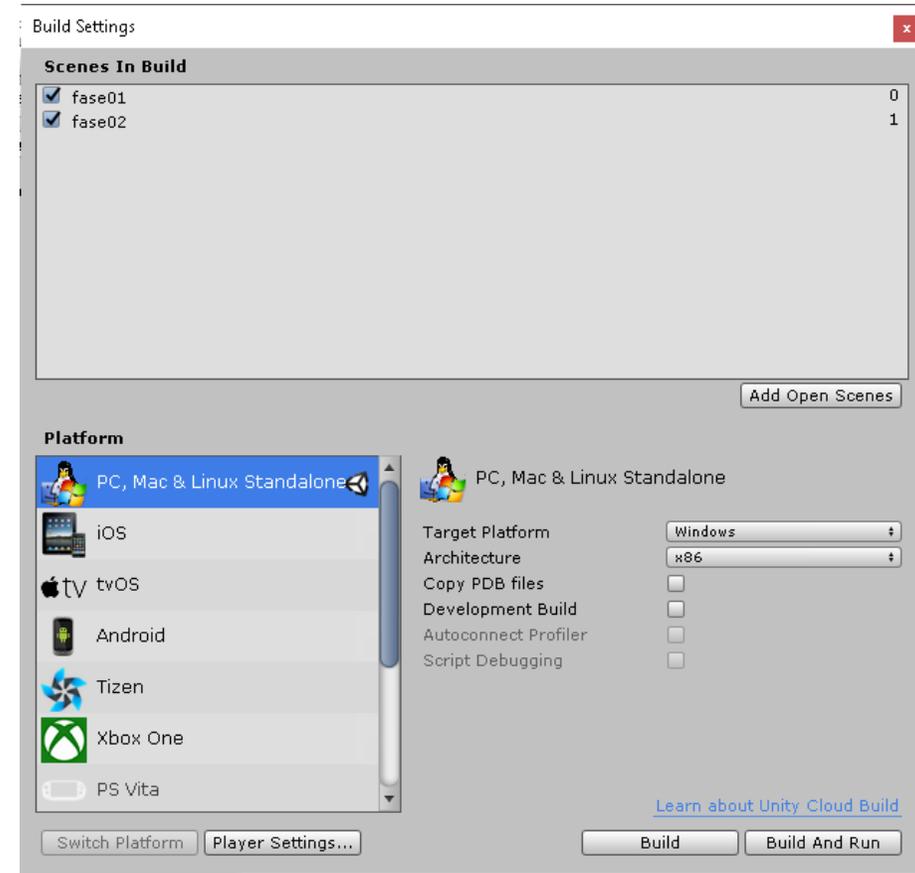
Switch Platform Player Settings... Build Build And Run

Clique na opção Add openScenes

Verifique que o cenário aberto Será adicionado a lista.

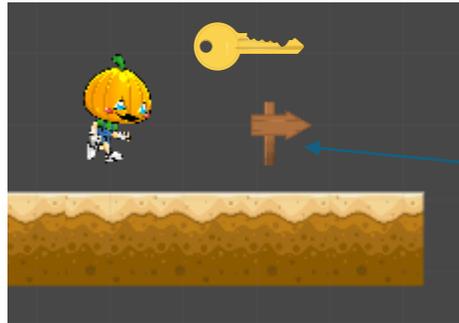
# Scenes in Build

- Todas as cenas que poderão ser carregadas no jogo devem ser configuradas no Build Settings.
- Apenas a posição da primeira fase é importante, pois o jogo será iniciado a partir da primeira fase da lista.
- As outras cenas serão carregadas via programação



# Configurações

- Configura a placa da seta com o boxcollider.



Marque o elemento que determina o fim do jogo, ou  
A morte do personagem com uma tag

A seta foi marcada com a tag **Fim**

# Importe a biblioteca

```
using UnityEngine;
```

```
//Acréscete a biblioteca que permite utilizar o SceneManager para  
trocar de cena
```

```
using UnityEngine.SceneManagement;
```

# Atributos

```
string TagObjetoTocado;

bool PegouChave01;

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

# Start()

```
void Start () {  
    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;  
  
    // 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 máxima 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;  
}
```

# Uso Simples

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;  
  
    if (TagObjetoTocado == "fim" ) {  
        SceneManager.LoadScene ("fase02");  
    }  
  
    if (TagObjetoTocado == "fim"  && PegouChave1 == true ) {  
        SceneManager.LoadScene ("fase02");  
    }  
  
    if (TagObjetoTocado == "chave01 ") {  
        PegouChave01 = true;  
        Destroy (objetoTocado.gameObject);  
    }  
}
```

Quando personagem tocar na tag "Fim"  
O novo cenário será carregado.

```
void OnCollisionEnter2D(Collision2D objetoTocado ){  
    TagObjetoTocado = objetoTocado.gameObject.tag;
```

Uso Elaborado

```
1  if (TagObjetoTocado == "fim" && PegouChave1 == true ) {  
    SceneManager.LoadScene ("fase02");  
}
```

```
2  if (TagObjetoTocado == "chave01 ") {  
    PegouChave01 = true;  
    Destroy (objetoTocado.gameObject);  
}
```

```
}
```

Observe que para trocar de cena é necessário tocar na tag “fim” e o valor da variável **PegouChave1** deve ser true.  
No start **PegouChave1** é inicializada com false, verifique no segundo if que se o jogador tocar na “chave01” o valor de **PegouChave1** é mudado para verdadeiro.

# Use a criatividade

## Exemplos criativos para carregar novas cenas

```
if (TagObjetoTocado == "fim" && QtdEstrelas > 10) {  
    SceneManager.LoadScene ("fase02");  
}
```

```
if (TagObjetoTocado == "fim" && PegouChave1 == true && PegouChave2 ==true) {  
    SceneManager.LoadScene ("fase02");  
}
```

```
if (TagObjetoTocado == "fim" && Energia >=75) {  
    SceneManager.LoadScene ("fase02");  
}
```

Para que os exemplos funcionem é necessário criar, inicializar e modificar o valor de outras variáveis não apresentadas no trecho acima. Use lógica para fazer com que as ideias apresentadas sejam realidade