

Prof. Me. Hélio Esperidião.

Tipos de Colisão

Tipos básicos de Colisão

- `OnCollisionEnter2D(Collision2D objetoTocado)`
 - Executa quando um colisor toca em outro colisor. A variável `objetoTocado` possui o elemento que foi tocado.
- `OnCollisionStay2D(Collision2D tocandoNoObjeto)`
 - Executa quando um colisor permanece tocando em outro colisor. A variável `tocandoNoObjeto` possui o elemento que permanece em contato.
- `OnCollisionExit2D(Collision2D objetoQueParouTocar)`
 - Executa quando um colisor para de tocar em outro colisor. A variável `objetoQueParouTocar` possui o elemento que não está mais em contato.

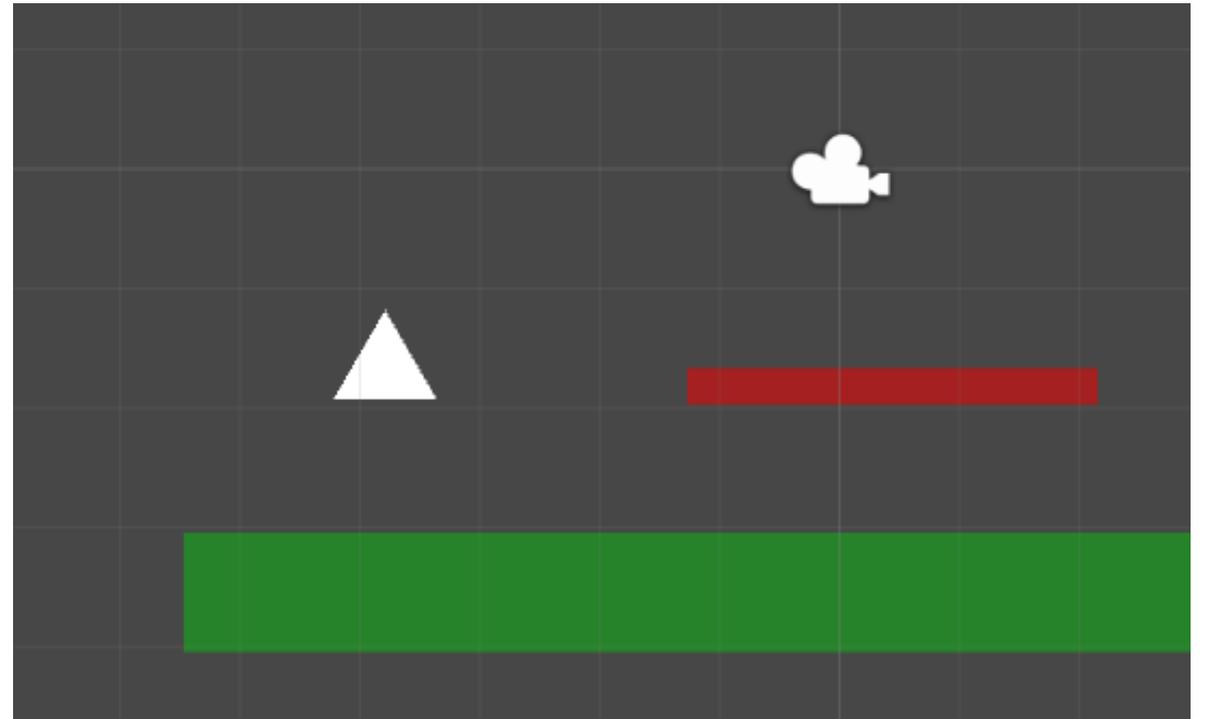
OnCollisionEnter2D()

É chamado quando um objeto com um Collider2D entra em contato com outro objeto com um Collider2D, gerando uma colisão.

Esse método é útil para detectar colisões físicas, como quando um jogador colide com uma parede, uma bola atinge uma plataforma ou dois objetos colidem em um jogo de quebra-cabeça

Configuração inicial

- Triângulo(Personagem):
 - Corpo Rígido
 - Colisor
- Chão
 - Colisor
- Chão Vermelho
 - Colisor
 - Tag: lama



Atributos

```
public class tempBimestre2 : MonoBehaviour {
    string TagObjetoStay; //Armazena a tag que o personagem está tocando
    string TagObjetoExit; //Armazena a tag que o personagem parou de tocar
    int TotalPulos;
    PhysicsMaterial2D MaterialSemAtrito;
    int ContadorPulos;
    SpriteRenderer Renderer;
    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;
```

```
void Start () {
```

```
TagObjetoStay="";  
TagObjetoExit="";
```

```
MaterialSemAtrito = new PhysicsMaterial2D();
```

```
TotalPulos = 1;
```

```
ContadorPulos = 0;
```

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

```
PegouChave01 = false;
```

```
TagObjetoTocado = "";
```

```
ApertouBotaoPular = false; // inicia a variável com false;
```

```
EstaTocandoAlgumColisor = false; // inicializa a variável 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;
```

```
}
```

start

OnCollisionStay2D(Collision2D objetoStay)

```
//Evento que acontece quando um colisor PERMANECE tocando outro colisor.  
  
//A variável objetoStay possui o objeto que o colisor permanece tocando.  
  
void OnCollisionStay2D(Collision2D objetoStay){  
    TagObjetoStay = objetoStay.gameObject.tag;  
    if (TagObjetoStay == "lama") {  
        VelocidadeHorizontalMaxima = 1.0f;  
    }  
}
```

OnCollisionExit2D(Collision2D objetoExit)

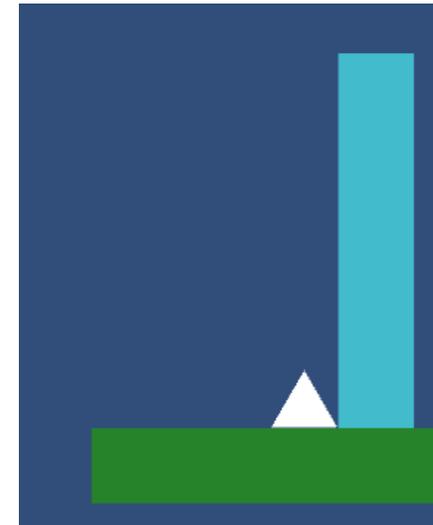
```
//Evento que acontece quando um colisor para de tocar outro colisor.  
  
//A variável objetoExit contem o objeto que foi parado de tocar pelo colisor.  
  
void OnCollisionExit2D(Collision2D objetoExit){  
    TagObjetoExit = objetoExit.gameObject.tag;  
    if (TagObjetoExit == "lama") {  
        VelocidadeHorizontalMaxima = 5;  
    }  
}
```

OnTriggerEnter2D

- O `OnTriggerEnter2D()` é chamado quando um objeto com um `Collider2D` entra em contato com o "trigger" de outro objeto.
- Os "triggers" são volumes de colisão que não possuem física de colisão, mas emitem um sinal quando um objeto colide com eles.
- Esse método é útil para detectar eventos, como quando um jogador entra em uma área específica do jogo, quando um inimigo detecta a presença do jogador ou quando um item é coletado ao entrar em contato com ele.

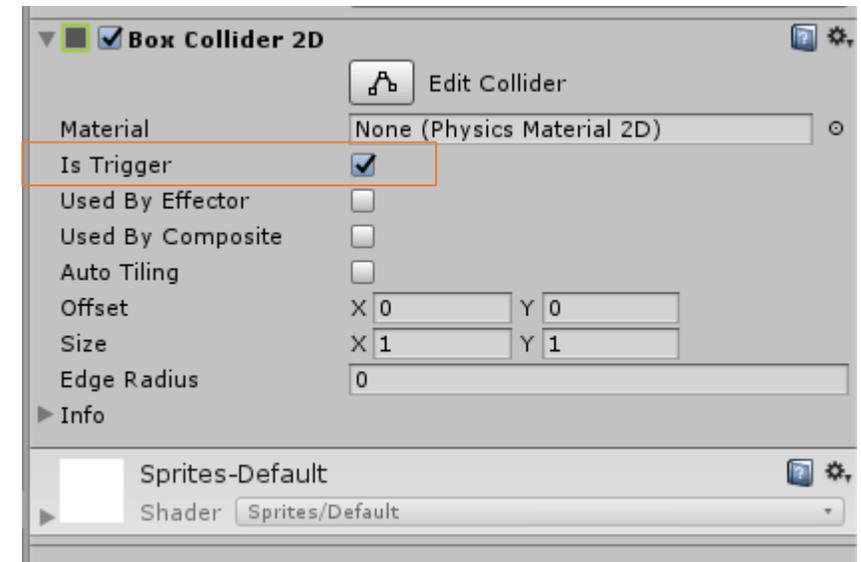
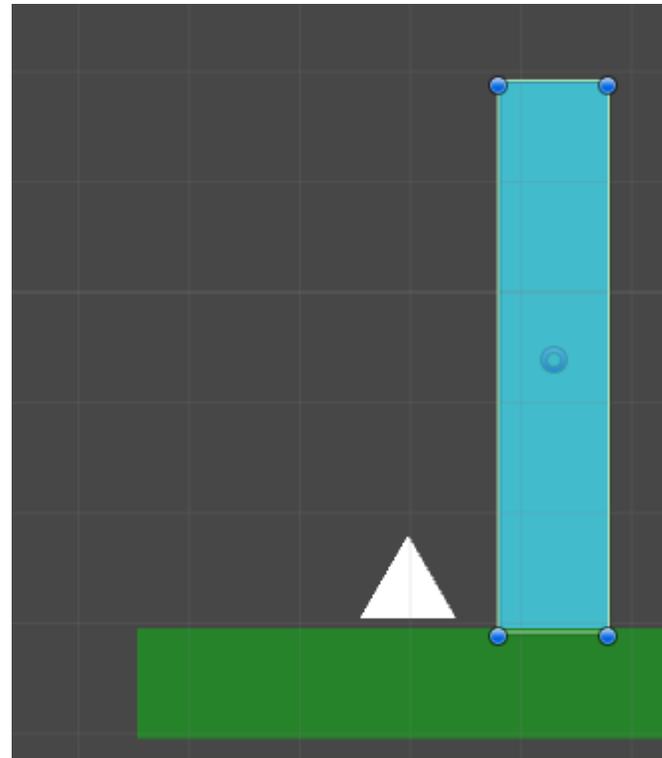
OnCollisionEnter2D vs OnTriggerEnter2D

- Em resumo, OnCollisionEnter2D() é usado para colisões físicas enquanto OnTriggerEnter2D() é usado para detectar eventos em "trigger volumes" sem física.
- No **collision** **há** uma colisão física.
- No **Trigger** **não há** colisão física
- **No trigger o evento é disparado sem que exista uma representação física.**
- É possível verificar que na figura de acima acontece o trigger e na figura de baixo acontece o collision.



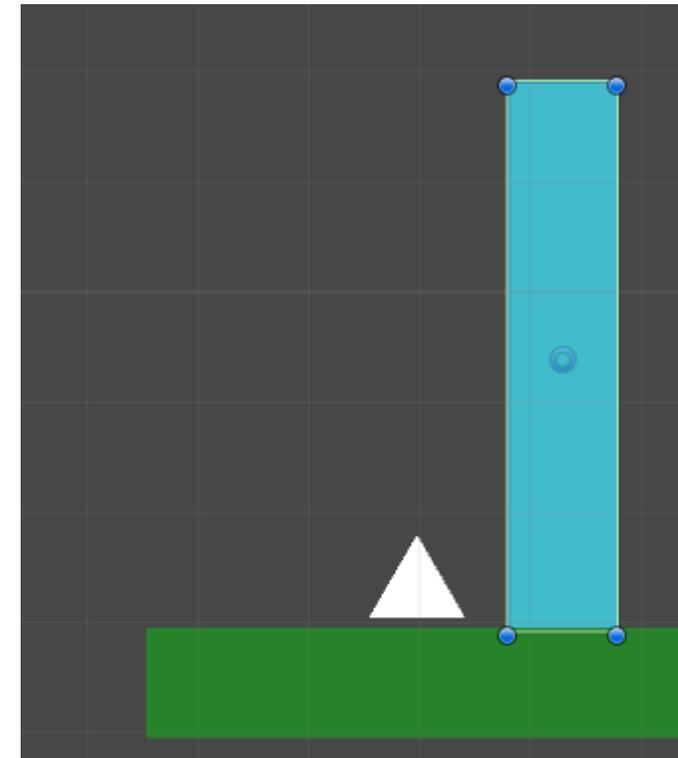
Como configurar um colisor como trigger?

- Exemplo: Selecione o retângulo azul, no inspector marque a propriedade “Is Trigger”.



Configuração inicial:

- Triângulo(Personagem):
 - Corpo Rígido
 - Colisor
- Chão
 - Colisor
- Parede Azul
 - Colisor
 - Is Trigger : true (marcado)
 - Tag: paredeAzul



```
public class tempBimestre2 : MonoBehaviour {
```

```
    string TagTriggerStay;  
    string TagTriggerEnter;  
    string TagTriggerExit;
```

```
    string TagObjetoTocando;  
    string TagObjetoParouTocar;
```

```
    int ContadorPulos;
```

```
    SpriteRenderer Renderer;
```

```
    string TagObjetoTocado;
```

```
    bool PegouChave01;
```

```
    bool ApertouBotaoPular; // para saber se o botão de pular foi apertado
```

```
    float VelocidadePuloSimples; // variável 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
```

Atributos

Start ()

```
void Start () {  
    TagTriggerStay = "";  
    TagTriggerEnter = "";  
    TagTriggerExit = "";  
  
    TagObjetoTocado = "";  
    TagObjetoParouTocar = "";  
  
    ContadorPulos = 0;  
    Renderer = GetComponent<SpriteRenderer> ();  
    PegouChave01 = false;  
    TagObjetoTocado = "";  
    ApertouBotaoPular = false; // inicia a variavel com false;  
    EstaTocandoAlgunColisor = 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;  
}
```

OnTriggerEnter2D(Collider2D objetoTriggerTocado)

```
// É um evento que é executado quando entra na área de colisão do trigger
void OnTriggerEnter2D(Collider2D objetoTriggerEnter){
    TagTriggerEnter = objetoTriggerEnter.gameObject.tag;
    if (TagTriggerEnter == "paredeAzul") {
        Destroy (objetoTriggerEnter.gameObject);
        print ("OnTriggerEnter2D: " + TagTriggerEnter);
    }
}
```

OnTriggerStay2D(Collider2D objetoTriggerStay)

```
// Evento que é executado quando permanece na área de colisão do trigger  
  
void OnTriggerStay2D(Collider2D objetoTriggerStay){  
    TagTriggerStay = objetoTriggerStay.gameObject.tag;  
    if (TagTriggerStay == "paredeAzul") {  
        print ("Trigger Stay: " + TagTriggerStay);  
    }  
}
```

OnTriggerExit2D(Collider2D objetoTriggerExit)

```
// Evento que é executado quando o colisor sai da área de colisão do
trigger

void OnTriggerExit2D(Collider2D objetoTriggerExit){
    TagTriggerExit = objetoTriggerExit.gameObject.tag;
    if (TagTriggerEnter == "paredeAzul") {
        print ("Trigger Exit: " + TagTriggerExit);
    }
}
```