

V_a



$v_b = 0$

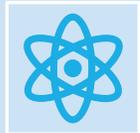
SISTEMA DE COLISÃO

PROF. ME. HÉLIO ESPERIDIÃO

Colisores



Para que as colisões aconteçam os objetos envolvidos devem ser configurados como colisores.



Colisões apenas acontecem quando dois elementos são configurados como colisores.



O Unity permite algumas formas de colisores como:

Circle Collider 2D - Possui a forma circular ou oval.

Box Collider 2D - Possui a forma de um retângulo ou quadrado.

Polygon Collider 2D – Tenta se adaptar a forma do asset.

Tipos de colisão

OnCollisionEnter2D()

- Quando o personagem toca em outro componente.

OnCollisionStay2D()

- Quando o personagem permanece sobre um componente.

OnCollisionExit2D()

- Quando o personagem para de tocar um determinado componente.

OnCollisionEnter2D()

É um evento que é disparado automaticamente quando dois colisores se chocam.

Exemplo: Quando a caixa de colisão do personagem toca na caixa de colisão de outro elemento um elemento configurado como vida.

O que acontece quando tocar no elemento configurado como vida?

O que acontece quando tocar no elemento configurado como espinho?

Quais as possibilidades de tocar em algo e acontecer alguma coisa?

Configuração inicial

Personagem (circulo)

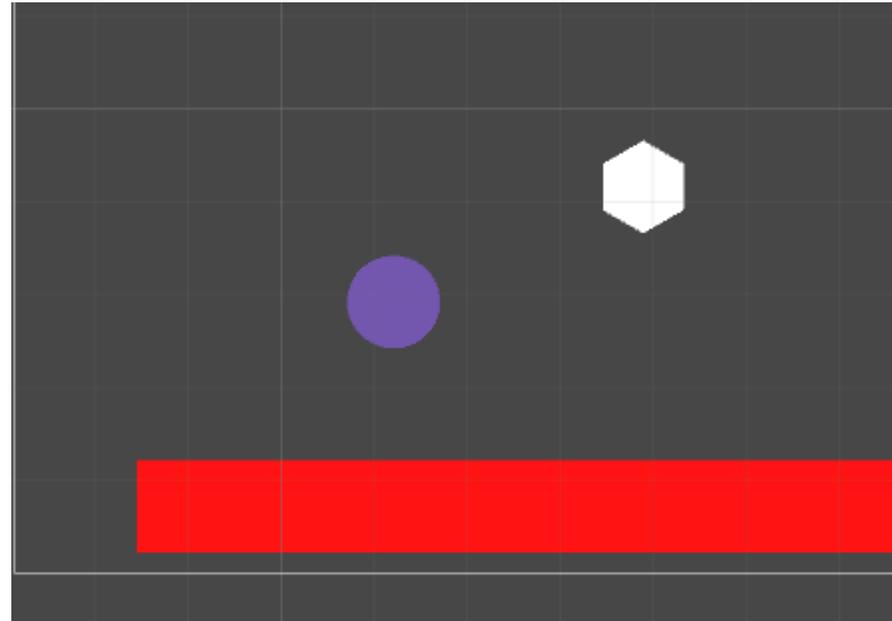
- Component >> Physic2D>>BoxCollider2D
- Component >> Physic2D>>RigidBody2D

Chão

- Component >> Physic2D>>BoxCollider2D

Hexágono(Objeto para tocar)

- Component >> Physic2D>>BoxCollider2D
- Component >> Physic2D>>RigidBody2D



Tags

É um recurso que permite nomear objetos de cena.

Vários objetos podem possuir o mesmo nome.

Em caso de colisão com um elemento é possível identificar o elemento por meio de uma tag.

Exemplos possíveis de tags:

- Espinho, vida, bônus, estrela, etc.

Caso o personagem toque em um elemento de cena configurado com a tag “espinho” ele pode perder vida.



Configuração do hexágono

Vamos definir um tag para o hexágono. Essa será chamada de “nomeTag”

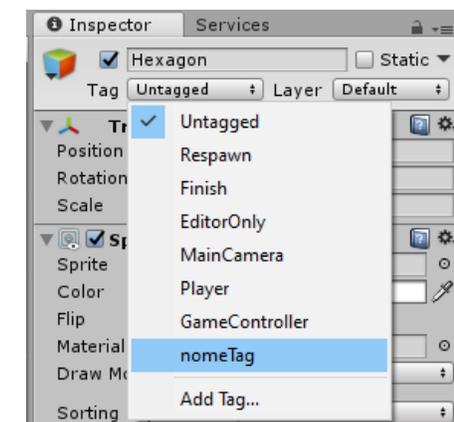
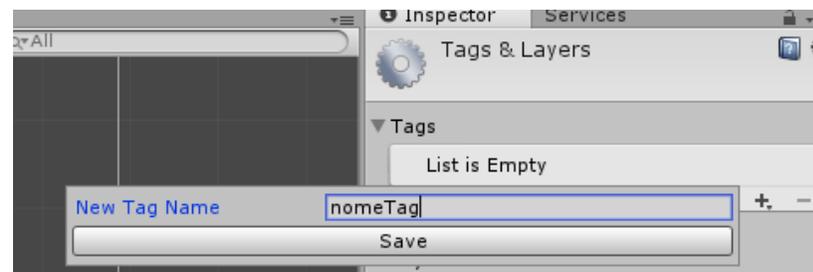
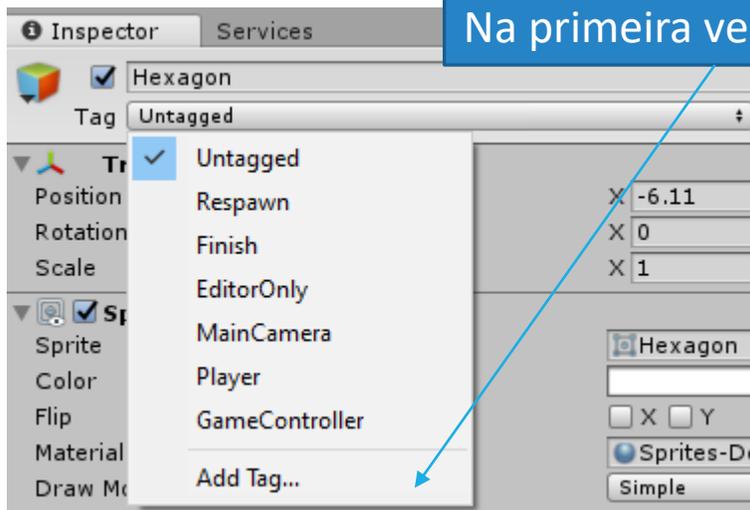
Tag: Define um nome para um componente ou grupo de componentes.

- Clique na hexágono e vá para o **Inspector**
- Procure a caixa de combinação chamada “Tag”
- Defina a mesma tag (“nomeTag”) para os objetos desejados.

Na primeira vez add a Tag

Clique em “+” Digite o nome da tag e salve

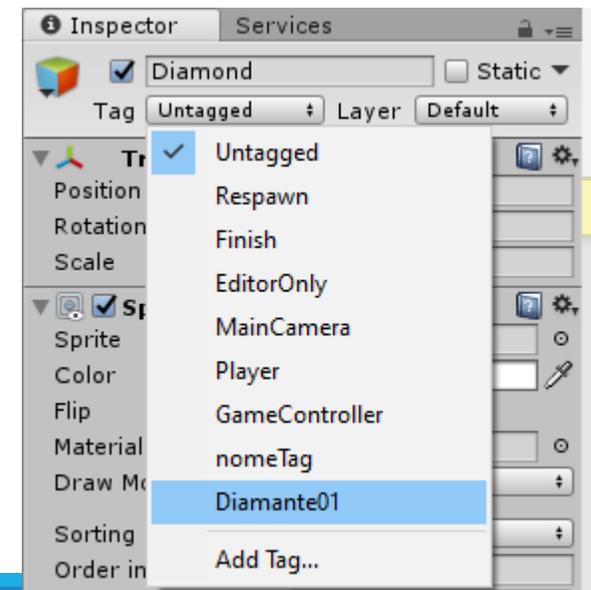
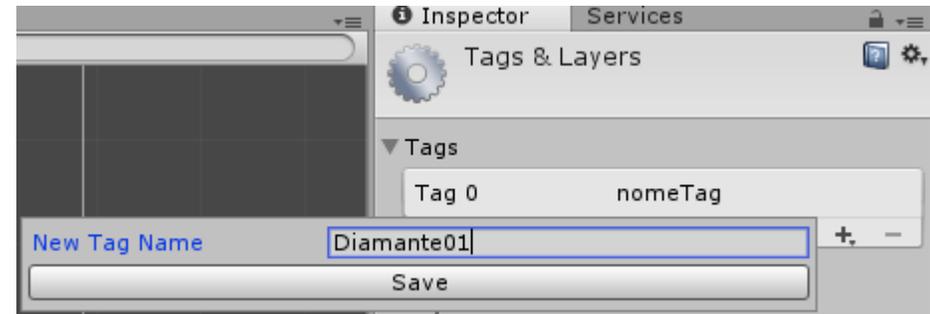
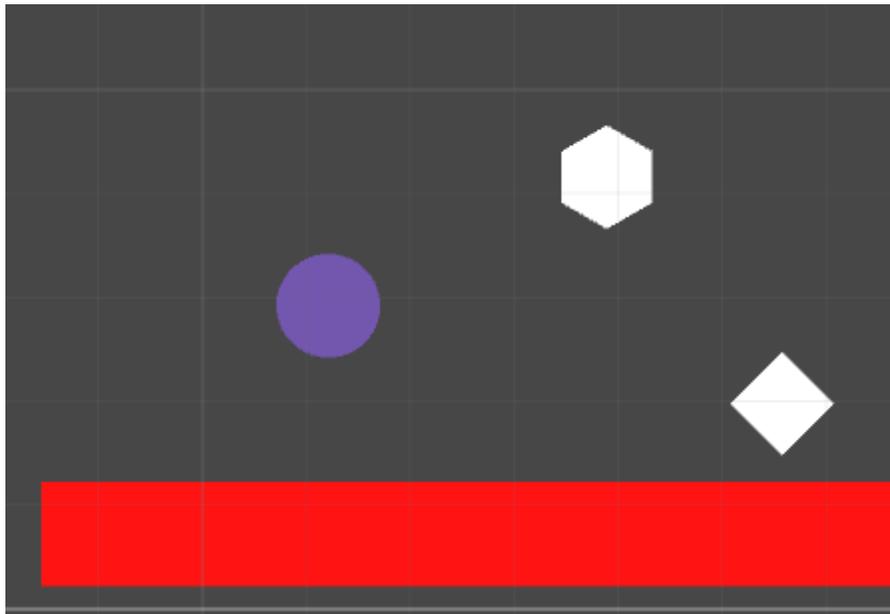
Clique novamente no objeto e escolha a tag criada anteriormente



Você pode criar outras tags

No Exemplo podemos verificar a criação das tags:

- “nomeTag” e “Diamante01”



OnCollisionEnter2D

Esse método(“Bloco”) é executado toda vez que o componente colide com outro.

É chamado automaticamente quando um colisor atinge o outro.

A variável **objetoTocado** possui informações sobre o objeto tocado

- Exemplo de informações armazenadas na variável:
 - Corpo rígido, **tag**, etc

Você precisa digitar o método como no exemplo abaixo:

```
void OnCollisionEnter2D(Collision2D objetoTocado){  
    //é executado sempre que um colisor toca em outro colisor  
    //a variável objetoTocado possui dados do objeto tocado.  
}
```

OnCollisionEnter2D

Método("Bloco") que é chamado toda vez que o componente toca em outro. Ambos devem possuir colisor para que O método seja chamado.

```
//não precisa ser chamado no update  
//executa automaticamente toda vez que colidir com algo  
void OnCollisionEnter2D(Collision2D objetoTocado){  
    string tagObjetoTocado = objetoTocado.gameObject.tag;  
    if (tagObjetoTocado == "Diamante01") {  
  
        //só executa o bloco entre {} se o objeto tocado  
        //possuir a tag "Diamante01"  
        Destroy(objetoTocado.gameObject);  
  
    }  
}
```

Todo código do bloco azul só é executado se a tag do objeto tocado é igual a "Diamante01"

Destrói imediatamente o objeto de jogo do objeto tocado

Verifica se colidiu com um componente de tag "Diamante01"

Código completo

```
public class JogadorColisao : MonoBehaviour {
    float Vx;        // velocidade em x
    float Vy;        // velocidade em y
    float DirecaoHorizontal;
    float VelocidadeAndar;    //velocidade de andar
    Rigidbody2D CorpoRigido;    //corpo rígido do elemento que possui o script
    void Start () { //executado quando o jogo inicia
        Vx = 0;                //inicia velocidade em x
        Vy = 0;                //inicia velocidade em y
        DirecaoHorizontal = 0;
        VelocidadeAndar = 5;
        CorpoRigido = GetComponent<Rigidbody2D>(); //inicia com o corpo rígido do elemento que possui o script
        CorpoRigido.freezeRotation = true;        // impede rotação no eixo
    }
    void Update () {          //executa a cada frame
        Andar();              //chama para execução: Andar() a cada frame
    }
    void Andar(){
        DirecaoHorizontal = Input.GetAxis ("Horizontal");
        Vx = VelocidadeAndar * DirecaoHorizontal;
        Vy = CorpoRigido.velocity.y;
        Vector2 velocidadeX = new Vector2 (Vx, Vy);
        CorpoRigido.velocity = velocidadeX;
    }
    //não precisa ser chamado no update
    //executa automaticamente toda vez que o colisor do elemento que possui o script colidir com outro colisor
    void OnCollisionEnter2D(Collision2D objetoTocado){
        string tagObjetoTocado = objetoTocado.gameObject.tag; //retorna a tag do elemento que foi tocado
        if (tagObjetoTocado == "Diamante01") {
            Destroy (objetoTocado.gameObject); //destrói o gameObject(Objeto de jogo) do elemento tocado
        }
    }
}
```

```

public class JogadorColisao : MonoBehaviour {
    float Vx; // velocidade em x
    float Vy; // velocidade em y
    float DirecaoHorizontal; //depende das setas direcionais
    float VelocidadeAndar; //velocidade de andar
    float VelocidadePulo; //Velocidade do pulo
    Rigidbody2D CorpoRigido; //corpo rígido do elemento que possui o script
    CircleCollider2D Colisor;
    void Start () { //executado quando o jogo inicia
        Vx = 0; //inicia velocidade em x
        Vy = 0; //inicia velocidade em y
        DirecaoHorizontal = 0;
        VelocidadePulo = 7;
        VelocidadeAndar = 5;
        CorpoRigido = GetComponent<Rigidbody2D>(); //inicia com o corpo rígido do elemento que possui o script
        Colisor = GetComponent<CircleCollider2D>();
        CorpoRigido.freezeRotation = true; // impede rotação no eixo
    }
    void Update () { //executa a cada frame
        Andar(); //executa Andar() a cada frame
        PuloSimples(); //executa PuloSimples() a cada frame
    }
    void Andar(){
        DirecaoHorizontal = Input.GetAxis ("Horizontal");
        Vx = VelocidadeAndar * DirecaoHorizontal;
        Vy = CorpoRigido.velocity.y;
        Vector2 velocidadeX = new Vector2 (Vx, Vy);
        CorpoRigido.velocity = velocidadeX;
    }
    void PuloSimples(){
        bool apertouBotao = Input.GetButtonDown ("Jump");
        bool estaTocandoAlgo = Colisor.IsTouchingLayers();
        if (apertouBotao == true && estaTocandoAlgo ==true) {
            Vx = CorpoRigido.velocity.x;
            Vy = VelocidadePulo;
            Vector2 velocidadePulo = new Vector2 (Vx, Vy);
            CorpoRigido.velocity = velocidadePulo;
        }
    }
    void OnCollisionEnter2D(Collision2D objetoTocado){
        string tagObjetoTocado = objetoTocado.gameObject.tag;
        if (tagObjetoTocado == "Diamante01") {
            Destroy (objetoTocado.gameObject);
        }
    }
}

```

Código completo com pulo

Destroy

```
//não precisa ser chamado no update
//executa automaticamente toda vez que colidir com algo
void OnCollisionEnter2D(Collision2D objetoTocado){
    string tagObjetoTocado = objetoTocado.gameObject.tag;
    if (tagObjetoTocado == "Diamante01") {
        Destroy(objetoTocado.gameObject, 1.5f);
    }
}
```

Pra representar um número float acrescente um f ao seu final.

Destroy pode receber um valor do tipo float
O valor recebido é o tempo para a destruição do elemento.
No exemplo o objeto é destruído 1,5 segundos depois do toque.
Observe que só é destruído se o objeto tocado possuir a tag
“Diamante01”

Arremessando objetos

Quando o personagem tocar em algo que possua a tag "Diamante01". O componente marcado com a tag "Diamante01" será arremessado a uma determinada velocidade de 5 em x e 5 em y

```
//não precisa ser chamado no update  
//executa automaticamente toda vez que colidir com algo  
void OnCollisionEnter2D(Collision2D objetoTocado){  
    string tagObjetoTocado = objetoTocado.gameObject.tag;  
  
    if (tagObjetoTocado == "Diamante01") {  
  
        //armazena na variável o corpo rígido do objeto que foi tocado.  
        Rigidbody2D corpoRigidoObjetoTocado = objetoTocado.rigidbody;  
        //cria um novo vetor de velocidade para o objeto tocado.  
        Vector2 novaVelocidade = new Vector2 (5, 5);  
        //a nova velocidade será de 45 graus  
        corpoRigidoObjetoTocado.velocity = novaVelocidade;  
    }  
}
```

Ângulos de “velocidade”

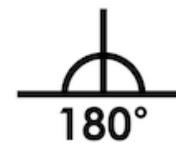
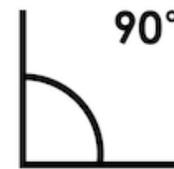
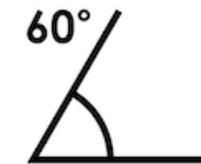
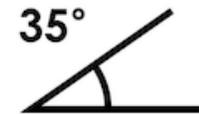
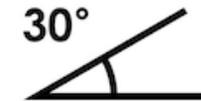
Análise simples:

Se $x = y$ ângulo igual a 45°

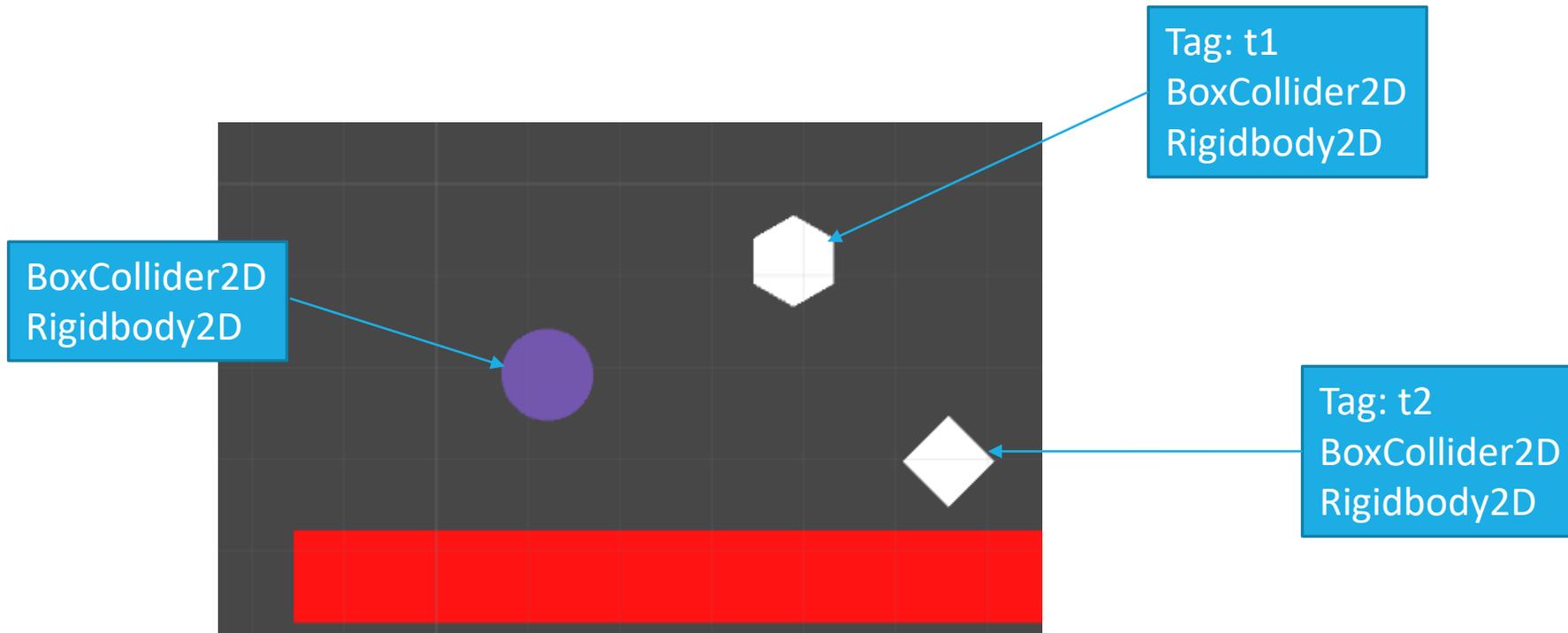
Se $x=0$ e $y > 0 \rightarrow$ ângulo = 90°

Se $x > y$ então ângulo menor que 45°

Se $x < y$ então ângulo maior que 45° .



Trabalhando com múltiplas tags



Crie um “if” para detectar cada tag.

```
void OnCollisionEnter2D(Collision2D objetoTocado){
    string tagObjetoTocado = objetoTocado.gameObject.tag;
    if (tagObjetoTocado == "t1") {
        Rigidbody2D corpoRigidoObjetoTocado = objetoTocado.rigidbody;
        Vector2 novaVelocidade = new Vector2 (5, 5);
        corpoRigidoObjetoTocado.velocity = novaVelocidade;
    }
    if (tagObjetoTocado == "t2") {
        Destroy (objetoTocado.gameObject);
    }
}
```

O código abaixo permite acompanhar o posicionamento do personagem para aplicar uma velocidade ao objeto tocado.

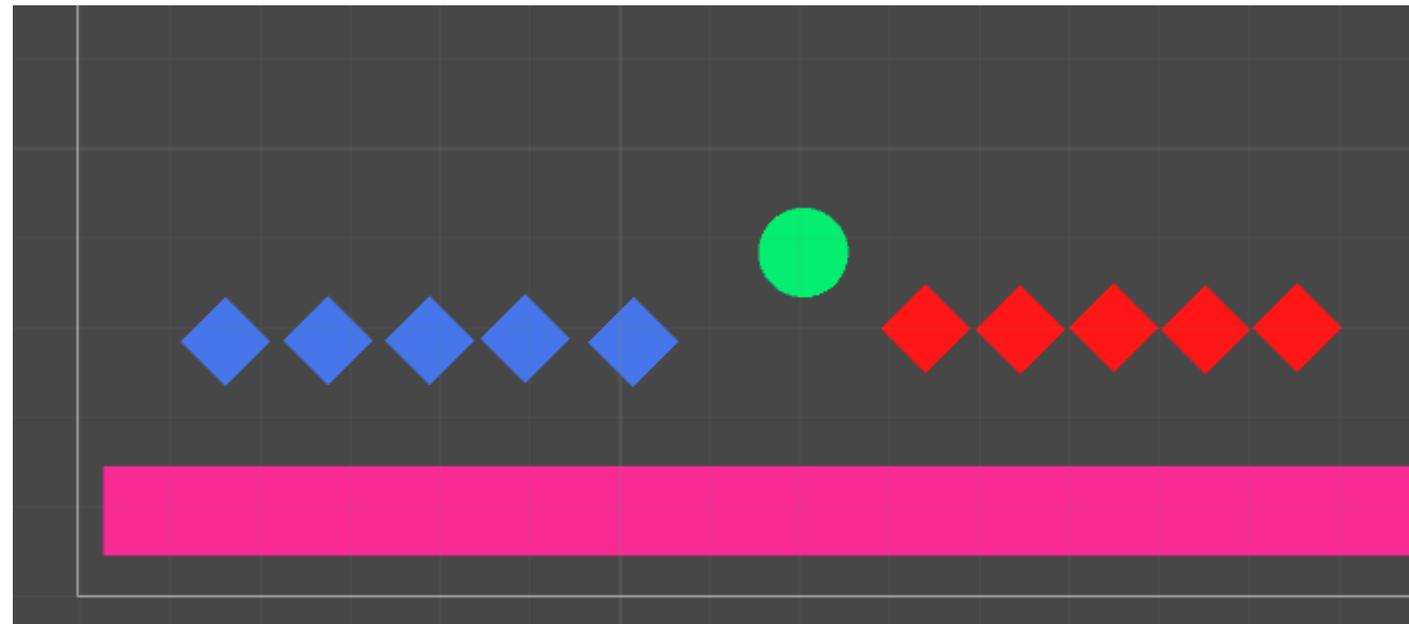
```
void OnCollisionEnter2D(Collision2D objetoTocado){
    string tagObjetoTocado = objetoTocado.gameObject.tag;
    if (tagObjetoTocado == "t1") {
        //armazena na variável "corpoRigidoObjetoTocado" o corpo rígido do objeto que foi tocado.
        Rigidbody2D corpoRigidoObjetoTocado = objetoTocado.rigidbody;
        //cria um novo vetor de velocidade para o objeto tocado.
        float tempX = 5 * Mathf.Sign(Vx); //Sign recupera o sinal da velocidade Vx 1 ou -1
        float tempy = 5 * Mathf.Sign(Vy); //Sign recupera o sinal da velocidade Vy 1 ou -1
        Vector2 novaVelocidade = new Vector2 (tempX, tempy);
        //a nova velocidade será de 45 graus
        corpoRigidoObjetoTocado.velocity = novaVelocidade;
    }
    if (tagObjetoTocado == "t2") {
        Destroy (objetoTocado.gameObject);
    }
}
```

Aumentar Velocidade de Movimento

Ao tocar em “t1” o personagem ganha velocidade.
Ao tocar em “t2” o personagem perde velocidade.

Configurações:

- Diamante azul
 - Colisor
 - Corpo Rígido
 - Tag: “tipo1”
- Diamante vermelho
 - Colisor
 - Corpo Rígido
 - Tag: “tipo2”



Aumentar Velocidade

```
void OnCollisionEnter2D(Collision2D objetoTocado){
    string tagTocada = objetoTocado.gameObject.tag;
    if (tagTocada == "tipo1") {
        if (VelocidadeAndar < 10) {
            VelocidadeAndar = VelocidadeAndar + 1;
        }
        Destroy (objetoTocado.gameObject);
    }
    if (tagTocada == "tipo2") {
        if (VelocidadeAndar >= 2) {
            VelocidadeAndar = VelocidadeAndar - 1;
        }
        Destroy (objetoTocado.gameObject);
    }
}
```

A velocidade é aumentada em +1 somente se for tocado no elemento com a tag "Tipo1 " e a velocidade for menor que 10

A velocidade é Diminuida em -1 somente se for tocado no elemento com a tag "Tipo2 " e a velocidade for maior que 2.