Teletransporte

Prof. Me. Hélio Esperidião

Teletransporte

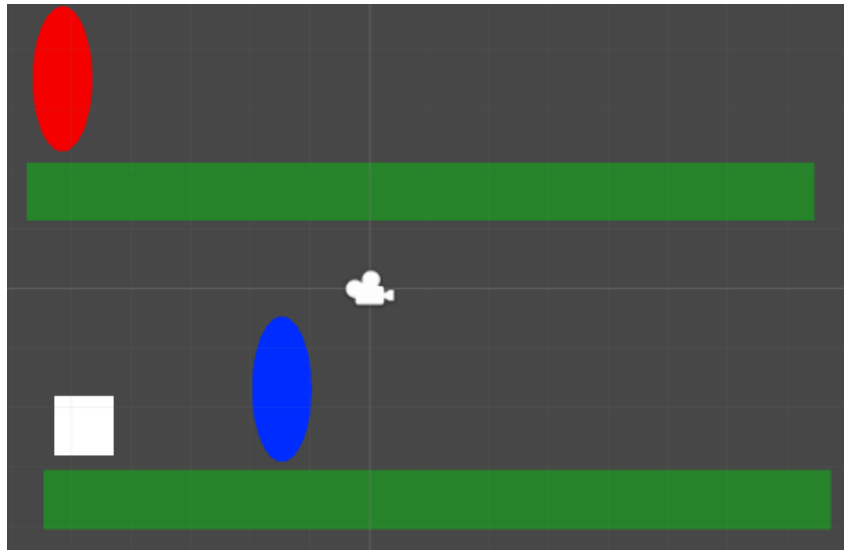
- Teletransporte ou teleporte (teleport em inglês) é um comando usado para mover um elemento de um ponto ao outro no mesmo mapa.
- Ele está presente em muitos jogos.
  - Maria quando ele entra pelo cano

Transform

- No Unity cada GameObject contém um componente fixo chamado **Transform**.
- Esse componente possui a propriedade: **Position.**
- **Position** determina a posição do GameObject na cena nos eixos X, Y e Z com valores que indicam o quanto o centro do GameObject está deslocado em cada eixo.

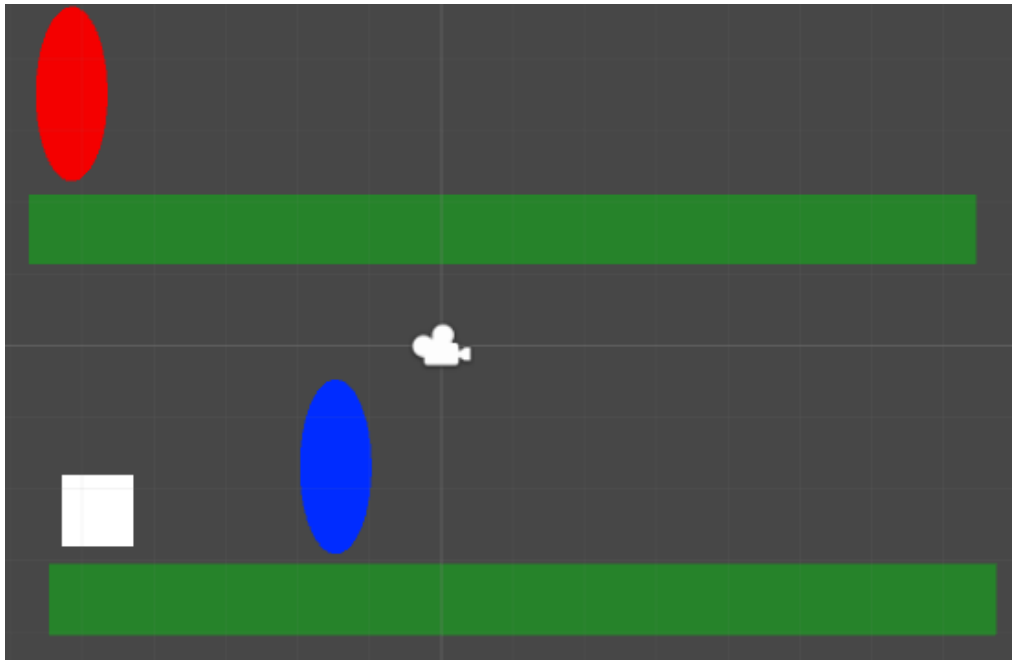
Configuração

- Portal Azul:
  - Component >> Physic2D>>PolygonCollider2D
  - Is Trigger : True.
  - Tag: portal01
- Portal Vermelho:
  - Component >> Physic2D>>PolygonCollider2D
  - Is Trigger : True.
  - Tag: portal02



Problema

- Observe que se o quadrado tocar no portal01 ele deve ser posicionado(teleportado) para a posição do portal02.
- Mas quando o personagem toca no portal02 ele é teleportado para o portal01, ou seja, temos um **loop infinito**.
- Para **resolver o problema** quando o personagem tocar no portal01 ele deve ser teleportado para uma posição a direita (x) de portal02, e quando o personagem tocar no portal02 ele deve ser teleportado para uma posição a esquerda de portal01.



Atributos

```
float Vx;
float Vy;
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;

    }
```

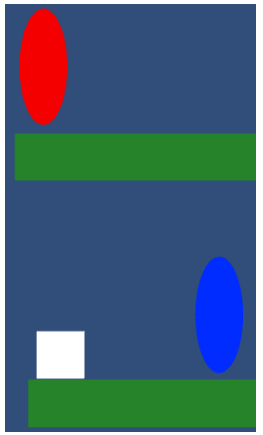
void Update ()

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

void OnTriggerEnter2D

```
void OnTriggerEnter2D(Collider2D objetoTriggerTocado){
    string tagTocadaTrigger = objetoTriggerTocado.gameObject.tag;
    if (tagTocadaTrigger == "portal01") {
        //recupera o vetor com a posição do elemento configurado com a tag "portal02"
        Vector2 posicaoDestino = GameObject.FindGameObjectWithTag ("portal02").GetComponent<Transform> ().position;
        //recupera o valor de x e y do destino do teleport.
        float xDestino = posicaoDestino.x;
        float yDestino = posicaoDestino.y;
        xDestino = xDestino + 1.5f;
        //cria um novo vetor de posição com (XDestino+1,yDestino)
        transform.position = new Vector2(xDestino,yDestino);
    }

    if (tagTocadaTrigger == "portal02") {
        Vector2 posicaoDestino = GameObject.FindGameObjectWithTag ("portal01").GetComponent<Transform> ().position;
        float xDestino = posicaoDestino.x;
        float yDestino = posicaoDestino.y;
        xDestino = xDestino - 1.5f;
        transform.position = new Vector2(xDestino,yDestino);
    }
}
```



É subtraído “-1.5f” pelo mesmo motivo e é somado +1.5f . O personagem será posicionado a esquerda do portal 01

é somando + 1.5f um ao valor de x do destino, como a posição de destino também é um portal se o personagem for teletransportado para exatamente a posição de outro portal, ele será teletransportado de novo e você ficará em um loop infinito

void OnCollisionEnter2D()

```
void OnCollisionEnter2D(Collision2D objetoTocado)
    {
        ContadorPulos = 0;
        string tagObjetoTocado = objetoTocado.gameObject.tag;

    }
```

void MovimentoHorizontalFlip()

```
void MovimentoHorizontalFlip(){
    DirecaoHorizontal = Input.GetAxis ("Horizontal");
    // 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;
    }
}
```

void PuloDuplo()

```
void PuloDuplo(){
    //verdadeiro(true) se o botão de Jump foi pressionado caso contrário igual a false
    bool apertou = Input.GetButtonDown ("Jump");
    // 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;
    }
}
```