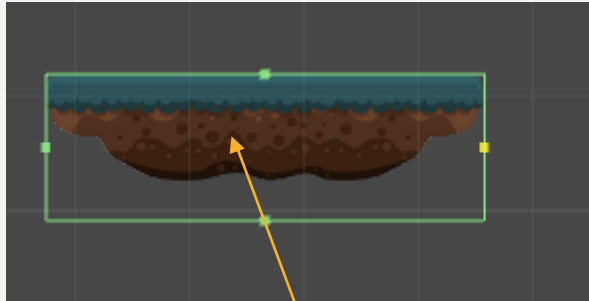


PLATAFORMAS

**PROF. ME.
HÉLIO
ESPERIDIÃO**



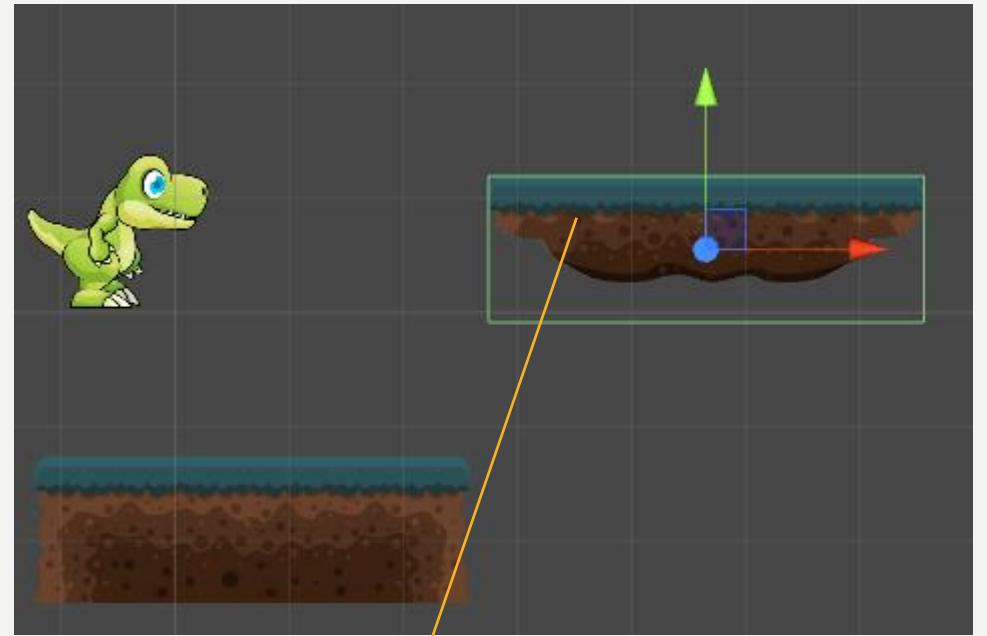
Box Collider 2D

Edit Collider

Material: None (Physics Material 2D)

Is Trigger:

Used By Effector:



Box Collider 2D

Edit Collider

Material: None (Physics Material 2D)

Is Trigger:

Used By Effector:

Used By Composite:

Auto Tiling:

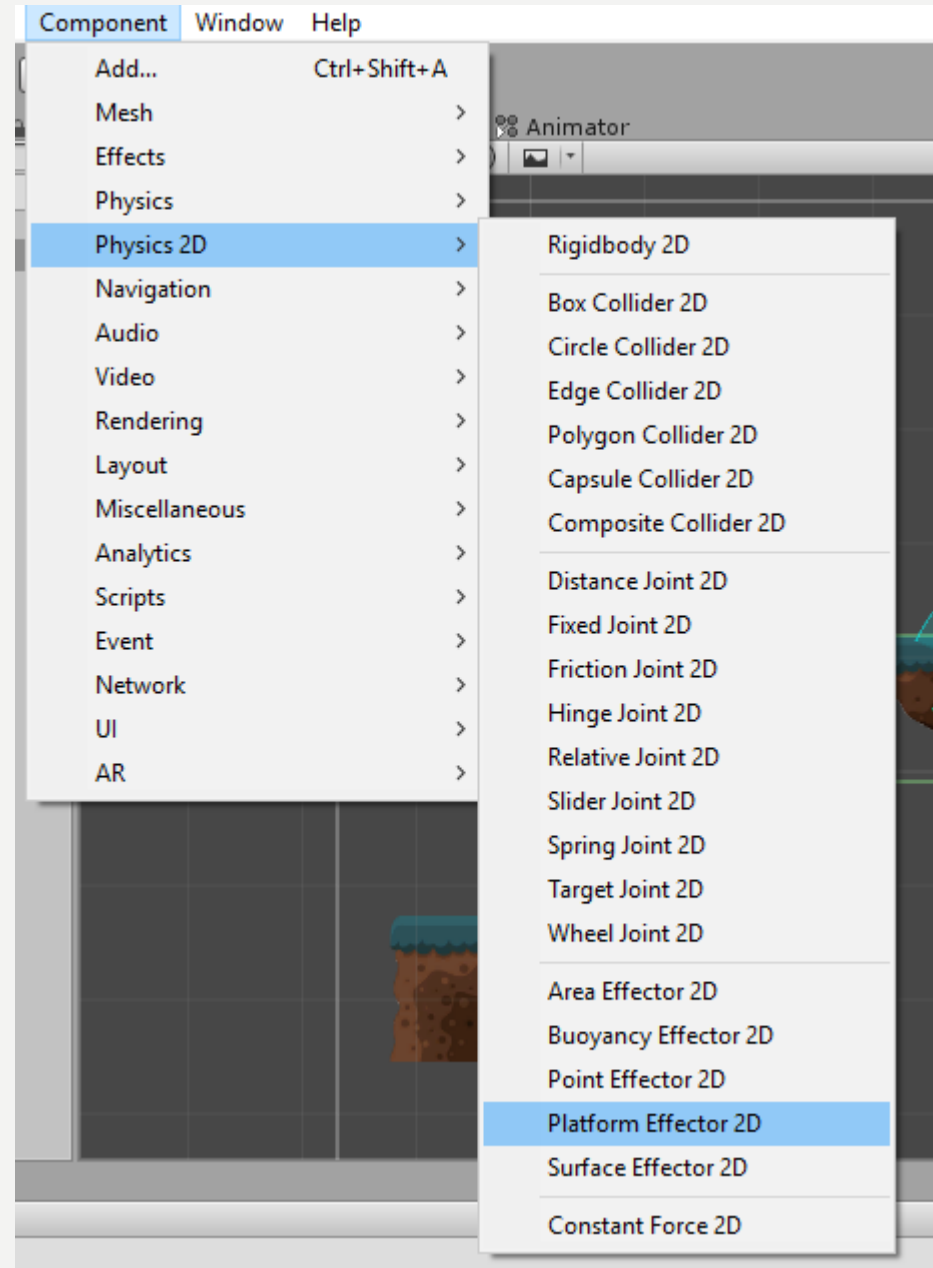
Offset: X 0 Y 0

Size: X 1.28 Y 1.28

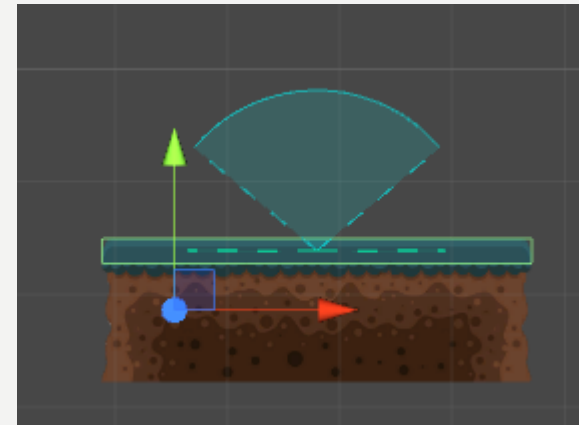
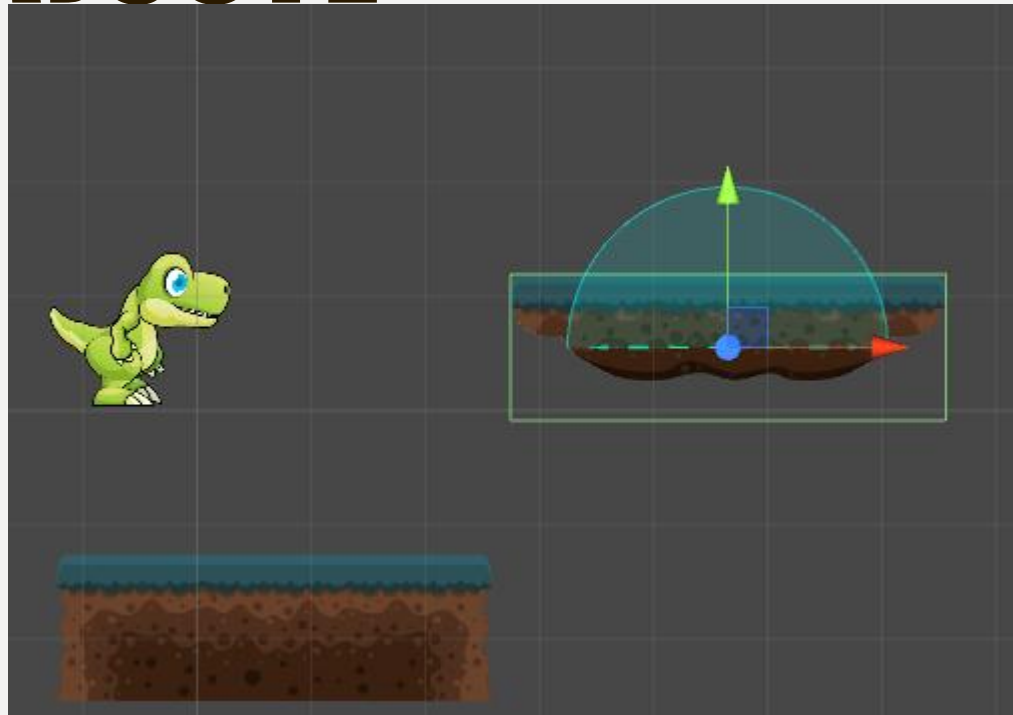
Edge Radius: 0

Info

- Adicione na plataforma:



AJUSTE



▼ Platform Effector 2D ? ⚙

Use Collider Mask

Collider Mask Everything ▾

Rotational Offset 0

▼ One Way

Use One Way

Use One Way Grouping

Surface Arc 120

► Sides

```
public class Plataforma : MonoBehaviour {
    float velocidadeMofimento = 5f;
    bool moverParaDireita = true;
    //início do movimento na posição de x = -4
    public float inicio = -4;
    //Término do movimento na posição de x = 4
    public float termino = 4;

    // Update is called once per frame
    void Update () {
        //caso passe da posição de termino inverte e volta
        if (transform.position.x > termino) {
            moverParaDireita = false;
        }
        //caso passe da posição de inicio inverte e retorna.
        if (transform.position.x < inicio) {
            moverParaDireita = true;
        }

        if (moverParaDireita == true) {
            //movimenta o personagem para a direita
            //é multiplicado pelo deltaTime para corrigir desempenho de placas de vídeo distinta,
            transform.position = new Vector2(transform.position.x + velocidadeMofimento * Time.deltaTime, transform.position.y);
        }else {
            transform.position = new Vector2(transform.position.x - velocidadeMofimento * Time.deltaTime, transform.position.y);
        }
    }
}
```

```
void MovimentoHorizontal(){
    //Recebe zero quando está parado até -1 quando move para
    //esquerda e até +1 quando move para direita.
    this.direcaoHorizontal = Input.GetAxis("Horizontal");
    //Multiplica a direção d movimento pela intensidade.
    float x = this.direcaoHorizontal * this.intensidadeMovimentoHorizontal;
    //recebe a velocidade em y no momento da movimentação.
    float y = personagemRigidbody2D.velocity.y;
    //Crie um vetor de velocidade com componentes x, y do plano.
    Vector2 movimento = new Vector2(x, y);
    //Adiciona o vetor de movimento ao corpo rígido do personagem.
    this.personagemRigidbody2D.velocity = movimento;
    //se o jogador está indo para a esquerda
    //espelhe o personagem e X
    if (this.direcaoHorizontal > 0) {
        this.personagemSpriteRenderer.flipX=false;
    }
    if (this.direcaoHorizontal < 0){
        this.personagemSpriteRenderer.flipX = true;
    }
}
```

```
//Executa quando o objeto ou o personagem toca
//em algum elemento com caixa de colisão
int qtdBonut = 0;
void OnCollisionEnter2D(Collision2D objeto) {

    this.contadorPulos = 0;
    if (objeto.gameObject.tag == "plataforma")
    {

        //o transform do personagem fica igual ao Transform da plataforma
        this.personagemRigidbody2D.transform.SetParent(objeto.transform);
    }

}

void OnCollisionExit2D(Collision2D objeto)
{

    if (objeto.gameObject.tag == "plataforma")
    {

        //Retira o transform do personagem da plataforma
        this.personagemRigidbody2D.transform.SetParent(null);
    }

}
```