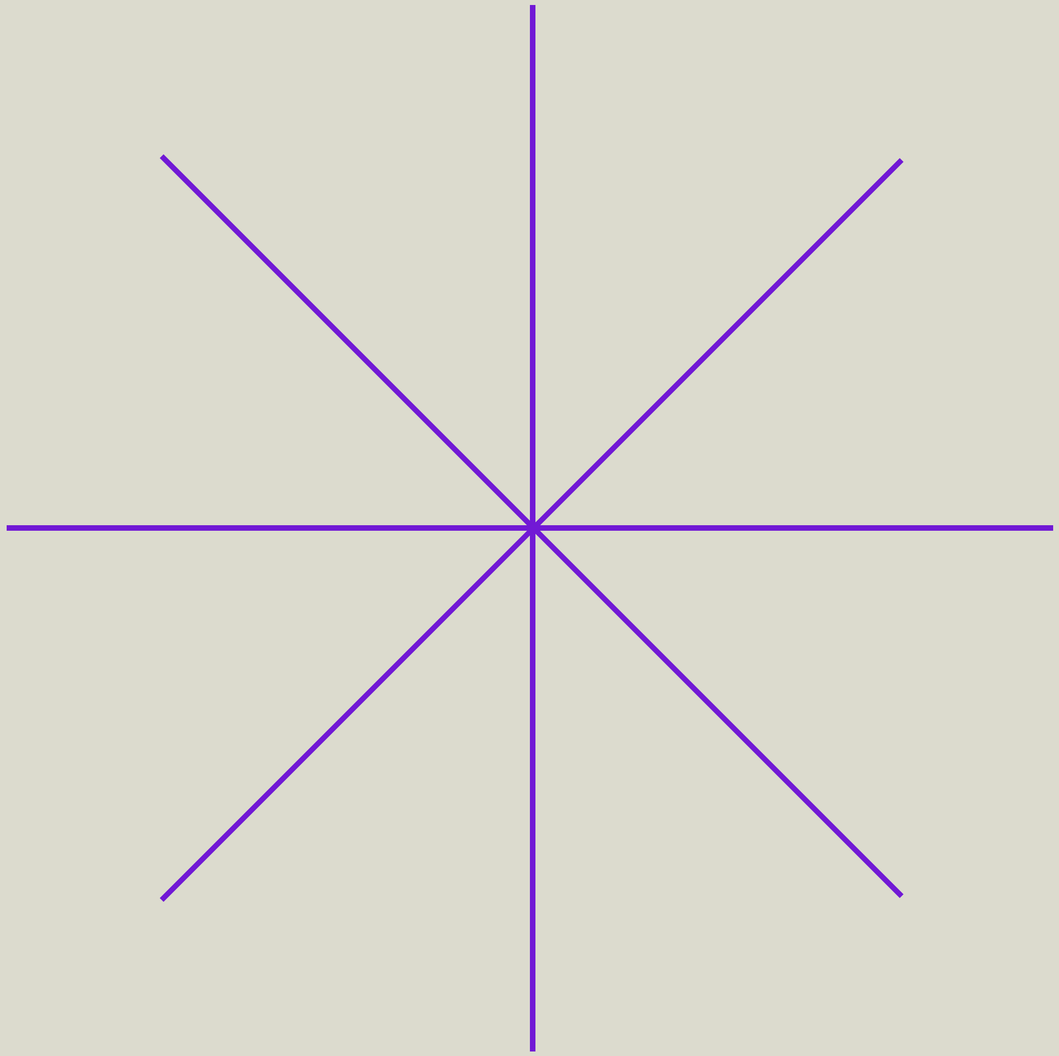
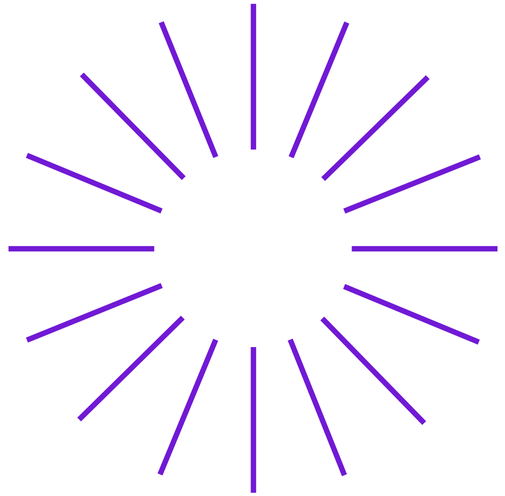
TECNOLOGIAS E ANIMAÇÃO DE JOGOS DIGITAIS



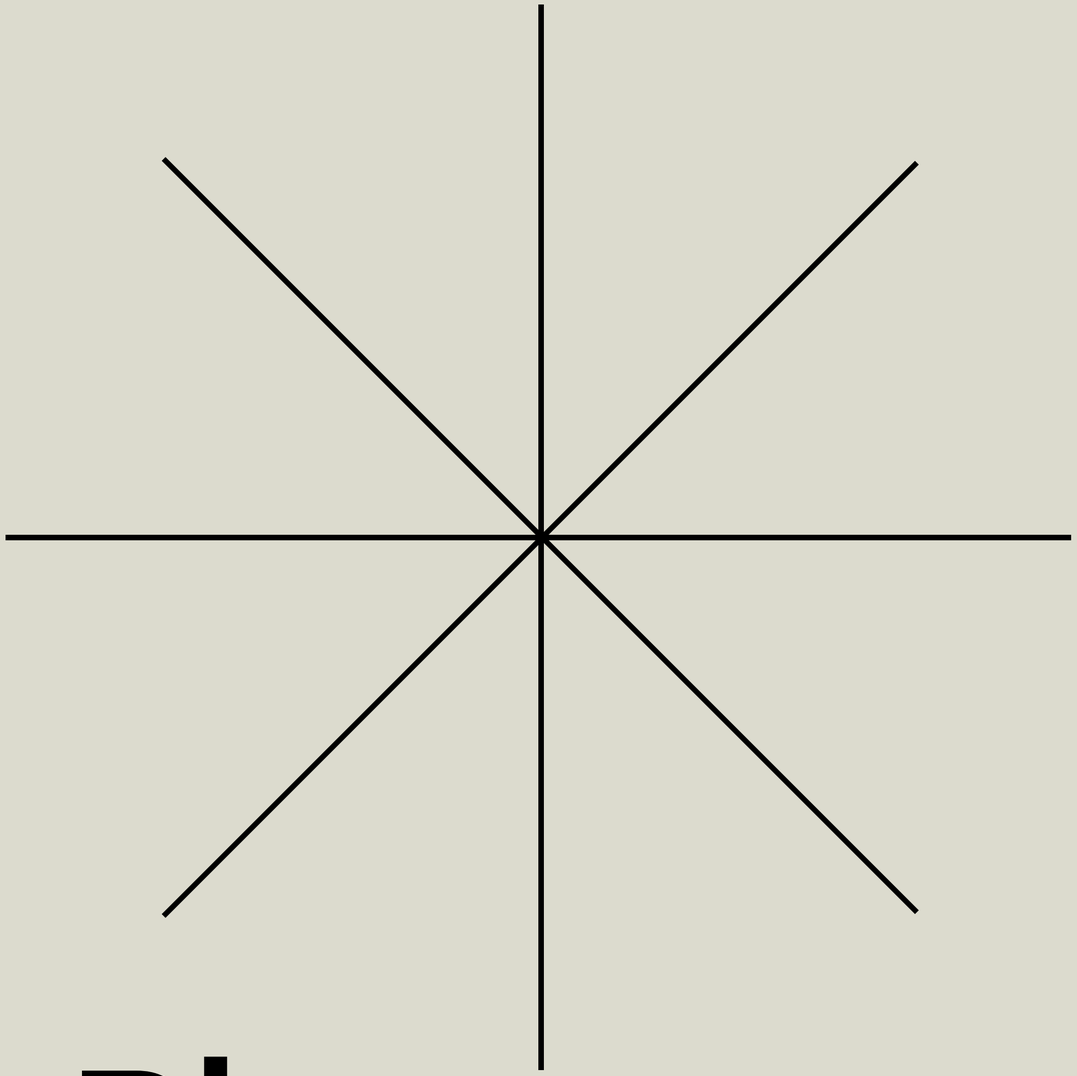
05. Space Invaders - Parte 2

DANIELA MONTEIRO

ÍNDICE



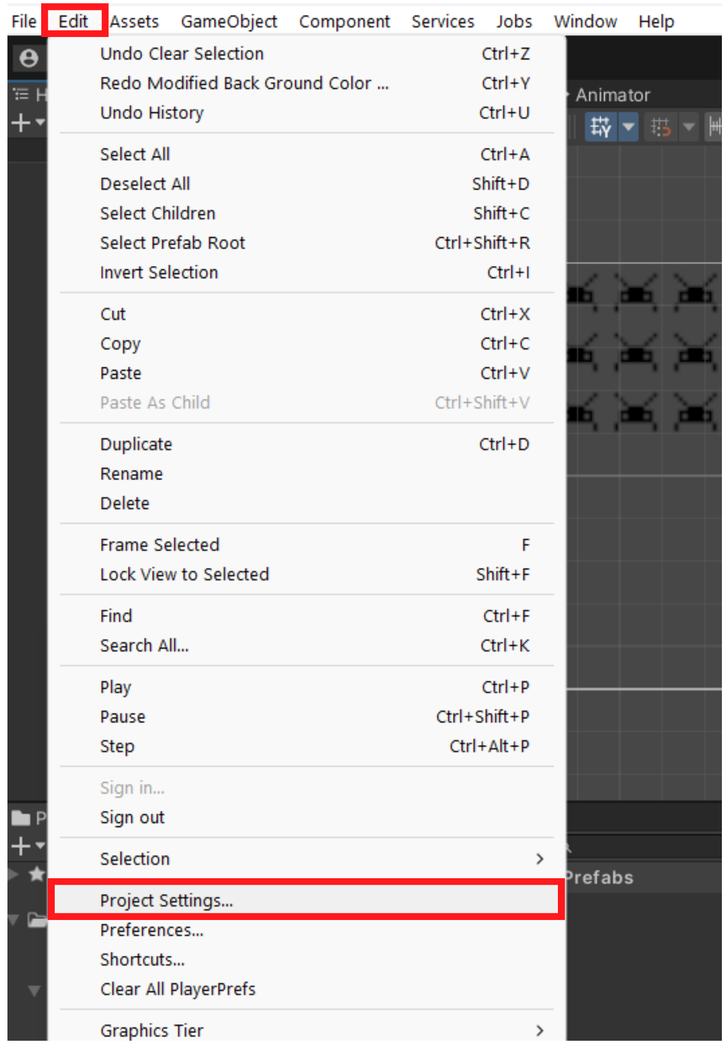
TECNOLOGIAS E ANIMAÇÃO DE JOGOS DIGITAIS

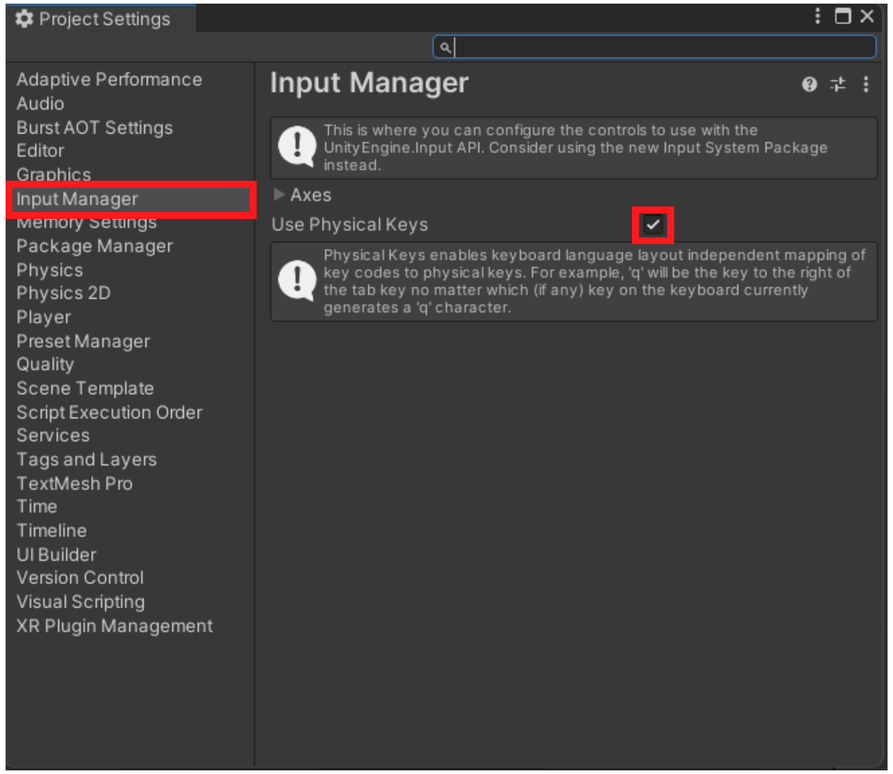


01
Movimento do Player

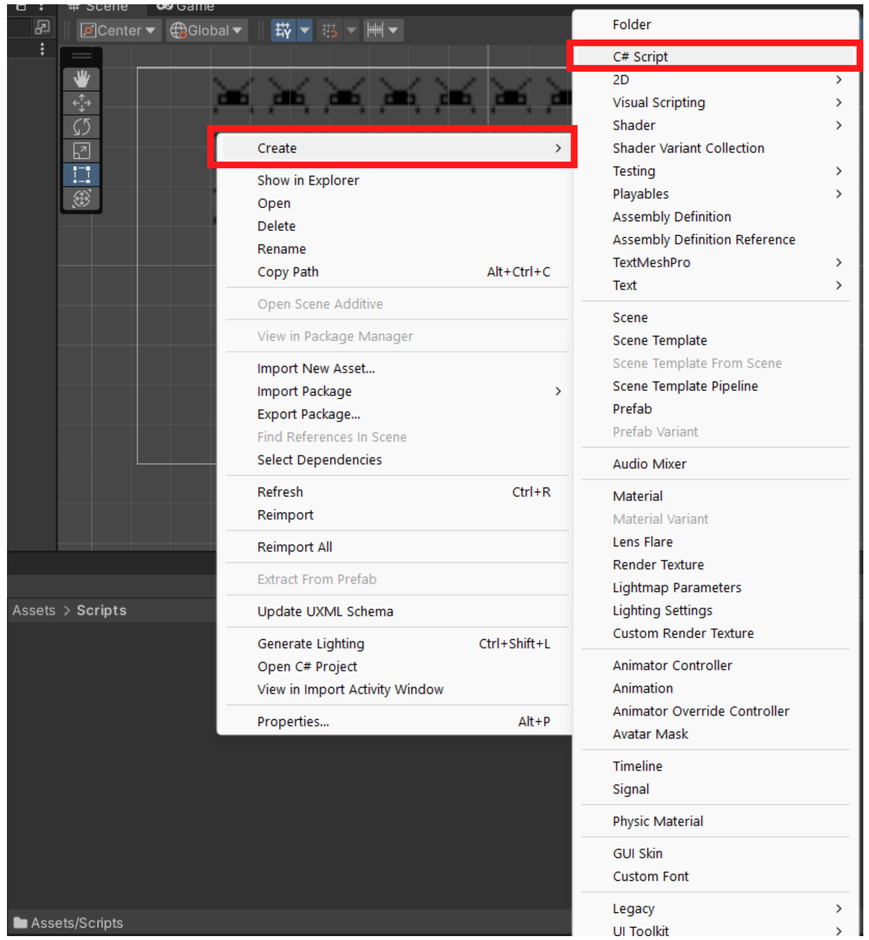
DANIELA MONTEIRO

PERMITE O USO DAS TECLAS



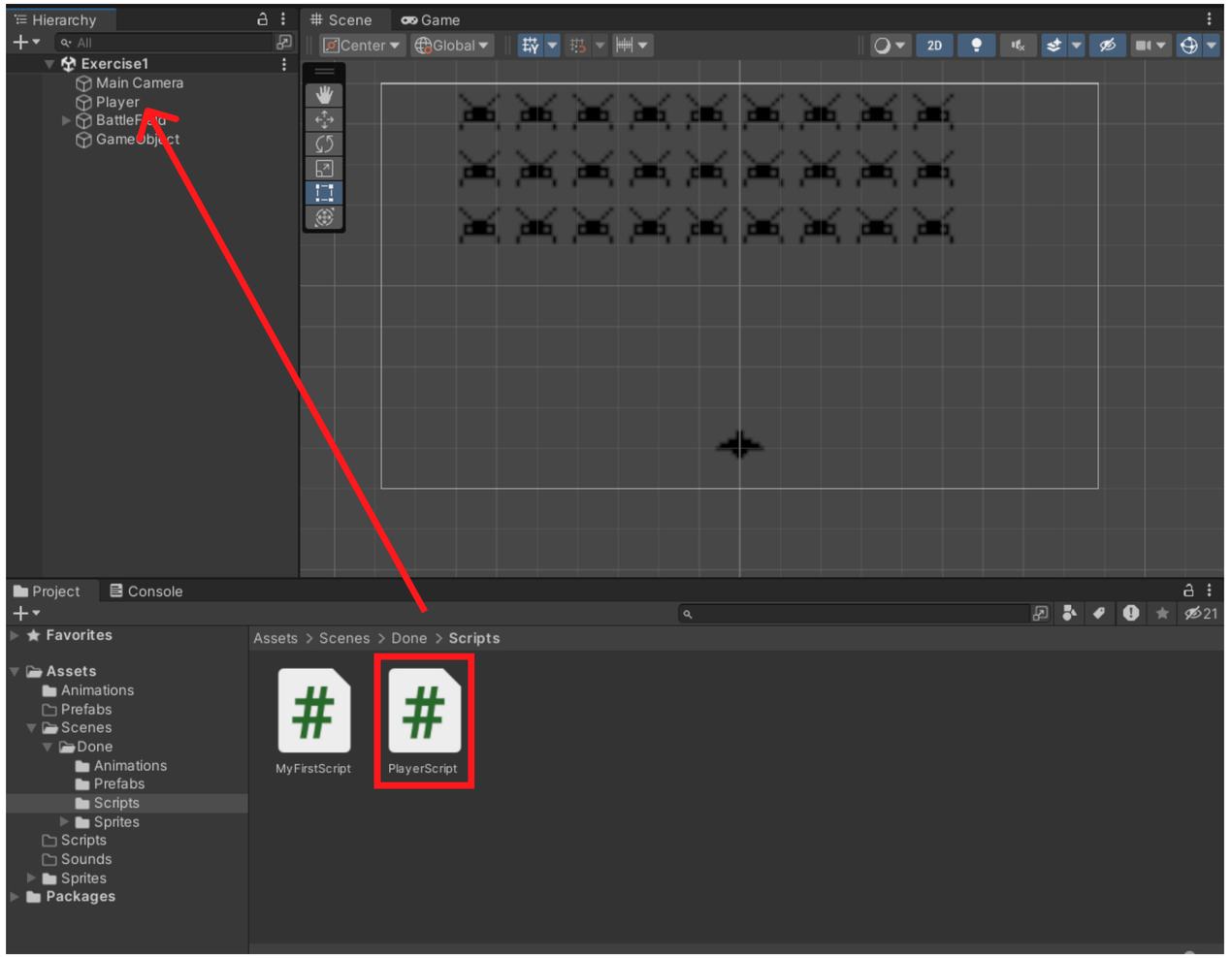


CRIA UM SCRIPT PARA O MOVIMENTO



Cria um Script C# chamado ‘PlayerScript’, na tua pasta ‘Scripts’.

ASSOCIA O SCRIPT AO TEU PLAYER



INICIALIZA AS VARIÁVEIS NECESSÁRIAS

```
public class PlayerScript : MonoBehaviour
{

    //inicializar as variaveis
    Animator playerAnim;
    [SerializeField]  float speed = 0.1f;
    Transform playerTransform;
```

Vão ser precisas as variáveis referentes a: o RectTransform do objeto a controlar (player); o animador do player e um valor para a velocidade.

INICIALIZA AS VARIÁVEIS NECESSÁRIAS

```
void Start()
{
    playerTransform = this.gameObject.transform;
    playerAnim = this.GetComponent<Animator>();
}
```

Às variáveis relativas ao jogador, necessitamos de as associar aos respetivos componentes.

CRIA UMA VARIÁVEL PARA O INPUT HORIZONTAL

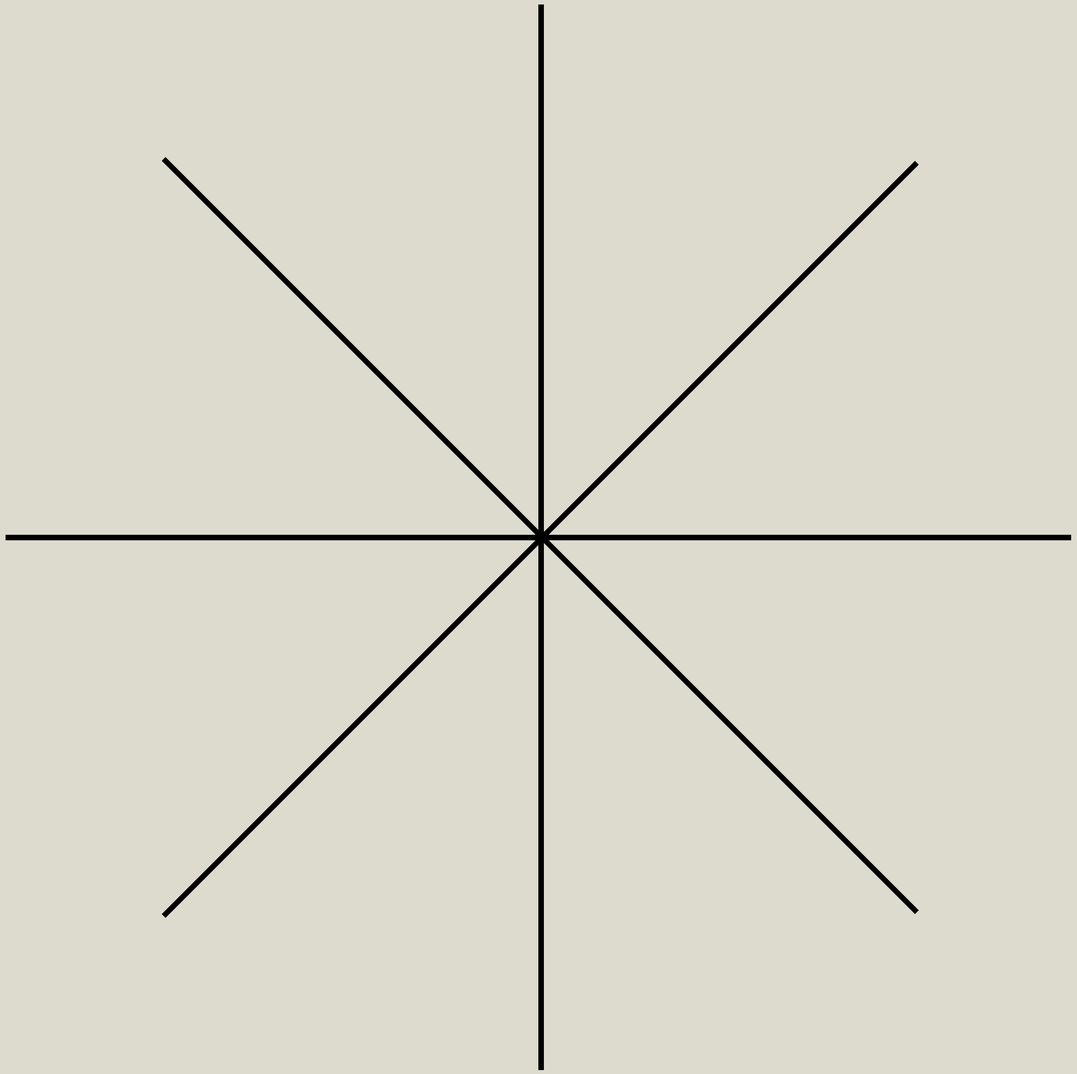
```
void Update()
{
    //buscar o input "setas horizontais" do jogador
    float horizontalInput = Input.GetAxis("Horizontal");
```

Cria uma variável do tipo ‘float’ dentro do método ‘Update’, para guardar o input horizontal das setas do teclado.

MUDA A POSIÇÃO DO PLAYER

```
void Update()
{
    //buscar o input "setas horizontais" do jogador
    float horizontalInput = Input.GetAxis("Horizontal");

    playerTransform.position =
        new Vector2(playerTransform.position.x + horizontalInput * speed,
        playerTransform.position.y);
}
```

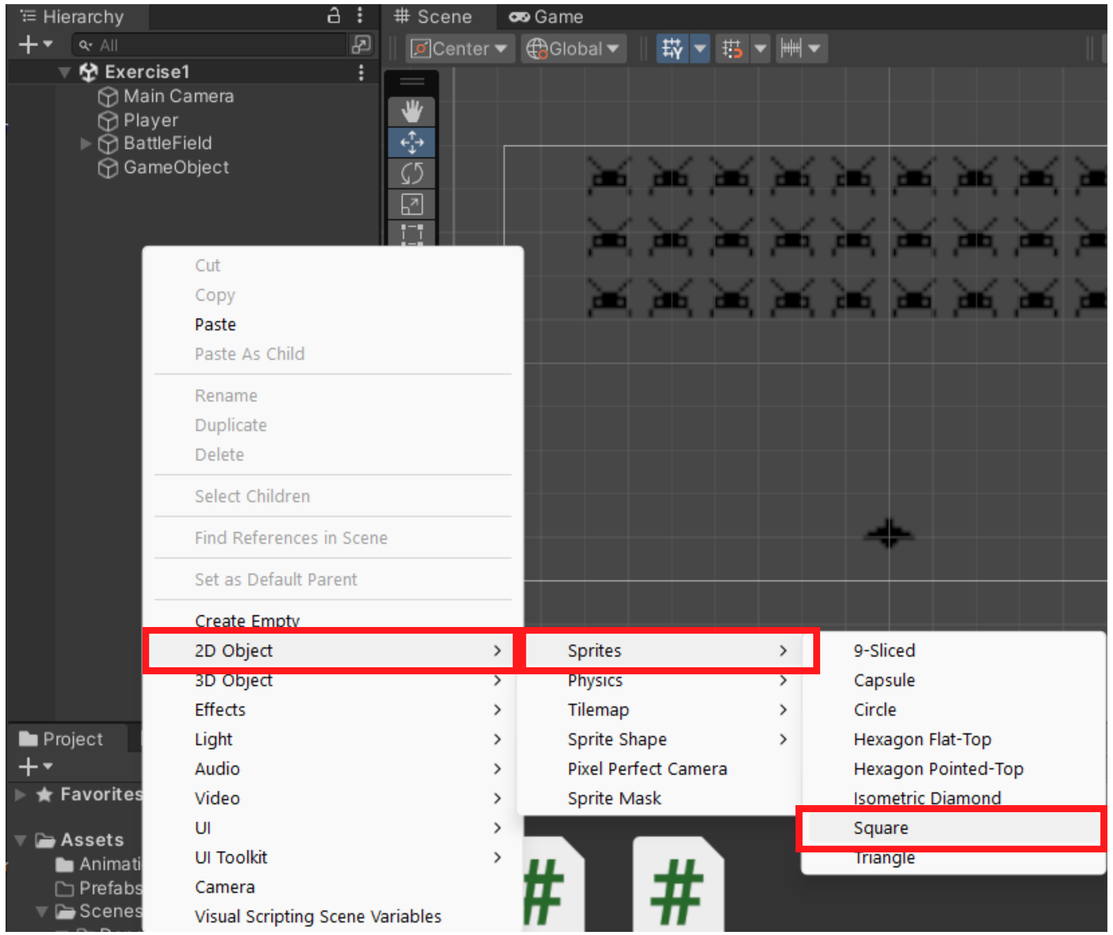
Muda a posição do playerTransform, mas apenas o eixo do x.



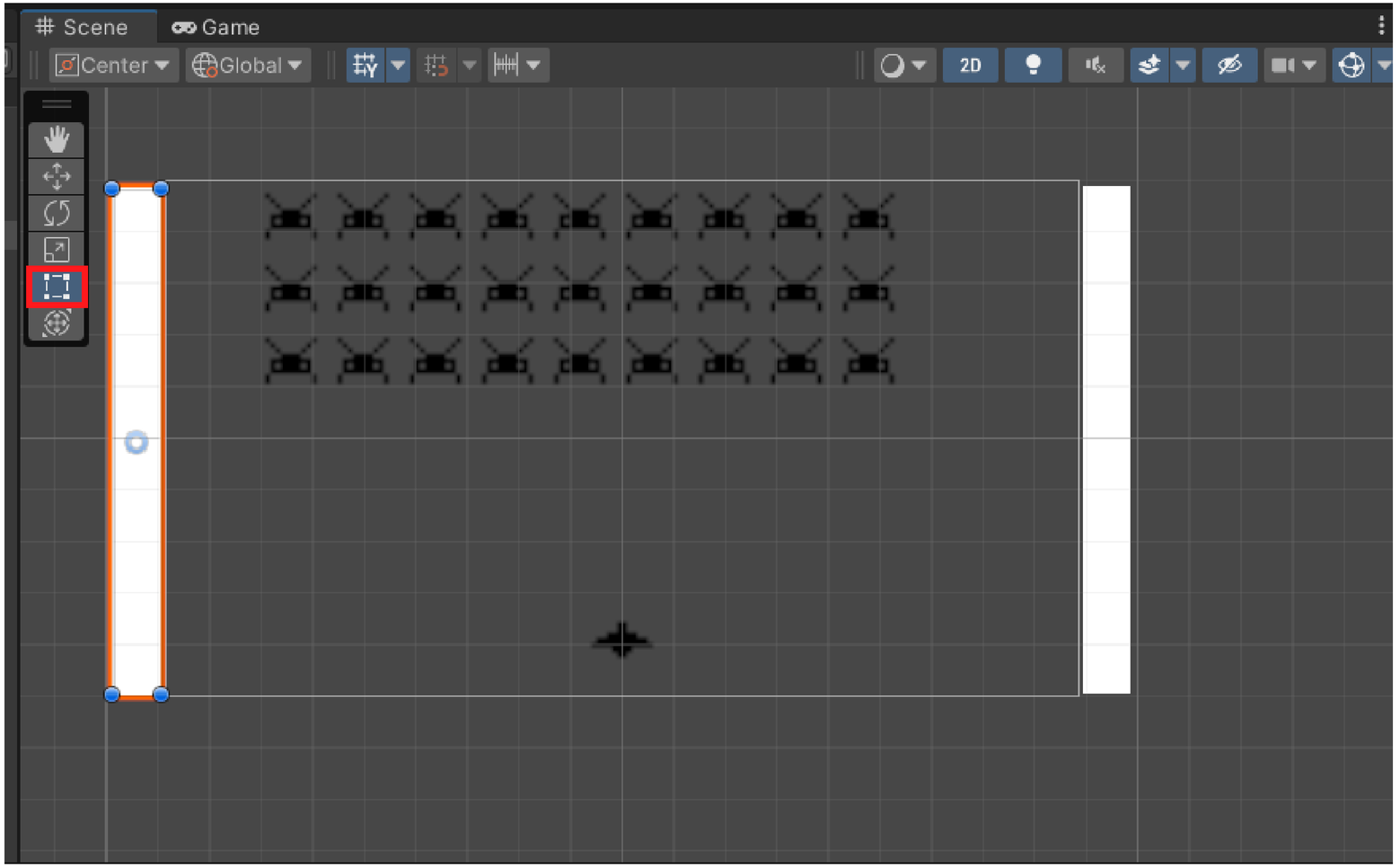
02
Barreiras de Jogo

DANIELA MONTEIRO

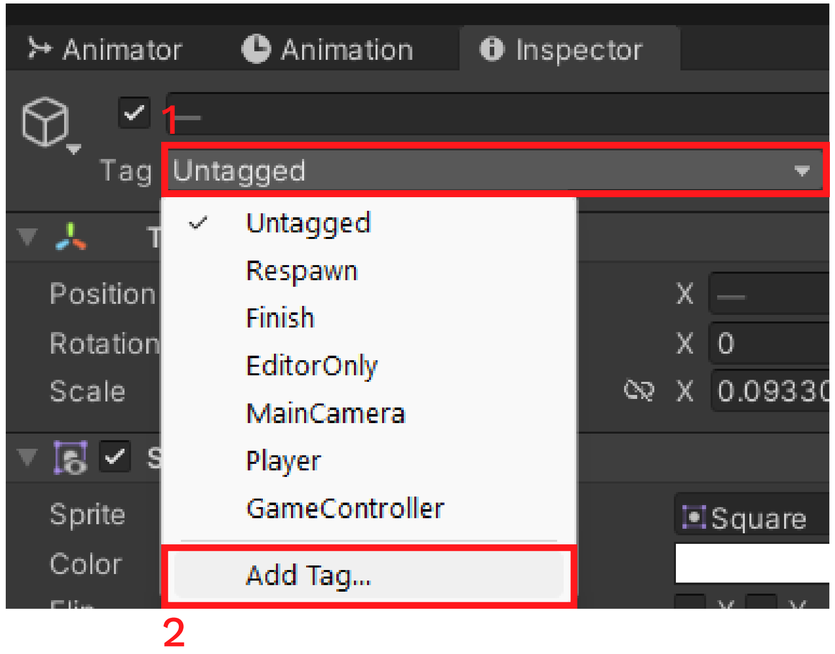
ADICIONA DOIS SPRITES QUADRADOS

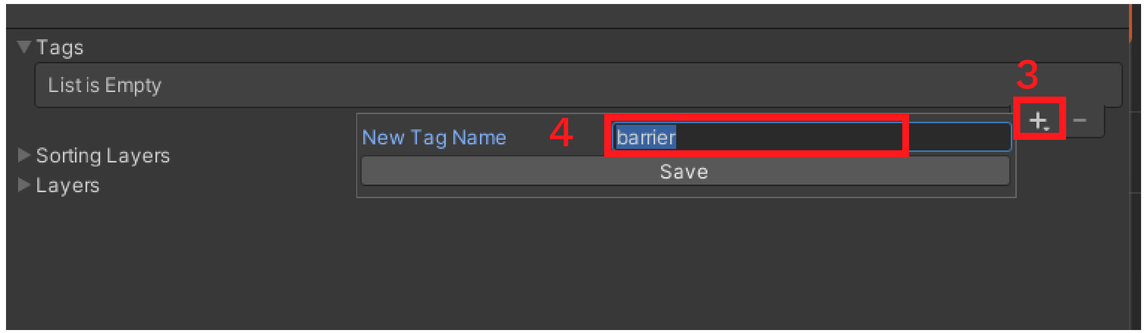


AJUSTA OS SEUS TAMANHOS

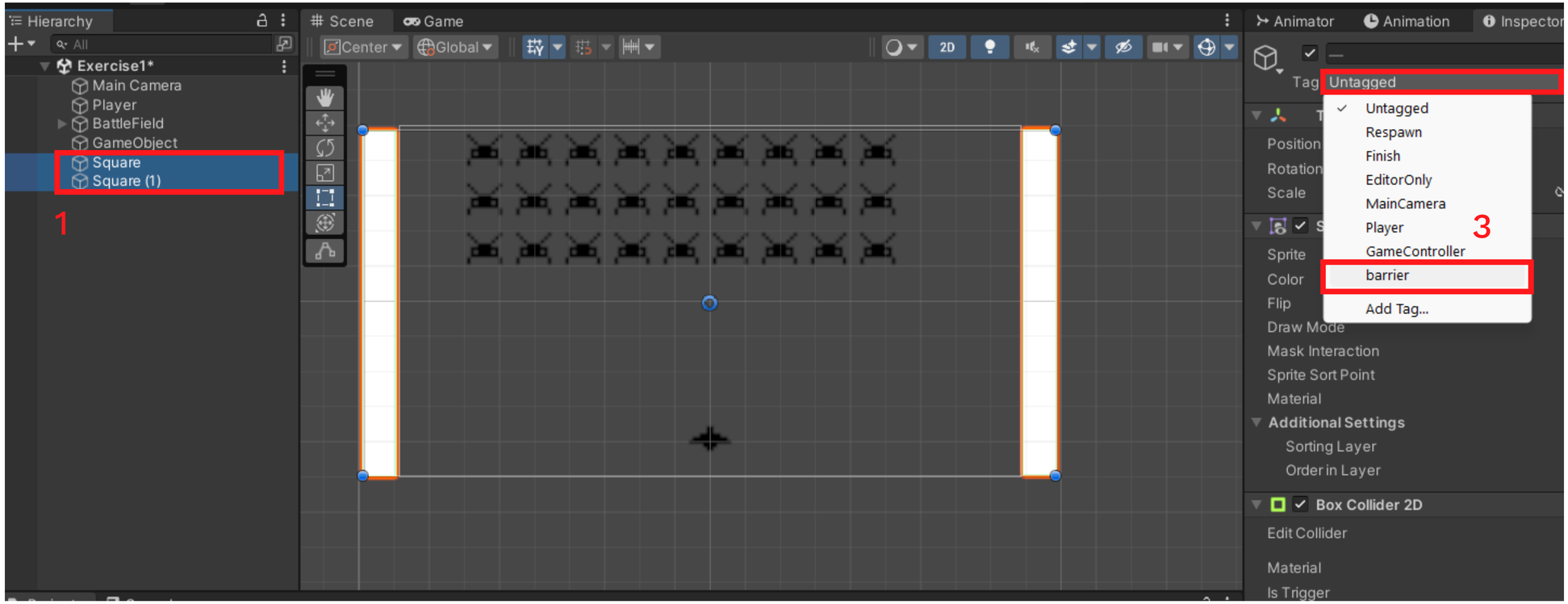


ADICIONA UMA TAG PARA AS BARREIRAS

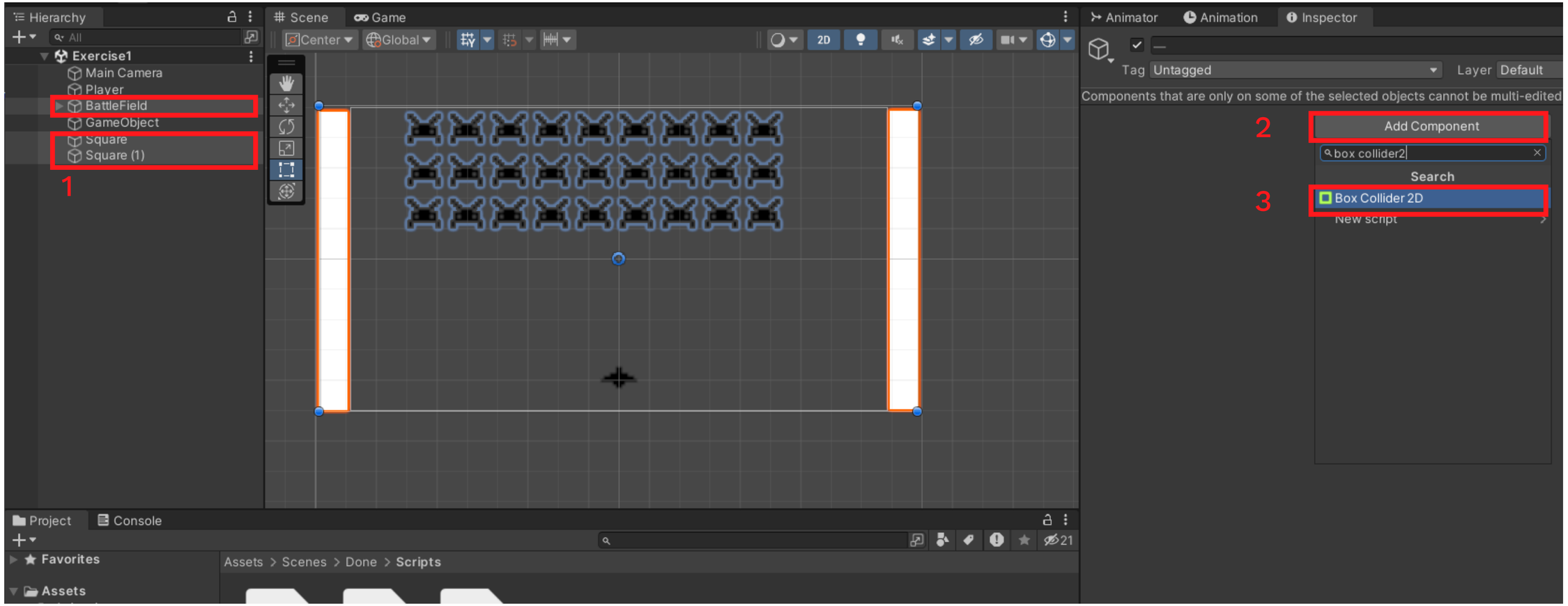




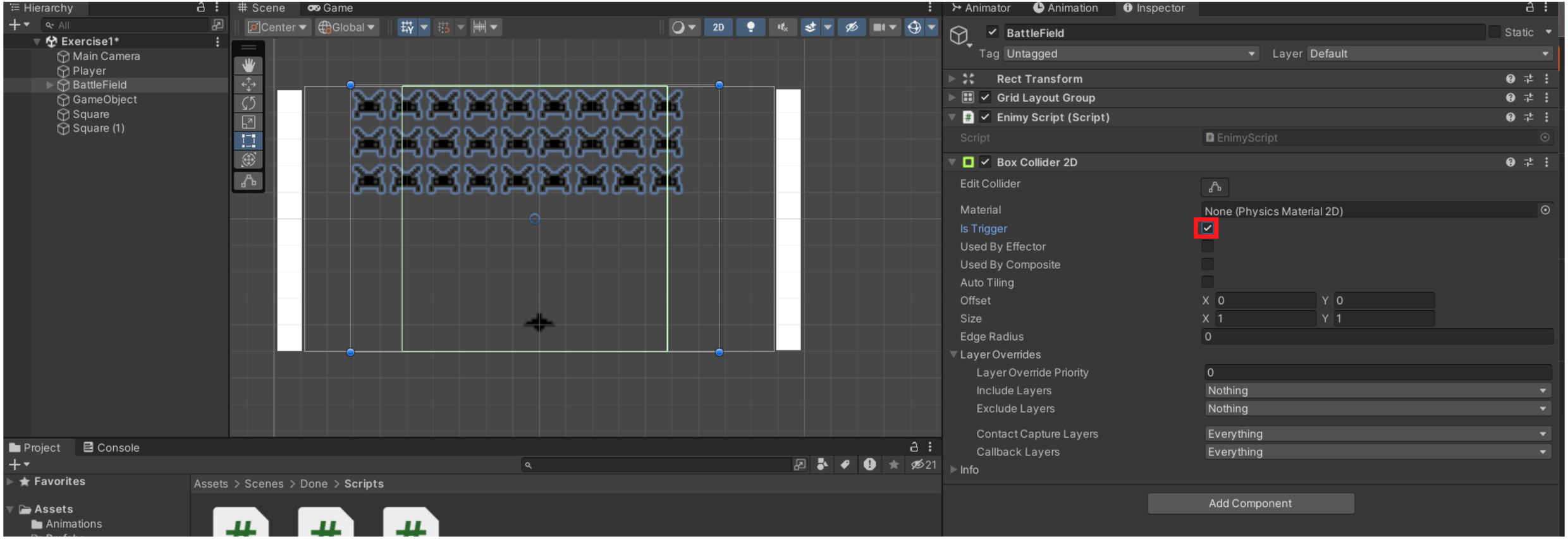
ADICIONA A TAG CRIADA ÀS BARREIRAS



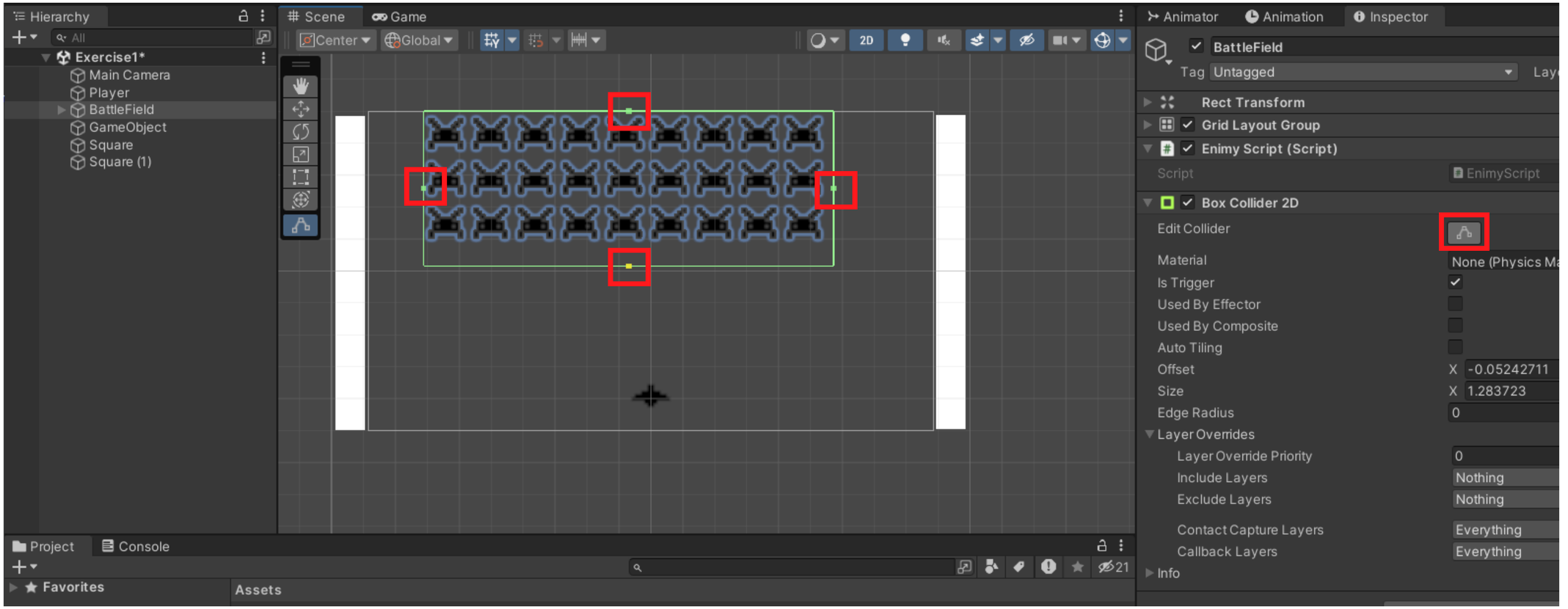
ADICIONA UM ‘BOX COLLIDER 2D’ NAS BARREIRAS E NO ‘BATTLEFIELD’



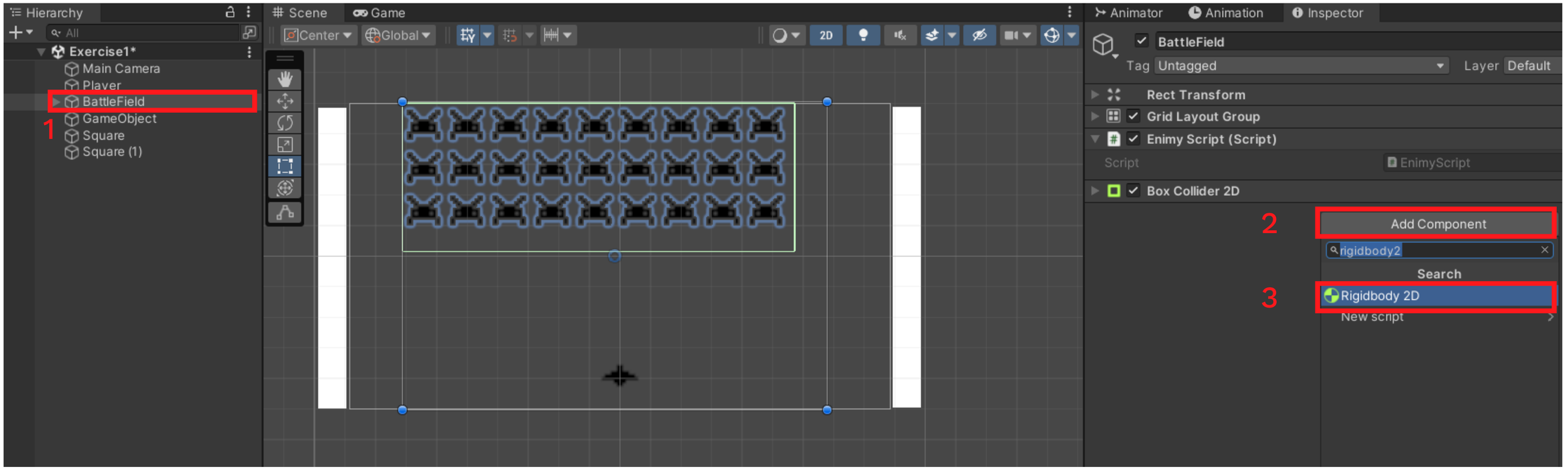
ACIONA A OPÇAO ‘IS TRIGGER’ DO COLLIDER DO ‘BATTLEFIELD’



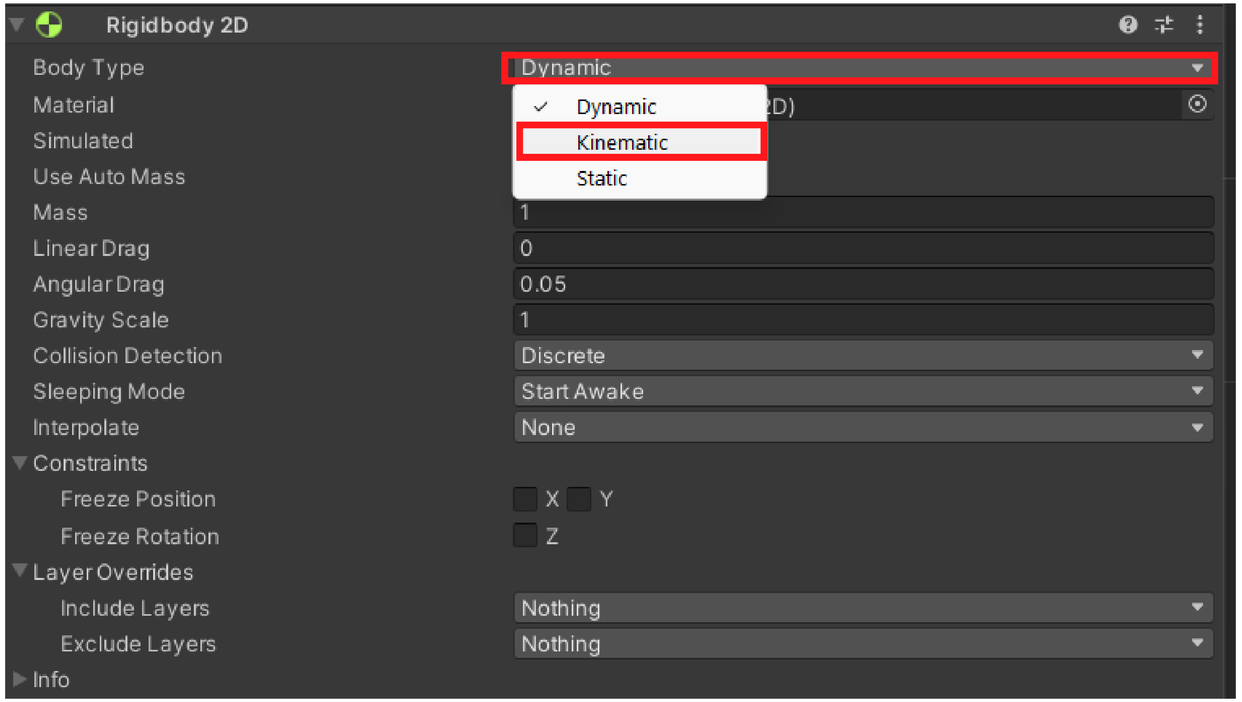
AJUSTA O TAMANHO DO BOX COLLIDER

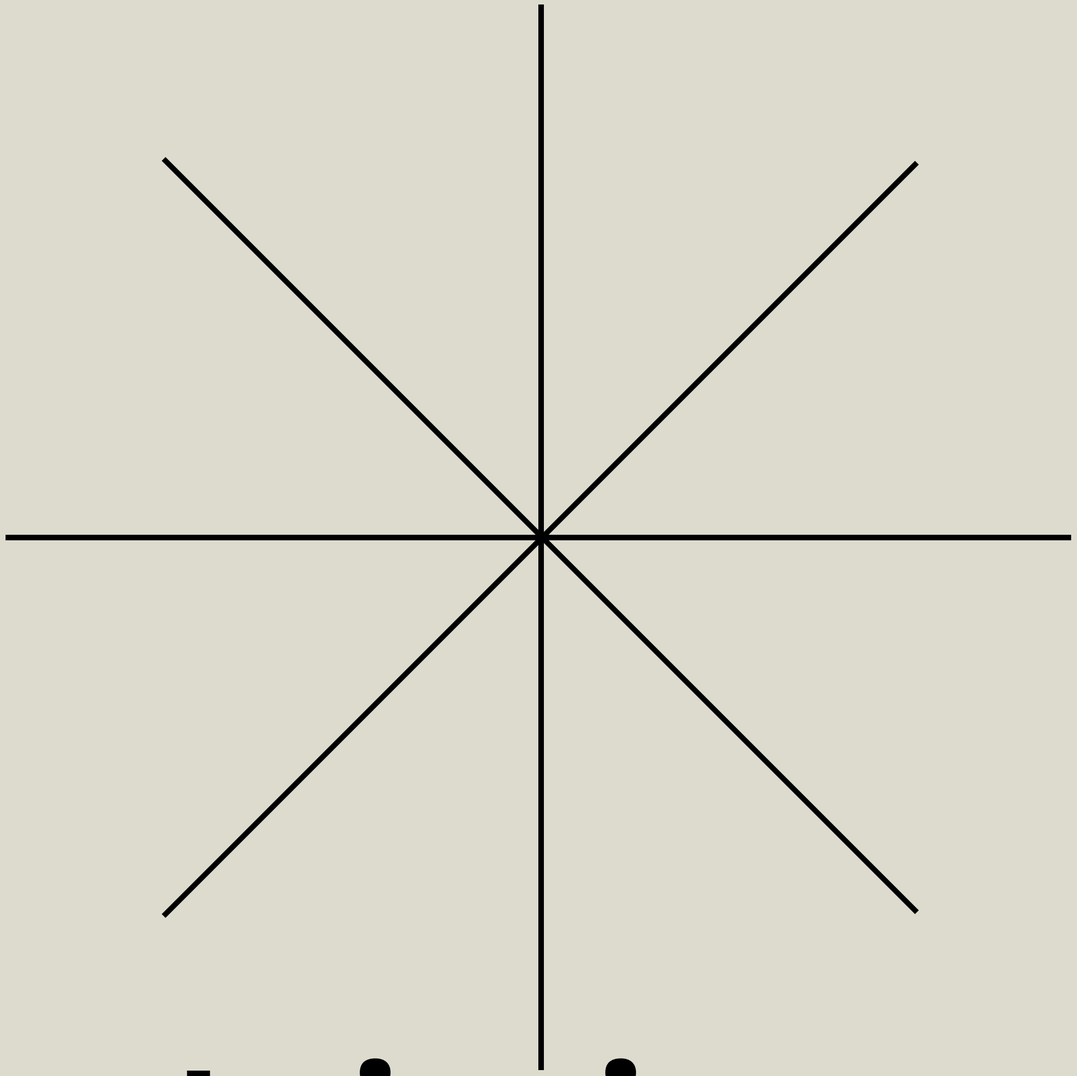


ADICIONA O COMPONENTE ‘RIGIDBODY2D’



MUDIFICA O BODY TYPE DO ‘BATTLEFIELD’ PARA ‘KINEMATIC’

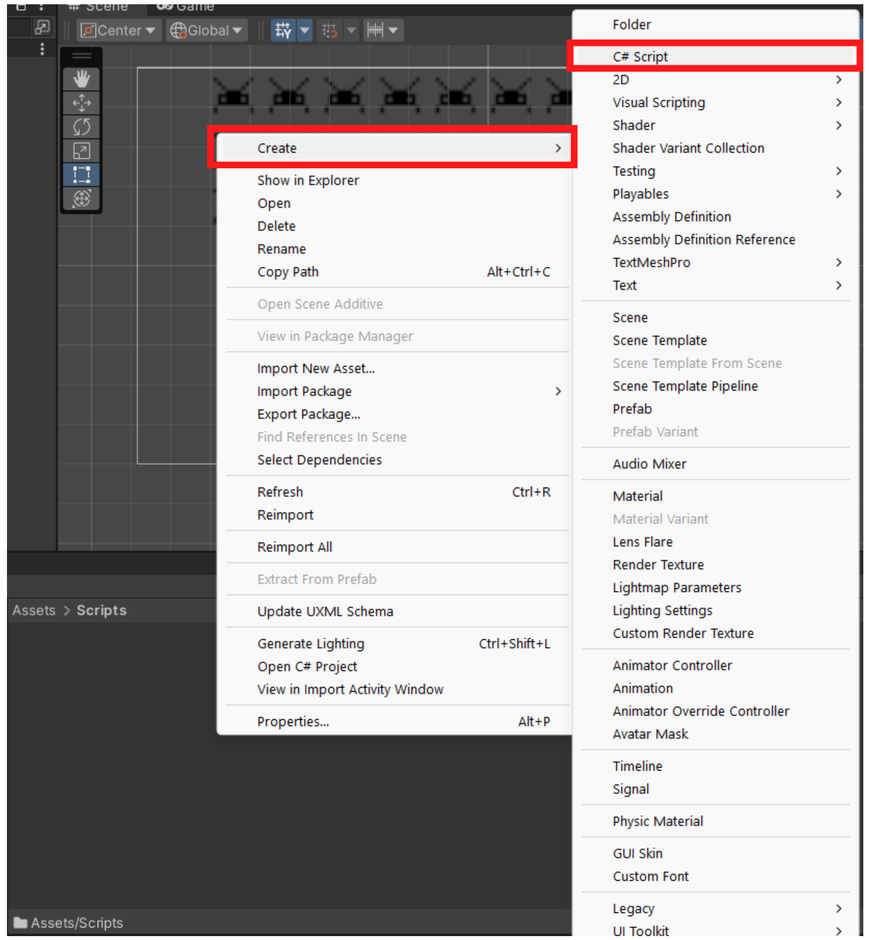




03
Movimento dos Inimigos

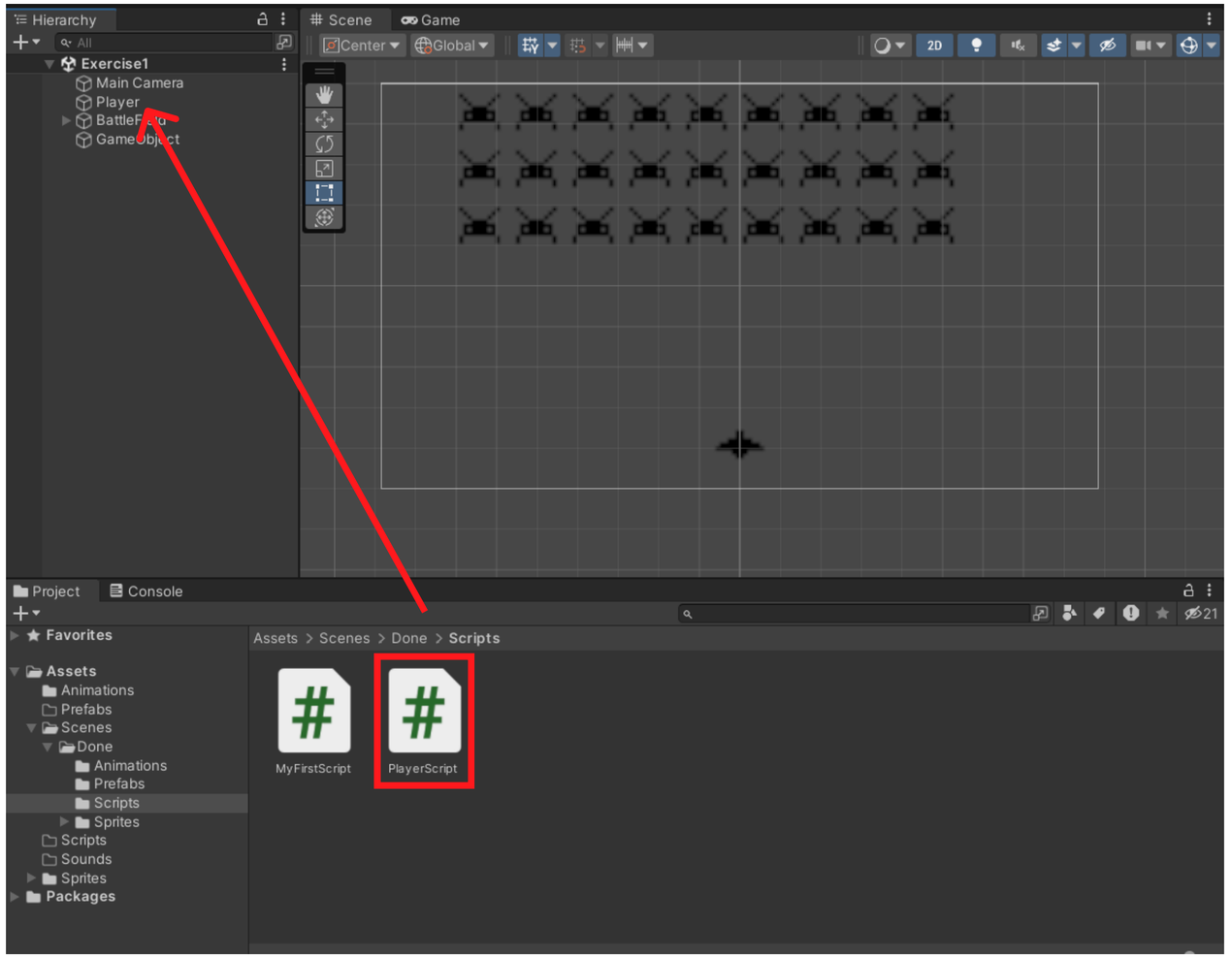
DANIELA MONTEIRO

CRIA UM SCRIPT PARA O MOVIMENTO DOS INIMIGOS



Cria um Script C# chamado ‘EnimyScript’, na tua pasta ‘Scripts’.

ASSOCIA O SCRIPT AO TEU ‘BATTLEFIELD’



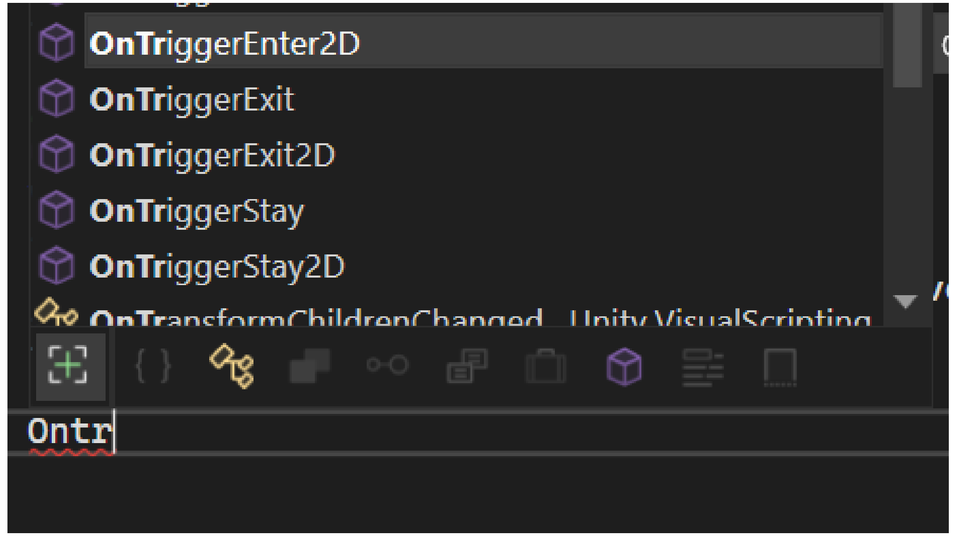
INICIALIZA AS VARIÁVEL RELATIVAS À VELOCIDADE E O SENTIDO DO MOVIMENTO

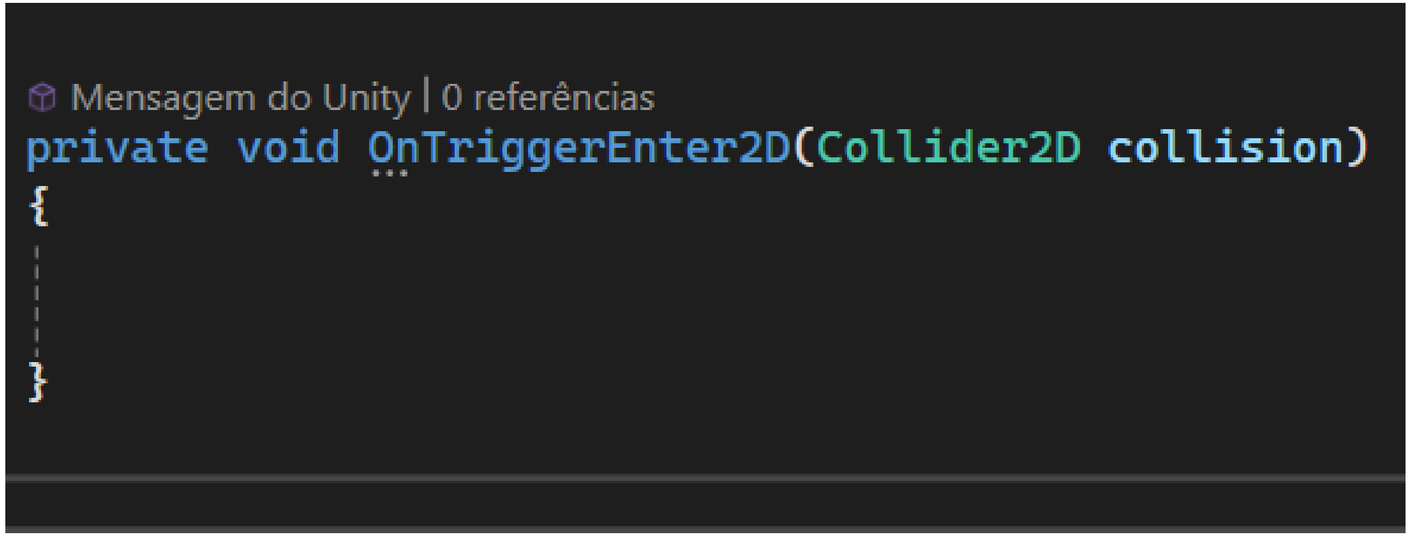
```
public class EnimyScript : MonoBehaviour
{
    float moveSpeed = 0.1f;
    float moveDirection = 1;
```

ADICIONA UMA TRANSLAÇÃO PARA A DIREITA

```
// Update is called once per frame
Mensagem do Unity | 0 referências
void Update()
{
    transform.Translate(Vector2.right * moveSpeed * moveDirection * Time.deltaTime);
}
```

ADICIONA UM METODO ‘ON TRIGGER ENTER 2D’



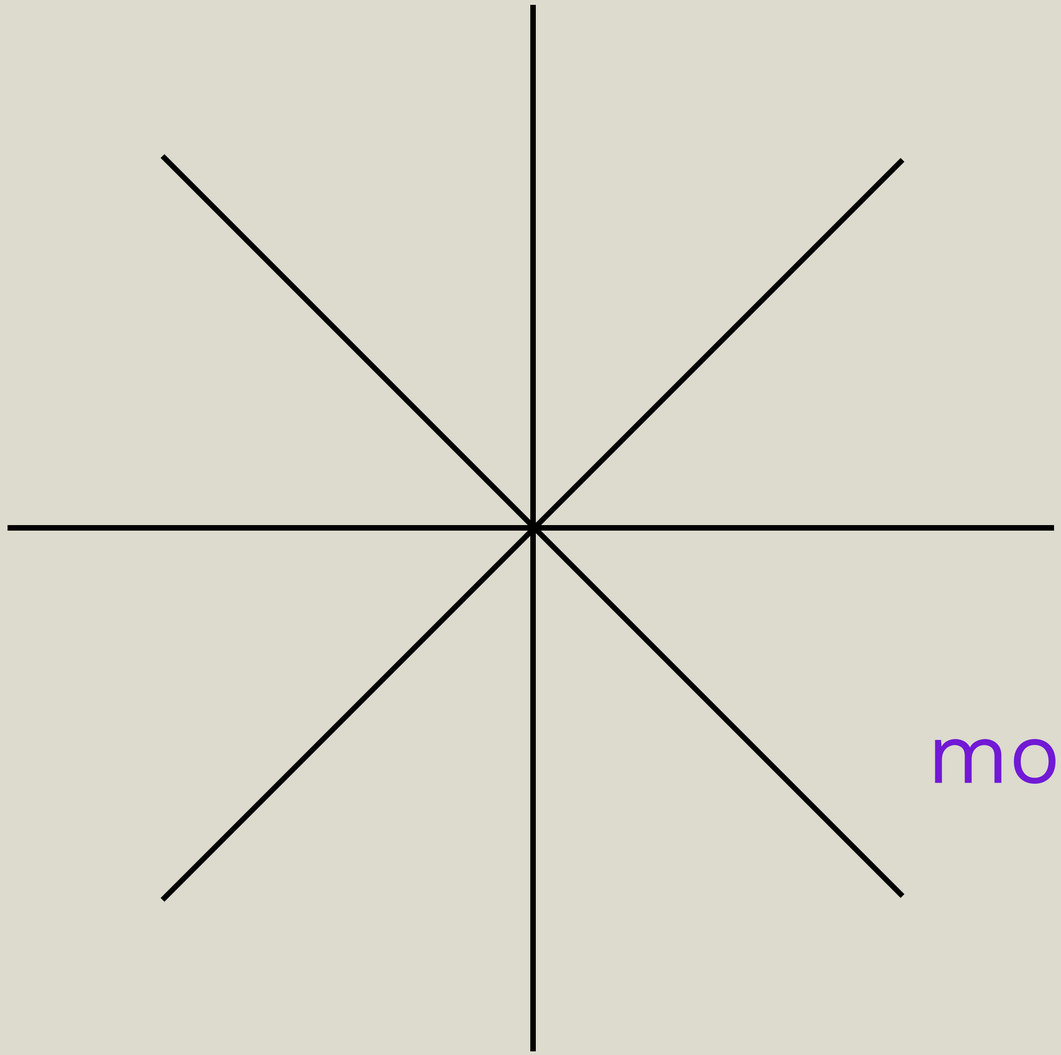


ADICIONA UMA CONDIÇÃO PARA TROCAR O SENTIDO DO MOVIMENTO

```
Mensagem do Unity | 0 referências
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.tag == "barrier") {
        moveDirection *= -1;
    }

}
```

MUDA A POSIÇÃO DO Y PARA O MOVIMENTO SE SEGUIR UM POUCO MAIS EMBAIXO

```
//quando colide com uma barreira
if (collision.tag == "barrier") {

    //o sentido inverte
    moveDirection *= -1;

    //a posição y vai um pouco mais para baixo
    transform.position = new Vector2(transform.position.x, transform.position.y - 1);
}
```



Obrigada!

Não te esqueças onde encontrar este ppt:

motamdaniela.github.io/tajd