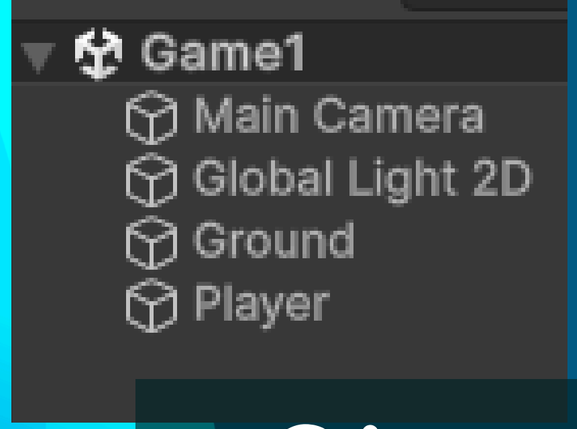
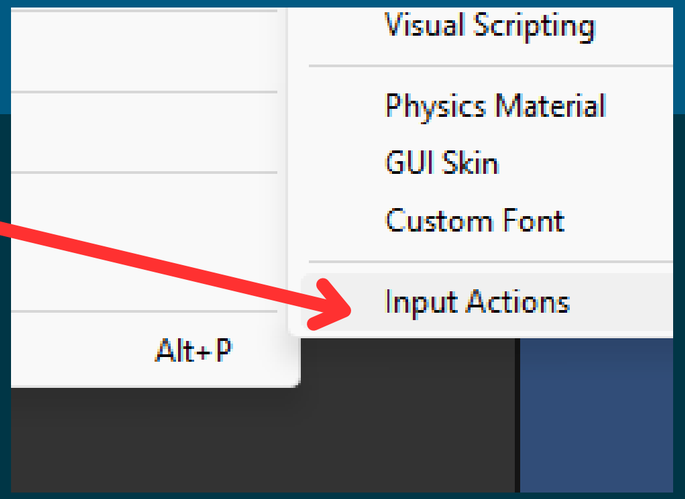
Mini Jogo 2D

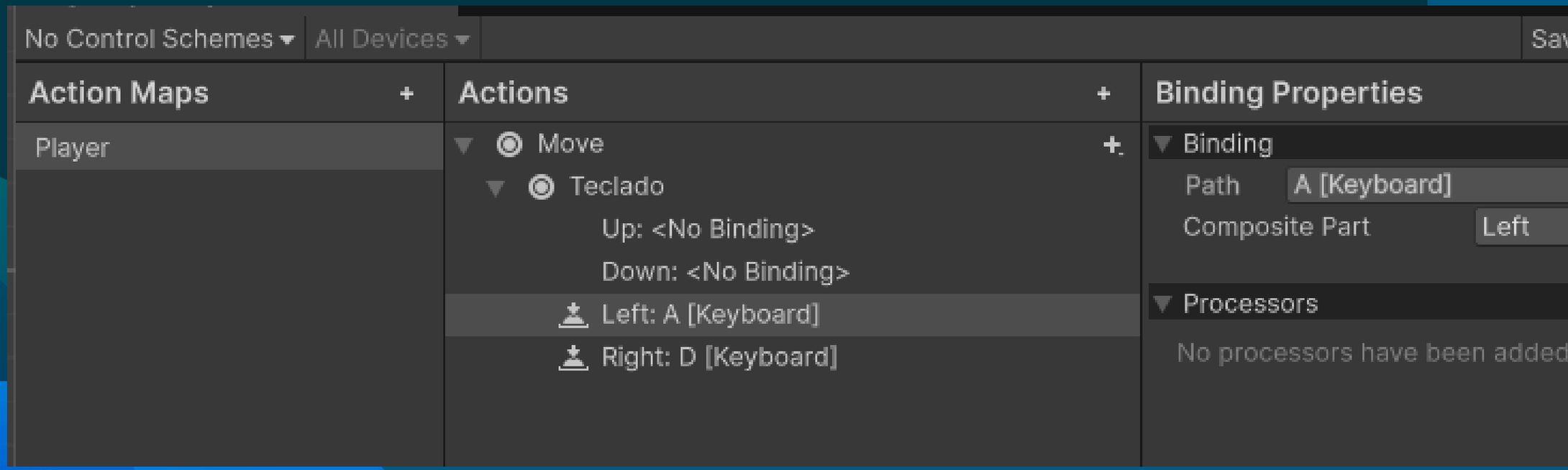


Crie o projeto 2d Chamado MiniJogo2D

- Crie uma Cena Chamada Game1
- Faça um GameObjeto Ground (Crie Tag “ground” e coloque no objeto Ground)
- Box collider no ground
- Crie um Game objeto Capsule renomeie para Player com um rig, collider Capsule e Script Chamado “MovePlayer”

Crie Input Actions com nome PlayerInput





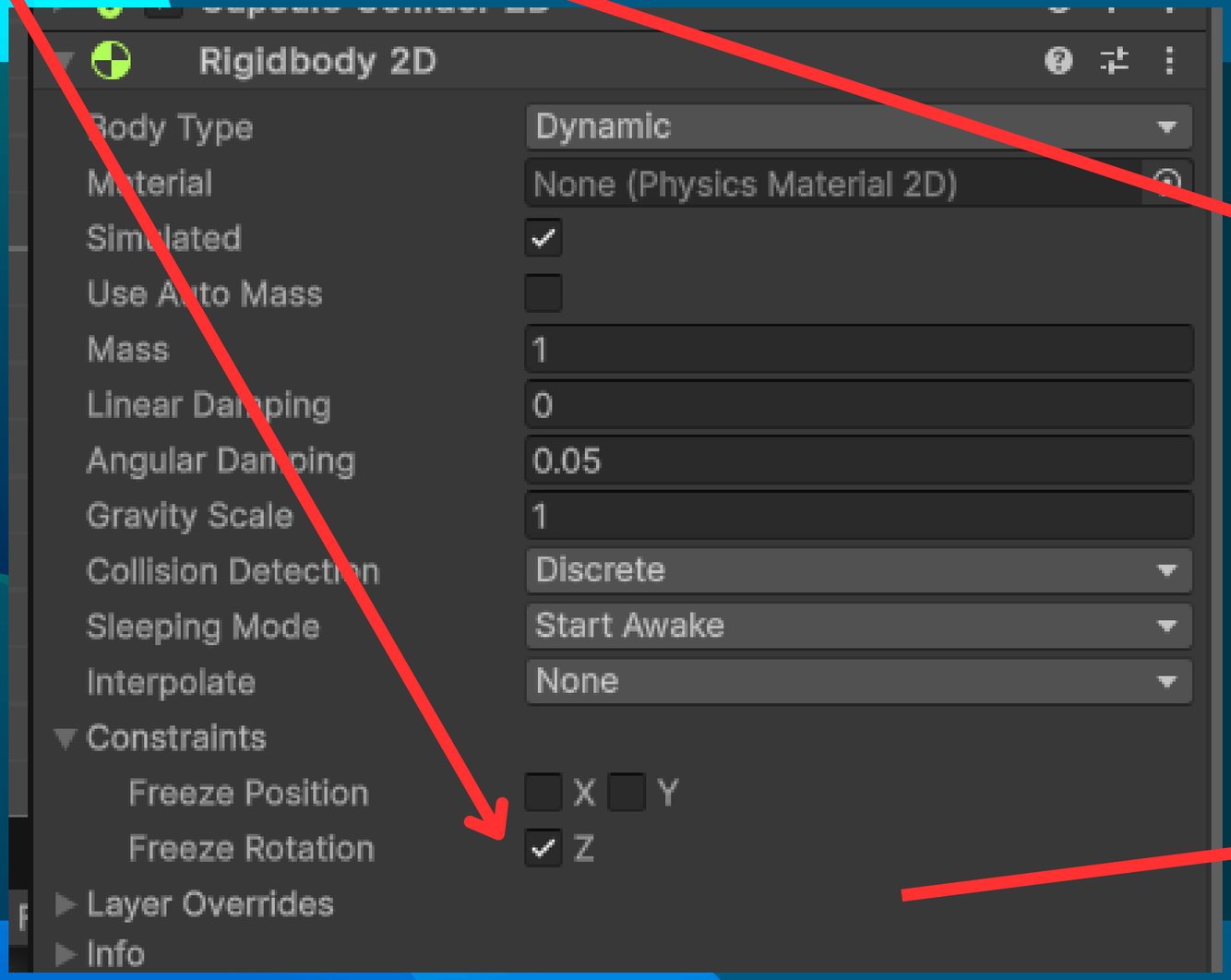
Programação de movimento

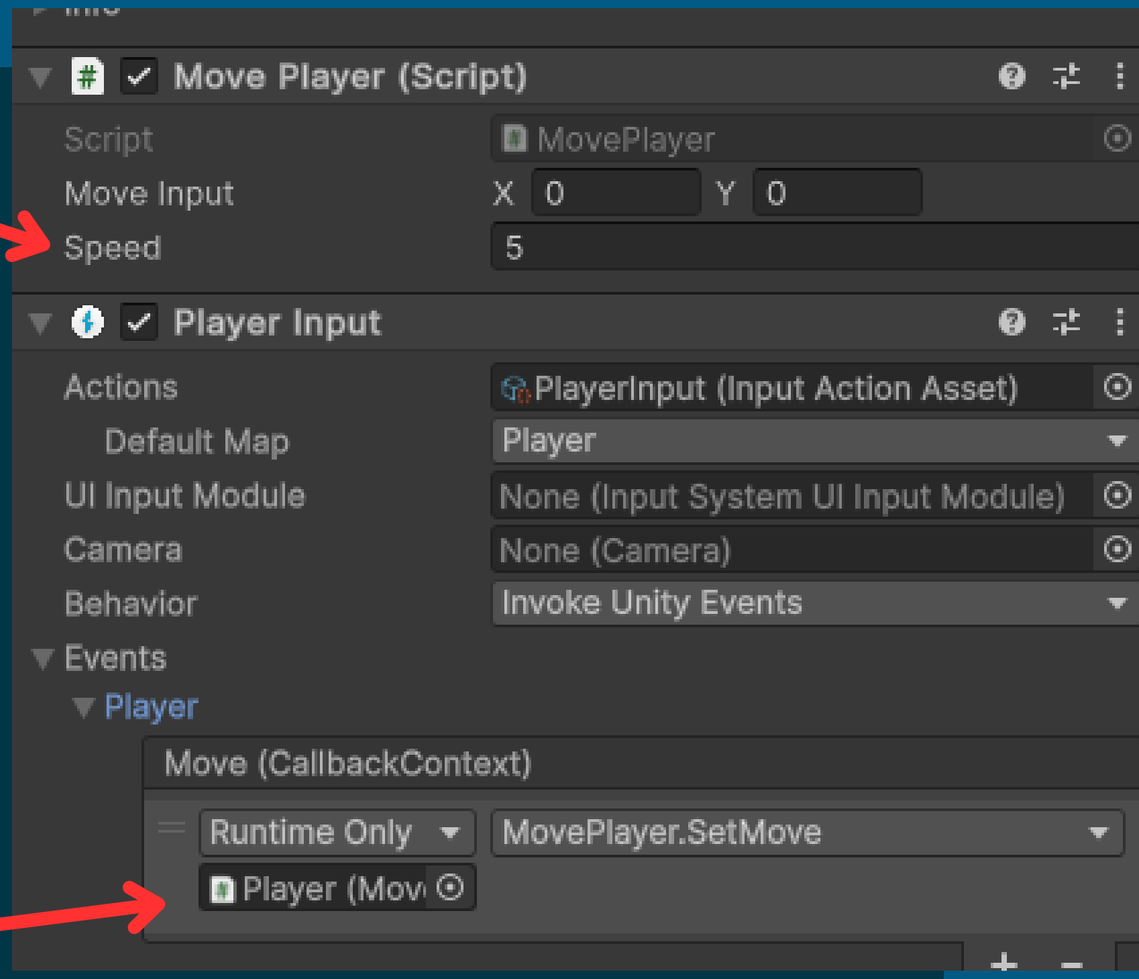
```
public class MovePlayer : MonoBehaviour
{
    Rigidbody2D _rb;
    [SerializeField] Vector2 _moveInput;
    [SerializeField] float _speed;
    Mensagem do Unity | 0 referências
    void Start()
    {
        _rb=GetComponent<Rigidbody2D>();
    }

    // Update is called once per frame
    Mensagem do Unity | 0 referências
    void Update()
    {
        _rb.linearVelocity = new Vector2(_moveInput.x* _speed, _rb.linearVelocity.y);
    }

    0 referências
    public void SetMove(InputAction.CallbackContext value)
    {
        _moveInput = value.ReadValue<Vector2>();
    }
}
```

Configure Player

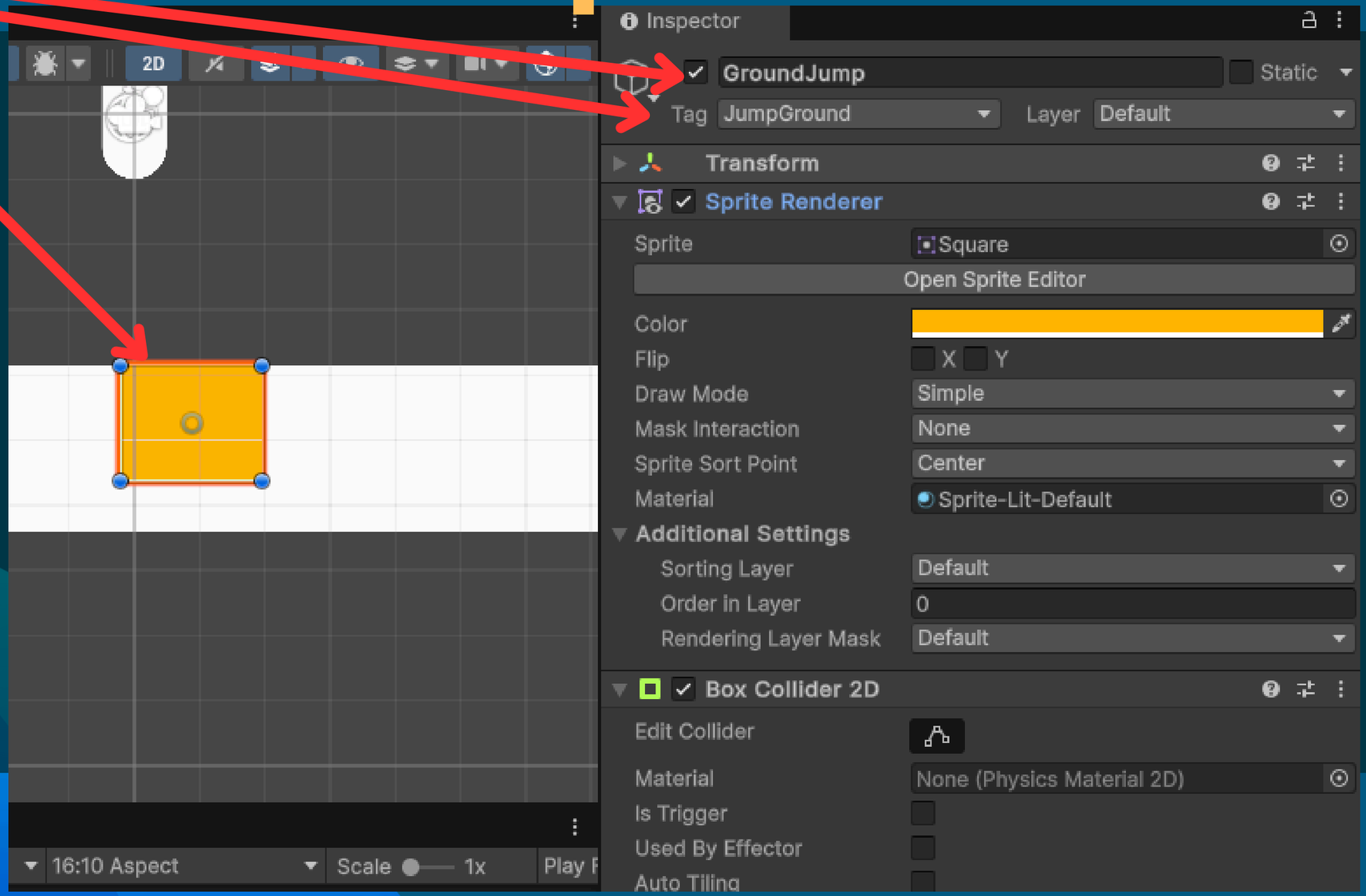




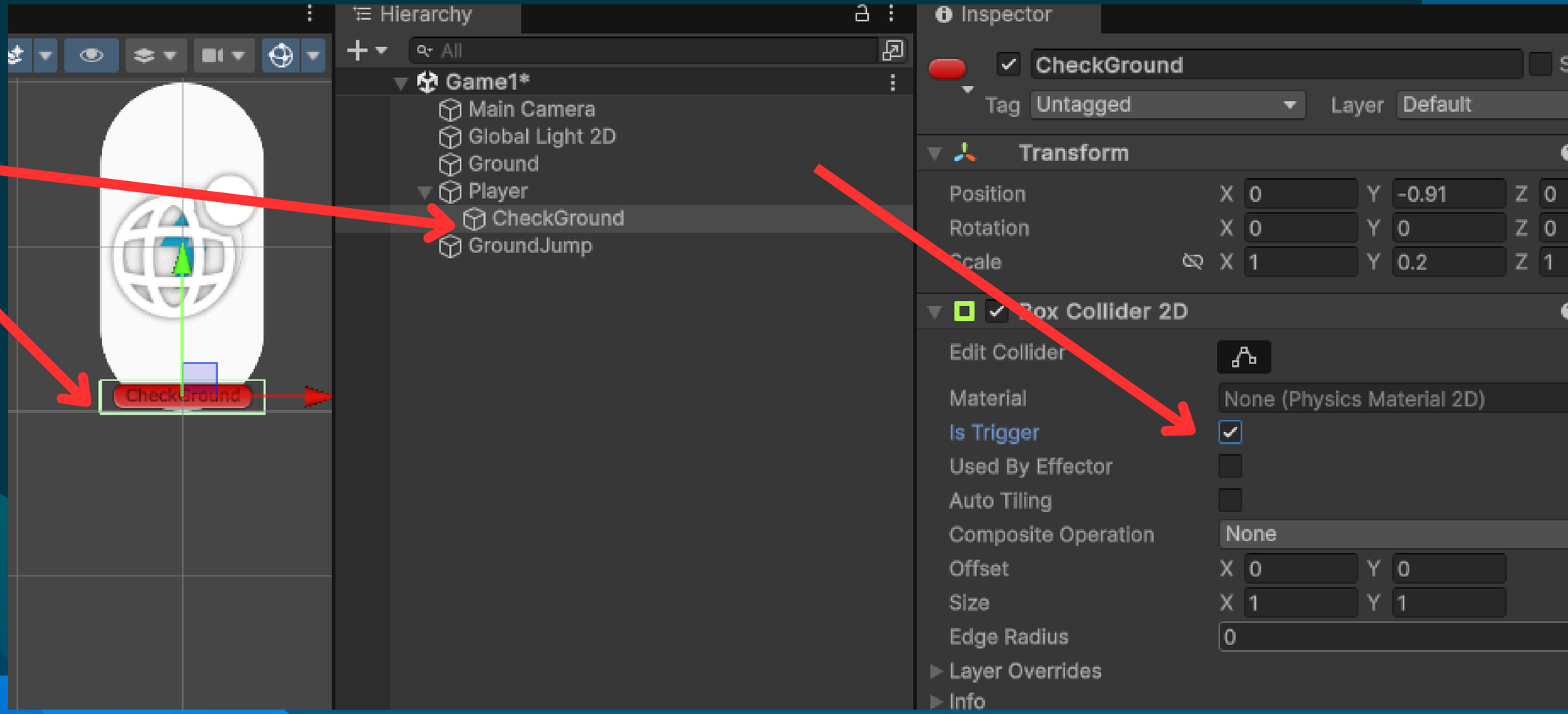
Configure Trigger

```
Mensagem do Unity | 0 referências
private void OnTriggerEnter2D(Collider2D collision)
{
    if (collision.gameObject.CompareTag("JumpGround"))
    {
        Debug.Log("Pular");
    }
}
```

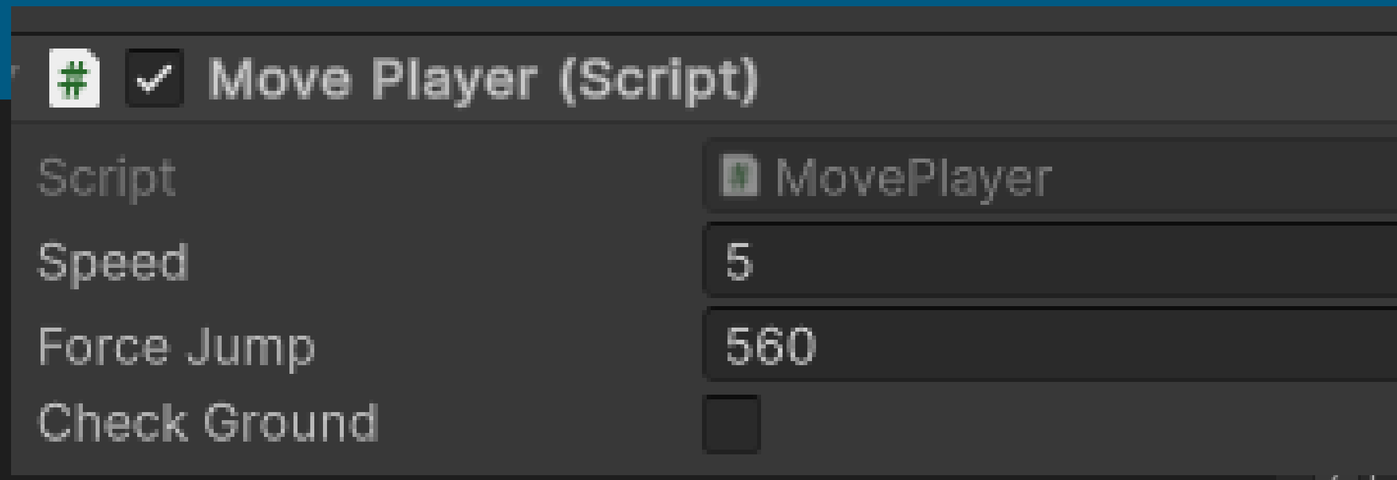
Configure GameObject GroundJUmp com a TaG



Configure Player com CheckGround

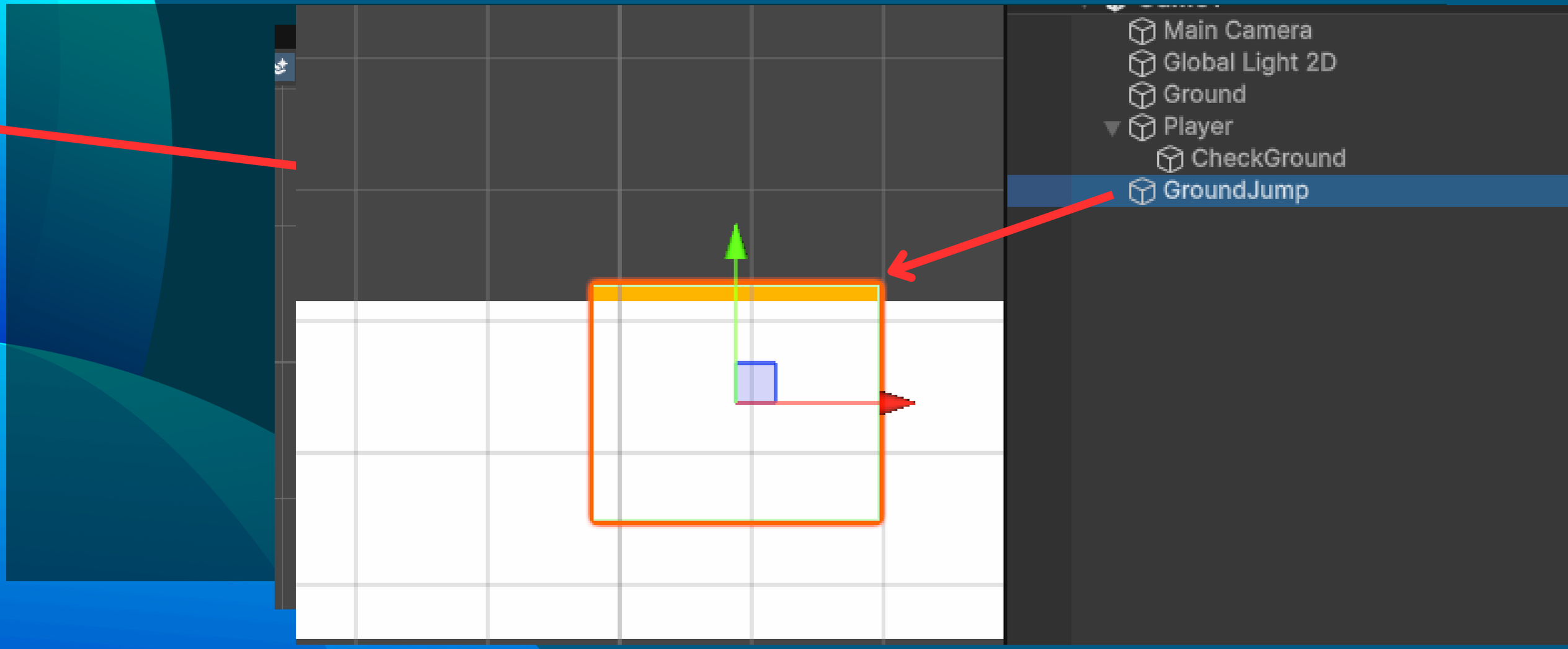


Declare Variavel Float _forceJump




```
1 referência
void Jump()
{
    _rb.linearVelocityY = 0;
    _rb.AddForceY(_forceJump);
}
```

Posicione o GroundJump



Crie uma classe
Com nome ObjectPool

https://learn.unity.com/tutorial/introduction-to-object-pooling

Copie e cole no ObjectPool

```
public static ObjectPool SharedInstance;
public List<GameObject> pooledObjects;
public GameObject objectToPool;
public int amountToPool;

void Awake()
{
    SharedInstance = this;
}

void Start()
{
    pooledObjects = new List<GameObject>();
    GameObject tmp;
    for(int i = 0; i < amountToPool; i++)
    {
        tmp = Instantiate(objectToPool);
        tmp.SetActive(false);
        pooledObjects.Add(tmp);
    }
}
```

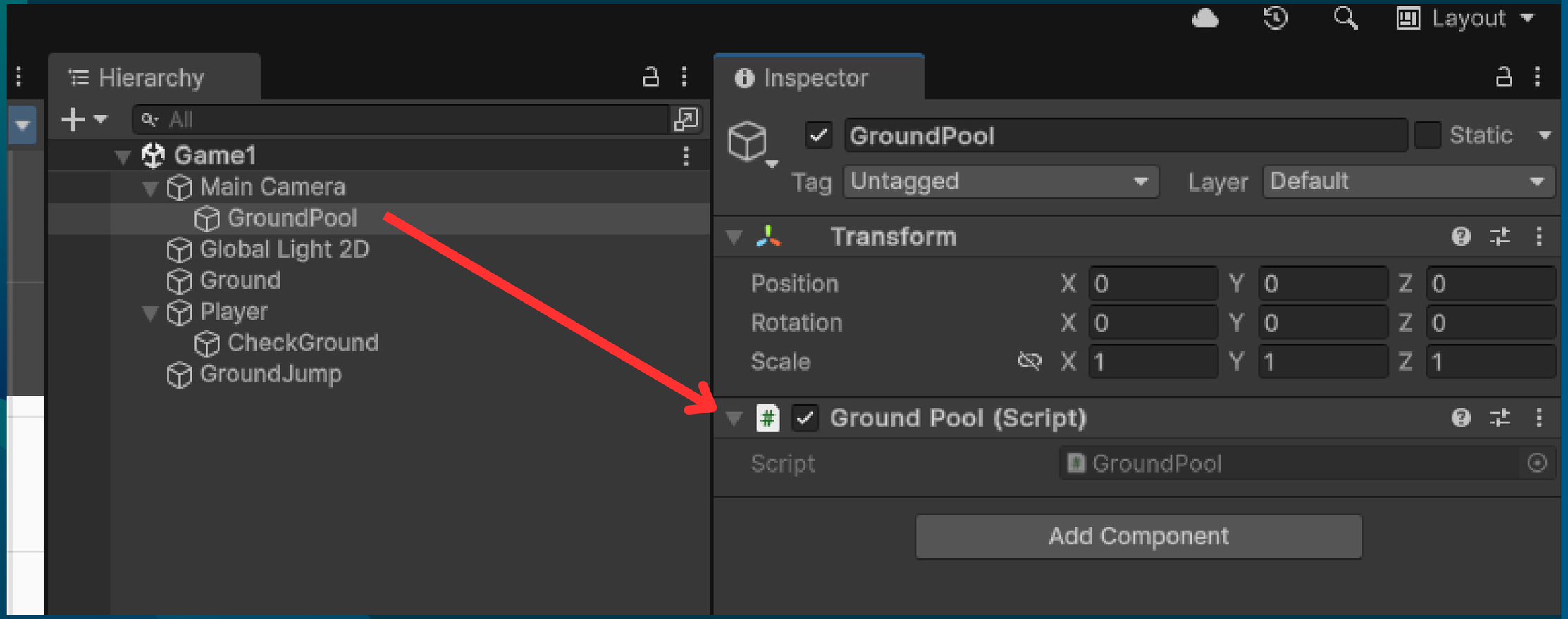
```
public GameObject GetPooledObject()
{
    for(int i = 0; i < amountToPool; i++)
    {
        if(!pooledObjects[i].activeInHierarchy)
        {
            return pooledObjects[i];
        }
    }
    return null;
}
```

Mude no ObjectPool

```
Script de Unity | 0 referências
public class ObjectPool : MonoBehaviour
{
    //public static ObjectPool SharedInstance;
    [SerializeField] protected List<GameObject> pooledObjects;
    [SerializeField] protected GameObject objectToPool;
    [SerializeField] protected int amountToPool;

    Mensagem do Unity | 0 referências
    void Awake()
    {
        //SharedInstance = this;
    }
```

Crie Classe GroundPool



Coloque virtual no ObjectPool

```
public class GroundPool : ObjectPool
{
    public static GroundPool _groundPool;

    Mensagem do Unity | 2 referências
    public override void Awake()
    {

        base.Awake();
        _groundPool = this;

    }
```

Coloque virtual no ObjectPool

```
//public static ObjectPool SharedInstance;
[SerializeField] protected List<GameObject> pooledObjects;
[SerializeField] protected GameObject objectToPool;
[SerializeField] protected int amountToPool;

Mensagem do Unity | 2 referências
public virtual void Awake()
{
    //SharedInstance = this;
}
```