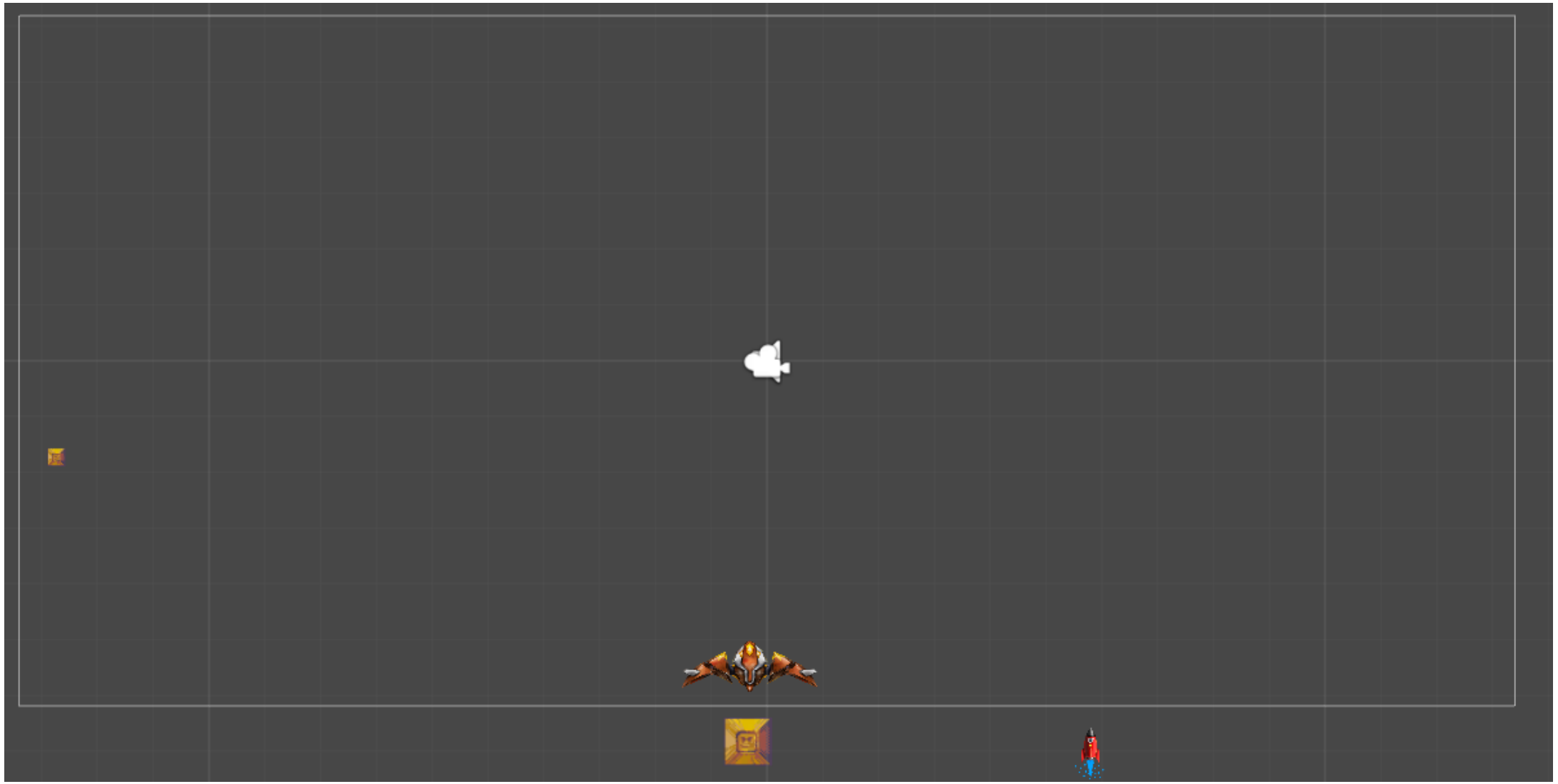
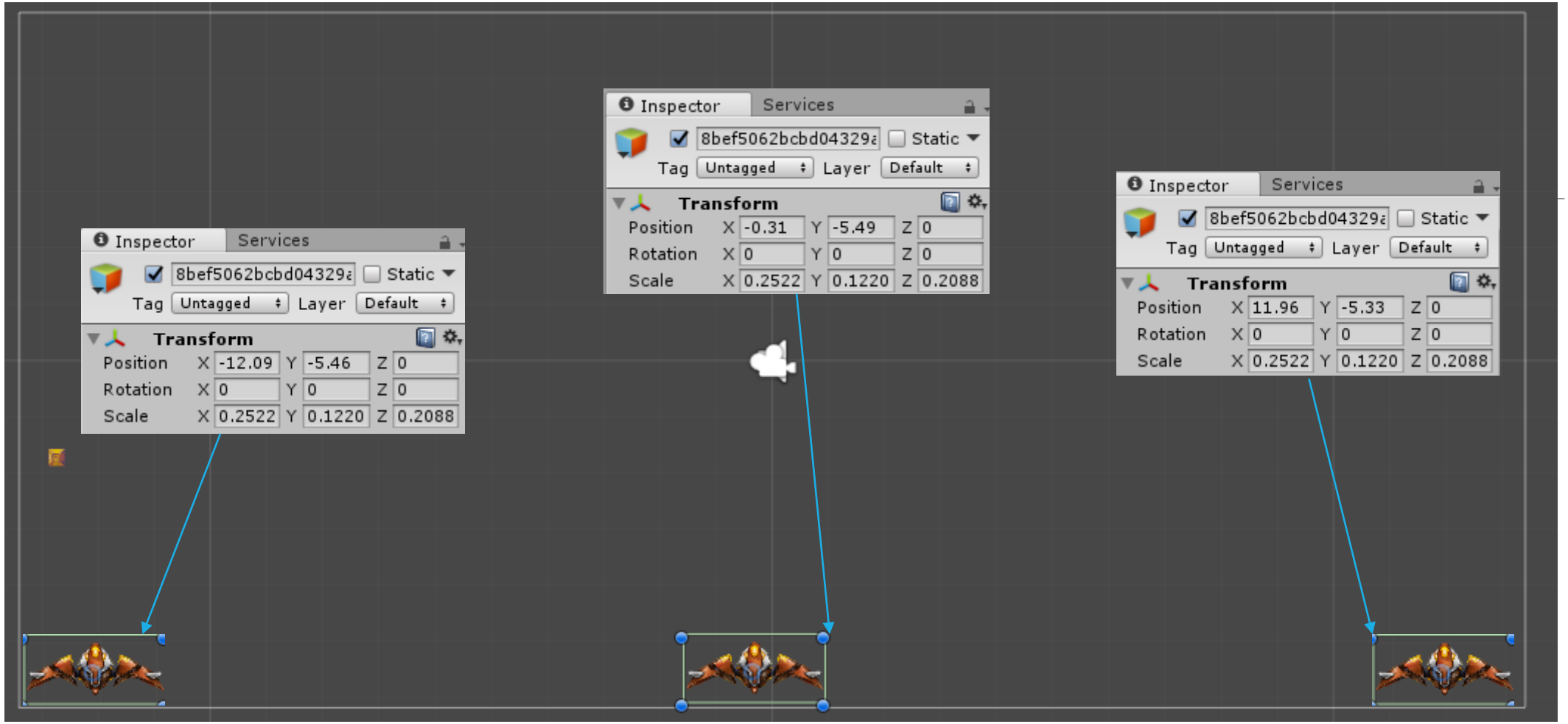
GERANDO INIMIGOS E DISPAROS AUTOMATICAMENTE

PROF. ME. HÉLIO ESPERIDIÃO




Inspector
Services
8bef5062bcbd04329a
Static
Tag
Untagged
Layer
Default
Transform
Position
X -12.09
Y -5.46
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 0.2522
Y 0.1220
Z 0.2088
Inspector
Services
8bef5062bcbd04329a
Static
Tag
Untagged
Layer
Default
Transform
Position
X -0.31
Y -5.49
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 0.2522
Y 0.1220
Z 0.2088
Inspector
Services
8bef5062bcbd04329a
Static
Tag
Untagged
Layer
Default
Transform
Position
X 11.96
Y -5.33
Z 0
Rotation
X 0
Y 0
Z 0
Scale
X 0.2522
Y 0.1220
Z 0.2088

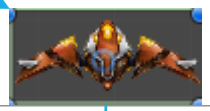
```
public class Personagem : MonoBehaviour {
    // recebe  corpo rígido do personagem (Componentes>Física 2d > Corpo Rígido)
    public Rigidbody2D personagemRigidbody2D;
    //recebe o transform do personagem
    private Transform personagemTransform;
    //recebe o SpriteRenderer do elemento de jogo ou do personagem
    private SpriteRenderer personagemSpriteRenderer;
    // Recebe zero quando está parado até -1 quando move para esquerda
    //e até +1 quando move para direita.
    private float direcaoHorizontal = 0;
    // determina a intensidade de caminhar.
    public float intensidadeMovimentoHorizontal = 15f;
    void Start () {
        //atribui o corpo rígido a variável.
        this.personagemRigidbody2D = gameObject.GetComponent<Rigidbody2D>();
        // Atribui o SpriteRenderer a variável
        this.personagemSpriteRenderer = gameObject.GetComponent<SpriteRenderer>();
        // Atribui o Transform a variável
        this.personagemTransform = gameObject.GetComponent<Transform>();
        // impede que o personagem ou elemento rotacione.
        this.personagemRigidbody2D.freezeRotation = true;
    }
```

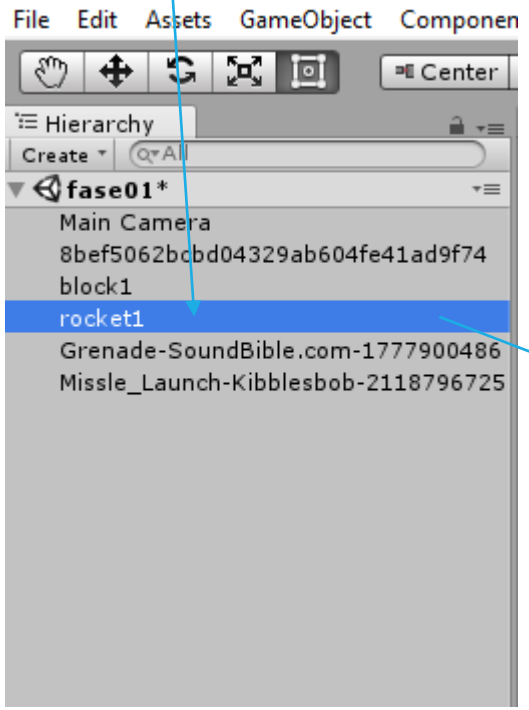
```
// Update is called once per frame
void Update () {
    MovimentoHorizontal();
    AtirarMissel();

}
```

```
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 compoenentes x, y do plano.
    Vector2 movimento = new Vector2(x, y);
    if (this.transform.position.x <= -12){
        this.transform.position = new Vector3(-10, this.transform.position.y, 0);
    }
    if (this.transform.position.x >= 12) {
        this.transform.position = new Vector3(10, this.transform.position.y, 0);
    }
    //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
}
```

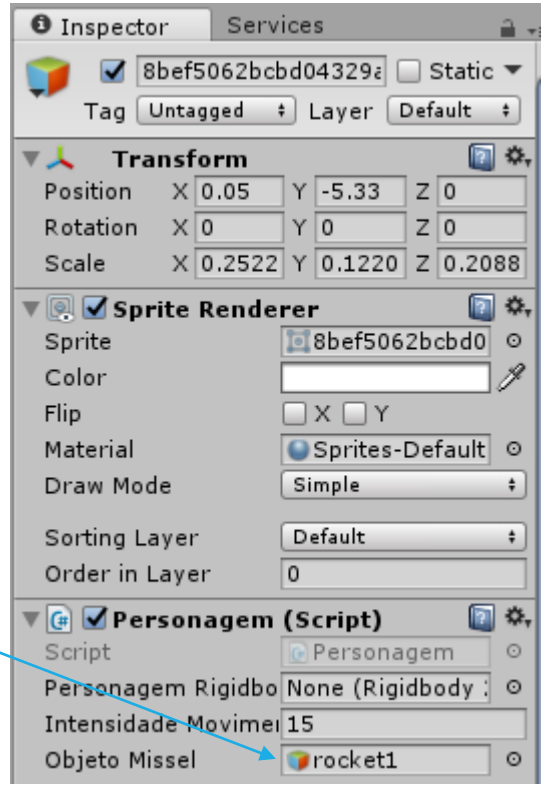
```
void AtirarMissel(){
    if (Input.GetKeyDown(KeyCode.Space)) {
        //inicializa um novo objeto missel.
        GameObject novoMissel = Instantiate(this.objetoMissel);
        //recupera o transform do novo missel.
        Transform transformNovoMissel = novoMissel.GetComponent<Transform>();
        Rigidbody2D rigidbodyNovoMissel = novoMissel.GetComponent<Rigidbody2D>();
        //recura a posição da nave
        Vector2 vetorPosicaoNave = gameObject.GetComponent<Rigidbody2D>().position;
        //Armazena a posição onde está a nave para depois
        //utilizar como referência para disparar o missel.
        Vector2 vetorPosicaoNovoMissel = vetorPosicaoNave;
        //acrescenta +1 na posição de y do missel para ele não
        //colidir com a própria nave que efetuou o disparo.
        vetorPosicaoNovoMissel.y = vetorPosicaoNovoMissel.y + 1;
        //adiciona o vetor de posicionamento no transform do missel
        transformNovoMissel.position = vetorPosicaoNovoMissel;
        //cria um vetor de velocidade para o missel
        //componente singnificativo apenas em y
        Vector2 vetorVelocidadeMissel = new Vector2(0, 20);
        rigidbodyNovoMissel.velocity = vetorVelocidadeMissel;
        //destroy o novo missel depois de 5 segundos
        Destroy(novoMissel, 5f);
    }
}
```

Ao clicar na nave
Veja suas propriedades





Arraste o Objeto Rockt1
que representa o míssil
para dentro do Objeto Míssil
Que está dentro do script
do personagem

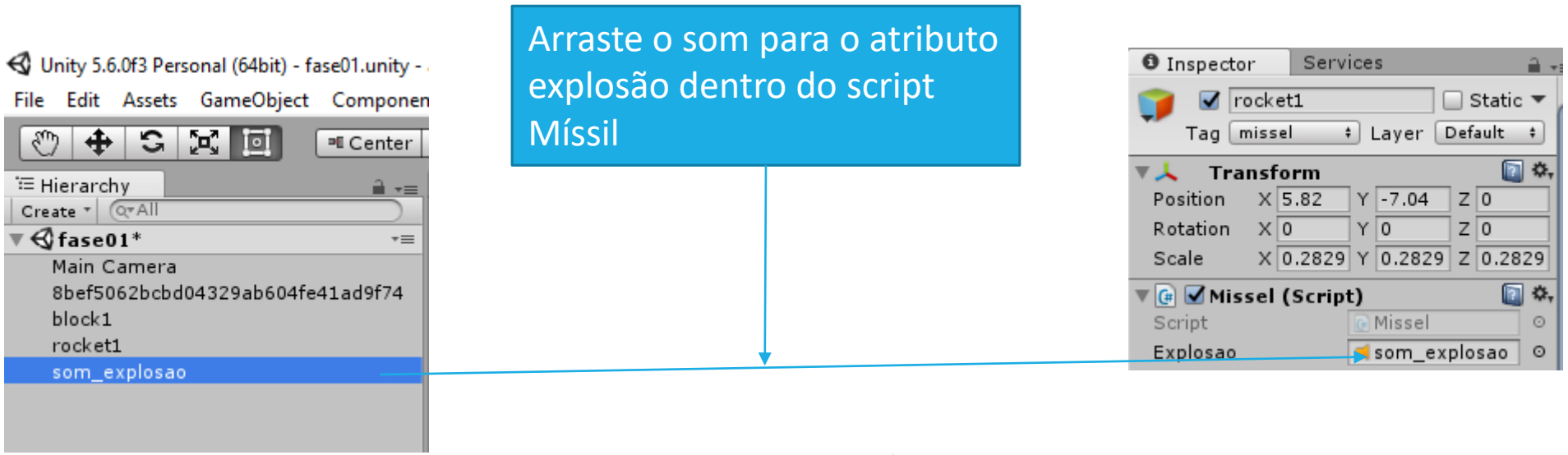


Míssil

Acrescente o script dentro do míssil

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Missel : MonoBehaviour {
    public AudioSource explosao;
    void Start () {

    }
    void Update () {

    }
    private void OnCollisionEnter2D(Collision2D collision){
        if (collision.gameObject.tag == "pedra"){
            explosao.Play();
            Destroy(collision.gameObject);
        }
    }
}
```

Arraste o componente de áudio para dentro do scene.



Não esqueça de desmarcar a opção de áudio Play on awake

```
void GerarInimigo(){
    //gera um número aleatório entre 0 e 8;
    //como a nave pode se movimentar de -12 até 12
    //o intervalo gerado permite que naves sejam geradas
    //entre -8 e 8
    int posicaoMapa = Random.Range(0, 8);
    //entre zero e 8 tempos apenas números positivos.
    //vamos fazer um sorteio de 0 a 100
    //caso o sorteado seja maior que 50 multiplicamos
    //o valor de 0 a 8 por -1
    int direitaOuEsquerda = Random.Range(0, 100);
    if (direitaOuEsquerda > 50){
        posicaoMapa = posicaoMapa * -1;
    }
    // o comando Instantiate gera uma cópia de um gameObjet
    GameObject novaPedra = Instantiate(inimigo);
    //Armazena o Rigidbody2D da nova pedra que foi criada;
    Rigidbody2D novaPedraRigidbody2D = novaPedra.gameObject.GetComponent<Rigidbody2D>();
    //gera um valor aleatório que será inserido no gravityScale da nova pedra.
    novaPedraRigidbody2D.gravityScale = Random.Range(0.01f,0.1f);
    //cria um vetor de posição para a nova pedra.
    //o vetor possui o número sorteado entre -8 e 8 parax
    //y é igual a 6 para ficar fora do campo de visão da câmera
    Vector2 posicaoNovaPedra= new Vector2(posicaoMapa, 6);
    //acrecenta o vetor de posicionamento na novaPedraRigidbody2D
    novaPedraRigidbody2D.position = posicaoNovaPedra;
    //A nova pedra será automáticamente destruida depois de 10 segundos
    Destroy(novaPedra, 10f);
}
```

Acrescente o script inimigo na câmera

Arraste o bock1 que será utilizado como inimigo para a propriedade Inimigo do script Inimigo.