

# Desenvolvendo "Foguinho e Aguinha" na Unity 3D

Criar, atualizar e personalizar um sumário

O jogo Foguinho e Aguinha é um jogo de plataforma cooperativo onde dois personagens, um de fogo e outro de água, precisam colaborar para superar desafios. Este tutorial guiará você pelo processo de desenvolvimento na Unity 3D (versão 6000), usando C# e a biblioteca padrão da Unity.

## Configuração do Projeto

1. Abra a Unity e crie um novo projeto 2D.
2. No menu Game, configure a resolução para 640x480 (ou outra de sua preferência).
3. Importar Sprites e Assets: Adicione sprites para os personagens, plataformas, portais e botões.

## Configuração dos Sprites

1. Abra o sprite e renomeie como informado



Coloque o nome de UI\_Player\_WaterGirl\_Walking\_01 a UI\_Player\_WaterGirl\_Walking\_08

## Criando os Personagens

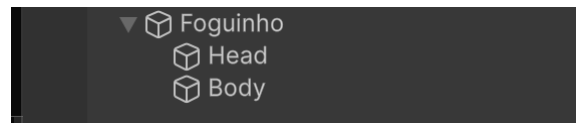
Vamos criar os dois personagens do jogo

### Criar o Foguinho e a Aguinha

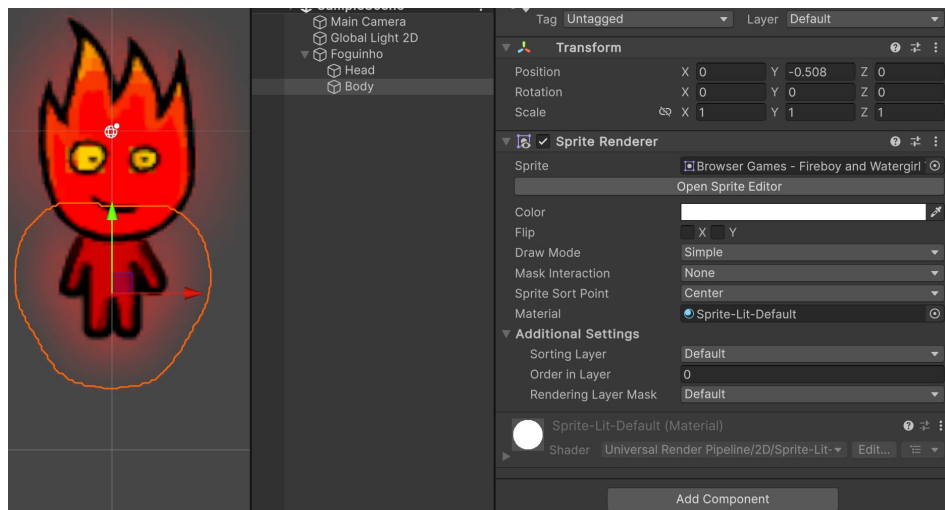
Repita os passos acima para a Aguinha. No Sumário, selecione uma entrada de Nível 1 inteira

1. No Hierarchy, crie um objeto vazio e renomeie como Foguinho.

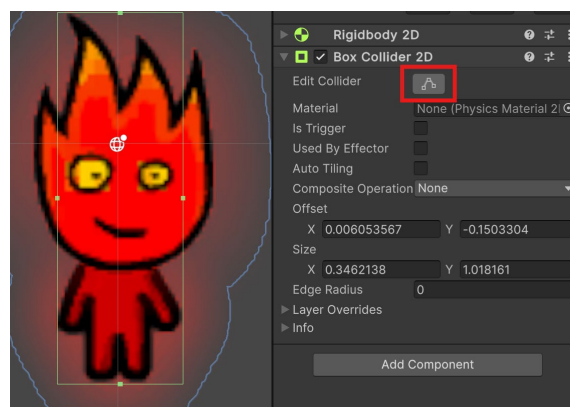
2. Crie outros dois objetos vazios e coloque o nome de Head e Body
3. Arraste os dois objetos para o gameobject pai Foguinho



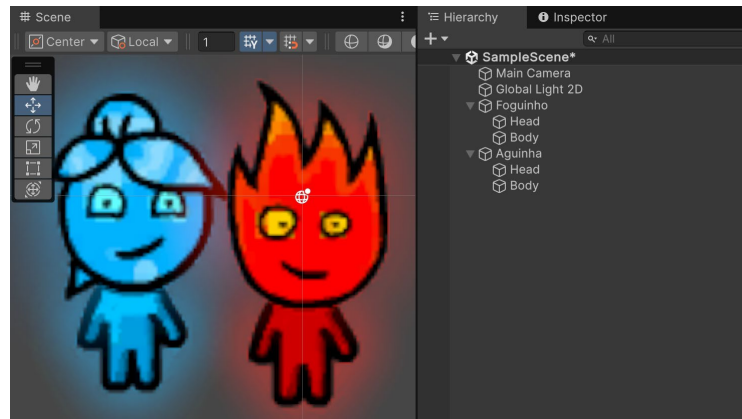
4. Adicione um Sprite Renderer no Head e Body e atribua a imagem correspondente na qual o personagem fique de frente.



5. Adicione um Rigidbody2D (modo "Dynamic") no gameobject Foguinho.
6. Adicione um BoxCollider2D e modifique a colisão para cobrir toda a área do personagem.



7. Repita os passos acima para a Aguinha. Para facilitar faça uma cópia de Foguinho e renomeie o objeto para Aguinha e mude os sprites. Se o corpo ficar muito para baixo, puxe para cima.



## Criando o Script de Movimento e Animação

Os personagens devem responder a controles específicos e ter animações.

No script Player.cs, adicione:

```
using UnityEngine;
```

```
public class Player : MonoBehaviour
```

```
{
```

```
    public float speed = 3f;
```

```
    public float jumpForce = 5f;
```

```
    private Rigidbody2D rb;
```

```
    private bool isGrounded;
```

```
    public KeyCode leftKey;
```

```
    public KeyCode rightKey;
```

```
    public KeyCode jumpKey;
```

```
    public string characterType;
```

```
    private Animator animator;
```

```
    void Start()
```

```
{
```

```
        rb = GetComponent<Rigidbody2D>();
```

```
        animator = GetComponent<Animator>();
```

```
}
```

```
    void Update()
```

```

{
    float move = 0;

    if (Input.GetKey(leftKey))
    {
        move = -speed;

        animator.SetBool("isWalking", true);

        transform.localScale = new Vector3(-1, 1, 1);
    }

    else if (Input.GetKey(rightKey))
    {
        move = speed;

        animator.SetBool("isWalking", true);

        transform.localScale = new Vector3(1, 1, 1);
    }

    else
    {
        animator.SetBool("isWalking", false);
    }

    rb.velocity = new Vector2(move, rb.velocity.y);

    if (Input.GetKeyDown(jumpKey) && isGrounded)
    {
        rb.velocity = new Vector2(rb.velocity.x, jumpForce);

        isGrounded = false;

        animator.SetTrigger("Jump");
    }
}

private void OnCollisionEnter2D(Collision2D collision)
{
    if (collision.gameObject.CompareTag("Ground"))
    {
        isGrounded = true;
    }
}

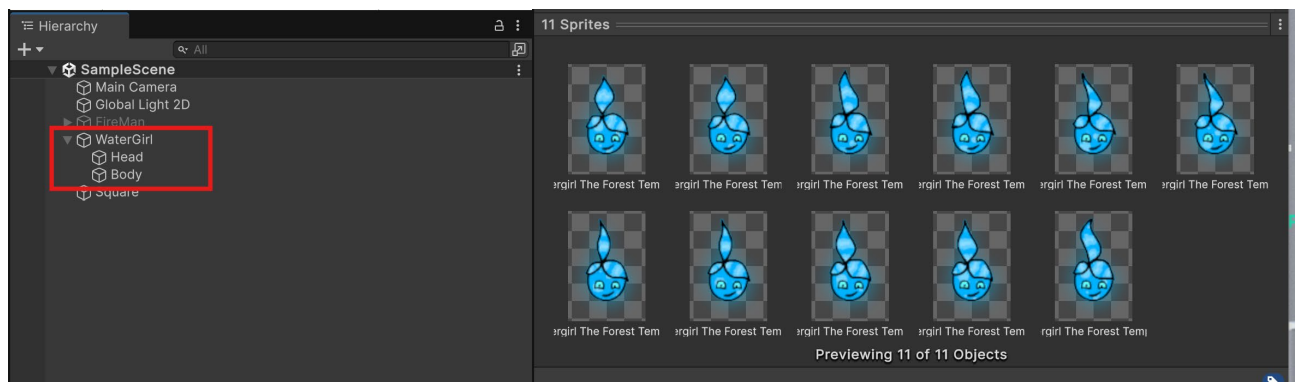
```

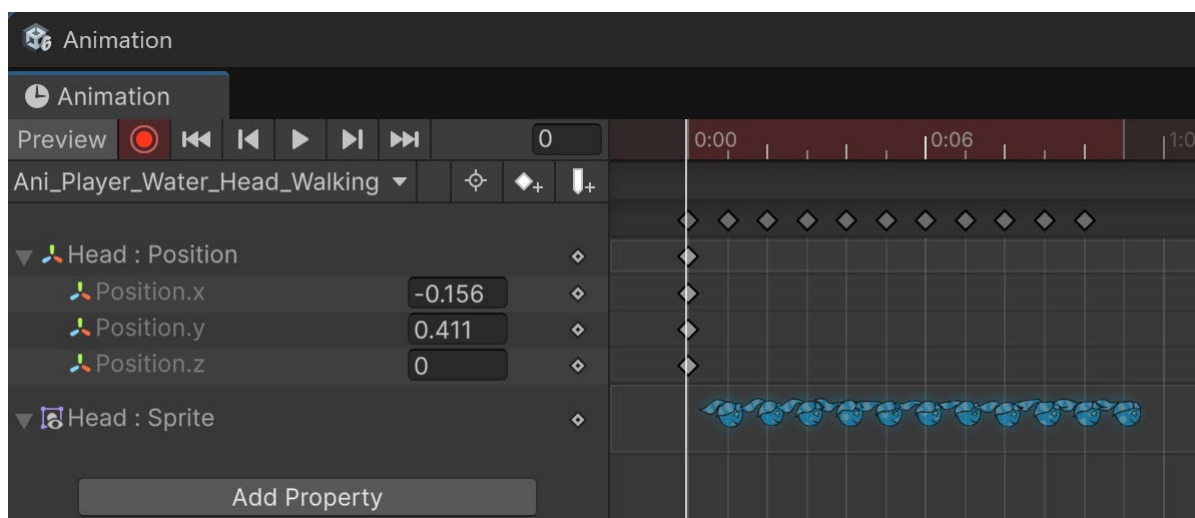
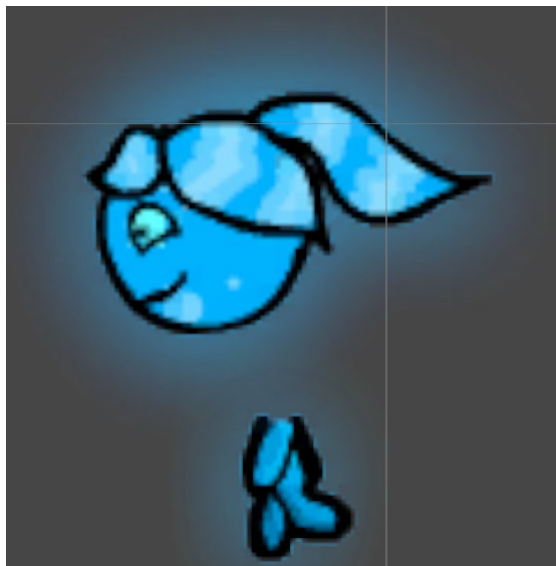
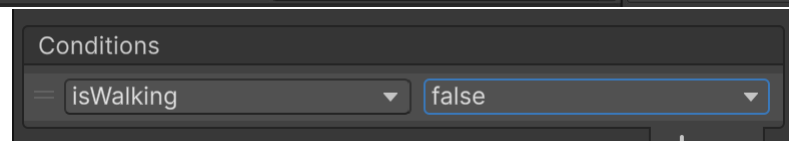
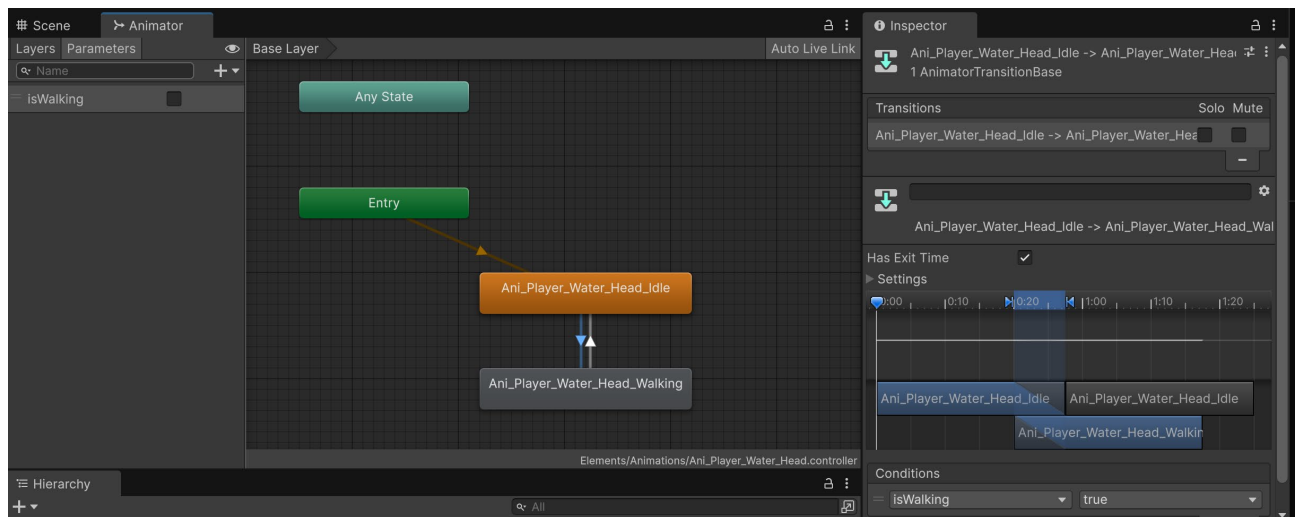
```

private void OnTriggerEnter2D(Collider2D collision)
{
    if ((characterType == "Fogo" && collision.CompareTag("Agua")) ||
        (characterType == "Agua" && collision.CompareTag("Fogo")) ||
        collision.CompareTag("Goo"))
    {
        animator.SetTrigger("Dying");
        Destroy(gameObject, 0.5f); // Pequeno atraso para mostrar animação de morte
    }
}
}

```

8. Associe o script ao Foguinho e Aguinha.
9. No Inspector, configure leftKey, rightKey e jumpKey para os respectivos controles.
10. Criação das Animações: Adicione um Animator a cada personagem.
11. Crie os seguintes estados no Animator Controller: Idle (Parado), Walking (Andando), Jumping (Pulando), Dying (Morrendo), Configure os triggers e bools de acordo com o script acima.





# Criando o Cenário

---

Vamos agora criar a sua fase

## Criar o chão

Será o chão do seu jogo

**12.** No Hierarchy, crie um novo GameObject, adicione um BoxCollider2D e renomeie para Chão.

**13.** Marque a tag Ground.

## Criar Plataformas e Obstáculos

Será a plataforma e obstáculos

**14.** Use objetos 2D para criar plataformas.

**15.** Adicione BoxCollider2D para colisões.

# Criando os Portais e Portas

---

Criação dos portais e portas do jogo

**16.** No Hierarchy, crie dois objetos vazios chamados PortalFogo e PortalAgua.

**17.** Adicione BoxCollider2D e marque como "Trigger".

**18.** No script Doors.cs, adicione:

```
using UnityEngine;
```

```
public class Doors : MonoBehaviour
```

```
{
```

```
    public string requiredTag;
```

```
    private bool isOpen = false;
```

```
    private void OnTriggerEnter2D(Collider2D collision)
```

```
{
```

```
        if (collision.gameObject.CompareTag(requiredTag))
```

```
{  
    isOpen = true;  
}  
}  
}
```

**19.** Associe o script às portas e configure requiredTag conforme necessário.

## Criando Botões para Portões

---

Criação dos botões e portões

**20.** Crie um objeto vazio chamado Botão e adicione BoxCollider2D (como "Trigger").

**21.** No script Gates.cs, adicione:

```
using UnityEngine;  
  
public class Gates : MonoBehaviour  
{  
    public GameObject gate;  
    private bool platePressed = false;  
  
    private void OnTriggerEnter2D(Collider2D collision)  
    {  
        if (collision.CompareTag("Fogo") || collision.CompareTag("Agua"))  
        {  
            platePressed = true;  
            gate.SetActive(false);  
        }  
    }  
}
```

**22.** Associe o script ao botão e arraste a porta no Inspector.

## Criando o Sistema de Reinício e Vitória

---

Essa parte o jogo irá reiniciar e poderá ganhar o jogo



**23.** Adicione um botão no UI para reiniciar o jogo.

**24.** No script GameManager.cs, adicione:

using UnityEngine;

```
public class Gates : MonoBehaviour
{
    public GameObject gate;
    private bool platePressed = false;

    private void OnTriggerEnter2D(Collider2D collision)
    {
        if (collision.CompareTag("Fogo") || collision.CompareTag("Agua"))
        {
            platePressed = true;
            gate.SetActive(false);
        }
    }
}
```

**25.** Associe o script ao botão e arraste a porta no Inspector.