

## Personagem no jogo

### Transcrição

Anteriormente, colorimos o cenário. Agora, adicionaremos **personagens** a ele. Lembrando que os objetos serão extraídos da pasta "Prefabs", na qual já estão configurados. Para acessar a pasta de personagens, clicaremos em "Assets > Prefabs > Personagens".

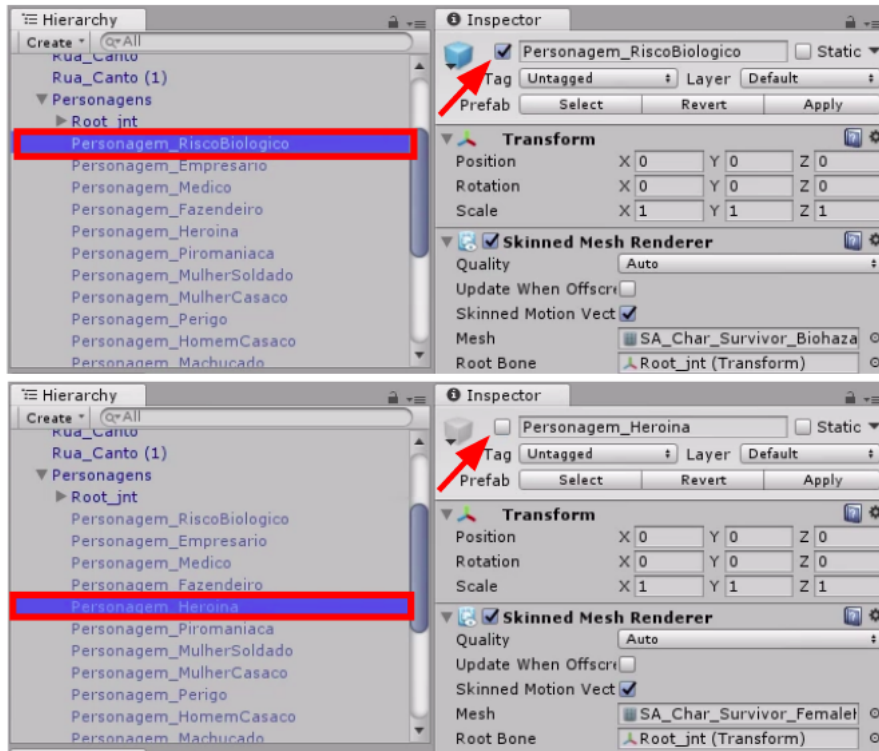
Na visualização da pasta, clicaremos em "Personagens" e arrastaremos para "Hierarchy". Podemos não encontrá-lo de primeira, mas como ele está selecionado em "Hierarchy", para **localizá-lo** no cenário:

- clicaremos em "Scene";
- teclaremos "F", que é o atalho de localização de personagem;
- teclaremos "W", para movê-lo para o centro do cenário.



Notem que em "Hierarchy", o nome está como "Personagens", indicando que há mais de um. Para visualizar os outros, clicaremos na seta à esquerda de "Personagens". Abrirá um menu com diversos tipos. O título do primeiro está com a fonte em azul escuro, enquanto os outros estão em azul claro. Essa coloração distingue os personagens ativos dos inativos. No caso, **para trocarmos o personagem**, precisamos:

- **desativá-lo**, desmarcando a caixa de seleção à esquerda do nome dele ("Personagem\_RiscoBiologico"), em "Inspector";
- selecionar e **ativar** "Personagem\_Heroína", marcando a caixa de seleção dela.

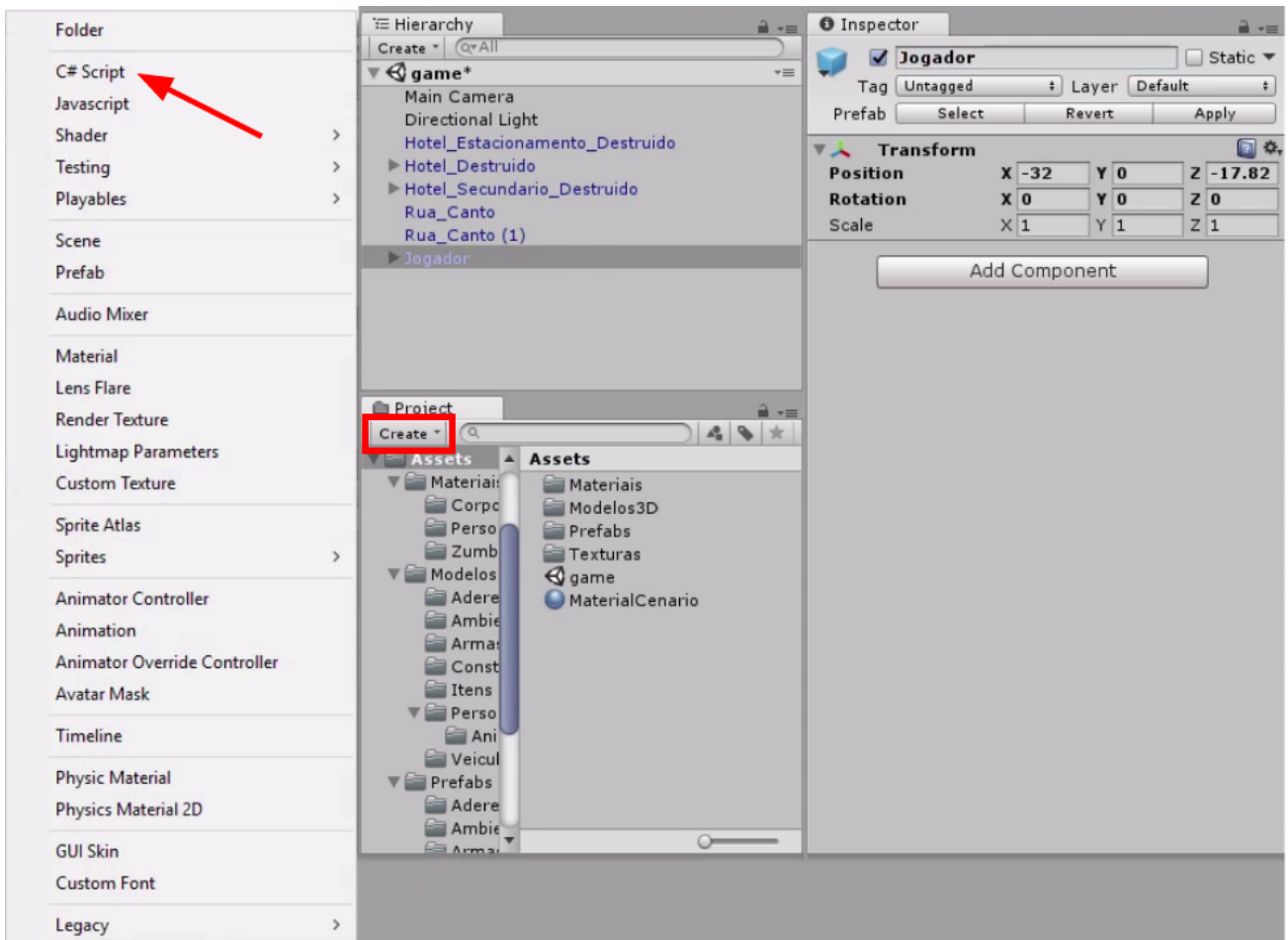


Após aplicar a troca, em "Scene", veremos a "Personagem\_Heroina". Se preferirem, podem trocar, ativando e desativando os personagens em "Inspector".

Renomearemos o objeto para facilitar a associação ao papel dele no cenário. Clicaremos em cima de "Personagens", em "Hierarchy", para que fique editável, e digitaremos "Jogador". A renomeação também pode ser aplicada em "Inspector", à direita da caixa de seleção que ativa e desativa os personagens.

O próximo passo será fazer o personagem andar pelo cenário a partir de um *script*, um **código** que faz a Unity executar as ações especificadas nele. Para inseri-lo, em "Project":

- abriremos a pasta "Assets";
- clicaremos em "Create", abaixo da identificação da aba "Project";
- selecionaremos "C# Script".



Nomearemos como "ControlaJogador" e clicaremos duas vezes nele para abrir. Notem que abrirá no *Visual Studio Code* ou no *MonoDevelop*, se utilizar Mac. No editor de texto encontraremos:

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

public class ControlaJogador : MonoBehaviour {

    // Use this for initialization
    void Start () {

    }

    // Update is called once per frame
    void Update () {

    }
}
```

Tudo que escrevermos ficará abaixo da classe que contém o mesmo nome ( *ControlaJogador* ) do *script* que criamos, entre as chaves ( `{ }` ) de:

```
public class ControlaJogador : MonoBehaviour {}
```

Começaremos deletando o seguinte trecho, para manter o foco naquilo que utilizaremos nesse momento:

```
// Use this for initialization
void Start () {

}
}
```

Editaremos o trecho que contém `Update`, referente à parte da Unity que está sempre rodando, aquela que é executada para que o jogo aconteça. Nela, especificaremos os comandos que farão a personagem se movimentar. Antes, minimizaremos o editor de texto para observar um detalhe do movimento do personagem em "Inspector".

Notem que:

- se teclarmos "W" e movimentarmos "Jogador" no eixo Z, o valor de "Position Z" mudará de acordo com o movimento;
- "Position" ("Posição" em português) está dentro de "Transform", ou seja, para editar "Position Z", temos que entrar em "Transform".

Assim, no código:

- digitaremos `transform`;
- para entrar nele, utilizaremos ponto ( `.` );
- digitaremos `translate` para mover o objeto;
- acrescentaremos parênteses;
- dentro dos parênteses, adicionaremos `Vector3`, que agrega os três eixos (x, y e z);
- utilizaremos ponto ( `.` ) para entrar nele;
- digitaremos `forward`, que contém ( `0,0,1` ), ou seja, movimenta-se somente no eixo z, pois:
  - eixo x = 0
  - eixo y = 0
  - eixo z = 1
- encerraremos a linha de código com ponto e vírgula ( `;` ).

O código ficará da seguinte forma:

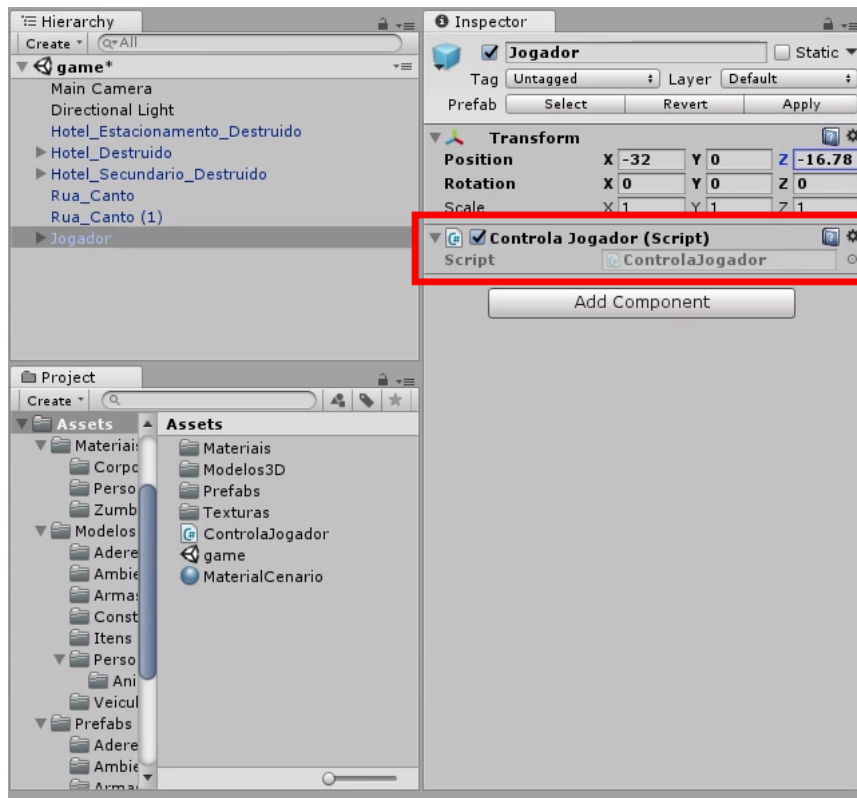
```
public class ControlaJogador : MonoBehaviour {

// Update is called once per frame
void Update () {
    transform.Translate(Vector3.forward);
}
}
```

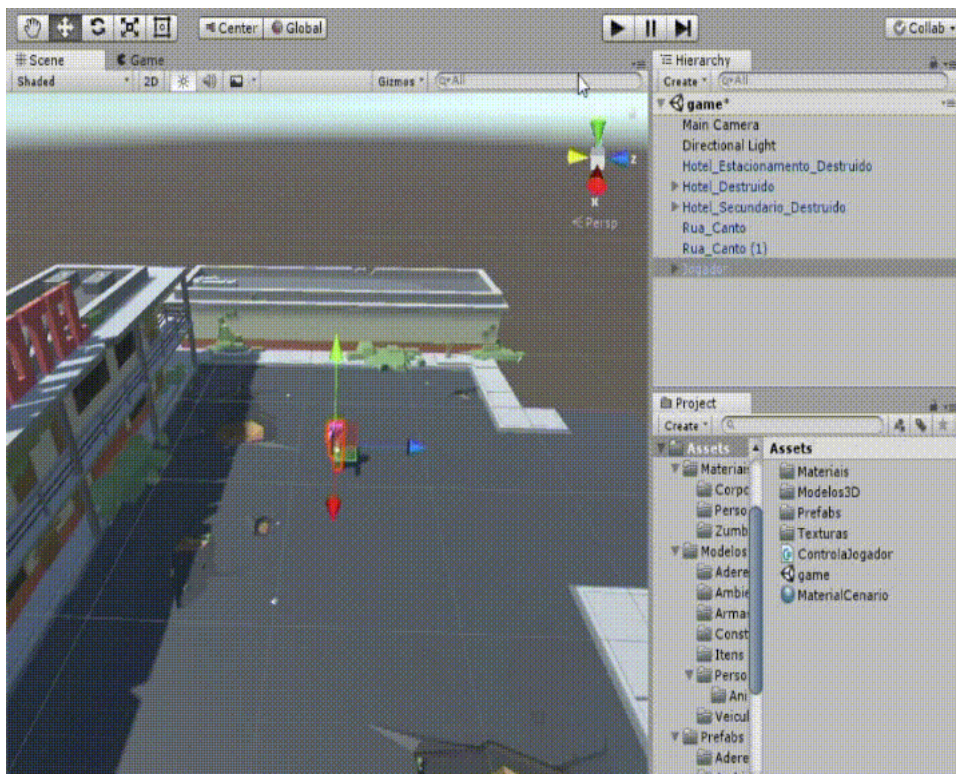
Assim, movimentaremos o personagem no eixo z. Se clicarmos no botão "Play", no topo da Unity, indicado pelo desenho de uma seta apontando para a direita



, nada acontecerá, porque precisamos especificar em **qual objeto** será aplicado o movimento. Para isso, clicaremos em "ControlaJogador", em "Assets", e arrastaremos para "Inspector" de "Jogador". Após arrastá-lo é possível vê-lo abaixo de "Transform".

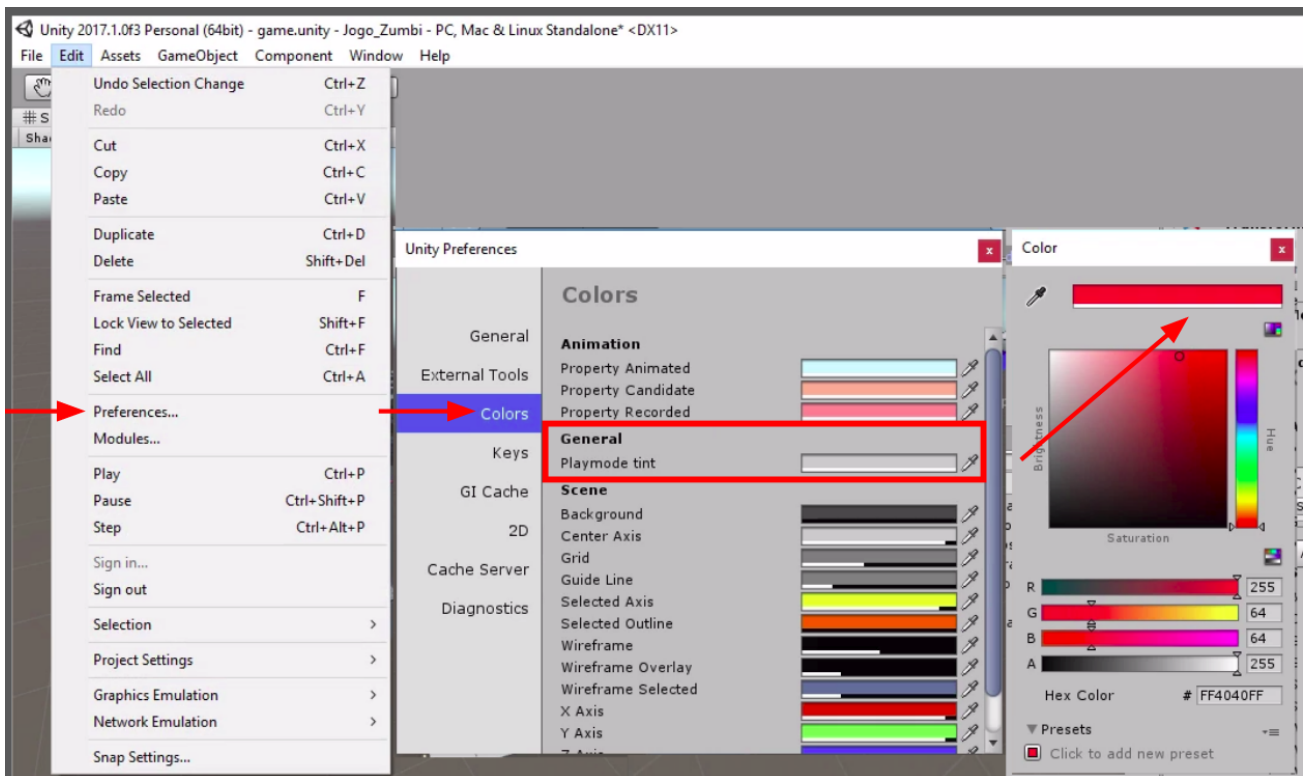


Se clicarmos em "Play" novamente, veremos "Jogador" deslizar ininterruptamente para frente, rumo ao horizonte.

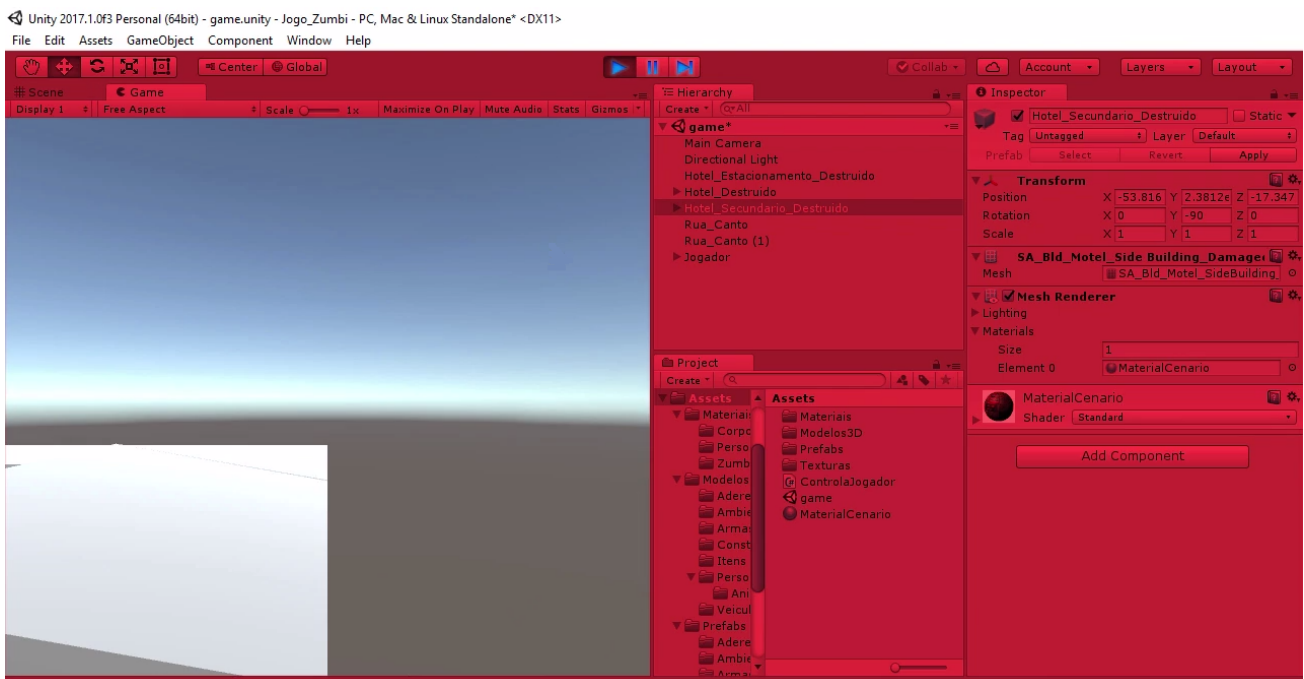


Notem que ao clicar em "Play", a aba "Scene" vai para "Game", à direita dela, na qual vemos o jogo rodar. Precisamos prestar atenção à aba em que estamos e ao botão "Play", se está ativado ou não, pois **se editarmos "Scene" com o botão "Play" ativado**, ao desativá-lo, todas as **edições serão perdidas**.

A Unity escurece um pouco a tela quando "Play" está ligado, mas é quase imperceptível. Podemos utilizar outra cor para garantir se está ativado ou não, clicando em "Edit > Preferences... > Colors" e trocando a cor de "Playmode tint", em "General", arrastando o círculo da cor cinza para vermelho.



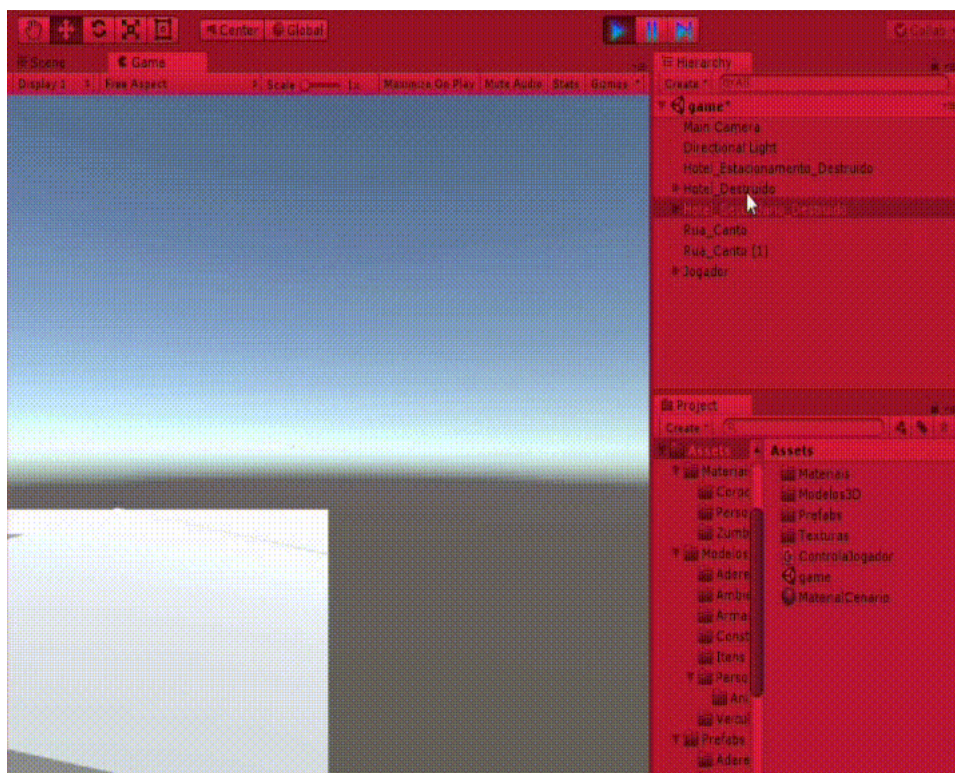
Assim, se o "Play" estiver ligado, não passará despercebido, pois a Unity inteira, com exceção da janela de "Game" estará tingida de vermelho.



Dessa forma, ficará fácil de distinguir quando podemos aplicar alterações no cenário ou não.

Em "Scene", com "Play" ativado, se teclarmos:

- **duas vezes** a letra "F", **acompanharemos** a personagem se deslocando no eixo z;
- **uma vez** a letra "F", **interromperemos o acompanhamento** e a personagem continuará o movimento ininterrupto para frente.



E assim, começamos a movimentar a personagem, dentro da Unity.