

Reiniciar o Jogo no Unity

Nosso inimigo já ataca o jogador, agora temos que fazer o famoso Game Over no nosso código e reiniciar o jogo. Vamos ver como podemos fazer isto?

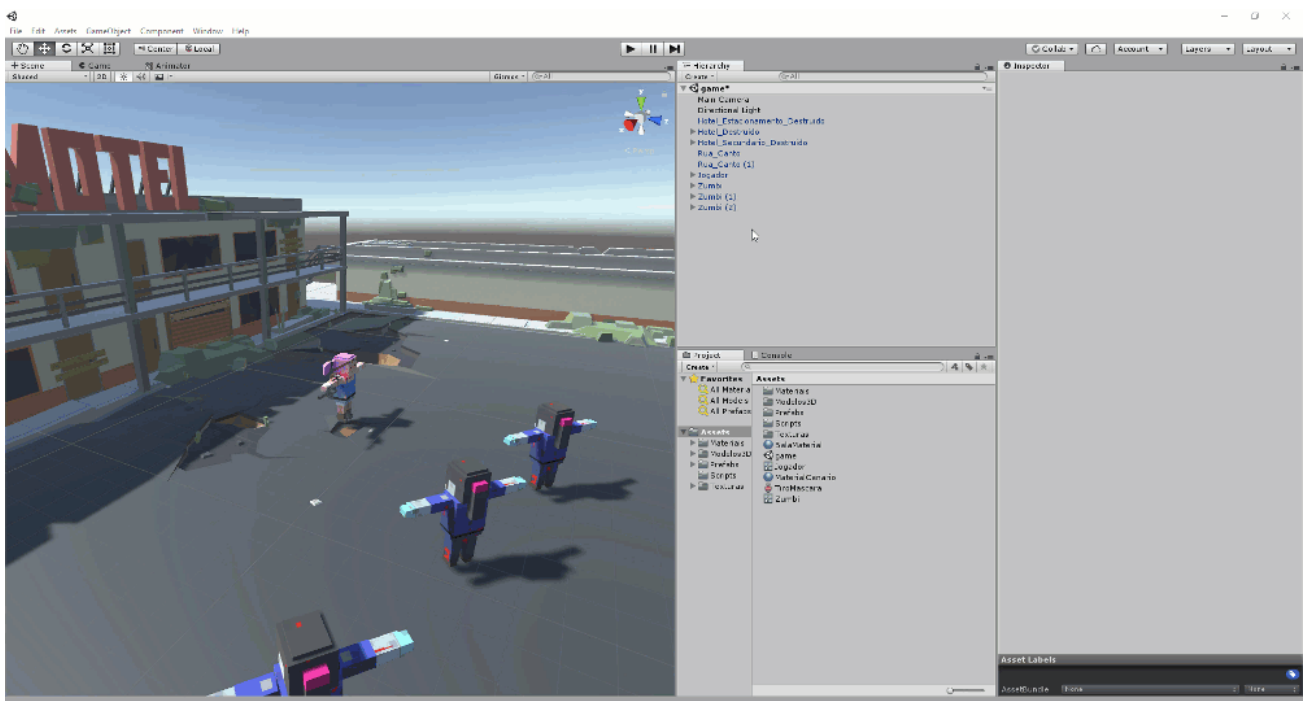
Inicialmente o que vamos fazer é no método do nosso *Script* `ControlaInimigo` vamos no método `AtacaJogador` e vamos pausar o jogo para isto com a linha `Time.timeScale = 0;`

`timeScale` é escala de tempo do Unity, como zeramos o valor isso significa que nada irá acontecer no nosso jogo. Seu código fica da seguinte forma:

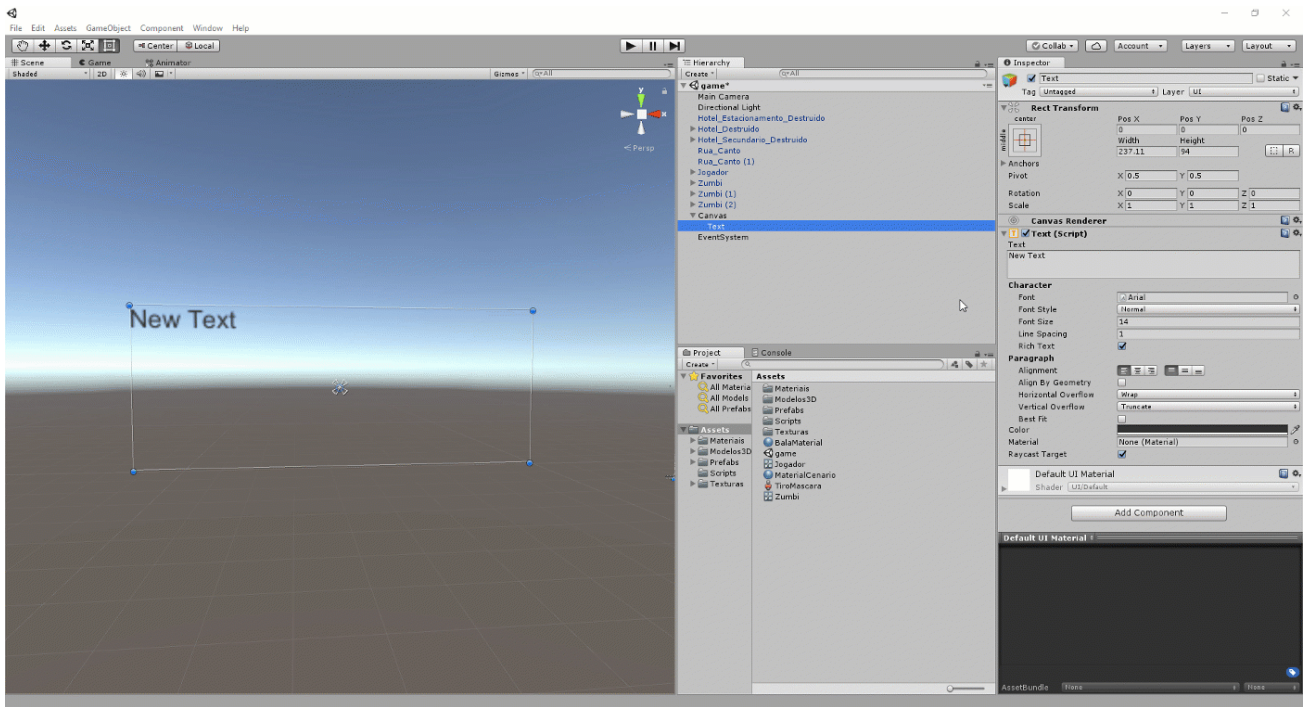
```
void AtacaJogador ()
{
    Time.timeScale = 0;
}
```

Só que se o jogador pausar o jogo, não pode parecer que o jogo deu Game Over. Então vamos colocar uma mensagem na tela?

Para isto temos que criar uma Interface no jogo então vamos na *Hierarquia* e acessar o menu **Create > UI > Text** e vamos aumentar esse texto com a ferramenta de **Transformação de Retângulos (Rect Transform)** que o atalho é **T** e vamos aumentar este texto e zerar a posição em X, Y e Z.



Além disso vamos no Componente de Texto e vamos trocar a mensagem para **Você Perdeu! Clique para recomeçar,** aumentar a fonte no **Font Size**, no **Alignment** alinhamos o texto ao centro e vamos trocar a cor para Branco.



Legal agora temos uma mensagem para o Jogador saber que ele perdeu, mas essa mensagem não pode ficar o tempo todo na tela, certo? Vamos então desativá-la pelo código. Para isso vamos até o *Script* `ControlaJogador` e junto das variáveis públicas que já temos vamos criar mais duas.

```
public GameObject TextoGameOver;
public bool Vivo = true;
```

Lembre-se de preencher a variável `TextoGameOver` com o seu Texto da Interface no Inspector.

Agora temos que no *Script* `ControlaInimigo` usar estas variáveis que criamos no método `AtacaJogador` passando a variável de caso o Jogador esteja `Vivo` para falso e desabilitando o GameObject do texto.

```
void AtacaJogador ()
{
    Time.timeScale = 0;
    Jogador.GetComponent<ControlaJogador>().TextoGameOver.SetActive(true);
    Jogador.GetComponent<ControlaJogador>().Vivo = false;
}
```

Legal agora resta a nós reiniciarmos nosso jogo para isto bem para isso no *Script* `ControlaJogador`.

Para iniciar o que vamos fazer é bem no começo do código fora da Classe do nosso código vamos digitar a linha.

```
using UnityEngine.SceneManagement;
```

Essa linha faz com que no nosso código possamos utilizar funções de manejo de Cenas. Então agora no nosso Update vamos testar se o jogador está vivo, se não estiver temos que clicar para reiniciar o jogo, carregando a cena novamente.

```
if(Vivo == false)
{
    if(Input.GetButtonDown("Fire1"))
    {
```

```
SceneManager.LoadScene("game");  
    }  
}
```

Sim! Podemos fazer `if` dentro de `if`, bem legal né?

O nome da minha Cena é **game** lembre-se de substituir pelo nome que você deu para a sua.

Agora só falta uma pequena linha, quando reiniciamos nosso jogo ele está iniciando pausado, para isto ser resolvido temos que voltar o `timeScale` para `1` então no **Start** deste *Script* vamos digitar a linha.

```
Time.timeScale = 1;
```

Você pode fazer download do projeto Unity desta aula [clcando aqui \(https://github.com/alura-cursos/unity-shooter/raw/master/Final_Aula4.zip\)](https://github.com/alura-cursos/unity-shooter/raw/master/Final_Aula4.zip).