

## Balanceamento do Jogo

Você pode fazer download do projeto Unity como ele está ao final desta aula [clcando aqui](https://github.com/alura-cursos/unity-shooter-3/raw/master/Final_Aula03.zip) ([https://github.com/alura-cursos/unity-shooter-3/raw/master/Final\\_Aula03.zip](https://github.com/alura-cursos/unity-shooter-3/raw/master/Final_Aula03.zip)).

Limitamos o número de zumbis que podem ser criados, mas isso fez com que novos zumbis parem de surgir já que nunca diminuimos a quantidade de zumbis vivos.

Para fazer isso, temos que fazer a verificação por gerador. Cada gerador vai cuidar dos seus zumbis e verificar se deve ou não criar mais.

Então temos que atrelar o zumbi ao gerador, e para isso vamos criar uma variável no *script* de `ControlaInimigo` que guarda o gerador daquele zumbi:

```
public GeradorZumbis meuGerador;
```

Ela é pública, porque a melhor forma de preencher essa variável é no momento que criamos o zumbi no *script* `GeradorZumbis`. Vá até a linha do `Instantiate` e vamos fazer a troca por:

```
ControlaInimigo zumbi = Instantiate(Zumbi, posicaoDeCriacao, transform.rotation).GetComponent<ControlaInimigo>();  
zumbi.meuGerador = this;
```

Ou seja, no momento que criamos o zumbi já guardamos seu *script* de `ControlaInimigo`. Também preenchemos a variável criada anteriormente com esse gerador, então agora esse zumbi sabe de qual gerador ele veio.

Temos então que criar no `GeradorZumbis` um método para fazer o gerador perder um número na quantidade de zumbi vivos, quando nosso zumbi morrer.

```
public void DiminuirQuantidadeZumbisVivos ()  
{  
    quantidadeDeZumbisVivos --;  
}
```

Este método deve ser chamado no método `Morrer` do *script* `ControlaInimigo`:

```
meuGerador.DiminuirQuantidadeZumbisVivos();
```

Agora que estamos fazendo os geradores funcionarem de forma adequada, vamos fazer com que quantidade de zumbis máximos possa aumentar com o tempo de jogo, aumentando assim a dificuldade.

Temos que criar então duas variáveis, uma para guardar de quanto em quanto tempo vamos aumentar a dificuldade, e uma outra para marcar quando devemos fazer essa alteração como um contador.

```
private float tempoProximoAumentoDeDificuldade = 30;  
private float contadorDeAumentarDificuldade = 0;
```

Basta utilizarmos essas variáveis no `Update` para fazermos os contadores. Vamos ver um contador diferente do que estamos acostumados: neste, vamos utilizar o `Time.timeSinceLevelLoad` ao invés de utilizarmos do `Time.deltaTime` do Unity e atribuí-lo para uma variável.

Nós vamos utilizar o próprio `Time.timeSinceLevelLoad` do Unity para contar o tempo, e só conferir se o tempo que a fase está chegando alcançou o momento de aumentar a dificuldade.

Em seguida, atualizamos o tempo do próximo aumento e a quantidade máxima de zumbis daquela gerador em `1`.

```
if(Time.timeSinceLevelLoad > contadorDeAumentarDificuldade)
{
    contadorDeAumentarDificuldade = Time.timeSinceLevelLoad + tempoProximoAumentoDeDificuldade;
    quantidadeMaximaDeZumbisVivos++;
}
```

Lembre-se que não queremos aumentar a dificuldade no `0` e sim no primeiro tempo de aumento, então temos que inserir uma linha no `Start`.

```
contadorDeAumentarDificuldade = tempoProximoAumentoDeDificuldade;
```