

Mantendo a Vida Máxima

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_Aula01.zip) (https://github.com/alura-cursos/unity-shooter-3/raw/master/Final_Aula01.zip).

Já estamos pegando o nosso Kit e recuperando vida, só temos um problema: a vida que pegamos pode superar a nossa vida máxima passando dos 100 que colocamos como o valor padrão inicial. Temos então que limitar a nossa vida para este valor que definimos no início. Para isso, vamos verificar se o valor da vida que ganhamos ultrapassa a vida inicial e depois, em outra linha, manteremos no valor da vida inicial.

```
public void CurarVida(int quantidadeDeCura)
{
    statusJogador.Vida += quantidadeDeCura;
    if(statusJogador.Vida > statusJogador.VidaInicial)
    {
        statusJogador.Vida = statusJogador.VidaInicial;
    }
    scriptControlaInterface.AtualizarSliderVidaJogador();
}
```

Também há outras boas formas de fazer isso, como utilizar o [Mathf.Clamp](https://docs.unity3d.com/ScriptReference/Mathf.Clamp.html) (<https://docs.unity3d.com/ScriptReference/Mathf.Clamp.html>) se você quiser pesquisar.

Com o nosso Kit pronto, temos que fazer ele soltar a partir dos zumbis, certo? Quando o zumbi morrer ele gerar um Kit Médico.

Vamos no zumbi então e criar uma variável que fará referência ao objeto:

```
public GameObject PrefabKitMedico;
```

E vamos gerar um Kit quando nosso zumbi morrer:

```
Instantiate(PrefabKitMedico, transform.position, Quaternion.identity);
```

Nossos zumbis agora geram os objetos, legal! Só que temos um problema: não é para cada zumbi gerar um quando morre, isso é estranho. O ideal seria calcular um valor aleatório e se basear neste valor para gerar ou não o objeto.

Para fazer isto, vamos criar uma variável equivalente a chance em porcentagem de o nosso zumbi gerar um Kit Médico.

```
private float porcentagemGerarKitMedico = 0.1f;
```

A partir dessa porcentagem, vamos calcular se devemos ou não gerar o Kit. Vamos criar um método para isolar esse comportamento?

```
void VerificarGeracaoKitMedico (float porcentagemGeracao)
{
    if(Random.value <= porcentagemGeracao)
    {
```

```
Instantiate(PrefabKitMedico, transform.position, Quaternion.identity);  
    }  
}
```

Lembre-se que o [Random.value](https://docs.unity3d.com/ScriptReference/Random-value.html) (<https://docs.unity3d.com/ScriptReference/Random-value.html>) gera um valor aleatório que vai de 0 a 1, então se o valor gerado for menor que 0.1(10%) nós criamos o objeto.

Agora é só uma questão de chamar este método quando o zumbi morrer.

```
GeracaoKitMedico(porcentagemGerarKitMedico);
```

Legal! Agora geramos o Kit aleatoriamente mas temos um novo problema: quando o Kit é gerado, ele fica lá para sempre até ser pego isso faz com o jogador possa manter vários Kits para facilitar o Jogo. Então vamos fazer eles se autodestruírem em alguns segundos?

Então vamos criar uma variável para isso no nosso kit:

```
private float tempoDestruir = 5;
```

E vamos no `Start` falar para o kit se autodestruir em 5 segundos, que é valor atual da nossa variável:

```
void Start()  
{  
    Destroy(gameObject, tempoDestruir);  
}
```