

Game Over ao Perder toda a vida

Já temos o método e a chamada do método para dar dano ao nosso personagem. Vamos melhorar este método utilizando parâmetros. Os parâmetros fazem com que possamos passar informações para o método dinamicamente, podendo trocar o valor informado a cada chamada se quisermos.

Vamos então criar um parâmetro dentro dos parênteses do nosso método:

```
public void TomarDano (int dano)
{
    Vida -= dano;
}
```

Além disso, vamos criar uma variável que gera um valor aleatório de dano para utilizarmos na chamada do método. Assim, se o jogador tiver muita sorte, ele pode perder com 5 ataques, mas o normal seria ele perder com 4.

Para isso vamos utilizar o `Random.Range` que é um método do Unity que gera números aleatórios para nós, de um mínimo até um máximo.

```
void AtacaJogador ()
{
    int dano = Random.Range(20, 30);
    Jogador.GetComponent<ControlaJogador>().TomarDano(dano);
}
```

As linhas de Game Over não fazem mais sentido no método `AtacaJogador` do nosso inimigo, já que elas tem que ser atreladas a vida do nosso personagem. Então, vamos trazer estas linhas para o código do personagem. Não precisamos buscar pelo personagem, já que essas variáveis são deste script. Lembre-se de testar se a vida é menor ou igual a zero para dar o *Game Over*:

```
public void TomarDano (int dano)
{
    Vida -= dano;
    if(Vida <= 0)
    {
        Time.timeScale = 0;
        TextoGameOver.SetActive(true);
    }
}
```