

## Partícula no código

Você pode fazer download do projeto Unity como ele está ao final do curso [clikando aqui \(https://github.com/alura-cursos/unity-shooter-4/raw/master/Final\\_Aula04.zip\)](https://github.com/alura-cursos/unity-shooter-4/raw/master/Final_Aula04.zip).

Agora que já temos a partícula, podemos utilizar a programação para chamá-la. Para isso, podemos mudar nossa interface para implementar posição e rotação para nossas partículas.

Vamos então criar um novo método para Instanciar a partícula no nosso Zumbi e Chefe:

```
public void ParticulaSangue (Vector3 posicao, Quaternion rotacao)
{
    Instantiate(ParticulaSangueZumbi, posicao, rotacao);
}
```

Lembre-se que `ParticulaSangueZumbi` é uma variável que deve ser declarada, e tem que ser preenchida com o Prefab da Partícula para assim funcionar.

Para termos essa posição e rotação, vamos na bala e modificar algumas coisas.

A primeira é calcular uma rotação para a partícula. Para o sangue sair para a direção oposta, essa rotação da partícula deve ser o inverso da frente da nossa bala. Faça isso no `OnTriggerEnter` :

```
Quaternion inversaoDoSentidoDeRotacao = Quaternion.LookRotation(-transform.forward);
```

Assim, quando tomamos o dano, podemos utilizar essa variável e o `transform.position` como posição para criar a bala no local correto, ou seja, no momento que o personagem for atingido. Como temos que executar duas ações nos nossos Inimigos podemos salva-los numa variável e chamar os dois métodos em seguida:

```
switch(objetoDeColisao.tag)
{
    case "Inimigo":
        ControlaInimigo inimigo = objetoDeColisao.GetComponent<ControlaInimigo>();
        inimigo.TomarDano(1);
        inimigo.ParticulaSangue(transform.position, rotacaoOpostaABala);
        break;
    case "ChefeDeFase":
        ControlaChefe chefe = objetoDeColisao.GetComponent<ControlaChefe>();
        chefe.TomarDano(1);
        chefe.ParticulaSangue(transform.position, rotacaoOpostaABala);
        break;
}
```