

## Iniciando o novo gerador de Inimigos

Vamos começar a trabalhar num novo modelo de gerador de inimigos? Esse modelo é para não termos problema com o zumbi ser criado dentro de outros objetos ou zumbis.

Para isso, vamos no nosso script **GeradorZumbis** e vamos começar a trabalhar nele.

Vamos iniciar extraíndo o `Instantiate` do `Update` e criando um método para isso, já que vamos começar a trabalharativamente nesse Gerador.

```
void Update () {
    contadorTempo += Time.deltaTime;

    if(contadorTempo >= TempoGerarZumbi)
    {
        GerarNovoZumbi();
        contadorTempo = 0;
    }
}

void GerarNovoZumbi ()
{
    Instantiate(Zumbi, posicaoDeCriacao, transform.rotation);
}
```

Assim como fizemos com o nosso zumbi, vamos gerar uma posição aleatória para criar outros zumbis e assim testar algumas posições e verificar se elas estão ocupadas.

Para fazer isso vamos criar um método que gera uma posição aleatória baseada na posição do Gerador.

```
Vector3 AleatorizarPosicao ()
{
    Vector3 posicao = Random.insideUnitSphere * 3;
    posicao += transform.position;
    posicao.y = 0;

    return posicao;
}
```

Agora temos que utilizar este método para gerar uma posição aleatoriamente.

```
void GerarNovoZumbi ()
{
    Vector3 posicaoDeCriacao = AleatorizarPosicao();

    Instantiate(Zumbi, posicaoDeCriacao, transform.rotation);
}
```

Agora que estamos gerando uma posição aleatória, temos que verificar se esta posição está sendo ocupada por algum zumbi. Para isso vamos utilizar o método `Physics.OverlapSphere` da Unity.

Ele testa pegando uma posição e criando uma esfera naquela local, também salvando todos os colisores numa coleção.

Por isso temos que criar uma variável diferente: uma variável que guarda uma coleção de colisores e é conhecida como `Array`.

É legal também recebermos só os objetos que queremos. Como só queremos os zumbis, vamos criar uma variável de Camada para eles.

```
public LayerMask LayerZumbi;
```

E vamos colocar os zumbis numa camada separada e marcar na variável.

Agora podemos utilizar o método de verificar se há zumbis na área. Para isso, vamos levar em consideração a posição que iremos criar o zumbi e um raio de esfera de `1`, que é um valor parecido com o raio do colisor zumbi.

E vamos salvar os objetos encontrados num array de colisores.

```
void GerarNovoZumbi ()
{
    Vector3 posicaoDeCriacao = AleatorizarPosicao();
    Collider[] colisores = Physics.OverlapSphere(posicaoDeCriacao, 1, LayerZumbi);

    Instantiate(Zumbi, posicaoDeCriacao, transform.rotation);
}
```

Temos que agora testar se esse nosso array de colisores tem algum elemento, pois se ele tem significa que temos um zumbi naquela área e temos que gerar outra posição para criação e verificar a área novamente. Lembre-se que estamos utilizando um array porque podemos ter vários zumbis, e um array nesse caso vai guardar exatamente isso, vários colisores de zumbis numa mesma variável.

Para verificar se temos algum zumbi no array, usamos a variável `length`, que nos devolve quantos elementos nossa coleção tem.

```
void GerarNovoZumbi ()
{
    Vector3 posicaoDeCriacao = AleatorizarPosicao();
    Collider[] colisores = Physics.OverlapSphere(posicaoDeCriacao, 1, LayerZumbi);

    if(colisores.Length > 0)
    {
        posicaoDeCriacao = AleatorizarPosicao();
        colisores = Physics.OverlapSphere(posicaoDeCriacao, 1, LayerZumbi);
    }

    Instantiate(Zumbi, posicaoDeCriacao, transform.rotation);
}
```

