

## Zumbi olhar para o Jogador

Nosso Zumbi está nos perseguindo mas ele não olha pro nosso jogador quando está fazendo isso nem para quando está perto, como podemos realizar essa alteração?

No nosso código temos que fazer o Zumbi olhar para a direção que ele está se movendo para isso vamos construir uma variável que valerá essa rotação que ele tem que ter quando olha para esta posição.

Dentro do `FixedUpdate` do código **ControleInimigo** crie a variável `Quaternion novaRotacao;` .

```
7 public GameObject Jogador;
8 public float Velocidade = 5;
9
10 // Use this for initialization
11 void Start () {
12
13 }
14
15 // Update is called once per frame
16 void FixedUpdate () {
17     Vector3 direcao = Jogador.transform.position - transform.position;
18
19     GetComponent<Rigidbody>().MovePosition
20     (GetComponent<Rigidbody>().position + direcao.normalized * Velocidade * Time.deltaTime);
21
22 }
23
24 }
25
```

No Unity rotação é uma variável do tipo `Quaternion` , que são estruturas baseadas em números complexos e tem muitas facilidades ao utilizar eles para a rotação, mas não se preocupe todos os cálculos que vamos fazer para chegar nessas rotações serão feitos pelo próprio Unity.

Temos que fazer o Zumbi olhar para a direção que ele está se movendo e já temos esta direção através da variável `direcao` então vamos utilizar uma estrutura que o Unity nos fornece que calcula uma rotação através de uma posição no nosso código passando este valor para a variável que criamos anteriormente atualizando a linha para:

```
Quaternion novaRotacao = Quaternion.LookRotation(direcao);
```

Aqui temos a nossa variável de rotação sendo igual a uma estrutura definida dentro da parte de rotação do Unity, o *\*Quaternions* e chamando um método chamado `LookRotation` que dentro dos parênteses recebe uma posição para qual queremos calcular a rotação.

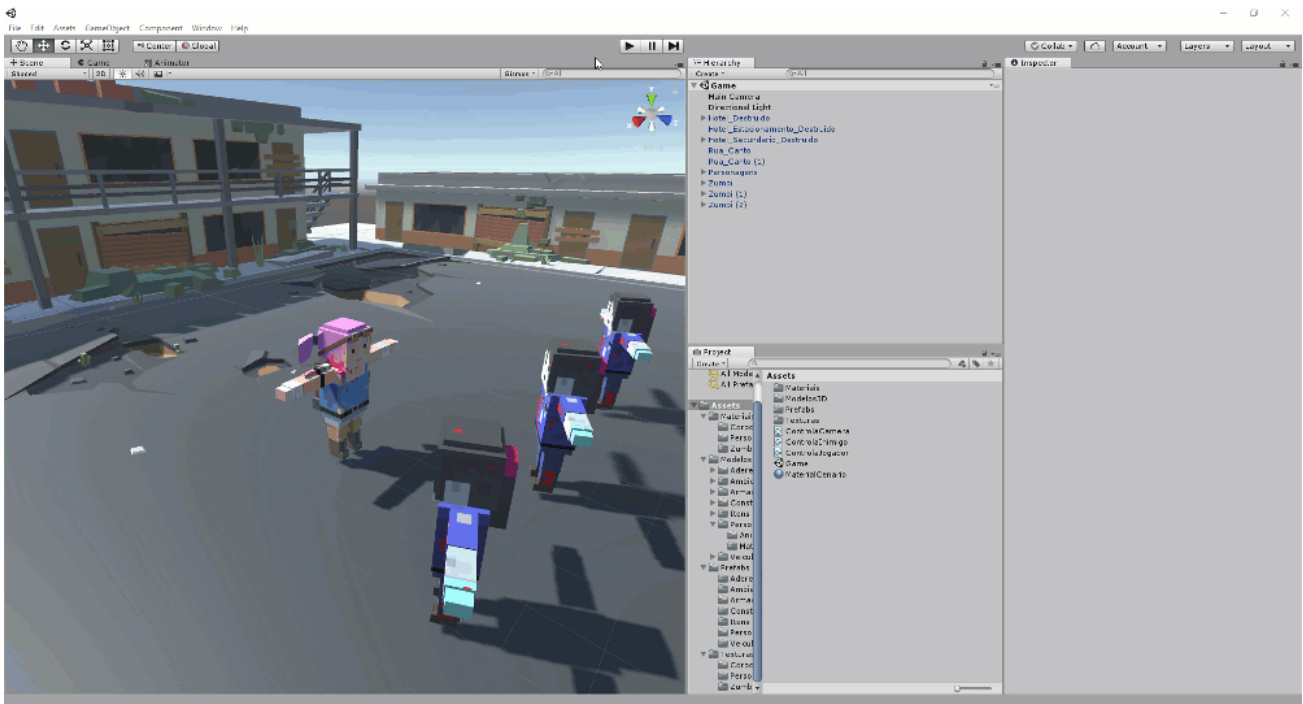
Então temos guardado na variável `novaRotacao` o valor de rotação que nosso Zumbi tem que ter para olhar na direção que ele está indo, agora só temos que passar esta rotação para a rotação do nosso Zumbi.

Para isso entrar no **Rigidbody** do Zumbi afinal se estamos movimentando pela física é ideal rotacionarmos também. Dentro do **Rigidbody** vamos utilizar o método de rotação `MoveRotation` que recebe um `Quaternion` de rotação para

rodar o personagem, então no nosso código temos:

```
GetComponent<Rigidbody>().MoveRotation(novaRotacao);
```

Aqui temos nosso código finalizado e pronto agora para o Zumbi andar e olhar para o personagem.



Agora para ele parar quando está perto basta colocar este código dentro de um `if` para ele parar quando chegar a uma certa distância.

Calculamos a distância utilizando o método `Vector3.Distance` que a Unity fornece que calcula o valor entre duas posições.

```
float distancia = Vector3.Distance(transform.position, Jogador.transform.position);
```

Agora coloque o seu código dentro de um `if` que testa se o inimigo está a uma certa distância:

```
void FixedUpdate ()
{
    float distancia = Vector3.Distance(transform.position, Jogador.transform.position);

    if (distancia > 2.5)
    {
        Vector3 direcao = Jogador.transform.position - transform.position;
        GetComponent<Rigidbody>().MovePosition(GetComponent<Rigidbody>().position + direcao.normalized * 10f);

        Quaternion novaRotacao = Quaternion.LookRotation(direcao);
        GetComponent<Rigidbody>().MoveRotation(novaRotacao);
    }
}
```

