02

Refatoração da movimentação

Nossos *scripts* estão ficando grandes, e logo podemos perder o controle de onde está cada feature. Então vamos trabalhar na refatoração de algum deles? Quebrar os elementos em métodos e manter tudo conciso e organizado.

Vamos começar pelo *script* do inimigo?

Temos que quebrar comportamentos em métodos. Mais do que isso, uma das boas práticas da programação é tentar não escrever a mesma coisa duas vezes. Ambos, inimigo e personagem, andam igual; vamos então quebrar isso em um código diferente e usar nos dois?

Crie o *script* **MovimentoPersonagem**, e vamos começar a definir um método de movimentar com as linhas de movimentação do inimigo só que passando dois parâmetros: a direção e velocidade.

```
public class MovimentoPersonagem : MonoBehaviour {

    public void Movimentar (Vector3 direcao, float velocidade)
    {
        rigidbodyJogador.MovePosition (rigidbodyJogador.position + (direcao * velocidade * Time
    }
}
```

Faltou a variável do **Rigidbody**, que vamos colocar no método `Awake` para garangir que ela rode antes do método `Start`, permitindo que o personagem já se movimente antes.

```
public class MovimentoPersonagem : MonoBehaviour {

    private Rigidbody rigidbodyJogador;
    void Awake()
    {
        rigidbodyJogador = GetComponent<Rigidbody>();
    }

    public void Movimentar (Vector3 direcao, float velocidade)
    {
        rigidbodyJogador.MovePosition (rigidbodyJogador.position + (direcao.normalized * velocid
    }
}
```

Legal! Agora é só irmos no nosso Inimigo e chamar este método no **ControlaInimigo**. Para isso vamos jogar esse *script* para o Inimigo, e vamos criar uma referência a ele guardando numa variável.

```
private MovimentoPersonagem movimentoPersonagem;
```

E no `Start` vamos preencher essa variável:

```
movimentoPersonagem = GetComponent<MovimentoPersonagem>();
```

Agora toda aquela linha de movimentação será substituída por uma chamada:

```
movimentoPersonagem.Movimentar(direcao, Velocidade);
```

Vamos fazer a mesma coisa com a rotação agora no script **MovimentoPersonagem**, e vamos criar um método para a rotação:

```
public void Rotacionar (Vector3 direcao)
{
        Quaternion novaRotacao = Quaternion.LookRotation(direcao);
        rigidbodyJogador.MoveRotation(novaRotacao);
}
```

E fazemos a troca da parte da rotação de rotação no script **ControlaInimigo** para:

```
movimentoPersonagem.Rotacionar(direcao);
```