

Para saber mais - Magic Number e Literal Strings

Números Mágicos

Algo que pode acontecer comumente no nosso código é deixar números soltos, que não são atrelados a nenhuma variável, para gerar algum comportamento. Isso é conhecido como **Número Mágico (Magic Number)**, eles são uma péssima prática de programação.

Para evitarmos os **Magic Number** o ideal é sempre extrairmos esse comportamento para uma variável, por exemplo, no nosso código da `Bala` quando vamos dar dano aos nossos Zumbis nós passamos o valor de `1`.

Se pensarmos a curto prazo pode ser algo simples, mas ao fazer isso várias vezes se torna um vício que é melhor ser corrigido no início.

Então o ideal é evitarmos totalmente essa prática sempre extraindo estes números para variáveis, neste caso da `Bala` podemos fazer:

```
private int danoDoTiro = 1
```

E ao chamar o método fazemos uso dessa variável:

```
inimigo.TomarDano(danoDoTiro);
```

Assim temos mais controle e facilita a busca pelo local onde estamos definindo quanto de dano nossa bala dá no personagem.

String Literais

Outro péssimo comportamento no nosso código é o uso de **Strings Literais (Literal Strings)** fazemos isso ao utilizar as Tags, por exemplo. Toda vez que utilizamos uma Tag no nosso código nós escrevemos seu nome:

```
Jogador = GameObject.FindWithTag("Jogador");
```

Isso é ruim porque se errarmos uma letra na hora de digitar fica difícil de identificar o erro ou se queremos alterar o nome da Tag temos que mudar em vários locais do código.

Novamente o ideal seria extrair isto para uma variável mas no caso das Tags vamos praticar uma abordagem bastante comum, a de gerar uma classe para as Tags.

O que vamos fazer é criar um novo script para guardar o nome das Tags evitando assim que tenhamos que digitar toda vez e nos dando a facilidade de mudar o nome das Tags quando queremos.

```
public class Tags : MonoBehaviour
{
}
```

E vamos criar uma variável constante (`const`) para cada Tag, o modificador constante nos dá uma facilidade de acesso fácil assim como o `static` mas a variável não pode trocar de valor em durante a execução do jogo. Como uma

Tag é algo que não muda constantemente de nome e nem em execução este modificador é perfeito para este uso.

```
public class Tags : MonoBehaviour
{
    public const string Jogador = "Jogador";
}
```

Pronto! Agora toda vez que precisamos utilizar alguma Tag é só incluir a variável neste script e usa-la da seguinte forma:

```
Jogador = GameObject.FindWithTag(Tags.Jogador);
```