

Programação do menu

Agora que já temos o nosso menu montado, vamos começar a trabalhar com a programação do menu. Vamos começar fazendo os métodos que temos que fazer para os dois botões funcionarem. Que tal iniciarmos pelo "Jogar"? Para não misturar nossos *scripts*, vamos criar um outro código chamado `ControlaInterfaceMenu`. Note que o código de `ControlaInterface` tem várias dependências de `Sliders` de vida entre, outras coisas, e isso pode confundir. Ao invés de sempre rodar pra uma cena como no método `Reiniciar` que temos no outro *script*, podemos fazer algo mais genérico: um método que troca para Cena que queremos mediante um parâmetro.

```
void MudarCena (string nomeDaCena)
{
    SceneManager.LoadScene(nomeDaCena);
}
```

E vamos criar mais um para jogar o jogo.

```
public void JogarJogo ()
{
    MudarCena("game");
}
```

Agora é só colocarmos essas funções para serem chamadas no botão de Jogar através do `OnClick`

Agora no botão de sair vamos fazer algo diferente. Como estamos mirando como plataforma WebGL (que funciona no browser), na teoria não temos a necessidade de um botão de sair, mas talvez você queira enviar seu jogo para outras plataforma. Então para não termos o botão o tempo todo, mas somente em algumas plataformas, vamos colocar uma configuração conhecida como **Compilação dependente da plataforma (Platform dependent compilation)**.

Vamos então desativar o botão, e mostrá-lo somente em algumas plataformas. Vamos mostrá-lo aqui no editor também, para que possamos testá-lo.

```
void Start ()
{
    #if UNITY_STANDALONE || UNITY_EDITOR
        botaoSair.SetActive(true);
    #endif
}
```

Agora com o botão funcionando no editor, vamos criar um método para que possamos sair do nosso jogo, tanto no editor quanto no jogo compilado.

```
public void SairDoJogo ()
{
    Application.Quit();
    #if UNITY_EDITOR
        UnityEditor.EditorApplication.isPlaying = false;
    #endif
}
```

```
#endif
```

```
}
```

Vamos também colocar este método para ser chamado no botão como fizemos anteriormente!