05

Game Loop

Transcrição

Agora já temos tudo pronto. Podemos desenvolver o nosso **Jogo da Forca**.

Conhecendo o jogo

O jogo da forca também é um jogo de adivinhação, mas no caso o usuário deve adivinhar uma **palavra secreta**. No arquivo **forca.py**, já há a função `jogar` definida, onde ficará o código do nosso jogo:

```
def jogar():
    print("*********************************")
    print("***Bem vindo ao jogo da Forca!***")
    print("*********************************")

    print("Fim do jogo")
```

E caso executemos o programa como programa principal, executamos essa função `jogar`. Verificamos isso através de um `if`:

```
if(__name__ == "__main__"):
    jogar()
```

Definindo a palavra secreta

Como no jogo devemos adivinhar uma palavra secreta, nada mais justo do que defini-la em uma variável. Por enquanto a palavra será fixa, com o nome de **banana**:

```
def jogar():
    print("*********************************")
    print("***Bem vindo ao jogo da Forca!***")
    print("*********************************")

    palavra_secreta = "banana"

    print("Fim do jogo")
```

Mais à frente deixaremos essa palavra secreta mais dinâmica.

Primeiros passos do jogo

Agora, enquanto o usuário estiver jogando, devemos ficar verificando se o usuário acertou a palavra secreta ou se ele perdeu (se "enforcou"), mas devemos fazer isso em quantas iterações?

O usuário continua jogando enquanto ele não acertar a palavra e enquanto ele ainda tiver tentativas para acertar a palavra, ou seja, enquanto ele não "se enforcar". Para representar esse loop, já conhecemos o `while`:

```
palavra_secreta = "banana"

while(não acertou E não enforcou):
    print("Jogando...")
```

Mas o que colocamos como condição do `while`, está em português e não existe em Python, logo isso não funcionará.

No caso de se enforcar, podemos ter dois valores, ou o usuário se enforcou, ou não. Logo, podemos ter os valores **verdadeiro** ou **falso**.

O tipo bool

Nas linguagens de programação, e no Python não é diferente, há um tipo específico para representar esses valores, o tipo `bool`. Ele pode ter os valores `True` (**Verdadeiro**) e `False` (**Falso**). Então vamos definir uma variável `enforcou`, que começará com o valor `False`:

```
palavra_secreta = "banana"

enforcou = False

while(não acertou E não enforcou):
    print("Jogando...")
```

Criaremos também a variável `acertou`, que representa se o usuário acertou ou não a palavra. Ela também começará com o valor `False`, pois quando o jogo começa, o usuário ainda não acertou a palavra secreta:

```
palavra_secreta = "banana"

enforcou = False
acertou = False

while(não acertou E não enforcou):
    print("Jogando...")
```

Agora podemos modificar a condição do `while`, mas a condição é **não** acertar a palavra e **não** se enforcar. Para representar o **não**, no Python existe a palavra chave de negação, o `not`. E para representar o `E`, existe o operador lógico `and`:

```
palavra_secreta = "banana"

enforcou = False
acertou = False

while(not acertou and not enforcou):
    print("Jogando...")
```

Então, enquanto não acertamos a palavra e enquanto não nos enforcamos, continuamos jogando.

Daremos continuação à programação do nosso jogo no próximo capítulo :)