

EDUARDO CUNHA

CodeNCode

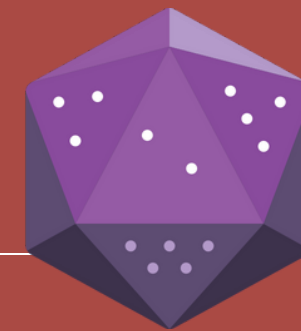
Linguagem de programação de RPG

Motivações



Hobby Pessoal

RPG é um hobby pessoal muito importante para mim, sendo um dos meus principais hobbies hoje.



Melhorar a Experiência

A ideia da linguagem é permitir que o “Mestre” (responsável por coordenar o jogo), consiga se preocupar menos em alguns detalhes técnicos, agilizando o ritmo do jogo

Características

Tipada

A linguagem é tipada, os possíveis tipos são INT e STRING

Funções

CnC possui funções, porém não é possível criar funções dentro de funções

Condições e Loops

Além de funções, também estão presentes condicionais e loops.

Função de Rolar Dado

CnC, possui uma função que simula o rolar de um dado, basta fornecer qual tipo de dado deve ser rolado.

Curiosidades

- O nome da linguagem é uma referência ao principal RPG moderno “Dungeons and Dragons (DnD)”
- Inicialmente, os jogos de RPG eram vistos como algo “ruim” ou do mal.
- Os jogos de RPG modernos surgiram em meados do século XX, se popularizando principalmente no final desse século
- Atualmente, além de ser possível jogar RPG, popularizou-se a praticar de assistir RPG por meio de lives e vídeos,

Exemplo 1

Nesta região declaramos as variáveis do programa.

```
def rollAndAttack(diceSides int, attackedEnemy string, attackedEnemyHealth int) int {
    int damage = roll(diceSides);
    print(attackedEnemy . " got hit! Damage dealt: " . damage);

    act attackedEnemyHealth - damage;
}

string knight = "Charles";
string enemy = "Dragon";
int enemyHealth = 10;

enemyHealth = rollAndAttack(10, enemy, enemyHealth);
print("Enemy is now at: " . enemyHealth . " life!");
```

```
C:\Users\eduar\Insper\setimo\logcomp\logcomp-projeto>python main.py input-0.cnc
Dragon got hit! Damage dealt: 5
Enemy is now at: 5 life!
```

Aqui declaramos uma função que aleatoriza um valor de um dado de “diceSides” lados, e então retira esse valor da vida de um inimigo, simulando um ataque no jogo.

Podemos ver que neste caso, o valor no dado foi 6, e o inimigo tomou este dano corretamente

Exemplo 2

Aqui temos o uso de um loop, o qual acontece enquanto a vida do gigante ou do feiticeiro for maior que 0, além disso, a variável time é atualizada, informação que é útil para o mestre administrar efeitos que dependem de tempo.

```
string sorcerer = "Daar";
string giant = "Fenir";
int sorcererLife = 5;
int giantLife = 10;

int attackRoll;
int time;

combat; time = 0; while sorcererLife > 0 && giantLife > 0 progress time = time + 1 {
    print("Current combat time: " . time . "s");
    print("");

    if (time % 2 == 0) {
        print(sorcerer . "'s turn");
        attackRoll = roll(8);
        giantLife = giantLife - attackRoll;
        print(sorcerer . " attacked " . giant . " with " . attackRoll . " points of damage!");
    } else {
        print(giant . "'s turn");
        attackRoll = roll(4);

        sorcererLife = sorcererLife - attackRoll;
        print(giant . " attacked " . sorcerer . " with " . attackRoll . " points of damage!");
    }
}
if (sorcererLife > 0) {
    print("The monster was defeated!");
} else {
    print("The player was defeated!");
}

print("Battle ended");
```

```
C:\Users\eduar\Insper\setimo\logcomp\logcomp-projeto>python main.py in
Current combat time: 0s

Daar's turn
Daar attacked Fenir with 5 points of damage!
Current combat time: 1s

Fenir's turn
Fenir attacked Daar with 3 points of damage!
Current combat time: 2s

Daar's turn
Daar attacked Fenir with 6 points of damage!
The monster was defeated!
Battle ended
```

Por fim é verificado quem venceu o combate.



OBRIGADO