

A stylized fantasy landscape illustration. In the upper left, a black dragon flies against a pale yellow sky. Several small black birds are scattered across the sky. On the right, a large, bright yellow sun is partially obscured by a black silhouette of a castle with multiple spires. The foreground features dark green and black silhouettes of mountains and trees. The overall color palette is muted, with earthy tones and a soft, hazy atmosphere.

DnD Language

Eric Possato

Table of contents

The background features a warm, orange-toned sky with stylized white clouds. Several small black birds are scattered across the sky. In the foreground, there are dark, silhouetted mountains and a dark body of water. The overall aesthetic is minimalist and artistic.

01

Motivação

02

Características

03

Curiosidades

04

Exemplos

Motivação

- Linguagem baseada e em Dungeons & Dragons, feita para simular comportamento de um RPG de mesa
- Como todos componentes de RPGs podem ser representados por texto e números, uma implementação de uma implementação de uma linguagem como essa acaba sendo similar a uma linguagem de programação comum

Características

- Atribuição de variáveis (tipagem)
 - *narration* (texto)
 - *stat* (inteiros)
- Operações matemáticas de soma, subtração, multiplicação e divisão
- Condicionais com operação *check*
- Loop com operação *turns*
- Output com *say()*
- Randomização para simular rolagem de dados *roll()*

Curiosidades

- O jogo DnD possui condicionais na forma de *Skill Checks* onde o jogador precisa superar um valor estipulado com um d20 + modificadores, todos estes se resumem a comparações de \geq e foram feitos com DC
- Como toda ação possui resultados, tanto em sucesso quanto falha, a linguagem exige uma condição de *success* e *consequence* em um *check*, mesmo que estejam vazias

Exemplos

Variavel stat e narration

```
stat str = 12
stat dice
stat enemyHP = 10
narration aviso = "Cuidado!"
```

check

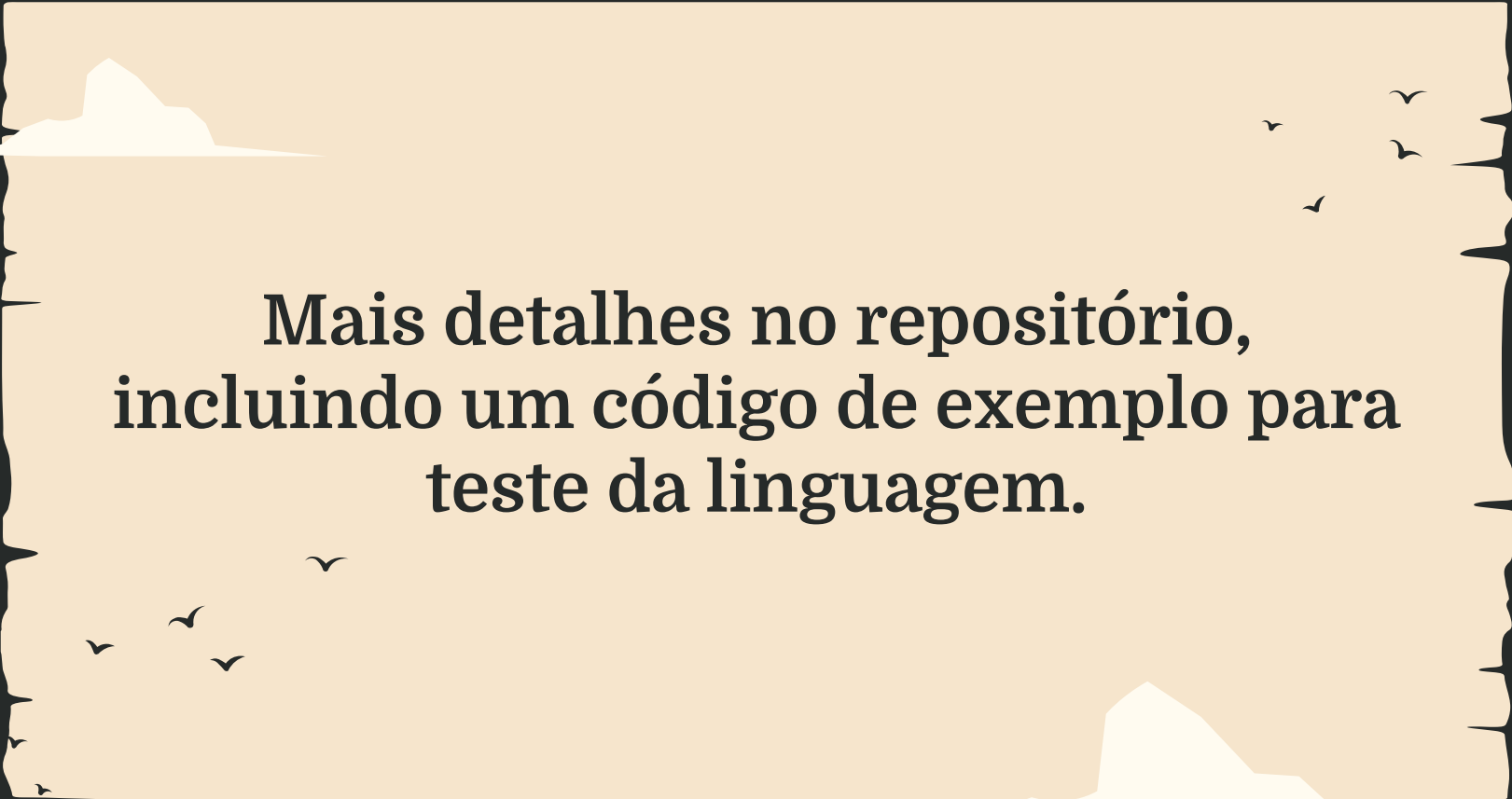
```
check dice DC 10
success
  say("Sucesso")
consequence
  say("Falhou")
rest
```

roll

Para simular um dado
de 6 lados:
dice = roll(6)

Loop turns

```
turns (enemyHP DC 0) action
  dice = roll(6) + STR
  enemyHP = enemyHP - dice
  say(enemyHP)
rest
```



**Mais detalhes no repositório,
incluindo um código de exemplo para
teste da linguagem.**