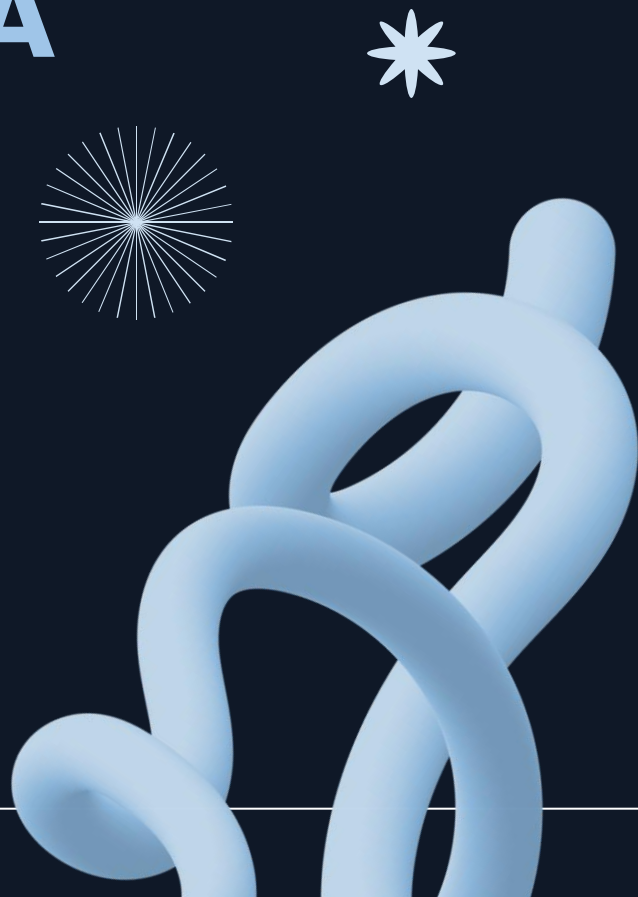

CRIAÇÃO DE UMA Linguagem de programação

Motivações, curiosidades, etc.



MOTIVAÇÃO



O que me motivou?

A escolha do tema de Hollow Knight foi baseada em gostos pessoais pelo jogo. Criar uma linguagem de programação capaz de identificar se um inimigo morreu após alguns ataques é algo funcional.



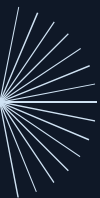
Por que Hollow Knight?

Hollow Knight é um jogo repleto de funcionalidades, mecânicas e inimigos. Fazer uma linguagem capaz de identificar isso é algo desafiador.



CARACTERÍSTICAS

PROGRAMA	CONDICIONAIS	GAMELOOP	FUNÇÕES	RESULTADO
O programa sempre deve ser iniciado pela declaração de variáveis começando com "soul" e finalizando com "eaten"	A condicional da linguagem simula um "if", onde entrará apenas se a condição seguida por "lifeislessthan" que indica se a vida do inimigo é menor que o número a seguir.	Os loops funcionam similarmente ao "While", em que há uma condição de parada.	Há criação e chamada de funções. A criação é feita com a palavra inicial "interaction" e a chamada com a palavra "action".	O resultado que será printado no terminal é a condição da vida do inimigo "ALIVE" ou "DEAD" e a vida dele caso ainda esteja vivo.
01	02	03	04	05



EXEMPLOS

```
souls
player:hollowknight lives
enemy:aspid lives
eaten
if aspid lifeislessthan 21
    begin hollowknight uses purenail causes playerdmg being 1 to aspid
empty
interaction:life_condition teste1 entities enemy:enemy1, player:player1
purevessel
    usingsoul 20
        begin player1 uses purenail causes playerdmg being 1 to enemy1
        5
    endusingsoul
    return enemy1
sealedvessel
action teste1 entities aspid, hollowknight done
```

No exemplo ao lado há algumas coisas que devem ser notadas:

- Há a declaração de duas variáveis hollowknight e aspid dos tipos player e enemy, respectivamente;
- Há uma condicional que será executada somente se a vida de **aspid** for menor que 21, sendo esse um valor implícito no código do compilador;
- Há uma criação de função em “interaction” de nome “teste1” que irá ser responsável por atribuir o ataque de **player1** à **enemy1**;
- Ainda na função **teste1**, ela utiliza um **gameloop** que irá rodar **20/5** vezes, realizando sempre um **ATAQUE** que é indicado sempre por **begin**.
- No fim, a função **teste1** é chamada e seu valor retornado, sendo este a vida resultante do inimigo e seu estado **DEAD** ou **ALIVE**.

EXEMPLOS

```
souls
player:hollowknight lives
enemy:hopper lives
eaten
interaction:life_condition ataque1 entities enemy:enemy1, player:player1
purevessel
  usingsoul 10
  if enemy1 lifeislessthan 10
    begin player1 uses purenail causes playerdmg being 1 to enemy1
  empty
  begin player1 uses purenail causes playerdmg being 1 to enemy1
  5
endusingsoul
return enemy1
sealedvessel
action ataque1 entities hopper, hollowknight done
```

No exemplo ao lado há algumas coisas que devem ser notadas:

- Há a declaração de duas variáveis hollowknight e hopper dos tipos player e enemy, respectivamente;
- Há uma criação de função em “interaction” de nome “ataque1” que irá ser responsável por atribuir o ataque de **player1** à **enemy1**;
- Ainda na função **ataque1**, ela utiliza um **gameloop** que irá rodar **10/5** vezes, realizando sempre um **ATAQUE** que é indicado sempre por **begin**.
- Dentro desse loop, há uma condicional que será executada somente se a vida de **hopper** for menor que 10, sendo esse um valor implícito no código do compilador;
- No fim, a função **ataque1** é chamada e seu valor retornado, sendo este a vida resultante do inimigo e seu estado **DEAD** ou **ALIVE**.

CURIOSIDADES

01

Desenvolvimento

O jogo Hollow Knight foi desenvolvido por apenas 3 pessoas e é o jogo indie com o maior número de jogadores e o mais conhecido no ramo.

02

Música

A música e trilha sonora do jogo foi composta por exclusivamente uma única pessoa, sendo esta Christopher Larkin.

03

Tempo

O jogo começou a ser desenvolvido em 2013 e foi publicado em 2017/18, um tempo consideravelmente curto considerando os detalhes do jogo e quantas pessoas trabalharam nele.

