



Exercício Programa 2 – Design de Software 2017.1

Vocês terão de desenvolver um jogo chamado **Inspèrmon!**

Nesse jogo você deve começar com uma implementação simples e depois ir evoluindo com novas funcionalidades.

O Jogo:

Fase 1

Bem vindos ao Inspèrmon! Um jogo completamente inovador, nunca visto antes, onde você anda por um mundo virtual e encontra os Inspèrmons!

Cada Inspèrmon tem minimamente os seguintes atributos:

- Nome
- Poder de ataque
- Nível de defesa
- Pontos de vida

Como funciona o jogo? Bem, basicamente você deve escolher, a cada turno, se você vai:

Passear: Seu inspèrmon estará passeando pelos campos (ou campus? ou campi ?). Inevitavelmente encontrará um Inspèrmon selvagem e a batalha começará;

Dormir: Você vai dormir. Até a próxima!

Ao passear você encontrará um Inspèrmon e batalhará! As batalhas funcionam assim:

- Seu Inspèrmon ataca o adversário. Remova da vida do adversário o valor do seu ataque menos o valor de defesa do oponente.
- Se o adversário sobreviveu, ele ataca seu Inspèrmon do mesmo jeito.
- Esses rounds continuam até que a vida de um Inspèrmon chegue a zero.

Ou seja, a lógica do jogo básico é a seguinte:

```
Loop infinito:
  Mostra opções e Pergunta: passear ou dormir?
  Se dormir: sai do loop
  Escolhe inimigo aleatório
  Mostra inimigo
  Faz a Batalha e recebe o resultado
  Se você morreu:
    Imprime alguma mensagem adequada
    Sai do loop
```

Inspêr

A função batalha deverá receber os dois Inspèrmons e retornar o resultado da batalha, segundo a lógica abaixo:

```
Função batalha:
  Enquanto VidaJogador > 0 e VidaOponente > 0:
    VidaOponente = VidaOponente - ( PoderJogador - DefesaOponente )
    Se VidaOponente <= 0:
      Retorna que jogador ganhou
    VidaJogador = VidaJogador - ( PoderOponente - DefesaJogador )
    Se VidaJogador <= 0:
      Retorna que jogador perdeu
```

Fase 2

Poxa, este jogo está muito simples! Vamos melhorar esse jogo?

Funcionalidade 1: Vamos criar o Inspèrdex!

O Inspèrdex guarda uma estrutura de dados idêntica à estrutura que armazena a lista original de Inspèrmons, mas contendo apenas os Inspèrmons que foram encontrados pelo jogador. Adicione uma opção no menu principal para exibir o conteúdo do seu Inspèrdex.

Funcionalidade 2: Vamos fugir da batalha!

Adicione uma opção para fugir da batalha. Se a fuga for bem-sucedida, a batalha termina em empate. Senão, a batalha continua e você acabou de perder seu round de ataque!

Funcionalidade 3: A sorte está lançada!

Como você percebeu os combates não envolvem sorte: o resultado sempre é o mesmo. Adicione um elemento de sorte ao combate.

Funcionalidade 4: Vamos evoluir Inspèrmons!

Crie o atributo de experiência do seu Inspèrmon! Você deve definir como ganhar experiência, e a partir de qual nível de experiência seu Inspèrmon evolui. Para simplificar, vamos definir que a evolução não necessariamente troca de tipo de Inspèrmon, apenas aumenta seus atributos vitais.

Funcionalidade 5: Salve o jogo!

Gere um par de rotinas para carregar e salvar o estado do jogo, ou seja: o Inspèrdex, o nível de experiência e o Inspèrmon do jogador.

Funcionalidade 6: Vamos inventar!

Crie uma nova funcionalidade no jogo! Mas seja criativo e mostre que é um programador evoluído: faça uma adição que impressione!

Seu objetivo é implementar quantas funcionalidades você conseguir!



Para carregar seus dados

O arquivo inicial de dados contém apenas um Inspêrmon: o "picaxu". Os dados estão armazenados em um arquivo do tipo JSON (JavaScript Object Notation), que é um jeito padronizado de escrever estruturas de dados (como listas e dicionários) em um arquivo texto. Parece muito com Python puro!

Eis aqui o conteúdo do arquivo "inspermons.json":

```
[
    {
        "nome" : "picaxu",
        "poder" : 20,
        "vida" : 200,
        "defesa" : 12
    },
    etc...
]
```

Aqui está um exemplo de jogo salvo:

```
{
    "ipmon_jogador" : {
        "nome" : "picaxu",
        "poder" : 20,
        "vida" : 200,
        "defesa" : 12
    },
    "xp_jogador" : 0,
    "insperdex" : {}
}
```

Aqui está uma amostra de código que consegue ler o arquivo de Inspêrmons:

```
import json

def mostra_ipmon(ipmon):
    print("Inspêrmon : {}".format(ipmon["nome"]))
    print("poder = {}".format(ipmon["poder"]))
    print("vida = {}".format(ipmon["vida"]))
    print("defesa = {}\n".format(ipmon["defesa"]))

with open('inspermons.json') as arquivo:
    inspermons = json.load(arquivo)

for ipmon in inspermons:
    mostra_ipmon(ipmon)
```

Rubrica de Avaliação:

Conceito	Descrição
I	Não conseguiu implementar a fase 1.
D	Implementou a fase 1, mas nenhuma outra funcionalidade.
C	Implementou a fase 1, e uma funcionalidade de 1 a 5.
B	Implementou a fase 1, e três funcionalidades de 1 a 5.
A	Implementou a fase 1, e todas as funcionalidades.

Entrega:

- Este projeto deverá ser realizado em duplas. Exercícios copiados ou muito parecidos podem gerar notificações de infrações do código de ética Insper.
- Para entregar, inclua seu código em um repositório no Github. Envie no BlackBoard o endereço do seu repositório Github.
- Os *commits* deverão minimamente refletir as fases de implementação do projeto. Ou seja, ao menos um *commit* estará relacionado ao término de cada fase ou funcionalidades.
- **Data máxima da entrega: 24/04/2017**
 - O professor irá clonar o código no Github ao final desse dia.