

# MAC0115 - IFUSP - 2o semestre 2020

## Terceiro Exercício-Programa

### Sobre os projetos anteriores

Graças às suas contribuições no âmbito das pesquisas de Pokémons, outros cientistas descobriram que as criaturinhas são abundantes em outros planetas e luas também.

Descobriu-se que a Lua contém uma grande variedade de Pokémons, todos da variante “Shiny”, muito rara e cobiçada pelos treinadores terrestres. Agora, seu objetivo é auxiliar a desenvolver um módulo matricial bidimensional para captura de tais Pokémons espaciais, que será usado nas próximas missões.

### Sobre o Programa

Um dos objetivos desta fase é você aprender a utilizar arquivos e listas de listas em Python (que vão representar as matrizes). O seu programa deverá ler as dimensões de uma matriz  $m$  por  $n$  (em metros) e as características dos Pokémons presentes nela a partir de um arquivo texto fornecido (no VPL serão fornecidos seis arquivos de entrada para testes `entrada1.txt` - `entrada6.txt`). A matriz representa um plano em que o eixo  $x$  corresponde à distância e o eixo  $y$ , à altura. O programa deverá popular a matriz (cada índice correspondente a um metro) de acordo com a posição e raio dos Pokémons, bem como guardar as características deles em uma lista. Em seguida, receberá do usuário a quantidade de Pokébolas disponíveis e, enquanto houver pokébolas e pokémons livres, lerá as entradas do usuário em relação à velocidade (em m/s) e ângulo (em graus) referentes ao lançamento. O lançamento deverá ocorrer conforme os exercícios anteriores, produzindo uma representação gráfica matricial, indicando por pontos os locais em que a Pokébola passou (mais detalhes na seção correspondente abaixo). Ao capturar um Pokémon, deve imprimir qual Pokémon foi capturado ou informar que o lançamento não capturou Pokémon algum, em caso negativo.

Finalmente, o programa imprime uma mensagem referente ao motivo pelo qual se encerrou (nenhuma pokébola ou nenhum Pokémon livre disponível).

### Lendo do arquivo de entrada

Desta vez, ao invés de ler os dados da entrada pelo teclado, você seu programa irá ler os dados de um arquivo de entrada que tem o nome “default” (padrão) “`entrada.txt`”. Para isso, você pode usar a função `open()` de Python da seguinte forma:

```
arquivo = open("entrada.txt", "r")
```

Não se esqueça de invocar a função para fechar o arquivo antes de encerrar seu programa:

```
arquivo.close()
```

Outras funções úteis para manipular listas e strings referentes à leitura do arquivo de entrada são:

```
arquivo.readline() # lê a próxima linha do arquivo e devolve
                  # como string

arquivo.readlines() # lê todas as linhas restantes no arquivo
                  # e devolve como uma lista de strings

string.split() # devolve uma lista a partir de string
              # separando a partir de seus espaços

lista[1:] # corta o primeiro item de uma lista
```

## Representação dos Pokémons na matriz

Um Pokémon é representado no programa por meio de uma lista do tipo `[nome, raio, x, y]`, onde `x` e `y` indicam as coordenadas na matriz de seu ponto central, enquanto que `raio` indica quantas casas deverão ser preenchidas ao seu redor, a partir do centro. Um Pokémon de raio zero, é representado por um quadrado 1x1. Raio um, 3x3. Raio dois, 5x5 e assim por diante.

Abaixo segue uma figura exemplificando tal representação:

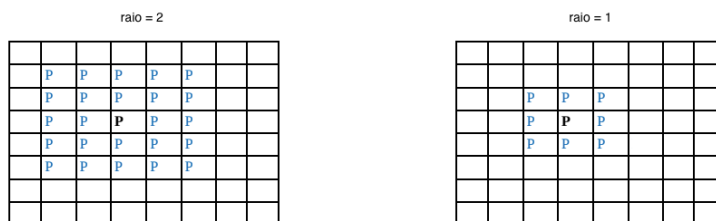


Figura 1: À esquerda, representação de um Pokémon de raio 2. À direita, representação para raio 1. As posições da matriz indicadas por P devem receber o id do Pokémon correspondente.

## Simulação do lançamento

Enquanto o treinador possuir pokébolas e existirem pokémons livres na representação de seu mundo, ele seguirá efetuando arremessos. Antes de cada arremesso, o programa deve:

- Ler a coordenada  $x$  (no chão,  $y = 0$ ) onde o treinador se posicionará. No caso em que o arremesso anterior capturou um Pokémon, o treinador deve se deslocar até a coordenada  $x$  em que a Pokébola capturou o Pokémon (ao capturar, ela perde todo seu momento e despenca verticalmente) e obrigatoriamente efetuar seu próximo arremesso deste ponto.
- A posição do treinador deve ser marcada com um “T” na matriz.

- Imprimir quantas pokébolas ainda restam e o estado atual da matriz.
- Copiar a matriz para uma nova, que será modificada para incluir os caracteres ('o') referentes à trajetória do lançamento.
- Ler velocidade (em m/s) e ângulo de lançamento (em graus, partindo de zero representando o sentido do semi-eixo  $+x$ , ou seja, lançamento para a direita, e com 180 representando o sentido do semi-eixo  $-x$ , ou seja, lançamento para a esquerda).
- Recomenda-se o uso da biblioteca matemática de Python. Funções como seno (`sin`), cosseno (`cos`) e a constante `pi` serão úteis. Lembre-se que as funções seno e cosseno recebem um ângulo em radianos.

A simulação ocorrerá como nos exercícios anteriores, atualizando os valores de `x`, `y` e `vy` para cada passo de tempo `dt` (a constante será definida em uma seção abaixo).

Considere como condições de parada:

- $y < 0$  (ou seja, se a pokébola ultrapassar o chão) e
- $x < 0$  ou  $x > n-1$  (ou seja, se a pokébola ultrapassar da moldura vertical da matriz)

Note que não há problemas que a pokébola saia da matriz “por cima”, visto que ela ainda pode acertar um Pokémon dentro da matriz, dada a natureza parabólica do movimento.

Usaremos um passo de tempo igual a 0.1, que eventualmente produzirá números reais como resultado da simulação ponto a ponto, incompatíveis com a natureza inteira dos índices da matriz. Neste caso, utilize `round(x)` e `round(y)` como índices ao lidar com valores de `x` e `y` passíveis de serem `floats`.

Finalmente, imprima a representação gráfica do lançamento, seguida por uma mensagem de captura de Pokémon, explicitando qual foi, ou de falha. Mais informações estarão na seção mais abaixo sobre a saída do programa.

## Constantes

Assuma como constantes (variáveis globais):

```
DELTA_T = 0.1
GRAVIDADE = 2
```

Para o valor de `pi`, use a constante presente na biblioteca matemática.

## Funções a serem implementadas

Todas as funções abaixo são de uso obrigatório em sua `main()`, exceto por `preenche_pokemon()`, função auxiliar para `popula_matriz()`

```

def leArquivo(nomeArquivo = 'entrada.txt'):
'''
Esta função lê um arquivo ('entrada.txt' por default) e
retorna uma lista de listas.
Entrada: arquivo cujo nome está armazenado em nomeArquivo.
        Por default, é 'entrada.txt'
Saída: uma lista de listas, onde o primeiro elemento é uma
        lista de inteiros [m, n] (dimensões da matriz) e os
        elementos subsequentes são listas que representam as
        característica lidas dos Pokémons na forma:
            [nome, raio, x, y]
'''

def criaMatriz(m, n):
'''
Esta função cria e retorna uma lista de listas.
Entrada: dois inteiros que representam o número de linhas e
        o número de colunas da matriz.
Saída: uma lista de m listas, cada uma com n elementos, todos
        inicializados com zeros.
'''

def populaMatriz(matriz, pokemons):
'''
Esta função recebe uma matriz e uma lista contendo listas que
representam os pokémons na forma [nome, raio, x, y] e preenche-a
os pokémons conforme a representação retangular considerando os
raios da representação.
Entrada: matriz representada por uma lista de listas
Saída: A matriz fornecida é modificada.
'''

def preenchePokemon(matriz, id, x, y, raio):
'''
Esta função é auxiliar da função populaMatriz. Ela insere
um Pokémon na matriz de acordo com sua representação retangular
baseada no raio ao redor do ponto central (x,y)
Entrada: matriz representada por uma lista de listas
        id é o número a preencher a matriz; para o
        primeiro pokémon na lista (de índice zero),
        usa-se 1 e assim subsequentemente.
        x,y são as coordenadas do ponto central
        raio é a distância a ser guardada a partir do
        ponto central.
Saída: A matriz fornecida é modificada.
'''

def removePokemon(matriz, id, pokemons):
'''
Esta função recebe uma matriz, o numeral que representa o pokémon

```

```

a ser removido da matriz (id) e a lista contendo as listas que
representam pokémons, substituindo os numerais id por zero
Entrada: matriz representada por uma lista de listas
        id é o número a preencher a matriz; para o
        primeiro pokémon na lista (de índice zero),
        usa-se 1 e assim subsequentemente.
        pokemons lista contendo as listas que representam pokémons
Saída: A matriz fornecida é modificada.
'''

def imprimeMatriz(matriz):
'''
Esta função imprime a matriz dada.
Note que a matriz deve ser impressa com espelhamento vertical,
pois a primeira linha representa o chão.
Entrada: matriz representada por uma lista de listas
'''

def atualizaPosicao(x, y, vx, vy, dt=DELTA_T):
'''
Esta função calcula as atualizações das posições de x e y usando
as velocidades escalares respectivamente dadas por vx e vy.
Entrada: As posições x e y dadas em metros, as velocidades vx e
vy em metros por segundo e o intervalo de tempo em segundos.
Saída: Dois valores: o valor atualizado de x e o valor atualizado de y.
'''

def atualizaVelocidade(vx, vy, dt=DELTA_T):
'''
Esta função calcula e atualiza as velocidades vx e vy para o
próximo intervalo de tempo.
Entrada: As velocidades vx e vy em metros por segundo e o
intervalo de tempo em segundos.
Saída: Dois valores: o valor atualizado de vx e o valor atualizado de vy.
'''

def grau2Radiano(theta):
'''
Esta função converte o ângulo theta em graus para radianos.
Entrada: ângulo theta
Saída: ângulo theta em radianos
'''

```

## Atenção à saída

A saída de seu programa deve estar formatada identicamente às saídas deixadas como exemplo. Este passo será essencial para a avaliação correta de seu exercício-programa, portanto atente-se a isto. Note que, internamente, a matriz contém 0s nas posições vazias, mas na hora de imprimir, eles são substituídos

por “.” (ponto), para ajudar na visualização.

Para facilitar a checagem e comparação, utilize o seguinte website, onde você pode copiar da tela e colar a saída do seu programa:

- <https://www.diffchecker.com>

#### **Sobre os programas e a entrega**

- Atente-se à especificação das entradas e saídas do seu programa. Mais precisamente, siga à risca os formatos de entrada e de saída descritos no enunciado. Isso facilita a correção do programa pelo avaliador automático do Laboratório Virtual de Programação no edisciplinas. Observe que as acentuações foram omitidas das saídas impressas por simplicidade.
- Não será possível entregar o trabalho após as 23:55 do dia de entrega. Por isso, programe-se para não exceder o prazo. A data de entrega pode ser vista na parte EP3 do edisciplinas.
- O Python tem muitas bibliotecas que facilitam a programação, mas seu uso nos EPs criam problemas para sua correção, criam desequilíbrios, dificultam a atribuição de nota etc. Assim, antes de usar qualquer coisa que não foi ensinada em aula, pergunte no fórum.

## Exemplos de execução:

(os números pintados de vermelho foram digitados pelo usuário)

```
Digite o nome do arquivo: entrada2.txt
Digite o numero N de pokebolas: 2
Digite a coordenada x do treinador: 4
pokebolas disponiveis = 2
Estado atual do jogo:
.....
.....
.....
...111...
...111...
...111...
.....
.....
.....
....T....
Digite a velocidade de lancamento em m/s: 5
Digite o angulo de lancamento em graus: 45
Representacao grafica do lancamento:
.....
.....
.....
...111...
...111...
...111...
.....oo
.....oo..
.....o....
....T....
O lancamento nao capturou pokemon algum
Digite a coordenada x do treinador: 1
pokebolas disponiveis = 1
Estado atual do jogo:
.....
.....
.....
...111...
...111...
...111...
.....
.....
.....
.T.....
Digite a velocidade de lancamento em m/s: 8
Digite o angulo de lancamento em graus: 50
Representacao grafica do lancamento:
.....
.....
.....
...111...
...111...
...11o...
.....o....
...o.....
..o.....
.T.....
Um Nidoran foi capturado!
Parabens! Todos pokemons foram capturados
```

```
Digite o nome do arquivo: entrada4.txt
Digite o numero N de pokebolaz: 2
Digite a coordenada x do treinador: 7
pokebolaz disponiveis = 2
Estado atual do jogo:
.....33333.111.....
.....33333.111.....
.....33333.111.....
.....33333.....
.....33333.....2.....
.....
.....
.....
.....T.....
.....
Digite a velocidade de lancamento em m/s: 15
Digite o angulo de lancamento em graus: 80
Representacao grafica do lancamento:
.....33333.111.....
.....33333.111.....
.....33333.111.....
.....33333.....
.....33o33.....2.....
.....o.....
.....o.....
.....o.....
.....o.....
.....o.....
.....o.....
.....T.....
Um Quilava foi capturado!
pokebolaz disponiveis = 1
Estado atual do jogo:
.....111.....
.....111.....
.....111.....
.....
.....2.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....T.....
Digite a velocidade de lancamento em m/s: 12
Digite o angulo de lancamento em graus: 45
Representacao grafica do lancamento:
.....111.....oo..
.....111.....ooo..
.....111.....oo...
.....o.....
.....oo...2.....
.....oo.....
.....o.....
.....o.....
.....o.....
.....oo.....
.....o.....
.....oo.....
.....O.....
.....T.....
O lancamento nao capturou pokemon algum
Jogo encerrado
```