

Especificação da recuperação do do trabalho 1

Quem pode fazer?

Todos que tiraram nota abaixo de 7 no primeiro trabalho. Este trabalho valerá 7 pontos e substituirá o anterior.

Ou seja:

1. Tirou 7 ou mais? Não pode fazer. Sua nota é a do trabalho 1.s
2. Tirou abaixo de 7? Pode fazer e a nota que terá aqui substituirá a anterior se for maior. Por exemplo, se tirou 6 no primeiro trabalho e agora tirou 7, computarei sua nota do trabalho 1 como 7. Pegarei a maior das duas.

Objetivos

1. Usar os conhecimentos abordados para o projeto de funções, laços e condicionais.

O Problema

Os engenheiros da Nintendo estavam desenvolvendo o jogo Pokemon. Eles resolveram começar o jogo projetando a simulação da trajetória das pokebolas em um plano cartesiano, bidimensional (X e Y), sem atrito.

No início da simulação o treinador pokemon terá N pokebolas ($N > 0$), que poderão ser usadas para capturar pokemons. Dadas as posições iniciais do treinador (x_t, y_t) e de um pokemon(x_p, y_p), o valor da atração gravitacional (gravidade, g) e a velocidade do lançamento $v_0 = (v_{xb}(0), v_{yb}(0))$, o módulo deverá calcular, segundo a segundo, a

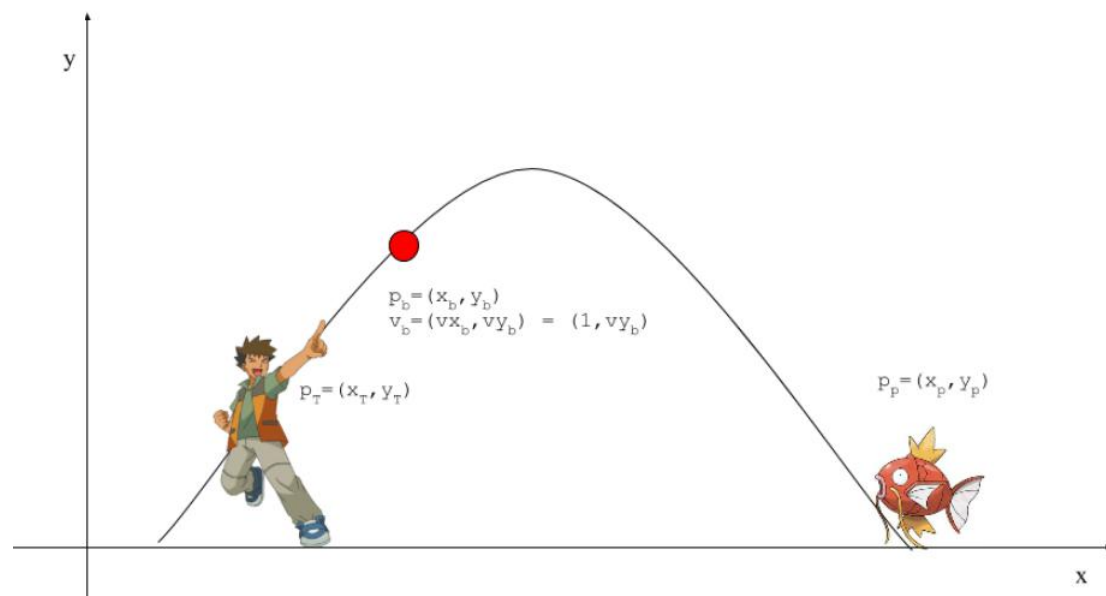
trajetória da pokebola ($x_b(t)$, $y_b(t)$) e determinar se ela colidiu com o pokemon. Caso a pokebola não tenha atingido o pokemon, o usuário poderá digitar novos valores de posição do treinador e velocidade de lançamento. A simulação será executada novamente e esse procedimento deve se repetir até que não haja mais pokebolas ou a pokebola não tenha atingido o pokemon.

Os cálculos

A simulação deve ser realizada atualizando a posição da pokebola ($x_b(t)$, $y_b(t)$) a cada segundo. Ou seja, a variação de tempo é $\Delta t = 1s$. O cálculo da posição da pokebola ($x_b(t)$, $y_b(t)$) é realizado de forma iterativa, usando a posição e velocidade no instante anterior, $t-1$, segundo as fórmulas abaixo:

- $x_b(t) = x_b(t-1) + vx_b(t-1) * \Delta t$.
- $y_b(t) = y_b(t-1) + vy_b(t-1) * \Delta t - g/2 * \Delta t^2$.
- $vx_b(t) = vx_b(t-1)$.
- $vy_b(t) = vy_b(t-1) - g * \Delta t$.
- $(x_b(0), y_b(0)) = (x_t, y_t)$

A figura abaixo demonstra o problema com as equações.



Note que os valores do componente y da velocidade (vy_b) são atualizados. O componente x da velocidade (vx_b) é

uniforme. A gravidade não varia. E não há qualquer outro efeito na simulação, só o movimento descrito nos cálculos acima.

O comportamento do programa

O programa deve ler inicialmente o número de pokebolas N , o valor da gravidade g e as coordenadas do pokemon(x_p, y_p) nessa ordem. Para cada tentativa, o programa deve ler as coordenadas iniciais do treinador(x_t, y_t) e o valor inicial do componente y da velocidade ($vy_b(0)$). Suponha que o valor do componente x da velocidade é 1 ($vx_b(0) = 1$).

O programa deverá calcular a posição da pokebola a cada segundo, e imprimir linhas contendo o instante t , a velocidade ($vy_b(t)$) e as coordenadas $x_b(t)$ e $y_b(t)$ da pokebola, usando **exatamente** o padrão seguinte:

```
> t=    0    vy=    0    x=    0    y=    0
```

Ou seja, com um espaço de 4 dígitos para impressão do valor e outro espaço de 4 dígitos separando do valor anterior.

A simulação deve ocorrer até uma das situações seguintes ser atingida:

- A pokebola bater no chão (posição y_b menor ou igual a zero).
- A pokebola estar na mesma posição horizontal x_b que o pokemon (x_p), ou depois.

Para determinar se a bola atingiu o pokemon as coordenadas x e y da bola e do pokemon devem ser idênticas. O programa deve imprimir ao final de cada tentativa uma mensagem indicando se o pokemon foi atingido.

Observações

- As coordenadas digitadas devem ser maiores ou iguais a 0.
- Informe apenas valores inteiros.
- Assuma que g é par.

Exemplos de execução

```
Digite o numero N de pokebolas: 3
Digite o valor da gravidade: 10
Digite a coordenada x (inteiro >= 0) do pokemon: 5
Digite a coordenada y (inteiro >= 0) do pokemon: 20
```

Tentativa 1:

```
Digite a coordenada x (inteiro >= 0) do treinador: 0
Digite a coordenada y (inteiro >= 0) do treinador: 2
Digite a componente y da velocidade de lancamento: 3
> t= 0    vy= 3    x= 0    y= 2
> t= 1    vy= -7   x= 1    y= 0
```

A pokebola nao atingiu o pokemon.

Tentativa 2:

```
Digite a coordenada x (inteiro >= 0) do treinador: 3
Digite a coordenada y (inteiro >= 0) do treinador: 50
Digite a componente y da velocidade de lancamento: 10
> t= 0    vy= 10   x= 3    y= 50
> t= 1    vy= 0    x= 4    y= 55
> t= 2    vy= -10  x= 5    y= 50
```

A pokebola nao atingiu o pokemon.

Tentativa 3:

```
Digite a coordenada x (inteiro >= 0) do treinador: 3
Digite a coordenada y (inteiro >= 0) do treinador: 20
Digite a componente y da velocidade de lancamento: 10
> t= 0    vy= 10   x= 3    y= 20
> t= 1    vy= 0    x= 4    y= 25
> t= 2    vy= -10  x= 5    y= 20
```

A pokebola atingiu o pokemon.

Digite o numero N de pokebolas: 3
Digite o valor da gravidade: 4
Digite a coordenada x (inteiro ≥ 0) do pokemon: 3
Digite a coordenada y (inteiro ≥ 0) do pokemon: 7

Tentativa 1:

Digite a coordenada x (inteiro ≥ 0) do treinador: 0
Digite a coordenada y (inteiro ≥ 0) do treinador: 0
Digite a componente y da velocidade de lancamento: 12
> t= 0 vy= 12 x= 0 y= 0
> t= 1 vy= 8 x= 1 y= 10
> t= 2 vy= 4 x= 2 y= 16
> t= 3 vy= 0 x= 3 y= 18

A pokebola nao atingiu o pokemon.

Tentativa 2:

Digite a coordenada x (inteiro ≥ 0) do treinador: 1
Digite a coordenada y (inteiro ≥ 0) do treinador: 3
Digite a componente y da velocidade de lancamento: 7
> t= 0 vy= 7 x= 1 y= 3
> t= 1 vy= 3 x= 2 y= 8
> t= 2 vy= -1 x= 3 y= 9

A pokebola nao atingiu o pokemon.

Tentativa 3:

Digite a coordenada x (inteiro ≥ 0) do treinador: 1
Digite a coordenada y (inteiro ≥ 0) do treinador: 2
Digite a componente y da velocidade de lancamento: 5
> t= 0 vy= 5 x= 1 y= 2
> t= 1 vy= 1 x= 2 y= 5
> t= 2 vy= -3 x= 3 y= 4

A pokebola nao atingiu o pokemon.

Nota

Estes são os aspectos do seu trabalho que podem ser avaliados para nota:

- Estilo do código (20%):
 - Certifique-se de seguir as diretrizes de estilo do Python que apresentamos e as convenções de codificação do Python que temos usado ao longo do semestre.
 - Todas as funções, incluindo funções auxiliares, devem ter docstrings completas, incluindo pré-condições, quando você achar que elas são necessárias.
- Corretude (80%):
 - Suas funções devem ser executadas conforme especificado.

Como submeter

A submissão é feita no sigaa. Está liberada a ressubmissão.

Construa um arquivo compactado, tem que ser **.zip**, **não serão aceitos outros tipos de compactação (como rar, 7zip e etc)**. Por exemplo, o aluno de matrícula 000000, vai gerar o arquivo 000000_Trabalho.zip contendo o seu código.