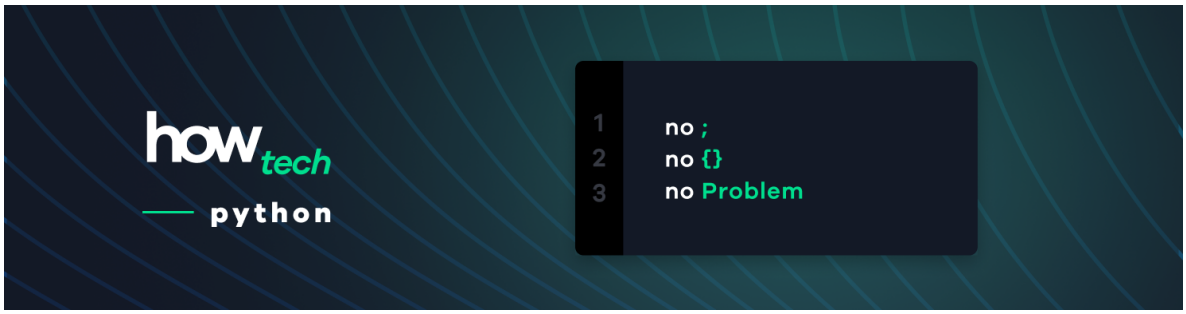

Bootcamp Python - /código[s] - Stone

Code War

Projeto Labirinto

Definições do Projeto

Implementar um sistema em Python que efetue o comportamento de um robô tentando encontrar a saída de um labirinto.

As inspirações para esse projeto foram: aspiradores de pó automáticos (como o Roomba), navegação no Google Maps e o problema clássico do Labirinto do Minotauro.

O objetivo do projeto é posicionar um robô em um labirinto e desenvolver a lógica para que ele percorra esse labirinto em busca da saída, avançando pelas células em branco, respeitando as paredes e retornando por um caminho caso esteja encurralado. O robô não pode avançar 2 vezes por um mesmo caminho, assim, ao descobrir que está encurralado pode retornar pelas células percorridas, mas não deve avançar novamente por este caminho.

O enunciado a seguir traz instruções referentes à implementação do projeto, mas as equipes podem fazer algumas alterações como, por exemplo, optar por outra padronização de números ou caracteres para representar o labirinto e o robô. As matrizes representadas neste documento são apenas sugestões. Opte por um modelo e realize sua codificação.

O intuito é cumprir o objetivo da navegação do robô no labirinto.

O labirinto

O labirinto pode ser representado por uma matriz de inteiros ou caracteres representando, por exemplo:

- 0 ou " ": espaço em branco por onde o robô pode caminhar
- 1 ou "#": parede, o robô não pode avançar para essa posição
- 2 ou ".": posição já percorrida pelo robô
- 3 ou "S": saída do labirinto
- 4 ou "X": posição atual do robô

O labirinto pode ser montado em uma matriz de inteiros ou caracteres, onde a combinação dos números ou dos caracteres em posições específicas definirá o seu desenho. Os caminhos

desenhados devem permitir que o robô encontre uma saída e o robô deve respeitar os limites da matriz.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	0	0	0	1
2	1	1	1	1	0	1	1	1	0	1	0	1
3	1	0	0	0	0	0	0	1	0	1	0	1
4	1	0	1	1	1	0	0	1	0	1	0	1
5	1	0	1	0	1	1	1	1	0	1	0	1
6	1	0	1	0	0	0	0	0	0	1	0	1
7	1	0	1	0	1	1	0	1	0	1	0	1
8	1	0	0	0	0	1	0	0	0	1	0	1
9	1	1	1	1	1	1	1	1	1	1	0	3

	0	1	2	3	4	5	6	7	8	9	10	11
0		#	#	#	#	#	#	#	#	#	#	#
1						#						#
2	#	#	#	#		#	#	#		#		#
3	#							#		#		#
4	#		#	#	#			#		#		#
5	#		#		#	#	#	#		#		#
6	#		#							#		#
7	#		#		#	#		#		#		#
8	#					#				#		#
9	#	#	#	#	#	#	#	#	#	#		S

Representações do labirinto: matriz de inteiros, matriz de caracteres, desenho

Após apresentar o labirinto ao usuário, deve-se pedir que ele informe a posição em que o robô será inserido, devendo essa posição representar um espaço em branco, obrigatoriamente. Uma vez posicionado, o robô deve iniciar seu passeio no labirinto a fim de encontrar a saída.

Por exemplo: iniciar o robô na posição 4,10 (linha 4, coluna 10).

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	0	0	0	1
2	1	1	1	1	0	1	1	1	0	1	0	1
3	1	0	0	0	0	0	0	1	0	1	0	1
4	1	0	1	1	1	0	0	1	0	1	4	1
5	1	0	1	0	1	1	1	1	0	1	0	1
6	1	0	1	0	0	0	0	0	0	1	0	1
7	1	0	1	0	1	1	0	1	0	1	0	1
8	1	0	0	0	0	1	0	0	0	1	0	1
9	1	1	1	1	1	1	1	1	1	1	0	3

	0	1	2	3	4	5	6	7	8	9	10	11
0		#	#	#	#	#	#	#	#	#	#	#
1						#						#
2	#	#	#	#		#	#	#		#		#
3	#							#		#		#
4	#		#	#	#			#		#	X	#
5	#		#		#	#	#	#		#		#
6	#		#							#		#
7	#		#		#	#		#		#		#
8	#					#				#		#
9	#	#	#	#	#	#	#	#	#	#		S

Representações do labirinto com o posicionamento do robô

Para que o usuário consiga acompanhar o passeio do robô, o labirinto deve ser apresentado na tela a cada passo dado. O robô pode avançar por um caminho nunca percorrido e pode retornar por um caminho já percorrido. Não pode, em nenhuma situação, avançar por um caminho já percorrido.

Para garantir esse comportamento, a cada passo válido dado pelo robô, faz-se a troca do número 0 pelo número 2 no labirinto, indicando que esta posição já foi visitada, como se ele deixasse uma marca nas posições em que já passou.

Caso o robô precise retornar, não é necessário trocar o número 2 já registrado no labirinto.

Por exemplo: imagine que o robô andou 3 casas para cima e uma casa para a esquerda. Lembre-se que é preciso marcar como percorrida a posição em que ele iniciou o percurso.

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	1	1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0	0	0	4	2	1
2	1	1	1	1	0	1	1	1	0	1	2	1
3	1	0	0	0	0	0	0	1	0	1	2	1
4	1	0	1	1	1	0	0	1	0	1	2	1
5	1	0	1	0	1	1	1	1	0	1	0	1
6	1	0	1	0	0	0	0	0	0	1	0	1
7	1	0	1	0	1	1	0	1	0	1	0	1
8	1	0	0	0	0	1	0	0	0	1	0	1
9	1	1	1	1	1	1	1	1	1	1	0	3

	0	1	2	3	4	5	6	7	8	9	10	11
0		#	#	#	#	#	#	#	#	#	#	#
1						#				X	.	#
2	#	#	#	#		#	#	#		#	.	#
3	#							#		#	.	#
4	#		#	#	#			#		#	.	#
5	#		#		#	#	#	#		#		#
6	#		#							#		#
7	#		#		#	#		#		#		#
8	#					#				#		#
9	#	#	#	#	#	#	#	#	#	#		S

Representações do labirinto com avanço do robô

O robô

O robô nada mais é do que o número 4 ou o caracter "X" dentro do labirinto. Ele pode caminhar no labirinto apenas na horizontal e na vertical.

Uma vez posicionado, o robô deve executar sua regra de movimentação:

- testar a sequência das quatro direções possíveis (acima, à direita, abaixo ou à esquerda) tentando encontrar a saída (valor 3 na matriz). Se encontrá-la, avançar na direção correta e encerrar o jogo;
- se não encontrar a saída, testar a sequência das quatro direções para avançar para uma casa em branco (valor 0 na matriz). Na primeira direção em que for possível avançar, empilhar a sua posição atual, marcar a posição atual como percorrida (valor 2 na matriz) e avançar;
- caso não consiga avançar para uma posição em branco (valor 0 na matriz) após testar as 4 direções, deve retornar para sua posição imediatamente anterior. Para isso, deve desempilhar a última posição empilhada e retornar para ela.

O robô deverá repetir a regra acima até que encontre a saída.

O robô deve saber a sua posição atual (linha e coluna na matriz) e saber a sequência de passos que representa um caminho percorrido (pilha de posições).

A pilha

Uma pilha é uma estrutura de dados bem definida que segue o comportamento LIFO: Last In First Out (a última coisa empilhada é a primeira a ser desempilhada). Como em uma pilha de livros, se você empilhar os livros A, B, C e D, nessa ordem, irá desempilhá-los na ordem D, C, B e A.

Nesse projeto, a representação da pilha pode ser um vetor onde elementos são inseridos e removidos do seu final. Cada elemento deverá representar uma posição da matriz, com valor para linha e coluna, representando as posições que o robô já visitou (mas sem a sua posição atual).

Por exemplo: partindo da posição (4, 10), ao avançar 4 casas para cima e uma para a esquerda, a pilha do caminho percorrido será:

(1, 10) topo da pilha
(2, 10)
(3, 10)
(4, 10) base da pilha

Se o robô precisar retornar, desempilhará a posição (1, 10) e a pilha será atualizada para:

(2, 10) topo da pilha
(3, 10)
(4, 10) base da pilha

Empilhar uma posição significa registrar uma posição já visitada pelo robô, sendo que ele saiu dela e avançou para outra. O robô só avança para posições na matriz que possuem o valor 0 (espaço em branco). Desempilhar uma posição significa que o robô ficou encurralado e deve retornar para a última posição em que estava. O robô só retorna para posições na matriz que possuem o valor 2 (posição já percorrida).

Caso seja empilhada uma sequência de posições, significa que o robô avançou por um caminho. Caso seja desempilhada uma sequência de posições, significa que o robô retornou por um caminho. Lembre-se, entretanto, que o robô avança ou retorna apenas uma casa do caminho. A cada passo dado, executa sua regra de movimentação.

Nesse projeto, o robô não deve avançar por um mesmo caminho mais de uma vez, senão perderia tempo testando caminhos que já sabe que não levam à saída. Esse comportamento é garantido com o uso da pilha e com a marcação do valor 2 na matriz.

Agora é a sua vez de programar!

O projeto é proposto com 4 níveis de dificuldade. Leia as opções dos níveis e identifique aquele que acredita ser capaz de implementar.

Registre seu nome na planilha de níveis (insira seu nome completo apenas na aba que representa o nível escolhido) e confira no dia do Code War a equipe da qual fará parte. Você irá trabalhar em uma sala específica do Discord para a sua equipe.

- O link para a planilha está disponibilizado no Discord no canal #CodeWar.

Caso a equipe opte por trocar de nível, é possível fazer isso em qualquer momento. Ressaltamos apenas que toda a equipe deverá concordar com a troca pois os integrantes deverão codificar o projeto juntos.

As equipes deverão ser compostas por 5 alunos.

Nível 1

A equipe pode baixar um arquivo pré-codificado

Acesse no Discord o link para baixar um arquivo pré-codificado e com instruções sobre a codificação do projeto.

Observações:

- será fornecida uma matriz 10x10 com o robô já posicionado na matriz; e
- deverá ser implementada toda a lógica de movimentação do robô.

Nível 2

A equipe deverá desenvolver todo o código do projeto, contando com o auxílio de um fluxograma de apoio lógico

Acesse no Discord o link para baixar o fluxograma que apresenta o roteiro para a implementação do seu código.

Observações:

- é permitido representar o desenho do labirinto diretamente no código;

- deverá ser solicitado ao usuário que informe a posição inicial do robô no labirinto; e
- deverá ser implementada toda a lógica de movimentação do robô.

Nível 3

A equipe deverá desenvolver todo o código do projeto, fazendo a leitura do desenho do labirinto em um arquivo texto

Não é necessário baixar arquivos nesse nível visto que a equipe deverá fazer todo o desenvolvimento do projeto.

Observações:

- o desenho do labirinto deve ser feito em um arquivo texto, com a padronização que a equipe escolher;
- deverá ser solicitado ao usuário que informe a posição inicial do robô no labirinto; e
- deverá ser implementada toda a lógica de movimentação do robô.

Nível 4

A equipe deverá implementar um gerador de labirinto

Não é necessário baixar arquivos nesse nível visto que a equipe deverá fazer todo o desenvolvimento do projeto.

Observações:

- o labirinto deverá ser gerado automaticamente, tendo um desenho que permita o passeio do robô;
- posicionar o robô em uma posição aleatória; e
- deverá ser implementada toda a lógica de movimentação do robô.

Durante o Code War

A equipe deverá definir o aluno responsável pela comunicação com os instrutores e com a postagem do código final.

Caso precisem de auxílio no desenvolvimento do código, o aluno responsável deverá inserir seu nome e o número da sua equipe na planilha da Fila de Atendimento.

- O link para a planilha está disponibilizado no Discord no canal #CodeWar.

Caso a equipe deseje receber comentários sobre o seu código, o aluno responsável deverá postar o código no seu GitHub e indicá-lo na planilha da Fila de Atendimento. Serão avaliados os projetos postados até o dia 12/04.

Após o dia 12/04 serão postados os gabaritos dos 4 níveis de dificuldade do projeto.