

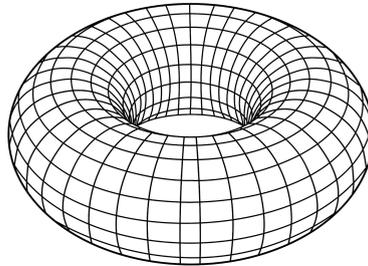
Terceiro Exercício-Programa - Parte A

BCC 2018 - MAC0110 - Entrega: até 11/06/2018 23:55 pelo PACA

Sobrevivendo no Mundo de Wumpus

Introdução

Nesse EP vamos criar estratégias para sobreviver no sombrio Mundo de Wumpus! Na nossa versão, adaptada do jogo criado por Gregory Yob em 1975, o mundo é um tabuleiro em forma de um toro $N \times N$, como na figura abaixo.



Nesse mundo, cada quadrado representa uma sala, conectada a quatro outras salas, que podem estar livres, seladas com um muro, conter um poço sem fundo ou um terrível Wumpus. As personagens nesse mundo devem se unir para eliminar todos os Wumpus e assim poderem voltar sãs e salvas para casa. Ao percorrerem as salas do mundo, as personagens recebem dados sensoriais que as ajudarão a compreender o que pode estar nas salas adjacentes, e assim poderem evitar alguns destinos trágicos, como caírem em um poço ou serem devoradas por um Wumpus. Cada personagem possui uma flecha, que pode ser disparada uma única vez, preferencialmente quando a personagem tiver acumulado evidências suficientes de que existe um Wumpus à sua frente. Personagens também podem se encontrar e trocar conhecimento sobre as partes do mundo que já exploraram.

Uma das dificuldades que as personagens encontram nesse mundo é a quantidade limitada de informações sensoriais que podem usar para se guiarem. Como o mundo não recebe luz, as personagens não enxergam, e como o mundo não possui bordas ou quinas, não existem referências absolutas de posição. Assim cada personagem constrói o seu conhecimento do mundo relativamente à sua posição/orientação inicial e a partir das informações que consegue coletar no caminho. Ao chegar numa sala, a personagem recebe uma coleção de percepções a respeito dos arredores: se ela estiver em uma sala adjacente a um Wumpus ("W"), sentirá um fedor ("F") medonho; ao passar nas adjacências de um poço ("P") sem fundo, sentirá uma brisa ("B"); se ela tentar entrar numa sala que está selada por um muro ("M"), ela sentirá um impacto ("I") e não conseguirá entrar; e finalmente, se ela conseguir disparar um tiro certo, em todo o mundo se ouvirá um urro ("U") do Wumpus agonizante, que depois de morto não exala mais seu fedor característico. Se outra personagem estiver na mesma sala, ela lhe dirá seu nome (diferente de "F"/"B"/"I"/"U"). As ações no Mundo de Wumpus também são bem restritas: cada personagem pode tentar andar ("A") para a sala imediatamente à sua frente, girar 90° para a sua direita ("D"), girar 90° para a sua esquerda ("E"), atirar ("T") sua flecha para a frente, ou compartilhar ("C") informações com outros personagens que estiverem na mesma sala.

A Parte A desse EP consiste em adaptar uma implementação existente de uma personagem, trocando o atual sistema de ações guiadas pelo usuário por alguma estratégia automática, que você irá definir. Não se preocupe: você não precisa saber IA pra fazer esse EP! Bastará entender a mecânica do jogo e as regras mínimas que sua estratégia precisa satisfazer. Sua personagem será compartilhada com toda a classe, para que todos possam usá-la no desenvolvimento de um gerenciador mais sofisticado do mundo virtual, que será o objeto da Parte B do EP3.

(1,4) ou à nossa esquerda (1,1). Nenhum desses passos é muito seguro, então talvez seja melhor voltarmos para trás e tentarmos um outro caminho por uma das salas não visitadas que sabíamos estar livre (ou seja, (0,4), (0,1) ou (4,0)).

O arquivo mundo.py fornece um gerenciador minimal de um mundo particular onde existem duas personagens: você e uma personagem chamada “Dummy”. O código gerenciador do mundo não é um arquivo executável, mas sim um programa em Python perfeitamente legível. Você não precisa se preocupar com ele nesta Parte A, nem alterá-lo e muito menos submetê-lo no PACA, mas ele será bem útil para a Parte B do EP3, onde você irá implementar um gerenciador muito mais legal que este. A única parte do código mundo.py que talvez você possa ter interesse em mexer mais para a frente é a linha “`__DEBUG__ = False`”: trocando esse valor para True você passará a enxergar a realidade externa, que a sua personagem não enxerga. Isso será quase como entrar na Matrix (tenha muito cuidado para preservar a sua sanidade!).

Tente percorrer mais um pouco o labirinto de salas desse Mundo de Wumpus e anote seus raciocínios e conclusões num papel; anote todas as informações percebidas ao longo do caminho e também as salas que você ainda não visitou mas já sabe que estão livres. Se encontrar com a personagem Dummy, peça para ela compartilhar o que conhece. Se sentir um fedor, prepare sua flecha! Essa será a melhor maneira de compreender o que exatamente você deve implementar nesta Parte A do EP3.

Especificação

Seu trabalho na Parte A do EP3 será definir uma estratégia de sobrevivência para a sua personagem no Mundo de Wumpus, completando o código do arquivo personagemNUSP.py (não esqueça de renomear esse arquivo substituindo o seu próprio número USP). Essa parte do EP será pública, ou seja, após o prazo todos os alunos terão acesso ao seu código submetido no PACA. Que isso sirva de estímulo para vocês fazerem documentações bem caprichadas!

Nossa personagem corresponde a um módulo Python, que possuirá variáveis globais (que serão usadas em todo o módulo) e funções. Essas variáveis já estão declaradas (e amplamente documentadas) no arquivo, e também são declaradas dentro das funções (para vincular esses nomes ao escopo do módulo). Duas dessas variáveis fazem parte do protocolo de comunicação com o mundo, e por isso não devem ser modificadas pela personagem: essas são as variáveis **nFlechas**, que contém o número de flechas que a personagem possui, e **mundoCompartilhado**, que possui uma representação do mundo obtida de outra personagem através da ação compartilhar (“C”). As outras variáveis globais existentes estão sugeridas para facilitar o trabalho de vocês. São elas:

- **mundo**: representa o conhecimento da personagem em relação ao Mundo de Wumpus. Essa é uma matriz de $N \times N$ onde a personagem toma notas de suas percepções ao longo do caminho que percorre, indicando os muros e as salas livres/visitadas, bem como a proximidade de perigos (poços e Wumpus). Cada entrada mundo[i][j] é uma lista de anotações/rótulos correspondentes às informações encontradas ou deduzidas pela personagem sobre o conteúdo da sala (i,j).
- **posicao**: representa a posição relativa da personagem no mundo. Cada personagem começa em uma posição aleatória (e desconhecida por ela, pois não possui GPS ou equivalente) do Mundo “real” de Wumpus, e por isso precisa usar um sistema de coordenadas pessoal para se orientar. Por convenção, sua posição inicial é representada como [0,0] (canto superior esquerdo) nesse sistema. Como o mundo não tem bordas, podemos usar sempre os índices $i=0, \dots, N-1$ e $j=0, \dots, N-1$ para indexar todas as salas possíveis do Mundo de Wumpus, usando o operador módulo (%) para corrigir a posição quando um passo nos levar a uma sala de índice <0 ou $\geq N$.
- **orientacao**: juntamente com a posição relativa, permite à personagem manter o histórico das salas visitadas. Essa orientação é independente da orientação “real” que o mundo usa para coordenar a ação de todas as personagens. Por convenção, todas as personagens indicam sua orientação inicial como “para baixo” ou “sul”, correspondente à direção [1,0] (direção do eixo vertical).

Você pode criar quantas outras variáveis globais ao módulo quiser, e inicializá-las na função **inicializa()**. Tudo o que você precisa saber sobre o mundo nesse momento é que ele aciona a função **inicializa()** no início da simulação, e depois entra em um laço acionando a cada rodada duas funções para cada personagem: as funções **planejar()** e **agir()**, que são o cerne da estratégia de sobrevivência de cada personagem. A primeira recebe uma lista de informações sensoriais do mundo, referentes à posição atual da personagem e também a outros eventos, e atualiza a representação interna que a personagem constrói do mundo (sua memória, por assim dizer); a função **planejar()** também é responsável por deduzir e lançar hipóteses sobre o conteúdo de salas vizinhas, tais como:

- se não sinto nem fedor (“F”) nem brisa (“B”), posso concluir que as salas vizinhas estão livres (“L”);
- se sinto fedor, é certo que existe um Wumpus em alguma das salas vizinhas; devo marcar esse conhecimento com “W?” nas salas vizinhas cujo conteúdo já não seja conhecido (porque algumas das salas vizinhas podem já ter sido visitadas, ou deduzidas como livres, muros, poços, etc.);
- analogamente, se sinto uma brisa, é certo que existe um poço em uma das salas vizinhas e desconhecidas, e devo marcar esse conhecimento com “P?”;
- se sinto um impacto (“I”), significa que minha última ação (andar) foi mal-sucedida, e não mudei de posição; além disso, posso concluir que há um muro (“M”) à minha frente;
- se atirei uma flecha estando ao lado de um Wumpus e escuto um urro (“U”) e não sinto mais fedor (“F”), isso significa que ele está morto;
- se uma outra personagem compartilhou comigo informações sobre outras salas do mundo, posso incorporar essas informações à minha própria representação do mundo, seja transcrevendo-as (se eu confiar na informação), ou posfixando-as com “?” para indicar que não são certas (por exemplo, se uma personagem sentiu um fedor numa sala pode ser que esse fedor não exista mais, certo?).

Como as possibilidades são inúmeras, sua função **planejar()** precisa, no mínimo:

- marcar salas visitadas como “V” (o arquivo personagemNUSP já faz isso, mas tem um erro);
- incorporar os dados da percepção (“F”/“B”) à representação do mundo;
- marcar como “L” as salas adjacentes a qualquer sala que não tenha “M”, “F” ou “B”; além disso você deve criar uma lista com as salas livres e não visitadas para lembrar de as explorar no futuro;
- incorporar as inferências correspondentes (“W?”/“P?”) para as salas vizinhas de “F”/“B” (quando já não forem “V”/“L”), bem como marcar com “M” os muros encontrados;
- usar informação compartilhada por outras personagens quando disponível.

A segunda função, **agir()**, consulta a representação do mundo conhecido e outros dados, como o número de flechas disponíveis e informações compartilhadas por outras personagens, para tomar uma decisão sobre realizar uma das cinco ações disponíveis (“A”, “D”, “E”, “T” ou “C”); a cada rodada a personagem deve realizar uma dessas ações, pois não existe a opção de “não agir”. A estratégia específica será desenvolvida por você, e não precisa ser sofisticada; seus únicos requisitos formais são:

- ser baseada na representação do mundo e não usar decisões aleatórias;
- só dar passos arriscados (para salas “W?” ou “P?”) quando não houver outra opção, como compartilhar (“C”) ou tentar voltar para explorar uma das salas livres não visitadas;
- compartilhar informação sempre que for possível;
- usar informação compartilhada quando disponível.

Last but not least... lembre-se sempre de:

- Ler as Instruções para Entrega de EPs no PACA;
- Não postar códigos no Fórum Geral, nem os compartilhar com colegas;
- Não deixar o EP para a última hora;
- Divertir-se programando!