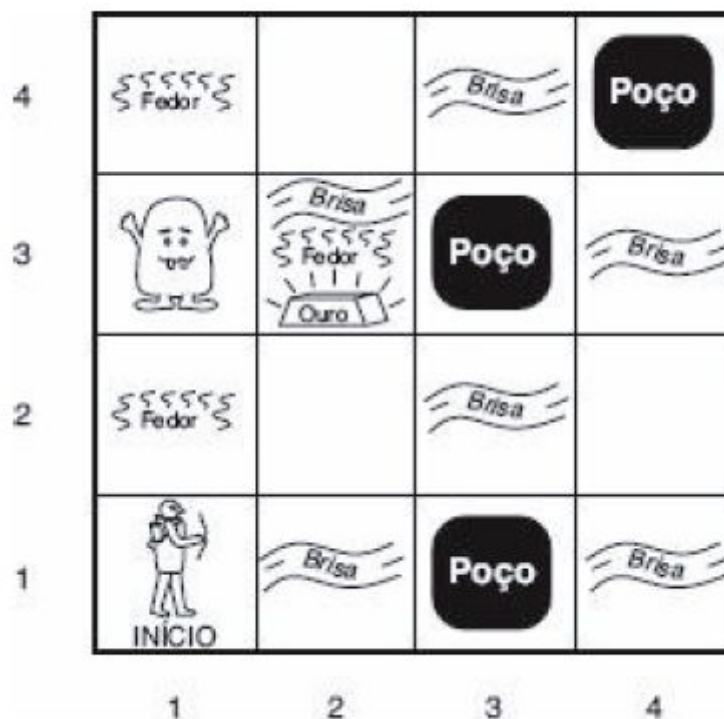


Game em Python com Inteligência Artificial

HUNT THE WUMPUS



O mundo de wumpus é uma caverna que consiste em salas conectadas por passagens. À espreita em algum lugar na caverna está o terrível wumpus, um monstro que devora qualquer guerreiro que entrar em sua sala. O wumpus pode ser atingido por um agente, mas o agente só tem uma flecha. Algumas salas contêm poços sem fundo nos quais cairá qualquer um que vagar por elas (com exceção do wumpus, que é muito grande para cair em um poço). A única característica que abranda esse ambiente desolador é a possibilidade de encontrar um monte de ouro. Embora o mundo de wumpus seja bastante domesticado para os padrões dos modernos jogos de computador, ele ilustra alguns pontos importantes sobre a inteligência.



Medida de desempenho: +1.000 para sair da caverna com o ouro, -1.000 se cair em um poço ou for devorado pelo wumpus, -1 para cada ação executada e -10 pelo uso da flecha. O jogo termina quando o agente morre ou quando sai da caverna.

Ambiente: Uma malha de salas 4×4 . O agente sempre começa no quadrado identificado como [1,1], voltado para a direita. As posições do ouro e do wumpus são escolhidas ao acaso, com uma distribuição uniforme, a partir dos outros quadrados, diferentes do quadrado inicial. Além disso, cada quadrado com exceção do inicial pode ser um poço, com probabilidade 0,2.

Atuadores: O agente pode mover-se ParaFrente, VirarEsquerda 90° ou VirarDireita 90° . O agente terá uma morte horrível se entrar em um quadrado contendo um poço ou um wumpus vivo (é seguro, embora fedorento, entrar em um

quadrado com um wumpus morto). Se um agente tenta mover-se para a frente e esbarra em uma parede, o agente não se move. A ação Pegar pode ser usada para levantar um objeto que está no mesmo quadrado em que se encontra o agente. A ação Atirar pode ser usada para disparar uma flecha em linha reta diante do agente. A flecha continuará até atingir (e, conseqüentemente, matar) o wumpus ou até atingir uma parede. O agente só tem uma flecha e, portanto, apenas a primeira ação Atirar tem algum efeito. Finalmente, a ação Escalar pode ser usada para sair da caverna, mas apenas do quadrado [1,1].

Sensores: O agente tem cinco sensores, cada um dos quais fornece um único bit de informação:

- No quadrado contendo o wumpus e nos quadrados diretamente adjacentes (não em diagonal), o agente perceberá um Fedor.
- Nos quadrados diretamente adjacentes a um poço, o agente perceberá uma Brisa.
- No quadrado onde está o ouro, o agente perceberá um Brilho.
- Quando caminhar para uma parede, o agente perceberá um Impacto.
- Quando o wumpus é morto, ele emite um Grito triste que pode ser percebido em qualquer lugar na caverna.

Percepções: As percepções serão dadas ao programa do agente sob a forma de uma lista de cinco símbolos; por exemplo, se houver um fedor e uma brisa, mas nenhum brilho, impacto ou grito, o programa do agente receberá a percepção [Fedor, Brisa, Nada, Nada, Nada].

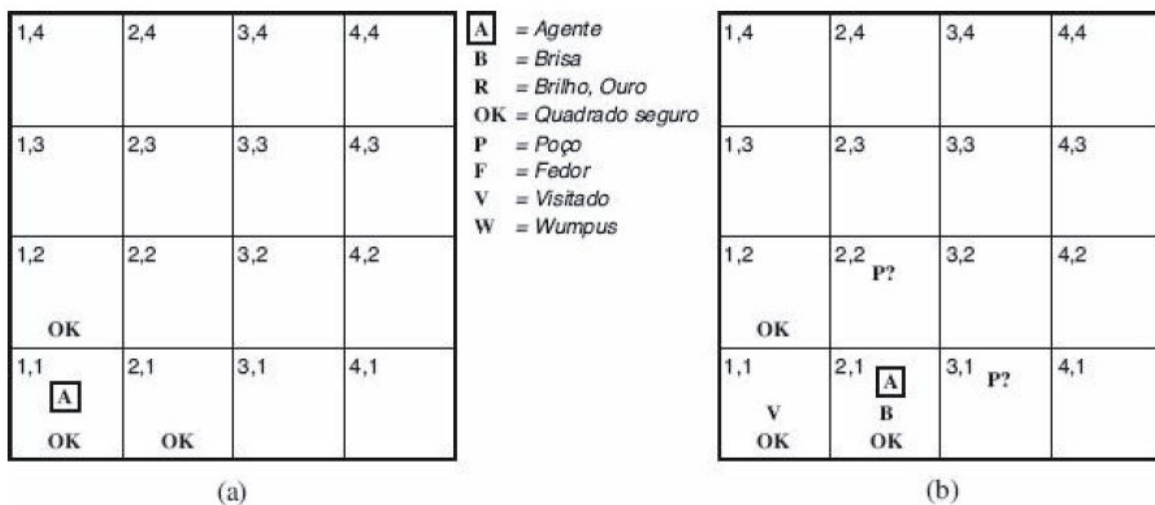
Podemos definir o ambiente de wumpus de acordo com as várias dimensões. Claramente, é discreto, estático e de agente único (o wumpus não se move, felizmente). É sequencial, porque as recompensas podem vir somente após muitas ações serem tomadas. É parcialmente observável, porque alguns aspectos do estado não são diretamente perceptíveis: a localização do agente, o estado de saúde do wumpus e a disponibilidade de uma flecha. Com relação aos locais dos poços e do wumpus: poderíamos tratá-los como as partes não observáveis do estado que são imutáveis — e, nesse caso, o modelo de transição para o ambiente é completamente conhecido ou poderíamos dizer que o modelo de transição em si é desconhecido porque o agente não sabe quais ações ParaFrente são fatais — em cada caso, descobrir a localização dos poços e do wumpus completa o conhecimento do agente do modelo de transição.

Para um agente no ambiente, o principal desafio é a sua ignorância inicial da configuração do ambiente; superar essa ignorância parece exigir raciocínio lógico. Na maioria das instâncias do mundo de wumpus, é possível para o agente recuperar o ouro em segurança. Ocasionalmente, o agente deve escolher entre ir para casa de mãos vazias e se arriscar a morrer para encontrar o ouro. Cerca de

21% dos ambientes são totalmente injustos porque o ouro está em um poço ou cercado por poços.

Vamos acompanhar um agente de wumpus baseado em conhecimento em sua exploração do ambiente. Usamos uma linguagem de representação do conhecimento informal que consiste em escrever símbolos em uma grade. A base de conhecimento inicial do agente contém as regras do ambiente, como descrito anteriormente; em particular, ele sabe que está em [1,1] e que [1,1] é um quadrado seguro; isso está indicado com um “A” e “OK”, respectivamente, no quadrado [1,1].

A primeira percepção é [Nada, Nada, Nada, Nada, Nada], a partir da qual o agente pode concluir que seus quadrados vizinhos, [1,2] e [2,1], são seguros — estão OK.



Um agente cauteloso só se moverá para um quadrado se souber que o quadrado está OK. Vamos supor que o agente decida se mover para a frente até [2,1]. O agente detecta uma brisa (indicada por “B”) em [2,1] e, assim, deve haver um poço em um quadrado vizinho. O poço não pode estar em [1,1], de acordo com as regras do jogo, e, portanto, deve haver um poço em [2,2], [3,1] ou ambos. A notação “P”? na figura (b) indica um possível poço nesses quadrados. Nesse momento, existe apenas um quadrado conhecido que está OK e que ainda não foi visitado. Desse modo, o agente prudente voltará, retornará a [1,1] e depois prosseguirá para [1,2].

O agente percebe um fedor em [1,2], resultando no estado de conhecimento mostrado na figura (a). O fedor em [1,2] significa que deve haver um wumpus por perto. No entanto, o wumpus não pode estar em [1,1], pelas regras do jogo, e não pode estar em [2,2] (ou o agente teria detectado um fedor quando estava em [2,1]). Portanto, o agente pode deduzir que o wumpus está em [1,3]. A notação W! indica essa inferência. Além disso, a falta de uma brisa em [1,2] implica que não existe nenhum poço em [2,2]. O agente já inferiu que deve haver um poço em [2,2] ou

[3,1], então isso significa que ele tem de estar em [3,1]. Essa inferência é bastante difícil porque combina o conhecimento obtido em diferentes instantes em diferentes lugares e se baseia na falta de uma percepção para dar um passo crucial.

1,4	2,4	3,4	4,4
1,3 W!	2,3	3,3	4,3
1,2 A S OK	2,2 OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

A = Agente
B = Brisa
R = Brilho, Ouro
OK = Quadrado seguro
P = Poço
F = Fedor
V = Visitado
W = Wumpus

1,4	2,4 P?	3,4	4,4
1,3 W!	2,3 A S G B	3,3 P?	4,3
1,2 S V OK	2,2 V OK	3,2	4,2
1,1 V OK	2,1 B V OK	3,1 P!	4,1

(a)

(b)

O agente agora provou a si mesmo que não existe nem poço nem wumpus em [2,2], então é OK mover-se para lá. Não mostraremos o estado do agente baseado em conhecimento em [2,2]; apenas supomos que o agente se volta e vai para [2,3], gerando a figura (b). Em [2,3], o agente descobre um brilho e, assim, deve agarrar o ouro e voltar para casa. Observe que, em cada caso no qual o agente tira uma conclusão a partir das informações disponíveis, essa conclusão tem a garantia de ser correta se as informações disponíveis estiverem corretas. Essa é uma propriedade fundamental do raciocínio lógico.

Vamos jogar?

Seja Bem-vindo à Caçada ao Wumpus!

Este jogo é um conjunto de módulos Python para o game Hunt the Wumpus, com módulo de Inteligência Artificial.

Descrição do jogo (em inglês): http://en.wikipedia.org/wiki/Hunt_the_Wumpus

Como jogar

O agente está localizado em uma caverna com 16 quartos (uma grade 4x4). O objetivo é encontrar o ouro em um desses quartos e voltar à entrada. Mas a caverna contém também o Wumpus, uma besta que come qualquer um que entra em sua sala, além de zero ou mais poços, que serão uma armadilha para qualquer pessoa que entrar na sala, exceto para o Wumpus.

O agente pode perceber várias informações de acordo com sua localização atual:

1. No quadrado contendo o wumpus e nos quadrados adjacentes diretamente (não diagonais), o agente perceberá um Stench (cheiro ruim).
2. Nos quadrados diretamente adjacentes a um poço, o agente perceberá uma Brisa.
3. Na praça onde está o ouro, o agente perceberá um Brilho.
4. Quando o wumpus é morto, ele emite um grito que pode ser percebido em qualquer lugar na caverna.

O agente pode executar cinco ações diferentes:

1. Um simples movimento para a frente.
2. Uma simples virada à esquerda por 90°.
3. Uma simples virada à direita por 90°.
4. A ação Pegar pode ser usada para pegar ouro quando estiver na mesma sala do ouro.
5. A ação Atirar pode ser usada para disparar uma flecha em uma linha reta na direção atual que o agente está seguindo. A flecha continua até atingir e matar o wumpus ou atingir uma parede.

Requisitos

Para jogar o Caçada ao Wumpus você precisa ter o Python 3.x instalado. Descompacte o arquivo zip em anexo em um diretório, abra o prompt de comando e navegue até o diretório com os arquivos.

Iniciar o jogo

Para iniciar, basta executar o script do módulo principal (que faz a chamada a todos os outros módulos) digitando:

python game.py

Você também pode deixar o módulo de Inteligência Artificial jogar por você, acrescentando o argumento '-ai':

python game.py -ai

Finalmente se você quiser jogar com a mesma configuração várias vezes você pode especificar o seed:

python game.py -seed 0

Obs: Você pode alterar a permissão do arquivo game.py e torná-lo um executável e desta forma, executa o game sem a necessidade de usar a palavra python antes do nome do arquivo.

No terminal Linux, digite:

chmod 755 game.py (para alterar a permissão do arquivo)

./game.py

./game.py -ai

./game.py -seed 0

Referências:

Livro: Inteligência Artificial

Autor: Peter Norvig

PS: Está disponível na sala virtual da disciplina no SIGAA alguns scripts Python e um Jupyter Notebook com os exemplos de agentes lógicos, procedimentos de avaliação de operadores lógicos, definição de operadores lógicos em Python e definição de bases de conhecimento proposicionais e de lógica de primeira ordem.

Analise os scripts e estude atentamente o Jupyter Notebook.

Você também encontrará a implementação completa deste game de Inteligência Artificial que cobre o conteúdo estudado até aqui.