

INSTRUÇÕES

- **Preencha HONESTAMENTE o formulário abaixo.**
- **Certifique-se que o trabalho em avaliação foi baixado do moodle.**
- Não esqueça de indicar os nomes dos avaliados e dos avaliadores.
- O **avaliado** deve carregar a versão PDF da **SUA AVALIAÇÃO** no moodle (se foi feita em duplas, cada avaliado deve carregar uma cópia).
- A nota começa em 10 e será descontada segundo os critérios abaixo descritos.
- Tempo máximo: 40 minutos

DADOS DA AVALIAÇÃO

Nomes dos avaliados:

1. Eduardo Vanderlei
2. Weslley Picone

Nome dos avaliadores:

1. Luca Almeida Magalhães
- 2.

NOTA FINAL

Preencher a coluna pontuação para cada um dos itens e somá-las ao final da avaliação.

Critério	MÁX	Pontuação
Formulação do problema	3	3
Solução e execução	3	3
Arquitetura agente	4	0,95
TOTAL	10,0	6,95

FORMULAÇÃO DO PROBLEMA

Enunciado: requisitos diversos de implementação

VALOR TOTAL (VT): 3,0

Critério	Descontar	Desconto
Na classe Agente, foram invocados os métodos abaixo corretamente? -Problema. defEstIni (8,0) -Problema. defEstObj (2,8)	-0,2 por invocação faltante ou incorreta	0
O atributo Problema. crencaLabir foi corretamente instanciado com o labirinto do enunciado?	Até -0,2 em função das diferenças em relação ao pedido.	0
O método Problema.suc(Estado, Acao) calcula corretamente o estado sucessor? - o sucessor está dentro dos limites do tabuleiro? - o sucessor não coincide com paredes?	Caso não, -0,2 Caso coincida, -0,2	0

	SOMATÓRIO (A)	
	NOTA PARA O ITEM = VT + A	

Comentários:

SOLUÇÃO E EXECUÇÃO

Enunciado: o agente deve conter um plano armazenado e executar uma ação por vez.

VALOR TOTAL (VT): 3

Critério	Descontar	Desconto
Na classe Agente , há uma solução (plano) armazenado formado por uma sequência de ações a serem executadas pelo agente?	-1,0	0
Foi implementado o método Agente.executarIr(direção) ? Caso implementado, este método invoca o método Model.ir(direção) ?	-0,5 -0,3	0
Foi implementado o método Agente.sensorPosicao() ? Caso implementado, este método invoca o método Model.lerPos() ?	-0,5 -0,3	0
O método Agente.deliberar() consome uma ação por vez (a cada invocação) da solução?	-1	0
No método Agente.deliberar() foram implementados todos os prints solicitados: - estado atual - ações possíveis - ação escolhida - custo acumulado	-0,1 por print faltante	0
	SOMATÓRIO (A)	
	NOTA PARA O ITEM = VT + A	

Comentários:

ARQUITETURA: GOAL-BASED AGENT

Enunciado: mapear a arquitetura de agente baseado em objetivos para a implementação Java fornecida pelo professor.

VALOR TOTAL (VT): 4

Critério	Descontar	Desconto
2. Mapeamento: figura x implementação do professor	-2 se não fez	-2
Environment → class Model	-0,1	
Sensors → Estado Agente.sensorPosicao()	-0,1	
Actuators → Agente.executarIr(direção)	-0,1	

State → crenças do agente sobre o estado do mundo (Problema.crencaLabir e estAtu)	-0,2	
What the world is like now → é a atualização do atributo do agente estAtu	-0,1	
How the world evolves e What my actions do → é a função sucessora que diz qual o próximo estado baseado no estado atual e uma ação escolhida.	-0,2	
What it will be like if I do action A → é a combinação de ações possíveis em um estado s com o resultado da função sucessora $suc(s, a): (s, a) \rightarrow s'$. Primeiro o agente deve saber quais ações são possíveis e, em seguida, conhecer a consequência (o estado sucessor) se executasse cada uma delas	-0,2	
Goals → é o estado objetivo (só tem um objetivo neste problema)	-0,1	
What action I should do now → Agente.deliberar()	-0,2	
3.1. Quantos planos de ação são possíveis para sair de S_0 e alcançar S_g ? R: <i>infinitos planos são possíveis, já que o agente pode ir e voltar inúmeras vezes de uma posição a outra.</i>	Se respondeu: muitos: -0,25 alguns: -0,40 um ou menos: -0,5	-0,4
3.2. Qual o tamanho do espaço de estados e como pode ser calculado? R: <i>linhas x colunas - posições ocupadas pelas paredes = 81 - 28 = 53</i>	Se errou a fórmula: -0,5 Se errou somente os valores da fórmula e obteve valor diferente de 53: -0,2	0
3.3. Quais são os conhecimentos/crenças que o agente deve ter acerca do ambiente para que possa executar o plano? R: crenças sobre: - estado inicial - estado objetivo - posição atual - crenças sobre o labirinto: tamanho e posições das paredes	-0,1 por item faltante	-0,3
3.4. Em todo e qualquer problema, as crenças do agente sempre correspondem ao estado real ou simulado do mundo? R. <i>Não, as crenças do agente podem divergir em relação ao estado real do ambiente.</i>	-0,2 se disse que não divergem em todo e qualquer problema	-0,2
O que ocorre no caso de divergências entre a representação que o agente possui do ambiente e o estado real do ambiente? R. <i>O agente pode tomar decisões erradas e tentar se movimentar para fora do labirinto e/ou atravessar uma parede.</i>	-0,15 se não disse que as ações podem ser equivocadas	-0,15

De onde podem vir estas divergências? <i>R. O agente pode ter recebido um mapa diferente da realidade em tempo de projeto. Os sensores do agente podem ser defeituosos, retornando uma posição diferente da real. O atuador pode falhar, fazendo com que o agente não se movimente ou vá para a posição errada. Em todas estas situações, se o agente crê estar em um ambiente determinístico, então ele utilizará uma função sucessora que retornará um estado inconsistente com realidade.</i>	-0,15 se não respondeu	0
	SOMATÓRIO (A)	
NOTA PARA O ITEM	= VT + A	

Comentários: