SINGULAR®
ENGENHARIA, TECNOLOGIA E GESTÃO

Publicação: Fevereiro de 2019

Digital Object Identifier https://doi.org/10.33911/singular-etg.v1i1.16

Modelando um esconde-esconde no *Minecraft* utilizando *Hidden Markov Model*

RONALDO E SILVA VIEIRA,
MATHEUS RODRIGUES LEAL e
LUIZ CHAIMOWICZ.

Universidade Federal de Minas Gerais (UFMG) (e-mail: {ronaldo.vieira, matheus.leal, chaimo}@dcc.ufmg.br

RESUMO - A criação de agentes inteligentes é uma área da inteligência artificial que busca, dentre outros objetivos, construir entidades capazes de desempenhar ações semelhantes às de seres humanos. Dessa forma, é possível modelar cenários e avaliar a tomada de decisão de entidades envolvidas nestes cenários. Este trabalho consiste na construção de agentes inteligentes para participarem da brincadeira de esconde-esconde. Para isso, é desenvolvida uma abordagem probabilística baseada em um *Hidden Markov Model*, utilizando o jogo *Minecraft* como plataforma para sua aplicação. Os experimentos realizados demonstram que a estratégia desenvolvida permite a interação entre os agentes de forma adequada às regras da brincadeira.

PALAVRAS-CHAVE - Agentes Inteligentes, Sistemas Multiagente, Hidden Markov Model, Minecraft.

I. INTRODUÇÃO

Um objetivo fundamental da inteligência artificial (IA) é criar agentes inteligentes que realizem tarefas ou resolvam problemas de forma autônoma, com eficiência no mínimo equiparável à de seres humanos. Em termos gerais, tais agentes obtém informações de seu ambiente através de seus sensores e as interpretam inteligentemente para escolher as ações adequadas para atingir seus objetivos [6].

Dessa maneira, não é incomum a proposta de criação de agentes que procurem agir de forma tão similar quanto possível à de seres humanos, em detrimento de estrita eficiência em alcançar algum objetivo. Neste contexto, o presente trabalho apresenta uma tentativa à simulação com múltiplos agentes da brincadeira tradicional de esconde-esconde através de uma abordagem probabilística.

O esconde-esconde inicia com uma pessoa aguardando, de olhos vendados, outras se esconderem pelas redondezas. Ao som de um sinal ou após um tempo acordado previamente, esta pessoa sai à procura das outras. As outras, no entanto, possuem como objetivo chegar sem ser pegas pela primeira no local onde ela aguardou – a base – a fim de "se salvar". Caso alguém seja pego, esta pessoa ajudará seu algoz a encontrar e pegar as outras. A brincadeira geralmente acaba quando todas as pessoas foram pegas ou salvas.

Neste trabalho, uma versão simplificada da brincadeira de esconde-esconde é usada. Nela, apenas dois agentes se movimentam por um cenário, utilizando funções de utilidade para definir seus próximos destinos. Cada agente considera em sua função de utilidade (i) uma estimativa probabilística da localização do outro agente obtida a partir de um *Hidden Markov Model* (HMM), a fim de possibilitar que os agentes persigam ou evitem o outro; e (ii) a distância do agente em questão para a base, a fim de encorajá-lo a escolher destinos próximos da base para guardá-la ou "se salvar".

A seção II aborda o referencial teórico necessário para o restante do trabalho. Na seção III, alguns trabalhos relacionados são listados. Após, na seção IV, é descrita detalhadamente a metodologia utilizada para reproduzir a brincadeira em meios computacionais utilizando os conceitos de inteligência artificial. Em seguida, na seção V, os resultados obtidos são apresentados e discutidos. Por fim, a Seção VI apresenta as conclusões obtidas a partir dos resultados do trabalho e define etapas futuras que podem ser desenvolvidas para aprimorá-lo.

II. REFERENCIAL TEÓRICO

No contexto de inteligência artificial, processos Markovianos são quaisquer processos estocásticos cuja determinação do próximo estado depende apenas de seu estado atual. O *Hidden Markov Model* (ou, em português, Modelo Oculto de Markov) se trata de uma forma de modelar um processo cujo estado atual não é observável mas pode ser inferido através de evidências indiretas, se baseando na suposição de que é um processo Markoviano. [6]

Neste trabalho, o HMM é utilizado para inferir a localização do agente adversário a partir de evidências: os locais já observados até o momento. Além da lista de locais possíveis e de evidências, o modelo utiliza como parâmetros

(i) uma distribuição de probabilidades inicial da localização do agente adversário; (ii) uma matriz de transição, que descreve como a distribuição se altera ao longo do tempo; e (iii) uma matriz de observação, que descreve como a distribuição se altera de acordo com novas evidências.

Dessa forma, a cada passo de tempo, utiliza-se de operações matriciais eficientes para atualizar a estimativa de localização do agente adversário. Para a criação do cenário gráfico onde os experimentos são executados, utiliza-se o projeto Malmö.

O projeto Malmö [2] é uma plataforma de código aberto para experimentos de IA construída sobre o jogo *Minecraft*. A plataforma oferece a possibilidade da criação de agentes que atuam como um personagem do jogo em um mundo aberto, dando acesso a informações de sua visão e permitindo ações como andar, pular, olhar para direções específicas ou usar itens.

Embora a plataforma ofereça maior suporte a tarefas de aprendizado por reforço, qualquer aplicação de IA com um ou múltiplos agentes é realizável. Em [9], por exemplo, um agente é treinado com aprendizado por reforço profundo para sobreviver contra uma horda de zumbis, e em [5] outro agente é treinado para escapar de labirintos gerados aleatoriamente.

III. TRABALHOS CORRELATOS

Com o recente avanço no desenvolvimento de pesquisas na área de inteligência artificial, tornou-se necessária a busca por cenários adequados e desafiadores para experimentação de novas técnicas. Nos últimos anos, um dos cenários que vem sendo utilizados para esta finalidade é o jogo *hide-and-seek* (em português, esconde-esconde). A partir disso, os trabalhos buscam aprimorar as estratégias dos agentes envolvidos no jogo, ou modelar um problema real de acordo com as regras do jogo e utilizar estratégias já conhecidas para resolvê-los.

Em [1], o jogo foi utilizado para estudar a interação entre robôs e humanos, com ênfase na tarefa de busca envolvida no jogo. Para o desenvolvimento da estratégia, foi utilizado um Processo de Decisão de Markov de Observação Mista (MOMDP) com o intuito de reduzir o espaço de busca do problema. A partir disso, foram realizadas algumas simulações para comparar o desempenho de métodos padrão com uma nova heurística também criada pelos autores. Os resultados comprovam que a heurística desenvolvida supera as estratégias concorrentes em praticamente todos os cenários.

Em [3], o jogo também foi utilizado para estudar novas estratégias. No entanto, nesse contexto as características e as regras do jogo foram incrementadas. A principal diferença é relacionada ao papel do agente que se esconde, denominado *Hider*, que precisa desempenhar algumas atividades para conseguir "sobreviver" enquanto tenta fugir. Além disso, todas as atividades desempenhadas por ambos os agente emitem um "ruído", que pode ser identificado pelo agente adversário. Dado o cenário proposto e após um conjunto de simulações, foi obtida uma solução analítica para todos os casos.

Além de trabalhos cujos objetivos são encontrar melhores formas de jogar o jogo, existem trabalhos focados na modelagem de problemas reais em cenários do jogo, a fim de aplicar as estratégias já conhecidas para resolvê-los. Esse é o caso de [8], que consiste em uma ferramenta de modelagem específica para o esconde-esconde. Essa ferramenta permite definir o cenário do jogo, assim como a quantidade de agentes participantes e os seus respectivos objetivos. Além disso, também fornece suporte para simulação, visualização e análise das estratégias de ambos os tipos de agentes sob diversos ambientes.

IV. METODOLOGIA

Embora a brincadeira original possa envolver uma quantidade indeterminada de pessoas em ambientes diferentes, aqui limita-se a participação em apenas dois agentes: um para procurar, denominado *Seeker*, e outro para se esconder, denominado *Runner*. Além disso, utiliza-se um cenário 2D em formato de labirinto como local do esconde-esconde. Como plataforma gráfica, utiliza-se o jogo *Minecraft* [4] através da plataforma Malmö.

Como forma de discretizar o ambiente contínuo fornecido pelo jogo, foi criado um grafo onde cada nó representa um local possível do cenário e arestas definem quais locais são acessíveis desde quais. Da mesma forma, guarda-se ainda a informação de visibilidade dos locais, isto é, quais locais são visíveis a partir de quais. A Figura 1 mostra o cenário 2D e o grafo propriamente dito.

Na simulação, o agente *Seeker* tem como posição inicial sua base, disposta em uma das extremidades do labirinto, enquanto o agente *Runner* inicia em um dos locais mais próximos da extremidade oposta. O *Seeker* possui como objetivo encontrar o *Runner*, enquanto o *Runner* deseja chegar à base sem ser pego. É importante notar que, embora o cenário tenha sido discretizado, o tempo e a movimentação dos agentes ainda é contínua e "ser pego" é interpretado como ambos os agentes estando perto o suficiente.

Ao iniciar a simulação, os agentes devem escolher sequencialmente novos locais para ir de acordo com algum critério. Deseja-se que este critério culmine em movimentos inteligentes dos agentes em prol de seus objetivos (chegar à base ou encontrar o outro). Para tal, foi sugerida uma abordagem probabilística a ser descrita a seguir.

Um agente pode se mover apenas para algum dos locais vizinhos do local em que está. Dada essa restrição, permanece ainda a escolha de para qual local ir. E para realizar essa decisão, atribui-se a cada local vizinho um valor de utilidade, isto é, um valor que determina quão bom é ir para um determinado local, segundo o próprio agente com as informações que possui. O local então é escolhido de forma probabilística com valores de probabilidade proporcionais aos respectivos valores de utilidade.

O valor de utilidade de um local para os agentes é calculado com base em onde ele acredita que o outro está e o quão longe o local é da base. Mais especificamente, na probabilidade do outro agente estar no local considerado e

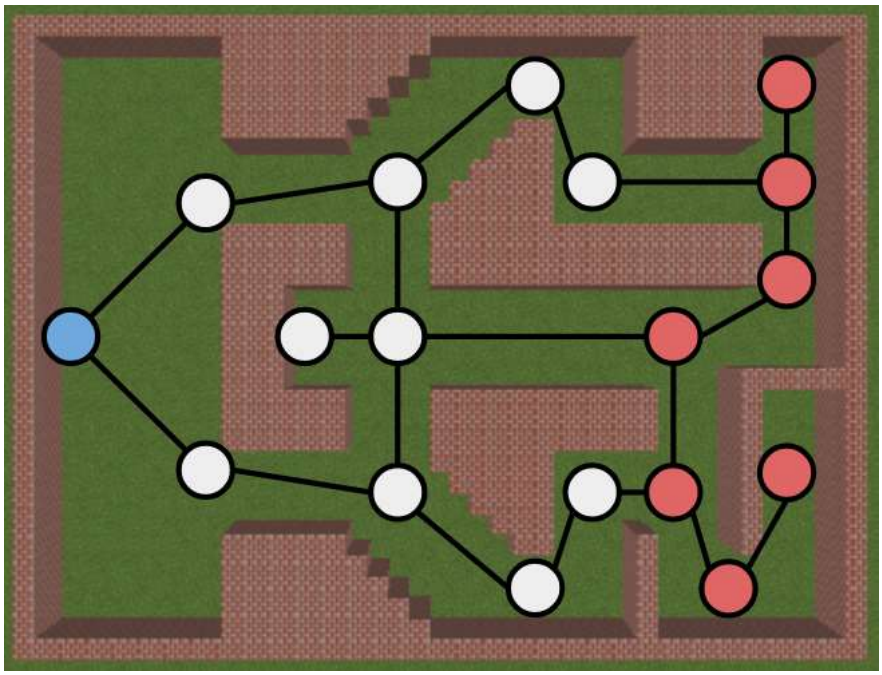

Figura 1. Cenário 2D utilizado como ambiente da simulação e grafo de locais. Em azul e vermelho, respectivamente, as posições iniciais dos agentes *Seeker* e *Runner*. O último inicia em uma das posições destacadas escolhida de forma aleatória.

na distância euclideana deste para a base. Formalmente, o valor de utilidade $U(s)$ de um estado s é dado pela seguinte equação:

$$U(s) = \begin{cases} \alpha \mathrm{P}(\text{loc. o. ag.} = s) + \dfrac{\beta}{d(s, \text{base})} & \text{para o } \textit{Seeker} \\ \alpha \mathrm{P}(\text{loc. o. ag.} \neq s) + \dfrac{\beta}{d(s, \text{base})} & \text{para o } \textit{Runner} \end{cases}$$

Onde α e β representam pesos a serem dados aos dois fatores envolvidos no cálculo. É importante notar que, enquanto o *Seeker* deseja encontrar o outro agente, este não deseja ser encontrado. Por isso, o *Runner* considera o inverso da probabilidade utilizada pelo *Seeker*. Por outro lado, ambos utilizam o inverso da distância do local até a base, de forma a encorajá-los a ir para locais próximos da base – um para guardá-la e outro para atingir seu objetivo.

Enquanto o cálculo da distância euclideana entre locais seja trivial e possa até ser pré-calculado, o mesmo não é válido para a probabilidade, por sua natureza dinâmica. Portanto, para calcular tal probabilidade, foi utilizado um *Hidden Markov Model* [7].

Em sua modelagem, utiliza-se como evidências a localização do agente em questão e dos locais que ele consegue enxergar de acordo com as informações de visibilidade do grafo. Com essas evidências, deseja-se inferir $\mathbf{P}(\text{loc. o. ag.})$, isto é, a distribuição de probabilidades que indica a localização do outro agente. A Figura 2 apresenta a modelagem em forma gráfica, com a inclusão do fator temporal.

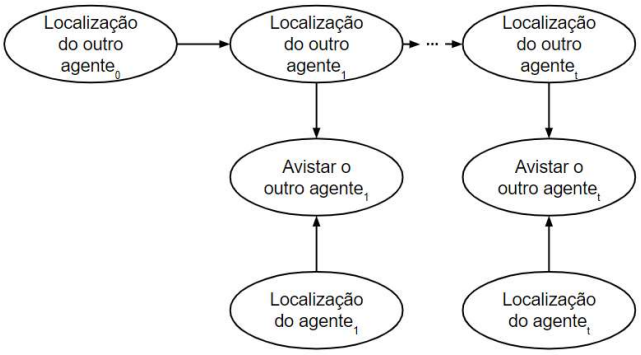

Figura 2. Modelo markoviano utilizado para o cálculo da distribuição de probabilidades da localização do agente adversário.

O primeiro dos parâmetros necessários para um HMM é distribuição de probabilidades $\mathbf{P}(\text{loc. o. ag.}_0)$, isto é, a probabilidade para cada local no início da simulação. Estas são definidas de forma homogênea, pois sem nenhuma evidência, o outro agente pode estar em qualquer um dos dezoito locais com a mesma probabilidade.

Outro parâmetro é o modelo de observação. Ele se refere à aplicação das evidências: a forma como as probabilidades de localização do agente adversário mudam conforme os locais que o agente em questão consegue enxergar. O modelo de observação é definido, informalmente, como:

(i) Se o agente adversário está em um dos locais que o

agente em questão consegue enxergar, é atribuída probabilidade 1 ao local onde ele está e 0 para todos os outros.

(ii) Se o agente adversário **não** está em um dos locais que o agente em questão consegue enxergar, é atribuída probabilidade 0 aos locais visíveis pelo agente, enquanto os locais não visíveis mantém sua distribuição de probabilidades.

Dessa forma, ao ver o outro agente, o modelo se torna certo de sua posição. Em contrapartida, caso não o esteja vendo, pode-se apenas ter certeza de que ele não está nos locais visíveis.

Por último, é necessário definir o modelo de transição – como a distribuição de probabilidades se comporta com o passar do tempo. Em outras palavras, deseja-se calcular $\mathbf{P}\text{loc. o. ag.}_t|\text{loc. o. ag.}_{t-1}$ e, para tal, é necessário calcular uma quantidade de probabilidades do tipo $P\text{loc. o. ag.} = s'|\text{loc. o. ag} = s$, para todos os estados s' e s.

Conforme o tempo passa, é improvável que o agente esteja no mesmo local e mais provável que esteja em algum de seus vizinhos, com peso maior para os vizinhos mais próximos. Dessa forma, o modelo de transição possui o comportamento de decair a probabilidade do agente estar em um local, se ele estava nele anteriormente, e elevar a probabilidade do agente estar em um estado vizinho. Esta alteração se dá de forma proporcional à distância entre os dois locais em consideração.

Com a probabilidade inicial e os modelos de observação e transição especificados, todos os componentes necessários para a execução do algoritmo estão definidos. No início da execução, os agentes são dispostos no labirinto em suas respectivas posições iniciais e recebem como conhecimento inicial sobre a localização do adversário uma distribuição de probabilidades uniforme sobre todos os estados. Os agentes realizam os seguintes passos:

1) Cada agente verifica se o adversário está em seu campo de visão.
2) Caso esteja, a matriz do modelo de observação é definida para indicar a certeza da localização do adversário em um estado específico.
3) Se não, a matriz do modelo de observação é definida para indicar a certeza de que o adversário não está nos estados vizinhos.
4) Em seguida, a matriz do modelo de transição é utilizada para atualizar a distribuição de probabilidades de acordo com a possível movimentação do adversário.
5) Por último, a matriz do modelo de observação é multiplicada pela distribuição de probabilidades resultante, atualizando o conhecimento do agente de acordo com a sua observação.
6) A partir disso, são calculados os valores de utilidade dos estados vizinhos. A próxima localização dos agentes é então selecionada probabilisticamente.

A cada iteração, estes passos são repetidos até que o *Seeker* alcance o *Runner* ou o *Runner* consiga chegar à base.

V. RESULTADOS

Pela natureza do problema, uma análise qualitativa foi realizada para avaliação da simulação. Esta análise foi executada diversas vezes e observações de comportamentos adequados ou inadequados foram notadas. Os parâmetros α e β que culminaram em maior realismo foram, respectivamente, $0{,}9$ e $0{,}1$. A Figura 3 mostra alguns momentos das simulações. Duas execuções representativas de ambos os agentes completando seus objetivos está disponível em https://youtu.be/SkSCwN-mF44 e https://youtu.be/Er4B11D6_3E.

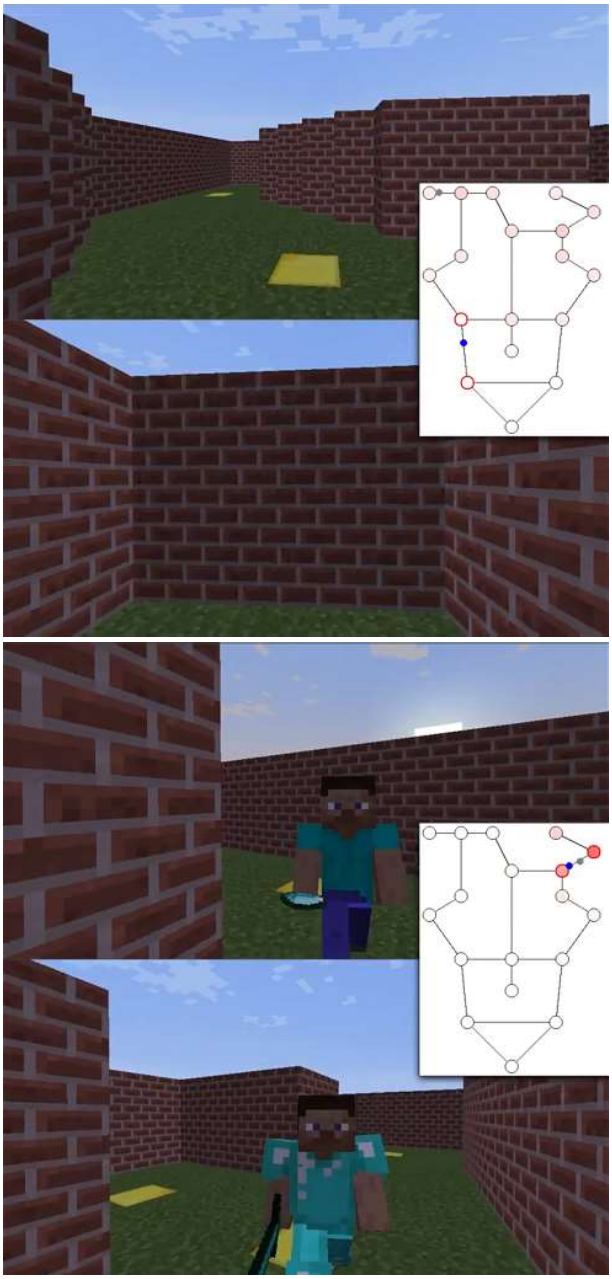

Figura 3. Dois momentos capturados nos experimentos. Em cada momento, as visões de cima e de baixo são, respectivamente, do *Seeker* e do *Runner*. A janela mais à direita contém uma representação visual do grafo de locais, probabilidades e posicionamento dos agentes.

Pôde-se perceber que a aplicação da abordagem probabilística proposta resulta em dois comportamentos desejados: o agente *Seeker* procurando ativamente o *Runner* e este, por sua vez, procurando fugir – principalmente quando visto. No entanto, o processo de fuga frequentemente termina no agente encurralado por seu adversário, apesar de ser encorajado a ir em direção da base.

Percebe-se que a fonte de alguns comportamentos inadequados como fugir em direção a um beco sem saída e não aproveitar chances de chegar próximo à base advém da capacidade do agente de apenas considerar probabilidades localmente, isto é, não levar em consideração a probabilidade dos vizinhos de um local para calcular sua probabilidade.

Além disso, por ser uma abordagem probabilística, é incomum, porém possível, que os agentes tomem decisões claramente ruins. Por outro lado, este pode ser um fator beneficente ao que a abordagem almeja: realismo em vez de perfeição. Dado que na maioria absoluta das vezes o comportamento esperado dos agentes seria também o comportamento esperado de seres humanos na brincadeira.

VI. CONCLUSÃO

Foi proposta uma abordagem probabilística para simular uma brincadeira de esconde-esconde com agentes inteligentes. Modelou-se um cenário 2D em formato de labirinto com estratégias para discretizar o espaço e o tempo. Os agentes no cenário se movimentam em função de um cálculo de qual é o melhor local para ir que, por sua vez, utiliza um *Hidden Markov Model* e cálculos de distâncias.

A simulação atinge os resultados esperados, embora faça emergir algumas sequências de comportamentos improváveis em um cenário real. Uma melhoria natural para a escolha do próximo local a ir é a utilização de um Processo de Decisão de Markov para que se efetuem cálculos de utilidade de forma global.

Além disso, apesar da simplicidade obtida ao discretizar o espaço através de um grafo, isto restringe a quantidade de movimentos possíveis dos agentes e também acarreta no comportamento improvável de nunca mudar de direção uma vez que inicia um percurso entre um local e outro. Um próximo passo sugerido é ampliar o escopo da simulação para utilizar mais agentes. Este passo naturalmente necessitaria de implementações de políticas de cooperação e comunicação entre agentes, como descrito em [10].

O código-fonte deste trabalho está disponível em https://github.com/j-ufmg/mine-and-seek.

Referências

[1] A. Goldhoorn, A. Sanfeliu, and R. Alquézar. Comparison of momdp and heuristic methods to play hide-and-seek. In CCIA, 2013.
[2] M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The malmo platform for artificial intelligence experimentation. In IJCAI, pages 4246–4247, 2016.
[3] G. McCormick and G. Owen. A composite game of hide and seek. Central European Journal of Operations Research, 27(1):1–14, Mar 2019.
[4] Mojang. Minecraft, 2009.
[5] M. Monfort, M. Johnson, A. Oliva, and K. Hofmann. Asynchronous data aggregation for training end to end visual control networks. In Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, pages 530–537. International Foundation for Autonomous Agents and Multiagent Systems, 2017.
[6] S. J. Russell and P. Norvig. Artificial intelligence: a modern approach. Malaysia; Pearson Education Limited,, 2016.
[7] R. L. Stratonovich. Conditional markov processes. Theory of Probability & Its Applications, 5(2):156–178, 1960.
[8] A. Tandon and K. Karlapalem. Medusa: Towards simulating a multi-agent hide-and-seek game. In Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18, pages 5871–5873. International Joint Conferences on Artificial Intelligence Organization, 7 2018.
[9] H. Udagawa, T. Narasimhan, and S.-Y. Lee. Fighting zombies in minecraft with deep reinforcement learning. Technical report, Technical report, Stanford University, 2016.
[10] M. Wooldridge. An introduction to multiagent systems. John Wiley & Sons, 2009.

RONALDO E SILVA VIEIRA Possui graduação em Ciência da Computação pela Universidade Federal Rural do Rio de Janeiro (2017). Atualmente é mestrando em Ciência da Computação pela Universidade Federal de Minas Gerais, pesquisador, sob a orientação do professor Luiz Chaimowicz, no Laboratório de Inteligência Artificial Aplicada a Jogos do Departamento de Ciência da Computação na mesma instituição. Tem experiência na área de Inteligência Artificial, com ênfase em Inteligência Artificial Aplicada a Jogos e Aprendizado por Reforço.

MATHEUS RODRIGUES LEAL Possui graduação em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (2017). Atualmente é mestrando em Ciência da Computação pela Universidade Federal de Minas Gerais, pesquisador, sob a orientação da professora Gisele Lobo Pappa, no Laboratório de Inteligência Computacional do Departamento de Ciência da Computação na mesma instituição. Tem experiência na área de Ciência dos Dados e Aprendizado de Máquina, com ênfase em Processamento de Linguagem Natural e Representação do Conhecimento.

LUIZ CHAIMOWICZ Professor Associado do DCC - UFMG e Bolsista de Produtividade em Pesquisa Nível 2 do CNPq. Ele é doutor em Ciência da Computação pela UFMG (2002) e realizou um pós-doutorado em Robótica na Universidade da Pennsylvania, EUA (2003-2004). Suas áreas de interesse incluem robótica móvel, coordenação de múltiplos robôs, plataformas para programação e simulação de robôs móveis e desenvolvimento de jogos digitais, especialmente inteligência artificial aplicada a jogos.