

## DESENVOLVIMENTO DE SIMULAÇÕES BASEADAS EM AGENTES E MELHORIAS NA FERRAMENTA NETLOGO

Luiz Fernando de Jesus Portela<sup>2</sup>, Fernando dos Santos<sup>3</sup>

<sup>1</sup> Vinculado ao projeto “Aprimorando o Desenvolvimento de Simulações Baseadas em Agentes por meio de Blocos de Construção Abstratos”

<sup>2</sup> Acadêmico do Curso de Engenharia de Software - CEAVI - Bolsista PIVIC

<sup>3</sup> Orientador(a), Departamento de Engenharia de Software - CEAVI - [fernando.santos@udesc.br](mailto:fernando.santos@udesc.br)

O paradigma de modelagem baseada em agentes autônomos aborda a interação entre agentes autônomos simulando ambientes singulares, nos quais o comportamento é normalmente obscuro. Ele permite reproduzir cenários que seriam complexos de serem reproduzidos de forma prática, provendo casos específicos e com uma abordagem apropriada (O'MALLEY; DELOACH, 2002).

As técnicas de aprendizagem por reforço são formas de treinar os agentes para que eles adquiram conhecimento ao serem expostos ao ambiente onde estão situados. Quando o agente está em contato com o ambiente, ele é exposto a um estado e conseqüentemente uma ação é tomada, então um novo estado é direcionado ao ambiente. As recompensas de aprendizagem por reforço são utilizadas nos algoritmos que auxiliam a detectar qual a melhor ação para determinado estado. Dessa maneira, o agente consegue aprender qual a melhor forma de responder ao ambiente (BUŞONIU; BABUŠKA; SCHUTTER, 2010).

Devido a sua relevância, há diversas ferramentas sendo produzidas e evoluídas voltadas para o desenvolvimento de simulações baseadas em agentes. Uma delas é a plataforma NetLogo, sendo extremamente poderosa para modelagem científica, de fácil uso e ótima documentação (LYTINEN; RAILSBACK, 2012). Além de possuir bibliotecas nativas, também possui diversas opções externas desenvolvidas pela comunidade e que promovem resultados robustos. A técnica de aprendizagem por reforço é disponibilizada por meio de uma biblioteca desenvolvida por Bazzanella (2022). Esta biblioteca permite que os seguintes algoritmos de aprendizagem por reforço sejam utilizados diretamente no NetLogo: Q-Learning, SARSA, e Actor-Critic.

O presente trabalho teve como objetivo evoluir a biblioteca de aprendizagem por reforço de Bazzanella (2022), para permitir o armazenamento e a recuperação de dados de aprendizagem dos agentes que foi gerado na plataforma NetLogo, conforme pode ser visto na Figura 1. Dessa forma, o desenvolvedor NetLogo consegue usufruir e aplicar um refinamento mais robusto no aprendizado, visto que pode realizar a etapa de aprimoramento a partir da última edição de aprendizado gerada pelos agentes. Além disso, facilitando na obtenção dessas informações e replicar os resultados obtidos por meio de outros métodos a critério do usuário.

```
public class ExtensionCommands extends DefaultClassManager { // Eloisa Bazzanella +1 *
    public static void main(String[] args) throws Exception { new *
        new SaveLearningCommand().report( args: null, context: null);
    }

    public void load(PrimitiveManager primitiveManager) { // Eloisa Bazzanella +1 *

        //COMANDOS SETUP
        primitiveManager.addPrimitive( name: "Learning-rate", new LearningRateCommand());
        primitiveManager.addPrimitive( name: "discount-factor", new DiscountFactorCommand());
        primitiveManager.addPrimitive( name: "lambda", new LambdaCommand());
        primitiveManager.addPrimitive( name: "action-selection", new ActionSelectionCommand());
        primitiveManager.addPrimitive( name: "action-selection-greedy", new ActionSelectionEGreedyCommand());
        primitiveManager.addPrimitive( name: "action-selection-random", new ActionSelectionRandomCommand());
        primitiveManager.addPrimitive( name: "reward", new RewardCommand());
        primitiveManager.addPrimitive( name: "end-episode", new IsEndEpisodeCommand());
        primitiveManager.addPrimitive( name: "actions", new ActionsCommand());
        primitiveManager.addPrimitive( name: "state-def", new StateDefinitionCommand());
        primitiveManager.addPrimitive( name: "state-def-extra", new StateDefinitionExtraCommand());
        primitiveManager.addPrimitive( name: "define-algorithm", new DefineAlgorithmCommand());
        primitiveManager.addPrimitive( name: "setup", new SetupCommand());

        //COMANDOS GO
        primitiveManager.addPrimitive( name: "Learning", new LearningCommand()); // TODO debug=true
        primitiveManager.addPrimitive( name: "decay-epsilon", new DecayEpsilonCommand());
        primitiveManager.addPrimitive( name: "get-episode", new GetEpisodeCommand());
        //primitiveManager.addPrimitive("get-qtable", new GetQTableCommand()); // TODO
        primitiveManager.addPrimitive( name: "get-learning-details", new GetQTableCommand());
        primitiveManager.addPrimitive( name: "save", new SaveLearningCommand());
        primitiveManager.addPrimitive( name: "load", new LoadLearningCommand());
    }
}
```

**Figura 1.** Trecho de código NetLogo apresentando os dois novos comandos criados

Para evoluir a biblioteca de aprendizagem, o método consistiu no estudo do código-fonte da BURLAP, que é uma biblioteca utilizada Java que disponibiliza diversos algoritmos de aprendizagem por reforço (BAZZANELLA, 2022). Foram pesquisados mecanismos nativos na BURLAP para salvar os dados de aprendizado dos agentes contidos na memória do programa e um método capaz de carregá-los para uso posterior em um agente ou análise de dados. Foi descoberto que a estrutura é armazenada em arquivo no formato YAML, que é amplamente utilizado em diversas tecnologias. Após o conhecimento adquirido, a biblioteca de aprendizagem NetLogo foi refatorada, para disponibilizar dois novos comandos que permitem salvar e ler o conhecimento aprendido pelo agente.

**Palavras-chave:** Simulações baseadas em agentes. Plataforma NetLogo. Aprendizagem por reforço.

### Referências

BAZZANELLA, E. **Expansão da biblioteca de Aprendizagem por Reforço para desenvolvimento de agentes inteligentes na plataforma Netlogo.** 2022. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Software) – Universidade do Estado de Santa Catarina, Ibirama, 2022.

BUŞONIU, L.; BABUŠKA, R.; SCHUTTER, B. Multi-agent Reinforcement Learning: An Overview. **Studies in computational intelligence**, p. 183–221, 2010.

LYTINEN, S.; RAILSBACK, S. **The Evolution of Agent-based Simulation Platforms: A Review of NetLogo 5.0 and ReLogo**. [s.l.: s.n]. 2012.

O'MALLEY, Scott A ; DELOACH, Scott. Determining When to Use an Agent-Oriented Software Engineering Paradigm. **Lecture notes in computer science**, p. 188–205, 2002.