A Pedagogical Framework for Modeling and Simulating Intelligent Agents and Control Systems

Dan Tappan

College of Engineering, Idaho State University

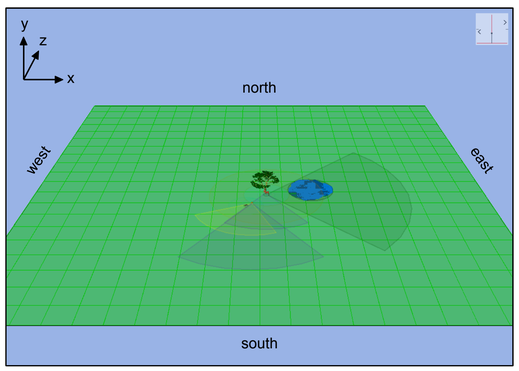
Introduction

Classroom assignments are an effective way for students to investigate many important and fun aspects of AI. Unfortunately, for any project of reasonable breadth and depth, especially involving multiple agents and graphics, most of the programming goes into tedious, error-prone administrative tasks. Time constraints also usually preclude a reusable, extensible design, so this overhead repeats itself throughout the semester.

This Java-based, pedagogy-oriented modeling-and-simulation API framework provides the necessary support capabilities to get students up to speed quickly on playing with AI content. It contains extensive, highly configurable, yet user-friendly, engineering, physics, and communication models for arbitrary components within a definable task environment. These components are managed automatically in a stochastic simulation that allows students to define, test, and evaluate their performance over a wide range of controlled experiments.

Environment

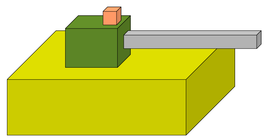
The environment is a three-dimensional tabletop of arbitrary scale. Agents, including the viewer, may be situated anywhere in real time.



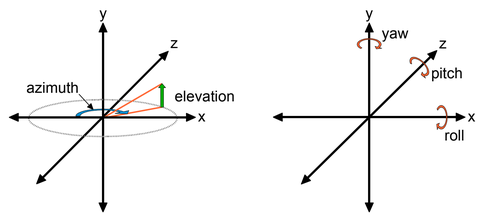
The default model supports naive physics and kinematics like velocity, acceleration, and gravity [1,2]. Students may also substitute their own.

Components

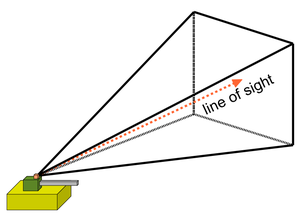
Components are student-defined, three-dimensional functional units that hierarchically compose an agent, such as a tank with a hull, turret, gun, and observation sensor:



They support interconnection models with variations on 5 and 6 degrees of freedom to support practically any mechanical motion:



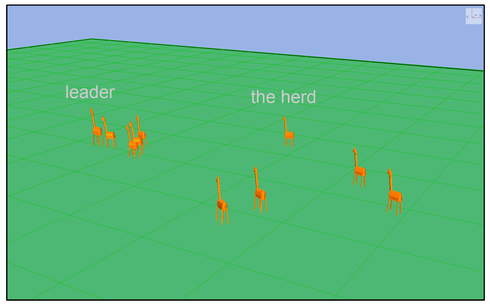
The most common usage is for sensors, which can model optical, thermal, or sonic observation, or anything students can define. A sensor sees within a field of view, with horizontal and vertical angular limits (like eyes), that can optionally move over a field of regard (like eye sockets or neck movement).



This combination reasonably supports any sensory configuration found in nature. It allows students to experiment with variations on different vision models (like eagles, iguanas, and insects) and investigate their relative performance characteristics, pros and cons, etc. It offers a practical and entertaining perspective on nature-inspired computing.

Agents

An agent is a coordinated collection of components for input (sensors), processing (AI core), and output (actuators) that plays some experimental role in the task environment. Wayfinding, leader-following, swarming, flocking, targeting, intercepting, and other behaviors are common scenarios [2]. For example, this herd of giraffes is lackadaisically following the lead alpha wherever he goes:



The AI core is facilitated by a flexible command-and-control framework based heavily on object-oriented design patterns for modeling the structure, assembly, behavior, and communication of agents [3]. It is derived from a large-scale, accredited simulation architecture developed for the Department of Defense and other AI-related research [5,4].

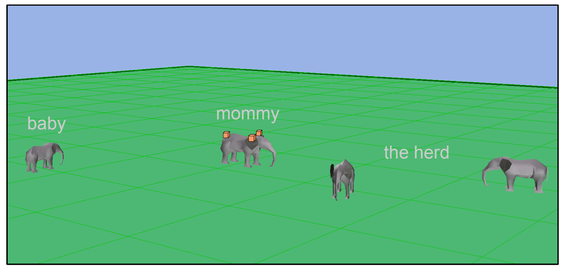
Simulation and Analysis

Agents operate interdependently in real time within a stochastic, discrete-event simulation for controlled experiments. It allows students to determine the baseline performance of their models, then to modify them purposefully in various ways and evaluate the results of the changes.

The framework logs all data of interest for analysis. It automatically manages the stochastic nature of multiple runs to produce individual and aggregate results. Students then use external tools like Excel for analysis and reporting.

Conclusion

Preliminary deployment of this software in an upper-division undergraduate AI class was received well. A number of fun and insightful programming assignments gave students the opportunity to investigate and experiment with more aspects of AI than would have otherwise been possible. For example, can a mother elephant follow the herd and keep a watchful eye on her baby better with three eyes?!



Suggestions for improving the API and integrating it further into the course are being incorporated. Once stable, it will be made freely available for academic use.

References

[1] Bourg, D. 2002. *Physics for Game Developers*. O'Reilly, Sebastopol: CA.

[2] Bourg, D. and Seemann, G. 2004. *AI for Game Developers*. O'Reilly, Sebastopol: CA.

[3] Gamma, E., Helm, R., Johnson, R., and Vlissides, J. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley, Indianapolis: IN.

[4] Tappan, D. 2004. Knowledge-Based Spatial Reasoning for Automated Scene Generation from Text Descriptions. Ph.D. diss. Dept. of Computer Science, New Mexico State University, Las Cruces, NM.

[5] Tappan, D. and Engle, J. 2005. The AMSAA SURVIVE Model. In Proc. *U.S. Army 16th Ground Vehicle Survivability Symposium*, Monterrey, CA.