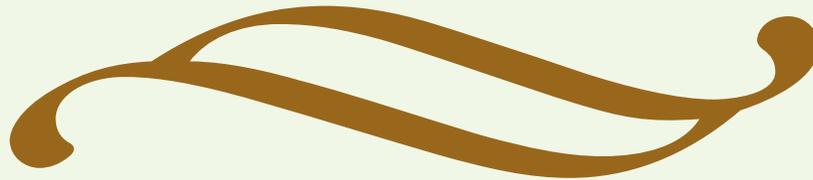Gal A. Kaminka, Manuela M. Veloso, Steve Schaffer, Chris Sollitto, Rogelio Adobbati, Andrew N. Marshall, Andrew Scholer, and Sheila Tejada
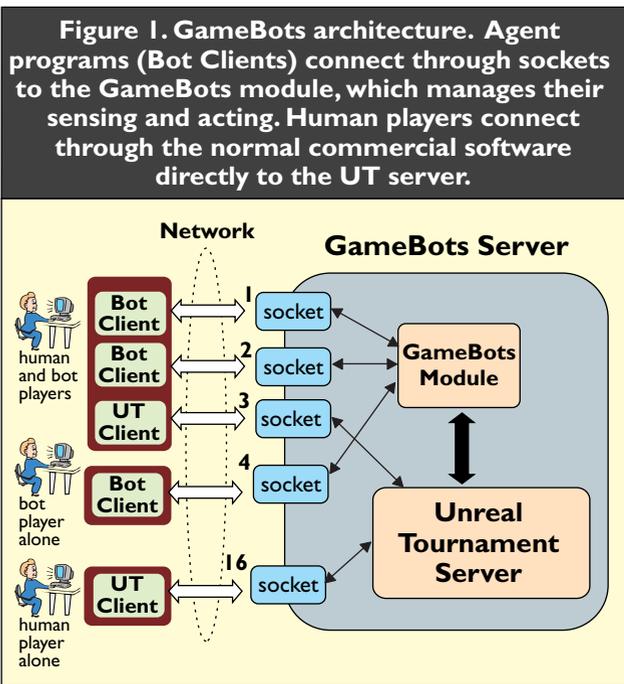
GameBots: A Flexible Test Bed for Multiagent Team Research

GameBots [1] is a virtual reality platform that allows the creation and evaluation of intelligent agents that interact with a rich 3D continuous dynamic environment. As opposed to previous test beds that focus on a single task and environment (such as soccer simulation [4]), GameBots does not define a single benchmark task. Instead, the GameBots platform comes with a wide variety of predefined tasks and environments and allows anyone to extend these in various ways, or create new challenges. This enables multiagent systems (MAS) and artificial intelligence researchers to explore a wide variety of algorithms and techniques, in areas such as spatial navigation, learning, dynamic resource allocation, multiagent planning, plan-recognition, collaboration, distributed adversarial planning, and human-machine teamwork.

GameBots is composed of two components. The first of these is a freely-available open source extension of the commercial Unreal Tournament game engine [3]. It defines a socket-based API allowing anyone to create agents that can participate in any Unreal Tournament games. The second component is a set of development tools, sample source code, and nonviolent graphics (replacements for the default graphics) that form a basic development environment to help users get started in using GameBots.
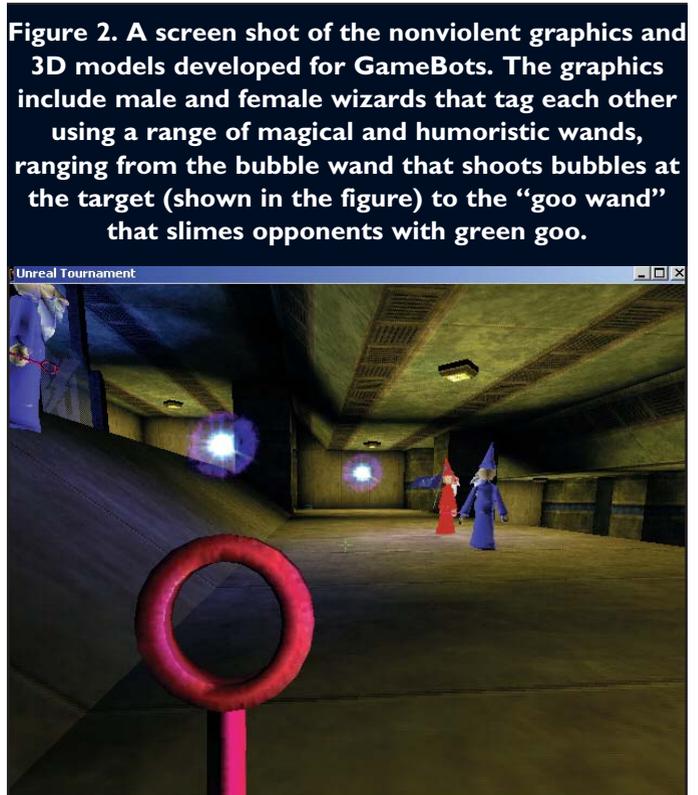
and the environment. These interactions take place in the context of tasks (typically, games) that are defined within the test bed software: The agent programs, through the sensors and actuators, participate in these games. To achieve their goals, agents must cooperate or compete with one another in order to win. The definition of sensor/actuator interfaces is meant to put agent programs and human game-players on nearly equal footing, supporting mixed human-machine studies (see Figure 1). Indeed, this capability for exploring mixed human-agent scenarios is an important characteristic of GameBots.

Another important feature we added to the commercial game was a set of freely available tools and code libraries that can be useful in building agents for GameBots. Sample agent code is available in Tcl/Tk, Java, C/C++, and more. Tools for visualizing the terrain in which agents interact and for logging agents' actions are also available. The game itself comes with a variety of tools supporting the

A quick summary of how GameBots works follows. Gambots supports multiple agents (both human and synthetic) that interact in a variety of multi-agent games. Two interesting built-in games are Domination (where two teams of agents compete with each other for the control of selected locations), and Capture-the-Flag (where two teams of agents attempt to steal each other's flag). The Unreal engine features real-time 3D graphics, as well as sound and agent movement, in different user-designed terrains. The commercial game is modifiable using the built-in programming language to support the introduction of different games, additional 3D models, sounds, alternative terrain maps, and agents with different capabilities.

To turn the commercial software into a test bed, we modified the game using its published programming interfaces. Like other MAS test beds, GameBots clearly defines the sensing and acting capabilities of agents without restricting the reasoning processes. This is achieved by defining socket-based interfaces to which a single program (agent) can connect in a client/server fashion. Through the interface, the agent program can accept sensory information from a GameBots server and can send actuation information back. In other words, the agent program contains the "brains" of a single agent, and the GameBots server controls the "body." The Gamebots server controls the environment and facilitates interactions between agents, and between agents

Figure 2. A screen shot of the nonviolent graphics and 3D models developed for GameBots. The graphics include male and female wizards that tag each other using a range of magical and humoristic wands, ranging from the bubble wand that shoots bubbles at the target (shown in the figure) to the "goo wand" that slimes opponents with green goo.

creation of additional environments, graphics, and extensions of the tasks. For instance, to address our own and general educational concerns we have created nonviolent versions of the graphics and sound (see the screen shots in Figures 2 and 3).

There are several features of the commercial game that we have found particularly useful. First, of

course, is its programming interfaces that allow drastic redesigns of the multiagent tasks (which we use as benchmarks). This means that when new benchmarks are introduced in GameBots, the effort

Figure 3. This screen shot shows the effect of using the bubble wand to "shoot" a bubble at the blue wizard. A bubble-like sound accompanies the action.

involved in creating the infrastructure for building and connecting the agent does not have to be repeated: The agents continue to interact with the same test bed software, but can now engage in a new game. Indeed, the Unreal Tournament gaming community has enjoyed extending the game for years, and thus many modifications, replacement graphics, and additional terrain areas are freely available. Other useful features include the built-in game server, which can be deployed once and left running, supporting continued usage by researchers or students without administrative overhead. The platform runs on Linux, Windows, and the Macintosh OS, and is capable of real-time play over the Internet, allowing distributed development and friendly competition with other researcher groups. Some such competitions have already taken place.

Several different universities and research organizations are utilizing GameBots in their research and education activities. At Carnegie Mellon University, GameBots is being used in undergraduate research projects. SRI uses GameBots in multiagent planning and believability research. GameBots is currently being used in Ph.D. and Master's thesis investigations of planning in the U.K. and Mexico. Exploiting GameBot's flexibility, Evans [2] created a virtual Clue game (also known as Cluedo) in which the famous board game characters come to life to interact with the human players. In addition, some friendly competitions have already taken place. Given the attention GameBots has received from teachers of programming, AI, and MAS courses, there are plans for end-of-semester competitions that will serve to motivate students and focus researchers. Announcements about such events, as well as all software, tools, and documentation, are available at www.cs.cmu.edu/~galk/ GameBots/. Our team is committed to continuing its development of the GameBots platform and the agent development environment. **C**

REFERENCES
1. Adobbati, R., Marshall, A.N., Scholer, A., Tejada, S., Kaminka, G.A., Schaffer, S., Sollitto, C. GameBots: A 3D virtual world test bed for multiagent research. In *Proceedings of the Second International Workshop on Infrastructure for Agents, MAS, and Scalable MAS,* (Montreal, Canada, 2001).
2. Evans, T. *Cluedo.* Master's Thesis, Centre for Virtual Environments, ISI, University of Salford, 2001.
3. Gerstmann, J. Unreal tournament: Action game of the year, 1999. GameSpot; www. gamespot.com/features/1999/p3_01a.html
4. Kitano, H., Tambe, M., Stone, P., Veloso, M., Coradeschi, S., Osawa, E., Matsubara, H., Noda, I., and Asada, M. The RoboCup synthetic agent challenge '97. In *Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence, (*Nagoya, Japan, 1997).

GAL A. KAMINKA (galk@cs.cmu.edu) is a post doctoral fellow in the Computer Science Department at Carnegie Mellon University in Pittsbugh, PA.
MANUELA M. VELOSO (veloso@cs.cmu.edu) is an associate professor in the Computer Science Department at Carnegie Mellon University.
STEVE SCHAFFER (steve.schaffer@jpl.nasa.gov) is a member of the Artificial Intelligence technical staff at NASA Jet Propulsion Laboratory.
CHRIS SOLLITTO is a recent graduate of the Computer Science Department at Carnegie Mellon University.
ROGELIO ADOBBATI (rogelio@isi.edu) is a recent graduate of the Information Sciences Institute at the University of Southern California.
ANDREW N. MARSHALL (amarshal@isi.edu) is a staff member in the Information Sciences Institute at the University of Southern California.
ANDREW SCHOLER (ascholer@usc.edu) is a Master's student in the Information Sciences Institute at the University of Southern California.
SHEILA TEJADA (tejada@isi.edu) is a Ph.D. candidate in the Information Sciences Institute at the University of Southern California.