

John E. Laird

# RESEARCH IN HUMAN-LEVEL AI USING COMPUTER GAMES



**T**he goal of my research group is to understand what is required for human-level artificial intelligence. A key component of our methodology is developing AI systems that behave in complex, dynamic environments with many of the properties of the world we inhabit. Although robotics might seem an obvious choice, research in robotics requires solving many difficult problems related to low-level sensing and acting in the real world far removed from the cognitive aspects of intelligence. Simulated virtual environments make it possible to bypass many of these problems, while preserving the need for intelligent real-time decision-making and interaction. Unfortunately, development of realistic virtual environments is an expensive and time-consuming enterprise onto itself and requires expertise in many areas far afield from artificial intelligence. However, computer games provide us with a source of

cheap, reliable, and flexible technology for developing our own virtual environments for research.

Over the last four years, we have been pursuing our research in human-level AI using a variety of computer game engines: Descent 3, Quake II, and Unreal Tournament. Outrage Entertainment, the developer of Descent 3, created an interface for us to test the viability of using a mature AI engine to control a character in the game. Descent 3 is a fun and challenging game that involves 3D control of a spaceship through tunnels and caves. Although it was a useful first step, we abandoned it for Quake II in which the AI system could control more human-like characters. In Quake II, players (including AI bots) attempt to shoot each other and collect “powerups” such as health items, ammunition, and weapons. Quake II has a dynamically linked library (DLL) that allows access to Quake II’s internal data structures and



Figure 1. Example Unreal Tournament character.

controls for the computer-controlled bots. We interface our AI engine (Soar) through the DLL to control a bot that a human plays against [2]. One attractive feature of Quake II is there are editors available for users to create their own game environments.

Our goal in using Quake II was to discover what was necessary to create an AI bot that played the game in much the same way a human plays the game. We designed our bots to use sensory information similar to what is available to a human, use the controls similar to those used by a human, and use some of the tactics that humans use. For sensing, the bots can “see” other players and items that are not obstructed by other entities or features (such as walls) in the environment. However, it is difficult to extract spatial information about the physical environment from the game, such as walls and doors, which in the game’s internal data structures are just sets of polygons. The bot needs this spatial information to avoid moving into walls and

to create internal maps of its environment. To overcome this difficulty, the bot gets range information about the nearest polygons to the front, back, and to both sides. The bot then builds up a map as it explores the level it later uses for moving from room to room, finding the best path to pick up a given powerup, or hiding in corners to surprise the enemy. The bot can also hear noises made by other nearby characters. For movement, the bots can move left, right, forward, and back, as well as turn, using commands that map directly onto the actions humans can make by moving their mouse and pressing keys on their keyboard.

The reasoning in our bot is done by programs written in the Soar AI architecture. Programs in Soar consist of sets of rules that support knowledge-rich reactive and goal-driven behavior through the elaboration of the situation, and the proposal, selection, and application of operators. For example, rules can elaborate the internal representation of the current situation, such as detect-

ing the bot is too close to a wall, or a useful weapon is nearby. Proposal rules test the current situation, including elaborations to suggest either primitive or complex operators to perform, such as proposing to pick up a nearby powerup (weapon, health, or ammunition item). Additional rules select among proposed operators, such as preferring to pick up the best powerup if there are multiple powerups nearby. Finally, application rules generate the actions involved in performing the operator such as sending a motor command to move forward, or turn in a certain direction.

We did some informal studies where humans compared the behavior of human players to different configurations of the bots to determine if changes in decision speed, tactics, aggressiveness,

ply the process, our bot creates an internal representation of what it thinks the opponent's internal state is, and then uses its own tactics to predict the opponent's behavior. It continues to predict until it finds a situation in which it can get to one of the opponent's destinations first and set an ambush, or there is so much uncertainty in what the opponent will do that it is not worth projecting its behavior any further. After adding anticipation, playing the bot shifts from being a purely tactical game of trying to get the best weapons and shoot the fastest, to a more strategic and intriguing game, where you are always wondering if the bot has already second guessed you and is hiding in ambush on the other side of the next door.

Although action games such as Quake are one of the most popular game genres, there are inherent limits in the complexity of behaviors required to create compelling bots that are essentially computerized punching bags. Furthermore, these types of games limit the human gaming experience to violent interactions with other humans and bots. Therefore, we are currently working to develop nonviolent plot-driven computer games where we really need complex AI characters. The behavior of these characters cannot be determined by a simple script, but must be driven by the interaction of their body with the environment, their goals, their knowledge of the world

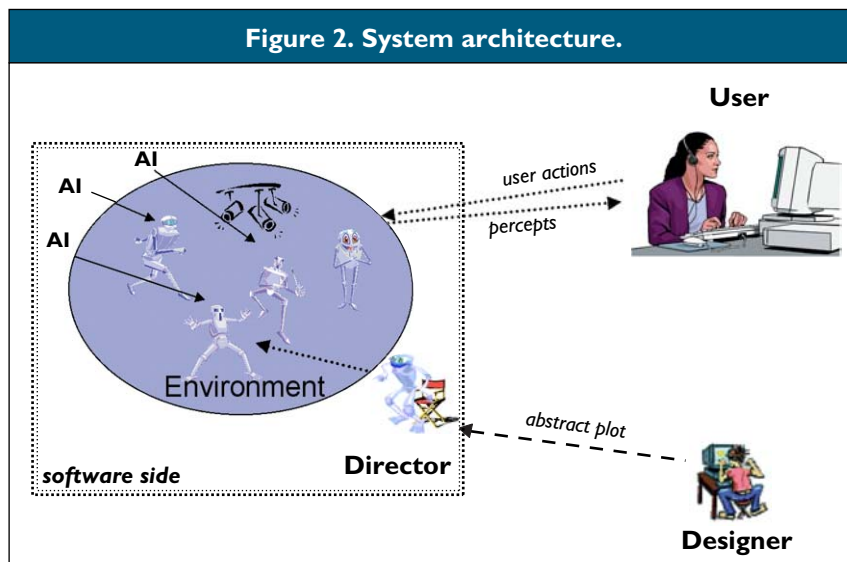
and aiming skill influence the humanness of the bots. The trends in the results were that bots with extremely accurate aiming or extremely fast (<25msec.) decision speed appeared less human than ones with less accurate aiming skills and slower (100msec.) decision speed.

We also did a qualitative analysis of the behavior and noticed that expert players attempt to anticipate the actions of their opponents. Anticipation is a form of planning similar to the look-ahead search performed by AI programs that play classic games such as chess and checkers; however, the challenge in a game like Quake II is that when a decision is made to perform an action (such as when to turn) it is often as important as the choice of action to take. Moreover, in contrast to chess where a player can see the complete board, in Quake II there is only partial information about the state of each player. To sim-

they inhabit, their own personal history, their interactions with human players, and real-time advice from a director. Our hope is that complex AI characters will lead to games where the human players are faced with challenges and obstacles that require meaningful interactions with the AI characters.

We are building on one of the oldest genres of computer games, sometimes called "interactive fiction" or adventure games. These games involve having the human player overcome obstacles and solve puzzles in pursuit of some goal—games such as Adventure, Zork, Bladerunner, and the Monkey Island series. One weakness of these games is the behavior of nonplayer AI characters is scripted; the interactions with them are stilted and not compelling. Our challenge will be to create AI characters whose behavior is not only human-like but also leads to engaging game play.

Figure 2. System architecture.



Using Unreal Tournament (UT), we are creating a game where the player takes on the persona of a ghost-like energy creature trapped in a house (see Figure 1). UT is an action game similar to Quake II with an underlying graphics and game engine that is extremely flexible. For just the cost of the game (\$20), you get access to level editors for defining the environment, a scripting language (UnrealScript) for defining the physics of the world and the way objects in the world interact, and the ability to import your own objects into the game.

In our game, the human player's goal as the ghost is to escape the house and return home to an underground cavern. The ghost is severely limited in its ability to manipulate the environment. It can move or pick up light objects, such as a match or a piece of paper, but it can't move or manipulate heavy objects. Moreover, contact with metal drains the ghost's energy, so the ghost must avoid metal objects. These constraints force the player to entice, cajole, threaten, or frighten the AI characters into manipulating the objects in the world, which in turn forces us to

develop AI characters that have enough intelligence to make these social manipulations possible and realistic. We are also attempting to incorporate a computer director that watches the game as it unfolds and then provides direction to the AI characters in the game—see Figure 2. A similar approach is also being explored by Michael Young's group at North Carolina State University [5].

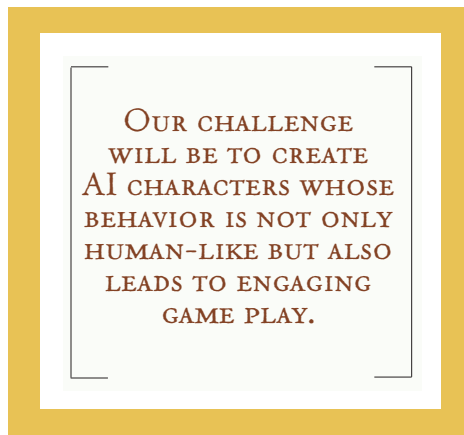
With the AI characters playing such a central role, they must be well grounded in their environment. For example, there is an evil scientist who is immune to fear but is weak and easily fatigued by exertion or cold and wants to capture the ghost character, and there is also a lost hitchhiker (we aren't trying to have the most original story ever) who is easily frightened by the ghost, but is physically strong and driven by curiosity. The game will push our research to integrate the knowledge-based, goal-oriented reasoning we have concentrated on in the past, with emotions, personality, and physical drives that have been used in simple, knowledge-lean agents in other systems [1, 3, 4].

To support the physical drives, we have extended Unreal Tournament so all of our characters have a model of physiological responses to the environment and to their internal processing. For example,

just as these games already have a measure of ambient light level, we have added ambient temperature. Different regions of the game have different ambient temperatures: outside it is very cold; inside it is moderately cold; and when a fire is lit in the fireplace, it is very warm near the fire. All of the physiological properties serve as input into the AI characters, that is, the character is aware of their values. However, the character can only change them indirectly by the actions it performs. For example, the characters have a body temperature that can be raised by exertion, by changing the clothes they wear, or by moving to different regions of the level that have different temperature levels, such as near the fire. Changes in one of these attributes can affect others, so that a significant drop in body temperature can make them more tired.

In conclusion, using computer games provides us with a flexible, robust, and inexpensive environment for exploring the development of human-level AI in complex environments. Our hope is that we inspire others to pursue human-

level AI characters and new types of games that those characters make possible. **C**



## REFERENCES

1. Gratch, J. and Marsella, S. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. In *Proceedings of the 5th International Conference on Autonomous Agents* (Montreal, Canada, June 2001), 278–285.
2. Laird, J.E. Using a computer game to develop advanced AI. *Computer* (July 2001), 70–75.
3. Loyall, A.B. and Bates, J. Personality-rich believable agents that use language. In *Proceedings of the First International Conference on Autonomous Agents*, (Feb. 1997, Marina del Rey, California), 106–113.
4. Macedonia, M. Using technology and innovation to simulate daily life. *Computer* (Apr. 2000), 110–112.
5. Young, R.M. An overview of the Mimesis architecture: Integrating intelligent narrative control into an existing gaming environment. *AAAI 2001 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment* (Mar. 2001), AAAI Technical Report SS-01-02, 77–81.

---

JOHN E. LAIRD (laird@umich.edu) is a professor at the University of Michigan in Ann Arbor.

---

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.