

# AI Characters and Directors for Interactive Computer Games

Brian Magerko, John E. Laird, Mazin Assanie, Alex Kerfoot, Devvan Stokes

University of Michigan

1101 Beal Ave.

Ann Arbor, MI 48109-2110

magerko, laird, mazina, akerfoot, stokesd @umich.edu

## Abstract

We are creating an environment for investigating the role of advanced AI in interactive, story-based computer games. This environment is based on the Unreal Tournament (UT) game engine and the Soar AI engine. Unreal provides a 3D virtual environment, while Soar provides a flexible architecture for developing complex AI characters. This paper describes our progress to date, starting with our game, *Haunt 2*, which is designed so that complex AI characters will be critical to the success (or failure) of the game. It addresses design issues with constructing a plot for an interactive storytelling environment, creating synthetic characters for that environment, and using a story director agent to tell the story with those characters.

## Introduction

To date, every successful computer game is an existence proof that you can create good games without human-level AI. Our research is focused on determining whether or not the addition of human-level AI can facilitate the creation of new types of games: games where the AI characters are central to the player's experience. Our current work builds on previous research on AI in games where the AI characters were enemies for actions games such as *Quake 2* and the characters had many of the same capabilities as human players, including the ability to use many tactics, create internal maps of the level, and anticipate their enemy's behavior (Laird 2001). Although action games such as *Quake* are one of the most popular game genres, there are inherent limits in the complexity of the 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, ignoring many other types of more subtle social interactions. Therefore, we are currently working to develop non-violent plot-driven computer games where we really need complex AI characters and directors.

Ours is but one approach in the growing field of *interactive drama*, which strives to provide a narrative

experience to a human player that has a strong connection between the player's actions and the story (Laurel 1986). Our approach to plot-driven games assumes that there is a single human player that is the focus of the game, as in the *Holodeck* of *Star Trek, The Next Generation*. The player moves through the game, performing actions in the world. The story is defined by challenges and conflicts that arise from interactions with the other (synthetic) characters that populate the world. In order to ensure that the player's experience is a coherent story, we do not want to rely on arbitrary interactions that may or may not arise between the player and the synthetic characters. Instead, we propose relying on a pre-written script that is available to a computer director. It becomes the responsibility of a computer director to ensure that the script is followed during the game experience so that a coherent narrative is experienced by the player. In order to support an immersive and realistic experience, the synthetic characters must be responsive to the interaction of their body with the environment, their goals, their knowledge of the world they inhabit, their own personal history, their interactions with human players, and real-time advice from a director.

To explore the development of computer-based directors and characters, we are building on one of the oldest genres of computer games, sometimes called interactive fiction or adventure games, where the human player must overcome obstacles and solve puzzles in pursuit of some goal – games such as *Adventure*, *Zork*, *Bladerunner*, and the *Monkey Island* series. One weakness of the traditional adventure games is that the behavior of non-player AI characters is scripted; therefore, the interactions with them are stilted and not compelling. One of our challenges is to create AI characters whose behaviors are not only human-like but also lead to engaging game play.

In this paper we describe our initial results in developing human-level AI characters and directors within computer games. Our challenge is to demonstrate that for at least one genre, human-level AI can make a difference so that with human-level AI, the game play is qualitatively different (and still entertaining). This paper covers the following: the overall software architecture, our initial script/game scenario, the design of the physiology and sensing of our characters which in turn forces us to support a combination of goal-driven and environmentally-driven behavior, and the

structure of the story director. We then conclude with a look forward to future research problems.

## Our Story: Haunt 2

The game we are creating in Unreal Tournament has the player taking on the persona of a ghost trapped in a bed & breakfast. The human player's goal as the "ghost" is to determine what happened to it (it was murdered), who murdered it, and then bring the murderer to justice. The ghost is severely limited in its ability to manipulate the environment. It can move throughout the building (through walls even), but it can't move or manipulate objects, nor can it talk to other characters, but it can appear and disappear. 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. Initially we are avoiding issues with natural language understanding and using a small fixed grammar for communication.

To provide more interaction, the ghost is able to "possess" an AI character as long as the AI character isn't too scared. Possession allows the player, as the ghost, to see some of the AI characters' thought processes and influence its decisions when the AI character does not have a strong bias. However, whenever the ghost influences a decision, it feels a bit weird to the AI character, raising the possessed character's level of anxiety. The ghost can maintain possession of relatively calm characters; too much manipulation leads to the ghost being expelled from the AI characters. In order to make the possession fun and engaging, we will have to develop characters whose internal processing appears human-like to a player possessing them

## Requirements for Haunt 2

When considering our system design in the context of building an interactive drama, there are six main requirements we strive to fulfill:

1. **Generalizable design:** Our goal is not to build a specific game, but to explore research issues and then to develop solutions that can be applied more generally. The design needs to be modular so that individual components can be independently developed. Moreover, we are exploring approaches where an author can easily change the current story, add new characters, or create an entirely new plot. The system described in this paper should be applicable to the telling of many narratives; Haunt 2 is merely one of those many.
2. **Flexible, efficient system:** The software architecture should be low-cost, and easy to develop and use.
3. **Believable behavior:** Synthetic characters have to exhibit behavior that is dramatically and realistically believable, whether in performing a simple action such as picking up a drink because they are thirsty, or having

more complicated interactions with other characters, the player, or the world.

4. **Directability:** The synthetic characters should be able to receive story direction from an exterior agent that is responsible for the narrative. Each synthetic character will need to balance the story's goals with its individual personal goals, giving preference to the story. However, even when responding to direction, the characters must generate behavior that is coherent, in character, and thus believable.
5. **Fully-structured story:** Our goal is to create experiences in which complete dramatic stories can be experienced, which is in contrast to systems that depend on a story, or at least the story's structure, to emerge. Initially, we are assuming that the story is created by a human author and that the content in the story is structured according to the author's desires.
6. **Interactive player experience:** Although there is a pre-defined story, the user's action should have significant impact on the details of how the story unfolds. Different player behaviors within the environment should elicit different, however so slightly, narratives. If the player executes some series of actions that are unexpected or go against the plot, the system should mediate the differences to elicit a coherent narrative.

We address how these requirements relate to Haunt 2 in the following sections.

## System Software Design

To support the development, experimentation, and evaluation of our AI characters, we need a game environment that is flexible and efficient (requirement 2):

- ◆ Flexible and low-cost development:
  - We should use existing game software and development tools (level and character editors) whenever possible. The underlying software should be easily modifiable so that we can test out alternative designs.
- ◆ Support a full debugging/development environment:
  - Our design should support all of our existing debugging and development tools for creating AI characters.
- ◆ High performance:
  - We should not sacrifice performance in the AI engine or in its interface to the game. We should support a large number of AI characters in the game without sacrificing graphics performance. We should be able to run the game and the AI engine on a single laptop.

We achieved flexibility and low cost in the game environment by following in the footsteps of other projects (NCSU: Mimesis (Young 2001), ESC Online (Martin 2001), Deus Ex, Gamebots (Kaminka et al. 2002)) and using the Unreal Tournament (UT) engine. UT provides an off-the-shelf 3D game engine that can be easily extended. A copy of Unreal Tournament costs about ~\$20. Moreover there are many free level editors available for creating your own virtual environment; furthermore, the game physics and interface are coded in a powerful internal scripting language (Unrealscript) that is completely accessible.

Each AI character is implemented as rules in Soar (Newell 1990). The characters share the same basic knowledge base to support interacting with the world and other characters; however, specific characters have different physiology (see below), goals and background knowledge.

Designing the interface between the environment and the AI engine is challenging because of two often contradictory goals: high performance and supporting a powerful development environment. Achieving high performance usually demands that the AI engine run as an embedded application in the same process as the game environment, eliminating access to a powerful development environment. In an attempt to finesse these problems, we have developed a high-level interface that is really three low-level interfaces. The high-level interface hides the differences at the low-level so that the user can select from among the three low-level interfaces at run time without requiring any changes to the game engine and the AI engine. The overall interface is called Soar General Input/Output (SGIO) (Laird et al. 2002) and is a domain independent interface between the Soar AI engine and an external environment – in our case Unreal Tournament. SGIO supports across-machine and across-process communication via sockets, allowing us to use our full development environments. For high performance, SGIO supports the direct embedding of multiple copies Soar within the UT process using a C API.

This project builds directly on related work for modeling human behavior in military simulation that shares the same core infrastructure, which consists of UT, SGIO, Soar, and navigation knowledge (Wray and Laird 2002). Although there are many components in common, the other research emphasized encoding military doctrine and tactics with very short but intense interactions among human participants and AI characters. There was no notion of story or director.

## AI Characters as Actors

For the AI characters to be believable (requirement 3), they must have many general cognitive capabilities including perception, internal reasoning, and acting in the world. In *Haunt 2*, each character has a long-term knowledge base, implemented as rules in Soar, perceptual and motor systems, and a simple model of physiology. The current situation is represented declaratively in working memory, which holds what is perceived from sensors (including physiology), current goals and operators, and the character's understanding of the current situation. The long-term knowledge can be broken down into situation-assessment knowledge that reactively classifies sensor data into relevant categories, and task knowledge, which is organized hierarchically as a set of operators. The operators cover both abstract tasks, such as exploring the Inn, and more primitive tasks such as talking to other actors (using a set of templates) or manipulating objects in the world.

## AI Character Physiology

A critical part of making the actors believable is having them respond to their environment. Although one possible way to implement this is for the synthetic actors to “act” as if they have needs and drives; however, our approach is to give them a synthetic physiology so that they have needs and drives. For example, in *Haunt 2* the characters that are cold (and have no other pressing needs or goals) will go into the parlor room, where there is a fire to warm them up. Similarly, a thirsty character may ask the innkeeper for a drink. The game will push our research to integrate the knowledge-based, goal-oriented reasoning that 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 (e.g. the Oz project (Loyall and Bates 1997), Gratch's work at ICT (Gratch and Marsella 2001), and the Sims (Macedonia 2000)).

To support the physical drives, we have extended Unreal Tournament so that all of our characters have a model of physiological responses to the environment and to their internal processing. Moreover, the environment has attributes that influence the physiology of the characters. For example, just as games 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 and inside it is moderately cold, although 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 characters are aware of their values. However, the character can only indirectly change them 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.

Physiological effects that we have implemented include: temperature, exertion, fatigue, sleepiness, hunger, and thirst. There are other attributes that impact a character's actions, such as its strength, speed and dexterity. Commercial computer games have had complex physiological effects for the human player's character as well as many character attributes (an important part of role-playing games). For example, the Sims has a set of attributes for the computer characters which drives their behavior. We are attempting to extend the set to be more comprehensive and to explore the interplay between physiological drives and goal-driven behavior, which other systems in interactive drama lack.

## Environmental Sensing and Action

As in our previous work with Quakebots (Laird 2001), we are committed to giving our characters realistic sensing and actions in their environments. However, this is challenging because of the difficulty of sensing walls (and doors) in these environments. To enable sensing of the rooms, we

annotate the map with regions that give the name of each room so that the characters can directly sense which room they are in. We are creating nodes in the map that are placed at important locations (doors, windows). The characters use these nodes for navigation between rooms, but move more freely within a room based on their sensing of objects and other characters.

### Autonomy and Directability

The agents in *Haunt 2* are first and foremost autonomous; they can move about in the synthetic world on their own, gathering knowledge and attempting to fulfill whatever their current goals are. The characters are at a base-level defined by their goal-oriented behaviors. This way of viewing our agent design in an interactive drama follows Mateas and Stern's view of *strongly autonomous* characters (Mateas and Stern 2000). However, as they point out, using strongly autonomous agents for storytelling can be problematic. Characters performing in a drama may base some of their decisions on their internal state and personal goals, but should give priority to doing the proper actions given the current state of the story. There have been attempts to create interactive dramas that employ a director agent to monitor the story state, which makes use of low-bandwidth communications with the characters to occasionally give them directions to alter their behavior, thus performing story elements (Weyhrauch 1997; Young 2001). There have also been projects focused on *weakly autonomous* agents that have high-bandwidth communications with the director (Mateas and Stern 2002). In this design, the director has more control (and responsibility) to continually monitor all of the agents and provide them with detailed directions. This allows for a finer grain of control over the agents, with the goal being to create a more coherent presentation of the drama, but requires much more communication and forces more complexity into the director.

In order to fulfill both design requirements 3 and 4, we have opted towards a hybrid approach for incorporating synthetic characters in *Haunt 2*'s plot. The characters are *semi-autonomous*, which follows closest to the philosophy laid out by Blumberg and Galyean (1997). In *Haunt 2*, we imbue the characters with autonomous behaviors that the director can manage using high-level directions for major plot points of the story, with the characters executing the directions on their own. However, the director also has the ability to drill down and give low-level direction when the details really matter. This approach is possible because long-term knowledge in *Haunt 2* characters is defined by a hierarchy of operators, which includes a range of general operators to very specific ones that execute atomic actions. The director can give very general commands (e.g. "be social") that correspond to abstract operators, or very specific ones ("move to area X, face the Jon character, and perform dialogue line 126) that execute lower-level agent behaviors. This design allows for flexibility in the story representation not seen in other interactive drama approaches; the story author can write story events with a

*varying degree of specificity*, which we discuss later in the paper.

With this design, a synthetic character's performance can be the result of story direction from the director, the dynamic selection of a new internal goal based on new inputs, a change in the agent's physiology, or most likely some combination of these affects. In the end, the story dictates an agent's performance, which in turn is coordinated by the director. The story author can make use of a character's knowledge or physiology as a tool in telling the story he wishes to tell. The next section discusses the director and its relationship with the story and synthetic characters.

### Story Director

As mentioned above, our approach to interactive drama centers around building semi-autonomous agents that perform actions largely dependent on director commands. Drama managers have been used in varying ways in other interactive narrative projects as well (Weyhrauch 1997; Young 2002; Mateas and Stern 2002; Blumberg and Galyean 1995). There are also alternative systems that rely mainly on autonomous behaviors or predefined dramatic principles to elicit an interesting story (Sgorous 1999; Cavazza, Charles and Mead 2002). The story director in *Haunt 2* uses player prediction to determine if the player's actions will endanger the plot. It is this capability that distinguishes it most from other interactive narrative systems, such the MIMESIS architecture developed by the Liquid Narrative Group (Young 2001).

MIMESIS uses a fully-structured plot, represented as a partial-order plan, and either incorporates unplanned player actions into the story or avoids them altogether if incorporating them is infeasible. What Mimesis, as well as other interactive narrative approaches, does not address is the preemptive alteration of the story state in subtle ways to avoid problematic actions in the future. Other approaches to this problem have taken a more modular approach to plot construction (Mateas and Stern 2002; Weyhrauch 1997; Sgorous 1999). They rely on heuristically choosing plot elements as the player moves through the space of possible stories. While these approaches provide a greater number of possible story orderings, our design focuses on providing different possible plot content from one experience to the other with the game. This section will discuss the story representation used to encode the plot content, how that representation is used by the director and actors, and how player prediction fits into our design.

The story used in *Haunt 2* is written by a human author and given as an initial input to the director (requirement 4). The representation used is similar to that in a partially-order plan. A story's scene can be broken down into atomic events called *plot points*, which can be partially-ordered to construct a scene. A plot point represents some important, story-relevant change in the world. Each plot-point can be split into a set of preconditions and postconditions. The preconditions describe what has to be in true in the world in

order for this plot point to be relevant. The postconditions describe what actions should take place once these conditions are met. Just as in any real-time dramatic experience, timing is important. We also provide a special precondition, called a *timing constraint*, for plot points that indicate a specific time or range of times that the other preconditions must be met by. We have found this structured approach to be particularly sensible when considering creating an interactive narrative for teaching or training applications, when creating an experience where realism and interactivity are important, but also that specific teaching goals are met (Magerko and Laird 2002).

The *Haunt 2 Bed & Breakfast* story begins with the player awakening from a brutal murder at the bed & breakfast as a ghost. He knows little about where he is, or even who the murderer is, so he begins by exploring the house, invisible and immaterial. His first encounter with the characters in the building is overhearing a conversation between Sally and the Innkeeper in the main lobby. Figure 1 illustrates how this event would be represented in the story structure. It describes that both Sally and the Innkeeper should be in the same room and that the player is within earshot. Another more modular way this plot point could be written would be to tell the characters to “engage in small talk,” and let them pick out of a set of pre-written dialogues. One of the director’s roles is to step through the plot and give characters their “cues” for different performances (e.g. dialogue, a new goal, or information). Therefore, the director will notice when the conditions are met for this plot point and move the plot forward by directing the relevant characters to perform specific lines of dialogue.

If the player behaves precisely as the author expected when constructing the plot, then the story will be told without any problems. One of the main issues that arises when working with a fully-structured plot is that the player, and plausibly the synthetic characters, can execute actions in the world that threaten a future plot point’s preconditions from being met. For example, if the player explores the house for longer than two minutes, then the preconditions in the plot point in Figure 1 would be violated. This problem introduces one of the director’s main roles: giving direction to either the world or to the synthetic characters to change their state or their behavior to encourage the satisfying of some precondition(s). In our current representation, the director’s commands are not directly connected to the individual preconditions, but are

|   |  |
|---|--|
| <pre>Location( Sally, x ) Location( Innkeeper, x ) Proximity(Player, Sally, 2) Timing( &lt;= 2 min. )</pre> | <pre>Talk(Sally, Innkeeper, small_talk) Talk(Innkeeper, Sally, small_talk)</pre> |
| <b>preconditions</b>  | <b>postconditions</b>  |

Figure 1: An example plot point in *Haunt 2*.

categorized by the type of situation for which they are applicable. For instance, if a *proximity* precondition, as in Figure 1, is unfulfilled when the plot point’s timing constraint is violated, then a director action that addresses the positions of characters relative to the player may be chosen (e.g. a director command labeled “proximity to player” which would order Sally and the Innkeeper to move closer to the player before engaging in small talk). Some director commands may be general across narratives or types of stories, but it is up to the author of the story to write any needed director commands and have them fit in with the overall narrative. Future work may focus on what sets of strategies for different situations would be appropriate, and how to choose between them.

Requirement 6 reflects the common goal in interactive drama systems to create a narrative that hinges on the player’s actions; different player actions should yield different narratives. In terms of story representation, we try to meet this goal within the constraints of having a fully-structured plot along two axes: *content variability* and *temporal variability* (Magerko 2003). Content variability refers to “what happens in a story” being different across experiences. Temporal variability refers to when those things happen.

In our representation, plot points are partially-ordered by the author. The preconditions for a given plot point can only be considered after all of the preconditions of the points that temporally proceed that point have been met and their postconditions performed. This allows for some temporal variability between experiences, but is constrained by our earlier-stated requirement for having a fully-structured plot. There have been less structured approaches to representation that are more flexible in providing temporal variability (Mateas and Stern 2002; Weyhrauch 1997; Sgorous 1999).

Content variability is an issue brought up far less in interactive drama systems, in part due to the fact that if a system has a high degree of temporal variability, the player is likely to see different plot content between experiences. Systems such as MIMESIS or Façade provide mechanisms for choosing between different pre-written plot points at run-time, but the content of those plot points themselves is fixed; “who,” “what,” and “where” go unchanged. In *Haunt 2*, we make a simple, but effective, design choice in allowing the story author to leave plot content uninstantiated when it is given to the director.

Another role of the director is to recognize missing plot content and instantiate it properly. For example, the final plot point of our scene in *Haunt 2* is shown in Figure 2. At this point, the player has discovered his dead body in the library, and has helped another character stumble upon it. Which character the player ends up with in the final scene depends on who the player has decided to coax into the room with him (e.g. by materializing and scaring them out of the lobby and into the library). A variable can be shared across plot points, allowing for the instantiation of a variable early on the plot to affect plot content throughout the rest of the narrative. Therefore, the variable representing

which character should discover the body alongside the player could be shared with earlier plot points, or bound by the director based on the player’s previous experiences (e.g. instantiate the variable to be the character the player has gathered the least information about and then direct that character to find its way to the dead body). This representational feature is a very simple example of a significant strength of our story representation: it creates a strong connection between the player’s interactions with the world and the story that they experience. The author has the flexibility with this representation to be as exact or as vague as he wishes to be with the use of temporal ordering, content variables, and abstract agent operators.

|   |   |
|---|---|
| <pre>Location( Sally, x ) Location( Innkeeper, x )  Proximity(Player, Sally, 2)  Timing( &lt;= 2 min. )</pre> | <pre>Talk(Sally, Innkeeper, small_talk)  Talk(Innkeeper, Sally, small_talk)</pre> |
| preconditions   | postconditions  |

Figure 2: Ending plot point for Haunt 2.

In requirement 5, we commit to a fully-defined plot structure, as opposed to dynamically creating structure (Mateas and Stern 2002; Sgorous 1999; Weyhrauch 1997). This commitment gives our system the capability to make use of a predictive model of player behavior. The director can run an internal simulation of player behavior and hypothesize about whether or not the story can reach completion in a coherent, timely manner. The player model is implemented as an internal rule-based simulation within the director. The long-term memory of the model is written as production rules in a similar fashion as the synthetic characters are defined, like implementing the top-level operators *listen* or *explore*. As the player moves through the world, the director hypothesizes what declarative knowledge he is acquiring through his experiences. It also keeps an up-to-date internal structure describing the state of the world. Periodically, the director runs a simulation of the user model and the world state, observing what new knowledge the player might gather and what actions he might take. The next step is to have the modeling triggered by events, such as the player moving into a new room, acquiring some knowledge, or too much time passing.

If the model is a success, it means that the player is predicted to execute actions that encourage or at the very least don’t hinder the story from progressing from the current story state to the conclusion. Model failure indicates that there is some precondition on a plot point that is being threatened by predicted player action. Model failure is the same as a precondition being violated by some player or character action; the annotated director actions are executed to attempt to ensure that the state of the world is focused back on achieving this particular condition. For example, before trying to coax one of the characters into the

library to discover the dead body, the player has to know a) that the body exists, b) that the room exists, and c) that the body is in that room. The director queries the model, asking “based on the current hypothesis of player knowledge, will the player behave in a manner that will keep the plot from continuing?” In this example, if the player is missing any of the above knowledge, the model is likely to fail; the player model will execute a series of irrelevant actions and violate a timing constraint. The user model that we employ is a simple one, designed to show that user prediction is a useful module in an interactive drama architecture. Future work would involve revising the model using experimental data. We will also investigate choosing different directions for when the model fails as opposed to when an action in the actual environment violates the plot specification.

## Conclusions & Future Work

This research has gone from the initial desire of exploring the development of human-level AI agents, to pursuing that desire within the context of computer games, to embedding AI agents as actors and directors. We have followed this with the creation of an environment for developing the AI agents. We have sufficient implementations of these agents so that the actors can perform actions on their own as well as take actions sent by the director. The director monitors the agents, the player, and the plot, reasons about how they may interact in the future, and sends directions to the agents based on that reasoning and any story elements that are currently applicable. We are at a stage where we have demonstrated the feasibility of this overall approach, and now the real fun begins as we can explore the deeper research issues that arise in developing more capable actors and directors and extending the system to a complete story that can be played by human users.

The primary research problem we are pursuing with the actors is focused on improving their ability to manage the tension between carrying out their own goals and responding to commands from the director. To maintain believability, the actors must not only stay in character, but also appear to have coherent behavior. If the Innkeeper receives a command to offer a drink to the player who is across the room, the Innkeeper should not immediately abandon his current activity, such as pouring a drink for another customer, and then shouting at the player. Instead, the Innkeeper must replan its actions, determining when it can smoothly incorporate the new direction while still maintaining coherent behavior (Assanie 2002).

Our current plans for the director include having it not only model the player’s actions and respond to problems when they arise, but also to anticipate when the player is unable to move forward in the story. When predicting player behavior, the director asks “Is the player likely to move the plot along?” If the answer is “no” to this question, the director currently stops there and acts on that result. A more intelligent approach to the design would be to have a second, complete search of the possible story futures and ask “Is it even *possible* for the story to be completed?”

Some action may have been executed, either by the player or a synthetic character, which prevents the story from ever reaching a conclusion. Knowing whether or not the story's end can actually be reached could be a key piece of information for the director in deciding what direction to give the characters. If there is a relatively short path to the next plot point, the director's decision may be different than if there were only a single, long path or none at all.

Our future work in *Haunt 2* will not only focus on evaluating the system once it is in a more mature state, but also on developing methodologies for such an evaluation. It is hard to categorize how much more entertaining or "fun" a system is. We intend to instead rely on fulfilling the reasonable design requirements we have laid out for our view of an interactive narrative, as well as conducting experiments on the effectiveness of story direction. The director's effectiveness can be evaluated in terms of how often temporal constraints are violated by players (i.e. how often plot flow lags because of player behavior) and how often the player executes actions that threaten preconditions of future plot points. If the prediction model is effective, these measures will be lower than the same system with player prediction removed from the director's capabilities.

We plan on extending our story beyond a single scene to a complete interactive drama (on the order of 20-30 minutes of game play). The player should be able to replay the experience several times with noticeable differences in the narrative each time due to different plot instantiations, character behavior choices, and player behavior. That will stress all components of the system and let us evaluate whether or not this approach to incorporating human-level AI into computer games can be successful.

## References

- Assanie, M. 2002. *Directable Synthetic Characters*, AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Blumberg, B. and Galyean, T. 1995. *Multi-level Direction of Autonomous Characters for Real-Time Virtual Environments*. In Proceedings of SIGGraph 1995.
- Blumberg, B. and Galyean T. 1997. *Multi-level Control for Animated Autonomous Agents: Do the Right Thing...Oh, Not that*. In *Creating Personalities for Synthetic Actors*, Ed. R. Trappl and P. Petta, Springer.
- Cavazza, M., Charles, F., and Mead, S. 2002. *Sex, Lies, and Video Games: an Interactive Storytelling Prototype*. AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Gratch, J. and Marsella, S. 2001. *Tears and Fears: Modeling emotions and emotional behaviors in synthetic agents*. In *Proceedings of the 5th International Conference on Autonomous Agents*, Montreal, Canada, 278-285.
- Kaminka, G. A., Schaffer, S.; Sollitto, C., Adobbati, R., Marshal, Andrew N., Scholer, Andrew, S., and Tejada, S. 2002. *GameBots: the ever-challenging multi-agent research test-bed*. In *Communications of the ACM*. Vol. 45, No. 1.
- Laird, J. E. 2001. *It knows what you're going to do: Adding anticipation to a Quakebot*. Proceedings of Agents 2001, 385-292, Montreal, CA.
- Laird, J. E., Assanie, M., Bachelor, B., Benninghoff, N., Enam, S., Jones, B., Kerfoot, A., Lauver, C., Magerko, B., Sheiman, J., Stokes, D., and Wallace, S. 2002. *A Test Bed for Developing Intelligent Synthetic Characters*. AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Laird, J. E., A. Newell, Rosenbloom, P.S. 1987. *Soar: An architecture for general intelligence*. *Artificial Intelligence* 33(3), 1-64.
- Laurel, B. 1986. *Toward the Design of a Computer-Based Interactive Fantasy System*. Ph.D. diss., Drama Department, Ohio State University.
- Loyall, A. B. and Bates, J. 1997. *Personality-Rich Believable Agents That Use Language*. Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, California, 106-113.
- Macedonia, M. 2000. *Using Technology and Innovation to Simulate Daily Life*, Computer, 110-112.
- Magerko, B. and Laird, J.E. 2002. *Towards Building an Interactive Training Simulator*, In the Proceedings of the Eleventh Conference on Computer-Generated Forces and Behavior Representation, Orlando, FL.
- Magerko, B. 2003. *Building an Interactive Drama Architecture*. 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment, Darmstadt, DE.
- Martin, M. 2001. *EscOnline: a Venue for Believable Agents*. AAAI 2001 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Mateas, M. and A. Stern. 2002. *A Behavior Language for Story-Based Believable Agents*. AAAI 2002 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment.
- Newell, A. 1990. *Unified Theories of Cognition*. Cambridge, Massachusetts, Harvard University Press.
- Sgorous, N. M. 1999. *Dynamic Generation, Management, and Resolution of Interactive Plots*. *Artificial Intelligence*, 107, 29-62.
- Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. diss., Dept. of Computer Science, Carnegie Mellon.
- Wray, R. E., Laird, J. E., Nuxoll, A., and Jones, R. M. 2002. *Intelligent Opponents for Virtual Reality Training*. Inter-service/Industry Training, Simulation, and Education Conference (I/ITSEC), Orlando, FL.
- Young, R. M. 2001. *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, March 2001: AAAI Technical Report SS 01-02, 77-81.