Beyond Adversarial: The Case for Game AI as Storytelling

David L. Roberts, Mark O. Riedl, Charles L. Isbell
School of Interactive Computing, Georgia Institute of Technology
Atlanta, GA USA
{robertsd, riedl, isbell}@cc.gatech.edu

ABSTRACT
As a field, artificial intelligence (AI) has been applied to games for more than 50 years, beginning with traditional two-player adversarial games like tic-tac-toe and chess and extending to modern strategy games, first-person shooters, and social simulations. AI practitionershave become adept at designing algorithms that enable computers to play games at or beyond human levels in many cases.

In this paper, we argue that the traditional goal of AI in games—to win the game—is not the only, nor the most interesting goal. An alternative goal for game AI is to make the human player's play experience "better."AI systems in games should reason about how to deliver the best possible experience within the context of the game. The key insight of this paper is that approaching AI reasoning for games as storytelling reasoning makes this goal much more attainable.

We present an overview of traditional game AI techniques as well as a few more recentAI storytelling techniques. We also provide afoundation for describing and reasoning about games as stories, citing a number of examples. We conclude by discussingthe implications forfuture directions.

Author Keywords
Artificial intelligence, machine learning, story telling, narrative, drama management

INTRODUCTION
There has been a significant increase over the past several years of interest in research focused on the applications of artificial intelligence to computer games. Dating back to the 1950s with early efforts in computer chess, approaches to game artificial intelligence (AI) have been designed around *adversarial*, or *zero-sum* games. The goal of AI agents in these cases is to maximize their payoff. Simply put, they are designed to win the game. In recent years similar approaches have been applied to newer games of real-time strategy, first person shooters, and more. Despite the relative complexities of these environments compared to chess, the fundamental goals of the AI agents remain the same: to win the game.

Central to the vast majority of techniques in AI and Machine Learning (ML) is the notion of *optimality,* implying that the best performing techniques seek to find the solution to a problem that will result in the highest (or lowest) possible evaluation of somemathematical function. In adversarial games, this function typically evaluates to symmetric values such as +1 when the game is won and -1 when the game is lost.

In this paper, we argue that the focus of game AI design should not be on developing new algorithms to win more often, but developing new algorithms that will make the human player enjoy the game more. Winning or losing the game is an outcome or an end. While theremay be a long sequence of actions that actually determine who wins or loses the game, for all intents and purposes, it is a single event that is evaluated and "maximized." While the outcome is important, it isn't the only aspect of a game that a player evaluates. How they reach the ending can often be just as, if not more, important than what the ending is. A hard fought battle that results in a loss can be more enjoyable than an easy win. We argue that for game AI techniques to deliver more enjoyable experiences to players, these AI techniques should operate to maximize the interestingness of sequences of states; that is, they should focus on telling the player a good story.

In AI and ML, problems are modeled as state spaces, where every point in this abstract space is a particular configuration of the game environment.A sequence of states forms a ***trajectory*** through the space. How do we evaluate the optimality of a trajectory?One way is to identify certain patterns of state transitions that, when considered together correlate to increased player enjoyment.Because a narrative is therecounting of a non-random sequence of events[31],a trajectory through a state space is a narrative. Making the connectionbetween a trajectory of states in a game and a narrative provides us with a set of tools for thinking about and evaluating sequences.

There has been 30 years of research on AI and ML techniques for story representation, generation, understanding, and interactive storytelling. Recent developments in computational storytelling technologies provide a powerful framework for AI systems to represent and reason about computer games in general. We argue that formalizing computer games in a manner similar to a story provides the

Beyond Adversarial: The Case for Game AI as Storytelling. Proceedings of DiGRA 2009

© 2009Authors & Digital Games Research Association (DiGRA). Personal and educational classroom use of this paper is allowed, commercial use requires specific permission from the author.

groundwork for developing “better” game AI that moves beyond the traditional notions of optimal.

In this paper, we first present an overview of traditional AI techniques applied to games as motivation for our later claims. Then, we address several issues.First, we describe how games—both classic board games and modern graphical computer games—can be represented as state spaces and trajectories.We further describe how previous and emerging work in game AI and drama management can be applied to our representation. Additionally, we present our case for conceptualizing state space trajectories as narratives. We conclude with an enumeration of open research questions entailed by our approach that, if met, may dramatically improve the quality of game AI in the future.

AI OVERVIEW

Much of modern artificial intelligence is dominated by the notion of rationality. That is, an agent should always act in its best interest. The notion of *bounded rationality* that has dominated design prescribes that agents will act in their own best interest subject to the constraints imposed by their processing capabilities[41]. The simplest AI is the reflex agent, which makes decisions through the application of “perception-action” mappings.Perception-action mappings are hand-authored IF-THEN rules that dictate what the agent does in any given situation.

More sophisticatedAI techniques typically operate over *state spaces*. A state space is a (often concise) representation of the configuration of the environment. In cases where the perceptual capabilities of agents are limited, such as in robotics, it is difficult to represent and update state information accurately. AI techniques used in these cases are typically of the reactive control variety where all state information is encoded solely by the actual environment and not in a model maintained by the agent itself[8, 24]. In game settings, however, the agents have the advantage of getting accurate and timely state information from the game environment without sensor error. As a result, game AI techniques can easily reason about and maintain state.

In AI, a *problem* is a goal and a set of means for achieving that goal. S*earch* is the process of exploring the possible ways in which these means can be applied to realize the goal[39]. Search was the first tool leveraged by AI researchers to create computer agents that were more sophisticated than simple IF-THEN rules. Despite its early inception, search through state space remains a powerful tool for game AI designers today.

State, Trajectories, and Search

Consider thesimple grid-world environment in Figure 1. Each of the squares is a discrete location and solid lines represent walls that prevent movement. In this environment, we can use *x* and *y* coordinates to efficiently represent state. One state is designated as a start state *S*and one as a goal state *G*.

Knowing something about the structure of thisenvironment, it is relatively straightforward to write down a concise set of rules for how the agent should act: move in the direction that makes the agent’s state (*x*, *y*) closer to *G*. In general, however, state spaces can be arbitrarily complex and therefore hand-specifying rules for agent behavior can be impossible for a programmer. When an agent is confronted with a number of immediate options of unknown value, it can hypothesize about sequences of actions that will lead to states of known value and then pick the best one. That process is search [39].

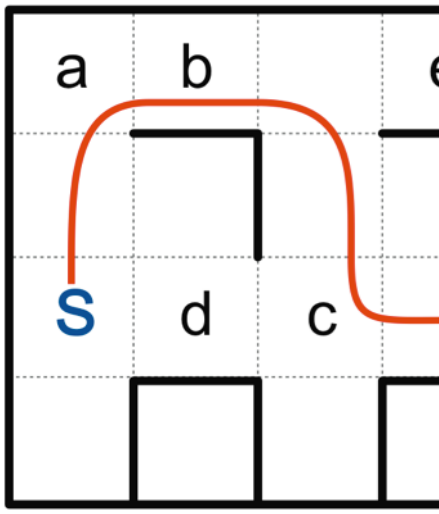

Figure 1: A simp grid-world with barrier intermediate goa (lower case letters), start state *S*, and a go state *G*.

Actions transition the agent from one state to another. A sequence of states and transitions is known as a ***trajectory***. In Figure 1, the red arrow leading from the start state to the goal state depicts one possible trajectory. In traditional AI techniques, trajectories are an artifact of the goal search process. Some trajectories are *solutions* in that they transition the agent from *S* to *G*. It is often the case that there are many trajectories that are solutions, in which case we might want to know whether one trajectory is better than another. A common generic metric is trajectory length, under the assumption that the shortest trajectory is always the best. A heuristic is a function that can evaluate the “goodness” of a state, often as an estimate of how far that state is away from the goal.

In some cases it is helpful to develop heuristics that encode other criteria for comparing trajectories. In Figure 1, some states are labeled *a*, *b*, …,*f*. Suppose there is value associated with passing through these states. We might favor a trajectory that passes through as many of these states on the way from *S* to *G* while minimizing redundant navigation.

Adversarial Search

Adversarial search is a specialized search technique designed for two-player turn-based games such as chess and checkers. Basic adversarial search is applicable to games where there is no element of chance and the game state is completely observable to the AI system.The most prevalent adversarial search algorithm is *minimax*[39].The idea behind minimax is that an AI player looks ahead to predict the outcome of the different possible moves available to them.The player’s goal is to maximize payoff, while the

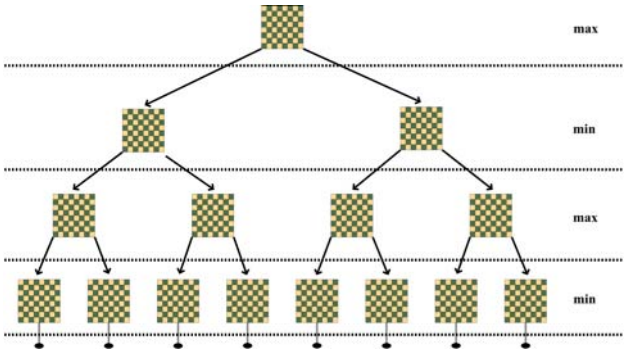

Figure 2: A simplified representation of the search tree for chess.

opponent's goal is to maximize their own payoff which amounts to minimizing the player's payoff (hence, the zero-sum nature of the game). For any given turn, the player looks at all possible moves and the successor states that result from their application.Each successive state similarly has successors based on possible moves by the opponent (unless the end of the game is reached).Search continues until there are no more possible moves or a pre-defined search limit is reached. The agent then picks the move that maximizes payoff according to a heuristic function given the opponent is trying to minimize that same function. The opponent moves, and the search process repeats from scratch.

Figure 2 shows a simplified hypothetical search tree for the game of chess.Each state is represented in the figure by a small board and arrows represent moves by the player or the opponent. In this example, the "branching factor" is two, indicating both players have two legal moves. In actuality, the branching factor in chess is an order of magnitude larger. Each level, or "ply," of the tree represents one move by one of the players. The "root", or top node of the tree, is a maximizing node and every subsequent ply of the tree alternates between minimizing and maximizing representing alternating turns by the two players.

Extensions to minimax allow for games of chance. Another approach is to model uncertainty as *transition dynamics*. Designers can use *Markov Decision Processes* (MDPs) as another means for reasoning under uncertainty[17]. In MDPs, an agent pre-computes a *policy*—a function that maps states to actions. Partially Observable MDPs (POMDPs) are used when the game state is not fully observable [16], that is, when the agent cannot know with certainty what state it is in. Although the algorithmic techniques behind MDPs and POMDPs are different than minimax, the goals are similar: choose an action that will result in the highest expected payoff of the AI system.

AI STORYTELLING

Artificial intelligence is not only used for rational decision making; artificial intelligence has been applied to problems of aesthetics as well. In particular, there are a variety of automated techniques for generating and manipulating stories. There have been a number of systems developed for story generation. Space precludes an in-depth discussion of these techniques, but we can make certain general observations. Narrative generation systems can often be classified as using one of two approaches: simulation or deliberation.*Simulation-based* narrative generation systems (also referred to as *emergent systems*[4]) are those that generate narrative by simulating a story world full of autonomous character agents. Simulation-based systems include [4, 5, 9, 25]. *Deliberative* narrative generation systems search for a sequence of character actions from the perspective of an author. The narrative is the output of this procedure. Deliberative systems include [12, 20, 30, 32].One advantage of deliberative approaches over simulation-based approaches, for the thesis of this paper, is that a deliberative system can reason about the global structure of a narrative. That is, a deliberative system can in theory ask "if I were to have character *c* perform action *a* at time *t*, will this increase or decrease the dramatic structure of the narrative?"

AI AND INTERACTIVE NARRATIVE

Recently, there has been growing interest in stories as artifacts that can augment interactive entertainment products such as video games. This interest in storytelling for interactive entertainment has lead to the development of the notion of an interactive narrative. An interactive narrative is an approach to interactive entertainment in which a system attempts to tell a story to an interactive participant. In order to distinguish interactive narrative systems from other types of interactive entertainment, an interactive narrative allows the user to make decisions that directly affect the direction and/or outcome of the story being told by the system. They are the modern-day equivalent of the Bantam Books Choose-Your-Own-Adventure novels that balanced narrative coherence and user self-agency by separating the reading experience into non-interactive narrative interspersed with decision-points. Mateas surveys early work in the field [23]while Roberts & Isbell [38] and Riedl *et al.* [34]provide more recent surveys.

The goal of an interactive narrative system is to balance the competing requirements for narrative coherence and perceived user self-agency [33]. A *drama manager* (first proposed by Laurel [19] and more generally known as an *experience manager* [34]) is an AI system that attempts to coerce the state of the world such that a coherent narrative unfolds over time without reducing the agency of the interactive participant.

Interactive narratives typically occur in virtual worlds, defined by a model of world dynamics (*e.g.*, what is possible and when it is possible). Like chess and checkers, the vir-

tual world has state. Not all states are of equal value in terms of the player's needs/desires or in terms of narrative quality. Consequently, a drama manager is often explicitly or implicitly provided with a set of author goals; therefore, the AI system is a surrogate for a designer who sets the parameters for defining a good narrative experience. Author goals—usually aesthetic in nature, but can also be pedagogical (*c.f.*, [22, 26, 34])—define a set of qualities or features of the narrative that are preferred or necessary. That is, author goals provide guidance for an AI system to distinguish a good narrative from a bad narrative or at least prefer one possible narrative to another.

A successful drama manager will steer the player away from parts of the state space that, in conjunction with the sequence of past states result in narratives that violate the author's goals. To accomplish this, drama managers must reason about narrative structures based on***state*** and ***history*** information.

BEYOND AVERSARIAL SEARCH

Industrial game developers cite "fun" as the primary design consideration. An AI opponent that acts rationally always has the goal of beating the player. This is true for AI opponents in real-time strategy games, individual non-player characters (NPCs) in first-person shooters, and so on. Realizing that an AI that beats the player is not always fun to play against, designers often take steps to "dumb down" the AI systems by limiting their computational resources or perceptual abilities [21].West proposes that the use of "intelligent mistakes" is a better approach [43].

From an AI perspective, a "mistake" is a sub-optimal decision. We interpret West's intelligent mistakes to be the intentional selection of sub-optimal moves. We extend West's argument by proposing *formal* AI approaches to make calculated "sub-optimal" decisions in the service of improving the player's experience. But how does the system know when and which sub-optimal decisions to make? We propose that many of the computational techniques that have been applied to storytelling can provide the framework for reasoning about when to act sub-optimally (*e.g*, to make "intelligent mistakes") in the context of a game. We argue that under this approach the AI system is actually acting rationally if the goal is to deliver an enjoyable and engaging *global* experience. But this can manifest itself as apparently sub-optimal *local* moves, assuming the human player believes the AI opponent to be rational.

A narrative is an ordered sequence of events that change the story world in meaningful ways. Indeed, the simplest narrative is two states and a transition—or move—between them [18]. Therefore, any sequence of state transitions—trajectories—can be conceived of as a narrative. The question is whether that narrative is "good" from the perspective of the user's experience. Trajectories provide us with a set of tools for reasoning about games and how they relate to enjoyable experiences for players.We hypothesize that an AI opponent that reasons about trajectories as narrative is in fact reasoning about how to make "intelligent mistakes" because it is making choices based not on whether a move increases payoff, but on whether the narrative experience is improved. There are two questions that must be addressed: First, how do we algorithmically search for moves that maximize the utility of trajectories?Second, how do we determine the utility of a trajectory?

Trajectory Space Search

On the surface, searching for a move that maximizes the utility of a trajectory appears complicated.A move transitions the agent from one state to another state.How does the agent know what the trajectory will look like?Part of the trajectory is the history of all moves that preceded the current move and another part of the trajectory is a projection of possible future states.This is further complicated by the presence of the human player, who is not guaranteed to act rationally or predictably.Thus any projection into the future that we perform may not come to pass. One way to deal with these questions is to recast the agent as a system that searches through ***trajectory space***.Trajectory space is like state space, except every point in the space is a partial or complete trace of the game.

Trajectory space search is not a novel idea. One form of search, partial-order planning (POP), is search through plan space, where a plan is a sequence of operators selected to achieve a given goal.That is, a plan is a complete or partial trajectory. Another technique is Targeted Trajectory Distribution Markov Decision Processes (TTD-MDPs), an extension of Markov Decision Processes that reasons about trajectories instead of single states [37].

Narrative Planning

A discussion of how POP algorithms work is beyond the scope of this paper; however many story generation systems utilize some version of planning and thus utilize some form of trajectory space search (*c.f.*, [20, 30, 32]). Others have noted the similarities between plans and narratives [47]. Riedl and colleagues [32, 35] present a specialized deliberative generation system, Fabulist, which reasons about whether a sequence of events will be comprehended as believable. Fabulist has an implicit user model that analyzes proposed solutions based on cognitive psychological principles. This is the first narrative generation system that reasons about narrative from the perspective of the user. However, at the time of writing Fabulist does not have a model of "goodness" that extends beyond the notion of comprehensibility.

Because Fabulist is a narrative generator, it is not designed to handle an interactive user who is making moves in the virtual world. To make planning capable of responding to interactivity, Mimesis [48] and the Automated Story Director [34] utilize a technique called*narrative mediation* [33]. A narrative mediation system determines possible

moves that the user can make in the virtual world. For each user move that is projected to interfere with the progression of a narrative plan, a contingency narrative plan is generated and stored in case real-time execution is required. The result of this process is a tree of contingency plans, where each child is a revision of the parent narrative plan to accommodate a user action that disrupts the parent narrative plan. This narrative mediation tree can be computed prior to execution time and isanalogous to a policy.

Targeted Trajectory Distribution MDPs

Originally developed for drama management, *Targeted Trajectory Distribution Markov Decision Processes* (TTD-MDPs) are specifically designed to provide a framework for reasoning about trajectories [6, 36, 37]. Based on the Declarative Optimization-based Drama Management (DODM) formalism[27, 28, 44], the TTD-MDP formalism reasons about sequences of underlying world states. The solution to a TTD-MDP is a probabilistic policy providing a distribution over action choices in any given world state. TTD-MDPs break from the traditional MDP formalism by redefining the notion of an optimal policy. Rather than solve for a policy that provides anaction at every pointthat will maximize long-term expected discounted reward, the agent solves for a probabilistic policy that matches a target distribution over complete trajectories.

To solve for a TTD-MDP policy, a series of local decisions that involve comparing the current partial trajectory to the desired distribution over complete trajectories is made. In making this comparison the system is able to make globally desirable decisions with only local computation. It may be the case that a local decision appears to be highly suboptimal, but results in access to a part of the trajectory space that has many desirable qualities. In one authorial idiom for TTD-MDPs, examples of desirable trajectories are used to define the goals for the system [36]. If a careful selection of these example trajectories includes those that illustrate the intelligent mistakes an author wants the system to make, then the system can efficiently target those qualities requiring only local computation of decisions. Thus, the TTD-MDP formalism provides a computational framework for easily targeting desirable qualities in trajectories for games.

Evaluating Trajectories

To implement trajectory-space search a system must have the ability to distinguish between trajectories and provide a deterministic (partial) ordering. That is, the system must be able to determine that one trajectory is categorically better than or equal to another trajectory. How do we mathematically determine whether one trajectory is better than another? Another way to ask this question is, what is the heuristic that defines the relative "goodness" of a trajectory? Weyhrauch and others [27, 44] have identified a number of metrics that are believed to correspond to aesthetics of the dramatic arc. A few examples: *Thought flow* measures the coherence of thoughts associated with plot events in a narrative; if a narrative has a bunch of tightly grouped subsequences that evoke similar underlying concepts it is believed to make a better narrative. *Location flow* is similar and is based on physical location. *Plot mixing* is the degree to which multiple sub-plots are explored at the onset of the narrative experience and *plot homing* is the degree to which one coherent sub-plot forms the latter portion of the narrative experience.

We can derive other, theoretical heuristics from narrative and dramatic practices. *Dramatic arc* is one common model of story structure that correlates plot structure to the tension that an audience feels over time[2]. As obstacles between a protagonist and her goals mount, the tension of the audience increases. The climax is the point in which it is clear that the protagonist will overcome adversity and tension decreases. Dramatic arcs can only be identified by looking at the relative tension between successive events (*e.g.*, by looking at sequences); however, it is not computationally straightforward to correlate state transitions to rising or falling tension because tension is an affective state phenomenon felt by an observer andin a game, the protagonist is only the digital representation of the human player.Still, there are models of emotion that lend themselves to computation. In particular, *appraisal theory*[42]asserts that emotion is a specific response to a cognitive assessment of an event. The exact manifestation and intensity of the affect is a result of factors including perceived value of the event and locus of control (*e.g.*, who is responsible). To date, appraisal theory has been used in some research systems (*c.f.*, [1, 3, 13]).

Another concept that comes from the dramatic arts is *suspense,*the feeling of anxiety that occurs when one believes that the likelihood of avoiding a negative outcome is perceived to be small or nonexistent [11]. Suspense is a term used to describe close sporting events; it is a reasonable heuristic for computer games. (See [10] for a system that generates suspense by choosing which events of a narrative to tell). In general, the AI opponent must be able to assess what the player believes will happen and choose the move that increases the likelihood that the game will enter a state of maximal suspense for the player. Given a move that eliminates all possibility of the human player winning, or a move that leaves an opening, the computer player should choose the one that leaves the opening, especially if it is made clear to the player.

Implications

The notion of plot is included in many modern computer games. Generally speaking, however, games that include notions of plot completely decouple the plotline from what non-player characters (NPCs) do. In that sense, the plot is a device to motivate the player. Even if decoupled, the NPC control system can still use trajectory-based reasoning to make each encounter more interesting for the player. Many games with plotlines use various techniques to keep the plotline "on rails" such as limiting the exploration of the player and using triggers to control encounters. These techniques are very effective; it is interesting to note that these techniques can be thought of as creating a hard-coded trajectory with narrative properties.

Our proposal generalizes and formalizes this approach, which we hypothesize will lead to engaging gameplay without hard-coded rails. Reasoning and acting to encourage certain trajectories with narrative properties means the AI system is at times is adversarial—the computer controls an opponent to the player, so there must be an attempt to create the sense of adversity. At other times the AI system will be non-adversarial—the system makes "intelligent mistakes" that are in favor of the human player.

CASE STUDIES

In this section, we will discuss two games not traditionally thought of in terms of narrative: chess and poker. We will highlight how such games can be modeled as narratives and how storytelling and drama management techniques can be applied to create better experiences for players. While our discussion will be limited to more traditional types of games, our approach is not. All genres of games lend themselves to a state-based representation including MMORPGs and FPSes. Such games do present different types of challenges for AI systems including having multiple players acting concurrently; however, there are techniques available to mitigate these complexities to allow for the application of our approach.

Chess

Chess has long been studied by AI researchers for a number of reasons including its rigid and well-defined rules and the enormous space of possible board configurations. In creating a narrative AI for chess, we must design systems that incorporate aesthetic characteristics in its decision making process. Similar to narratives, there are features of a chess game that can make it more or less enjoyable to experience. In the case of narrative, these features correspond to aesthetics of the dramatic arc.Analogues can be applied to chess. Three examples are *forks*, *discovered attacks*, and *pins*. A fork occurs when one player is simultaneously attacking two of their opponent's pieces with only one of their own. A discovered attack is an attack that occurs when one piece moves out of the way of another allowing the second piece to attack the opponent. Apiece is pinned-whenthe playercannot move it without enabling an attack

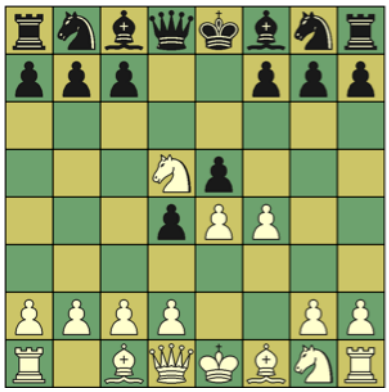

Figure 3: A board position from the Falkbeer Counter-Gambit where black can force white to lose a knight.

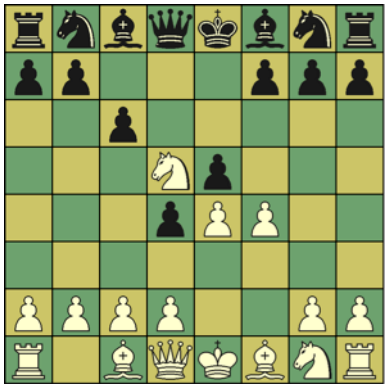

Figure 4: A board position from the Falkbeer Counter-Gambit where black has forced white to drop a knight.

on another—typically more important—piece. While the features of chess bear little resemblance to narrative features, they can form the basis upon which narrative reasoning for chess can occur by correlating them to known narrative heuristics such as location flow and plot homing/mixing or determining how they relate to evaluations of suspense and dramatic arc.

Figures 3and4depict an example taken from a well-studied opening sequence in chess known as the Falkbeer Counter-Gambit [29]. We have elected to use this example as it highlights a subtle concession the computer may make and in doing so make an intelligent mistake. In Figure 5, white has just moved its knight into the center of the board, giving it a reasonably strong position. There is exactly one move that black can make to turn the position into one of immediate weakness for white (shown in Figure 6). By moving its pawn, black has forced white to lose its knight. If the computer were playing white, moving its knight to the cen-

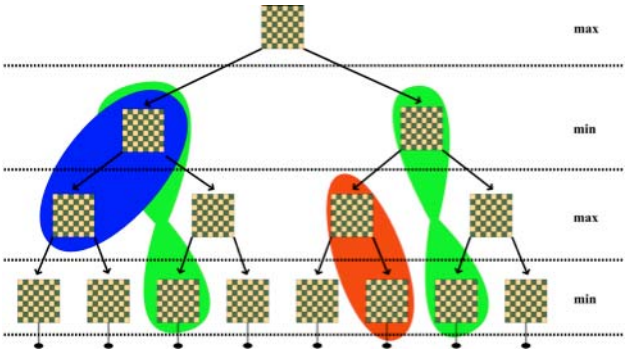

Figure 5: A simplification of the search tree for chess where board positions with desirable aesthetic features have been highlighted.

ter would be an excellent "intelligent mistake" as it is a seemingly strong position for white but provides an opportunity for black to gain an advantage three moves later. On the other hand, if the computer were playing as black, it may choose not to attack with the pawn as it would mercilessly take advantage of white's positional blunder.

Figure 5 is a simplified search tree for chess. A trajectory in this diagram is a path from the root to one of the leaves of the tree. We have highlighted regions of the graph suggesting portions of complete trajectories that have narrative-relevant features. For example the green region on the right might represent a situation similar to that of Figures 3and4.

Designing and implementing a narrative AI for chess amounts to finding trajectories in which certain features desirable from the perspective of the user's experience are present. These features (forks, discovered attacks, and pins) can be used in conjunction with some of the more traditional narrative features like thought flow. Setting up and executing the fork may require an extended sequence of moves all related to the same thought process. Thus, thought flow (as well as other features of narratives) can be made meaningful in the chess setting.

Poker: A Game of Imperfect Information

Despite the relative simplicity of the game rules, poker is still a very difficult game for AI systems due to the hidden state information. In poker, a player knows only what cards it has been dealt and what cards are available to everyone but does not know what cards its opponents have or are left in the deck.Additionally, these systems do not know how their opponents' actions correspond to what cards they may have.

The most obvious use of AI systems for poker is as a player.There has been a range of extremely sophisticated statistical and game theoretic techniques developed for use in a poker-playing computer[7, 14, 15, 49]. These approaches are designed to win as much money from an opponent as possible and state-of-the-art systems can play effectively against highly skilled human players in one-on-one matches.

However, the hidden information in poker provides an affordance for a different type of game AI system—a system that controls the flow of the game by dealing the cards according to an algorithm rather than (pseudo)-randomly as the rules of poker require. In short, everything that is not observable by the human player can be manipulated by the AI system. Both a poker playing and a poker dealing AI system are ripe to benefit from narrative design ideas.

As in chess, there are certain features of a poker game that can contribute to the quality of the experience: bluffing, "slow betting", "bad beats", "catching a card", "flopping the nuts", etc. All of these concepts can be leveraged to build a "fun" poker-playing opponent for the player. Unfortunately, the computer player alone can't control all of the strategies we would like if the card deal is truly random. For example, in order to "flop the nuts" the right card(s) must be revealed to the player at exactly the correct time. The only way to ensure this happens is to maintain control over the deal and to observe the player's hand. The purpose of this is not for the AI system to "cheat" in the classic sense of getting the upper hand over its opponent. Rather, the goal is to be able to make decisions that lead to the emergence of interesting narrative features. The additional necessary caveat is that the AI system must not get caught cheating which can happen in the case of too many coincidences, leading to the player's loss of suspension of disbelief. In that sense, the computer program that deals the cards and manages the money in the pot is a co-creator of the narrative experience for the human player along with the AI and human players.

CONCLUSIONS

All of the algorithmic techniques described in this paper already exist; however, we should note that certain challenges remain. First, the techniques we suggest in this paper will benefit from more accurate estimates of player responses to system actions.For example,work on evaluating player satisfaction [40, 45, 46] may lead to more general models of game content and its effect on player's physiological and cognitive responses. These types of models, once implemented may contribute further to the narrative approach to game AI.

Second, search through trajectory space is computationally more expensive than search through state space. For example, a finite state space can have an infinite trajectory space if the player is allowed to loop through the same states. Under certain circumstances there are average-case efficiency gains when searching trajectory space instead of state space (*e.g.*, partial order planning). We hypothesize that reasoning about the features of trajectories is an advantage when reasoning about player experience; however,

without the existence of robust heuristic evaluation functions that can discriminate the utility of narratives, trajectory space search may not scale. Certain authoring paradigms exist that can additionally help to circumvent this problem [36].

In conclusion, we have argued for an approach to developing game AI systems that target "intelligent mistakes" as a means of increasing the enjoyment of the player. This approach is believed to be more general than approaches that attempt to maximize payoff (*e.g.*, play to win) but with "handicapped" AI or random errors. This approach can be thought of as a process akin to generating a story or creating a "good" story in an interactive narrative setting.

We have described a number of examples of games where narrative AI could be used to create a better game experience. The case studies are not meant to be exhaustive, but suggestive. Modern computer games are often fast-paced and allow concurrent action by all users. However, the key insight is that when a game is represented as a state space (as is commonly done) the paths, or trajectories, through that space can be analyzed in narrative terms. The moves that dictate the path through state space are the building blocks of the narrative and the subset as well as the order in which they occur affects the quality of the narrative (or game experience). This insight allows a game designer to bring the full power of AI storytelling and interactive narrative technologies to bear on creating new and engaging game experiences for players in genres not traditionally thought of as narrative.

REFERENCES

[1] Appling, D. S. and Riedl, M. O. 2009. Representations for Learning to Summarize Plots. Proceedings of the AAAI Spring Symposium on Intelligent Narrative Technologies II.

[2] Aristotle. 1992 The Poetics. Prometheus Books.

[3] Aylet, R. and Paiva, A. 2006. An Affectively-Driven Planner for Synthetic Characters. Proceedings of the 2006 International Conference on Automated Planning and Scheduling (ICAPS 06).

[4] Aylett, R. 1999. Narrative in Virtual Environments—Towards Emergent Narrative. Narrative Intelligence: Papers from the AAAI Fall Symposium (Technical Report FS-99-01).

[5] Aylett, R., Louchart, S., Dias, J., and Paiva, A. 2005. FearNot! an Experiment in Emergent Narrative. Proceedings of the Fifth International Conference on Intelligent Virtual Agents (IVA 05).

[6] Bhat, S., Roberts, D. L., Nelson, M. J., Isbell, C. L., and Mateas, M. 2007. A Globally Optimal Algorithm for TTD-MDPs. Proceedings of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS07).

[7] Bowling, M., Johanson, M., Burch, N., and Szafron, D. 2008. Strategy Evaluation in Extensive Games with Importance Sampling. Proceedings of the 25th Annual International Conference on Machine Learning (ICML-08).

[8] Brooks, R. 1991. How to Build Complete Creatures Rather Than Isolated Cognitive Simulators. Architectures for Intelligence. 225-239.

[9] Cavazza, M., Charles, F., and Mead, S. 2002. Planning Characters' Behaviour in Interactive Storytelling. Journal of Visualization and Computer Animation. 13.

[10] Cheong, Y.G. and Young, R. M. 2008. Narrative Generation for Suspense: Modeling and Evaluation. Proceedings of the International Conference on Interactive Digital Storytelling (ICIDS 08).

[11] Gerrig, R. J. 1993. Experiencing Narrative Worlds: On the Psychological Activities of Reading. Yale University Press.

[12] Gervas, P., Diaz-Agudo, B., Peinado, F., and Hervas, R. 2004. Story Plot Generation based on CBR. Journal of Knowledge-Based Systems. 18(4-5).

[13] Gratch, J. and Marsella, S. 2004. A Domain-independent Framework for Modeling Emotion. Journal of Cognitive Systems Research. 5(4), 269-306.

[14] Johanson, M., Zinkevich, M., and Bowling, M. 2008. Computing Robust Counter-Strategies. Advances in Neural Information Processing Systems 20 (NIPS-08). 721–728.

[15] Johanson, M. and Bowling, M. 2009. Data Biased Robust Counter Strategies. Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics (AISTATS-09).

[16] Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. 1998. Planning and Acting in Partially Observable Stochastic Domains. Artificial Intelligence Journal. 101, 99–134.

[17] Kaelbling, L. P., Littman, M. L., and Moore, A. P. 1996. Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research. 4, 237–285.

[18] Labov, W. 1972. Language in the Inner City: Studies in the Black English Vernacular. University of Pennsylvania Press.

[19] Laurel, B. 1986. Toward the Design of a Computer-Based Interactive Fantasy System. Ph.D.

[20] Lebowitz, M. 1985. Storytelling as Planning and Learning. Poetics. 14.

[21] Liden, L. 2003. Artificial Stupidiy: The Art of Intentional Mistakes. AI Game Programming Wisdom 2. 41-48.

[22] Magerko, B., Wray, R., Holt, L., and Stensrud, B. 2005. Customizing Interactive Training through Individualized Content and Increased Engagement. Proceedings of the 2005 Interservice/Industry Training, Simulation, and Education Conference.

[23] Mateas, M. 1999 An Oz-Centric Review of Interactive Drama and Believable Agents. In M. Woodridge and M. Veloso,Eds. Springer.

[24] McLurkin, J. 2008. Analysis and Implementation of Distributed Algorithms for Multi-Robot Systems. Massachusets Institute of Technology. Ph.D.

[25] Meehan, J. R. 1977. TALE-SPIN, An Interactive Program that Writes Stories. Proceedings of the Fifth International Joint Conference on Artificial Intelligence (IJCAI-77). 91–98.

[26] Mott, B., McQuiggan, S., Lee, S., Lee, S., and Lester, J. 2006. Narrative-Centered Environments for Guided Exploratory Learning. Proceedings of the AAMAS 2006 Workshop on Agent-Based Systems for Human Learning. 22-28.

[27] Nelson, M. J. and Mateas, M. 2005. Search-based Drama Management in the Interactive Fiction Anchorhead. Proceedings of the First Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE-05).

[28] Nelson, M. J., Mateas, M., Roberts, D. L., and Isbell, C. L. 2006. Declarative Optimization-Based Drama Management in the Interactive Fiction Anchorhead. IEEE Computer Graphics and Applications (Special Issue on Interactive Narrative). 26(3), 30–39.

[29] Pandolfini, B. 1989 Chess Openings: Traps and Zaps. Simon and Schuster.

[30] Pérez y Pérez, R. and Sharples, M. 2001. Mexica: A Computer Model of a Cognitive Account of Creative Writing. Journal of Experimental and Theoretical Artificial Intelligence. 13, 119-139.

[31] Prince, G. 1987. A Dictionary of Narratology. University of Nebraska Press.

[32] Riedl, M. O., Saretto, C. J., and Young, R. M. 2003. Managing Interaction Between Users and Agents in a Multi-agent Storytelling Environment. Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-03). 741--748.

[33] Riedl, M. O. 2004. Narrative Generation: Balancing Plot and Character. Computer Science Department, North Carolina State University. Ph.D.

[34] Riedl, M. O. and Young, R. M. 2004. An Intent-Driven Planner for Multi-Agent Story Generation. Proceedings of the 3rd International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-04).

[35] Riedl, M. O., Stern, A., Dini, D., and Alderman, J. 2008. Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training. International Transactions on Systems Science and Applications(Special Issue on Agent Based Systems for Human Learning).4(2).

[36] Roberts, D. L., Nelson, M. J., Isbell, C. L., Mateas, M., and Littman, M. L. 2006. Targeting Specific Distributions of Trajectories in MDPs. Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-06).

[37] Roberts, D. L., Bhat, S., St. Clair, K., and Isbell, C. L. 2007. Authorial Idioms for Target Distributions in TTD-MDPs. Proceedings of the 22nd Conference on Artificial Intelligence (AAAI-07).

[38] Roberts, D. L. and Isbell, C. L. 2008. A Survey and Qualitative Analysis of Recent Advances in Drama Management. International Transactions on Systems Science and Applications(Special Issue on Agent Based Systems for Human Learning).4(2).

[39] Russell, S. J. and Norvig, P. 2003. Artificial Intelligence: A Modern Approach. Pearson Education.

[40] Sharma, M., Ontanon, S., Strong, C., Mehta, M., and Ram, A. 2007. Case Based User Modeling and Search Based Drama Management for Interactive Stories. Proceedings of the 20th Flordia Artificial Intelligence Research Society Conference (FLAIRS-07).

[41] Simon, H. 1957. A Behavioral Model of Rational Choice. Models of Man, Social and Rational: Mathematical Essays on Rational Human Behavior in a Social Setting.

[42] Smith, C. A. and Lazarus, R. S. 1990. Emotion and Adaption. Handbook of Personality: Theory and Research.

[43] West, M. 2008. Intelligent Mistakes: How to Incorporate Stupidity Into Your AI Code. Game Developer Magazine—Digital Edition.

[44] Weyhrauch, P. 1997. Guiding Interactive Drama. School of Computer Science. Ph.D.

[45] Yannakakis, G. N. and Hallam, J. 2008. Entertainment Modeling through Physiology in Physical Play. International Journal of Human-Computer Studies. 66(10), 741–755.

[46] Yannakakis, G. N., Hallam, J., and Lund, H. H. 2008. Entertainment Capture through Heart Rate Activity in Physical Interactive Playgrounds. User Modeling and User-Adapted Interaction(Special Issue on Affective Modeling and Adaptation). 18(1-2), 207–243.

[47] Young, R. M. 1999. Notes on the Use of Plan Structures in the Creation of Interactive Plot. Proceedings of the AAAI Fall Symposium on Narrative Intelligence.

[48] Young, R. M., Riedl, M. O., Branly, M., Jhala, A., Martin, R. J., and Saretto, C. J. 2004. An Architecture for Integrating Plan-Based Behavior Generation with Interactive Game Environments. Journal of Game Development. 1(1), 51-70.

[49] Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. 2008. Regret Minimization in Games with Incomplete Information. Advances in Neural Information Processing Systems 20 (NIPS-08). 1729–1736.