Player Modeling Evaluation for Interactive Fiction

Manu Sharma, Manish Mehta, Santiago Ontañón, and **Ashwin Ram**
Cognitive Computing Lab (CCL)
College of Computing, Georgia Institute of Technology
Atlanta, Georgia, USA
{manus, mehtama1, santi, ashwin}@cc.gatech.edu

Abstract

A growing research community is working towards employing drama management components in story-based games that guide the story towards specific narrative arcs depending on a particular player's playing patterns. Intuitively, player modeling should be a key component for Drama Manager (DM) based approaches to succeed with human players. In this paper, we report a particular implementation of the DM component connected to an interactive story game, Anchorhead, while specifically focusing on the player modeling component. We analyze results from our evaluation study and show that similarity in the trace of DM decisions in previous games can be used to predict interestingness of game events for the current player. Results from our current analysis indicate that the average time spent in performing player actions provides a strong distinction between players with varying degrees of gaming experience, thereby helping the DM to adapt its strategy based on this information.

Introduction

There has been a growing interest in creating story-based games where the player is provided an active role in the ongoing narrative. The underlying goal is to provide the player with a better play experience by gently guiding him towards certain story arcs. These components, called Drama Manager (DM) (Nelson *et al.* 2006) or Director (Magerko *et al.* 2004), employ a set of actions to guide the player towards more enjoyable story-lines. Previous approaches to drama management have either not been connected to a concrete world (Weyhrauch 1997), ignored the player model altogether or employed a hand crafted player model to predict the next player action during the interaction (Nelson *et al.* 2006). Further, these approaches have not been evaluated using real human players interacting with an actual game. In an experiential interactive system, DM should not only use player modeling for actual players but also incorporate results of evaluations to guide its strategies.

Our approach to drama management, previously presented in (Sharma *et al.* 2007), used a player preference modeling component to model interestingness of the intermediate game events and the overall story encountered during the interaction. Our approach is based on the underlying fact that preferences from real players must be used by the DM component to provide a better play experience. In this paper, we specifically focus on results from evaluating the player preference modeling component. In particular, we evaluate two different issues about player modeling. First, we validate the main assumption behind our player modeling approach: if the current player's actions follow a pattern that closely resembles the playing patterns of previous players, then their interestingness rating for stories would also closely match. Next, we present the results from our analysis to find the features that can be used by our player modeling technique to differentiate between different types of players (e.g., based on gaming experience). We also investigate the key features that can be extracted from the player trace to improve the performance of the player preference modeling.

In order to evaluate our overall approach to drama management, we created an intervention where the player is asked to play the game twice, once with the DM included as part of the game and once where there is no DM. We used the player's subjective opinion during both scenarios to evaluate the success of the DM in creating a better play experience. Evaluation with real players has aided us to obtain valuable information that we could not have noticed otherwise. Our analysis of the results presented in this paper are:

- Player modeling is a key factor for the success of the DM based approaches in interactive games.
- During any given game episode, the current DM action trace for the game can be used to predict an interestingness value for upcoming game events (we call them 'plot points') for the current player during the game. Amongst all the key features used to represent the player trace, there is a strong correlation between trace of actions taken by the DM during the game and the corresponding interestingness rating of plot points from the players.
- While finding similarity between two player traces, amongst all the features used for describing the player action trace, average time spent in performing player actions provides a strong distinguishing measure to judge the player's previous gaming experience. Using a player model that incorporates suggestions on previous gaming

experience, the DM can adapt its strategy to improve interestingness for the current player.

The rest of the paper is organized as follows. First, we provide an overview of the existing approaches in player modeling for adaptive games while citing their relationship with our approach in Section 'Player Modeling'. Next, we detail our technical approach in Section 'Drama Management Approach' and follow up with the details of our evaluation procedure and the results in Section 'Evaluation and Analysis'. Finally, we conclude with some future directions we plan to undertake.

Player Modeling

Player modeling is generally accepted as a prerequisite towards achieving adaptiveness in games (Houlette 2004; Charles & Black 2004). Different approaches towards player modeling can be classified in two groups, namely:

- *Direct-measurement* approaches, that employ physiological measures to directly monitor player's emotional state during the game playing episode.
- *Indirect-measurement* approaches, that try to infer (in opposition to directly monitoring player's emotional state) information about the current player (e.g., skill level, and preferences) by computing a set of features from the playing pattern during the interaction in the game episode.

An example of the former, is used in the approach of (Prendinger, Mori, & Ishizuka 2005) where sensed data is used to modify the behavior of an empathic virtual agent situated in a job environment. In our approach towards player modeling, indirect measurements were better suited as the game is currently a text-based interaction, where emotional reactions might not be as rich as in a 3D environment. Further, we were interested in modeling the player from the data that can be derived from the actions taken in the game. Previous work on indirect-measurement techniques for player modeling focuses on modeling the player's skill for automatic adjustment of game level. (Cowley *et al.* 2006) present a decision theoretic framework to model the choices that players make in the well known *pacman* game. They observe the player's deviation from the optimal path and use that to model the player's skill level. One of the results from the evaluation of our DM, presented later, suggested that the skill level of the player is an important measure for a better playing experience. However, our current approach models player preferences and not skill level.

(Togelius, Nardi, & Lucas 2006) present a player modeling technique applied to a racing game. The player models captures the behavior of a player for a given track. Instead of generating hand-made tracks, they use the learned models to automatically generate new tracks that exhibit similar characteristics (speed achieved, difficulty, etc.) when the learned player models are used to drive in the tracks. In their work, they build *player-action* models (i.e., modeling the behavior of the players) whereas we focus on modeling the preferences of the player to provide a better playing experience.

(Yannakakis & Maragoudakis 2005) present results on the usefulness of having a player model. Specifically, they define a set of rules by which the game *pacman* can be considered interesting, and an evolutionary algorithm for the behavior of the enemies in the game that is able to make the game interesting. In that framework, they show the usefulness of a player model to help the evolutionary algorithm to achieve more interesting games (based on the predefined set of rules). In our work, we focus on obtaining the interestingness from the player feedback instead of defining a set of rules. The feedback is in form of an overall game rating, confidence on the rating and a measure of liking for intermediate game events encountered during the interaction.

Finally, there is also a body of work that identifies sets of general heuristic rules that are common for all the players. These rules model whether a game is going to be interesting or not. (Sweetser & Wyeth 2005) present the *GameFlow* model, that combines eight different heuristics (concentration, challenge, skills, control, clear goals, feedback, immersion, and social interaction) to evaluate how much a player is going to enjoy a particular game. Although the heuristics included in GameFlow could be included into our approach, the main assumption of our work is that preferences strongly vary from player to player and thus player modeling is required for an accurate interestingness prediction of story arcs. Further, these heuristics are used as an evaluation metric and not employed directly to adapt the game in order to provide better play experiences for future players.

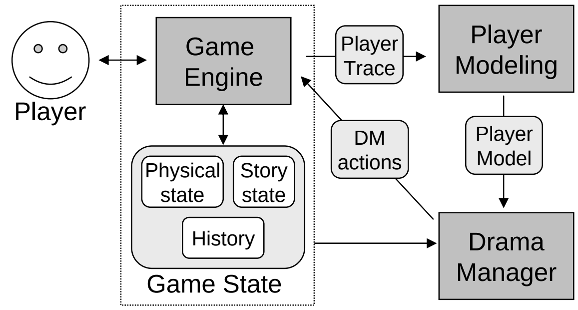

Figure 1: Basic scheme of the three modules that compose a game in our approach.

Drama Management Approach

Our approach to drama management consists of three modules (shown in Figure 1), namely: a *game engine*, responsible for actually running the game and interacting with the player; a *player modeling module*, responsible for analyzing the actions of the current player at run-time and developing a player model; and a *drama management module*, influencing the development of the story arcs in the game and making it more appealing to the player. A comprehensive technical overview of our approach to drama management can be found in (Sharma *et al.* 2007). The next sections briefly explain these three modules.

Game Engine

The game engine is responsible for the traditional duties like running the game, enforcing rules of the game, interacting with the player, maintaining the game state, and presenting game elements to the player (via an audiovisual interface). Specifically, its maintains the *story state*, represented as a collection of events relevant to the game known as ‘plot points’ (Weyhrauch 1997), the *physical state*, representation of the physical objects and the characters in the game, and the *history*, that maintains the evolution of the game to the current situation.

Player Modeling Module

The player modeling module (PMM) builds and constantly maintains a player model for the current player of the game. In particular, it builds a *player preference model* that models the stories that the player is likely to enjoy (see Figure 2). At the end of each game episode, the player is asked to provide an interestingness rating for the overall game experience as well as for intermediate story events encountered during the interaction.

We employ a case-based (Aamodt & Plaza 1994) approach for building the player preference model. Each case in our system, collected at the end of a game, consists of the provided interestingness feedback along with the game trace. During later game episodes, the PMM retrieves the closest matching cases (i.e. those with the most similar playing pattern) and builds a player model for the current player. Using this player model, interestingness of plot points for the current player can be predicted. The predicted interestingness for individual plot points can be computed as a function of annotations contained in the retrieved cases. The underlying assumption behind our approach is that if a player’s playing pattern closely matches with other previous players, then the interestingness rating expressed by all these players would be similar. From each player feedback form, the system can build a case in the following way (see Figure 2):

- Player annotations of interest for each plot point pp_j are converted to a number $\delta(pp_j)$ using the mapping: strongly dislike = -1, dislike = -0.5, indifferent = 0, like = 0.5 and strongly like = 1.
- The overall score provided by the player is also converted to a number $s \in [-1, 1]$ in the same way. Similarly, the confidence provided by the player is converted to a number $c \in [0, 1]$.
- The interestingness of each plot point pp_j is computed as $ppi(pp_j) = \frac{\delta(pp_j)+s}{2}$, i.e. the average between the overall score and the particular annotation for that plot point.
- A new case consists of the player trace, the interestingness values for each plot point $ppi(pp_j)$, and the confidence c.

Drama Management Module

Given the player preference model from the PMM, the current game state, and the author specified story guidelines, the Drama Management Module (DMM) plans story arcs with narrative coherence. Specifically, at every game cycle the DMM uses this information to select, if necessary, a particular *drama manager action* (DM action).

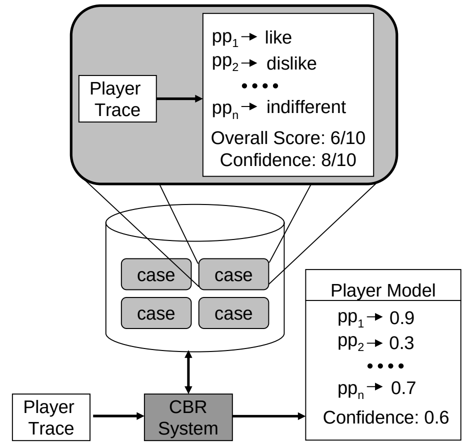

Figure 2: The Player Modeling Module uses a Case Based approach to build the player model.

We classify the DM actions as *causers*, designed to lead the player towards a specific direction in the story, and *deniers*, designed to prevent the player from moving towards a specific direction in the story. A causer can be either a hint or a direct causer and a denier can either have temporary or permanent effects. For instance, one of the DM actions available in our Anchorhead implementation was a hint that induced the player to offer an amulet to a particular character in the game (the bum). Specifically, that hint makes the bum directly ask the player about the amulet.

Given the player preference model from the PMM, the DMM uses an *expectimax* method (Michie 1966) to decide the DM action that leads to the most interesting story arc for the current player. With the current game state as the starting node, the DMM opens a search tree, where the odd plies indicate application of a DM action (including a null action) and the even pies indicate application of a player action. At each leaf node l_j of the tree, the DMM computes the combined interestingness of author-specified guidelines $a(l_j)$ and player preference $p(l_j)$ as $nodei(l_j) = c \times p(l_j) + (1 - c) \times a(l_j)$, where c is the confidence suggested by the player model. These interestingness values are propagated upwards in the search tree, resulting in the DM action that suits the player’s interests the most.

Evaluation and Analysis

We recruited sixteen participants (P1-P16) with a range of genders (4 females and 12 males), races, education levels, and ages (from 22 to 37 with an average age of 23.88). Seven of these participants had absolutely no or low gaming experience. Each participant was provided with an explanation on Anchorhead and asked to sign a consent form before starting the game. The player filled a background questionnaire to provide information such as previous gaming experience or types of games they like to play. The evaluation was conducted in four phases: In Phase I, P1 to P6 played without the DM. From their feedback, we obtained six cases (C1 . . . C6) for the PMM. In Phase II, P7 to P11 played first with-

Player Model Feature	Correlation with Feedback
F1	-0.021495
F2	0.074829
F3	0.258544
F4	-0.077339

Table 1: The correlation of the four different features of the player trace with player feedback.

out the DM and then with the DM (using cases C1 . . . C6). Then, In Phase III, P12 to P16 played first with the DM that used the same cases C1 . . . C6 and then without the DM. The different orders in phases II and III helps in accounting for any possible discrepancy in results due to the order in which they play the game with or without the DM. Finally, in Phase IV, P1 to P6 played the game again with DM that used cases C1 . . . C26 (cases C7 to C26 were collected from phases II and III, one case per completed game trail).

During each episode, a researcher logged his observation of player actions and any unusual reactions. On an average, the complete player interaction (both episodes) lasted for about 45 minutes. At the end of each game episode, the player was asked to provide an interestingness value and their confidence value on a 5 point Likert scale (0 - Strongly Dislike, 1 - Dislike, 2 - Don't Care, 3 - Like, and 4 - Strongly Like) for the overall game experience as well as the intermediate story events that were encountered during the interaction. After playing twice, participants were interviewed about their experience.

Evaluating Player Modeling

There was a significant increase in the averaged overall rating (0 - Strongly Dislike and 4 - Strongly Like) of the game experience from 2.56 when played without the DM (in Phase I) to 3.05 when played with the DM (in Phase IV). This clearly shows an increase in the interestingness of players playing Anchorhead enable with our DM. Furthermore, we observed that the improvement in player rating is dependent on the player modeling component. With respect to the results from Phases II and III, the average overall rating for players playing without DM was 2.92, while the average overall rating of players playing with the DM was 3.03, showing a small increase. The increase in overall game playing experience was found to be dependent on the number of cases in the PMM. Comparing the percentage increase in the overall rating with the number of cases in the PMM, the increase in phases II and III (played with 6 cases) is just a 3.6%, while the increase between phases I and IV (played with 26 cases) is larger, 19.2%. This shows that as the number of cases increase, our drama management approach is able to find story arcs that are better suited to the player's gaming characteristics; underscoring the importance of player modeling to improve the overall player experience.

These results provide a further motivation to perform elaborate evaluation of our player modeling (i.e., the PMM). In the next two sections, we first validate our player modeling assumption of similar preferences for similar playing characteristics. As pointed out earlier, the DM should adapt strategies according to player's previous gaming experience. Hence, the DM should be able to detect the player's experience by observing the player actions and the reactions to the DM strategies.

Detecting Relevant Playing Characteristics

We correlate the player preference ratings for individual plot points and overall experience with different player features used to model them during any game episode. We divided the feature set into four groups, namely: Average number of unique actions performed when the player visits a location in the game for the first time (F1), average time taken by a player to perform an action during the game (F2), DM action trace (F3), and other general player trace features (F4). As a secondary aim we wanted to validate our assumption, that similarity in the player's playing characteristics results in a similar interestingness in the game events.

The motivation behind choice of feature sets F1 and F2 was to get an insight on the gaming experience of the player. We observed that typically, players with low gaming experience spend a lot of time moving in different locations without performing a lot of possible actions at those locations. In our implementation of Anchorhead, F1 incorporates 51 player actions that can be taken at 12 different locations. As a typical episode for a gamer is of shorter duration than a non-gamer, we used feature F2 to capture the information about the gaming experience. Defined only when the game used a DM, the DM action trace (F3) is represented using 7 different features that include the length of trace, number of hints, non-hint causers, permanent deniers, temporary deniers and associated re-enabling actions, and the DM actions that have no effect in the game state. F4 incorporates 9 features of player action trace. These include general trace features like length of the trace, number of unique actions, duplicate actions and cardinality of the most occurring duplicate action as well as the number of actions from each of the five possible player action classes (movement, conversation, picking objects, using objects and no effect on game state).

As we observe from Table 1, most of the features (except F3) did not provide appreciable correlation values. One of the possible reasons for a low correlation value can be understood with an example from our evaluation study. One of the players (P7) enjoyed a particular challenge presented to him during the game and accordingly provided a high rating for the plot points associated with that challenge. When another player (P9), with very similar playing characteristics, interacted with the game, the DM first decided a sequence of actions that lead P9 to the story arc that made him confront the challenge (that P7 earlier enjoyed). In order to guide P9 towards the story arc that contained a challenge, DM provided the player certain hints. During the interview, P9 indicated that as a result of being provided with hints, he did not enjoy the challenge as much (compared to P7). As a result, even though P9 had similar playing characteristics, he provided a lower interestingness rating for the plot points associated with the challenge.

The best correlation was achieved between player feed-

Model	2 Classes		3 Classes		4 Classes	
Feature	**INTRA**	**INTER**	**INTRA**	**INTER**	**INTRA**	**INTER**
F1	0.0784	0.0962	0.0337	0.1153	0.0410	0.1057
F2	0.0640	0.2194	0.0642	0.1682	0.0867	0.1520
F3	0.0899	0.9489	0.0952	0.7921	Not Possible	
F4	0.2448	0.2949	0.2616	0.2879	0.2678	0.2869

Table 2: Spectral clustering on the player models over the four features and increasing number of classes. INTRA and INTER respectively represent the average intraclass distance for all classes and the average distance between the classes.

back and DM action trace features (F3). The high correlation indicates that when two players are presented with similar DM actions during a game episode, they provide a similar overall preference for the game and the plot points. For future DM design, the above result indicates that DM action set employed for a particular player could provide one of the features to measure similarity between different player models. This can be used by the player modeling component to compute the interestingness value of a particular plot point for the player under consideration.

In terms of other features tested, due to low correlation values and different DM strategies employed, our results could not conclusively find a subset of these features (i.e. F1, F2, and F4) that would be useful for detecting the interestingness value of a given plot point for the current player. These results indicate that further research needs to be carried out in DM design to find the correct subset of primitive features from the feature sets F1, F2, and F4 to be used for player modeling. Simple techniques based on decision trees coupled with information gain can be used for discovering the best subset amongst the 17 primitive features (represented as four sets, F1 to F4) and aid in predicting interestingness value for plot points.

Detecting Previous Gaming Experience from Player Traces

This experiment attempts to discover necessary features to distinguish between playing patterns to aid the player preference modeling. We perform spectral clustering on the player traces with the identified feature sets (F1 to F4). This involves measuring the similarity in the playing patterns of the 32 player traces collected from the 16 participants of our evaluation. During clustering, we experimented by increasing the number of possible classifications for a given feature set. This aids our understanding to discover the playing patterns and game characteristics that maintain classification irrespective of the increase in the number of classes.

The results of spectral clustering (see Table 2) lead to certain interesting observations. While clustering the player traces with the average number of unique actions taken when the player visits a particular location for the first time (F1), we observed that most participants with no or low gaming experience (non-gamers) were classified in a single class irrespective of the increase in the number of classes. This result conforms with our observational analysis that non-gamers typically did not perform multiple actions at a given location and move to other parts of the game. Afterwards, once they are hinted by the DM, they revisit the previous locations to perform actions they decided to skip in the previous visit.

The spectral clustering for the feature set denoting the average time taken by the player to perform an action in the game (F2) provided a strong classification of the player traces. Close observation of the each class suggested the influence of factors like gaming experience and the number of times the participant has played the game. These factors were immune to the increase in the size of classification. The clustering lead to the following three such classes:

- Since feature set F2 classifies based on player action time, one class consisted of traces from players who took a long time to complete the game. These included non-gamers who were lost in the game and certain gamers who specifically explored a lot in the game.
- Another class of player traces belonged to those participants that took the least amount of time, on average, to select a player action. Its interesting to note that this class included the second game episode for 10 out of the 16 participants (recall that the protocol involved each participant to play the game twice). This suggests that the average time taken to select a player action is fairly similar (irrespective of the previous gaming experience) when the participants played the game for the second time.
- The last class of player traces showed a common feature of being instances of games played with an active DM.

Feature set of DM action trace, F3, provides a good clustering. This is expected as the spectral clustering is able to identify the cases that contain DM action trace well. Since, there are just two classes (DM action trace is either present or absent in the case), spectral clustering is not able to classify the player models in four or more classes. The final feature set (F4), consisting of the 9 general player action trace features and the ending state of the game, does not provide a very good clustering. Our reasoning for its failure to classify well is that a better analysis is needed to find a representative subset of features that performs a good playing pattern classification.

Our evaluation has shown (from the researcher's observations and the interviewing after the game) that DM strategy should adapt with change in the number of times the interaction has occurred. Further, the players did not enjoy receiving more obvious hints from the DM when they were associated with game elements that they already discovered in the previous interactions. We performed manually clustering of player traces with two different classifications: instance of

Model	Gamer/Non-Gamer		First/Second game	
Feature	**INTRA**	**INTER**	**INTRA**	**INTER**
F1	0.0842	0.0897	0.0877	0.0875
F2	0.1150	0.1097	0.1086	0.1197
F3	0.5760	0.5413	0.4825	0.6211
F4	0.2666	0.2859	0.2773	0.2801

Table 3: Two possible manual clusterings on the player models over the four feature sets (F1 to F4). INTRA and INTER represent the same meaning as in Table 2.

interaction with the game (i.e., first or second) and previous gaming experience (i.e., gamer or non-gamer). Results (see Table 3) indicate that the player model (reconstructed using the four feature sets on the player traces) has not explicitly captured these classifications. This suggests that the problem of detecting whether the player is having an interaction with the game for the first time is not easy to deduce automatically from the player modeling. To find this information, one easy way to achieve this would be to ask the player to enter the information regarding his gaming experience and the interaction with the game at the start of the game (e.g., logging into the game).

Future Work and Conclusions

In this paper we have presented an evaluation of our player modeling approach for a turn-based interactive fiction game called Anchorhead. Our results indicate that there is a strong correlation between trace of actions taken by the DM during the game and the corresponding interestingness rating of the plot points, indicating that DM action trace in the current game episode can be used to predict interestingness value of a plot point for the player during the game. Further our results indicate that amongst all the features used for detecting similarity in the player traces, average time spent in performing actions by the player provides a strong feature to distinguish between gamers and non-gamers, thereby helping the DM to adapt its strategy based on players previous gaming experience.

In terms of other features tested, due to low correlation values and different DM strategies employed, our results could not conclusively find a subset of the features (i.e. F1, F2, F4) that would be useful for detecting the interestingness value of a given plot point for the current player. As a future step we want to use a decision tree based approach, using the information gain (i.e. entropy) to find a feature set that could further help us predict interestingness value for a plot point. Moreover, we want to incorporate the results from these experiments into our DM and player preference modeling components and conduct player evaluations to further validate our findings. Another improvement towards player modeling is by constructing an explicit *player action model* that can predict the next player action and thus aid in pruning the search space of the DM. Thus, achieving both a deeper and faster search.

Acknowledgments

The authors would like to thank the sixteen participants who helped us evaluate our system.

References

Aamodt, A., and Plaza, E. 1994. Case-based reasoning: Foundational issues, methodological variations, and system approaches. *Artificial Intelligence Communications* 7(1):39–59.

Charles, D., and Black, M. 2004. Dynamic player modeling: A framework for player-centered digital games. In *Proc. of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 29–35.

Cowley, B.; Charles, D.; Black, M.; and Hickey, R. 2006. Using decision theory for player analysis in pacman. In *SAB'06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, 41–50.

Houlette, R. 2004. Player modeling for adaptive games. In *AI Game Programming Wisdom II*. Charles River Media. chapter 10.1, 557–566.

Magerko, B.; Laird, J.; Assanie, M.; Kerfoot, A.; and Stokes, D. 2004. AI characters and directors for interactive computer games. In *Proceedings of the 2004 Innovative Applications of Artificial Intelligence Confercence*.

Michie, J. 1966. Game-playing and game-learning automata. In Fox, L., ed., *Advances in Programming and Non-Numerical Computation*, 183–200.

Nelson, M.; Mateas, M.; Roberts, D.; and Isbell, C. 2006. Declarative optimization-based drama management in interactive fiction. *IEEE Computer Graphics and Applications* 26(3):33–41.

Prendinger, H.; Mori, J.; and Ishizuka, M. 2005. Recognizing, modeling, and responding to users affective states. In *User Modeling 2005*, number 3538 in Lecture Notes in Computer Science. Springer-Verlag. 60–69.

Sharma, M.; Ontañón, S.; Strong, C.; Mehta, M.; and Ram, A. 2007. Towards player preference modeling for drama management in interactive stories. In *FLAIRS 2007*. AAAI.

Sweetser, P., and Wyeth, P. 2005. Gameflow: a model for evaluating player enjoyment in games. *Comput. Entertain.* 3(3):3–3.

Togelius, J.; Nardi, R. D.; and Lucas, S. M. 2006. Making racing fun through player modeling and track evolution. In *SAB'06 Workshop on Adaptive Approaches for Optimizing Player Satisfaction in Computer and Physical Games*, 61–70.

Weyhrauch, P. 1997. *Guiding Interactive Drama*. Ph.D. Dissertation, Carnagie Mellon University.

Yannakakis, G. N., and Maragoudakis, M. 2005. Player modeling impact on players entertainment in computer games. In *User Modeling 2005*, number 3538 in Lecture Notes in Computer Science. Springer-Verlag. 74–78.