Using Data Mining to Model Player Experience

Ben G. Weber
UC Santa Cruz
Santa Cruz, CA
bweber@soe.ucsc.edu

Michael Mateas
UC Santa Cruz
Santa Cruz, CA
michaelm@soe.ucsc.edu

Arnav Jhala
UC Santa Cruz
Santa Cruz, CA
jhala@soe.ucsc.edu

ABSTRACT
Using data mining to analyze telemetry enables designers to measure player experience in games, providing quantitative measures. These measures can be used to implement telemetry-supported game design, where a human designer makes decisions with the support of insights learned from mining gameplay data. We model player retention as a measure of player experience and present a technique for identifying which gameplay elements have the most significant impact on this measurement. Our approach builds regression models and applies unique effect analysis to identify these features, which are then used to provide a designer with recommendations. We present two case studies applying this technique: *Madden NFL 11* and *Infinite Mario*, and discuss how our results can be used to provide insights to the game designer.

Author Keywords
Game Telemetry, Data Mining, Player Modeling

ACM Classification Keywords
H.2.8 Database Applications: Data Mining; K.8.0 Personal Computing: Games

General Terms
Measurement, Algorithms, Design

INTRODUCTION
Collecting and analyzing game telemetry has recently become an integral aspect of game development, with applications including bug tracking, gameplay balancing, and level design [6, 15]. There has also been recent interest in the use of telemetry to evaluate player experience in games [14], because it provides analysts with quantitative data. Several sources of telemetry are available including physiological readings, such as Galvanic skin response, as well as gameplay metrics, such as frequency of player deaths [12].

There are several advantages to incorporating telemetry analysis in game user research [1]. First, collecting telemetry enables large-scale analysis of players, because data can be collected from every user interaction. Second, it enables analysis of players in real-world interactions, as opposed to controlled environments. Finally, it enables the analysis of long-term player behavior, where player experience can be evaluated across days, weeks, and months.

We explore the use of telemetry to model player engagement in games. Specifically, we present a data mining approach to model player retention. We introduce several measures of player retention and represent retention as a quantitative metric of player experience. The goal of our approach is to determine how to maximize this metric based on analysis of gameplay telemetry. Our system determines which gameplay features have the most significant impact on player retention, and these features are used to provide game designers with recommendations.

Our approach, Robust Unique Effect Analysis (RUnEA), utilizes data mining to build predictive models of player retention. These player models are used to evaluate the relation between different gameplay features and retention. The intended use of the technique is situations in which there are no strong correlations between individual gameplay features and retention. It analyzes the unique effect of each gameplay feature and identifies which features are most influential in maintaining player retention. The output of the system is used to provide designers with recommendations for increasing player retention.

RELATED WORK
Game telemetry has been applied to several aspects of game development. Zoeller demonstrates that telemetry can be used during game development in order to track bugs, find level design problems, and detect balance issues [15]. A difference from our work is that we are analyzing telemetry post-release, and rely on a franchise title for iteration. Telemetry can also be used to track and visualize metrics, both during development and post release [5]. Our work differs from a metrics focused approach [3], because we are building models of player experience for evaluating the impact of design decisions independent of the game.

Previous work has explored the use of game telemetry to build several types of player models: Drachen et al. analyzed gameplay traces from *Tomb Raider: Underworld* and identified four unique types of players [2], Thurau and Bauckhage mined statistics from *World of Warcraft* and discovered patterns in the evolution of guilds [8], and Weber and Mateas

EPEX 2011, June 28–July 1, 2011, Bordeaux, France.

built classifiers for predicting opponent strategies in *StarCraft* [10]. Our work differs from previous work on player modeling, because we are predicting usage patterns as opposed to in-game behavior [4].

Telemetry is also utilized in game user research. Zammitto et al. are exploring the integration of gameplay telemetry with additional quantitative data sources including eye-tracking and physiological data [14].

PLAYER RETENTION

We represent player retention as a measurement of player experience. Retention refers to how long a player continues to play a game across the player's complete gameplay history, as opposed to individual play sessions. While retention is not a direct measure of user experience, it does provide a quantitative measure of engagement.

There are two main classes of measurements for player retention based on end-user telemetry: aggregate and individual. Aggregate retention refers to measurements taken across the entire player base, such as active monthly users. Individual retention refers to measurements applied to a specific player, such as number of gameplay sessions and total time in game.

Aggregate retention statistics are a useful measurement of player experience in casual games, because the iterative deployment environment enables developers to determine the impact of design decisions. However, aggregate retention statistics are a poor measurement of player experience in console games, because the game generally cannot be modified post release. To usefully apply retention as a measure of player experience in console games, it is necessary to use individual measurements. The goal of our approach is to determine the impact of design decisions without the need for iteration by building models based on individuals.

TELEMETRY-SUPPORTED GAME DESIGN

One of the potential uses of telemetry analysis is to assist in the game design process. Telemetry provides quantitative data sources that can be used to build models of player experience, enabling player-based feedback to designers. However, implementing this approach requires addressing several challenges: dealing with the bias of individual players, providing data with a sufficient resolution, and integrating player-based feedback with qualitative findings.

Specifically, a player experience model is a predictive model for forecasting how design decisions will impact player experience independent of the game runtime. Therefore, an experience model enables direct feedback from players to designers, without the need for rapid iteration. While an experience model is capable of providing useful feedback to designers, it still suffers from the fallacy of being able to identify the *what* of player behavior, but not explain *why* [7].

Nacke at al. define playability as the interaction between a designer and a game, and player experience as the interaction between a player and the game [7]. The interaction between the designer, game and player entities is shown in Figure 1. We propose a new interaction between these entities; an experience model that provides feedback to the designer.

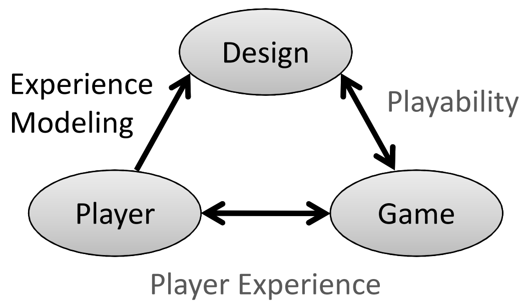

Figure 1. The interactions between player, game, and designer. We supplement the playability and player experience interfaces identified by [7] with a feedback loop from the player to the designer.

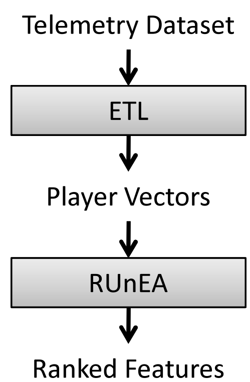

Figure 2. Our data mining workflow includes an extract-transform-load (ETL) component which encodes telemetry into player vectors, and the RUnEA component which selects the most influential features.

ROBUST UNIQUE EFFECT ANALYSIS

Robust Unique Effect Analysis (RUnEA) is a technique for building and analyzing player experience models. It uses an ensemble of regression algorithms to build models for predicting player retention, and then analyzes the impact of each gameplay feature on player retention. The output of RUnEA is a ranked list of features, from highest influence to lowest influence, which are interpreted by an analyst who provides the designer with a set of recommendations. RUnEA was developed as part of a pilot study with Electronic Arts and applied to the task of predicting player behavior in Madden NFL 11 [9].

The RUnEA algorithm is incorporated into a data mining workflow in which telemetry data is collected, transformed, and analyzed. An overview of the workflow is shown in Figure 2. The input into the process is a telemetry dataset collected from a game. The first component (ETL) transforms the telemetry dataset into a set of feature vectors, where each player is encoded as a single vector. The second component (RUnEA) takes the player vectors as input, builds and analyzes an experience model, and then ranks gameplay features from most to least significant.

Player Representation

The output of the ETL component is a collection of feature vectors. Each feature vector is a list of floating-point values that represent the behavior of a single player. The ETL component encodes a player's complete gameplay history into a vector using a game-specific transformation. Each vector also contains a label corresponding to a measure of retention.

Feature Ranking

RUnEA is a game-independent technique for selecting which features have the most significant impact on the retention measurement. It uses regression algorithms to build predictive models of retention and ranks features by computing the unique effect. The unique effect of a feature is computed by varying the value of a single feature, while holding all other features fixed, and measuring the expected change in retention. Features with a large effect cause significant change in expected retention, while features with a small effect minimally impact expected retention.

Our approach uses several regression algorithms which are provided by the Weka toolkit [11]: linear regression, regression trees, and additive regression. In the case of linear regression, the unique effect is equivalent to the regression coefficient of a feature. For the other regression algorithms, a more complex approach is required to measure the unique effect of a feature. RUnEA computes the unique effect by measuring the variance of the expected retention over a range of values.

RUnEA is suitable for building experience models, because it can scale to a large number of players. Using a large number of players to train experience models helps to alleviate player-feedback problems including individual player bias and low-resolution data. The main scalability limitations of the approach are the training times of the regression algorithms and the number of player vectors that can be provided by the ETL component.

CASE STUDY: MADDEN NFL 11

Madden NFL 11 is a commercial American football game released for the Xbox 360 and PlayStation 3 platforms[1]. It is the 22nd installment of the Madden NFL series, which is released during the NFL pre-season each year. Because a new Madden NFL title is produced each year, the designers are able to apply an iterative design process where feedback from the previous year can be incorporated into the next iteration. Previously, the designers relied on feedback from a small set of players, which were not representative of the player base. RUnEA improves this process, because it enables the investigation of players at a much larger scale.

RUnEA was applied to Madden NFL 11 telemetry to determine which gameplay features have the most significant impact on maintaining player retention. It also identifies whether each feature has a negative or positive correlation with retention. The output of the algorithm is a ranked list of features and their correlations, which is used to provide the designers with recommendations.

The Madden NFL 11 telemetry dataset contains gameplay data about millions of players from August 10th, 2010 to November 1st, 2010. It contains session-based data, where each session corresponds to a single game played, and contains a summary of every play in the game. For each play, the summary provides information about the starting conditions of the play, the formations and playcalls executed by each team, a subset of the actions executed during the play, and the outcome of the play. For our analysis we used a single random sampling of 25,000 players.

Player Representation

In order to apply RUnEA to the Madden NFL 11 dataset, it is necessary to develop an encoding for a player's complete gameplay history as a feature vector. Our feature vector representation captures a player's mode preferences, control usage, performance, and playcalling style. In total, there are 46 features which describe a player in our representation. Our approach creates a feature vector for each of the sampled players. The label assigned to each vector is the number of games played, which we use as a metric for measuring player experience.

Mode preference features describe a player's preferences for different game modes. Our representation includes two features for each mode, which capture the usage ratios of the different game modes as well as the player's win ratio for each mode. In Madden NFL 11 there are eight different game modes which include several single player and multiplayer variants. For multiplayer modes, an additional features is included which specifies the ratio of opponent quits and disconnects.

Control usage features encode a player's competency with the controls in Madden NFL 11. It includes two types of features: features describing the player's usage of pre-snap commands, such as audibles for changing the current play, and intra-play commands, such as controlling the path of a specific player during a play.

Performance features describe the ability of the player to make successful plays and gain an advantage over opponents. Features in this category include turnovers (changes in possession), average yards gained, average yards allowed, ratio of possession, and ratios of down conversions. For turnovers, our representation contains two features for both interceptions and fumbles, which correspond to offense and defense roles.

Playcalling features are used to describe a player's playcalling preferences. In Madden NFL 11, players can either manually choose a play to execute or use *Gameflow* which automatically selects a play for the player based on the current game situation. Our representation includes a feature that records the ratio of manual versus *Gameflow* playcalling. There is also a feature that describes the player's ratio of running versus passing plays. In order to capture the variety

[1]Madden NFL 11 was developed by EA Tiburon and published by EA Sports. Trademarks belong to their respective owners.

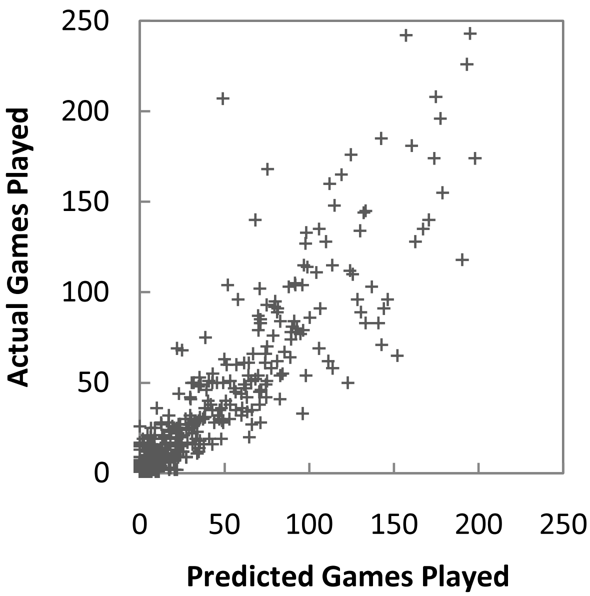

Figure 3. The number of predicted versus actual number of games played for the additive regression model shows a strong correlation.

of plays called by a player, the representation also includes features for play diversity. Play diversity is defined as the ratio of unique plays called to the total number of plays called, and there is a feature for *Offense Play Diversity* and *Defense Play Diversity*.

Results

The first phase of RUnEA builds regression models for predicting the games played given the description of a player's behavior. We used the Weka toolkit [11] to build and analyze several regression models. The most accurate model was additive regression, which achieved a correlation coefficient of 0.9 when using ten-fold cross validation. A visualization of predictions from the model for a subset of the sampled players is shown in Figure 3. Given the accuracy of the models, it is possible to analyze the impact of a feature, by modifying the value of the feature in the player feature vectors.

The second phase of RUnEA analyzes the regression models to determine the impact of each feature. For each feature, RUnEA holds all feature values fixed, except for the feature being evaluated and computes the predicted number of games played over the range $[0, 1]$. A visualization of this process is shown in Figure 4. The impact of each feature is then computed based on the deviation of the predicted number of games from the mean predicted value. RUnEA sorts the features from highest to lowest impact and outputs a ranked list of features.

The features with the highest impacts in Madden NFL 11 are shown in Table 1. Overall, the *Offense Play Diversity* and *Defense Play Diversity* features had the most significant impact on the predicted number of games played. Features corresponding to making successful plays, including *Interceptions Caught* and *Sacks Made*, also had a large impact. Additionally, the *Online Franchise Win Ratio* and *Multiplayer Win Ratio* features were highly influential in maintaining player retention.

Table 1. Features RUnEA identified as having the highest impact on player retention. The direction indicates whether the feature was positively or negatively correlated with the number of games played.

Feature	Correlation Direction	Unique Effect
Offense Play Diversity	(−)	55.4
Defense Play Diversity	(−)	34.2
Interceptions Caught	(+)	24.6
Online Franchise Win Ratio	(+)	15.7
Running Play Ratio	(+)	10.1
Multiplayer Win Ratio	(+)	9.3
Sacks Made	(+)	8.4
Defense Audibles	(+)	6.9
Peer Disconnects	(−)	6.3
In Possession	(+)	5.1

Based on these results, several recommendations were proposed to the *Madden NFL 12* designers. First, playbooks should be simplified, as least for novice players, because play diversity had a large negative impact on retention. Second, the controls should be clearly presented to players, because knowledge of the controls and the ability to make successful plays had a larger impact on retention than winning. Finally, provide players with the correct challenge, because the optimal win ratio is different for each game mode [9].

The Madden NFL 12 designers are now faced with the challenge of integrating traditional player feedback with the new source of feedback provided by RUnEA. To overcome this issue, the designers are exploring how to design the playcalling system to address the concerns of play diversity, while minimally impacting veteran players.

CASE STUDY: INFINITE MARIO

Infinite Mario is a public domain, Java clone of Nintendo's *Super Mario Bros* (1985) developed by Markus Persson. It uses procedural content generation to dynamically generate levels. We explore a variation of Infinite Mario that generates levels using a Probabilistic Multi-Pass (ProMP) algorithm and adapts to the skill of the player using dynamic difficulty adjustment[2]. The system generates levels online and rewards or punishes players based on their success: level completion causes the game to generate more challenging levels, while player deaths cause the game to generate easier levels. An example generated level is shown in Figure 5. RUnEA was applied to Infinite Mario telemetry to determine which features of the level generator have the most significant impact on maintaining player retention.

The Infinite Mario telemetry dataset contains information collected from September 10th, 2010 to February 1st, 2011. During this time 4,427 players participated in 20,433 gameplay sessions. The dataset contains streamed-based data, where game events are immediately logged to the server.

[2] `http://eis-blog.ucsc.edu/2010/09/mario/`

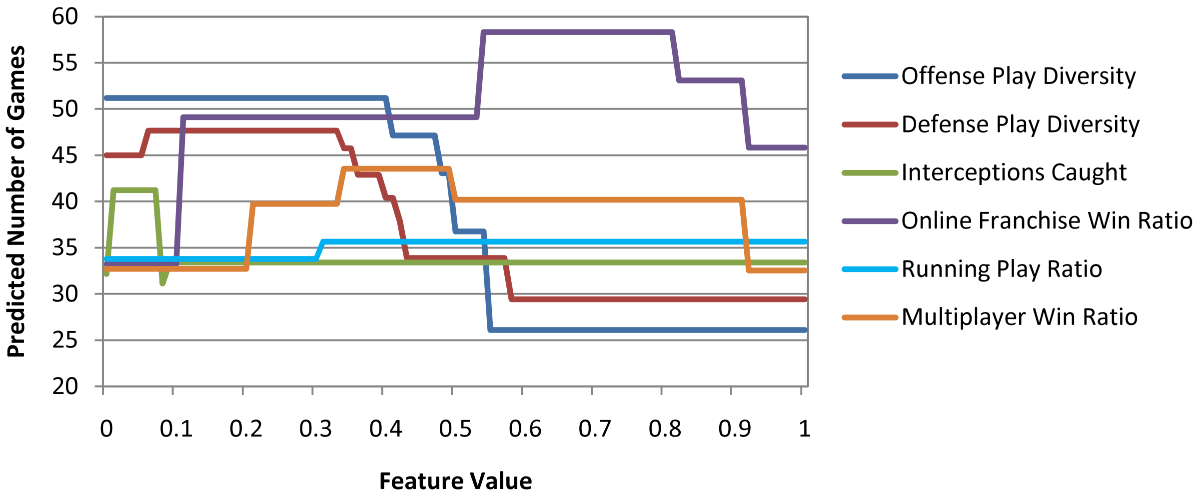

Figure 4. Unique effect analysis of the additive regression model on Madden NFL 11. The figure shows the result of holding all but a single feature fixed, which is evaluated over the range $[0, 1]$. Offense and defense play diversity are highly negatively correlated with retention.

Figure 5. A level generated by ProMP at the most challenging setting contains a large number of enemies and gaps.

The dataset contains data about level generation, level completion, and player deaths.

Player Representation
Our feature vector representation for Infinite Mario encodes both player and level generation features. It contains information about the minimum, average, and maximum levels generated for a player as well as minimum, average, and maximum levels completed by a player. The vector also includes features for capturing causes of death including enemies, pits, and the time limit. The label assigned to each vector is the number of gameplay sessions, which we use as a metric for measuring player experience. A new gameplay session is instantiated each time the player starts the game.

Results
The first phase of RUnEA found additive regression to be the most accurate predictive model with a correlation coefficient of 0.76. The second phase of RUnEA identified the maximum level generated, with a positive correlation, and minimum level generated, with a negative correlation, as the most significant features in maintaining player retention. A visualization of a subset of the features analyzed is shown in Figure 6.

Based on these results, two recommendations can be provided for the level generation process. First, the cause of player death had little effect on player retention and therefore the focus of the generator should be providing the player with the correct challenge. Second, it is necessary to quickly adapt to the skill of the player: the most significant retention patterns were at the ends of the skill spectrums. There are two ways in which these insights can be applied: a human designer, or a system that optimizes player experience [13].

CONCLUSIONS AND FUTURE WORK
We presented RUnEA, an approach for modeling player experience from gameplay telemetry, and showed that building models of player experience enables player-based feedback for designers. The RUnEA algorithm identifies gameplay features with the highest impact on a player experience metric. We represented retention as a player experience metric, and showed how results from RUnEA can be used to improve player retention.

We investigated the application of RUnEA in two case studies: our analysis on Madden NFL 11 resulted in recommendations for the designers of Madden NFL 12, and our analysis on Infinite Mario resulted in recommendations for the level generation process.

There are two main directions for future work. The first direction is to explore the integration of telemetry-supported game design with additional game user research methodologies including metrics visualization [5] and analysis of physiological data [14]. The second research direction is to investigate how approaches such as RUnEA can be used dur-

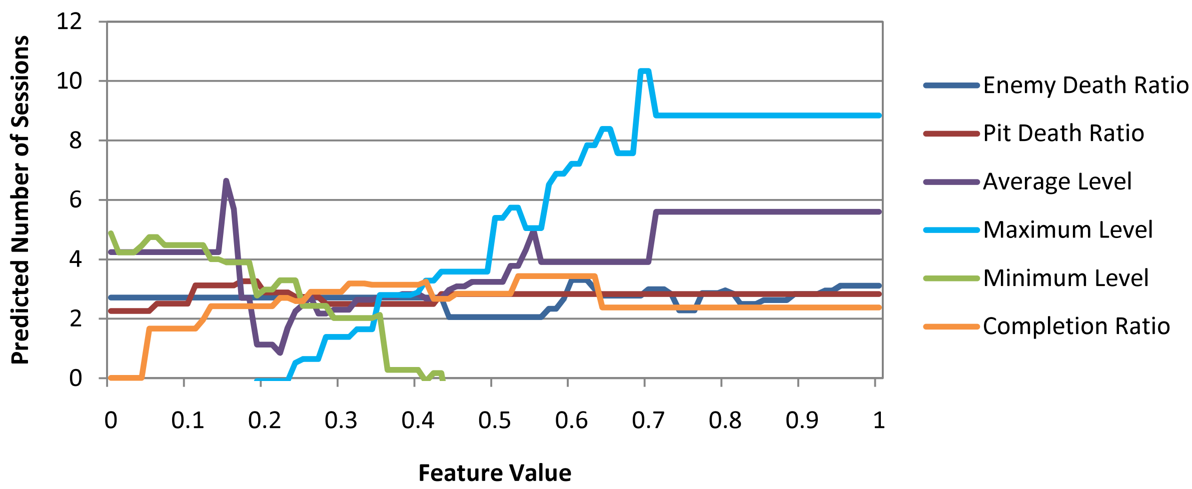

Figure 6. Unique effect analysis of the additive regression model on Infinite Mario. The maximum and minimum levels completed had the most significant impact on player retention, while cause of player death had minimal impact.

ing the development process. Several challenges need to be addressed to achieve this goal: dealing with a small data set, avoiding individual player bias, and reasoning about exogenous player behavior.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant Number IIS-1018954. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

1. Drachen, A., and Canossa, A. Towards Gameplay Analysis via Gameplay Metrics. In *Proceedings of the International MindTrek Conference*, ACM (2009), 202–209.
2. Drachen, A., Canossa, A., and Yannakakis, G. N. Player Modeling using Self-Organization in Tomb Raider: Underworld. In *Proceedings of Computational Intelligence and Games*, IEEE Press (2009), 1–8.
3. Kim, J. H., Gunn, D. V., Schuh, E., Phillips, B., Pagulayan, R. J., and Wixon, D. Tracking Real-Time User Experience (TRUE): A comprehensive instrumentation solution for complex systems. In *Proceeding of SIGCHI*, ACM (2008), 443–452.
4. Mahlmann, T., Drachen, A., Togelius, J., Canossa, A., and Yannakakis, G. N. Predicting Player Behavior in Tomb Raider: Underworld. In *Proceedings of Computational Intelligence and Games*, 178–185.
5. Medler, B., John, M., and Lane, J. Data Cracker: Developing a Visual Game Analytic Tool for Analyzing Online Gameplay. In *Proceedings of CHI* (2011).
6. Mellon, L. Applying Metrics Driven Development to MMO Costs and Risks. *Versant Corporation* (2009).
7. Nacke, L. E., Drachen, A., Kuikkaniemi, K., Niesenhaus, J., Korhonen, H. J., van den Hoogen, W. M., Poels, K., IJsselsteijn, W. A., and de Kort, Y. A. W. Playability and Player Experience Research. *Proceedings of DiGRA* (2009).
8. Thurau, C., and Bauckhage, C. Analyzing the Evolution of Social Groups in World of Warcraft. In *Proceedings of Computational Intelligence and Games*, IEEE Press (2010), 170–177.
9. Weber, B. G., John, M., Mateas, M., and Jhala, A. Modeling Player Retention in Madden NFL 11. In *Proceedings of IAAI* (2011), To appear.
10. Weber, B. G., and Mateas, M. A Data Mining Approach to Strategy Prediction. In *Proceedings of Computational Intelligence and Games*, IEEE Press (2009), 140–147.
11. Witten, I. H., and Frank, E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, San Francisco, California, 2005.
12. Yannakakis, G. N. How to Model and Augment Player Satisfaction: A Review. In *Proceedings of the ICMI Workshop on Child, Computer and Interaction*, ACM (2008).
13. Yannakakis, G. N., and Hallam, J. Real-Time Game Adaptation for Optimizing Player Satisfaction. *IEEE Transactions on Computational Intelligence and AI in Games 1*, 2 (2009), 121–133.
14. Zammitto, V., Seif El-Nasr, M., and Newton, P. Exploring Quantitative Methods for Evaluating Sports Games. In *CHI Workshop on Brain, Body and Bytes: Psychophysiological User Interaction* (2010).
15. Zoeller, G. Game Development Telemetry. In *Proceedings of the Game Developers Conference* (2010).