

When Players Quit (Playing Scrabble)

Brent Harrison and David L. Roberts

North Carolina State University
Raleigh, North Carolina 27606

Abstract

What features contribute to player enjoyment and player retention has been a popular research topic in video games research; however, the question of what causes players to quit a game has received little attention by comparison. In this paper, we examine 5 quantitative features of the game *Scrabble* in order to determine what behaviors are predictors of a player prematurely ending a game session. We identified a feature transformation that notably improves prediction accuracy. We used a naive Bayes model to determine that there are several transformed feature sequences that are accurate predictors of players terminating game sessions before the end of the game. We also identify several trends that exist in these sequences to give a more general idea as to what behaviors are characteristic early indicators of players quitting.

Introduction

As video games continue to grow in popularity, game developers are seeking new ways to encourage players to play their games. As such, there has been increasing attention paid to determining what motivates players to continue playing. The idea behind this research is that knowledge of why players play games can be used to guide the design of experiences that players want to experience. From this research, several models of player engagement (Ryan, Rigby, and Przybylski 2006; Schoenau-Fog 2011) and enjoyment (Sweetser and Wyeth 2005) have been developed. There are significant financial benefits for games companies to keep their players motivated to play. For example, in social games, the more time players spend playing, the more advertisements they see, the more money companies make. With consoles, the longer players play games, the longer it takes for the games to be resold, and the more companies profit from new copies. While knowledge of engagement and enjoyment can help designers to make better games, it doesn't provide insight into another fundamental aspect of game play: the fact that players stop playing at some point. Therefore, the question of why players quit playing is equally important as why they play to begin with.

In this paper, we present a data-driven approach to begin to answer a related and equally important question. Rather

than why, we ask the question "when do players stop playing a game?" If we can identify trends in when players tend to stop playing, we can gain early insight into why they stop. If game developers have the ability to determine when a player is likely to stop playing, they can implement adaptive retention techniques such as scaling AI difficulty to fit player skill (Hagelback and Johansson 2009; Spronck, Sprinkhuizen-Kuyper, and Postma 2004) or adapting the game to increase engagement (Yannakakis and Hallam 2009; Gilleade and Dix 2004). These techniques range from offering the player incentives to fundamentally altering the game so that the player wants to keep playing.

It is important to note that there are two main ways that a player can quit playing a game. The first of these is on the level of game *sessions*. For example, a player that begins a game session will eventually end the game session. This does not mean that this player will not play the game again in the future; it just means that the player has chosen to stop playing the game for the time being. The other way that players quit games is, often, more permanent. This occurs when players end a game session and never begin a new one. The former is a necessary, but not sufficient, condition for the latter. Therefore, we believe it is the more important of the two types of quitting to detect.

We use an implementation of the popular table-top word game Scrabble. Scrabble is representative of games in the social games space, a genre of casual games where game play duration translates directly to profit for games companies. We collect low-level analytics about game play as a means to identify patterns that predict session termination. We illustrate the difficulty of this problem by demonstrating how standard prediction techniques on raw analytics yield very poor accuracy (worse than simply guessing). We then present a feature transformation on the raw data that enables a simple naive Bayes model which yields an accuracy as much as three times higher than the baseline in certain cases.

Related Work

Most research that has been done has been in the area of massively multiplayer online role-playing games (MMORPGs) and has examined the more permanent type of quitting. Targ, Chen, and Huang (2008) studied the play tendencies of 34,524 World of Warcraft players during a two year period to determine if there are any indicators that a



Figure 1: A screenshot of the game *Scrabblesque*

player would quit playing. They determined that while predicting a player’s short term behavior is feasible, predicting long term behavior is a much harder problem. Tarng, Chen, and Huang (2009) performed a follow-up study on a different MMORPG using support vector machines (SVMs) as their predictor. They concluded that it is possible to predict when a player will quit in the near future as long as they do not possess wildly erratic play behavior.

Researchers have also studied the idea of *retention* and the factors that most contribute to it in video games. Retention refers to how long a player plays a game across their entire gameplay history. Weber, Mateas, and Jhala used regression to determine what features most contribute to retention in both Madden ’11 (2011) and infinite mario (2011).

The most closely related work was to identify secondary objectives that lead to people playing for a shorter time. Andersen *et al.* (2011) showed that the presence of secondary objectives can lead to players leaving the game prematurely. The main difference between their work and ours is they examine the presence or absence of objectives as a contributor, and therefore do not consider games as a time-series.

Data Collection and Cleaning

We created *Scrabblesque* based on the popular board game *Scrabble* (see Figure 1). *Scrabblesque* shares interaction characteristics with common social game environments by having a limited action set as well as having a limited number of ways that the user can interact with the game. The main difference between *Scrabblesque* and most other social games is that players do not have a human opponent in *Scrabblesque*; instead, the player competes against a computer controlled AI opponent.

To evaluate our hypothesis, we examined a set of 195 game logs gathered from *Scrabblesque*. We recruited participants using snowball sampling from mailing lists at our university, social networking sites, and popular technology forums on the internet and collected data for several weeks. In total, 94 different users played, which indicates that users played multiple games.

Scrabblesque logs several low-level features of gameplay. The features are:

- **Mouse Position:** The x and y coordinates of the player’s

Table 1: Summary of *Scrabblesque* dataset. Games played shows the number of games played in each category. Game length reports the average game length and standard deviation in game turns. Players shows the number of unique players in each category.

| | Games Played | Game Length | Players |
|------------|--------------|-------------|---------|
| Finished | 148 | 10.1 ± 2.5 | 56 |
| Unfinished | 47 | 4.4 ± 3.3 | 49 |

current mouse position

- **Mouse Clicks and Unclicks:** The x and y coordinates of mouse clicks or unclicks
- **Player Words and Score:** The word that a player played as well as that word’s point value
- **Computer Words and Score:** The word that the computer played as well as that word’s point value
- **Player Rack Status:** The current tiles in the player’s rack of letter tiles
- **Game Beginning and End:** When the player begins and ends a game as well as that game’s outcome

Additionally, each game was tagged as either *finished* or *unfinished* based on whether or not the player ended the session prematurely. If the game was allowed to continue until either the computer player or the human player had won, then the game was marked as *finished*. If the game ended before a winner was declared, then the game was marked as *unfinished*. Once this was completed, 148 games had been tagged as *finished* while 47 games had been identified as *unfinished*. A summary of our dataset can be found in Table 1.

Since we want to make predictions as the player plays the game, we treat this data as a time-series. In order to do this, the values of each of five features (described in the next section) were calculated at each turn and used to create a time-varying data set. For example, the training set for the first turn will contain feature values calculated using only data available on the first turn. The training set for the second turn will contain feature values calculated using data available on both the first and second turns, *etc.* If a game has ended on the first turn, its data will not be included in the training set for the second turn. As a result of this process, we have training sets representing 21 turns worth of game play. Thus, after each of a player’s turns, we can ask: “Based on the characteristics of their play so far, is the player going to continue to play to the end of the game?”

Hypotheses

We hypothesized that there are at least five features of gameplay that are predictive of when players will terminate play sessions. Using the above logged information, we derived the following set of five quantitative features:

- **Score difference:** The difference between the player’s and the computer’s score
- **Turn Length:** The amount of time that has passed from the beginning of a turn to the end of that turn (characterized by the final word that is accepted by the game)

- **Word Length:** The length of the last word played
- **Word Score:** The point value of the last word played
- **Word Submitted:** The length of time in between word submissions (because players can submit as many words as they want until one is accepted by the game, we looked at submissions separately from acceptances)

We selected these five features in part because they are relatively game-independent. Score difference and turn length are both very generic. While word length, word score, and word submitted are more particular to Scrabble, they do have more generic analogs in other casual games, like action complexity, previous round score increase/decrease, and action interval respectively. Although testing the generalizability of these features is beyond the scope of this paper, we are encouraged these features or features similar to these can be found in many other games. We wanted to determine if these five features could be used to determine if a player was going to end the current game prematurely.

Baseline Quantitative Feature Evaluation

We initially treated this problem as a classification problem. In these experiments, the five features will be used together to determine if a player will quit the current game early.

Methodology

To evaluate the effectiveness of these features, we chose three classification techniques and evaluated their resulting prediction accuracies. Specifically, we used a Bayesian Network (Friedman, Geiger, and Goldszmidt 1997), Multilayer Perceptron (Rosenblatt 1961), and C4.5 Decision Tree (Quinlan 1993). Because the different biases inherent in these algorithms make their suitability for individual data sets different, we felt they would be reasonably representative of existing classification methods. We assume that any measures made to keep the player playing will not have any adverse effect on players that were not going to quit prematurely; however, if we misclassify an unfinished game as finished, then no measures can be taken to retain that player, which is what we want to avoid. Thus we are mostly interested in our ability to accurately predict an unfinished game. To test each of these methods we used 10-fold cross-validation and recorded the classification accuracy.

Before the experiments were run, all values were discretized into 3 bins: low, medium, and high. These bins were created by using the following equation:

$$D(f_i(p, j)) = \begin{cases} low & \text{if } f_i(p, j) < B_{i,j} \\ med & \text{if } f_i(p, j) < 2B_{i,j} \text{ and } f_i(p, j) \geq B_{i,j} \\ high & \text{if } f_i(p, j) \geq 2B_{i,j} \end{cases} \quad (1)$$

In the above equation, $D(f_i(p, j))$ is the discretization function given the value $f_i(p, j)$ of feature i on turn j for player p and B_i is the size of each bin for feature i . $B_{i,j}$ is calculated by dividing the range of feature i on turn j by three. As you can see in the above equation, bin size is calculated by considering all values up to the current turn to determine the max and min values. We consider values that are greater

Table 2: Prediction accuracies for baseline set of experiments on each turn. After the turn 6 the prediction accuracy never increases. Values that are greater than our prediction threshold are bolded.

| | T1 | T2 | T3 | T4 | T5 | T6 |
|---------------|-------------|-------------|------|------|------|------|
| Bayes Net | 0.32 | 0.23 | 0.08 | 0.10 | 0.05 | 0.00 |
| Perceptron | 0.17 | 0.25 | 0.22 | 0.13 | 0.10 | 0.15 |
| Decision Tree | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |

than two standard deviations away from the average value of a feature as outliers and do not include them when determining the bin size. We do this to prevent outliers from skewing the size of each bin. These values are still used in the model learning and class prediction processes. As you can see in Equation 1, the discretization function D transforms every continuous value for every feature into either *low*, *medium*, or *high*. This technique for calculating bins will be relatively robust to outliers except in degenerate cases such as bimodal distributions of values. We empirically verified our data didn't fall into these degenerate cases.

Results and Discussion

The results of the cross validation over the course of 6 turns can be found in Table 2. For reported prediction accuracy, we are only considering how well the algorithms predict if a player will quit the game early and as such, we are only reporting each baseline classifier's performance on those games. The reason that we only show results for the first 6 turns is because the prediction accuracy for unfinished games does not improve passed that turn. For evaluation, we compare against the baseline of 24.1% prediction accuracy. This was chosen as our baseline because the *a priori* probability of any individual game resulting in early termination is 0.241 in our training data. If our models can produce a predictive accuracy of greater than 24.1%, then we are making predictions that are better than guessing. As can be seen in Table 2, these techniques result in prediction accuracy well below the baseline in all but one case.

These results show these baseline features are not sufficient to predict whether a player will quit a game before the game is over using standard classification techniques.

Feature Transformation

We further hypothesized that while the baseline features themselves may not be informative enough to make a prediction, the deviations in these values from what the average player would do will be useful. To test this hypothesis, we transformed the set of baseline features into a new set of features that measures how much each player diverges from the expected value of each feature.

Methodology

This set of experiments consisted of two phases. First, our original dataset was transformed into to capture each player's deviation from the expected value for a feature. Second, we derived a naive Bayes model and used it calculate the probability of quitting the game early.

Deviation Based Dataset In order to test our second hypothesis, we converted the baseline analytics features into deviations from the mean value for each feature. For each of the five features mentioned above, we used the following formula to calculate how much a player deviated from a feature’s average value as

$$f'_i(p, j) = \left| f_i(p, j) - \frac{\sum_{p'} f_i(p', j)}{n} \right|. \quad (2)$$

In the above equation, $f_i(p, j)$ is the value of feature i on turn j for player p as before, and n is the number of players in the training set. $f'_i(p, j)$ is the absolute value of the difference between the player j ’s value for feature i on turn j and the average value for feature i on turn j . We then discretized $f'_i(p, j)$ into equal-sized bins.

This process produced five separate training sets each corresponding to a different feature. To avoid issues with data sparsity, we chose to consider each feature in isolation. If we considered all five features together, the number of possible configurations on any given turn would be $3^5 = 243$. This means, that we would need on the order of 243 games to see every configuration of features for just one turn (assuming a uniform distribution of feature values). The number quickly explodes when more than one turn is considered at a time. In fact, if we consider a game that lasts 10 turns, it would take on the order of $243^{10} = 7.17897988 * 10^{23}$ games to explore the space of possible configurations (assuming a uniform distribution of feature values).

Even if we only consider each feature in isolation, it is possible that we can run into sparsity issues for longer games. If $f_i(p, j)$ can only take on three values, it would still take on the order of $3^{10} = 59,049$ games to explore the space of possible configurations for a 10 turn game. To alleviate this, we do not consider a player’s entire turn history and instead, consider just the last three turns. The reason that we chose to examine the last three actions was to balance between discriminatory power and data sparsity. If we examine the last two actions, we lose discriminatory power while having very few data sparsity issues (only requires on the order of $3^2 * 9 = 81$ games to explore a 10 turn game). If we examine the last four actions, then we gain discriminatory power but increase data sparsity issues (requires on the order of $3^4 * 7 = 567$ games to explore a 10 turn game).

A Naive Bayes Model Our hypothesis was that these transformed expectation-deviation features can be used as indicators for when a player will prematurely end a game. We chose to model this using a naive Bayes model:

$$P(c|i, s) = \frac{P(i|c)P(s|c)P(c)}{P(i, s)} \quad (3)$$

In the above equation, c is the class label (we’re specifically concerned with $C = \text{unfinished}$), i is the turn number, and s is a sequence of the last three feature values. We choose to calculate $P(c|i, s)$ using the naive Bayes model because computing it directly would have proven to be quite difficult. Since we have a small dataset, it is highly unlikely that we can compute the actual probability distribution for $P(c|i, s)$ based only on the 195 games collected. It is much easier, however, to calculate the four other probabilities:

Table 3: Average percentage of a game spent in *warning state*. *Warning state* is defined as any three action sequence that predicts the game ending prematurely with a probability of greater than our probability threshold. Therefore, these percentages show what percentage of three action sequences in a game predict that the current game will end prematurely.

| | Finished | Unfinished |
|------------------|----------|------------|
| Score Difference | 18.8% | 43.2% |
| Turn Length | 30.0% | 50.8% |
| Word Length | 22.2% | 51.7% |
| Word Score | 17.4% | 49.2% |
| Word Submitted | 32.2% | 65.2% |

- $P(i|c)$: The probability of it being turn i given the class label c . This can be calculated by finding the number of games that lasted at least until turn i in class c compared to the total number of turns in all games in the class.
- $P(s|c)$: The probability of observing the three-action sequence s given the class label c . This can be calculated by finding the number of times that the sequence s appears in games in the class compared to the total number of sequences in all games in the class.
- $P(c)$: The probability of class c . This is the number of examples of class c over the total number of training instances. In our data set, this was 0.241.
- $P(i, s)$: The joint probability of seeing sequence s on turn i . This is defined as $\sum_c P(i|c)P(s|c)P(c)$. In other words, it is the sum of the numerator in Equation 3 for all values of c —a normalizing factor.

Like before, we are using a probability threshold of 0.241. Recall that we use this value because that is the probability that a game drawn at random from the data set will be an unfinished game. Any sequences s such that $P(c|i, s) > 0.241$ are considered informative since they have predictive power greater than that of a random guess.

After finding these informative action sequences, they can be used to flag certain periods of game play as *warning states* that a player is more likely to quit the current game early. Once we had obtained these sequences, we used them to calculate the percentage of time that players spent in this warning state. To calculate this percentage, we examined all sequences of three consecutive actions in the same class and determined how many of them had $P(c|i, s) > 0.241$. We then divided this number by the total number of sequences in that class to determine the final percentage.

Results and Discussion

Using the above probability threshold, we found several sequences of actions for each of the five transformed features that are informative as to whether a player will end a game early or not. We calculated the percentage of a game that each player remained in a *warning state*, a state in which their probability of ending the game prematurely was higher than our threshold. The results for that set of experiments can be seen in Table 3. The important thing to note is that players who end games prematurely spend much more of

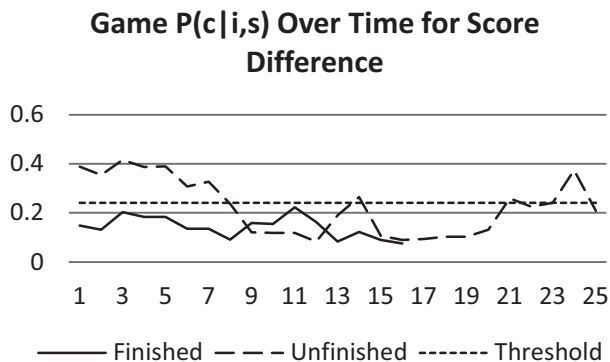


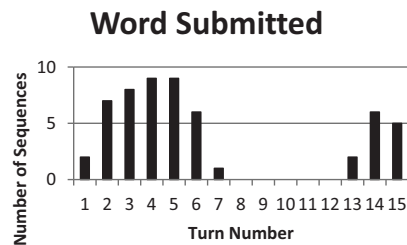
Figure 2: Probability the player will quit based on the score difference feature. The short dotted line is the baseline, the solid line is the probability of the player eventually quitting at each turn for an example finished game, and the long dotted line is the probability of the player eventually quitting at each turn for an example unfinished game.

Table 4: Number of informative sequences in the beginning, middle, and end of a game. Beginning is the first five turns, middle is six through ten, and end is 11 and on.

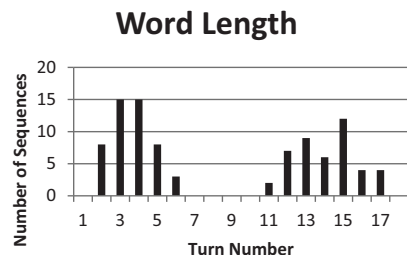
| | Beginning | Middle | End |
|------------------|-----------|--------|-----|
| Score Difference | 45 | 12 | 9 |
| Turn Length | 6 | 2 | 3 |
| Word Length | 46 | 3 | 44 |
| Word Score | 53 | 20 | 19 |
| Word Submitted | 35 | 7 | 13 |

their game time in a warning state than players that will finish the game normally—at least twice as often, and all greater than 43% compared to at most 32.2% for players who completed the game. Therefore, the longer that a player spends in a warning state can also act as a predictor of whether a player will end the current game prematurely. For a better illustration of this, see Figure 2. This figure shows how these probabilities change over the course of a game for a given feature (in this case, it is the score difference feature). Notice that for the finished game in Figure 2 $P(c|i, s)$ never rises above the threshold value whereas the probability in the unfinished game in Figure 2 stays above the threshold for a significant amount of time. These examples were chosen to be illustrative—not all features or games were as clear cut (as indicated in Table 3).

These sequences will be most useful to game designers and developers if they are able to identify games that are likely to end prematurely as early in the game as possible. Table 4 shows the distribution of sequences based on the turn that they occur on. As you can see, most informative sequences occur in the beginning of a game; however, the word length feature and the word submitted feature display bimodal behavior. In Figure 3(a) we show the word submitted occurrence histogram in more detail and in Figure 3(b) we show the word length occurrence histogram in more de-



(a) Word Submitted



(b) Word Length

Figure 3: Turn histograms for the word submitted and word length features. This shows how many informative sequences were found on each turn.

tail. As you can see, both of these have a high concentration of sequences found at the beginning and the end of the game with very few sequences found in the middle of the game. This implies that the characteristics of players’ play early and late in the game are the most important when it comes to determining they will end the game prematurely.

Finally, we sought to draw general conclusions about the sequences that are predictive of players quitting. A summary of the composition of the sequences is in Table 5. Note that there are two types of features: those correlated with score and those based on time. The score difference, word length, and word score features are correlated with the player’s performance. For the word length and word score features, predominantly high deviations in position one and lower deviations in positions 2 and 3 indicate an increased likelihood of quitting. Given that, it makes sense that the opposite is true for score difference. If lower score words are submitted, the gap between the computer’s and player’s scores is likely to widen. When considering the time-based features turn length and word submitted, we see a strong trend from low deviations towards high deviations from average being predictive of quitting. In this case, players likely either are turned away by the higher cognitive load as the board fills in, or become distracted and less interested in playing.

Future Work

Continuing our work, we are interested in examining techniques to procedurally make those changes. For example, perhaps it is enough to force the AI to make sub-optimal moves or play more slowly. Another question that arises is if it is enough to simply manipulate the actions that the

Table 5: Percentage of values at each position in an action sequence for each feature. Position 1 represents the earliest event looked at while Position 3 represents the most recent event looked at.

| | Position 1 | | | Position 2 | | | Position 3 | | |
|------------------|------------|--------|-------|------------|--------|-------|------------|--------|-------|
| | Low | Medium | High | Low | Medium | High | Low | Medium | High |
| Score Difference | 30.3% | 37.9% | 31.8% | 33.3% | 31.8% | 34.9% | 10.6% | 42.4% | 47.0% |
| Turn length | 72.7% | 27.3% | 0.0% | 45.5% | 18.2% | 36.3% | 27.4% | 36.3% | 36.3% |
| Word Length | 28.0% | 53.7% | 18.3% | 32.3% | 37.6% | 30.1% | 39.8% | 23.7% | 36.5% |
| Word Score | 26.1% | 26.1% | 47.8% | 17.4% | 54.3% | 28.3% | 28.3% | 41.3% | 30.4% |
| Word Submitted | 71.0% | 20.0% | 9.0% | 36.4% | 38.2% | 25.4% | 18.2% | 38.2% | 43.6% |

player will take such that they match low-probability sequences that were found. If we can find sequences of difference features that are highly-predictive of finishing (rather than quitting) a game, is it enough to manipulate the game so those sequences occur? Or do we need an ability to actually influence players' mental states? Another way of asking this question is are these difference features characteristic of mental states only? Or can these features actually influence mental states as well?

We can also see if this technique for identifying action sequences that signify that a player will end their game early generalizes to other games. There are many other types of games that fall in the category of *social games* so it would be informative to see if we can identify predictive sequences in these games. We can also test this in non-social game environments using more classic game genres such as first-person shooters and adventure games.

Conclusion

In this paper, we presented a set of features that can be used to describe gameplay in the game, *Scrabblesque*. While some of these features may be game specific, we are confident that several of them can transfer to other games of different genres. We illustrated the challenge of using these raw features to predict whether or not players will quite playing by using three standard classification algorithms, all of which resulted in poor accuracy. We applied a feature transformation to create a dataset based on each feature's deviation from the expected value. We then used this dataset to create a more accurate naive Bayes model for determining the probability that a player will quit the current game.

By examining these probabilities, we have determined that there exist several indicators as to whether a player will prematurely end a game or not. By examining these sequences of actions, we have provided concrete insight into what kinds of behavior are indicative of players ending a game session prematurely. Specifically, we found that in general increases from low deviations from average toward higher deviations from average in time-based features are generally good indicators that players are more likely to quite a game early. On the other hand, the opposite appears to be true for score-based features. This information gives game developers insight into when players are likely to quit and when effort should be put into retaining players.

References

- Andersen, E.; Liu, Y.; Snider, R.; Szeto, R.; Cooper, S.; and Popovic, Z. 2011. On the harmfulness of secondary game objectives. In *Proceedings of the 6th International Conference on Foundations of Digital Games*, 30–37.
- Friedman, N.; Geiger, D.; and Goldszmidt, M. 1997. Bayesian network classifiers. *Machine learning* 29(2):131–163.
- Gilleade, K., and Dix, A. 2004. Using frustration in the design of adaptive videogames. In *Proceedings of the 2004 ACM SIGCHI International Conference on Advances in computer entertainment technology*, 228–232. ACM.
- Hagelback, J., and Johansson, S. 2009. Measuring player experience on runtime dynamic difficulty scaling in an rts game. In *Computational Intelligence and Games, 2009. CIG 2009. IEEE Symposium on*, 46–52. IEEE.
- Quinlan, J. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.
- Rosenblatt, F. 1961. Principles of neurodynamics. perceptrons and the theory of brain mechanisms. Technical report, DTIC Document.
- Ryan, R.; Rigby, C.; and Przybylski, A. 2006. The motivational pull of video games: A self-determination theory approach. *Motivation and Emotion* 30(4):344–360.
- Schoenau-Fog, H. 2011. Hooked!—evaluating engagement as continuation desire in interactive narratives. *Interactive Storytelling* 219–230.
- Spronck, P.; Sprinkhuizen-Kuyper, I.; and Postma, E. 2004. Difficulty scaling of game ai. In *Proceedings of the 5th International Conference on Intelligent Games and Simulation (GAME-ON 2004)*, 33–37.
- Sweetser, P., and Wyeth, P. 2005. Gameflow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)* 3(3):3–3.
- Tarng, P.; Chen, K.; and Huang, P. 2008. An analysis of wow players' game hours. In *Proceedings of the 7th ACM SIGCOMM Workshop on Network and System Support for Games*, 47–52. ACM.
- Tarng, P.-Y.; Chen, K.-T.; and Huang, P. 2009. On prophesying online gamer departure. In *Network and Systems Support for Games (NetGames), 2009 8th Annual Workshop on*, 1–2.
- Weber, B.; John, M.; Mateas, M.; and Jhala, A. 2011. Modeling player retention in madden nfl 11. In *Twenty-Third IAAI Conference*.
- Weber, B.; Mateas, M.; and Jhala, A. 2011. Using data mining to model player experience. In *FDG Workshop on Evaluating Player Experience in Games*.
- Yannakakis, G., and Hallam, J. 2009. Real-time game adaptation for optimizing player satisfaction. *Computational Intelligence and AI in Games, IEEE Transactions on* 1(2):121–133.