

# Towards Deep Player Behavior Models in MMORPGs

**Johannes Pfau**  
Digital Media Lab, TZI  
University of Bremen  
Bremen, Germany  
jpfau@tzi.de

**Jan David Smeddinck**  
Open Lab, School of Comp.  
Newcastle University  
Newcastle upon Tyne, UK  
jan.smeddinck@ncl.ac.uk

**Rainer Malaka**  
Digital Media Lab, TZI  
University of Bremen  
Bremen, Germany  
malaka@tzi.de

## ABSTRACT

Due to a steady increase in popularity, player demands for video game content are growing to an extent at which consistency and novelty in challenges are hard to attain. Problems in balancing and error-coping accumulate. To tackle these challenges, we introduce *deep player behavior models*, applying machine learning techniques to individual, atomic decision-making strategies. We discuss their potential application in personalized challenges, autonomous game testing, human agent substitution, and online crime detection. Results from a pilot study that was carried out with the *massively multiplayer online role-playing game Lineage II* depict a benchmark between *hidden markov models*, *decision trees*, and *deep learning*. Data analysis and individual reports indicate that *deep learning* can be employed to provide adequate models of individual player behavior with high accuracy for predicting skill-use and a high correlation in recreating strategies from previously recorded data.

## CCS Concepts

•Information systems → Massively multiplayer online games; •Human-centered computing → User models; •Computing methodologies → Machine learning approaches;

## Author Keywords

Neural networks; deep learning; HMM; decision trees; games; player modeling; personalization; game testing; adaptive agents; dynamic difficulty adjustment

## INTRODUCTION

Video game production and maintenance, especially for flagship productions, is reaching the limits even of what large companies can sustain. Following the demand of players, games grow more complex in terms of content and mechanics, where the action spaces become nearly endless, greatly increasing the number of things that could potentially go wrong. This includes players facing unbalanced challenges, software execution or gameplay bugs that go undetected, connectivity

issues with large-scale systems, and cheating or other unethical behavior. We approach the closing of multiple unsolved gaps in these areas of concern for game research and development based on an uncommon building block: *deep player behavior modeling* (DPBM). We discuss the potential of DPBM with regard to the challenges indicated above. To establish apt representation techniques we also explore the potential of different machine learning techniques for player modeling in *massively multiplayer online role-playing games* (MMORPGs) and implement a pilot study which provides a first data set and enables the comparison between selected models. We hypothesize that different advantages can be attained from *Hidden Markov Models* (HMMs), *decision trees* (DTs) and *deep learning* (DL), in terms of analytic capabilities and prediction power. After outlining the concept of employing user modeling with machine learning for approaching challenges in game research and development, we present the selected techniques and illustrate the results of an exploratory study that was carried out with the established MMORPG *Lineage II*. Deep learning appears most adequate in terms of prediction accuracy and behavior representation similarity, whereas HMMs and DTs offer useful visualization and analysis features. These models also constitute the basis for ongoing subsequent future work focusing on the evaluation of user experience in different game modes.

Through this paper, we contribute a general discussion of potential application fields of DPBM and machine learning in the context of digital games, provide an overview of the state of the art in research and industry, and point out distinct advantages of player behavior models. Additionally, our exploratory study exemplifies an early working utilization, highlighting advantages and disadvantages of the different models.

## BACKGROUND

In related work, player behavior modeling has been approached mostly with the goal of facilitating *dynamic difficulty adjustments* (DDA) [1, 8, 20, 43, 50, 58]. Further application areas that have been discussed are the modeling of behavior impressions for informing game development [6, 9, 26] and the reproduction of atomic tasks [12, 47]). After briefly highlighting MMORPGs as an especially fitting class of games for the application of DPBM with machine learning in the next section, the subsequent sections introduce general categories that encompass the application areas of these isolated reports and discuss the potential of individual behavior models for tackling common challenges in games research and development **A-D**.

### The Case for Player Modeling in MMORPGs

MMORPGs typically encompass several aspects that highlight the potential benefits of the application of player modeling. The vast amount of data that each individual player is generating constantly along with the immediate opportunity to compare it to the global behavior data of a big community provides a rich basis for powerful but data-demanding machine learning algorithms. Moreover, the player behavior in MMORPGs can most often be broken down into *movement behavior* and *action selection* (skill usage). Each skill in this context is unique and discrete, which allows for less noisy behavioral state categorization compared to other popular video game genres, such as *first-person shooters* (FPS) or *real-time strategy* (RTS) games, where behavioral data quickly gets noisy [53]). Also, the continuously required internet connection simplifies centralizing and outsourcing the computational effort (e.g. through cloud computing). In return, player behavior models can enrich MMORPGs in various ways, such as increasing novelty and the prevalence of interesting challenges (see **A**), human tester relief and predicting the effects of game changes (see **B**), preventing game or match disruptions (see **C**), and preventing unfair or unethical player behavior which can not only harm the player community but can also cause financial losses for players and companies (see **D**). These aspects are further discussed in the following sections to clarify how these potential benefits of DPBM can come to play in (massively) online multiplayer games.

#### A. Personalized Challenges

Due to the complexity and scale of the game environments and interactions it can be difficult to consistently present motivating, well-perceived and evenly balanced challenges in player versus environment (PvE) modes of MMORPGs. Challenges frequently task players with defeating a large number of enemies at once (all substantially weaker than the player character) or with approaching powerful boss enemies (substantially stronger than the player characters) in a group or crowd of players. An even, one-to-one challenge is generally only found in player versus player (PvP) modes, when players may face other players on approximately the same skill level. To be able to compete in these matches, players have to constantly improve their skills and adapt to the specific situation and opponent, which in the end leads to a less repetitive gameplay providing more motivation for long-term commitment. As indicated by Fuster et al. [16], PvP-focused players spend significantly more hours per week on playing MMORPGs than PvE-focused players. PvE lacks this kind of a continuously changing challenge since non-human opponents are almost always constructed by “*simple rule-based finite and fuzzy-state machines for nearly all their AI needs*” [55], which quickly become predictable as there is no way to tailor their behavioral skill level to individual players, or to vary their behavior in a complex yet not overly random manner.

Thus, the guiding idea behind an ongoing, adaptive challenge is the maximization of *interestingness* (as introduced by Yannakakis et al. [56]) through adequate player modeling [5, 57]. The models proposed in this paper are able to represent player behavior individually and therefore result in the behavior of an agent on approximately the same skill level as the original

human player. Given such abilities, challenging “oneself” can present continuous and powerful DDA, since players have to adapt and overcome their own behavior in order to be less predictable. In this light, genuinely balanced challenges can be provided to players on a generative basis. Moreover, if the skill level of a player can be quantitatively assessed by means of player modeling, novel PvE modes are conceivable that confront the player with ever-changing enemies originating from a potentially large set of human players with adequate skill levels.

#### B. Autonomous Game Testing

Automatic simulations of video game play have proven to be usable in situations where human testing is too tedious or not exhaustive enough for the purpose of finding bugs and glitches [4, 15, 38, 45], parameter tuning [60], and assuring solvability [42]. Based on the insights and the potential of our previous work on a tool for completing and debugging adventure games [35]), we want to further extend the possibilities of autonomous game testing.

For developers, one of the most difficult and time-consuming phases of the game design process [22] is the balancing of different character classes. Following the definition of Sirlin [44], a multiplayer game is “*balanced if a reasonably large number of options available to the player are viable*” (where viability sets the requirement of having many meaningful choices throughout a game), while “*players of equal skill should have an equal chance at winning*”. Together with frequently desired asymmetrical character configuration possibilities this inherently leads to combinatorial explosions, which can become hazardous for the enjoyability of the game and the satisfaction of its players [39]. Even worse, balancing issues most of the time “*only become apparent after many months of play*” [19] and the trouble with these issues (in comparison to straightforward fixable bugs, glitches and solvability aspects) is that they do not only appear during the launch of a newly published game. Balancing is an ongoing and repeating task that is heavily influenced by the perceptions of the player community (“*after each patch, often the discussion begins again, factoring in new balancing or abilities for each class*” [26]). In the games industry this is most often approached through long-term expert analysis, excessive human play-testing, and persistent debates with the community.

Academic work presents approaches to tackling the issue of balancing different setups by simulation [3, 22] or genetic algorithms [25, 28, 29], yet without incorporating additional information about situated player behavior. Individual player models have the potential to unite automatic simulation methods with behavioral information. This gives developers the opportunity to receive practically immediate insights on a) which player strategies are popular, dominant and/or may require rework, b) how parameter tuning will likely alter the outcome of strategies before presenting it to the community and c) how to automatically balance game mechanics after large-scale permutations of classes, setups, parameters and behavior – in all stages of development.

### C. Human Player Substitution

Most multiplayer games throughout the popular genres have to handle disconnected players (or those who stop providing input for a longer time). In some cases, for balancing or game-experiential reasons, these lost players are replaced by computer-controlled agents (e.g., *Left 4 Dead* [52] (an FPS), *Heroes of the storm* [13] (a multiplayer online battle arena game), *Super Smash Bros. 4* [2] (a Beat 'em up), *Mario Kart 8* [10] (a racing game), *Civilization V* [17] (a turn-based strategy game), *Company of Heroes 2* [14] (an RTS), or *Rocket League* [37] (a sports game)). However, such substitution is frequently criticized, since the replacing agent is usually not able to compete with human players. On the other hand, developers cannot allow the deployment of computer-controlled agents that are clearly stronger than the replaced human player due to the obvious potential of abuse. Thus, the only satisfying replacement would be an agent that acts very much like the original human player and performs neither significantly better nor worse than her (at least until the original player returns). After the evaluation of application area A (approaching player behavior modeling for generative adaptive challenges), the computed models presented in this work will be used in order to assess whether computer-controlled agents perform on appropriate skill levels and present human-like behavior, controlling whether they a) even strike the attention of other players in the match and b) whether they can serve to replace lost players on an adequate level (i.e. showing a consistently comparable performance). At least one of the mentioned criteria should be accomplished in order to achieve temporary match disruption prevention.

Another possible application area is presented through the upcoming growth of asynchronous games [41], especially in the current age of mobile gaming: In many instances of this type of games players can challenge other players without the need of actually playing at the exact same time as the opponent. This allows the fast-paced and very situation-dependent world of mobile gaming to feign battles of various genres like strategy games, turn-based games, but also even real-time role-playing game battles so they can take place whenever it suits a player (cf. e.g. *Clash of Clans* [46], *Pokeland Legends* [51], *Star Wars: Commander* [21], *Goddess: Primal Chaos* [24]). As in the previously mentioned case, human opponents are represented by computer-controlled agents with the same character setup, equipment and/or further attributes, but lack individual decision making/behavior. This again leads to a misrepresentation of the actual human opponent's skill, a potentially unfair advantage for the attacking/challenging player, and consequently causes high, skill-independent fluctuations in leaderboards. Player behavior models as described in our approach could be integrated in asynchronous games or game modes to further extend the opportunities of this type of games and to give both the attacker a more appropriate challenge and the defender a better and fairer representation of herself in her absence.

### D. Cheating and Botting Detection

One of the major classification paradigms in which player modeling has successfully been used before is the detection of unwanted automated software (botting) in online

games [7, 18, 23, 31, 33, 48]. Malicious bot software has no or little access to the actual game variables and objects and is thus usually based on heuristic or predefined decision making. Above that, botting is used mostly in worthwhile areas and thus typically makes use of fixed paths, leading to rigid movement behavior. As such, differences between the classes of bot and human player can be identified quite accurately given the aforementioned techniques. Another problem in online games is the act of identity theft, where criminals gain unwanted access to user accounts. Existing approaches tackle the issue through different means of automatic detection [34, 54]. We argue that these approaches can be extended by employing in-depth player behavior models for the classification between real human account / character owners and imposters. Finally, competitive games are always prone to cheating or hacking. In such cases, DPBM can be employed to improve play-style analytics in order to classify suspicious or technically impossible behavior.

Lastly, beyond above major application areas, these models have the potential to aid in classifying player roles [11], to assess the player's experience [27] based on his behavior, and the live application of DPBM also opens the door for developers to enhance player experiences with completely novel game mechanics in existing or potentially newly created game modes.

### APPROACH

Following the definition by Yannakakis et al. [56], a game is only interesting when it "is neither too hard nor too easy", shows "diversity in [opponents'] behavior over the games" and "[opponents'] behavior is aggressive rather than static". That means that a) strictly optimal behavior is just as little interesting as conventional, predictable heuristic non-player character (NPC) behavior, b) opponents should evolve over time in order to constitute a dynamic challenge and c) players should experience a tension similar to the confrontation of a human opponent. To these ends, we aim for a model that displays increasing player behavior fitness when presented with increasing amounts of data showing similar behavior.

### Study

In order to assess which models and methods are sufficiently expressive and accurate, a pilot study has been conducted, resulting in a viable initial set of behavioral data. We chose to gather this data set in isolated play sessions in order to control the setting, collect verbal reports from participants, and to reduce confounding variables in comparison to noisy "in the wild" data. The convenient subject study participants were asked to maximize their score by defeating the highest possible number of enemies within 30 minutes in an open-world PvE game mode of the popular MMORPG *Lineage II* [32]. To model players with and without previous experience in the game, recruitment took place on a private server of this game and via email. As exemplary applications for investigating the applicability and performance of the different modeling approaches, we phrase the following assumptions for player behavior.

Players:

- prefer to use skills in *rotations*, i.e. sequentially
- use certain skills in certain situations, e.g.,
  - *initial* skills for each enemy (depending on whether the intention of the enemy is idling or attacking)
  - skills only viable when own/target HP is low/high
  - skills only viable when distance to the target is low/high
- choose different strategies in different game modes and against different enemies/classes

Thus, in order to allow the models to incorporate these cues, game states include (but are not limited to) the variables contained in these assumptions, such as previously used skill(s), health point (HP) conditions and distance between player and non-player character(s) (cf. Table 1), while game actions include all used skills and movement of the player.

#### Measures

An initial questionnaire asked for demographics and video game experience. During the task, we recorded movement data as continuous paths and skill usage by logging the most important character and target state information. After the play session, the participants were asked to complete a questionnaire containing *Player Experience of Need Satisfaction* (PENS) [40] and *Intrinsic Motivation Inventory* (IMI) [36] items in order to gather prediction training and validation data for later use. The participants were also asked to describe the strategies they employed when encountering enemies in detail while observing a replay of their player behavior. Skill icons and descriptions were shown during the replay for easier reference. The participants were asked to explain common skill rotations and rare / notably situated skill usages specifically. Lastly, the participants were asked to discuss their perception of the interestingness of computer controlled enemies in MMORPGs and to compare them to the experience of encountering a human player (in PvP). All data were evaluated in a pseudonymized fashion and stored in an encrypted file container.

#### Procedure

The study was executed in an online setting. Subjects were asked to download the game client in advance and met the experimenter on a TeamSpeak3 server, enabling voice communication throughout each session. Following informed consent and the pre-study questionnaire, participants chose between three different classes (Warrior, Archer or Wizard), were able to customize their skill configuration and test it on non-responding training dummy enemies without temporal restrictions (typically lasting 5-10 minutes). When the participant felt ready, the experimenter started the countdown of 30 minutes and teleported the player character to the treatment start location. In this place, a large number of common MMORPG enemies appeared that could be attacked in order to raise the participant's score. The score and the remaining time were displayed at all times. Upon death, the character was revived at the initial location. Throughout the whole in-game task no other player characters were present. After the countdown completed, the game shut down automatically and

the remaining questionnaires (PENS, IMI) were presented. This setting is representative for many tasks that MMORPGs present. In this pilot study we chose to focus on the assessment of single-player behavior first in order to benchmark the respective models before we broadening the scope to more noisy and also socially dynamic multiplayer settings.

variable	value
time	08.02.2018 15:26:16
<b>timeReuseAvailable</b>	08.02.2018 15:26:22
<b>skillID</b>	10771
skillName	Multiple Arrow
<b>casterID</b>	268492397
casterName	TestArcher
casterClassID	162
casterClassName	Archer (Yul Sagittarius)
<b>casterHPpercentage</b>	100
<i>locX</i>	-11965
<i>locY</i>	237519
<i>locZ</i>	-3213
targetID	23355
targetName	Armor Beast
<b>targetClassID</b>	-1
targetClassName	NPC
<b>targetHPpercentage</b>	100
<b>ai_intention</b>	AI_INTENTION_ATTACK
<b>distance</b>	309.36
<i>locXtarget</i>	-11936
<i>locYtarget</i>	237827
<i>locZtarget</i>	-3227
<b>zone</b>	Hellbound (Study)
score	34

**Table 1.** Example database entry for *skillLogs*. **Bold fields** are used in the behavior model computation (DT, deep learning). *Italic fields* are used in movement analysis along with further detailed path data, while the remaining variables serve purposes in readability and visualization. **Zone** consists of location and game mode information. **AI\_intention** describes the current aim of the target, most notably IDLE, ACTIVE, ATTACK or CAST.

#### Participants

In total,  $N = 24$  subjects completed the task (87.5% male, 12.5% female, 22 to 28 years of age ( $M=25.2$ ,  $SD=1.96$ ), yielding a fair representation of MMORPG demographics [59]. All of them stated being active gamers with 11 to 25 ( $M=16$ ,  $SD=3.96$ ) years of previous video game experience and 20 to 55 ( $M=30.4$ ,  $SD=12.4$ ) hours spent on games per week, while 83% also indicated that they had played *Lineage II* before.

#### Results

The final score varied greatly among participants (122 to 434 defeated enemies,  $M=287.4$ ,  $SD=92.6$ ), with a significant performance difference between participants with and without previous experience in *Lineage II* ( $p < 0.01$  with a Welch's t-test, Cohen's  $d = 1.69$ ,  $df = 22$ ). We found positive (Pearson) correlations between score and the surveyed PENS: in-game autonomy ( $r=0.37$ ), presence ( $r=0.38$ ) and IMI interest-enjoyment ( $r=0.48$ ) sub-scales. Overall, the notable variance in their performance indicates that the participants have chosen different strategies to approach the enemies. The analysis

and differentiation between strategies is further assessed in the next section. Regarding general performance as indicated by score, no significant differences due to sex, education or other demographic variables were found. When they were asked to discuss the interestingness of computer enemies in MMORPGs generally, subjects shared a common opinion, that NPCs are “predictable”, “no real enemies”, “not comparable to the experience of PvP”, “often boring [...] without good AI” and that “the only fun comes from the rewards, not the battle itself”.

**Data analysis & Modeling**

After testing several machine learning algorithms and fitting techniques, namely HMMs, DTs, deep learning, clustering, regression, splines, support vector machines (utilizing TensorFlow and DeepLearning4j) for their applicability to capture the individual behavior accurately, we selected to report on HMMs, DTs, and deep learning (DL), which bear distinct advantages in performance or visualization capabilities. All models were trained on a training set (80%) from the gathered study data (one model per player) and later used to predict action selection on a testing set (20%) from the corresponding player (see *Prediction Results*) and to classify behavior between players (see *Player Differentiation*). The prediction accuracy was compared between all models, with and without the incorporation of the heuristics that stem from former assumptions (see *Study*).

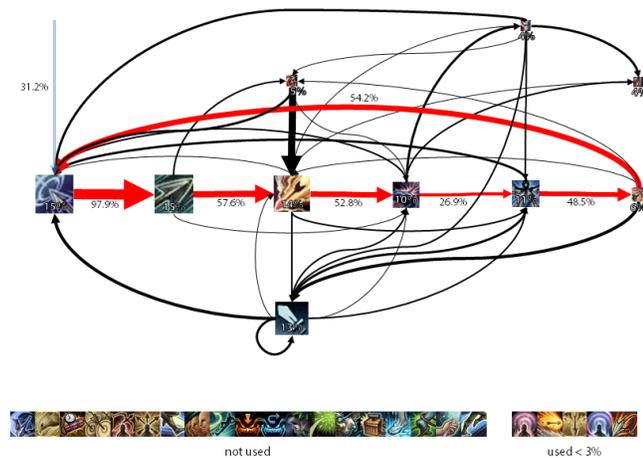


Figure 1. HMM skill transition graph of a single participant. White percentages and the skill icon sizes display the relative usage of the respective skill. The width of the transition arrows is proportional to the transition probability from one skill to another. The blue arrow shows the most likely skill to begin attacking each enemy, while red transition arrows depict the main rotation (transition probabilities are labeled black). Skills used in less than 3% of encounters are included in the calculations of the model but excluded from the visualization due to visibility reasons.

*Hidden Markov Models*

Following our first assumption (see *Study*), which was supported by similar reports in the post-test questionnaire, one major behavioral criterion is the adherence of individual skill rotations. Since HMMs shine in their capabilities of visualizing state sequences, we formulated the estimation of the main

rotation as a markov chain with the respective previous skills as observable variables, while the complex behavior strategy stays hidden.

Figure 1 displays the HMM for a single player and demonstrates the intuitive illustration of the probability of the skill successors given a previously known state. The most used skill together with its most probable successors constitute the individual main rotation, which differs from player to player (cf. Figure 2).



Figure 2. HMM-computed main rotations of all participants. Each line stands for the most likely rotation of a given player, from the first (most probable initial) skill across the following most likely successors. The highest transition probability of the last skill in line is the first skill again.

In order to increase the contextual integration capabilities of HMMs, we chose to extend the dimensionality of the original method by using 2nd order HMMs [49] at the cost of requiring more training data. We also integrated an *initial heuristic*, which assumes that players might prefer to attack enemies with certain skills initially (and thus, outside of their main rotation) and a *cooldown heuristic* which filters out idiosyncratic strategies that are not executable at the given point of time due to cooldown restrictions.

*Decision Trees*

While HMMs struggle with incorporating larger numbers of dimensions, DTs can break data down following the most discriminatory variables. This allows for pinning down decisive factors for skill usage accurately from a selection of many contextual game state factors that are potentially relevant. We included information about the enemy’s intention (IDLE, ACTIVE, ATTACK or CAST), the previously used skill and binary choices whether the player’s HP, the enemy’s HP or the distance between them is above or below the respective mean of the current player’s data (cf. Table 1, bold entries). Discriminativeness was calculated via Shannon entropy. As our outcomes show, this approach does not only yield a higher accuracy in predicting skill usage compared to HMMs, but it is even capable of “explaining” the intention of rarely used skills. By reversing the tree and collecting all paths ending in

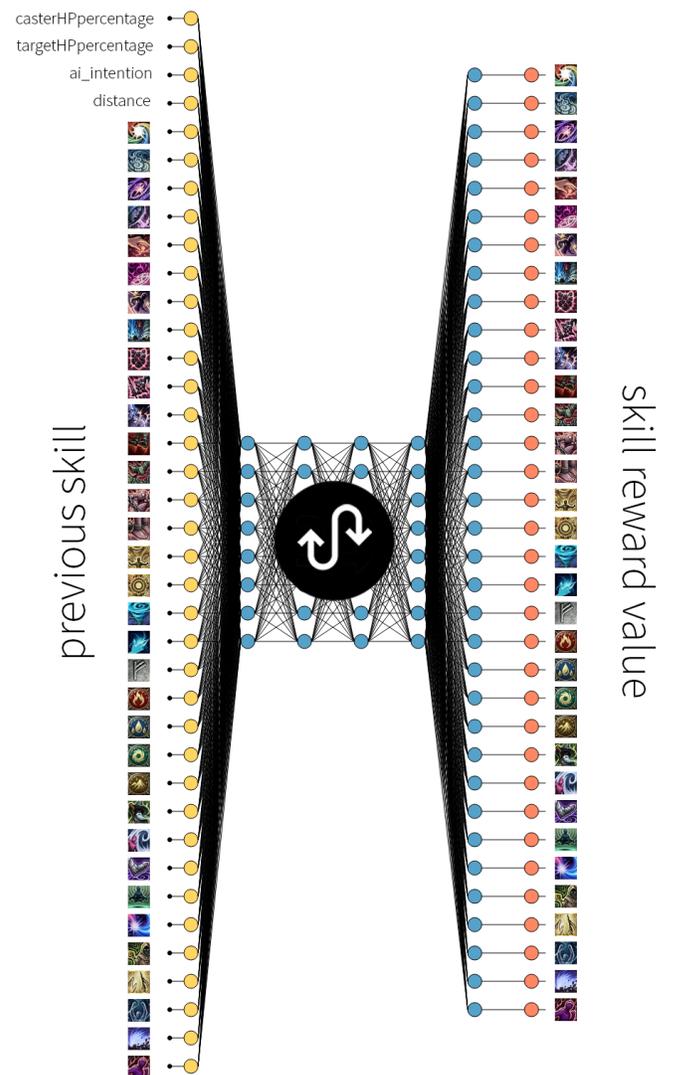
a particular skill leaf, the situational context of this skill can be illustrated.

For example, when asked for skills that were rarely used, one participant stated that he activated a skill (“Death Lord”) whenever his HP dropped to a low level, in order to transfer some of the enemy’s HP to his own. Reversing the tree returns “NOT HP above mean” (95.2% accuracy) as the top criterion for this skill, followed by “target HP above mean” (90.5%). The player’s contextual usage of this skill is thus accurately described by “having low HP while the enemy has high HP”. Furthermore, one player stated to use “Bow Strike” whenever an enemy gets too close, and thus knocking the enemy back - where the tree returned “NOT distance above mean” and “target HP above mean” as top criteria, explaining even more (since an approaching low HP target could be defeated quickly, but only approaching high HP targets are countered with the knock-back). One subject reported the usage of “Power Provoke” if – and only if – his HP are full and many enemies are around, since this skill taunts all of them to attack him, so that he can face all of them at the same time. The HP situation could be reflected, but the number of possible enemies to attack is a metric that was not logged, which should be considered in further research. Nevertheless, this process of reversing DTs produces a ranked set of meaningful variables in which particular skills are used and is capable of delivering clearly understandable insights to developers.

*Deep Learning*

Neural networks add further modeling performance since the learning process does not rely on manually defined discrimination criteria and all variables (again, cf. Table 1) contribute their real values instead of binary decisions, as is the case with decision trees. As a drawback, neither can the learned model be easily visualized nor can the process be reversed in order to describe situations in which particular skills are used. Above that, the computational and temporal effort of training and retrieving is considerably larger than for the former techniques. Nevertheless, the scalability of outputs beyond situations explicitly provided in the training data and high accuracy in prediction render deep learning a viable candidate for behavior generation. To establish general applicability we chose a multilayer perceptron with backpropagation and a logistic sigmoid activation function and trained one network for each participant, where input and output array sizes varied due to different numbers of skills used / available. Each network consisted of up to 38 input nodes and up to 34 output nodes (cf. Figure 3), while we ran a number of simulations for the best fit/effort ratio in terms of the number of hidden layers (1 to 5), nodes within (5 to 30) and training epochs (cf. Figure 2). Since skillIDs of previous skills are nominal and bear no meaning in their values, they had to be realized as individual input nodes. Target values were constituted by the use of particular skills given the situation defined from the input array. For the prediction afterwards, the computed output array is translated to a density function from which the guessed skill is picked probabilistically. Most fitting iterations did not improve significantly beyond 1000 training epochs, which were reached after about 7 minutes on a local i7-6700HQ CPU @2.60GHz

(using a single core). Retrieval time from a trained model did not exceed 20 milliseconds.



**Figure 3.** Example network for one participant. Real valued variables are mapped to the range from 0 to 1, previous skills are encoded as binary switches. Hidden layer and neuron count varied after optimizing for prediction accuracy.

*Prediction Results*

As shown in Figure 4, deep learning outshines our previous approaches with 55-97 % ( $M=71.4\%$ ,  $SD=13.2\%$ ) prediction accuracy across individual models. We did not compare the outcomes to complete random guessing, since the accuracy of random guessing in this high-dimensional action space would be <3 %. Rather, the baseline (BL) depicted stems from guessing with only the mere skill frequency probabilities of a given player, without further contextual information. HMMs succeed in extracting the most probable main rotations of the participants, but fail to explain the usage of rarely used skills. Second order HMMs (HMM<sup>2</sup>) yielded no significant difference in prediction compared to the former, while still inducing the cost of a considerably slower training curve. DTs

#	1	2	3	4	5	6	7	8	9	10
h	2	1	3	3	2	2	1	4	3	2
hn	22	5	15	12	3	13	8	5	17	12
acc	92	86	56	63	59	66	66	59	63	74
#	11	12	13	14	15	16	17	18	19	20
h	1	2	4	2	2	3	1	2	2	2
hn	6	8	14	7	6	12	17	21	17	9
acc	65	67	55	97	78	56	64	67	76	81
#	21	22	23	24						
h	3	2	2	1						
hn	8	11	7	12						
acc	66	89	97	71						

Table 2. Number of hidden layers (h) and nodes within (hn) used for each participant (#), resulting in the respective accuracy (acc) in %.

produce human-readable outcomes and facilitate attributing contextual factors to the situational usage of skills. However, they rely on manually defined criteria to accurately grasp borderline cases and introducing a large set of variables and/or decision criteria for these may harm the readability of reverse tree queries.

Among participants, prediction accuracy was significantly higher ( $p < 0.01$  with Welch’s t-tests, Cohen’s  $d > 1.03$  for all models) for experienced players compared to those who have not played *Lineage II* before (cf. Table 3). This indicates that the former stick more tightly to learned strategies and patterns whereas the latter are more eager in trying out different styles. Accordingly, this yields a slight correlation between score and prediction accuracy (HMM:  $r=0.19$ , DT:  $r=0.16$ , DL:  $r=0.21$ ).

		BL	HMM	HMM <sup>2</sup>	DT	Deep Learning
exp	M	36.5%	48%	47.9%	61.7%	73.4%
	SD	18.5%	17.1%	17.0%	16%	13.6%
inexp	M	19.3%	29.5%	29.5%	49.5%	61.8%
	SD	5.3%	7%	7%	4.7%	5.1%

Table 3. Average prediction accuracies between players with (exp.) and without (inexp.) previous *Lineage II* experience.

Regarding the former assumptions, the *cooldown heuristic* (temporarily discarding candidate skills from prediction that are not usable by design) significantly increased ( $p < 0.05$  with Welch’s t-test) the accuracy in HMMs and DTs, where no difference in the case of deep learning networks was found, since they inherently incorporated this information. The *initial heuristic* increased the accuracy in some participants’ cases, but decreased it in others, not leading to significant improvements.

#### Player Differentiation

Generally speaking, the study participants used skills differently. E.g., while some focused on defeating single enemies as quickly as possible, others attempted to gather larger group of enemies in order to utilize skills that damage multiple targets.

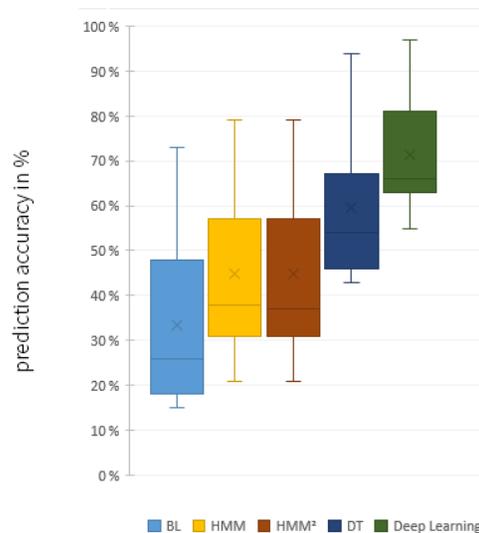


Figure 4. Prediction accuracies for BL, HMM, HMM<sup>2</sup>, DT and DL. Heuristics are included in the respective model variant if they yielded higher accuracy.

Certain players made efficient use of time-limited reinforcement skills (buffs), weakening skills (debuffs) and/or approximated the theoretical optimal damage rotation, whereas others stuck to personal preferences or even a seemingly random selection. Since our models should not only be able to predict skills from the trained player, but also be usable for differentiation between them, we benchmarked the respective model on the data from all other players (cf. Figure 5). In most of the cases (82.4%), player behavior is different enough so that it can not be predicted accurately from another model. However, 13.2% of the time, models explain considerable portions of the behavior of another player. Importantly, this does not necessarily reveal similarity between two players, since the explained behavior might only be a subset of the other player’s behavior. If we want to establish a similarity measure, which could be used in order to approach cheating or identity theft detection, we have to examine if the prediction accuracy is bidirectional - which is only the case in between subjects 3, 9, 10, 13 and 16.

#### Movement Analysis

We aimed to represent both fast-paced movement decisions as well as long-term movement plans. Engaging an enemy often evokes situational movement decisions depending on the individual strategy of the player (e.g., chasing an opponent or building up/maintaining a larger distance), therefore we included local movement decisions in the former presented skill usage model architecture by treating movement as a unique skill with distance and location parameters. Above that, players move according to their global intention [30] (e.g., reach certain points or areas), so decision making on a bigger scale has to be considered. We evaluated a number of fitting techniques and ended up with B-splines to approximate the overall movement behavior, compressing it to a smooth function (cf. Figure 6). The importance of global movement decision making will become more apparent in our follow-

testing data from subject #

#	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22
0	92%	0%	17%	0%	0%	13%	0%	0%	26%	0%	0%	0%	28%	0%	0%	0%	0%	0%	20%	0%	0%	0%	0%
1	0%	86%	0%	0%	28%	0%	0%	9%	0%	0%	23%	0%	0%	41%	0%	0%	11%	0%	33%	0%	0%	0%	37%
2	40%	0%	56%	0%	0%	22%	0%	0%	17%	0%	0%	0%	15%	0%	0%	0%	0%	0%	5%	0%	0%	0%	0%
3	0%	0%	63%	0%	0%	55%	0%	0%	61%	51%	0%	0%	92%	0%	41%	52%	0%	0%	73%	0%	75%	0%	0%
4	0%	30%	0%	0%	59%	0%	0%	23%	0%	0%	0%	27%	0%	61%	0%	0%	17%	0%	0%	51%	0%	74%	0%
5	21%	0%	15%	0%	0%	66%	0%	0%	11%	0%	0%	13%	0%	0%	0%	0%	0%	11%	0%	0%	0%	0%	0%
6	0%	50%	0%	0%	0%	66%	0%	0%	64%	52%	0%	0%	94%	0%	88%	49%	50%	0%	70%	0%	75%	0%	0%
7	0%	15%	0%	0%	23%	0%	0%	59%	0%	0%	0%	17%	0%	0%	20%	0%	0%	19%	0%	0%	35%	0%	29%
8	38%	0%	19%	0%	0%	13%	0%	0%	63%	0%	0%	0%	40%	0%	0%	0%	0%	43%	0%	0%	0%	0%	0%
9	0%	0%	48%	0%	0%	35%	0%	0%	74%	50%	0%	0%	92%	0%	84%	51%	0%	0%	68%	0%	77%	0%	0%
10	0%	0%	0%	51%	0%	0%	58%	0%	0%	66%	65%	0%	0%	92%	0%	55%	52%	0%	0%	73%	0%	62%	0%
11	0%	25%	0%	0%	33%	0%	0%	20%	0%	0%	0%	67%	0%	0%	58%	0%	0%	23%	0%	0%	38%	0%	71%
12	21%	0%	16%	0%	0%	19%	0%	0%	52%	0%	0%	0%	55%	0%	0%	0%	0%	45%	0%	0%	0%	0%	0%
13	0%	0%	0%	45%	0%	0%	52%	0%	0%	48%	47%	0%	0%	97%	0%	37%	42%	0%	0%	69%	0%	68%	0%
14	0%	30%	0%	0%	42%	0%	0%	26%	0%	0%	0%	34%	0%	0%	78%	0%	0%	19%	0%	0%	54%	0%	85%
15	0%	0%	0%	31%	0%	0%	32%	0%	0%	40%	32%	0%	0%	30%	0%	56%	31%	0%	0%	32%	0%	25%	0%
16	0%	0%	0%	49%	0%	0%	0%	0%	61%	53%	0%	0%	90%	0%	20%	64%	0%	0%	71%	0%	78%	0%	0%
17	0%	13%	0%	0%	29%	0%	0%	17%	0%	0%	17%	0%	0%	37%	0%	0%	67%	0%	0%	46%	0%	18%	0%
18	27%	0%	5%	0%	0%	8%	0%	0%	84%	0%	0%	0%	41%	0%	0%	0%	0%	76%	0%	0%	0%	0%	0%
19	0%	0%	0%	50%	0%	0%	46%	0%	0%	46%	46%	0%	0%	93%	0%	88%	47%	0%	0%	81%	0%	67%	0%
20	0%	26%	0%	0%	37%	0%	0%	28%	0%	0%	0%	23%	0%	0%	42%	0%	0%	21%	0%	0%	66%	0%	40%
21	0%	0%	0%	52%	0%	0%	33%	0%	0%	63%	53%	0%	0%	90%	0%	33%	53%	0%	0%	76%	0%	89%	0%
22	0%	27%	0%	0%	27%	0%	0%	21%	0%	0%	0%	35%	0%	0%	57%	0%	0%	16%	0%	0%	50%	0%	97%
BL	44%	22%	15%	29%	26%	17%	31%	14%	19%	42%	26%	26%	28%	73%	55%	15%	31%	21%	63%	48%	45%	69%	56%

used model trained on subject #

Figure 5. Prediction accuracy of participants using trained deep learning networks from different subjects. White values are equal or less than BL probabilities, yellow ones slightly better and blue ones significantly better. Green values depict the accuracy on the corresponding player, which is never surpassed by a different model.

ing research, which aims to facilitate categorizing decisions between points of interest and/or incorporating more complex gameplay choices in multiplayer settings. For now, the computed movement models were used for an approximate representation in the following replay section.

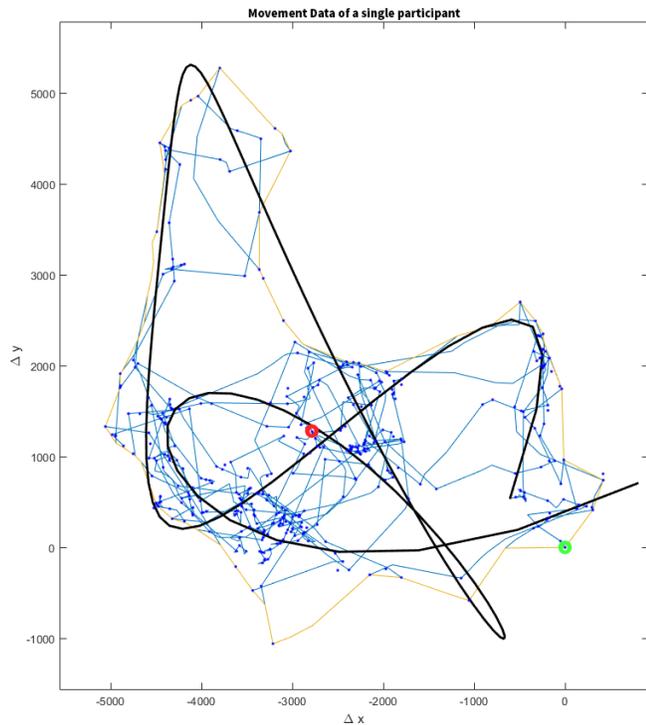


Figure 6. Movement data of a single participant. Blue lines visualize the users trajectory (starting at green, ending in the red spot), blue dots indicate skill usage. A yellow line encloses the travelled area. The thick black line shows the approximation of the movement behavior via a B-spline.

### Replay

Since we are lacking clear and distinctive automatically measurable criteria on “what is a good representation” of human behavior and cannot objectively draw a threshold from which percentage on theoretical prediction accuracy establishes close behavior in actual in-game situations, we chose to replay the study session for each individual participant with the difference that the behavior stemmed not from the player directly, but from the computed model for the respective player. The simulated agent followed the approximated path from **Movement Analysis**, targeted nearby enemies and acted according to the individual **Deep Learning** model, which computed the most probable action given the situational parameters. We compared scores from both groups (human agent and replicated behavior) using a paired t-test ( $p=0.42$ , Cohen’s  $d = -0.07$ ) and Pearson correlation ( $r=0.91$ ), which supports the quality of the model, since the outcome does not differ significantly and higher scores in the task completion correlate to higher scores in the replay.

### DISCUSSION

Testing the performance of the different candidate techniques for player behavior modeling, deep learning showed the best prediction accuracy for immediate skill use, while HMMs and DTs show clear causal paths leading to the prediction decision, which can provide benefits in those potential application categories that require human interpretation of the modeled decisions (mostly B, game testing; and partially D, cheating and botting detection). The ability of the models to support the exemplary applications for analysis and exploration in the pilot study delivers early evidence for the potential of DPBM in the context of the general application categories.

However, some limitations apply. Due to the explorative nature of this study a number of factors that potentially impact player behavior were not included. Participants reported some variables that they perceived to be influencing their decision making, such as the amount of enemies in their immediate surrounding, the presence of buffs applied to the player or debuffs to the enemy that were not yet recorded and represented in the training and validation datasets. When describing PvP combat, players also adapt their strategy to their enemies’ skill usage (and/or cool-downs). We did not model which enemies were attacked, which not and why not, yet this could have been helpful in order to further improve the replay session. Furthermore, the simple task did not bear any differences due to location that could be interpreted in movement analysis (aside from the area covered). In terms of the **replay** evaluation, a more expressive interpretation (such as analyzing human impressions by e.g. confronting players with videos of replay sessions and asking them to indicate which one represents themselves, without the information that the behavior is replicated) have to be considered in the future. For example, should players be able to correctly attribute model simulations to players for whom they were able to observe authentic gameplay behavior, it could be argued that the models appear to express characteristic and differentiable individual player behaviors.

### Future Work

In general, we are looking forward to extend the evaluation in terms of number of players captured and in terms of the observed play duration, while also improving the models in terms of accuracy and efficiency, as to develop a fully-fledged toolkit which can be effectively employed for academic and industrial use. We will examine further the application of deep learning methods, exploring and evaluating the use of recurrent, deep-belief, or more context-driven (e.g., long / short-term memory) networks. Regarding the most crucial limitations of this paper (i.e. the small sample size and the missing consideration of the influence of additional players), we will broaden the scope of observations substantially: Based on the insights gained from single-player behavior we seek to expand the models towards the inclusion of true multiplayer situations (both cooperating with and competing against human players). This includes gathering additional data from wide-ranging “in the wild” behavior rather than from heavily scoped tasks, which is more representative for real-world application. Within the models, we will add events as possible predecessors for skills (such as a major location change, receiving a buff or debuff, starting a match, dying, etc). Finally, we aim to evaluate the concept of deep player behavior models in settings that are more directly representative of the established categories **A-D**.

**Personalized Challenges** will be approached by recording and modeling behavior of individual players, who will then face the task of defeating an agent controlled by a generative DPBM. Since deep learning has shown to yield a high prediction accuracy while still providing fast output retrieval, it will be the prioritized model for implementation. We will assess whether there is a perceived difference in engaging an opponent that acts in the same fashion as the player, compared to traditional NPCs, human players, and agent behavior that stems from a blend of multiple DPBM (modeled from players that are on approximately the same skill level as the former player). Based on these observations, we will focus on the perceived challenge, interestingness and long-term motivation of the involved players.

For **Autonomous Game Testing**, we will utilize DPBM for simulations of player behavior in order to spot game balance issues and establish automatic parameter tuning. This will be carried out in an iterative fashion, in order to approximate a theoretically solid balancing.

To assess the possibilities of DPBM for **Human Agent Substitution**, we will replace players in running multi-player PvP matches with their individual models without notifying the affected or the opposing team. Afterwards, interviews with all participants can uncover a Turing-test-style impression, whether they actually recognized the substitution and in how far it was perceived as too weak/idiosyncratic, too strong/imbalanced, or as a fair representation.

For **Cheating and Botting Detection**, we seek to collect a ground truth of behavior data between players utilizing forbidden methods and tools that yield unfair advantages and regular players (e.g. Aim-/TriggerBots, Keyboard macros, NoClip-/Gravity-/Animation Hacks or Memory manipulation might be reflected in movement and action selection). Looking forward

to find discriminative variables to classify unwanted behavior, we will deploy and evaluate the resulting detection tool in live, real-world scenarios.

### CONCLUSION

We introduced the concept of deep player behavior models (DPBM) in order to analyze, explain, and generate behavior stemming from individual human players in the MMORPG *Lineage II*. Different machine learning techniques were shown to bear different advantages in visualization (Hidden Markov Models; most useful for main skill rotation extraction), analysis (decision trees; most useful for pinning down of skill usage in specific situations; explaining overall usage of particular skills by reversing trees), and performance (deep learning; yielding high accuracy overall and proved to replicate behavior close to the original strategies by human players). Based on the computational models and on verbal reports, we can support the formerly constructed assumptions that players use skills a) in rotations and b) adjusted to specific situations (e.g., own/target HP status), while the adherence to use initial skills remains player-dependent. We also provided a working example of movement behavior approximation, were successful in calculating a difference metric between player behavior, and in replicating play sessions that displayed comparable performance to the modeled players. These exploratory applications establish the potential of DPBM for analysis and for behavior generation that can be beneficial in both academic and industrial use cases.

### ACKNOWLEDGEMENTS

We would like to thank all participants, as well as the private *Lineage II* server *Imoriath* for advertising the study. This work was partially funded by the *German Research Foundation* (DFG) as part of Collaborative Research Center (Sonderforschungsbereich) 1320 EASE - *Everyday Activity Science and Engineering*, University of Bremen (<http://www.ease-crc.org/>), subproject H2 “Mining and Explicating Instructions for Everyday Activities”, as well as by the *German Federal Ministry of Education and Research* (BMBF) through the project *Adaptify*, and the *DAAD FITweltweit* program.

### REFERENCES

1. G. Andrade, G. Ramalho, H. Santana, and V. Corruble. 2005. Challenge-sensitive action selection: an application to game balancing. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*. 194–200. DOI : <http://dx.doi.org/10.1109/IAT.2005.52>
2. Sora Ltd. BANDAI NAMCO Studios Inc. 2014. *Super Smash Bros. for Nintendo 3DS / for Wii U*. Game [WiiU, 3DS]. (13 September 2014). BANDAI NAMCO Studios Inc., Tokyo, Japan. Last played 2017.
3. P. Beau and S. Bakkes. 2016. Automated game balancing of asymmetric video games. In *2016 IEEE Conference on Computational Intelligence and Games (CIG)*. 1–8. DOI : <http://dx.doi.org/10.1109/CIG.2016.7860432>
4. B. Chan, J. Denzinger, D. Gates, K. Loose, and J. Buchanan. 2004. Evolutionary behavior testing of commercial computer games. In *Proceedings of the 2004*

- Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, Vol. 1. 125–132 Vol.1. DOI : <http://dx.doi.org/10.1109/CEC.2004.1330847>
5. Darryl Charles and Michaela Black. 2004. Dynamic Player Modelling: A Framework for Player-centred Digital Games. *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education* (Jan. 2004).
  6. Darryl Charles, Michael Mcneill, Moira Mcalister, Michaela Black, Adrian Moore, Karl Stringer, Julian Kücklich, and Aphra Kerr. 2005. Player-centred game design: Player modelling and adaptive digital games. *Proceedings of DiGRA 2005 Conference: Changing Views - Worlds in Play* (Jan. 2005).
  7. Kuan-Ta Chen, Jih-Wei Jiang, Polly Huang, Hao-Hua Chu, Chin-Laung Lei, and Wen-Chin Chen. 2009. Identifying MMORPG Bots: A Traffic Analysis Approach. *EURASIP J. Adv. Signal Process* 2009 (Jan. 2009), 3:1–3:22. DOI : <http://dx.doi.org/10.1155/2009/797159>
  8. Gustavo Danzi, De Andrade, Hugo Pimentel Santana, André Wilson, Broto Furtado, André Roberto Gouveia, A. Leitão, Geber Lisboa Ramalho, and Luis Freire. 2004. Online Adaptation of Computer Games Agents: A Reinforcement Learning Approach.
  9. A. Drachen, R. Sifa, C. Bauckhage, and C. Thureau. 2012. Guns, swords and data: Clustering of player behavior in computer games in the wild. In *2012 IEEE Conference on Computational Intelligence and Games (CIG)*. 163–170. DOI : <http://dx.doi.org/10.1109/CIG.2012.6374152>
  10. Nintendo EAD. 2014. *Mario Kart 8*. Game [WiiU, Switch]. (29 May 2014). Nintendo EAD, Kyoto, Japan. Last played 2017.
  11. Christoph Eggert, Marc Herrlich, Jan Smeddinck, and Rainer Malaka. 2015. Classification of Player Roles in the Team-Based Multi-player Game Dota 2. In *Entertainment Computing - ICEC 2015*, Konstantinos Chorianopoulos, Monica Divitini, Jannicke Baalsrud Hauge, Letizia Jaccheri, and Rainer Malaka (Eds.). Springer International Publishing, Cham, 112–125.
  12. Roberto Flores Emmett Tomai. 2009. Adapting in-game agent behavior by observation of players using learning behavior trees - Semantic Scholar. (2009).
  13. Blizzard Entertainment. 2015. *Heroes of the Storm*. Game [PC]. (2 June 2015). Blizzard Entertainment, Irvine, California. Last played 2017.
  14. Relic Entertainment. 2013. *Company of Heroes 2*. Game [PC]. (25 June 2013). Relic Entertainment, Vancouver, Canada. Last played 2017.
  15. Julian Togelius Andy Nealen Fernando de Mesentier, Scott J Lee. 2017. AI as Evaluator: Search Driven Playtesting of Modern Board Games - Semantic Scholar. (2017).
  16. Hector Fuster, Xavier Carbonell, Andres Chamarro, and Ursula Oberst. 2013. Interaction with the Game and Motivation among Players of Massively Multiplayer Online Role-Playing Games. *The Spanish journal of psychology* 16 (Nov. 2013), E43. DOI : <http://dx.doi.org/10.1017/sjp.2013.54>
  17. Firaxis Games. 2010. *Civilization V*. Game [PC]. (21 September 2010). Firaxis Games, Hunt Valley, Maryland. Last played 2017.
  18. S. Hilaire, H. c Kim, and C. k Kim. 2010. How to deal with bot scum in MMORPGs?. In *2010 IEEE International Workshop Technical Committee on Communications Quality and Reliability (CQR 2010)*. 1–6. DOI : <http://dx.doi.org/10.1109/CQR.2010.5619911>
  19. K. Hullett, N. Nagappan, E. Schuh, and J. Hopson. 2012. Empirical analysis of user data in game software development. In *Proceedings of the 2012 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. 89–98. DOI : <http://dx.doi.org/10.1145/2372251.2372265>
  20. Robin Hunicke and Vernell Chapman. 2004. AI for Dynamic Difficulty Adjustment in Games.
  21. Disney Interactive. 2014. *Star Wars: Commander*. Game [iOS, Android, Windows Phone, PC]. (21 August 2014). Disney Interactive, Glendale, California. Last played 2017.
  22. Alexander Jaffe, Alex Miller, Erik Andersen, Yun-En Liu, Anna Karlin, and Zoran Popović. 2012. Evaluating Competitive Game Balance with Restricted Play. In *Proceedings of the Eighth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE'12)*. AAAI Press, 26–31. <http://dl.acm.org/citation.cfm?id=3014629.3014635>
  23. Hyungil Kim, Sungwoo Hong, and Juntae Kim. 2005. Detection of Auto Programs for MMORPGs. In *AI 2005: Advances in Artificial Intelligence (Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 1281–1284. DOI : [http://dx.doi.org/10.1007/11589990\\_187](http://dx.doi.org/10.1007/11589990_187)
  24. Koramgame. 2016. *Goddess: Primal Chaos*. Game [iOS, Android]. (2016). Koramgame, Hongkong, China. Last played 2017.
  25. Ryan Leigh, Justin Schonfeld, and Sushil J. Louis. 2008. Using Coevolution to Understand and Validate Game Balance in Continuous Games. In *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation (GECCO '08)*. ACM, New York, NY, USA, 1563–1570. DOI : <http://dx.doi.org/10.1145/1389095.1389394>
  26. Chris Lewis and Noah Wardrip-Fruin. 2010. Mining Game Statistics from Web Services: A World of Warcraft Armory Case Study. In *Proceedings of the Fifth International Conference on the Foundations of Digital Games (FDG '10)*. ACM, New York, NY, USA, 100–107. DOI : <http://dx.doi.org/10.1145/1822348.1822362>

27. Nicholas Liao, Matthew Guzdial, and Mark Riedl. 2017. Deep convolutional player modeling on log and level data. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*. ACM, 41.
28. T. Mahlmann, J. Togelius, and G. N. Yannakakis. 2012. Evolving card sets towards balancing dominion. In *2012 IEEE Congress on Evolutionary Computation*. 1–8. DOI : <http://dx.doi.org/10.1109/CEC.2012.6256441>
29. Joe Marks and Vincent Hom. 2007. Automatic Design of Balanced Board Games. (01 2007).
30. John L. Miller and Jon Crowcroft. 2009. Avatar Movement in World of Warcraft Battlegrounds. In *Proceedings of the 8th Annual Workshop on Network and Systems Support for Games (NetGames '09)*. IEEE Press, Piscataway, NJ, USA, 1:1–1:6. <http://dl.acm.org/citation.cfm?id=1837164.1837166>
31. Y. Mishima, K. Fukuda, and H. Esaki. 2013. An Analysis of Players and Bots Behaviors in MMORPG. In *2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*. 870–876. DOI : <http://dx.doi.org/10.1109/AINA.2013.108>
32. NCsoft. 2003. *Lineage II*. Game [PC]. (1 October 2003). NCsoft, Seongnam, South Korea. Last played April 2018.
33. J. Oh, Z. H. Borbora, D. Sharma, and J. Srivastava. 2013. Bot Detection Based on Social Interactions in MMORPGs. In *2013 International Conference on Social Computing*. 536–543. DOI : <http://dx.doi.org/10.1109/SocialCom.2013.81>
34. J. Oh, Z. H. Borbora, and J. Srivastava. 2012. Automatic Detection of Compromised Accounts in MMORPGs. In *2012 International Conference on Social Informatics*. 222–227. DOI : <http://dx.doi.org/10.1109/SocialInformatics.2012.69>
35. Johannes Pfau, Jan David Smeddinck, and Rainer Malaka. 2017. Automated Game Testing with ICARUS: Intelligent Completion of Adventure Riddles via Unsupervised Solving. In *Extended Abstracts Publication of the Annual Symposium on Computer-Human Interaction in Play (CHI PLAY '17 Extended Abstracts)*. ACM, New York, NY, USA, 153–164. DOI : <http://dx.doi.org/10.1145/3130859.3131439>
36. Robert W Plant and Richard M Ryan. 1985. Intrinsic motivation and the effects of self-consciousness, self-awareness, and ego-involvement: An investigation of internally controlling styles. *Journal of personality* 53, 3 (1985), 435–449.
37. Psyonix. 2015. *Rocket League*. Game [PC, XboxOne, PS4, Switch]. (7 July 2015). Psyonix, Satellite Beach, Florida. Last played 2017.
38. Stefan Radomski and Tim Neubacher. 2015. Formal Verification of Selected Game-Logic Specifications. *on Engineering Interactive Computer Systems with SCXML* (2015), 30.
39. Andrew Rollings and Dave Morris. 2003. Game Architecture and Design. (2003). <https://dl.acm.org/citation.cfm?id=1209229>
40. Richard M Ryan, C Scott Rigby, and Andrew Przybylski. 2006. The motivational pull of video games: A self-determination theory approach. *Motivation and emotion* 30, 4 (2006), 344–360.
41. Hannamari Saarenpää, Hannu Korhonen, and Janne Paavilainen. 2009. Asynchronous Gameplay in Pervasive Multiplayer Mobile Games. In *CHI '09 Extended Abstracts on Human Factors in Computing Systems (CHI EA '09)*. ACM, New York, NY, USA, 4213–4218. DOI : <http://dx.doi.org/10.1145/1520340.1520642>
42. M. Shaker, M. H. Sarhan, O. A. Naameh, N. Shaker, and J. Togelius. 2013. Automatic generation and analysis of physics-based puzzle games. In *2013 IEEE Conference on Computational Intelligence in Games (CIG)*. 1–8. DOI : <http://dx.doi.org/10.1109/CIG.2013.6633633>
43. Kyong Jin Shim, Richa Sharan, and Jaideep Srivastava. 2010. Player Performance Prediction in Massively Multiplayer Online Role-Playing Games (MMORPGs). In *Advances in Knowledge Discovery and Data Mining (Lecture Notes in Computer Science)*. Springer, Berlin, Heidelberg, 71–80. DOI : [http://dx.doi.org/10.1007/978-3-642-13672-6\\_8](http://dx.doi.org/10.1007/978-3-642-13672-6_8)
44. D. Sirlin. 2009. Balancing multiplayer competitive games. Game Developer's Conference 2009, San Francisco, CA.
45. Finnegan Southey, Gang Xiao, Robert C. Holte, Mark Trommelen, and John W. Buchanan. 2005. Semi-Automated Gameplay Analysis by Machine Learning. 123–128.
46. Supercell. 2012. . Game [iOS, Android]. (2 August 2012). Supercell, Helsinki, Finland. Last played 2017.
47. Gabriel Synnaeve and Pierre Bessière. 2011. Bayesian Modeling of a Human MMORPG Player. *AIP Conference Proceedings* 1305, 1 (March 2011), 67–74. DOI : <http://dx.doi.org/10.1063/1.3573658>
48. Ruck Thawonmas, Yoshitaka Kashifuji, and Kuan-Ta Chen. 2008. Detection of MMORPG Bots Based on Behavior Analysis. In *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology (ACE '08)*. ACM, New York, NY, USA, 91–94. DOI : <http://dx.doi.org/10.1145/1501750.1501770>
49. Scott M. Thede and Mary P. Harper. 1999. A Second-order Hidden Markov Model for Part-of-speech Tagging. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics (ACL '99)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 175–182. DOI : <http://dx.doi.org/10.3115/1034678.1034712>
50. Julian Togelius, Renzo De Nardi, and Simon M Lucas. 2006. Making racing fun through player modeling and track evolution. (2006).

51. UNINCONGAME. 2016. *Pokeland Legends*. Game [iOS, Android]. (2016). UNINCONGAME, China. Last played 2017.
52. Valve. 2008. *Left 4 Dead*. Game [PC, Xbox360]. (18 November 2008). Valve, Bellevue, Washington. Last played 2017.
53. Di Wang, Budhitama Subagdja, Ah-Hwee Tan, and Gee-Wah Ng. 2009. Creating human-like autonomous players in real-time first person shooter computer games. In *Proceedings, Twenty-First Annual Conference on Innovative Applications of Artificial Intelligence*. 173–178.
54. Jiyoung Woo, Hwa Jae Choi, and Huy Kang Kim. 2012. An automatic and proactive identity theft detection model in MMORPGs. *Appl. Math* 6 (Jan. 2012), 291S–302S.
55. Steven Woodcock. 2001. Game AI: the state of the art industry 2000-2001. *Game Developer* 8, 8 (2001), 36–44.
56. Georgios N Yannakakis and John Hallam. 2004. Evolving opponents for interesting interactive computer games. *From animals to animats* 8 (2004), 499–508.
57. Georgios N. Yannakakis and Manolis Maragoudakis. 2005. Player Modeling Impact on Player’s Entertainment in Computer Games. In *User Modeling 2005*, Liliana Ardissono, Paul Brna, and Antonija Mitrovic (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 74–78.
58. Georgios N. Yannakakis, Pieter Spronck, Daniele Loiacono, and Elisabeth André. 2013. Player Modeling. In *Artificial and Computational Intelligence in Games*, Simon M. Lucas, Michael Mateas, Mike Preuss, Pieter Spronck, and Julian Togelius (Eds.). Dagstuhl Follow-Ups, Vol. 6. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 45–59. <http://drops.dagstuhl.de/opus/volltexte/2013/4335> DOI: 10.4230/DFU.Vol6.12191.45.
59. Nick Yee. 2006. The Demographics, Motivations, and Derived Experiences of Users of Massively Multi-user Online Graphical Environments. *Presence: Teleoper. Virtual Environ.* 15, 3 (June 2006), 309–329. DOI: <http://dx.doi.org/10.1162/pres.15.3.309>
60. Alexander Zook, Eric Fruchter, and Mark O. Riedl. 2014. Automatic playtesting for game parameter tuning via active learning. In *FDG*.