Predicting Player Behavior in *Tomb Raider: Underworld*

Tobias Mahlmann, Anders Drachen, Julian Togelius, Alessandro Canossa and Georgios N. Yannakakis

***Abstract*—This paper presents the results of an explorative study on predicting aspects of playing behavior for the major commercial title Tomb Raider: Underworld (TRU). Various supervised learning algorithms are trained on a large-scale set of in-game player behavior data, to predict when a player will stop playing the TRU game and, if the player completes the game, how long will it take to do so. Results reveal that linear regression models and other non-linear classification techniques perform well on the tasks and that decision tree learning induces small yet well-performing and informative trees. Moderate performance is achieved from the prediction models, which indicates the complexity of predicting player behavior based on a constrained set of gameplay metrics and the noise existent in the dataset examined, a generic problem in large-scale data collection from millions of remote clients.**

Keywords: Player modeling, supervised learning, classification, Tomb Raider: Underworld

I. Introduction

User-oriented testing is a crucial phase of modern game development with the scope of iteratively enhancing the final game product that will be published [1], [2], [3]. Usually a carefully selected set of subjects, representative of the target audience, as well as professional testers are involved in a labor-intensive procedure testing the games and evaluating the quality of the gaming experience [1], [3]. One of the key components of user-oriented testing both during production and after game launch, is to evaluate if people *play the game as intended* and investigate how gameplay and game design impact the playing experience [1], [2]. The increasing focus on increasing player affordability in digital games [1] - freedom, choice - emphasizes the need for the development of reliable and effective user-testing procedures [2].

Being able to predict certain aspects of gameplay and playing experiences defines a vital component of the user testing procedure within game development [1], [3]. Prediction of playing patterns may rely on both qualitative and quantitative approaches to user testing [2], [4]. This paper examines the latter. Within the last five years, instrumentation data derived from player-game interaction — or *gameplay metrics* as they are referred to in game development — has gained increasing attention in the game industry as a source of detailed information about in-game player behavior [2], comprising detailed numerical data extracted from the interaction of the player with the game [5].

The application of machine learning and data mining on such data, with datasets often in the terabyte scale, and the inference of playing patterns from the data [6] can provide an alternative quantitative approach to and *supplement* traditional qualitative approaches of user- and playability testing [3]. Notably, the application of gameplay metrics permits much larger sample sizes to be used, and the data can potentially be collected outside of the laboratory environment. Furthermore, game metrics are highly detailed, permitting tracking and logging of the second-by-second behavior of players. Understanding patterns of game-playing behavior, and more specifically gameplay aspects such as where players encounter problems with progressing through a game, permits re-engineering of the game design and ensures the enhancement of playing experience.

In this paper, we explore the possibility of predicting particular aspects of playing behavior in the commercial game title *Tomb Raider: Underworld*[1] (TRU) via supervised learning. In particular we attempt to predict when a player will stop playing and, if alternatively the player completes the game, how long will it take the player to do so. The generated predictors are trained on player metrical data of the first two levels of the TRU game.

One of the perennial challenges of game design is to ensure inclusiveness — i.e. that as many different types or classes of players are facilitated in the design. Being able to predict when specific classes of players will stop playing a game is of interest in game development because it assists with locating problematic aspects of game design, i.e. features that hinder different classes or types of players from progressing through specific segments of the game, and ultimately complete the game. The ability to predict completion time for the players who do complete the game is of similar interest. For example, if a particular type of player completes a game very fast, there is a risk of disappointment with the game product. Identifying the different types of completion strategies and accounting for them in the game design is an important element ensuring customer satisfaction.

Earlier work on TRU metrics data has focused on the investigation of dissimilar playing patterns via self-organization in a moderate data set of 1365 players [6]. The experiments presented here are based on a large data set derived from 10000 players. Data was collected via the Square Enix Europe (SQE) Metrics Suite. The data collection process is completely unobtrusive since data was gathered directly via the Xbox Live![2] service, with subjects playing TRU in their natural habitat.

Several features that correspond to various key aspects of playing behavior, are extracted from the data, e.g. information about causes of player deaths. The specific features are

Authors are with the Center for Computer Games Research, IT University of Copenhagen, Rued Langgaards Vej 7, DK-2300 Copenhagen S, Denmark (email: {tmah, drachen, juto, alec, yannakakis}@itu.dk).

[1]http://www.tombraider.com
[2]http://www.xboxlive.com

978-1-4244-6297-1/10/$26.00 ©2010 IEEE

selected so that they incorporate knowledge for the player performance. A carefully selected set of various classification algorithms is employed to 1) predict the number of levels completed (i.e. the level number class) based on those features of play and 2) predict the game playing time of players that completed the TRU game. Our algorithms are tested on two tasks: 1) learn to predict based on playing features of level 1 of the game and 2) based on playing features of level 1 and 2. Results showcase the effectiveness of linear regression techniques as well as nonlinear classification approaches. It also appears that decision tree learning achieves moderate performance but provides a full degree of model expressiveness. Moreover, decision trees showcase that a very small number of playing features is adequate for achieving a moderate classification accuracy.

The findings directly address the industrial need of automated processes that could assist towards identifying dissimilar playing patterns and predicting forthcoming player actions and events. The main arguments supporting the commercial applicability of results include the *large-scale* training dataset consisting of 10000 players; the major commercial game used and the available industrial system for logging the data.

II. Game Metrics Mining

Viewing the mining of game data as a process towards player modeling [7], [8] we can identify few studies in the literature. Quantitative models of players have been built to assist the learning of basic non-player character (NPC) behaviors (e.g. moving, shooting) in Quake II [9], [10], [11]. In those studies self-organizing maps [10], Bayesian networks [11] and neural gas [9] approaches are employed for clustering game-playing samples. Similarly, self-organizing maps have been used for clustering players of the trails (player waypoints) of users playing a simple level exploration game [12]. Missura and Gärtner [13] investigate the use of k-means for clustering player data and support vector machines for predicting dynamic difficulty adjustment in a simple shooter game; data is derived from a small sample of 17 players.

The vast majority of the aforementioned approaches concentrates on a few specific scenarios (e.g. imitate human movement in a particular level of a game) while the game environments investigated are simple test-bed games or simplified versions of commercial games. Moreover, the studies focus on constructing models or predictors of playing behavior based on small-scale player-data collection experiments held in laboratories. Doing so questions the scalability of the obtained performance and leads to the simplification of the learning task — which in turn acts in favor of the learning approach.

Game data mining should consider large-scale data sets (ideally *live* player data sets) if the study wishes to ensure that the findings are representative and scalable. The existence of large-scale data, in turn, addresses the need for efficient and robust algorithms able to classify (or cluster) data successfully. Thawonmas et al. [14] used game metrics from the Massively Multiplayer Online Game (MMOG) *Cabal Online* to establish patterns of behavior among the player base, trying to identify aberrant patterns indicative of computer-controlled agents, i.e. game bots. The approach followed in that paper is based on simple frequency analysis. A similar approach was used to visualize the behavior of online game players in [15]. Ducheneaut and Moore [16] investigated interaction patterns between players in the *Star Wars Galaxies* MMOG utilizing action frequencies to group player behaviors. Conversely, Chen et al. [17] utilized the spatial behavior of avatars to establish models of bot and player behavior. None of the aforementioned studies moves beyond relatively simple statistical methods.

To the best of our knowledge the most related study to this research is the work of Weber and Mateas, mining game metrical data for the prediction of player strategy in the real-time strategy game *Starcraft* [18]. Replays from over 5000 expert players were compared using various classification algorithms for recognizing the player's strategy, and regression algorithms for the task of predicting when specific unit or building types will be produced. In [19] non-negative-matrix factorization is applied to mine 1.6 million images on *World of Warcraft* guilds. That study, however, does not consider live data of playing behavior rather than online player appearances. Our earlier study utilized self-organization for the identification of playing behavior clusters of 1365 TRU players [6].

III. Tomb Raider: Underworld

The Tomb Raider franchise is one of the most established in the digital games industry. The Tomb Raider games, a combination of adventure games and 3D platformers, have been published in different versions on all hardware platforms, including mobile devices, and the current game in the series, *Tomb Raider: Underworld* is the eighth to be published.

The main protagonist of the games, whom the player controls and interacts with the game world through, is Lara Croft. She is designed as a combination between an action heroine and Indiana Jones, who travels to exotic locations and enters forgotten tombs and lairs, solving puzzles and finding ancient treasures at the same time. The Tomb Raider game environments have been 3D from the beginning, and *Tomb Raider: Underworld* (TRU) is no exception. Tomb Raider: Underworld is a 3D platform game and is played in third-person perspective. The players are tasked with solving various navigational puzzles and apply strategic thinking in their navigational behavior (see Fig. 1).

The player faces different types of danger from the game environment and computer-controlled agents operating within it. Falling is an almost continuous risk in the game, and the player also encounters different types of mobile NPC enemies. The environment is also a danger, as it is filled with traps, hazardous substances, fire, etc., which can kill the player. The game consists of seven game levels plus a prologue. Each game level is set to a specific theme, for example *Thailand* or the *Arctic Sea*, subdivided into 71 map units (MU) of varying size.

Fig. 1. A screenshot from Tomb Raider: Underworld level "Thailand". Image is copyright of Crystal Dynamics/Square Enix Europe (2009).

IV. Data collection

The gameplay metrics data were obtained from the Square Enix Europe Metrics (SQE; the former EIDOS) Suite, which contains data from a range of SQE-produced games. The SQE Suite is an instrumentation/telemetry system developed to capture and store game metrics. Gameplay metrics are normally logged as event-based data, and each metric is associated with a range of descriptives (contextual information) such as time stamps, user IDs, IP addresses, etc.

An important aspect of the system is that it delivers live data, i.e. data from people playing these games in their natural habitats. The data collection is completely unobtrusive, providing detailed, quantitative information about how users play games free from any effects or bias imposed by experimental approaches to research [6], [20].

A. Data Preprocessing

The SQE Suite holds data from more than 1.5 million players of TRU. A sample was drawn covering all data collected from a two month period (1st Dec 2008 - 31st Jan 2009), providing records from approximately 203000 players (around 100 GB). The game was launched in November 2008, so the data represent a time period where the game was recently released to the public. The data was imported to dual Microsoft SQL Server databses. Such large data amounts require substantial computing power to analyze, and it was therefore chosen to extract a subsample of 10000 players for an initial study. The 10000 players provide a sample large enough to form the basis for developing analysis methods, while at the same time being manageable in terms of analysis runtime. The only criterion applied to the selection was that players in the sample must have completed the first level of TRU.

In terms of preprocessing, the main challenge was to transpose the data obtained from the Metrics Suite into a format we could use to analyze the data. To identify distinct players it was necessary to collect several messages to reconstruct their progress. The data in the sample were extracted in a series of tables, cleaned and transposed to a single table.

Because TRU was the first game that the Metrics Suite collected data from, there were a number of data cleaning issues such as the recording of negative values, missing timestamps, etc., which made the data cleaning process extensive. The 10000 player sample was also cleaned to remove e.g. instances where players had completed the game and then started playing the game again (approximately 1600 players did this). Additionally, instances where the Metrics Suite had missing data reported for a player from e.g. a specific game level or map unit or similar missing intermediate location times (those that were reported as not having spent any time in one or more locations that are part of a level they have completed), where removed. Missing data is discussed further in the last section of this paper.

B. Extracted features

Based on previous experience with a smaller sample of data from TRU [6], it was chosen to focus on game metrics that relate to the primary game mechanics and play features, as these are the most descriptive of the way TRU is played and how players can interact with the game system. TRU is a 3D platformer, with navigation being a major part of the gameplay, as is solving puzzles and fighting enemies. The features used for the current analysis relate to the core mechanics of the game. Eight categories of features were extracted, at two scales of resolution: **Map Unit** or **Game Level**, giving a total of 674 variables per player. Which resolution scale to use for each feature was chosen depending on the frequency of the specific variable, the distribution of use among the sampled players, the relation to the core game mechanics and its suitability for machine learning. Given the above-mentioned rationale the following features were extracted:

- **Playing time:** The time that each player spent playing the game, T. A total of 8.06 years of playtime were included in the dataset (including the game prologue), with an average playing time of 7.06 hours — with different levels/MUs of TRU taking different amounts of time to complete due to their varying size and/or puzzle difficulty. The total playing time per player varies between 21 minutes and 58.64 hours. The average time taken to complete the entire game was 10.23 hours.
- **Total number of deaths:** The total number of deaths for each player, D. There are 961403 instances of death registered, across all levels/MUs and death causes (96.14 average per player, varying from 0-1343 death events; $\sigma\{D\} = 83$). The death count is dependent on e.g. how much of TRU that a player has played, and the skill of the player.
- **Help-on-Demand:** The number of times help was requested, H. A key feature of TRU is the focus on navigational puzzle solving. A typical puzzle could be a door which requires specific switches to be pressed in order to open. Players need to solve the numerous puzzles in order to progress through the game. In order to avoid player frustration with the puzzles, a

native Help-on-Demand (HoD) system was added to TRU, from which a hint or solution can be requested in relation to puzzles. The sampled data indicate that players generally either request both hints and answers or no help at all for specific puzzles. Both hint and answer requests were therefore aggregated into the H value. A total of 329907 HoD-requests are recorded (32.99 average), this value is also highly dependent on how much of the game a player has played, and the player skill and playstyle.

- **Causes of death:** TRU features a variety of ways in which players can die. The causes of death can be grouped into three categories: Death via enemies (which can be subdivided into ranged- and melee-oriented enemies), from falling or from environmental hazards. Death events caused by game bugs, for example players dying during cinematic encounters, were not included.
  - Enemies (melee), D_m: the number of deaths caused by melee enemies. Those enemies include tigers, panthers, who attack Lara Croft in close combat. Dying from melee enemies comprise 3.03% of the total number of deaths recorded.
  - Enemies (ranged), D_r: the number of deaths caused by NPC enemies who attach using ranged weapons, e.g. mercenary snipers. Dying from ranged enemies comprise 4.14% of the total number of deaths recorded.
  - Environment, D_e: the number of deaths caused by environment-related causes of death such as player drowning, being consumed by fire, or killed in a trap, comprising 29.9% of the total number of deaths across all players.
  - Falling, D_f: the number of deaths caused by falling. This cause of death comprises the 62.92% of all death events making it the dominating way to die in TRU, as would be expected from the game design.

  These numbers vary from those reported in [6], reflecting the different properties of the underlying samples: in that study a sample of 1365 players was used who completed the game, whereas the current sample comprises of 10000 randomly selected players among those who completed the first level. The effect of sampling is seen in e.g. death from opponents only comprising 8.13% in the current dataset, but 28.9% in dataset of [6]. Enemies have a high impact in levels 5 and 6, levels that not all players in the current sample will have reached. Death by environmental causes comprised 13.7% in the earlier study, 29.9% in the current, which is likely again due to the different properties of the two samples. Death by falling is similar however: 57.2% reported in [6] vs. 62.92% in the current sample. Fig. 2 depicts the causes of death in TRU.
- **Adrenalin:** The number of times the adrenalin feature was used, A. This is an advanced gameplay feature of TRU that permits the player to temporarily slow down time while performing special attacks against enemies. When activated, a cursor has to be moved to the head area of the target, which will trigger a headshot event. The players in the sample used the adrenalin feature 72593 times, i.e. 7.26 per player. The use of adrenalin is highly varied between players: between 0 and 304 uses.

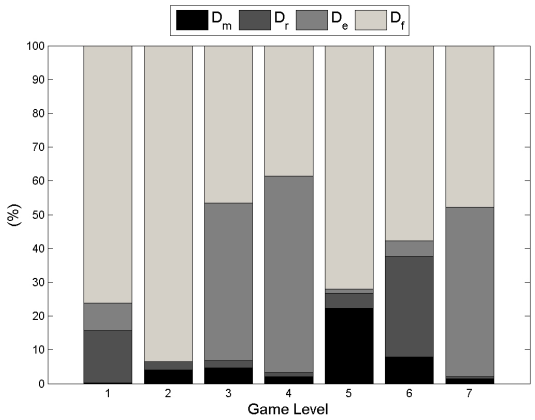

Fig. 2. Percentages of the four causes of death in Tomb Raider: Underworld across all seven game levels. Values are averages of all players (out of the 10000 players) that completed the corresponding level.

- **Rewards:** The number of rewards collected, R. The levels of Tomb Raider: Underworld are rife with ancient artifacts, shards and similar relics, which players have the opportunity to collect during the playing of the game. A total of 1120708 artefacts/shards were located by the players in the game (112.08 average ($\sigma\{R\} = 86.9$).
- **Treasure:** The number of treasures found, T. Most levels in TRU contain one or a few major treasures, which take particular exploration to locate. Thus, a high treasure count is indicative of explorative behavior in players. A total of 24927 treasures are located in the dataset ($\overline{T} = 2.49$; $\sigma\{T\} = 5.1$).
- **Setting changes:** Players can change various parameters of the TRU game. Among these, four directly impact on gameplay, and therefore are of interest to the current analysis:
  - Ammo adjustment, S_a: The number of times the player adjusts how much ammunition Lara Croft is able to carry. Changing this setting comprises 29.6% of the total amount of settings changes.
  - Enemy hit points, S_e: The number of times the player changes the amount of hit points that computer-controlled enemies have, either positively or negatively. Changing this setting comprises 31.5% of the total amount of settings changes.
  - Player hit points, S_p: The number of times the player adjusts how many hit points Lara Croft has, effectively making her harder vs. easier to kill.

Changing this setting comprises 19.5% of the total amount of settings changes.

- Saving grab adjustment, S_s: The number of times the player lowers the recovery time when performing platform jumps, increasing the time available to gain a handhold. Changing this setting comprises 19.4% of the total amount of settings changes.

There were 15317 settings changes made (max 104, 1.53 average); however, only 1740 players changed settings (8.8 average). Settings changes were vastly more common in the first two levels (comprising 34.71% and 37.82% of the changes, respectively), as compared to the later levels (8.02% for level 3, 10.89% for level 4, 4.89% for level 5, 1.21% for level 6, 2.47% for level 7). This pattern possibly reflects the players adjusting the difficulty parameters of the game early on, until they are satisfied, and then use the adjusted parameters throughout the rest of the game.

V. Methodology

After cleaning the 10000 player sample as described above, 6430 players remained. For these players, 30 features were collected relating to the performance of the player on level 1. These were the amount of time, T, spent in 19 different locations of the level (e.g. in the ship engine room and on the surface of the sea), and 11 other features relating to this level only: the number of deaths, the total reward, the number of help requests, the adrenalin used, the number of treasures found, and the number of deaths from the four different causes (melee, ranged weapons, environment, falling, and unknown).

From this set, a second, smaller set consisting of 3517 players who also completed level 2 was selected. For this set, 25 additional features were computed related to gameplay performance on level 2 following the principles of designing the level 1 dataset: the time spent on 14 locations of level 2 plus the 11 gameplay features used in dataset 1. All features are normalized to be in [0, 1] via a uniform distribution.

The target outputs for both data sets is a number indicating the last level completed by the player. We thereby assume there is an unknown underlying function between features of gameplay behavior on the first two levels and the last TRU level that was completed that a classification algorithm will be able to predict.

A third data set was created from the second data set, containing only the 1732 players that finished the whole game, and including the same features as the second data set. This data set was used for trying the predict the time taken to play through the game, assuming that there is some function between early playing behaviour and speed of completion.

To test the possibility of predicting both the TRU level the player completed last, and the time taken to complete game, we apply various classification and prediction algorithms using the *WEKA* machine learning software (version 3.6.2) from the University of Waikato [21]. WEKA is a comprehensive software package that includes versions of all the main prediction and classification algorithms from machine learning, as well as standard algorithms for preprocessing and unsupervised learning and regression techniques from statistics. This version of WEKA contains 76 algorithms applicable to classifying a nominal attribute (the final level played) from a vector of real-valued numeric attributes (the normalized location times, deaths etc. mentioned above) from 8 algorithm families. Somewhat fewer (34 algorithms) can predict a real value (time taken to finish the game) from a real-valued vector. This abundance of tools points to the maturity of the machine learning field, but means that all algorithms and all parameters cannot reasonably be tried on any particular problem.

Given the experimental aim, our approach was to try at least one algorithm from each of the families of algorithms on each dataset, and to spend extra effort on those classification algorithms that were included in the recent list of the most important algorithms in data mining: decision tree induction, backpropagation/multilayer perceptrons and simple regression [22]. Variants of those algorithms were explored and the space of parameters was searched manually. They were also used as components for ensemble classifiers and as subset evaluators for feature subset evaluation algorithms, in order to achieve maximum classification performance. In the following section, we only report the best and most interesting results we have obtained from this experimentation.

For all tested algorithms, the reported classification/prediction accuracy was achieved through 10-fold cross validation.

VI. Experiments

The first two sets of experiments aim to predict the last level finished for each player, based only on features from level 1 and based on features from level 1 and 2 combined. The second set of experiments aims to predict the total time the player took to finish the game, based either on only level 1 features or on both level 1 and 2 features.

A. Last level completed

Before trying to predict which will be the last level a player finishes, we need to establish the baseline accuracy: what would an optimal predictor predict in the absence of any attribute data? This number is equivalent to the number of samples in the most common class (i.e. level completed) divided by the total number of classes. As can be seen from Table I, for the dataset containing all 6430 players that finished level 1, the best guess — in the absence of further information — is that the player only finishes level 1, leading to a baseline prediction accuracy of 34.3%. For the 3571 [1] players that also finished level 2, the best guess is that a player finishes all the levels (last level finished is level 7), yielding a baseline prediction accuracy of 50%.

[1]This number is lower than would be expected by subtracting the players that only finished level 1 from the first dataset ($6430 - 2561 = 3869$) due to extra cleaning that was performed to remove players with missing level 2 features.

TABLE I
NUMBER OF PLAYERS (OUT OF THE 6430 FINISHING THE FIRST LEVEL) THAT STOPPED PLAYING THE GAME ON EACH LEVEL.

Level	1	2	3	4	5	6	7
No. of Players	2561	376	1045	393	56	267	1732

TABLE II
BEST ACCURACY (%) OF SEVERAL CLASSIFICATION ALGORITHMS ON PREDICTING FINAL LEVEL BASED ON FEATURES FROM ONLY LEVEL 1 OR FROM LEVEL 1 AND 2, USING DEFAULT OR LIGHTLY MANUALLY TUNED PARAMETERS. HIGHER VALUES ARE BETTER. NOTE THAT THIS IS JUST A SUBSET OF ALL ALGORITHMS THAT WERE TESTED.

Algorithm	Level 1	Levels 1 and 2
Logistic regression	48.3	77.3
MLP/Backpropagation	47.7	70.2
J48 (C4.5) decision tree (pruned)	48.7	77.4
REPTree decision tree (pruned)	48.5	77.2
Multinomial naive bayes	43.9	50.2
Bayes network	46.7	65.1
SMO Support vector machine	45.9	70.0
Baseline	39.8	45.3

As described above, a number of classification algorithms were brought to bear on the problem of predicting last finished level based on attributes from level 1 or from level 1 and 2. It was found to be easy to do substantially better than baseline accuracy. The best accuracy on predicting final level based on attributes from level 1 was 47.7% (baseline 39.8%), and from attributes from both level 1 and 2 it is 76.9% (baseline 50%).

The best results were found using logistic regression; several algorithms were able to achieve similar accuracy, but none could surpass this simple algorithm. The performance of a selected few algorithms can be seen in Table II.

Most of the tested algorithms had similar levels of performance (with the exception of a few algorithms, especially the Bayesian ones, which underperformed), and were able to predict substantially better than the baseline. In particular, when using features also from level 2, we were able to predict the last level with a much better accuracy than the baseline guess, suggesting that such predictors could be meaningfully used both for analyzing game mechanics and adapting the game online so as to keep the player playing. The difference in the predictive strength of using level 1 and 2 data as compared to only level 1 data is partly due to increased amount of features used in the second case, and of course to the fact that players who stopped playing before finishing level 2 are not part of the second data set. But it is also important to note that level 1 of TRU is designed as a form of "training level", with less varied hazards to the player. The main hazard is falling, which is also evident from the recorded causes of death for level 1 (see Fig. 2). Levels 2-7, while showing substantial variation in theme and design, are more homogenous in that they are varied in their navigational challenges and the challenges the players encounter.

Apart from accuracy, another important advantage of some machine learning algorithms is the transparency and the expressiveness of the acquired model. The models are more useful to a human game designer if they can be expressed in a form which is easy to visualize and comprehend, so that the consequences of changing particular design elements can be easily grasped. Multi-layer perceptrons are particularly limited from this perspective, and linear models with many free variables not so powerful either. However, decision trees of the form constructed by the ID3 algorithm and its many derivatives are excellent from this perspective, especially if pruned to a small size.

The following extremely small decision tree is produced by the REPTree algorithm constrained to tree depth 2, and has a classification accuracy of 47.3% when trained on data from the level 1 only:

L1-Seatop-$T < 10835.5$
$\rightarrow$ L1-$R < 25.5 : 1$
$\rightarrow$ L1-$R \geq 25.5 : 7$
L1-Seatop-$T \geq 10835.5 : 7$

On the set of players who completed both levels 1 and 2, the following tree has a classification accuracy of 76.7%:

L2-$R < 18.5$
$\rightarrow$ L2-Flushtunnel-$T < 9.858 : 2$
$\rightarrow$ L2-Flushtunnel-$T \geq 9.858 : 3$
L2-$R \geq 18.5 : 7$

The right arrow ($\rightarrow$) symbol depicted at the above trees indicates a branch under the tree-node which is right above the symbol. The number right to the colon symbol represents the predicted game level. The accuracy of these predictors is quite impressive given how extremely simple they are. The idea that it would be possible to guess which level a player will finish on much better than baseline, based simply on how long time the player spends on the surface of the sea (L1-Seatop-T; in seconds) in the first level and her total reward (L1-R) during the first level would seem rather outrageous if it was not supported by empirical evidence. The same goes for the idea that we could predict final level with a quite high accuracy based only on the amount of time spent in the Flush Tunnel room (L2-Flushtunnel-T) and the total rewards collected, for level 2 (L2-R).

What these two decision trees indicate is that the amount of time players spent within a given area early in the game and how well they perform is important for determining if they continue playing the game. Time spent can be indicative of problems with progressing through the game, which can lead to frustration. According to these trees the computer-controlled enemies of TRU do not appear to help in predicting when players will stop playing the game.

The fact that only very little performance can be gained from using all 30 (or 55) features rather than just 2 or 3, especially when those 2 or 3 features do not appear to be much more important than other features, suggests

that there is a very high degree of inter-correlation among those features. We, therefore, used the CFS feature subset evaluator [23], which rates a set of features depending on their correlation with the target class and the degree of redundancy between the features, together with a greedy search method (i.e. sequential forward features selection) which starts with adding the most significant feature and then adds one feature at a time until the feature subset cannot be improved . From all 55 features, this method selected only four (L1-Seatop-T, L2-Norsehall-T, L2-R and L2-H) confirming our assumption that the vast majority of features are highly inter-correlated.

B. Completion time

The next set of experiments aims at predicting the time taken to finish the game, based on the same features as above, either from level 1 only or from both levels 1 and 2.

As in the previous set of experiments we tried standard linear regression methods for the prediction of completion time. The feature (of all features from level 1 and 2) that correlates most with completion time is L1-Seafloor-T (positive correlation 0.35) and employing univariate linear regression from this feature to completion time yields an absolute relative error (RAE) of 92%. The RAE statistic is computed as the average difference between the predicted and the target value divided by the difference between the mean and the target value. Multivariate linear regression manages to reduce this error to 88.2% when only using features from level 1, and to 84.5% when using features from both levels 1 and 2 (see Table III).

These linear methods were contrasted a large number of nonlinear methods for numeric prediction from machine learning; selected results are shown in Table III. As can be seen some of the methods (SMO and REPTree combined with bagging) outperformed the linear methods by a notable amount of error. Attribute selection and ensemble classification were tried, as well as moderate parameter tuning, and the results reported in the table reflect the best configuration found for each algorithm (as above, results are only reported for selected algorithms). Like for the classification task, surprisingly poor (sub-baseline) performance was noted from an otherwise reliable algorithm, the MLP using backpropagation. This serves to underscore that experiments like these, which do not perform systematic search in parameter space, can only show that a particular algorithm can work for some type of problem, not that it cannot work.

The features that best predict the time taken to complete the game are unsurprisingly the times taken to complete various units of the first two levels. This can be seen both from which features best correlate with the game completion time, and which features best split the data set into binned classes in the REPTree classifier.

To summarize, we can predict the completion time substantially better than just random guessing, and using features from level 2 as well as well as from level 1 increases the accuracy of our predictions; the best predictor found is the support vector machine achieving a RAE of 82.4%. Given the results obtained the underlying function appears to be nonlinear.

TABLE III
RELATIVE ABSOLUTE ERROR (%) OF SEVERAL ALGORITHMS ON PREDICTING GAME COMPLETION TIME BASED ON FEATURES FROM ONLY LEVEL 1 OR FROM LEVEL 1 AND 2, USING DEFAULT OR LIGHTLY MANUALLY TUNED PARAMETERS. LOWER VALUES ARE BETTER.

Algorithm	Level 1	Levels 1 and 2
Simple linear regression	92.0	92.0
Multivariate linear regression	89.4	84.5
SMO Support vector machine	88.2	82.4
MLP/Backpropagation	107.2	111.5
REPTree decision tree (pruned)	92.5	91.8
Bagging REPTree (pruned)	85.2	83.5
M5Rules decision list	93.7	88.6
Gaussian processes	88.8	84.3
Baseline	100.0	100.0

The question that remains to be answered is exactly how useful these predictions are. Our best predictions still have $4/5$ as high errors as just guessing the average value, meaning that it is unlikely this information would really help in e.g. guiding real-time game adaption, but does provide useful feedback to guide game design. It might be possible to predict outliers – extremely high or low completion times — with higher accuracy, something we have not tried. But our main conclusion regarding completion time is that prediction algorithms are in need of more detailed gameplay metrics and more extracted statistical features.

VII. DISCUSSION

Despite the strong indication that prediction of player behavior based on quantitative measures of their early play performance is possible and the indication that it may be a few features of the behavior of the players that are the most important predictors, the predictive power of the models presented in the above is moderate.

We believe that one of the reasons for the moderate performances achieved in this paper is the existence of data noise both in terms of unreasonable outliers in unit times (which could be generated due to different patches of the game interlocking with the Metrics Suite) and in terms of missing information of players for some game levels. Even though we put substantial effort to remove noise from these large-scale datasets we cannot be entirely certain of the degree of noise that is still existent within those datasets. An additional issue is the limited number of variables available in the TRU-dataset, which do correlate with the core of the gameplay, but lack for example player movement paths. Improving on these points are likely to improve on the predictive strength of the algorithms used here.

In the future it would be our desire to have access to less noisy data via improved logging systems that use even more efficient server-client network communications. The data obtained from Tomb Raider: Underworld was among the first using the — by then — newly developed SQE Metrics Suite, which has since then been further developed.

Data from the newer games, which contain more variables compared to TRU, will form the focus of future research in our attempt to test the generality of the approach followed in this paper.

Future work will also focus on being able to predict when a player stops the game at a finer granularity. Thus, we would like to know not only at what level, but also at which map unit/specific situation the player stops playing. On that basis, supervised learning techniques could potentially perform better if we instead attempt to predict the type of situation in which the player is when she stops playing.

Future research will also investigate association mining, combining clusters of player behavior with gameplay metrics data to investigate if particular play-styles have an impact on game completion and the underlying reasons for why players stop playing before a game is completed. Finally, the 10000 player dataset used in the current study is only a fraction of the main dataset containing data from 203000 players, which in turn is a subset of the main SQE Metrics Suite database which contains data from over 1.5 million players. Future research will focus on testing clustering and classification methodologies on those *massive*-scale datasets.

The causes preventing players from completing a game are possibly game specific and maybe relate to particular playing styles [6], [4], although it is possible that there are principles that apply across specific subsets of games or digital games in general, for example a high difficulty (steep learning curve) early in a game. Ideas about how to keep players engaged are prevalent in the game industry, and increasingly backed by behavioral and cognitive psychology as user research is gaining importance in commercial game development; however, there is very limited publicly available empirical evidence, due to the general proprietary nature of such data. Studies such as the one presented here form a first step towards addressing this problem.

ACKNOWLEDGMENTS

This work would not be possible without the game development companies who are involved. The authors would like to thank their colleagues at Crystal Dynamics and IO Interactive (IOI) for continued assistance with access to the Square Enix Europe Metrics Suite and discussion of approaches, methods and results, including but certainly not limited to: Thomas Hagen and the rest of the Square Enix Europe Online Development Team, Janus Rau Sørensen and the rest of the IOI User-Research Team, Tim Ward, Kim Krogh, Noah Hughes, Jim Blackhurst, Markus Friedl, Thomas Howalt, Anders Nielsen as well as the management of both companies.

REFERENCES

[1] K. Isbister and N. Schaffer, *Game Usability: Advancing the Player Experience*. Morgan Kaufman, 2008.

[2] J. H. Kim, D. V. Gunn, E. Schuh, B. C. Phillips, R. J. Pagulayan, and D. Wixon, "Tracking real-time user experience (true): A comprehensive instrumentation solution for complex systems," in *Proceedings of CHI*, Florence, Italy, 2008, pp. 443–451.

[3] R. J. Pagulayan, K. Keeker, D. Wixon, R. L. Romero, and T. Fuller, *The HCI handbook*. Lawrence Erlbaum Associates, 2003, ch. User-centered design in games, pp. 883–906.

[4] A. Drachen and A. Canossa, "Towards Gameplay Analysis via Gameplay Metrics," in *Proceedings of the 13th MindTrek 2009*. Tampere, Finland: ACM-SIGCHI Publishers, September 2009.

[5] A. Tychsen and A. Canossa, "Defining personas in games using metrics," in *Proceedings of Future Play 2008*. Toronto, Canada: ACM publishers, 2008, pp. 73–80.

[6] A. Drachen, A. Canossa, and G. N. Yannakakis, "Player Modeling using Self-Organization in Tomb Raider: Underworld," in *Proceedings of the IEEE Symposium on Computational Intelligence and Games*. Milan, Italy: IEEE, September 2009, pp. 1–8.

[7] R. Houlette, *Player Modeling for Adaptive Games. AI Game Programming Wisdom II*. Charles River Media, Inc, 2004, pp. 557–566.

[8] D. Charles and M. Black, "Dynamic player modelling: A framework for player-centric digital games," in *Proceedings of the International Conference on Computer Games: Artificial Intelligence, Design and Education*, 2004, pp. 29–35.

[9] C. Thurau, C. Bauckhage, and G. Sagerer, "Learning human-like Movement Behavior for Computer Games," in *From Animals to Animats 8: Proceedings of the 8^{th} International Conference on Simulation of Adaptive Behavior (SAB-04)*, S. Schaal, A. Ijspeert, A. Billard, S. Vijayakumar, J. Hallam, and J.-A. Meyer, Eds. Santa Monica, LA, CA: The MIT Press, July 2004, pp. 315–323.

[10] ——, "Combining self organizing maps and multilayer perceptrons to learn bot-behaviour for a commercial game," in *GAME-ON*, 2003, pp. 119–123.

[11] C. Thurau, T. Paczian, and C. Bauckhage, "Is bayesian imitation learning the route to believable gamebots?" *International Journal of Intelligent Systems Technologies and Applications*, vol. 2, no. 2/3, pp. 284–295, 2007.

[12] R. Thawonmas, M. Kurashige, K. Iizuka, and M. Kantardzic, "Clustering of Online Game Users Based on Their Trails Using Self-organizing Map," in *Proceedings of Entertainment Computing - ICEC 2006*, 2006, pp. 366–369.

[13] O. Missura and T. Gärtner, "Player modeling for intelligent difficulty adjustment," in *Proceedings of the ECML–09 Workshop From Local Patterns to Global Models (LeGo–09)*, J. F. Arno Knobbe, Ed., Bled, Slovenia, September 2009.

[14] R. Thawonmas, Y. Kashifuji, and K.-T. Chen, "Detection of MMORPG Bots Based on Behavior Analysis," in *Proceedings of the 2008 International Conference on Advances in Computer Entertainment Technology (ACE)*. Yokohama, Japan: ACM, 2008, pp. 91–94.

[15] R. Thawonmas and K. Iizuka, "Visualization of online-game players based on their action behaviors," *International Journal of Computer Games Technology*.

[16] N. Ducheneaut and R. J. Moore, "The Social Side of Gaming: A study of interaction patterns in a Massively Multiplayer Online Game," in *Proceedings of the 2004 ACM conference on Computer supported cooperative work*. Chicaco, Illinois: ACM, 2004, pp. 360–369.

[17] H.-K. K. P. H.-H. C. Kuan-Ta Chen, Andrew Liao, "Game Bot Detection Based on Avatar Trajectory," in *Proceedings of the 7th International Conference on Entertainment Computing (ACE)*. ACM, 2008, pp. 94–105.

[18] B. Weber and M. Mateas, "A Data Mining Approach to Strategy Prediction," in *IEEE Symposium on Computational Intelligence in Games (CIG 2009)*, Milan, Italy, September 2009, pp. 140–147.

[19] C. Thurau, K. Kersting, and C. Bauckhage, "Convex non–negative matrix factorization in the wild," in *Proceedings of the 9th IEEE International Conference on Data Mining (ICDM–09)*, W. W. H. Kargupta, Ed., Miami, FL, USA, Dec. 6–9 2009.

[20] R. Rosenthal, "Covert communication in laboratories, classrooms, and the truly real world," *Current Directions in Psychological Science*, vol. 12, no. 5, pp. 151–154, 2003.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explorations*, vol. 11, no. 1, 2009.

[22] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, "Top 10 algorithms in data mining," *Knowl. Inf. Syst.*, vol. 14, no. 1, pp. 1–37, 2007.

[23] M. A. Hall and L. A. Smith, "Practical feature subset selection for machine learning," in *Australian Computer Science Conference*. Springer, 1998, pp. 181–191.