INTL JOURNAL OF ELECTRONICS AND TELECOMMUNICATIONS, 2023, VOL. 69, NO. 3, PP. 461-468
Manuscript received June 15, 2023; revised July, 2023. DOI: 10.24425/ijet.2023.144384

Method for the Player Profiling in the Turn-based Computer Games

Piotr Bilski, Izabella Antoniuk, and Rafał Łabędzki

Abstract—**The following paper presents the players profiling methodology applied to the turn-based computer game in the audience-driven system. The general scope are mobile games where the players compete against each other and are able to tackle challenges presented by the game engine. As the aim of the game producer is to make the gameplay as attractive as possible, the players should be paired in a way that makes their duel the most exciting. This requires the proper player profiling based on their previous games. The paper presents the general structure of the system, the method for extracting information about each duel and storing them in the data vector form and the method for classifying different players through the clustering or predefined category assignment. The obtained results show the applied method is suitable for the simulated data of the gameplay model and clustering of players may be used to effectively group them and pair for the duels.**

Keywords—**turn-based game; player profiling; data clustering; automated classification**

I. Introduction

GAME streaming is becoming a widely recognized phenomenon not only professional gamers but also casual players. Growing variety of streaming platforms, among which Twitch and YouTube are global leaders, extends the range of streaming and attracts constantly growing audience. As a rather new form of social media, game streaming is undergoing intense evolution. One of the paths of this process is the development of the integration between streamers and their audience. Few experiments of such a kind were carried out in last few years, like Twitch Plays Pokémon [1]. Interesting example of facilitating rich interaction between game stream participants is the Cherrystream project, which introduces AI empowered audience voting system.

This paper presents the novel methodology of configuring the computer gameplay on the audience voting, supported by the computational intelligence methods.

The core concept behind Cherrystream is an assumption, that interaction between streamers and their audience can take place in three ways: setting and achieving specific goals in the game, giving and receiving conditional donations, setting and dealing with special conditions in the gameplay like modified weather. The first way is also the most anticipated one, as the standard activity in stream's chat usually regards achieving specific goals in the game. Cherrystream provides tools for the audience, which allow them to easily create goals or goal sequences and present them to the streamer. Achieving those goals is not mandatory for the streamer, but can be used to cheer the audience or receive a conditional donation. Conditions for donation release can be various, from a single goal to complex scenarios. Conditional donations can be placed by a single viewer or a group of viewers, who would like to encourage streamer to reach certain performance targets in the game. The third way of interaction implies potentially crucial consequences to streamer's gameplay, as it introduces tools for modifying game environments. This can cause significant distortion in difficulty level of the game and as such make it too easy or impossible to deal with. As a result, stream can become unattractive to the audience, which is the opposite effect to the desired one, when gameplay environment alteration is set up. Therefore it is required to adjust possible level of gameplay conditions, altering it to the specific streamer's performance. It is achieved by Cherrystream's AI, which maintains a balance between gameplay attractiveness and available interaction level of the audience.

The theoretical scheme of the players and audience behavior is presented in detail. The framework for evaluating the gamer quality and competence is introduced and all its elements described. The framework was tested on the actual competitive logical game for mobile platforms, i.e. MatchUp Friends from Cherrypick Games company The experimental results show the potential of the proposed solution, which has to be further tested on additional set of games.

The paper content is as follows. In section II the state of the art in the online streamed games analysis is presented to show the current methodologies used to describe and analyze players and audience, their behavior and choices. Section III introduces the gameplay configuration system architecture, developed to connect the players with audience and allow selection of challenges for the former by the latter. In section IV the details of methodology for selecting challenges and impediments for the player based on his/her skills and decisions of the spectators are presented. Section V describes the experimental test stand, i.e. the details of the game itself and the framework deployment. In Section VI experimental results are presented, while section VII contains conclusions and future prospects.

II. EXISTING SOLUTIONS

Adjusting gameplay elements to better fit set requirements (i.e. player skill) is not a new topic, ranging from established difficulty levels player can choose from (present in almost every game) to complex methodologies adapting those elements to

Piotr Bilski is with Warsaw University of Technology, Poland (e-mail: piotr.bilski@pw.edu.pl).
Izabella Antoniuk is with Warsaw University of Life Sciences, Poland (e-mail: izabella_antoniuk@sggw.pl).
Rafał Łabędzki is with SGH Warsaw School of Economics, Poland (e-mail: rafal.labedzki@sgh.waw.pl).

specific attributes. At the same time, most of existing solutions focuses either on player actually immersed in computer game, or audience watching his/her performance.

A. Adjusting the player experience

In the first group of solutions main focus remains on the player. Perceived difficulty of the same challenge can vary greatly between players. To ensure, that each person has most rewarding experience, various approaches are adopted.

Number of works considers adapting different game elements to better fit specific player. In [2] authors introduce an adaptive approach for augmenting player satisfaction in real time, with Bug Smasher game as a test platform. In [3] the computational intelligence techniques to build player experience models were used, predicting player impression about the gameplay (i.e. fun, challenge, frustration, boredom) for a platform video game, using Infinite Mario Bros to test their method. In [4] authors design and evaluate an online game adaptation mechanism, that maximizes player's fun and recognizes different playstyles, although their model (based on previous research), achieves worse results for chosen features than previous approach. This work is later extended in [5]. Interesting work [6] considers generating quests based on player actions in game. Authors use components such as memories (obtained both from player and in-game NPC's), attributes, actions and proximity, to generate relevant quests that trace information about player, his/her actions, statistics and relations. Another group of solutions concerns serious games, where adaptation to specific player is important if only to better evaluate his/her progress [7].

Some solutions focus on Dota 2 – multiplayer battle arena game, taking different player and game parameters into consideration. [8] presents an recommendation engine to suggest heroes for the team based on characters picked by opposing group. Different system recommends order in which player should purchase items inside game [9]. In [10] authors evaluate player roles, depending on chosen hero. [11] presents two win predictors, that calculate possible outcome based on heroes selected by both teams. [12] considers AI algorithms for controlling non-player characters (NPCs). For overview and classification of different machine learning methods used for Dota 2 game for various applications see [13].

In [14] a design and requirements for a dynamic difficulty adjustment system were proposed. Also, their applications and possible implementation, using probabilistic techniques were discussed, that will dynamically evaluate and adjust obstacles, based on user performance. Unfortunately, only theoretical work was done, and its main focus remains on First Person Shooter (FPS) type of games.

For survey on present research related to game adaptivity in different fields see [15].

B. Audience participation games

The second group of solutions contains games focused only on audience. In those productions there is no specific player – entire game is directed by all audience members, with different means to achieve required level of control. In [16] authors describe techniques for interactive audience participation in simple games, such as audience movement tracking, object shadow tracking and laser pointer tracking. Illustrated techniques are also evaluated in dedicated games.

In [17] a study performed on CrowdChess game is presented. The application is entirely controlled by audience, without single player responsible for chess moves. Authors analyse different methods for vote counting (i.e. majority vote, expert, leader etc.), while playing against easy and medium difficulty AI, as well as analyse influence each voting method has over game outcome. [18] investigates, how audience of gaming livestreams can influence content. They conducted two case studies, from which first one used pen and paper role playing game as a test platform, while second one concerned game with concept similar to Twitch Plays Pokémon, providing audience with more options to organize their actions.

Some insight might also be drawn from other areas of research, such as recommendation system for videos used by YouTube [19].

III. MatchUp Friends GAME CHARACTERISTICS

The AI-based system proposed in the paper was prepared to analyze users playing the MatchUp Friends game (produced by the Cherrypick Games company) [20]. The game was written for the mobile platforms run under the Android operating system and the implementation of the traditional "memory" gameplay for two players (any of which may be a computer bot). Due to the fact that each player must be logged in to start the game his/her achievements are recorded, it was possible to combine the game engine with the analytical model, which, based on the collected data, would be able to profile players and propose challenges appropriate for them, i.e. with the proper difficulty level.

Fig. 1. Example screens from MatchUp Friends game developed by Cherrypick Games company [21]: initial game screen (left) and example game board during gameplay (right).

The game's goal it to win in PvP turn-based match against the opponent. Players uncover cards with various images, trying to find matching pairs. The game has four base difficulty levels implied by various grid sizes. The more experienced the player is, the bigger grid is available to play. However, the strongest factor influencing the difficulty level is the opponent's skill, which is represented by the ability to memorize previously revealed pictures. Test bots were developed in a way that allowed adjustment of their skill level and therefore various combinations of matchups were possible. Game conditions can be altered by using special boosters, which purpose is to give a hint of matching pair location or to misguide opponent. This is considered as a game environment change in terms of Cherrystream's third way of streamer - audience interaction, which is: setting and dealing with special conditions in the gameplay (for example game screens see Fig.1).

IV. GAMEPLAY CONFIGURATION SYSTEM ARCHITECTURE

The system is based on the assumptions that the player is able to stream the Memory gameplay and solve the particular tasks (challenges), which difficulty increases with time. This way it would be possible to maintain attention of the audience and allow for the bidirectional information flow. The generic architecture of the system is in Fig. 2. Here three modules are important: Players Profiling Module (PPM), Gameplay Control Module (GCM) and Audio-Video Module (AVM). The GCM is responsible for the two-way communication between the players (streamers) and the audience, enabling the video transmission (in one direction) and voting regarding the challenges (in the other). This pap0er focuses on the problem of evaluating the quality of the player, which imposes the analysis of his/her gameplays during the duels with other participants. The correct profiling would allow for the more precise player matching for the duel and selecting the challenges in the more accurate way. Overall, this would lead to the higher audience satisfaction. Although the system was designed for the specific game, it can easily be applied to other turn-based games, where each battle or duel is the discrete event which can be described using features similar to the ones presented in this paper.

The system's task is to connect audience with players and allow the latter to select the particular challenges for the former from the predefined set. It should contain tasks of the similar difficulty, adjusted to the player's capabilities. The GCM is responsible for two operations. The first one is pairing two players in a way that both would represent the similar experience level (which should ensure the maximum gameplay attractiveness). The second task is to select the challenges proposed by the audience and changing the gameplay according to their wishes. The resulting gameplay quality would then be evaluated by the audience, thus allowing for receiving the feedback about the system's accuracy.

The crucial part of the system is the module for profiling players, which is supposed to find and group similar users with the analogous experience. The process is done based on their characteristics, including parameters representing the game records, i.e. the number of games won, the overall number of challenges successfully completed, the size of the board played, etc. These are extracted from each gameplay and can be used for the player profiling. The latter can be performed in two ways. The first one uses the predefined number of 16 categories, generated based on two features, calculated from the gameplay vectors for each player: memory p={1,2,3,4} (the probability to remember the cards revealed during the game) and the manual capabilities m={1,2,3,4}, i.e. the average speed of uncovering the cards. Both parameters are combined, resulting in 16 classes of players, from which the opponents would be selected. The second method is to group similar players based on their vectors, not limiting the number of groups into which the players would be clustered. While the human players may be observed during the long-term interaction with the game, the bots are prepared in advance, differing in particularly adjusted skills to be a worthy opponent for each player. The characteristics of the i-th player's j-th gameplay is defined as the vector:

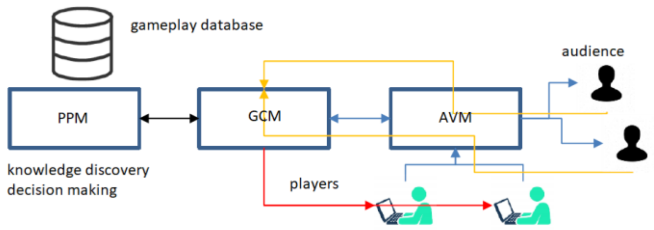

Fig. 2. The intelligent gameplay management systems for online streaming

$$x_{ij} = \left[\frac{t_s}{s}, \frac{r}{s}, s, c_a, c_s, c_c, \frac{2p}{s}, \frac{t_s}{r}\right] \qquad (1)$$

where s is the board size (number of cards, starting from 8x8 to 20x20), t_s is the gameplay duration [s], r is the number of cards' turning, p is the number of points awarded, c_a is the set of challenges available for the player, c_s is the set of challenges selected for the gameplay and c_c is the set of challenges completed during the gameplay. The sets of challenges are represented by the binary arrays where "1" stands for the challenge available, selected or completed, respectively. Vectors (1) are the main source of knowledge about the players' efficiency, thus allowing for their profiling using the unsupervised learning scheme.

The predefined category d of the particular player is calculated from all his/her vectors (1) in the following way:

$$\boldsymbol{p} = \frac{1}{|s|} \cdot \sum_{i=1}^{|s|} \alpha_i \cdot \frac{\bar{t}_{si}}{s_i} \qquad (2)$$

$$\boldsymbol{m} = \frac{1}{|s|} \cdot \sum_{i=1}^{|s|} \beta_i \cdot \frac{\bar{r}_i}{s_i} \qquad (3)$$

where $|s|$ is the number of different board sizes used by the particular player, $\bar{t}_{si}$ and $\bar{r}_i$ are average gameplay duration for the particular board size, and number of the card turns, respectively. The obtained values of $\boldsymbol{p}$ and $\boldsymbol{m}$ are then divided into 4 intervals, based on which the category number d can be inferred. The weighting coefficients α_i and β_i determine the importance of each board size during the classification. In the presented research they were set to equal values, i.e. $\alpha_i = \beta_i = \frac{1}{|s|}$. The proposed method allows for classifying the players to one of 16 categories depending on their skills expressed by (2) and (3) and disregarding the overall numbers of games played and the sizes of boards used during their activities in the game.

The Artificial Intelligence used inside the PPM includes the clustering algorithm, which allows for determining the number of player categories (provided it is not preset to 16). The applied algorithm was k-means [22] with the number of clusters being the result of the optimization procedure (based on the elbow criterion).

V. METHODOLOGY FOR CHALLENGES AND IMPEDIMENTS SELECTION

In this section we present our methodology for evaluating and assigning challenges and impediments for the player. Since presented tasks need to be adjusted both to audience preferences as well as actual player skills, entire process is divided into three repeatable phases: selecting initial challenges with prepared AI algorithm, audience voting on preferred tasks and evaluating player performance.

A. Audience participation games

Before proceeding to task selection, there are few factors related to games and tasks performed by player, that need to be considered.

The first aspect concerns overall tasks available in specific game. Depending from its complexity and structure, each game can contain different challenges that player can perform and impediments to hinder his progress. For each such task, two elements need to be defined: completion conditions (i.e. values for when current challenge is considered a success or a failure) and minimal player skill level (or task difficulty level). Example groups of available tasks based on MatchUp Friends game is presented in Table 1.

The second element is directly related to task difficulty and audience interest. It was assumed, that for game to be interesting, player should be faced witch challenges that are lying on the edge of his capabilities, and at the same time, final outcome of the game should be hard to predict. To achieve that, two elements need to be matched: skill of players facing each other (which needs to be taken into consideration before game begins), and difficulty level of tasks available for audience voting.

To summarize, each game can be described with following phases:

1) Initial task selection based on the player level.
2) Audience voting, selecting active tasks from initial list.
3) Task confirmation/selection by the player (simultaneous with assigning tips by audience).
4) Data collection during the game (depending from game type, task selection can also overlap with that phase).
5) Gameplay evaluation:
   - Player – based on collected data,
   - Game – based on audience questionnaire.
6) Updating player level and task popularity data.

The above six steps are then repeated, resulting in better adjustment of selected elements to both player skill (changing over time) and audience current preferences. In case of MatchUp Friends, all steps are contained inside single game, but it is also possible to extend this solution to other games without well-defined divisions, by adding time frames for each phase.

B. Task selection

At the beginning of each game (or time period set for Real-Time games), audience watching current stream is presented with list of available tasks (both challenges and impediments can be chosen). Elements selected for that list are based on player skill calculated from his previous games. Overall player level is a main factor specifying available tasks that are fitting to his previous statistics concerning different elements of gameplay (i.e. for MatchUp Friends we used factors such as average time to complete board of specific size, average count of cards turned before a pair is matched, average win/loose ratio etc.) – see Table I.

When it comes to difficulty level of single task, there are two things to consider: absolute difficulty level, and relative difficulty matched to player skill. First factor can be easily defined inside each group of tasks (i.e. finishing game on smaller board will be easier than completing it with the same conditions on larger board). For the second element more variables need to be considered. Different players can have different skill levels and the same task might be easy for one person and impossible for someone else. At this point the player's previous performance needs to be taken into consideration. Another issue is increased difficulty level when tasks are combined: i.e. finishing game with set board size might be easy, but when time limit is added, the same task will become more difficult. Therefore, difficulty levels of both selected tasks and task combinations need to be considered.

TABLE I
EXAMPLE CHALLENGES AND IMPEDIMENTS FOR MATCHUP FRIENDS GAME

ID	TYPE	Description	Completion conditions
1	Challenge	Finish board with size NxM.	Finishing game while playing on board with set size.
2	Impediment	Reduce time that the card images stay visible.	Win/finish game while impediment is active.
3	Challenge	Finish game with time under threshold (i.e. 5, 10, 15 minutes).	Win/finish game during set time threshold.
4	Impediment	Finish game with no more then set number of comparisons (i.e. 10, 5, 2).	Win/finish game matching each card without exceeding set threshold of comparisons.
5	Impediment	Finish game while card are moving across the board.	Win/finish game while impediment is active.
6	Challenge	Match N more pairs than opponent.	Win with at least N more pairs matched.
7	Challenge	Win N games in a row.	Win set number of games consecutively.

In our approach we adopted the following methodology. Each player is assigned to single class, calculated from his/her previous statistics. The player class is then used to define both available challenges, and possible opponents (we established maximum difference of 2 classes between players, of overall 16 categories). Each tasks is then assigned with difficulty level dependent on player skill. Total of five difficulty levels were defined in relation to player skill and game attractiveness:

1) d_0 – difficulty of this tasks is far lower than player skill, and poses no challenge – accepting such assignments doesn't increase game attractiveness (or can decrease it if only such challenges are used),

2) d_1 – difficulty of this task equals player skill level – accepting such assignments may (but does not have to) increase game attractiveness,
3) d_2 – difficulty level lying on the edge of players skills – best type of task from game attractiveness point of view (high probability that the task will end successfully and increase audience satisfaction),
4) d_3 – difficulty level lying above player skills – performing single assignments from that group may be possible, but it is not very probable (if task is perform it should increase game attractiveness greatly, but probability that player will fail is high),
5) d_4 – difficulty level far above player skills – evens single tasks are practically impossible to perform for current player.

Each task and task combination is rated according to above specification. Only assignments classified in first 3 groups, with sporadic elements from 4th group (no more than one at a time) will be taken into consideration. Initial tasks are then selected based on two factors: player level and assignment popularity (specifying how often specific task was chosen by viewers). Challenges and impediments from that list are then voted on by the audience.

C. *Player and game evaluation*

The Cherrystream is a system intended to improve communication between streamer and audience, enabling viewers to set different tasks (with possible rewards), as well as evaluate game quality. To achieve best results, two elements are required: ongoing evaluation of player performance, and verification of audience satisfaction. This section covers then former factor.

To follow how well streamer copes with presented challenges, in-game data related to player skill needs to be evaluated (assuming access to them is provided by game designer, otherwise only task completion and time required to finish current task set can be considered). In game supporting Cherrystream system, each task is directly connected to the set of in-game parameters concerning its completion conditions and player data related to it. During time period, when assignments are performed, we gather additional information about how well specific task is performed and compare it to skill level stored in player profile. If gathered statistic are within given threshold, nothing happens, but there are two moments, when task list will be altered: if player level is below given interval (in that case, specific task will be exchanged with the easier one, or removed) or above it (when difficulty level can be increased for specific task, or similar task can be added to available task list). Such approach ensures, that player will not be frustrated with unachievable assignments (for example when he/she has a bad day) or bored with to simple tasks (i.e. after reaching certain skill level). It also ensures, that while he/she progresses through game, learning how to play and achieving higher skill levels, audience will not be bored with the same tasks, or by watching streamer who easily defeats all obstacles.

After this phase is performed, the difficulty level from player point of view, as well as streamer statistics are updated. During the next gameplays the challenges' set will also be modified to better suit the current skills.

VI. EXPERIMENTAL TEST STAND

The presented system was verified based on data extracted from the simulated gameplays where two both have been battling each other. Their parameters and skills were selected randomly, but in a way that after categorizing them, the difference between classes would not be greater than 2 (which ensures the similar quality of both opponents). For each gameplay the set of challenges (as the ones from Table 1) has been selected, simulating the audience voting. After each game for both players the vectors (1) were calculated. They allow for profiling each player by collecting all information about his/her duels, which then can be used to update the competence level (which is used during the opponents selection and during the challenges preparation). All data extracted from games have been saved into the CSV file, which would then be processed by the hybrid Python/Matlab model, allowing for evaluating the obtained profiling results. The general experiment outline is in Fig. 3.

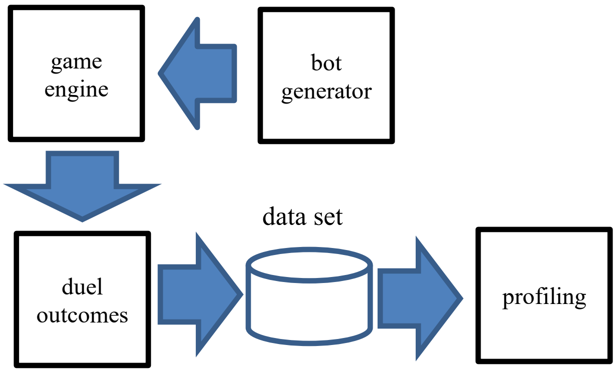

Fig. 3. The architecture of the players profiling module

The collected vectors can be processed in multiple ways, for instance, to extract optimal clusters, but also to evaluate the correctness of the challenges applied to each duel. In the following section analysis of the players' clustering is presented, being the main task in the process of the improvement of the gameplay attractiveness. The player's profile is calculated as the overall vector of the form (1) where features for each gameplay are added. This way it will be possible to calculate distances between different players. Also, these with small experience will be easily distinguishable from the ones spending more time inside the game (as their parameters will be smaller in absolute values). Finally, their features can be used to assign them to one of the predefined 16 classes, later used as the reference value for the clustering experiments.

VII. EXPERIMENTAL RESULTS

This section contains the simulation results for the player profiling scheme. Its aim is to verify if the proposed profiling scheme applied for the system from Fig. 2 is correct. Firstly, the data from the duels are presented to explain, what information has been extracted from the simulator. Secondly, the relation between the ability to successfully complete the particular challenges and the player category is established. Finally, the player clustering using the k means approach are presented.

A. Player statistics

The player bots have been observed for the predefined number of duels and statistics about them collected. Excerpt from the data set is presented in Table 2. Each player (identified by his/her ***pid***) has been disassembled into the corresponding board sizes ***s***, as they constitute the difficulty level of the game. For each board the average gameplay duration $\bar{t}_s$ and its standard deviation σ_t are calculated, as well as the average number of card turns $\bar{c}_t$ during the duel and the corresponding standard deviation σ_c. Values in the subsequent sizes for the selected player show that both the duel durations and number of card turns drastically change with the increase of ***s***. Also standard deviations are large for both features, which is caused by the large variability in the particular games (despite the constant size of the board each gameplay may be very different). This is also confirmed by comparing different players playing on the same board types (like b1 and b2). Differences are significant enough to assign them to different categories during the clustering.

TABLE II
PLAYER PROFILES DEPENDING ON THE SIZE OF THE BOARD PLAYED

pid	***s***	$\bar{t}_s$	σ_t	$\bar{c}_t$	σ_c
b1	64	281.488	242.305	85.485	69.892
b1	100	393.550	256.148	122.100	90.790
b2	64	134.146	67.411	77.263	40.461
b2	100	214.185	199.785	121.906	110.822
b3	100	320.675	219.491	116.739	79.147
b3	256	618.062	55.798	224.690	17.2847
b4	100	153.564	144.197	122.838	115.098
b4	256	281.203	38.960	226.537	18.616
b5	100	233.784	35.531	103.650	14.508
b5	256	521.004	46.602	231.702	14.331
b6	256	804.217	110.786	247.034	32.087
b6	400	1061.248	66.561	327.210	16.316
b7	256	416.311	54.406	237.676	23.841
b7	400	564.613	52.389	323.050	16.151

B. Duel analysis

The players were initially grouped into 16 classes, each depending on their recorded performance during the game with the randomly selected opponent (but differing in the class no more than). Fig. 4 shows the average duel duration depending on the player's classes. In general, the increasing competences lead to the longer game, though it also depends on the particular board generated (the variability is present in the input data in Tab. 2). The similar situation is for the number of card turns, as both boards selected for the game and the difficulty of the challenges increase (Fig. 5).

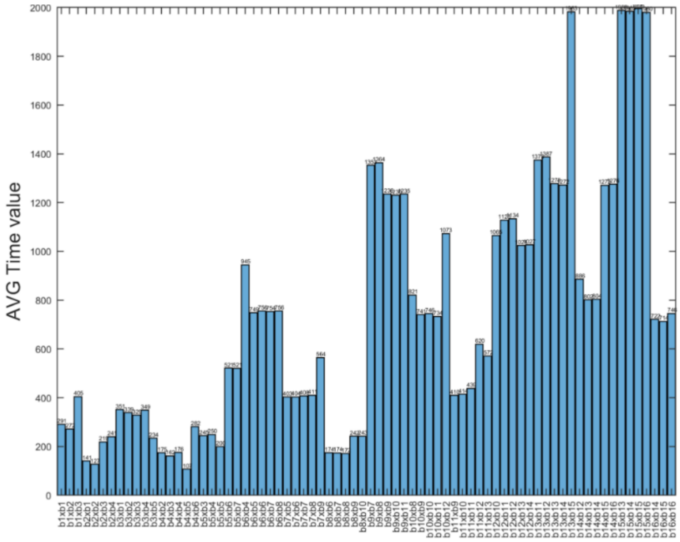

Fig. 4. The average duel duration in relation to the class of the matched players

In both cases there is the clear relation between the game difficulty (represented by the competence levels of players) and the gameplay intensity (measured by the number of card turns). Also, the average time must increase due to the more complex board. These results have to be confirmed for the human players, who may fall into the patterns established by the bots.

C. Clustering outcomes

The k means clustering was applied to the data set presented in section A to group players and obtain their categories. The reference point was the initial categorization to 16 classes based on the agility and memory characteristics assumed initially. The algorithm was repeated multiple times for different clusters created each time. As shown in Fig. 6, the increase in the number of clusters leads to smaller differences in their size. If there are only a few groups, one of them usually dominates over the others.

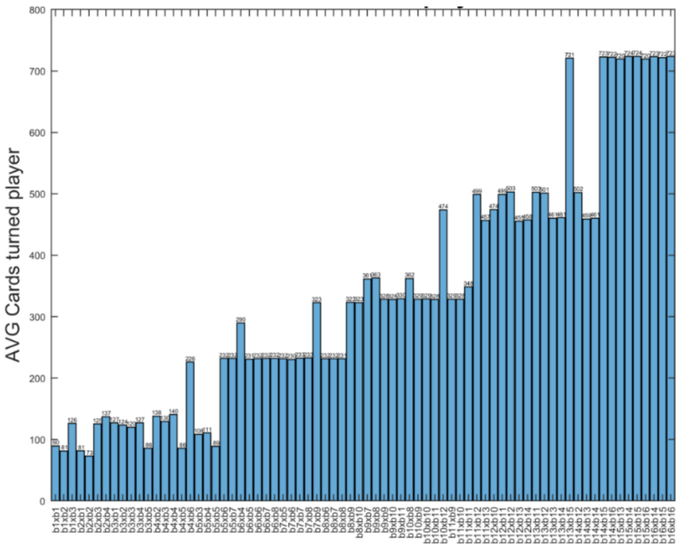

Fig. 5. The average number of card turns in relation to the class of the matched players

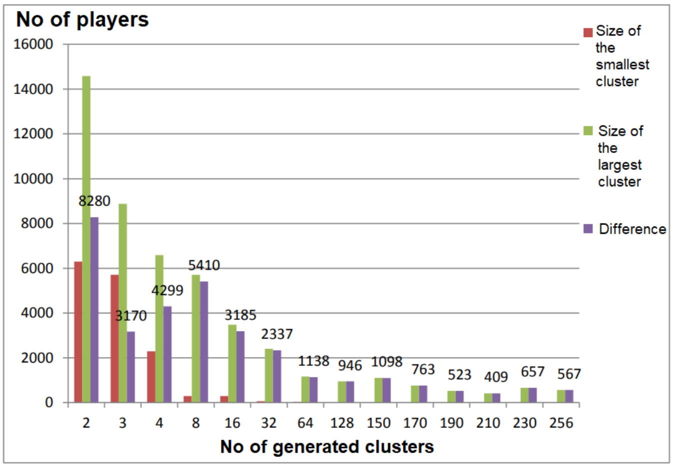

Fig. 6. Clustering density for different number of clusters

The optimal number of clusters may be inferred from Fig. 7, where the distance between the farthest example belonging to the cluster and its center is shown. The elbow curve suggests that the most representative number of groups is 8, which is below the assumed 16. This way the obtained clusters are better balanced (according to other criteria, such as cluster purity or Dunn's index). These results may be used to update the PPM (Fig.2) with the new number of player categories. They must be also confronted against the preestablished categories in the real gameplay environment, to prove which method is more practical.

Another factor to consider is the real gameplay evaluation by the audience. It was assumed that the attractive duel will be ensured by the appropriate selection of the players who should present the similar competence level. It is possible, however, that there are other factors that influence the human perception of the gameplay quality. This should be investigated in the future after collecting feedback from the real audience observing the actual gameplay.

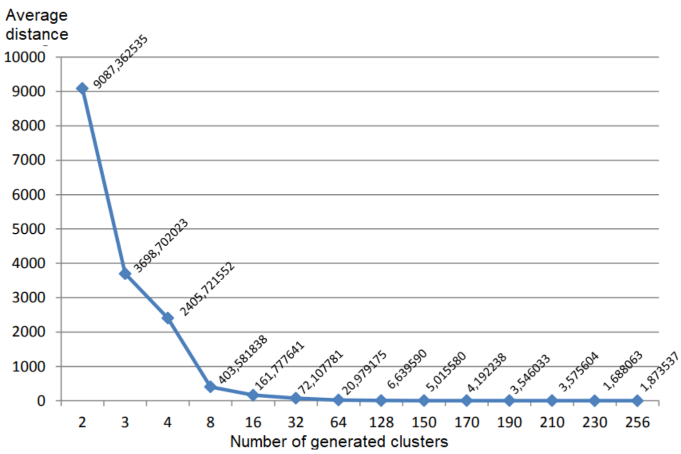

Fig. 7. Clustering quality for different number of clusters

VIII. EXPERIMENTAL RESULTS

The presented results show that profiling the turn-based game is a difficult task, as the variety among players is large, even for the ones classified to the same category. In general it is assumed that the main features describing the player are his/her agility and memory (disregarding the particular game title). The former allows for faster actions on the board, while the latter is responsible for planning these actions in advance.

Because the presented outcomes are from the modeled, simulated gameplay, the important task for the future is to confront them against the real-world gameplay. As the game is currently deployed and used by the players, the measurement data are coming in. After collecting the large enough data set it is intended to repeat the analysis and see if the modeling technique applied to obtain the data presented in the paper is accurate enough.

ACKNOWLEDGEMENTS

Results are related to the implementation of the project no. POIR.01.02.00-00-0114/16, entitled "Cherrystream - an intelligent system to support the creation of games allowing interaction between a streamer, an audience and a game environment", under Measure 1.2 of the Operational Programme Smart Growth 2014-2020 "Sectoral R&D Programmes", co-financed by the European Regional Development Fund.

REFERENCES

[1] Guinness World Records. 2015. Most Participants on a Single-Player Online Videogame. Internet. (2015), http://www.guinnessworldrecords.com/world-records/most-participantson-a-single-player-online-videogame , retr.17/04/2017.

[2] G. N. Yannakakis & J. Hallam, "Real-time game adaptation for optimizing player satisfaction.", in IEEE Transactions on Computational Intelligence and AI in Games, 1(2), pp 121-133, 2009. https://doi.org/10.1109/TCIAIG.2009.2024533

[3] C. Pedersen, J. Togelius & G N. Yannakakis, "Modeling player experience for content creation.", IEEE Transactions on Computational Intelligence and AI in Games, 2(1), pp 54-67, 2010. https://doi.org/10.1109/TCIAIG.2010.2043950

[4] N. Shaker, G. N. Yannakakis & J. Togelius, "Towards Automatic Personalized Content Generation for Platform Games.", in AIIDE, 2010.

[5] N. Shaker, G. N. Yannakakis, J. Togelius, M. Nicolau & M. O'neill, "Evolving Personalized Content for Super Mario Bros Using Grammatical Evolution," in AIIDE, 2012. https://doi.org/10.1609/aiide.v8i1.12501

[6] J. Pita, B. Magerko & S. Brodie, "True story: dynamically generated, contextually linked quests in persistent systems.", in Proceedings of the 2007 conference on Future Play, pp. 145-151, ACM, 2007. https://doi.org/10.1145/1328202.1328228

[7] G. A. Martin, C. E. Hughes, S. Schatz & D. Nicholson, "The use of functional L-systems for scenario generation in serious games.", in Proceedings of the 2010 Workshop on Procedural Content Generation in Games, p. 6, ACM, 2010. https://doi.org/10.1145/1814256.1814262

[8] K. Conley & D. Perry, "How does he saw me? A recommendation engine for picking heroes in Dota 2.", Np, nd Web, 7, 2013.

[9] W. Looi, M. Dhaliwal, R. Alhajj & J. Rokne, "Recommender System for Items in Dota 2.", IEEE Transactions on Computational Intelligence and AI in Games, 2018. https://doi.org/10.1109/TG.2018.2844121

[10] C. Eggert, M. Herrlich, J. Smeddinck & R. Malaka, "Classification of player roles in the team-based multi-player game dota 2.", in International Conference on Entertainment Computing, pp. 112125, Springer, Cham, 2015. https://doi.org/10.1007/978-3-319-24589-8_9

[11] N. Kinkade, L. Jolla & K. Lim, "Dota 2 win prediction.", Technical Report, tech. rep., University of California San Diego, 2015.

[12] M. Waltham & D. Moodley, "An analysis of artificial intelligence techniques in multiplayer online battle arena game environments.", in Proceedings of the Annual Conference of the South African Institute of Computer Scientists and Information Technologists, p. 45, ACM, 2016. https://doi.org/10.1145/2987491.2987513

[13] A. Semenov, P. Romov, K. Neklyudov, D. Yashkov & D. Kireev, “Applications of Machine Learning in Dota 2: Literature Review and Practical Knowledge Sharing.”, 2016.

[14] R. Hunicke, “The case for dynamic difficulty adjustment in games.”, in Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology, pp. 429-433, ACM, 2005. https://doi.org/10.1145/1178477.1178573

[15] R. Lopes & R. Bidarra, “Adaptivity challenges in games and simulations: a survey.”, IEEE Transactions on Computational Intelligence and AI in Games, 3(2), pp 85-99, 2011. https://doi.org/10.1109/TCIAIG.2011.2152841

[16] D. Maynes-Aminzade, R. Pausch & S. Seitz, “Techniques for interactive audience participation.”, in Proceedings of the 4th IEEE International Conference on Multimodal Interfaces, p. 15, IEEE Computer Society, 2002. https://doi.org/10.1109/ICMI.2002.1166962

[17] P. Lessel, A. Vielhauer & A. Krüger, “CrowdChess: A System to Investigate Shared Game Control in Live-Streams.” in Proceedings of the Annual Symposium on Computer-Human Interaction in Play, pp. 389-400, ACM, 2017. https://doi.org/10.1145/3116595.3116597

[18] P. Lessel, M. Mauderer, C. Wolff & A. Krüger, ”Let's Play My Way: Investigating Audience Influence in User-Generated Gaming Live-Streams.”, in Proceedings of the 2017 ACM International Conference on Interactive Experiences for TV and Online Video, pp. 5163, ACM, 2017. https://doi.org/10.1145/3077548.3077556

[19] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi,... & D. Sampath, “The YouTube video recommendation system.”, in Proceedings of the fourth ACM conference on Recommender systems, pp. 293-296, ACM, 2010. https://doi.org/10.1145/1864708.1864770

[20] Cherrypick Games web page: http://cherrypickgames.com/ , (accessed 20.08.2018).

[21] App Store webpage of MatchUp Friends game: https://itunes.apple.com/pl/app/matchup-friends-find-pairs/id1133956564?l=pl&mt=8 (Accesed: 28.08.2018).

[22] A. Drachen, R. Sifa, "Clustering Game Behavior Data," IEEE Transactions on Computational Intelligence and AI in Games, Vol. 7, No. 3, 2015, pp. 266-278, https://doi.org/10.1109/TCIAIG.2014.2376982.