Motion Gaming AI using Time Series Forecasting and Dynamic Difficulty Adjustment

Takahiro Kusano*, Yunshi Liu†
Graduate School of Information Science and Engineering
Ritsumeikan University
Kusatsu, Shiga, Japan
*is0212kf@ed.ritsumei.ac.jp, †is0388ik@ed.ritsumei.ac.jp

Pujana Paliyawan
Research Organization of Science and Technology
Ritsumeikan University
Kusatsu, Shiga, Japan
pujana.p@gmail.com

Ruck Thawonmas‡, Tomohiro Harada§
College of Information Science and Engineering
Ritsumeikan University
Kusatsu, Shiga, Japan
‡ruck@is.ritsumei.ac.jp §harada@ci.ritsumei.ac.jp

***Abstract*—This paper proposes a motion gaming AI that encourages players to use their body parts in a well-balanced manner while promoting their player experience. The proposed AI is an enhanced version of our previous AI in two aspects. First, it uses time series forecasting to more precisely predict what actions the player will perform with respect to its candidate actions, based on which the amount of movement to be produced on each body part of the player against each of such candidates is derived; as in our previous work. the AI selects its action from those candidates with a goal of making the player's movement of their body parts on both sides equal. Second, this AI employs Monte-Carlo tree search that finds candidate actions according to dynamic difficulty adjustment. Our results show that the proposed game AI outperforms our previous AI in terms of the player's body-movement balancedness, enjoyment, engrossment, and personal gratification.**

***Index Terms*—game AI, games for health, motion games, Monte-Carlo tree search, dynamic difficulty adjustment, time series forecasting**

I. Introduction

Full-body motion games are useful for preventing obesity [1] and promoting healthy exercises [2]. However, not much attention has been paid on possible adverse effects [3]. In motion games, repetition movement of few motions or unbalanced use of body parts can cause injuries such as dystonia [4], so there is need for a mechanism that encourages healthy exercises to suppress these adverse effects.

We previously developed a fighting game AI—named Health Promotion AI (HP-AI)—for being used as the opponent AI against human players that stealthy promotes their health during motion gameplay [5]. This AI determines its next action based on prediction on how each of its candidate actions will induce the player to move and how their health will be affected. Given the player's body movement data from UKI [6], the middleware in use, the AI's goal is to promote the balancedness in use of body segments of the player or in other words, to lead the player to use their body parts that are underused. HP-AI's overview is given in Fig. 1 where MCTS stands for Monte-Carlo tree search. However, this AI could not follow change in the players behavior well [7] and did not have any mechanism to adapt its strength to fit the ability of the player.

To overcome the above issues, in this research, we introduce two mechanisms to the above AI. The first mechanism is for more precise prediction of the player's counter actions by using time series forecasting, where four methods of time series forecasting—Auto Regressive Integrated Moving Average (ARIMA), Exponential Smoothing State Space Model (ETS), Naïve forecasting, and forecasting based on the average (Mean)—and the combination of the four methods are examined. The second mechanism is dynamic difficulty adjustment (DDA) to enhance player experience. After descriptions on research background and existing work in Section 2, the proposed AI is described in Section 3, followed by the conducted experiment and results in Section 4.

II. Research Background and Existing Work

A. FightingICE and UKI

FightingICE [8] is an open-source fighting game AI development platform developed for academic research purpose. It has been used as a platform for an annual AI competition by IEEE Conference on Computational Intelligence and Games since 2014, including IEEE Conference on Games 2019. This game covers basic features in the genre of fighting games, and there exist more than 30 actions for each game character.

FightingICE was used by us in several research studies of various themes, not limited to AI development. Some of these studies include a study on design of body motions for motion gaming [9], development of an intelligent assistant for providing instructions and recommending motions to players during full-body motion gaming [10] and development of a universal interface for motion gaming [6]. In addition, studies on the AI development aspect include application of a hybrid-reward deep-learning architecture [11] and investigation of

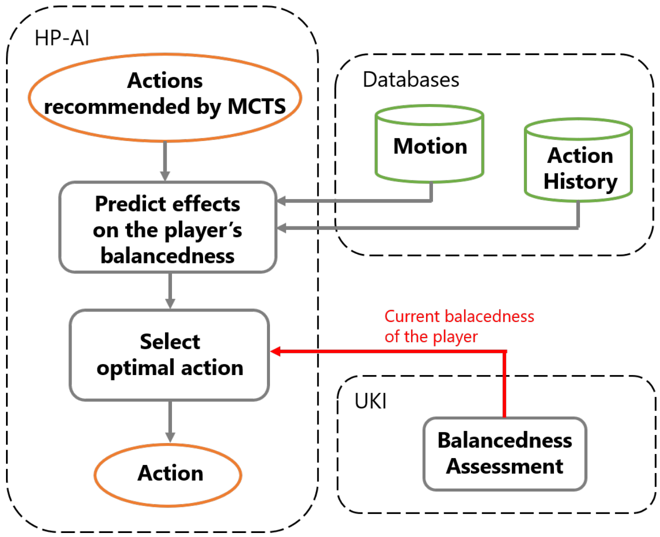

Fig. 1. Overview of HP-AI

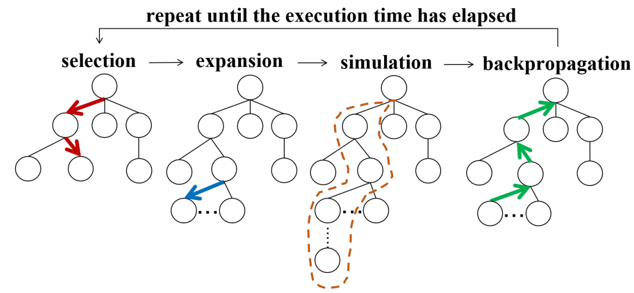

Fig. 2. Overview of MCTS

Kinect-based fighting game AIs that encourage their players to use various skills [12].

FightingICE was also used by many other researchers worldwide, especially on studies related to AI development. For instances, Majchrzak et al. [13] studied on an application of dynamic scripting to a FightingICE agent, and Demediuk et al. [14] developed dynamic difficulty adjustment AI agents. There existed several studies on MCTS AIs; for FightingICE, for examples, Pinto and Coutinho [15] combined hierarchical reinforcement learning with MCTS and Kim and Ahn [16] proposed a hybrid FightingICE AI using a genetic algorithm and MCTS. In addition, Konečný [17] introduced a dynamic 7-dimensional skill-capturing game player model and real-time metrics for modeling fighting-game players using FightingICE as a study platform.

The aforementioned UKI, standing for Universal Kinect-type-controller by ICE Lab (the short name for our laboratory), is the middleware that enables various games on PC to be played by body movement. It converts motions that the player performs in front of Kinect into keyboard input. Therefore, even with commercially available games that are not compatible with Kinect, operation by the whole body becomes possible. In this paper, UKI is applied to FightingICE.

B. Monte-Calro Tree Search (MCTS) and Dynamic Difficulty Adjustment (DDA)

MCTS is a best-first search technique that uses stochastic simulations. As a sample AI for the aforementioned AI competition, Yoshida proposed a fighting game AI, called MctsAi [18], that used open-loop MCTS [19]. Fig. 2 shows an overview of MCTS in general. In this figure, for the open-loop approach, the root node is the current game state and child nodes represent actions while they represent states in standard MCTS. There are four steps in MCTS: selection, expansion, simulation and backpropagation. These four steps are repeated until a given amount of execution time is elapsed. The description of each step is in the following.

1) Selection: select the child nodes with the highest UCB1 from the root node to a leaf node. UCB1 is expressed by

$$UCB1_i = \overline{X}_i + C\sqrt{\frac{2\ln N}{N_i}} \tag{1}$$

In eqn. (1), $\overline{X}_i$ is the average evaluation value of the i-th node (eqns. (2) and (3)), C is a balancing parameter, N_i is the number of visits at the i-th node, and N is the number of visits at the parent node of the i-th node.

$$\overline{X}_i = \frac{1}{N_i}\sum_{j=1}^{N_i} eval_j \tag{2}$$

$$\begin{aligned} eval_j = \quad & \left(afterHP_j^{my} - beforeHP_j^{my}\right) \\ & - \left(afterHP_j^{opp} - beforeHP_j^{opp}\right) \end{aligned} \tag{3}$$

In eqns. (2) and (3), $eval_j$ is the evaluation value of the j-th simulation, $afterHP_j^{my}$ and $beforeHP_j^{my}$ are Hit-Points (HP) of the character after and before the j-th simulation, $afterHP_j^{opp}$ and $beforeHP_j^{opp}$ are those of the opponent character. Accordingly, the more the AI reduces the opponent's HP and maintains its own HP, the higher the evaluation value becomes.

2) Expansion: create all child nodes of the leaf node at once if the number of visits at the leaf node exceeds N_{max}, a threshold, and the depth of the tree at the leaf node is less than D_{max}, another threshold.

3) Simulation: simulate the game for T_{sim} frames using a sequence of actions in the path from the root node to the leaf node as the AI's actions while using random actions as the opponent's actions. Then calculate $eval_j$ at the end of the simulation.

4) Backpropagation: update the UCB1 values of all nodes in the path from bottom to top.

Ishihara et al. proposed an MCTS AI that makes gameplay go within the so-called flow zone [20] by adjusting the strength of the AI according to the abilities of the player [21]. This was achieved by modifying $eval_j$ of the MCTS AI of Yoshida et al. to perform DDA and suppress unnatural behaviors such as intentionally taking damage with no defense at all. In their work, $eval_j$ is calculated as

$$eval_j = (1-\alpha)\,B_j + \alpha E_j \tag{4}$$

In eqn. (4), B_j is a term to suppress unnatural behaviors by promoting the AI's aggressiveness, E_j is a term related to difficulty adjustment, and α dynamically weighs which term should be emphasized. They are given as follows:

$$B_j = \tanh \frac{beforeHP_j^{opp} - afterHP_j^{opp}}{Scale} \tag{5}$$

$$E_j = 1 - \tanh \frac{\left|afterHP_j^{my} - afterHP_j^{opp}\right|}{Scale} \tag{6}$$

$$\alpha = \frac{\tanh\left(\frac{beforeHP_j^{my} - beforeHP_j^{opp}}{Scale} - \beta\right) + 1}{2} \tag{7}$$

In eqns. (5) and (6), $Scale$ is a scaling parameter. In eqn. (5), B_j will be large when the AI gives a high amount of damage to the opponent in the simulation, i.e., when the AI uses aggressive actions. In eqn. (6), E_j will be large when the HP difference between the AI and the opponent after the simulation is close to 0, i.e., when the AI uses actions which bring HP difference toward 0; in other words, when DDA is performed. In eqn. (7), the more the AI is winning against the opponent, the closer α reaches 1; on the contrary, the more the AI is losing, the closer α reaches 0. Thereby, the AI selects actions for difficulty adjustment when it is winning and aggressive actions when it is losing.

III. Proposed AI

The proposed AI, called Dynamic Difficulty Adjustment Health Promotion AI (DDAHP-AI), is aimed at improving HP-AI from the exercise-balance and player-experience perspectives by incorporating dynamic difficulty adjustment and time-series prediction methods into HP-AI. DDAHP-AI selects and executes actions according to the following three steps.

STEP 1

Obtain three candidate actions through MCTS with the evaluation function in eqn. (4), i.e., MCTS that performs DDA, whose use in the health promotion context is the first contribution of this work.

STEP 2

Predict which and how much body parts of the player will move when the AI performs each candidate action. Two types of databases are referred: the Motion database and the Action History database. The Motion database is the predefined database that stores how much each body part moves to perform a motion, assumed to be universal and shared among all players. The Action History database is the database that stores the probabilities of counteractions—one counteraction requires multiple motions—performed by the current player against the AI's actions. Here, the mechanism for the AI to predict the player's body movement against its candidate actions is the same as the one presented in Paliyawan et al.'s work [5], except for the part to calculate the probabilities in the Action History database; this part is new and is the second contribution of this work to be described in the next subsection.

STEP 3

Select the action predicted to have the highest likelihood in increasing the player's body balancedness due to the corresponding counteraction by the player, out of the three candidate actions obtained in STEP 1; this step is the same as the one used in Paliyawan et al.'s work [5].

A. Time Series Forecasting for Predicting Player's Counteraction

In STEP 2, the probability of each counteraction by the player in the Action History database is calculated by time series forecasting using the commonly used methods [22], [23] mentioned in Section 1, i.e., ARIMA, ETS, Naïve forecasting (using the previous value as the predicted one), and Mean (using the average value as the predicted one) used in HP-AI, The respective functions in the R package forecast are used, i.e., *auto.arima*, *ets*, *naive*, and *meanf*, respectively. At the beginning of each round (a round lasts 60 seconds in this work), the best of these four methods is selected according to time series cross validation [24] applied to the data obtained until the end of the previous round from the current player. More specifically, the root mean squared error (RMSE), between actual values and predicted values over all probabilities, of each method is calculated in time series cross validation, and the method with the smallest RMSE is selected. The selected method is called All.

To validate the effectiveness of All, we prepared 9 round data of a player (second-year master's student) playing against MctsAi in a pilot study and compared All with the other four methods. The RMSE of each method for each round from round 2 is shown in Fig. 3 where the x-axis and the y-axis represent the round number and the RMSE, respectively. It can be seen that the RMSE of ALL is the lowest from round 3, which indicates that DDAHP-AI has a higher prediction performance than HP-AI. Note that every method has the same performance at round 2 because when only the first round actual values are available, they all use these values in round 1 as the respective predicted values for round 2, and note that because methods selected for the probabilities of different counter actions are usually different, the RMSE of ALL at a given round is not the lowest value of those of the other four methods.

IV. Experiment

In this section, we describe a two-day experiment conducted to evaluate the proposed AI. Our experiment consisted of a pre-experiment and a main experiment. There were eighteen subjects who were third-year college students to first-year master's students in our university. The protocol of the experiment is shown in Fig. 4.

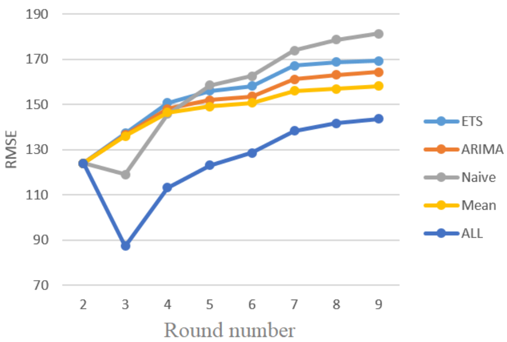

Fig. 3. Comparison of the time series forecasting methods

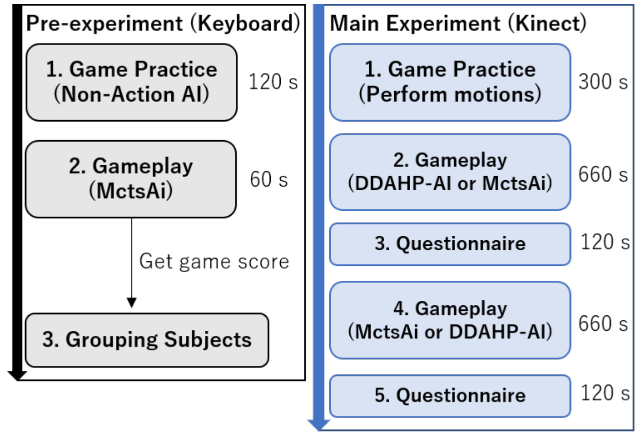

Fig. 4. Protocol of the experiment

Fig. 5. Game screenshot of a subject (P1) playing FightingICE via UKI against DDAHP-AI (P2)

A. Pre-experiment

The objective of this pre-experiment is for subject grouping. In the beginning, we explained the content of the whole experiment to the subjects and asked them to individually sign an informed consent. They were then asked to individually play FightingICE with the keyboard. The first session was for practicing, so they played FightingICE against an AI that did not act at all to get used to the game operation for two rounds. After completing the practice, they played one more round now against MctsAi. Based on the game score against MctsAi, the subjects were divided into two groups such that there was no significant difference between the two groups' average scores according to the Kruskal-Wallis test.

B. Main Experiment

On the second experiment day, the subjects were asked to individually played FightingICE against DDAHP-AI and MctsAi both with Kinect. After a practice session, those who belonged to one of the two groups played FightingICE against DDAHP-AI for three games and then played against MctsAi for another three games while those in the other group did these tasks in the opposite order. In this work, one game consists of three continuous rounds. A one-minute break was employed between each two consecutive games. In addition, after a session[1] against DDAHP-AI or MctsAi, the subjects were asked to answer a questionnaire. Fig. 5 shows a game screenshot of a subject (P1) playing FightingICE via UKI against DDAHP-AI (P2).

The questionnaire in use was based on the Game User Experience Satisfaction Scale (GUESS) [25]. Three factors, i.e., Enjoyment, Engrossment, Personal Gratification (PG), were measured in a 5-Likert scale (1: Strongly Disagree, 2: Disagree, 3: Neither, 4: Agree, 5: Strongly Agree). The evaluation of each factor was based on the value averaged from two questions shown in Table I.

C. Balancedness

We employed Bal, as done in Paliyawan et al.'s work [5], for evaluating balancedness in use of the body segments. To make the paper self-contained, its definition is given in eqn. (8) having the value in the range of [0, 1] with the value of 1 indicating that both sides of the body move in a perfectly balanced fashion.

$$Bal = 1 - 2 \times \frac{\sum_{s=1}^{4} gap_s}{\sum_{s=1}^{4} em_s} \tag{8}$$

where gap_s (eqn. (9)) is the gap between the expected momentum em_s and the actual momentum am_s of the sth segment in four segments of the body: Right Arm, Left Arm, Right Leg, Left Leg. In addition, am_s is an accumulated amount of

TABLE I
CONTENT OF THE QUESTIONNAIRE

Factor	Index	Content
Enjoyment	1	I think the game is fun
	2	I enjoy playing the game
Engrossment	3	I feel detached from the outside world while playing the game
	4	I do not care to check events that are happening in the real world during the game
PG	5	I am in suspense about whether I will succeed in the game
	6	I feel successful when I overcome the obstacles in the game

[1]http://www.ice.ci.ritsumei.ac.jp/%7eruck/ddahp-ai-cog2019.htm

movement of the segment since the current session (DDAHP-AI session or MctsAi session) starts, and em_s is given in eqns. (10) and (11).

$$gap_s = em_s - am_s \tag{9}$$

$$\begin{aligned} em_{RightArm} &= em_{LeftArm} \\ &= max(am_{RightArm}, am_{LeftArm}) \end{aligned} \tag{10}$$

$$\begin{aligned} em_{RightLeg} &= em_{LeftLeg} \\ &= max(am_{RightLeg}, am_{LeftLeg}) \end{aligned} \tag{11}$$

Whether or not DDAHP-AI could encourage the subjects to use the whole body in a well-balanced manner was evaluated by comparing Bal in the DDAHP-AI session and Bal in the MctsAi session.

D. AI configurations used in the experiment

The parameters of both MctsAi and DDAHP-AI were set to the same values as those in relevant previous work ([18], [21]), where they were empirically tuned for their tasks, as shown in Tables II and III, respectively.

TABLE II
MCTSAI'S CONFIGURATION

Notation	Meaning	Value
C	Balancing Parameter	3
N_{max}	Threshold of the number of visits	10
D_{max}	Threshold of the tree depth	2
T_{sim}	The number of simulations	60 frames
T_{max}	Execution time of MCTS	16.5 ms

TABLE III
DDAHP-AI'S CONFIGURATION

Notation	Meaning	Value
C	Balancing Parameter	0.42
N_{max}	Threshold of the number of visits	7
D_{max}	Threshold of the tree depth	3
T_{sim}	The number of simulations	60 frames
T_{max}	Execution time of MCTS	16.5 ms
$Scale$	Scaling parameter	30

V. RESULTS

The experiment results are shown in Table. IV, Figs. 6 and 7. It can be seen that the proposed AI surpasses MctsAi in terms of both mean and median of Bal and the mean of enjoyment, engrossment, and PG. Therefore, the proposed AI is more effective in encouraging players to use their whole body in a well-balanced manner and improving their play experience.

VI. CONCLUSIONS

We proposed a motion gaming AI with DDA and time series forecasting to encourage players to exercise in a well-balanced fashion in use of their body segments and to improve their play experience. Our results indicate that the proposed AI can increase not only balancedness of the body, but enjoyment, engrossment, and personal gratification during the game. Our future work includes adding functions to control the amount of calories burned and to analyze exercise intensity.

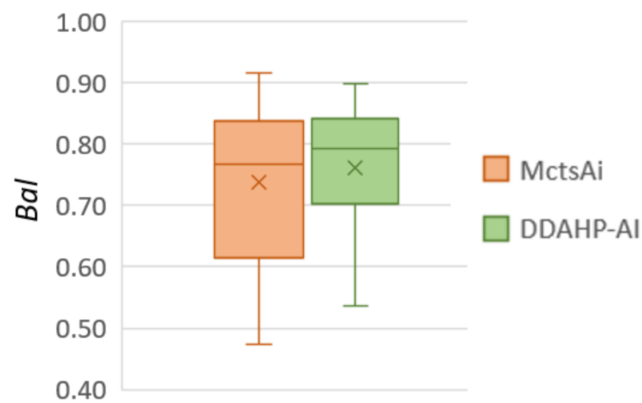

Fig. 6. Evaluation of balancedness (Bal).

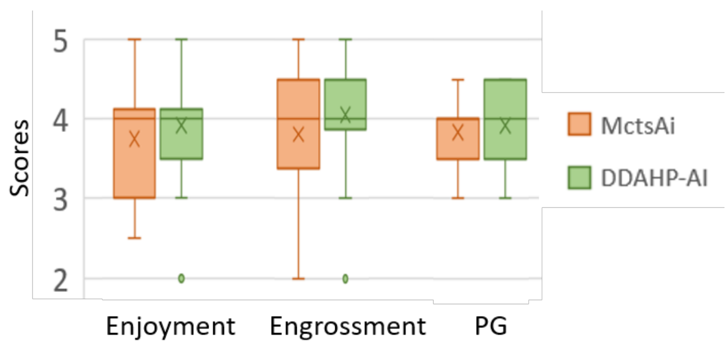

Fig. 7. Evaluation of Enjoyment, Engrossment, and Personal Gratification (PG) in the questionnaire.

ACKNOWLEDGEMENT

This research was supported in part by Strategic Research Foundation Grant-aided Project for Private Universities (S1511026), Japan, and by Grant-in-Aid for Scientific Research (C), Number 19K12291, Japan Society for the Promotion of Science, Japan.

REFERENCES

[1] Staiano, A. E., Abraham, A. A., and Calvert, S. L.: *Adolescent exergame play for weight loss and psychosocial improvement: a controlled physical activity intervention*, Obesity, 21(3), pp. 598-601, (2013).

[2] Staiano, A. E., Abraham, A. A., and Calvert, S. L.: *Motivating effects of cooperative exergame play for overweight and obese adolescents*, Journal of Diabetes Science and Technology, 6(4), pp. 812-819, (2012).

[3] Institute of Digital Media and Child Development Working Group on Games for Health.: *Games for health for children-Current status and needed research*, Games for health journal, 5(1), pp. 1-12, (2016).

[4] Cutsforth-Gregory, J. K., Ahlskog, J. E., McKeon, A., Burnett, M. S., Matsumoto, J. Y., Hassan, A., and Bower, J. H.: *Repetitive exercise dystonia: a difficult to treat hazard of runner and non-runner athletes* Parkinsonism and related disorders, Vol. 27, pp. 74-80, (2016).

[5] Paliyawan, P., Kusano, T., Nakagawa, Y., Harada, T., and Thawonmas, R.: *Health Promotion AI for Full-body Motion Gaming*, Proc. of the AAAI Spring Symposium Series (Well-Being AI: From Machine Learning to Subjective Oriented Computing), pp. 720-725, (2017).

[6] Paliyawan, P. and Thawonmas, R.: *UKI: universal Kinecttype controller by ICE Lab*, Software: Practice and Experience, 47(10), pp. 1343-1363, (2017).

TABLE IV
COMPARISON OF BALANCE, ENGROSSMENT, ENJOYMENT, AND PERSONAL GRATIFICATION (PG) BETWEEN THE TWO AIS

	Balance	Engrossment	Enjoyment	PG
MCTS-AI	0.738 ± 0.133	3.806 ± 0.788	3.750 ± 0.670	3.833 ± 0.383
DDAHP-AI	0.762 ± 0.103	4.056 ± 0.745	3.917 ± 0.691	3.917 ± 0.642

[7] Kusano, T., Paliyawan, P., Harada, T. and Thawonmas, R.: *Analysis of Relationship Between the Player's Behavior Change and the Effectiveness of a Health Promotion AI*, Proc. of the 2017 NICOGRAPH International (NicoInt 2017), pp. 92, (2017).

[8] Lu, F., Yamamoto, K., Nomura, L. H., Mizuno, S., Lee, Y. and Thawonmas R.: *Fighting game artificial intelligence competition platform*, Proc. of the 2nd IEEE Global Conference on Consumer Electronics (GCCE 2013), pp. 320-323, (2013).

[9] Paliyawan, P., Sookhanaphibarn, K., Choensawat, W., and Thawonmas, R.: *Body Motion Design and Analysis for Fighting Game Interface*, Proc. of the 2015 IEEE Computational Intelligence and Games (CIG 2015), pp. 360-367, (2015).

[10] Paliyawan, P., Kusano, T., and Thawonmas, R.: *Motion Recommender for Preventing Injuries During Motion Gaming*, IEEE Access, 12 pages, doi:10.1109/ACCESS.2018.2889166 (2018)

[11] Takano, Y., Ouyang, W., Ito, S., Harada, T., and Thawonmas R.: *Applying Hybrid Reward Architecture to a Fighting Game AI*, Proc. of the 2018 IEEE Conference on Computational Intelligence and Games (CIG 2018), pp. 433-436, (2018).

[12] Ishihara, M., Miyazaki, T., Paliyawan, P., Chu, C.Y., Harada, T. and Thawonmas, R.: *Investigating Kinect-based fighting game AIs that encourage their players to use various skills*, Proc. of the IEEE 4th Global Conference on Consumer Electronics (GCCE 2015), pp. 334-335, (2015).

[13] Majchrzak, K., Quadflieg, J., and Rudolph, G.: *Advanced dynamic scripting for fighting game AI*, Proc. of the 14th International Conference on Entertainment Computing (ICEC 2015), pp. 86-99, (2015).

[14] Demediuk, S., Tamassia, M., Raffe, W, L., Zambetta, F., Li, X., and Mueller, F, F.: *Monte Carlo Tree Search Based Algorithms for Dynamic Difficulty Adjustment*, IEEE the 2017 IEEE Conference on Computational Intelligence and Games (CIG 2017), pp. 53-59, (2017).

[15] Pinto I, P., and Coutinho, L, R.: *Hierarchical Reinforcement Learning with Monte Carlo Tree Search in Computer Fighting Game*, IEEE Transactions on Games, 6 pages, doi:10.1109/TG.2018.2846028 (2018).

[16] Kim, M, J., and Ahn, C, W.: *Hybrid fighting game AI using a genetic algorithm and Monte Carlo tree search*, Proc. of the Genetic and Evolutionary Computation Conference 2018 (GECCO 2018) Companion, pp. 129-130, (2018).

[17] Konečný, R.; *Modeling of fighting game players*, Master's Thesis, Utrecht University, (2016).

[18] Yoshida, S., Ishihara, M., Miyazaki, T., Nakagawa, H., Harada T. and Thawonmas, R.: *Application of Monte-Carlo Tree Search in a fighting game AI*, Proc. of the 5th IEEE Global Conference on Consumer Electronics (GCCE 2016), pp. 623-624, (2016).

[19] Liebana, D.P., Dieskau, J., Hunermund, M., Mostaghim, S., Lucas, S.: *Open Loop Search for General Video Game Playing*, Proc of the Genetic and Evolutionary Computation Conference 2015 (GECCO 2015), pp. 337-344, (2015).

[20] Chen, J.: *Flow in games (and everything else)*, Communications of the ACM, 50(4), pp. 31-34, (2007).

[21] Ishihara, M., Ito, S., Ishii, R., Harada, T., and Thawonmas, R.: *Monte-Carlo Tree Search for Implementation of Dynamic Difficulty Adjustment Fighting Game AIs Having Believable Behaviors*, Proc of the 2018 IEEE Conference on Computational Intelligence and Games (CIG 2018), pp. 46-53, (2018).

[22] Hyndman, R. J., and Athanasopoulos, G.: *Forecasting: principles and practice*, OTexts, (2018).

[23] Baldan, F. J., Ramirez-Gallego, S., Bergmeir, C., Benitez-Sanchez, J. M., and Herrera, F.: *A Forecasting Methodology for Workload Forecasting in Cloud Systems*, IEEE Transactions on Cloud Computing, 6(4), pp. 929-941, (2016).

[24] Hyndman, R. J.: *Cross-validation for time series*, https://robjhyndman.com/hyndsight/tscv/ (last accessed on May 30, 2019).

[25] Phan, M. H., Keebler, J. R., and Chaparro, B. S.: *The development and validation of the game user experience satisfaction scale (GUESS)*, Human factors, 58(8), pp. 1217-1247, (2016).