A Fighting Game AI Using Highlight Cues for Generation of Entertaining Gameplay

Ryota Ishii
Graduate School of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
is0245ve@ed.ritsumei.ac.jp

Suguru Ito
Graduate School of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
is0202iv@ed.ritsumei.ac.jp

Ruck Thawonmas
College of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
ruck@is.ritsumei.ac.jp

Tomohiro Harada
College of Information Science and Engineering
Ritsumeikan University
Shiga, Japan
harada@ci.ritsumei.ac.jp

***Abstract*—In this paper, we propose a fighting game AI that selects its actions from the perspective of highlight generation using Monte-Carlo tree search (MCTS) with three highlight cues in the evaluation function. The proposed AI is targeted for being used to generate gameplay in live streaming platforms such as Twitch and YouTube where a large number of spectators watch gameplay to entertain themselves. Our results in a user study conducted using FightingICE, a fighting game platform used in an international game AI competition since 2013, show that gameplay generated by the proposed AI is more entertaining than that by a typical MCTS AI. Detailed analyses of gameplay from all the methods assessed in the user study are also given in the paper.**

***Index Terms*—Monte-Carlo tree search, live streaming, highlight generation, fighting game AI, FightingICE**

I. Introduction

In recent years, live streaming platforms such as Twitch and YouTube are popular. A large group of spectators is "Let's Play" [1] who watch gameplay videos to entertain themselves. As a results, this type of spectators has gained a lot of interests by researchers in various areas.

Recently, Thawonmas and Harada in our group proposed a concept called procedural play generation (PPG) [2]. Their goal is to automatically generate gameplay according to spectators' preferences. PPG requires a system that analyzes and recommends gameplay and a mechanism or AI that generates various kinds of gameplay which entertain different types of spectators. In our previous work on PPG, a method was proposed using Monte-Carlo tree search (MCTS) [3-4] to generate playing styles [5] in a fighting game. We also focus on the AI part for the fighting game genre in this work.

In this paper, inspired by existing highlight generation methods that select exciting scenes for sports spectators, we propose a fighting game AI for generating entertaining gameplay where a combination of highlight indicators or cues is used in the evaluation function of MCTS. We show that gameplay generated by the proposed method is more entertaining than gameplay by a standard MCTS in a user study conducted on a fighting game platform called FightingICE [6], used in international game AI competitions since 2013 including one at CoG 2019. The proposed idea of introducing highlight cues in the evaluation function of MCTS for generation of entertaining gameplay can be applied to other games provided that game forward models are available and cues are modified accordingly.

978-1-7281-1884-0/19/$31.00 ©2019 IEEE

II. Related Work

A. Monte-Carlo Tree Search in Fighting Games

Although AIs using deep learning techniques [7-9] can be recently seen, MCTS, combining a Monte-Carlo method and game tree search using a given forward model, is a popular technique to implement a fighting game AI. A sample AI using MCTS for FightingICE [10] adopted the open loop approach [11]. In this work, we follow this recipe.

Figure 1 shows an overview of MCTS. In the open-loop approach, a node represents an action, except for the root node representing the current game state defined by information such as the Hit-Point (HP), energy, coordinates, and action of each character and the game remaining time. An edge represents the ongoing execution of an action of interest. Four steps exist: selection, expansion, simulation, and backpropagation. They are described in the following, respectively.

1) Selection: Nodes are selected from the root node until a leaf node is reached according to a selection criterion in use. We use Upper Confidence Bounds ($UCB1$) [12], which is widely used for this task, defined by the following equation:

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

where N_i is the number of times node (action) i has been visited, N is the number of visits to its parent node, and C is a constant. In addition, $\overline{X}_i$ is the average evaluation value of node i:

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

where $Eval_j$ is an evaluation function returning a reward value gained in the jth simulation from the perspective of the AI player. Note that every node of the tree contains the $UCB1$ value and a counter counting how many times it has been visited. In this work, the selected path is the one that contains

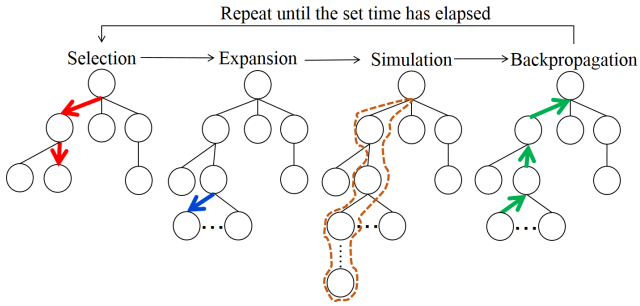


Fig. 1. An overview of MCTS for a fighting game

the nodes with the highest $UCB1$ value, from the root node until a leaf node.

2) Expansion: After a leaf node has been reached in the $Selection$ step, if the number of times it has been explored exceeds a threshold N_{max} and the depth of the tree is lower than a threshold D_{max}, all of its child nodes are created at once from it. Note that the initial tree consists of the root node and all of its direct child nodes.

3) Simulation: A simulation of the length of T_{sim} is performed by sequentially executing all the actions in the selected path while the opponent's actions are chosen for execution at random. If T_{sim} has not passed yet after those actions of the AI player have been executed, a rollout will be carried out until T_{sim} runs out using randomly selected actions. At the end of each simulation, the character's $Eval_j$ is calculated.

4) Backpropagation: The value of $Eval_j$ obtained in the $Simulation$ step is back-propagated from the leaf node to the root node during which the $UCB1$ value at each node along the path is also updated accordingly.

MCTS repeats the above four steps until a given time limit, T_{max}, is reached. An action is then chosen among all of the direct child nodes of the root node as the next action according to a given recommendation policy. In this work, it is action i^* that has the highest $\overline{X}_i$. As done in the open-loop approach, the chosen child node will be used as the next root node, now representing the state after the action is executed, and its sibling nodes will be pruned.

B. Highlight Generation

With a rapid increase in sports broadcasting, it is necessary to generate a highlight that allows audiences to see exciting scenes at their convenient timing. However, manual generation of highlights is time consuming, so a number of automatic methods for generating highlights have been proposed. Typically, scenes are evaluated based on several cues, and scenes with high evaluation values are selected for a highlight. For example, for boxing, a highlight can be generated based on the camera-flash timing and the distance between both players [13], i.e., a set of scenes with a close distance during flash light becomes a highlight.

Recently, a method was proposed that generates a highlight for basketball based on five cues [14]: "Audio," "Score Differential," "Player Ranking," "Basket Type," and "Motion," described in the following. Audio assesses scenes according to the loudness of spectators and commentators. Score Differential considers that scenes with a narrow gap in score near the end of the game are exciting. Player Ranking selects scenes where a shot is done by a high-ranking player. Basket Type ranks scenes according to their scoring shot types. We select and modify certain cues from these two previous studies for our work; worth mentioning is more recent work by Ringer and Nicolaou [15] that takes into account information on streamers' face and audio, which cannot be applied to our work.

III. Proposed Method

In this section, we describe our proposed method for generating gameplay to entertain spectators. Our research hypothesis is that a fight between two AI players would be exciting if each AI player selects their actions based on highlight cues, which in theory would make their fight a highlight from the beginning to the end. In this work, MCTS is used, and three highlight cues are introduced to form the evaluation function of MCTS: "Score Transition," "Action," and "Distance." All of them are expected to increase aggression in gameplay, which is what spectators demand in both traditional sports [16] and esports [17]. The details of the proposed method are given below.

A. Score Transition

Score Transition is a term that combines the elapsed time of the current round and the predicted damage value that the AI player gives to its opponent. This term, $Score$, is defined as follows:

$$Score = RoundTime \times Damage \tag{3}$$

where $RoundTime$ and $Damage$ represent the elapsed fight time and the damage value at the end of the current simulation, respectively. According to this term, the AI player prioritizes actions with high damage values, and this kind of an earnest fight is more prominent as the time is closer to the end of the round. The term is based on Score Differential [14], but has been adapted to the fighting game accordingly.

B. Action

Action prioritizes certain actions by the AI player and is defined as follows:

$$Action = \begin{cases} \frac{1}{2^{Rank-1}} & (belongs\ to\ RankAct) \\ 0 & (otherwise) \end{cases} \tag{4}$$

where $RankAct$ is a list of actions, and $Rank$ is the value associated to each action in the list. The list of actions and their rank are shown in Table I. In previous work [14], basket shots (dunk, three-point jumper, etc.) that have a high degree of excitement were selected to form the Basket Type cue. In this work, the actions in the list are those we consider have high visual effects. The value of an action that is not in the list is 0.

TABLE I
RankAct LIST IN FIGHTINGICE

RankAct	Skill content	*Rank*
STAND_D_DF_FC	Special Skill	1
STAND_F_D_DFB	Strong Upper	2
STAND_D_DB_BB	Sliding Kick	3
STAND_D_DF_FB	Shoot Strong Projectile Forward	4

C. Distance

Distance prioritizes a close-distance fight near the center of the screen and is defined as follows:

$$Distance = 1 - \left| \frac{center - Xpos}{center} \right| \tag{5}$$

where $center$ is the x coordinate at the center of the screen, and $Xpos$ is the AI player's x coordinate. $Distance$ will have a higher value when the AI player is positioned closer to the center of the screen. Note that if the distance between both AI players is only used, they might end up fighting at either edge of the screen, which we consider less exciting.

D. Evaluation Function

Finally, the evaluation function of MCTS in Eqn. (2) is concretely defined as follows:

$$Eval_j = \omega_s Score + \omega_a Action + \omega_d Distance \tag{6}$$

where $Score$, $Action$, and $Distance$ are normalized here to the range of 0 and 1, and their weights ω_s, ω_a, ω_d are discussed in the next section.

IV. EXPERIMENTS

In this section, we describe experiments conducted to verify the performance of the proposed method. We compare a number of AIs, which use a different combination of the three cues described above, with a typical MCTS AI (mAI) [10], which is a sample MCTS AI for the aforementioned competition using FightingICE where the difference in the HP decrease between the player AI and its opponent is used in the evaluation function. Results from a user study and gameplay analysis are given.

A. Environment

1) FightingICE: FightingICE is a real-time 2D fighting game platform used in a game AI competition (FTGAIC)[1] and for research [18-24, 26]. FightingICE has been originally developed from scratch without using a ROM emulator and publicly made available. As a result, use of this platform imposes no infringement issues to be concerned, in particular for places where there are no explicit law doctrines similar to $fair\ use$ in the US.

In FightingICE, a round lasts 60 seconds, and the game is rendered 60 frames per second. Due to receiving a delayed game state from the system, an AI player does not precisely know the timing that it can perform its next action and hence it has to decide and input an action every frame, by which the previous action awaiting execution will be overridden. The HP for both characters is initially set to HP_{max} and decreases when a character of interest is hit. A round ends when the fight is conducted for 60 seconds or the HP of at least one of the two characters becomes 0. The player of the character with the larger remaining HP at the end of a round is the round's winner. In our experiments, the value of HP_{max} was set to 400 according to the rule of the Standard Track of FTGAIC.

In FTGAIC, the aforementioned delay of the current game state provided to AI players is a constraint that has been imposed. More specifically, both AI players can only obtain each time a game state delayed by 15 frames (0.25 s), taking into account the delay of human perception. However, the delay constraint was removed in the experiments. This is because our work is focused on generation of entertaining gameplay, not on development of an AI player for fighting against another AI opponent in a competition or a human opponent in a fair fashion.

2) Parameter Settings: In the experiments, we used five AI players: mAI, AI1, AI2, AI3, and AI4. The last four AI players used a different combination of cues in the evaluation function as can be seen by the values of ω_s, ω_a, and ω_d shown in Table II. The values of other parameters in use are summarized in Table III where the first five parameters, determined following our previous work [5] except for C being set to 1 which is a typical value when the evaluation function is normalized, were shared among all the AI players and the sixth, $center$, was only for those AI players that included $Distance$ in the evaluation function, i.e., A1, A2, and A3.

TABLE II
WEIGHTS IN THE EVALUATION FUNCTION

Name	ω_s	ω_a	ω_d
AI1	1/3	1/3	1/3
AI2	0	1/2	1/2
AI3	1/2	0	1/2
AI4	1/2	1/2	0

B. User Study

In this user study, we evaluated the fun of gameplay generated by the aforementioned five AI players[2]. Participants were mainly college students and/or followers of the Facebook

TABLE III
PARAMETERS USED IN THE EXPERIMENTS

Notation	Description	Value
C	Balancing parameter	1
N_{max}	Threshold of the number of visits	10
D_{max}	Threshold of the tree depth	10
T_{sim}	Simulation-time budget	60 frames
T_{max}	Execution time of MCTS	16.5 ms
$center$	Center of the game screen	480 pixels

[1] http://www.ice.ci.ritsumei.ac.jp/%7eftgaic/
https://github.com/TeamFightingICE/FightingICE

[2] http://www.ice.ci.ritsumei.ac.jp/%7eruck/hlmcts-cog2019.htm

site of FTGAIC. They were asked to watch two gameplay video clips: a gameplay video clip where both AI players are of the mAI type and the other gameplay video clip where both AI players are of the same type selected from one of the other four AI players. Then, given no information on how each gameplay was generated, they were asked to answer which gameplay was more fun and to comment their reasons for selection. Three rounds were generated by two AI players of the same type, and the content in the last 30 seconds of a round was recorded. The user study was conducted on an online survey site which displayed a pair of gameplay video clips of the same but randomly selected round number. To test statistical differences between any two methods, the exact binomial test was used on amalgamated results from all three rounds.

Results and Discussions: Fig. 2 shows the results of the user study. In the first half of our study (Fig. 2-(1)) where we compared gameplay by mAI and gameplay by AI1, 66 participants (57 males, 9 females), with an average age of 24.1 ± 2.5, participated. In general, the majority of them preferred the AI1 gameplay for all rounds, indicating the positive effect of using all of Score Transition, Action, and Distance. The effectiveness of these terms can also be seen in representative comments by the participants in Table IV. A difference at the 1% level of significance ($p = 0.002$) was found between the gameplay of these two AI-player types. As a result, it can be said that the AI1 gameplay is more fun than the mAI gameplay.

TABLE IV
REASONS FOR CHOICE

AI	Reason
$AI1$	Frequent changes of the dominance side More engaging gameplay Closer-range fight
mAI	Well-thought out gameplay Curious to know how the fight will be unfolded Have both close-range and far-range fights

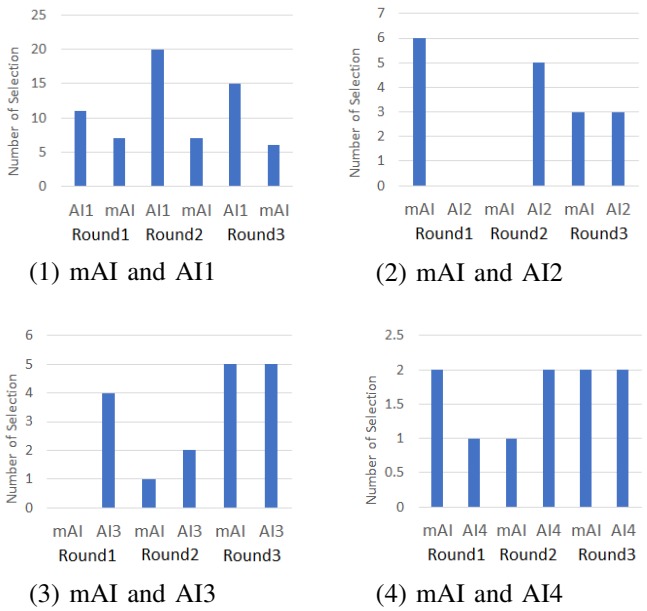
In the second half of our study conducted after the first one, we examined how the user evaluation toward gameplay would change if one of Score Transition, Action, or Distance was not included in the evaluation function, i.e., AI2 gameplay, AI3 gameplay, and AI4 gameplay, respectively. Our results indicated that the mAI gameplay was not statistically different from the AI2 gameplay and the AI4 gameplay, but statistically different from the AI3 gameplay at the 5% level of significance. For statistical significance tests, the number of participants was adjusted in each test to 66, the number of participants in the first half of the user study, for fair comparison. Below are details on the comparisons between mAI and AI2, mAI and AI3, and mAI and AI4. Note that due to random assignment of participants to one of the three comparisons, their numbers of participants were different and that in statistical tests, a significant difference can be more readily obtained with a higher number of participants.

The results of the comparison between the mAI gameplay and the AI2 gameplay are shown in Fig. 2-(2) where 17 participants (14 males, 3 females), with an average age of 23.1 ± 1.8, answered. For Round 2, all of the participants watching this round answered that the AI2 gameplay was more fun, but vice versa for Round 1. In Round 3, the number of participants who preferred either gameplay was the same. Due to the absence of Score Transition in AI2, a close-range fight using the actions in the $RankAct$ list (Table I) was continuously performed in the center of the screen throughout the fight. Therefore, the fight was sometimes repetitive causing the participants to not enjoy the gameplay. The exact binomial test, even after adjusting the number of participants to 66, showed no significant difference between the mAI gameplay and the AI2 gameplay. As a result, it can be said that the fun of the gameplay by these two AI player types is not different; in other words, Score Transition is an important element in the evaluation function.

Fig. 2-(3) shows the results of the comparison between the mAI gameplay and the AI3 gameplay where another group of 17 participants (14 males, 3 females), with an average age of 23.6 ± 2.1, participated. In Rounds 1 and 2, according to answers by participants the AI3 gameplay was more fun, but in Round 3, both types of gameplay had the same number of participants who preferred either of them. These results were obtained because there was no Action in AI3. In the gameplay by both AI3 players, a fight occurred near the center of the screen and became more tensed as the fight time increased, but the frequency of using actions, with high visual effects, in $RankAct$ decreased. The exact binomial test showed that there was a significant difference between the mAI gameplay and the AI3 gameplay at the 5% level of significance ($p = 0.019$) after the number of participants was adjusted to 66. Accordingly, it can be said that the AI3 gameplay is more fun than the mAI gameplay, but with less reliable than the difference between the mAI gameplay and the AI1 gameplay.

In the comparison between the mAI gameplay and the AI4 gameplay, 10 participants (9 males, 1 females), with an average age of 23.3 ± 1.8, participated, and the results are shown in Fig. 2-(4). In Round 2, many participants answered that the AI4 gameplay was more fun, but in Round 1, the trend became opposite. In Round 3, both types of gameplay had the same number of votes. This is because there was no Distance in AI4. Due to the absence of it, each of the AI4 players used actions in $RankAct$ throughout the round, causing a lot of damage to the opponent especially in the second half of the round, but they could not reduce the distance to the opponent. As a result, long-distance attacks were repeated, so there were not many changes in the fight causing the participants to find it not so fun. The exact binomial test, after the number of participants was adjusted to 66, showed no significant difference between both types of gameplay. From the above results, it can be said that the fun of both gameplay is not different; in other words, Distance is an important element in the evaluation function.

From all of the above results, it can be said that the AI1 gameplay where all the cue elements are used is more fun than



(1) mAI and AI1 (2) mAI and AI2

(3) mAI and AI3 (4) mAI and AI4

Fig. 2. Results of the user study

the mAI gameplay. Although when the evaluation function does not include Action, the AI3 gameplay outperforms the mAI gameplay with a statistical significance, inclusion of Action with the other cue elements contributes to a statistical significance with a more reliability. As a result, all of the proposed three cue elements forming the evaluation function are important to the fun of generated gameplay.

C. Gameplay Analyses

Here, we performed an analysis of gameplay generated by mAI, AI1, AI2, AI3, and AI4. In particular, we generated 1000 rounds of gameplay for each type and analyzed them using three criteria: "the Average Variance of the HP Difference" (AVHPD), "the Average Use of $RankAct$ Actions" (AURAA), and "the Average Distance Between the Two Characters" (ADBTC) defined as follows:

- AVHPD: the average among 1000 rounds of the variance of the HP difference between the two AI players among each 100 consecutive frames
- AURAA: the average among 1000 rounds of the number of times that each action in $RankAct$ is executed per one AI player throughout its fight
- ADBTC: the average among 1000 rounds of the average distance between the two characters throughout their fight

Results and Discussions: AVHPD for gameplay by each AI-player type is shown in Fig. 3. It can be seen that in the gameplay generated by the three AI-player types that use Score Transition (AI1, AI3, AI4), the HP difference between the two AI players varies more swiftly, causing a higher variance, near the end of round. The finding substantiates our discussions on Score Transition in the previous sub-section.

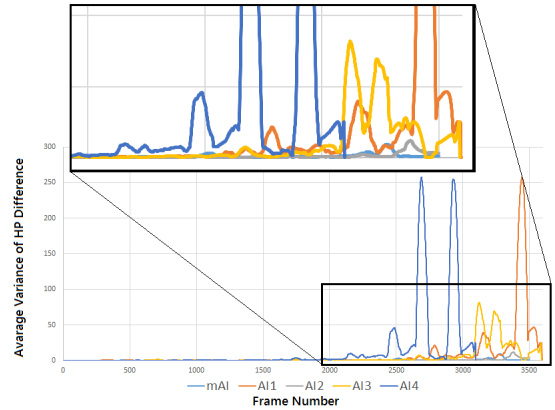


Fig. 3. The average variance of HP difference where the x-axis shows the frame number and the y-axis shows the average variance

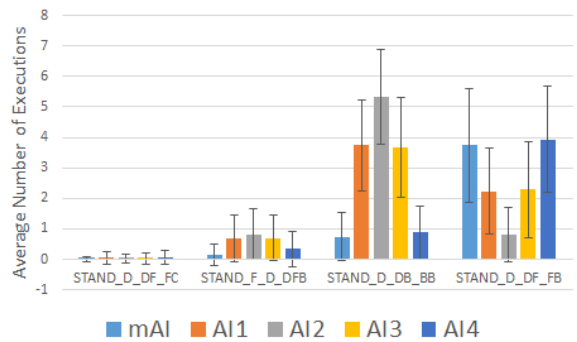


Fig. 4. The average use of $RankAct$ actions

AURAA for gameplay by each AI-player type is shown in Fig. 4 in the form of histograms with error bars. It can be seen that, due to an emergent behavior governed by each term in the evaluation function, AI1 performed the 2nd and 3rd ranked actions in the $RankAct$ list more frequently than mAI although the latter performed the 4th ranked action more frequently than the others except for AI4. This result in part substantiates the superiority of AI1 over mAI in the user study. However, the 1st ranked action in $RankAct$, the one with the highest visual effect which requires a large amount of energy to perform, was rarely executed by AI1 and its variants. As a result, there is room for improvement, e.g., how to promote execution of this kind of action, which is left as our future work.

ADBTC is shown in Fig. 5. It can be clearly seen that the gameplay generated by AI1, AI2, and AI3 that include Distance in the evaluation function represents a close-distance fight. In addition, mAI and AI4 with no Distance in the evaluation function generate fights where the two characters keep a further distance between each other. This finding shows that Distance successfully constrains the distance between the characters during their fight.

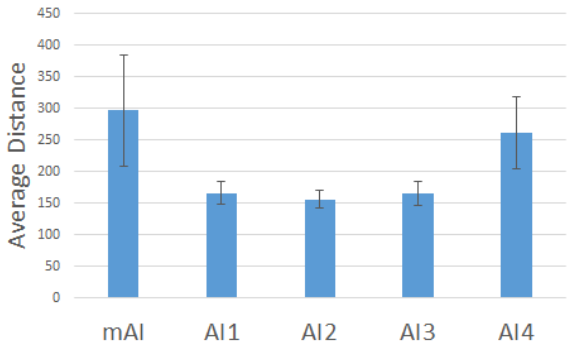


Fig. 5. The average distance between the two characters

V. Conclusions and Future Work

In this paper, we proposed a fighting game AI with the aim to generate entertaining gameplay. Based on existing highlight generation methods, we introduced three cues for use in the evaluation function of MCTS. Our results from the conducted user study showed that the proposed method that has all of the three highlight cues, AI1, generated gameplay that was more fun than gameplay generated by an existing MCTS AI that aims at decreasing its opponent's HP while maintaining its own HP. These results were also substantiated by the conducted gameplay analyses.

However, one issue is left as future work that is how to have an AI player of interest execute an action with a high visual effect that consumes a large amount of energy. Active learning for parameter tuning [25] might be useful. We are also interested in generation of believable gameplay [26-27].

Acknowledgement: This research was supported in part by Grant-in-Aid for Scientific Research (C), Number 19K12291, Japan Society for the Promotion of Science, Japan.

References

[1] T. Smith, M. Obrist and P. Wright, "Changes the (Video) Game," in *Proc. 11th European Conference on Interactive TV and Video (Live-Streaming)*, ACM, pp. 131-138, 2013.

[2] R. Thawonmas and T. Harada, "AI for Game Spectators: Rise of PPG," in *Proc. AAAI 2017 Workshop on What's next for AI in games*, San Francisco, USA, pp. 1032-1033, 2017.

[3] R. Coulom, "Efficient selectivity and backup operators in monte-carlo tree search," in *Proc. International Conference on Computers and Games*, pp. 72-83, 2006.

[4] C.B. Browne, E. Powley, D. Whitehouse, S.M. Lucas, P.I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A Survey of Monte Carlo Tree Search Methods," *IEEE Transactions on Computational Intelligence and AI in Games*, vol. 4, no. 1, pp. 1-43, 2012.

[5] R. Ishii, S. Ito, M. Ishihara, T. Harada and R. Thawonmas, "Monte-Carlo Tree Search Implementation of Fighting Game AIs with Personas," in *Proc. 2018 IEEE Conference on Computational Intelligence and Games (CIG 2018)*, Maastricht, The Netherlands, pp. 54-61, 2018.

[6] F. Lu, K. Yamamoto, L. H. Nomura, S. Mizuno, Y. Lee and R. Thawonmas, "Fighting Game Artificial Intelligence Competition Platform," in *Proc. IEEE 2nd Global Conference on Consumer Electronics (GCCE)*, pp. 320-323, 2013.

[7] Y. Takano, W. Ouyangy, S. Ito, T. Harada and R. Thawonmas, "Applying Hybrid Reward Architecture to a Fighting Game AI," in *Proc. 2018 IEEE Conference on Computational Intelligence and Games (CIG 2018)*, Maastricht, The Netherlands, pp. 433-436, 2018.

[8] S. Y. and K.-J. Kim, "Deep Q Networks for Visual Fighting Game AI," in *Proc. 2017 IEEE Conference on Computational Intelligence and Games (CIG 2017)*, New York City, USA, 3 pages, 2017.

[9] D.T.T. Nguyen, V. Quang and K. Ikeda, "Optimized Non-visual Information for Deep Neural Network in Fighting Game," in *Proc. 9th International Conference on Agents and Artificial Intelligence(ICAART 2017)*, Porto, Portugal, pp. 676-680, Feb. 2017.

[10] S. Yoshida, M. Ishihara, T. Miyazaki, Y. Nakagawa, T. Harada and R. Thawonmas, "Application of Monte-Carlo Tree Search in a Fighting Game AI," in *Proc. IEEE 5th Global Conference on Consumer Electronics (GCCE)*, pp. 623-624, 2016.

[11] D.P. Liebana, J. Dieskau, M. Hunermund, S. Mostaghim, S. Lucas, "Open Loop Search for General Video Game Playing," in *Proc. the 2015 Annual Conference on Genetic and Evolutionary Computation (GECCO' 15)*, pp. 337-344, 2015.

[12] P. Auer, N. Cesa-Bianchi, P. Fischer, "Finite-time analysis of the multiarmed bandit problem," *Machine Learning*, vol. 47, no. 2-3, pp. 235-256, 2002.

[13] B. Nunnapus, C. Nagul, J. Chuleerat, "Flashlight and player detection in fighting sport for video summarization," in *Proc. 2005 IEEE International Symposium on Communications and Information Technology (ISCIT 2005)*, pp. 441-444.

[14] B. Vinay, P. Caroline, E. Irfan, "Leveraging contextual cues for generating basketball highlights", in *Proc. 2016 ACM on Multimedia Conference*, pp. 908-917.

[15] C. Ringer and M. A. Nicolaou, "Deep unsupervised multi-view detection of video game stream highlights," in *Proc. the 13th International Conference on the Foundations of Digital Games (FDG'18)*, Article No. 15, 2018.

[16] R.T. Jewell, A. Moti, and D. Coates, "A Brief History of Violence and Aggression in Spectator Sports," in *Sports Economics, Management and Policy*, Volume 4, pp. 11-26, 2012.

[17] J. Hamari and M. Sjöblom, "What is eSports and why do people watch it?," in *Internet Research*, Vol. 27 Issue: 2, pp. 211-232, 2017.

[18] X. Neufeld, S. Mostaghim, and D. Perez-Liebana, "HTN fighter: Planning in a highly-dynamic game," in *Proc. 2017 Computer Science and Electronic Engineering (CEEC 2017)*, Colchester, pp. 189-194, Sep. 2017.

[19] S. Demediuk, M. Tamassia, Wi. Raffe, F. Zambetta, X. Li and F.F. Mueller, "Monte Carlo Tree Search Based Algorithms for Dynamic Difficulty Adjustment," in *Proc. 2017 IEEE Conference on Computational Intelligence and Games (CIG 2017)*, 2017.

[20] M.-J. Kim and K.-J. Kim, "Opponent Modeling based on Action Table for MCTS-based Fighting Game AI," in *Proc. 2017 IEEE Conference on Computational Intelligence and Games (CIG 2017)*, 2017.

[21] K. Majchrzak, J. Quadflieg, and G. Rudolph, "Advanced Dynamic Scripting for Fighting Game AI," in *Proc. Entertainment Computing (ICEC 2015)*, pp. 86-99, 2015.

[22] K. Asayama, K. Moriyama, K. Fukui, and M. Numao, "Prediction as Faster Perception in a Real-time Fighting Video Game," in *Proc. 2015 IEEE Conference on Computational Intelligence and Games (CIG 2015)*, pp. 517-522, 2015.

[23] N. Sato, S. Temsiririkkul, S. Sone. and K. Ikeda, "Adaptive Fighting Game Computer Player by Switching Multiple Rule-based Controllers," in *Proc. 3rd International Conference on Applied Computing and Information Technology (ACIT 2015)*, pp. 52-59, 2015.

[24] H. Park and K.J. Kim, "Learning to Play Fighting Game using Massive Play Data," in *Proc. 2014 IEEE Conference on Computational Intelligence and Games (CIG 2014)*, pp. 458-459, 2014.

[25] A. Zook, E. Fruchter, M. O. Riedl, "Automatic playtesting for game parameter tuning via active learning," in *Proc. the 9th International Conference on the Foundations of Digital Games 2014 (FDG'14)*, 8 pages, 2014.

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

[27] S. Devlin, A. Anspoka, N. Sephton, P.I. Cowling, "Combining Gameplay Data with Monte Carlo Tree Search to Emulate Human Play," in *Proc. Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference (AIIDE 2016)*, pp. 16-22, 2016.