Journal of Theoretical and Applied Information Technology
10th July 2014. Vol. 65 No.1
© 2005 - 2014 JATIT & LLS. All rights reserved·
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DYNAMIC DIFFICULTY ADJUSTMENT IN GAME BASED ON TYPE OF PLAYER WITH ANFIS METHOD

[1]KURNIAWAN SUTANTO, [2]SUHARJITO
[1]Master of Information Technology Program, Bina Nusantara University, Jakarta, Indonesia
[2]Assoc Prof., Master of Information Technology Program, Bina Nusantara University, Jakarta, Indonesia
E-mail: [1] kurn.stn08@gmail.com , [2] harjito@yahoo.com

ABSTRACT

The purpose of this research is to measure the accuracy and effectiveness of Adaptive Neuro Fuzzy Inference System (ANFIS) when it implemented as Dynamic Difficulty Adjustment, to manage the difficulty of a game according to the player's behavior, as the main problem is players need to switch the difficulty level manually, and it will be annoyed them. The inputs are the amount of enemies defeated and hero taken damages. There are five levels of difficulty. The measurement of accuracy done by Mean-Squared Error, meanwhile the measurement of effectiveness done by comparing the value of Mean-Squared Error to the Artificial Neural Network's. The result is ANFIS has better performance on accuracy and effectiveness, as the value of MSE is 0.010592338 compared to 0.026847068 of ANN, so it can be conclude that ANFIS has good accuracy and effectiveness when it implemented as Dynamic Difficulty Adjustment.

Keywords: *Dynamic Difficulty Adjustment, Game, ANFIS, Accuracy, Effectiveness*

1. INTRODUCTION

As population of gamer keep increasing, the amount of game also increased in order to fulfill gamers' desires. On a game, difficulty adjustment is needed to kept player feel challenged to play [12], so it won't be boring as only as follow the instruction from beginning to end [1]. In some genre of game, difficulty adjustment is provided to player so they can set it to their preference, but sometimes it can't be the solution. There will be some problems. Some expert players need an adaptation to an environment of a new game, and it's only costs a little amount of time, and it will be very annoying for them to restart the game just to adjust the difficulty level. Some of beginners also not realize that they are talented, and it will keep them bored if game was too easy for them. Dynamic Difficulty Adjustment provides the solution to the problems by adapting the user's behavior to adjust the difficulty of the game.

In this experiment, Dynamic Difficulty Adjustment implemented using Adaptive Neuro Fuzzy Inference System (ANFIS), to measure how accurate and efficient it will be to define the type of player.

Dynamic Difficulty Adjustment is a game's ability to detect player's skill level and adapt to it to change player's experience [1].

The type of player itself can be classified by two types: Core Gamer whose think the game as a challenge and enjoy the difficulty, and Casual Gamer whose only play it for fun and will give up when it was too difficult [1].

The research's objectives are to implement Dynamic Difficulty Adjustment with ANFIS then measure the accuracy and effectiveness of ANFIS as Dynamic Difficulty Adjustment, using the output of Mean-Squared Error. This research's merit is to develop a game that can adapt well with the player so he/she can comfortably play the game, and the scope is the game only an offline two dimensional shooter game.

2. RELATED WORKS

It often happened that the players did not know clearly know their level of playing, therefore K-means clustering method and Support Vector Machine is applied by Missura and Gartner [9] to classify players by type, based on previous data. Furthermore, Missura and Gartner [10] concluded that user defines the level of difficulty differently so they implement the method of Partially Ordered Sets that will update the value of difficulty based on player's game, and modifying the algorithm on previous research [10], Illici, Wang, Missura, & Gartner [5] created Dynamic Difficulty Adjustment model to Checkers and Chinese Chess where the enemy called as Master will recieve and evaluate the feedback from the way of user's play.

Baumgarten, Colton, and Morris [3] implemented an Artificial Intelligence bot on a nuclear war strategy game to build a new strategy based on previous record, using Decision Tree Learning. Using Matrix Formation as basic mathematical model and Dynamic Programming as artificial intelligence architecture, Jadon, Singhal, and Dawn [7] created more realistic bot in a military simulation where the bot's main behavior is Patrol, Active Search, and Attack, and it can be divided to several sub-behavior.

It was noticed that the static in-game AI can't adapt to a wide variety of users with different tactics, so Tan, Tan, and Tay [15] propose the application of the Adaptive Dynamic Difficulty Adjustment using Adaptive Chromosome Uni Chromosome and Adaptive Duo Chromosome with and test it with Heuristics and Neural Network. Using Adaptive Reinforcement Learning, Andrade et al [2] research how to adapt the difficulty of fighting game with comparison of Life of Agent (enemy) and Life of Player.

Using Temporal Difference Learning Method, Levene and Fenner [8] conducted an experiment to let computer learn about player's style in a chess game based on previously game record. Concluded that Dynamic Difficulty Adjustment must be measured not only from player's ability, but also player's willingness to take the risk, Hawkins, Nesbitt, and Brown [6] proposed Particle Filtering model to be implemented on the gameplay to observe player's characteristic.

A Motivational Behavior Game (MBG) is developed by Syufagi, Harini, Hariadi, and Purnomo [14] using Learning Vector Quantization to observe the input from player and interaction between player and game to determine the motivation level of player when play a certain game, and using Fuzzy Coordination method, Nugroho, Widiastuti, Hariadi, and Purnomo [11] developed Arificial Intelligence-Based agent to coordinate a team of Non-Playing Character (NPC) to behave based on how strong they are in a combat game. Syufagi, Hariadi, and Purnomo [13] also observe and classify player's motivation on a serious game using Petri Net Model.

3. METHODOLOGY

As explained in previous section, we use ANFIS for this experiment to find out the accuracy and effectiveness when it implemented on Dynamic Difficulty Adjustment on a game. ANFIS itself is one of the combination of Neural Network and Fuzzy Inference System - widely known as Neuro Fuzzy – that a graphical presentation of Takagi-Sugeno-Kang (TSK) Fuzzy Model where the rule is [3]:

IF $X_1 = A_i$ *and* $X_2 = B_i$, *THEN* $y = f(X_1, X_2)$.

The Framework of this research is drawn as on the Figure 1:

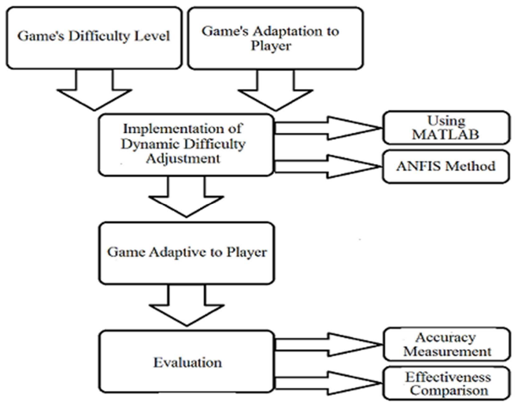

Figure 1: Framework of Research

In this experiment, we use a shooter game built on MATLAB software and make an algorithm analyzes the input from how many enemies defeated and how many damage had hero taken. The amount of training data and testing data is both 140 as a testing data use a value as input that close to the related training data's input value. The game's training also divided into two parts: normal and hard, as hard's input value is much bigger than the normal one. The calculation of output is:

AMOUNT OF ENEMIES IN A GROUP

- (ROUND(2 + (AOV / 5)) x AOE) if AOV > 5.
- (ROUND(1 + (AOV / 5)) x AOE) if 4 ≤ AOV < 5
- (1 x AOE) if 3 ≤ AOV < 4

- (ROUND(AOV / 5) x AOE) if 2 ≤ AOV < 3
- 1 if AOV < 2

ENEMY LEADER'S ATTACK SPEED

- (ROUND(1 + (AOV / 5)) x LAS) if AOV > 4
- (1 x LAS) if 3 ≤ AOV < 4
- (ROUND(AOV / 5) x LAS) AOV < 3

where ROUND is rounding to nearest, AOV is ANFIS Output Value, AOE is amount of enemies in a group (where in the first stage is random between 4 to 6, the total is 8 groups), and LAS is leader's attack speed.

For exception, the number that lower than expected ANFIS Output Value, but is very close (e.g 2.9 to 3) is rounded up to the expected value.

As the network structure, we use 5 x 5 networks for normal one and 3 x 3 networks for hard one which the input value can be described as:

Table 1: Rating for Normal Mode

Rating	Enemy Damaged	Hero Damaged
Very Good	31 - 40	0
Good	21 - 40	1 – 20
Fair	11 - 30	11 – 30
Bad	1 – 20	21 – 40
Very Bad	0	31 – 40

Table 2: Rating for Hard Mode

Rating	Enemy Damaged	Hero Damaged
Very Good	41 - 80	0
Good	21 - 70	1 – 60
Fair	0 - 50	31 – 80

For the rules to the input, 25 rules from 5 x 5 possibility are set up to evaluate the input, adapted from TSK in format of:

IF(Enemy Damaged Rating = X) AND (Hero Damaged Rating = Y) THEN DIFFICULTY = Z

Table 3: The Fuzzy TSK Rule for Normal Mode

Enemy Damaged Rating	Hero Damaged Rating	Difficulty
Very Good	Very Good	Very Hard
Very Good	Good	Hard
Very Good	Fair	Normal to Hard
Very Good	Bad	Normal
Very Good	Very Bad	Normal
Good	Very Good	Very Hard
Good	Good	Hard
Good	Fair	Normal
Good	Bad	Easy to Normal
Good	Very Bad	Easy
Fair	Very Good	Hard
Fair	Good	Normal
Fair	Fair	Easy to Normal
Fair	Bad	Easy
Fair	Very Bad	Very Easy
Bad	Very Good	Normal
Bad	Good	Easy to Normal
Bad	Fair	Easy
Bad	Bad	Easy
Bad	Very Bad	Very Easy
Very Bad	Very Good	Easy
Very Bad	Good	Very Easy
Very Bad	Fair	Very Easy
Very Bad	Bad	Very Easy
Very Bad	Very Bad	Very Easy

And the hard one is 9 rules from 3 x 3 possibility:

Table 4: The Fuzzy TSK Rule for Hard Mode

Enemy Damaged Rating	Hero Damaged Rating	Difficulty
Very Good	Very Good	Very Hard
Very Good	Good	Hard
Very Good	Fair	Normal
Good	Very Good	Very Hard
Good	Good	Hard
Good	Fair	Normal
Fair	Very Good	Normal to Hard
Fair	Good	Normal to Hard
Fair	Fair	Normal

For some rules, there are two possibilities occurs depends on the value generated from the input. For example, input Fair and Fair on Normal level will generates Easy to Normal, depends on the value of Fair. If the Enemy Damaged value is greater than the Hero Damaged values, it will generate the Normal difficulty level, but if the value is smaller will result in the Easy difficulty level.

The membership function of every input is drawn as below:

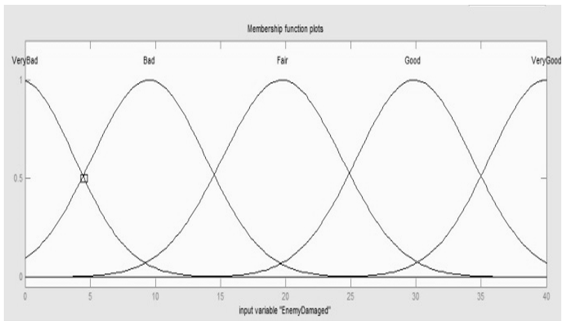

Figure 2: The Membership Function for input "EnemyDamaged" in Normal

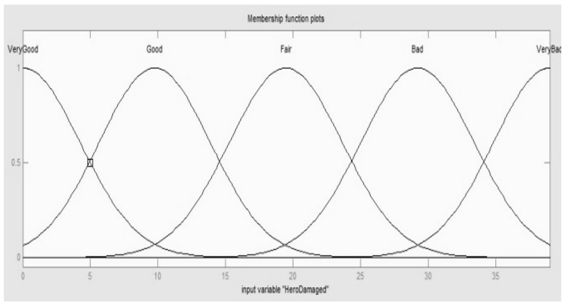

Figure 3: The Membership Function for input "HeroDamaged" in Normal

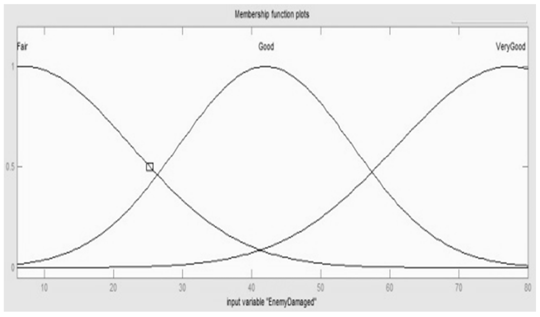

Figure 4: The Membership Function for input "EnemyDamaged" in Hard

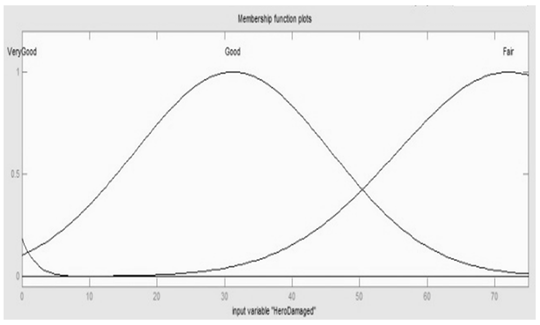

Figure 5: The Membership Function for input "HeroDamaged" in Hard

Training data at normal levels using 95 data, where on the data, the value of amount of defeated enemies is made sequentially from 40 (average amount of enemies in first stage) and the value of attacks on the hero is calculated by multiplying multiples of 3 or 1/3 (except for the difficulty level 5 where the number of attacks on the hero should be 0, and the level of difficulty 1 wherein the value of defeated enemies must be 0). The ANFIS output of Very Hard value is 5, and Hard, Normal, Easy, and Very Easy is consecutively 4, 3, 2, and 1.

For difficulty level in 4 and (Hard and Very Hard), the difficulty level can not be measured with normal data training because the number of enemies have far exceeded the limits of normal data training (in average of 80), therefore, a new training data created in order to determine the existing level of difficulty. The amount of data is 45.

Referencing to Adams [1], which states that Core Gamers play games seriously and need challenges, while Casual gamers play games just to spend leisure time and give up when the game is too difficult, then the level of the Hard and Very Hard are provided for Core Gamers who want to challenge continuously, while Easy and Very Easy level are provided for Casual Gamers to remain entertained in a game without having a frustrating fight enemies more and more.

4. EXPERIMENTS AND RESULTS

The experiment run on a computer with processor Intel Celeron 2.13 GHz, RAM 2.5 GB, Operating System in use is Microsoft Windows 7 Home Basic, and the algorithm was built on MATLAB 2013a.

The shooter game is heavily modified version of *Dave's Matlab Shooter* by David Buckingham (http://www.mathworks.com/matlabcentral/fileexch ange/31330-daves-matlab-shooter), where the purpose of modification is to implement and adapt the ANFIS algorithm in the game. There are two types of enemy boss: Colonel and General, where the Colonel start to appear from Stage 2 and General start to appear from Stage 3. The output is saved as a text file.

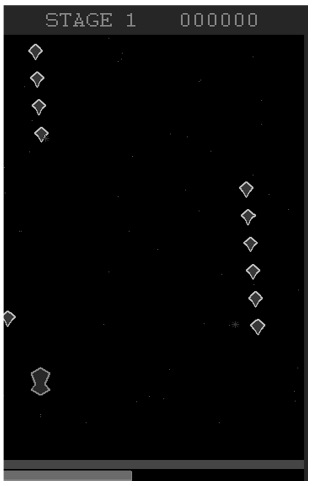

Figure 6: Game Interface

The results of the training data for normal levels by 80 epochs (only 80 epochs of training done because when the training is done in large quantities it produces higher accuracy but there are some fairly large anomalies in certain parts), showed the results of the training error by 0.079411.

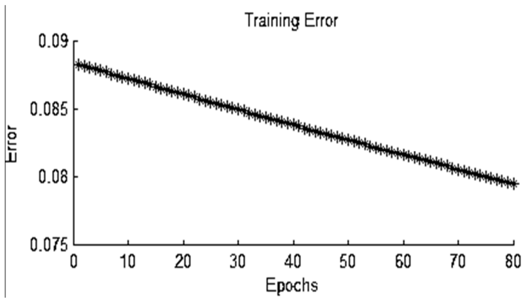

Figure 7: Training Data Result in Normal

The results of the training data for level 4 and 5 (Hard and Very Hard) in 2700 epochs, showed the results of the training error by 0.0025266

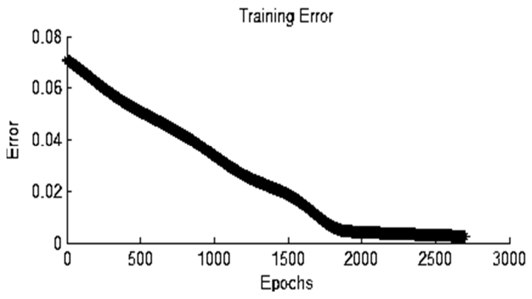

Figure 8: Training Data Result in Hard

For the testing data, the data that is taken is approaching the value of the number of enemies defeated and / or the number of attacks on the hero on training data.

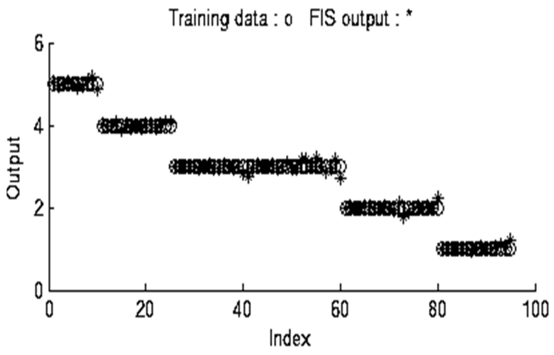

Figure 9: Testing Data in Normal compared to Training Data

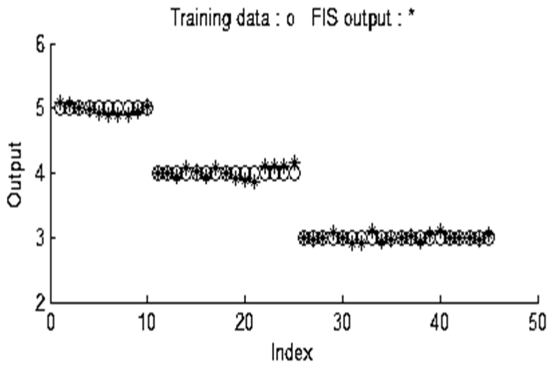

Figure 10: Testing Data in Hard compared to Training Data

To calculate the error, the method in used is Mean-Squared Error, and the following results are obtained:

Table 5: Mean-Squared Error

Level	Difficulty Prediction	Mean-Squared Error
Normal	5	0.018317621
Normal	4	0.024480593
Normal	3	0.011040508
Normal	2	0.026217356
Normal	1	0.001460445
4 dan 5	5	0.000004898
4 dan 5	4	0.000014687
4 dan 5	3	0.003202594
Mean		**0.010592338**

The results of the measurement error with Mean-Squared Error indicates that the MSE value is 0.010592338 or about 1.06% than what it should be. With the results be quietly approaching the expectation, the possibility of error in determining the degree of difficulty (and the characteristic of the player) is quite small.

To test the effectiveness of this method, then the accuracy of the data in ANFIS being compared with Artificial Neural Network (ANN) which has 25 hidden layer to normal level and 9 hidden layer for hard level, similar to the hidden layer to ANFIS, and the following results obtained:

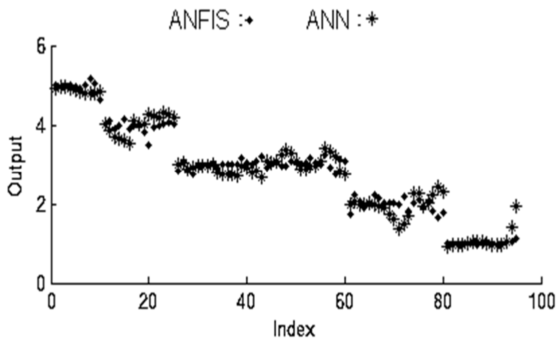

Figure 11: Result Comparison of ANFIS and ANN in Normal

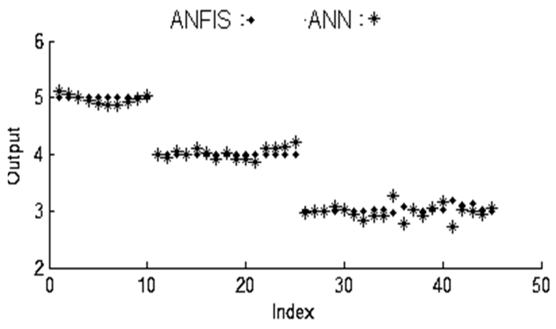

Figure 12: Result Comparison of ANFIS and ANN in Hard

From the plot above, it saw that ANFIS method has a higher degree of effectiveness as it has approached the training output value. To observe which one has the higher effectiveness, we compare their Mean-Squared Error value, and got the following result:

Table 6: Mean-Squared Error Comparison

Level	Difficulty Prediction	MSE ANFIS	MSE ANN
Normal	5	0.018317621	0.019584462
Normal	4	0.024480593	0.050989886
Normal	3	0.011040508	0.009104371
Normal	2	0.026217356	0.064340935
Normal	1	0.001460445	0.064623197
4 dan 5	5	0.000004898	0.000022
4 dan 5	4	0.000014687	0.000106147
4 dan 5	3	0.003202594	0.006005547
Mean		**0.010592338**	**0.026847068**

From the Mean Squared Error measurement result, the ANFIS method got a value of 0.010592338 and Artificial Neural Network got a value of 0.026847068. Because the average value of error in ANFIS is smaller, it could be seen that ANFIS method has higher value in effectiveness.

5. CONCLUSIONS

After the measurement of the accuracy and effectiveness of the ANFIS method in Dynamic Difficulty Adjustment to determine the characteristics of the player, so it is concluded as following:

- The level of accuracy of the ANFIS method for Dynamic Difficulty Adjustment has been pretty good, based from the results of the Mean-Squared Error that results in a fairly small value.
- ANFIS method has been effectively work when implemented for the purpose of Dynamic Difficulty Adjustment, seen from the comparison value to the Mean-Squared Error with Artificial Neural Network methods, ANFIS produces better value.
- Of the two conclusions above, it can be concluded that the Dynamic Difficulty Adjustment using ANFIS method is already well in use to determine the type and characteristics of the players on game.

Finally, we acknowledged our research's limitation, as for normal level training data, it can't be done with epoch in a large number because it will produce anomalies in certain parts. Modification of training data so it can be trained to the epoch in large number without generating anomalies and will resulting in higher accuracy is strongly recommended. Boolean rules are taken for ANFIS method is quite simple, only an "AND" rule. For the future research, it can be combined with the "NOT" rule and "OR" rule, to examine whether the level of accuracy and effectiveness will increase or decrease.

REFERENCES:

[1] Adams, E. (2010). Fundamentals of game design (2nd ed.). Berkeley, Calif.: New Riders.

[2] Andrade, G., Ramalho, G., Santana, H., & Corruble, V. (2005). Challenge-sensitive action selection: an application to game balancing. In Intelligent Agent Technology. *IEEE/WIC/ACM International Conference on*. 194-200.

[3] Baumgarten, R., Colton, S., & Morris, M. (2009). Combining AI Methods for Learning Bots in a Real-Time Strategy Game. *International Journal of Computer Games Technology*, *2009*(129075), 1-10.

[4] Du, K., & Swamy, M. N. (2006). Neural Networks in a Softcomputing Framework. London: Springer.

[5] Ilici, L., Wang, J., Missura, O., & Gartner, T. (2013). Dynamic Difficulty for Checkers and Chinese Chess. *In Computational Intelligence and Games (CIG)*, 2012 IEEE Conference on. 55-62.

[6] Hawkins, G., Nesbitt, K., & Brown, S. (2012). Dynamic Difficulty Balancing for Cautious Players and Risk Takers. *International Journal of Computer Games Technology*, *2012*(625476), 1-10.

[7] Jadon, S., Singhal, A., & Dawn, S. (2014). Military simulator - A Case Study of Behaviour Tree and Unity Based Architecture. *International Journal of Computer Applications*, *88*(5), 26-29.

[8] Levene, M., & Fenner, T. (2011). A methodology for learning players' styles from game records. *International Journal of Artificial Intelligence and Soft Computing*, *2*(4), 272-286.

[9] Missura, O., & Gärtner, T. (2009). Player Modeling for Intelligent Difficulty adjustment. *In Discovery Science*. 197-211.

[10] Missura, O., & Gärtner, T. (2011). Predicting Dynamic Difficulty. *Advances in Neural Information Processing Systems*, *24*(24). 2007-2015

[11] Nugroho, S. M. S., Widiastuti, I., Hariadi, M., & Purnomo, M. H. (2013). Fuzzy Coordinator based Intelligent Agents for Team Coordination Behavior in Close Combat Games. *Journal of Theoretical and Applied Information Technology*, *51*(2), 317-323.

[12] Rani, P., Sarkar, N., & Liu, C. (2005, July). Maintaining optimal challenge in computer games through real-time physiological feedback. *In Proceedings of the 11th International Conference on Human Computer Interaction*. 184-192.

[13] Syufagi, M. A., Hariadi, M. & Purnomo, M. H. (2013). Petri Net Model for Serious Games Based on Motivation Behavior Classification. *International Journal of Computer Games Technology*, *2013*(851287), 1-12.

[14] Syufagi, M. A., Harini, S. M., Hariadi, M., & Purnomo, M. H. (2013). Active Choice is Tendency of Players: Case studies of Motivation Classification in the Motivation Behavior Games. *Journal of Theoretical and Applied Information Technology*, *52*(1), 42-50.

[15] Tan, C. H., Tan, K. C., & Tay, A. (2011). Dynamic game difficulty scaling using adaptive behavior-based AI. *Computational Intelligence and AI in Games, IEEE Transactions on*, *3*(4), 289-301.