International Journal of Scientific and Technical Research in Engineering (IJSTRE)
www.ijstre.com Volume 7 Issue 4 ‖ Sep-Oct 2022.
ISSN: 2581-9941

How To Make NPC Learn The Strategy In Fighting Games Using Adaptive AI?

Crist Surya Kuriawan Lie[1], Wirawan Istiono[2]
[1,2]Informatics, Universitas Multimedia Nusantara, Indoensia
wirawan.istiono@umn.ac.id

Abstract : *In the video game, Artificial Intelligence (AI) is used to create non-player characters (NPCs) with the ability to generate responsive, adaptive or intelligent behaviour. In the fighting game, AI makes an enemy character that can fight the player, like when a player is fighting another player. However, using standard AI in the fighting game will make the enemy character have a static strategy. Because the AI keeps repeating the same strategy, it will decrease the game's repeatability when the player understands how to defeat the AI's strategy. One way to overcome this problem is to implement adaptive AI inside the fighting game so that the game AI can keep learning the player's strategy and adjust their own strategy. Therefore, in this research, a fighting game will be developed by implementing adaptive AI and then measuring the level of player satisfaction rate on the game that has been created. The game that has been built get a score of 73.03% from the GUESS calculation, which means the player are satisfied with the game that has been made*

Keywords - *adaptive AI, fighting Game, game AI, Artificial Intelligence*

I. INTRODUCTION

Artificial Intelligence (AI) is intelligence demonstrated by machines, as opposed to the natural intelligence displayed by humans [1]. In video games, AI refers to the ability of the computer to control game elements to make intelligent decisions when a condition has several choices with different outcomes, resulting in relevant, effective, and useful behaviour [2] [3]. An example of AI usage in a video game can be seen in controlling the enemy's character behaviour in a fighting game.

Traditionally, AI usage in video games focused on developing static strategies based on the game state to decide a set of actions with the highest probability of winning [4]. This type of AI is incompatible with a fighting game because fighting games often have no objectively correct move in any given situation. Action taken in the fighting game can be determined as good or bad depending on the enemy reaction. Other than that, this type of AI tends to be repetitive and transparent to the player because AI will use a fixed strategy [5]. Based on this problem, this research will develop adaptive AI in a fighting game that can adapt to the plater reaction, so game AI can provide better resistance. Other than that, with the use of adaptive AI over traditional AI, players will be pushed to keep creating new strategies to defeat enemy AI [6] [7]. Therefore, the created game will be less repetitive.

Similar research also had been done by Ricciardi, who developed online learning AI in the fighting game by implementing the Markov decision process. However, in the research that has been done, the state composition used only consists of seven types of moves that can be done and ignore other conditions inside the game environment. For example, the health status of AI and the player will not influence AI decision-making [8] [9]. In this study, adaptive AI will be developed using online machine learning using notify plugin. And then, in this study, the developed adaptive AI will take various game conditions as input for AI decision-making, for example, the health status of the AI and player, the position of the AI and player, and so on.

II. LITERATURE REVIEW

Adaptive AI refers to a dynamic non-player character (NPC) in which the computer is able to change its game behaviour in response to its enemy, either during gaming sessions or between sessions. In certain cases, dynamic game difficulty scaling uses adaptive AI to automatically adapt game parameters and behaviour in real-time according to the player's skill level in the game [10]. One example of an adaptive AI approach is case-based adaptive game AI, an approach to game AI in which domain knowledge is collected automatically by the game AI, and exploited directly to produce effective behaviour [11]. The effect of implementing adaptive AI in game to improve the gaming experience this is because adaptive AI can adapt to each individual to match the way players play and the goals of the player playing the game [6], [12] and Reducing development time and costs if a game is self-adapting, this is because game developers need less effort to predict all contingencies [8], [13].

Unsupervised online Machine learning in game AI is a technique that requires automatic learning process to be applied to the game AI while the game is being [12]. The application of online machine learning techniques in game will make game AI able to adapt to how players are playing the game. Therefore, by using online machine learning, AI in game can be adaptive. According from research that P. Spronck has done, there are two ways that can be used to apply online machine learning techniques to game AI, such as human-controlled online, which mean is an online machine learning technique that implements changes to game AI by processing direct feedback provided by players on all decisions made by AI. The feedback indicates whether the decision taken by the AI is desirable or not. And computer-controlled online learning which mean is an online machine learning technique that implements automatic changes to games by processing observations of the effects of game AI during live games [14].

Game user experience satisfaction scale (GUESS) is a method used to measure the level of player video game satisfaction. The GUESS method has 55 questions divided into 9 subclass: usability / playability, narratives, play engrossment, enjoyment, creative freedom, audio aesthetics, personal gratification, social connectivity, and visual aesthetic [15]. If the game that being evaluated using GUESS does not have narratives or social connectivity components, then narratives or social connectivity subscales can be omitted from the questionnaire [15]. In calculating GUESS score, it is recommended to average the rating of each item in order to obtain the average value of each subscale. Furthermore, the average value of each subscale can be summed to obtain a combined score of video game satisfaction. Questions from GUESS will be measured using a 7-point Likert scale. The lowest value is one, which means strongly disagree and the highest value is seven which means strongly agree.

III. METHODOLOGY

The research methodology that will be used in the development of this game, the first step is literature review; at this stage, will conduct a literature review by searching, studying, and reading previous studies from scientific journals, written works, and articles. Literature studies were conducted in fields related to fighting games, game design, game architecture, game design documents, GUESS, unity, online machine learning, artificial noedify, and Adaptive AI. And the second step is system Design, where at this stage, the design of the system to be built will be carried out, for example, a game environment, learning algorithms on agents, and so on. The third step is system programming, where at this stage, the program's implementation will be carried out based on the design that was made in the previous stage. Implementation will be done using Unity to build a game environment and training on agents will be done using noedify at runtime. The five step is testing, where at this stage, testing will be carried out on the games that have been made. The testing process will be carried out out to measure the level of player game satisfaction and measure the adaptive AI performance. The measurement player game satisfaction will be carried out by doing playtesting on the games that have been made on randomly selected respondents. Respondents who have played the game will then be given a GUESS (Game User Experience Satisfaction Scale) questionnaire using a 7-point Likert scale (1 = strongly disagree, 5 = moderately agree, 7 = strongly agree). The adaptive AI performance measurement will be carried out by comparing the adaptive AI that has been built with the static AI created using finite state machine (FSM) method. The final step is evaluation, where at this stage, the data obtained from filling out the GUESS questionnaire and the results of the match between adaptive AI and static AI obtained at the testing stage will be conducted. The results of the evaluation of the GUESS questionnaire will be used to measure the level of respondents satisfaction with the games that have been made. The results of the match between adaptive AI and static AI will be used to measure adaptive AI performance.

The game that will be built is named Extreme FighterZ, and the neural network model used in the Extreme FighterZ game was created using the notify plugin. The structure of the neural network model that is created consists of an input layer that accepts 17 types of input obtained from the game environment, an output layer that produces ten types of output to determine the action of the adaptive AI, and a hidden layer. After the system determines the game mode, the system will spawn the character selected by the player at the specified spawn position. Before starting each round, the initRound process will be carried out which will be described in figure 1. After the initRound process is complete, a countdown is carried out starting from three before each round starts. When the countdown process is complete, the round will start and the player can start the battle process.

During the battle process, the system will check player one and two's health point status. If the health point status of player one or player two reaches zero, the round will end and the endRound process will be carried out, as described in figure 3. After the endRound process is complete, the system will check player one and two scores. If the score from player one or player two reaches three, the match will be considered finished, then the system will display the winner's name and return to the character selection page. If player one and player two scores has not yet reached three, then the initRound process will be carried out again to start the next round.

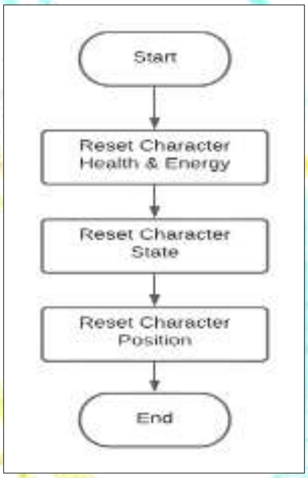


Fig 1. initRound Flowchart

Figure 1 is the flow of the initRound process. In the initRound process, the health points and energy status of player one and player two will be set back to 100. Then the positions of player one and player two will be reset to the specified spawn position.

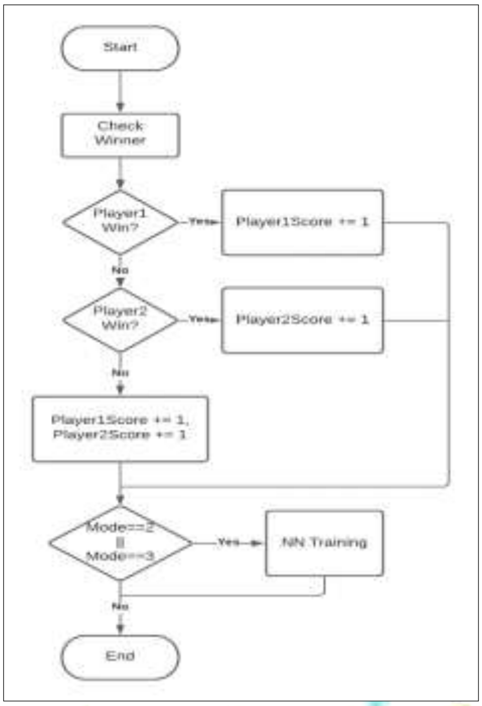


Fig 2. endRound Flowchart

Figure 2 is the flow of the endRound process. In the endRound process, the system will check the winner of the round that just ended. After the system determines the winner of the round, the system will add one score to the winner of that round. The system will award one point to both players if the round is considered a draw. After the point award process is complete, the system will check the mode of the game. If the game mode is mode two or mode three, the adaptive AI training process will be carried out using the training set obtained in that round.

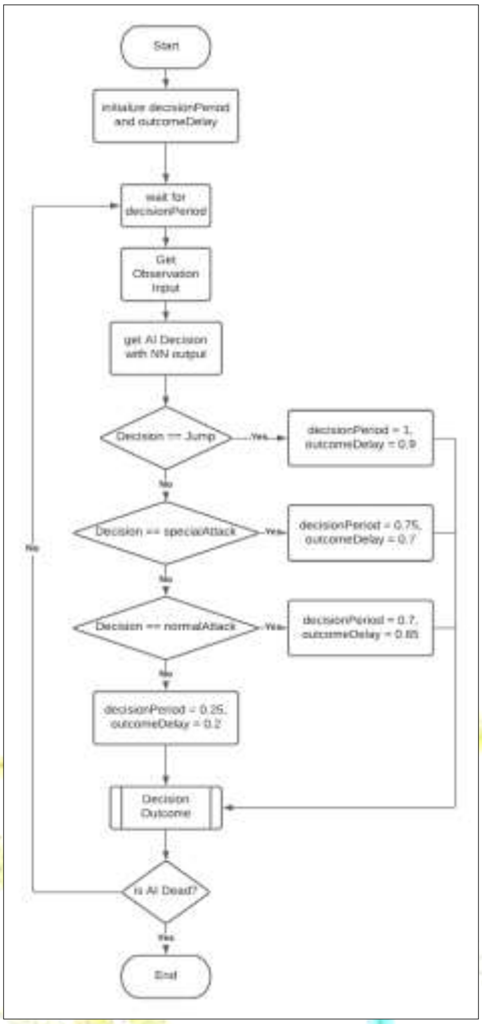


Fig 3. Adaptive AI Behaviour Flowchart

Figure 3 is the flow of adaptive AI behavior that shows the decision-making process carried out by adaptive AI. Whenever adaptive AI will make a new decision, adaptive AI will wait for the specified decisionPeriod. After reaching the decisionPeriod, the adaptive AI will make a new decision based on the calculated value from the neural network model. Based on the decision taken, the values of decisionPeriod and outcomeDelay will be changed. After the decision-making process is complete, the decisions taken will be weighted and used as a new training set through the Decision Outcome process, which will be explained in Figure 4. This process will be repeated until the match ends or the AI dies.

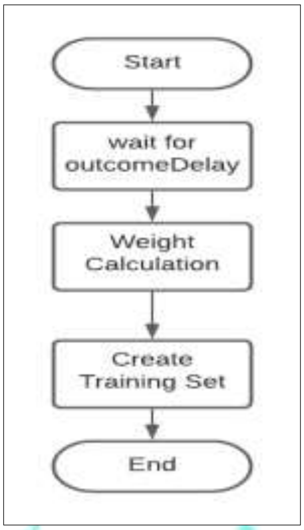


Fig 4. Decision Outcome Flowchart

Figure 4 is the flow of the Decision Outcome process. The Decision Outcome process is a process that is carried out to give weight to the decisions made by adaptive AI and form a training set based on the input, decision, and weight data. Each time before starting the process, the system will wait for the specified outcomeDelay. After reaching the outcomeDelay, the system will perform calculations to calculate the weight that will be given to the decision that has been taken based on the value provided. After the weight has been calculated, the weight will be clamped with zero and one limits. After all processes are complete, a training set will be made based on the input, output, and weight determined.

Fig 5. Static AI Behavior Flowchart

Figure 5 is a flow of static AI behavior that shows the decision-making process carried out by static AI. First, the AI will check the distance between the static AI and the opponent. If the distance obtained is smaller than the specified distanceTolerance, the aiState will be changed to closeState and vice versa if the distance is greater than the distanceTolerance, the aiState will be changed to normalState. When aiState is in closeState, the AI must wait for the specified closeRate before performing the close combat behavior. When aiState is in normalState, AI must wait for the specified normalRate before performing normal state behavior.

Close state behavior is a decision made by static AI when close to an opponent. When in close state behavior, AI will have a greater chance of carrying out attacks and defensive actions. Normal state behavior is a decision made by static AI when it is far from the opponent. When in a normal state behavior, the AI will not perform melee attacks and the AI will have a high probability of moving closer to the opponent.

IV. RESULT AND DISCUSSION

The results of the implementation will be shown in the video game display. In Figure 6a is a screenshot of the main menu of the Extreme FighterZ game. In the main menu display, there are four menus to choose from. The first menu is the Play menu which can be used to select the game mode. The second menu is the Options menu, which can open the configuration view. The third menu is the Command menu, which can open the command view. The fourth menu is the Quit menu which can be used to exit the Extreme FighterZ game. In figure 6b shown the configuration display of the Extreme FighterZ game. In the configuration display, there are two sliders that can be used to control the volume in the game. The first slider is a music slider that can control the background music volume in the game. The second slider is an sfx slider that can be used to control the volume of sound effects in games. In the configuration display there is also an ok button that can be used to close the configuration display

a. Main menu display b. Configuration Display

Fig 6. Main Menu Game Xtreme Fighter Z

Figure 7a is the command display of the Extreme FighterZ game. In the command view, there is a list of the various actions that the player can perform in the gameplay. Just like the configuration display, there is an ok button that can be used to close the configuration view. In figure 7b is the character selection page of the Extreme FighterZ game. In the middle of the character selection page, there are two characters that the user can select. At the top left corner of the character selection page, there is a back to main-menu button that can be used to return to the main menu.

a. Command Display

b. Character Selection Page

Fig 7. Information and selection menu Game Xtreme Fighter Z

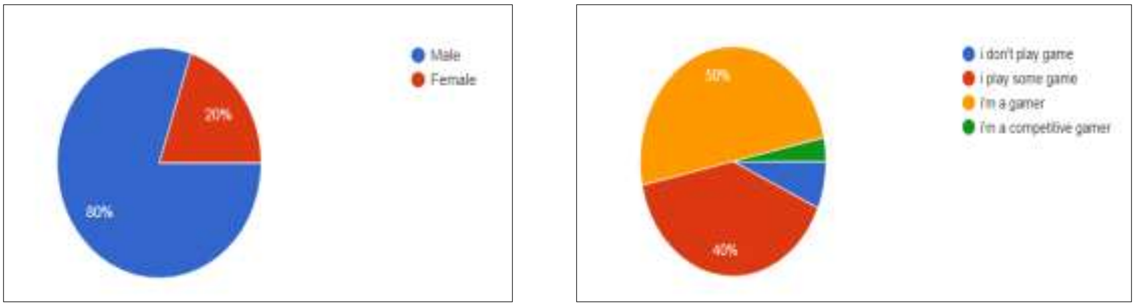
Figure 8a is the gameplay view of the Extreme FighterZ game. In the gameplay section, player one will control the character selected to defeat player two. Figure 8b shows a display of the training process carried out on adaptive AI at the end of each round. The training process is only carried out in modes that use adaptive AI.

a. Gameplay Display

b. Training Model Display

Fig 8. Gameplay in Game Xtreme Fighter Z

The evaluation of player game satisfaction was carried out using the GUESS questionnaire by Mikki Phan. The GUESS questionnaire consists of 55 questions divided into nine subscales which are measured using a seven-point Likert scale. A value of one on the Likert scale means strongly disagree and a value of seven on the Likert scale means strongly agree. GUESS's subscales consist of: Usability, Narratives, Play Engagement, Enjoyment, Creative Freedom, Audio aesthetics, Personal Gratification, Social connectivity, and Visual aesthetics. However, in this study, narratives and social connectivity subscales were not used because the game that being developed was not story-driven and is a single-player game. Therefore, the number of questions given to players is 44 questions. The test was conducted on 30 randomly selected respondents. The testing process is carried out by asking respondents to play the games that have been made, then filling out the GUESS questionnaire to measure the respondents' satisfaction with the games that have been made. In figure 9a shows is a pie chart that shows respondents based on gender and the level of respondents' preference for games. In figure, 9a shows 80% of the participants who took part in the evaluation were men and the remaining 20% were women. Figure 9b is a pie chart that illustrates the participant's preference for game. Based on the pie chart in Figure 14, it can be seen that 6.7% of participants do not play games, 40% of participants play games but are not gamers, 50% of participants are gamers, and 3.3% of participants are competitive gamers. Based on the data from Figures 9a and 9b, it can be concluded that most of the respondents who took part in this evaluation were men who liked to play games. The following is the calculation result from the GUESS evaluation



a. Respondent Gamer Level b. Respondent Gender Pie Chart

Fig. 9. Respondent category

The value of the player game satisfaction level can be obtained by adding up the results of each subscale and then dividing the sum by the number of subscales used. Table 1 shows the result of calculations from the GUESS questionnaire. Based on the results shown in Table 1, it can be concluded that the level of player game satisfaction is 73.03%, which means that players are satisfied with the Extreme FighterZ game. The majority of the subscale values are satisfied, except for the play engagement subscale, which gets a neutral score and the creative freedom subscale which gets a fairly satisfied score.

Table 1. Guess Calculation Result

Subscale	**Avarage**
Usability/Playability	79.957%
Play Engrossment	55.536%
Enjoyment	73.714%
Creative Freedom	65.374%
Audio Aesthetics	79.643%
Personal Gratification	77.460%
Visual Aesthetics	79.524%
Average	73.03%

The adaptive AI performance evaluation compares the adaptive AI that has been built using the noedify plugin with static AI created using a simple FSM. The matching process will be carried out on all characters in the game. Each match will be held in 100 rounds. The performance of adaptive AI can then be seen based on the number of rounds won by adaptive AI. Adaptive AI is able to win 70 to 91 matches in 100 rounds. This shows that the adaptive AI that has been built has better performance compared to static AI. Based on testing result, it can be concluded that, the adaptive AI built has better performance than static AI.

V. CONCLUSION

Based on the research that has been done, the implementation of adaptive AI in fighting games has been successfully carried out using the noedify plugin. Game development and design is carried out in the Unity 2019.2.12f1 game engine on a computer platform. Based on the evaluation results of the GUESS questionnaire conducted on 30 respondents, it can be concluded that players have a 73.03% satisfaction level for the Extreme FighterZ game. The majority of the subscale scores from the GUESS questionnaire were satisfied, except for the play engagement subscale, which scored neutral and the creative freedom subscale, which scored quite satisfied. The value obtained from the evaluation of the GUESS questionnaire indicates that players are satisfied with the Extreme FighterZ game. And for the results of the adaptive AI performance evaluation, adaptive AI is able to win more matches than static AI, adaptive AI is able to win 70 until 91 matches in 100 rounds. From the result, it can be conclude, adaptive AI has better performance compared to static AI.

VI. ACKNOWLEDGEMENTS

Thank you to the Universitas Multimedia Nusantara, Indonesia which has become a place for researchers to develop this journal research. Hopefully, this research can make a major contribution to the advancement of technology in Indonesia.

REFERENCES

[1] T. Kabudi, I. Pappas, and D. H. Olsen, "AI-enabled adaptive learning systems: A systematic mapping of the literature," *Computers and Education: Artificial Intelligence*, vol. 2, no. March, p. 100017, 2021, doi: 10.1016/j.caeai.2021.100017.

[2] M. Kopel and T. Hajas, "Implementing AI for Non-player Characters in 3D Video Games," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10751 LNAI, no. January, pp. 610–619, 2018, doi: 10.1007/978-3-319-75417-8_57.

[3] W. T. Ng *et al.*, "Application of Artificial Intelligence for Nasopharyngeal Carcinoma Management – A Systematic Review," *Cancer Management and Research*, vol. Volume 14, no. September 2021, pp. 339–366, 2022, doi: 10.2147/cmar.s341583.

[4] F. Dignum, J. Westra, W. A. Van Doesburg, and M. Harbers, "Games and agents: Designing intelligent gameplay," *International Journal of Computer Games Technology*, vol. 2009, no. 1, 2009, doi: 10.1155/2009/837095.

[5] W. Westera *et al.*, "Artificial intelligence moving serious gaming: Presenting reusable game AI components," *Education and Information Technologies*, vol. 25, no. 1, pp. 351–380, 2020, doi: 10.1007/s10639-019-09968-2.

[6] K. Okamura and S. Yamada, "Adaptive trust calibration for human-AI collaboration," *PLoS ONE*, vol. 15, no. 2, pp. 1–20, 2020, doi: 10.1371/journal.pone.0229132.

[7] Fedro and W. Istiono, "Mandarin Language Learning with Gamification Method," *International Journal of Advanced Trends in Computer Science and Engineering*, vol. 10, no. 5, pp. 3046–3052, 2021, doi: 10.30534/ijatcse/2021/171052021.

[8] A. Ricciardi and P. Thill, "Adaptive AI for Fighting Games," *Memory*, pp. 3–7, 2008.

[9] A. Rollings and D. Morris, *Game Architecture and Design : A New Edition*. 2004.

[10] S. S. Younes, "Examining the Effectiveness of Using Adaptive AI-Enabled e-Learning during the Pandemic of COVID-19," *Journal of Healthcare Engineering*, vol. 2021, 2021, doi: 10.1155/2021/3928326.

[11] S. C. J. Bakkes, P. H. M. Spronck, and H. Jaap van den Herik, "Opponent modelling for case-based adaptive game AI," *Entertainment Computing*, vol. 1, no. 1, pp. 27–37, 2009, doi: 10.1016/j.entcom.2009.09.001.

[12] P. Spronck, I. Sprinkhuizen-Kuyper, and E. Postma, "Online Adaptation of Game Opponent AI with Dynamic Scripting," *Proceedings of the 5th International Conference on Intelligent Games and Simulation (GAME-ON 2004)*, pp. 33–37, 2004.

[13] X. Zhu, "Behavior tree design of intelligent behavior of non-player character (NPC) based on Unity3D," *Journal of Intelligent and Fuzzy Systems*, vol. 37, no. 5, pp. 6071–6079, 2019, doi: 10.3233/JIFS-179190.

[14] P. Spronck, M. Ponsen, I. Sprinkhuizen-Kuyper, and E. Postma, "Adaptive game AI with dynamic scripting," *Machine Learning*, vol. 63, no. 3, pp. 217–248, 2006, doi: 10.1007/s10994-006-6205-6.

[15] M. H. Phan, J. R. Keebler, and B. S. Chaparro, "The Development and Validation of the Game User Experience Satisfaction Scale (GUESS)," *Human Factors*, vol. 58, no. 8, pp. 1217–1247, 2016, doi: 10.1177/0018720816669646.