Implementation of AlphaDDA-Based Dynamic Difficulty Adjustment for Game Balancing in The Congklak Board Game

Billdan Satriana Roseandree[1*], Hendriyana[2], Dian Anggraini[3], Yusmadi Yah Jusoh[4]

[1, 2, 3]Department of Software Engineering, Cibiru Regional Campus, Indonesia University of Education, Bandung, West Java, Indonesia
[4]Department of Software Engineering and Information System, Faculty of Computer Science and Information Technology, University Putra Malaysia, Serdang, Selangor, Malaysia
E-mail: [1*]billdansatriana@upi.edu, [2]hendriyana@upi.edu, [3]dian.anggraini@upi.edu, [4]yusmadi@upm.edu.my

(Received: 31 Aug 2026, revised: 3 Oct 2026, accepted: 5 Oct 2026)

Abstract

Traditional folk games such as Congklak are losing popularity, which motivates digital adaptations that remain balanced for players of different skill levels. AlphaDDA, an AlphaZero-based Dynamic Difficulty Adjustment (DDA) method that scales the number of Monte Carlo Tree Search (MCTS) simulations, has so far been evaluated on Connect4 and Othello, mainly in AI-versus-AI experiments. Its effect on human players, and its transfer to a game with non-binary states, relay sowing, capture, and extra turns such as Congklak, remain untested. This study implements AlphaDDA in the Congklak board game and evaluates its effect on game balance (score difference, Δscore) and on engagement (playtime, used as a limited behavioural proxy) following the Design Science Research Methodology (DSRM). A baseline AlphaZero model trained via self-play reached 1729.31 Elo in 50 round-robin tournaments, above the Random and MCTS agents and close to the Minimax agent. At runtime, difficulty was adjusted by changing the MCTS simulation budget according to value estimates, with parameters set by manual tuning. In a within-subject experiment with 10 male participants aged 21–22 years (convenience sampling), paired t-tests showed that DDA narrowed the mean Δscore to 11.80 points from 26.00 points in the non-DDA condition (one-tailed $p = 0.011$; $d = -0.87$), whereas playtime did not differ significantly ($p = 0.798$). Given the small and homogeneous sample, the findings are preliminary: AlphaDDA appears effective for game balancing in Congklak, but no evidence was found that it increases playtime, which was used as a limited behavioural proxy for engagement.

Keywords: Dynamic Difficulty Adjustment, AlphaDDA, AlphaZero, Congklak, Game Balancing.

I. INTRODUCTION

Research on Artificial Intelligence (AI) in gaming has advanced to the point where it can surpass the abilities of professional human players in various games. However, optimizing AI agents' performance solely to defeat players can undermine the player experience: agents that are too strong cause frustration, while those that are too weak lead to boredom. Therefore, the paradigm of game AI is shifting from "playing to win" to "playing for experience" through Dynamic Difficulty Adjustment (DDA), a mechanism that modifies the difficulty level in real time based on game conditions and player ability [1], [2], [3], [4].

Guo et al. define DDA as an automatic game difficulty control mechanism that evaluates both objective and subjective game difficulty and then modifies game tasks. The goal-based DDA framework emphasizes that DDA should be evaluated against specific game design goals. In this study, game balancing was chosen as the primary goal because, in competitive games, balance is necessary so that both sides have a more equal chance of winning [5], [6], [7].

In board games, applying DDA poses challenges because the evaluation of a player's ability relies on the board state. Fujita proposed AlphaDDA, a modification of AlphaZero that retains the Deep Neural Network (DNN) and Monte Carlo Tree Search (MCTS) but adjusts the agent's strength by varying the number of MCTS simulations based on the estimated value of the game position [4]. AlphaDDA has previously been evaluated on Connect4 and Othello, primarily through AI-versus-AI experiments; therefore, its effectiveness against human players and its generalizability to other domains still require empirical evidence.

Congklak was chosen as the research domain because its mechanics differ from those of Connect4 and Othello: the number of pieces in each hole is non-binary, and the game features relay sowing, capture, and extra turns. Furthermore,

DOI: 10.34148/teknika.v15i3.1562

the digitalization of Congklak is relevant to the preservation of traditional games; the 2024 Socio-Cultural Statistics report notes that 10.20% of the Indonesian population aged five and older still play traditional games [8], [9].

Research gap and novelty. Two gaps follow from the above. First, evidence for AlphaDDA comes mainly from AI-versus-AI experiments on Connect4 and Othello [4], so it is unknown whether the adjustment of search strength narrows the outcome gap when the opponent is a human. Second, it is unknown whether the approach transfers to a game such as Congklak, where hole contents are counted rather than binary states, and a single move can trigger relay sowing, captures, and extra turns. This study differs from previous AlphaDDA research in three respects: (i) it evaluates AlphaDDA against human players in a within-subject experiment; (ii) it applies AlphaDDA to a new game with a different state structure, for which the AlphaZero agent, the state representation, and the digital game were built from scratch; and (iii) it deploys the trained model in a Unity game with a capped simulation budget (Nmax = 300) so that the search runs in real time during play.

Engagement is operationalized in this study as playtime, defined as the accumulated think time of the human player during a session. In turn-based board games, think time is a behavioural trace of the cognitive effort a player invests in decisions, and it can be collected unobtrusively from game telemetry, which is why telemetry measures are widely used in games user research. Playtime is nevertheless only a proxy for engagement: longer deliberation can reflect interest but also confusion or excessive difficulty; shorter deliberation can reflect fluency or boredom, and playtime does not capture enjoyment or immersion. The findings on engagement are therefore interpreted as findings on behavioural engagement only (see the Discussion).

This study aims to (1) develop the AlphaDDA agent to play in the Congklak board game environment and (2) evaluate the effectiveness of AlphaDDA in terms of game balance and engagement, as measured by gameplay metrics such as score difference and play duration. Two directional hypotheses were tested: H1, the mean score difference (Δscore) is smaller in the DDA condition than in the non-DDA condition; and H2, the mean playtime is longer in the DDA condition than in the non-DDA condition. The contribution of this research is the application of AlphaDDA to the Congklak domain and its empirical evaluation on human subjects using game telemetry.

II. RESEARCH METHODOLOGY

A. Methodology and Experimental Design

The study employed the Design Science Research Methodology (DSRM), a process model for building and evaluating a technological artifact that addresses a stated problem [10], [11]. DSRM consists of six activities, which were mapped to this study as follows. (1) Problem identification and motivation: there is no empirical evidence of AlphaDDA with human players or on a game with Congklak's mechanics (Section I). (2) Objectives of a solution: an agent that adjusts its playing strength to narrow the score gap without reducing player engagement. (3) Design and development: Fujita's AlphaDDA artifact was adapted to a new context—Congklak—by building the AlphaZero agent, the AlphaDDA module, and the digital game (Sections II-B, II-C, and III-B). (4) Demonstration: the artifact was playtested by ten participants. (5) Evaluation: DDA and non-DDA conditions were compared quantitatively on Δscore and playtime. (6) Communication: this article.

The evaluation used a quantitative approach with a within-subjects design: each of the ten participants (n = 10) played under two conditions, a non-DDA control (static AI) and a DDA experimental condition (AlphaDDA). To control order effects, AB/BA counterbalancing was used, as shown in Table 1. Participants were divided into two groups: Group A played the non-DDA condition in Match 1 and the DDA condition in Match 2, whereas Group B played the two conditions in the reverse order. O1–O4 denotes the observations (Δscore and playtime) recorded after each match. Because each condition appears in both match positions, a systematic learning or fatigue effect is not confounded with the condition [12], [13]. The sample was selected through convenience sampling. All participants were male, aged 21–22 years; five had played Congklak before, and five had never played it [14]. The sample size was determined by the exploratory, single-session scope of the study, with a balanced split of experienced and inexperienced players, rather than by an a priori power analysis; a sensitivity analysis is reported in Section II-C, and the implications of the small and homogeneous sample are discussed under Limitations.

Table 1. Within-Subject Design with AB/BA Counterbalancing (Non-DDA = static AI; DDA = AlphaDDA)

Group	Match 1	Observation	Match 2	Observation
A	Non-DDA	O1	DDA	O2
B	DDA	O3	Non-DDA	O4

B. Implementation of AlphaZero and AlphaDDA

The AlphaZero Congklak agent is a policy–value neural network combined with MCTS. The Congklak environment is represented by a 3×2×8 canonical state tensor, i.e., the board as seen by the player to move. The circular board geometry is mapped onto a 2×8 grid. The policy head generates seven outputs corresponding to the player's seven active holes, while the value head generates a single output in the range [-1, 1] that estimates how favourable the position is for the player to move. This architecture uses 10 residual blocks, a 3×3 kernel, and 256 filters. The MCTS simulation during the training phase is set to Nsim = 800.

AlphaDDA retains the policy network, value network, selection, expansion, evaluation, and backpropagation in MCTS. Difficulty level adjustment is performed solely through a computational budget in the form of the number of MCTS simulations per turn. In the Congklak implementation,

Nh = 1 is used, so the values are derived from the current state only. The DDA parameters used throughout the testing are Asim = 1.5, X0 = -1.75, and Nmax = 300 (Table 2); these parameters were determined through manual tuning rather than a grid search, and their robustness was not analysed (see Limitations).

Table 2. AlphaDDA Configuration in Congklak

Component	Configuration
State Representation	3×2×8
Policy Head	7 outputs
Residual Blocks	10
Filter/Kernel	256 / 3×3
MCTS Training	800 simulations
DDA Runtime	Asim = 1.5; X0 = -1.75; Nmax = 300; Nh = 1

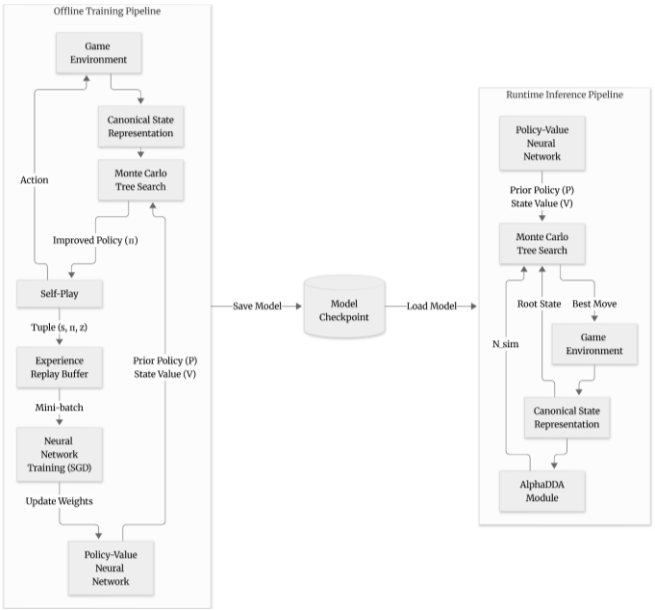


Figure 1. Architecture of the Congklak Alphazero Training Pipeline (Left) and the Alphadda Runtime Inference Pipeline (Right)

Figure 1 shows the complete pipeline: an offline AlphaZero training pipeline (left) and a runtime AlphaDDA inference pipeline (right), connected by a saved model checkpoint. The offline pipeline proceeds in sequence. (1) The game environment provides the current board state, (2) which is converted into canonical state representation. (3) MCTS searches from this state, using the policy–value neural network, which returns a prior policy (P) and a state value (V) for the states it visits, and produces an improved policy (π). (4) In self-play, an action is selected according to π and applied to the game environment; steps 1–4 repeat until the game ends. (5) Each position is stored in the experience replay buffer as a tuple (s, π, z), where s is the state, π the improved policy, and z the final game outcome. (6) Mini batches sampled from the buffer are used to train the network by stochastic gradient descent (SGD), (7) which updates the network weights; the updated network is used in the next self-play iteration. After 75 iterations, the model is saved as a checkpoint. In the runtime pipeline, (8) the checkpoint is loaded, and (9) the canonical state is passed both to MCTS as the root state and to the AlphaDDA module. (10) The AlphaDDA module uses the value estimate of the current state (Nh = 1) to set the number of simulations Nsim for the turn, bounded by Nmax. (11) MCTS runs Nsim simulations guided by P and V from the same trained network, and (12) the best move is returned to the game environment. AlphaDDA therefore adds only steps (10); the network and the search algorithm are unchanged.

C. Training, Deployment, and Telemetry

The AlphaZero model was trained via self-play. Training was stopped at the 75th iteration based on the "convergence until stable" stopping criterion after the trends in policy loss and value loss leveled off. The model was then evaluated in 50 round-robin tournaments against three reference agents: Minimax (a classical adversarial search algorithm), MCTS without a neural network (pure MCTS), and Random (an agent that selects a legal move at random). In a round-robin tournament every agent plays every other agent, and Elo ratings are updated from the results; the Elo rating is a relative strength scale in which a higher rating indicates a higher expected probability of winning against lower-rated agents. The selected model was exported from PyTorch (a deep-learning framework) to the Open Neural Network Exchange (ONNX) format and run in Unity using the Unity Inference Engine (Sentis).

Telemetry recorded turns, thinkTime, scoreP1, and scoreP2. Game balance was defined as the absolute difference in final scores, $\Delta score = |scoreP1 - scoreP2|$; values closer to zero indicate a more balanced game. Engagement was operationalized as playtime, defined as the accumulation of the human player's thinkTime during a single session (a behavioural proxy; see the Discussion). Statistical analysis used a paired samples t-test with the Shapiro-Wilk normality test and a significance level of $\alpha = 0.05$; effect sizes were reported using Cohen's d for paired differences (mean difference divided by the standard deviation of the differences) [15], [16], [17]. Because both hypotheses are directional (H1 and H2), one-tailed p-values are reported, and 95% confidence intervals (CI) of the mean paired difference are given. To contextualize the small sample, a sensitivity analysis was performed with the noncentral t distribution: a one-tailed paired t-test with n = 10 and $\alpha = 0.05$ reaches a power of 0.80 only for effects of $d \geq 0.85$.

III. RESULTS AND DISCUSSION

A. Training Results and Strengths of the Base Model

A total of 75 training iterations were obtained. In iterations 1–10, the policy loss decreased from 1.9420 to 1.0114, and the value loss decreased from 0.3020 to 0.0030. Iterations 59–72 showed stable convergence. The final model values were a policy loss of 0.8120 and a value loss of 0.0589. These results serve as evidence that the model is sufficiently stable to serve as the base model for AlphaDDA.

DOI: 10.34148/teknika.v15i3.1562
TEKNIKA, Volume 15(3), November 2026, pp. 497-503
ISSN 2549-8037, EISSN 2549-8045

At the end of 50 round-robin tournaments, Minimax achieved an Elo rating of 1,738.73, AlphaZero 1,729.31, MCTS 1,422.09, and Random 1,109.87 (Figure 2 and Table 3). The difference between AlphaZero and Minimax was only 9.42 points, while AlphaZero outperformed pure MCTS by 307.22 points. These results show that the AlphaZero Congklak model is competitive with Minimax and stronger than both MCTS and Random.

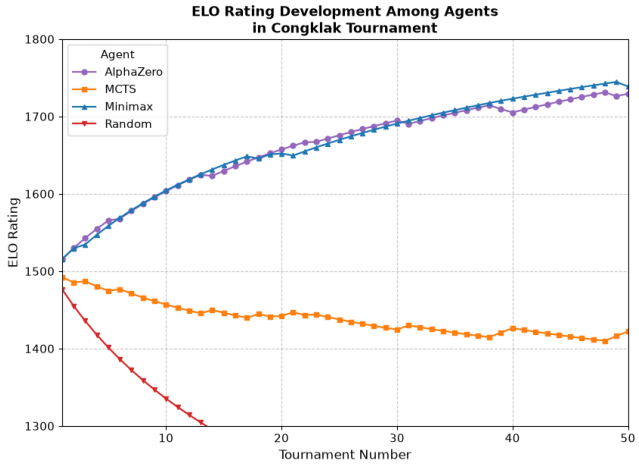


Figure 2. Changes in Elo Ratings Over the Course of 50 Round-Robin Tournaments (the Random Agent Falls Below the Plotted Range After Approximately Tournament 13)

Table 3. Final Elo Ratings of the Congklak Agents

Agent	Elo
Minimax	1738.73
AlphaZero	1729.31
MCTS	1422.09
Random	1109.87

Note: Elo is a relative strength rating (higher is stronger). Minimax = classical adversarial search; AlphaZero = the trained Congklak model; MCTS = Monte Carlo Tree Search without a neural network; Random = random legal moves.

B. Digital Congklak Game Artifact

The Congklak board game was developed in Unity as a testbed for AlphaDDA. The implementation preserves the mechanisms for initialization, piece distribution, capture, extra turns, game termination, and score calculation. Unit testing using the Unity Test Framework ran seven game logic test cases, and all of them passed.

The visual adaptation retains the wooden Congklak board and sea snail shells as references. In the digital version, physical interactions are transformed into haptic feedback, the rustling sound of the shells, and visual animations. Figure 3 is the reference image, while Figure 4 shows the digital adaptation of the board and pieces.

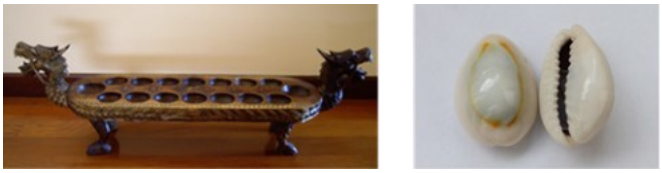


Figure 3. A wooden Congklak Board and Sea Snail Shells



Figure 4. A Digital Adaptation of the Congklak Board and Its Pieces

C. Gameplay Metrics Experiment Results

The experiment involved 10 participants. Three participants were 21 years old, and seven were 22 years old; all were male. Five participants had played Congklak before, and five had not. The Shapiro-Wilk test (Table 4) showed that Δscore ($W = 0.940$; $p = 0.551$) and playtime ($W = 0.971$; $p = 0.896$) were normally distributed ($p > 0.05$), so a paired samples t-test was used. Descriptive statistics for both conditions are given in Table 5, and the t-test results in Table 6.

Table 4. Shapiro-Wilk Test of Normality

Metric	W	p	Decision
Δscore	0.940	0.551	Normal
playtime	0.971	0.896	Normal

Note: W = Shapiro-Wilk test statistic (values close to 1 indicate normality); $p > 0.05$ means that the data do not deviate significantly from a normal distribution.

Table 5. Descriptive Statistics for the DDA and Non-DDA Conditions

Metric	DDA, Mean (Standard Deviation)	Non-DDA, Mean (Standard Deviation)
Δscore	11.80 (4.71)	26.00 (15.37)
playtime	164.70 (50.10)	197.35 (106.21)

Note: DDA = condition with AlphaDDA; Non-DDA = control condition with static AI. Δscore is in points and playtime in seconds.

Table 6. Results of the Paired-samples t-test (DDA vs. non-DDA)

Metric	t(9)	p (One-Tailed)	Cohen's d
Δscore	-2.762	0.011	-0.87
playtime	-0.874	0.798	-0.28

Note: t(9) = paired t statistic with 9 degrees of freedom; p = one-tailed p-value; d = Cohen's d for paired differences. Negative values indicate lower values in the DDA condition.

In the DDA group, the mean Δscore was 11.80 points (SD = 4.71), whereas in the non-DDA group it was 26.00 points

(SD = 15.37) (Figure 5). The mean paired difference was −14.20 points (95% CI [−25.83, −2.57]). This difference is statistically significant (t(9) = -2.762; one-tailed p = 0.011; two-tailed p = 0.022; d = -0.87) with a large effect size. Thus, H1, the hypothesis that the mean score difference in the DDA condition is smaller than in the non-DDA condition, is supported.

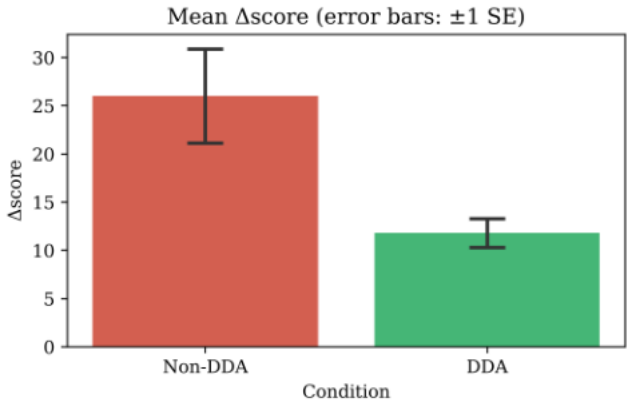


Figure 5. Mean Score Difference (Δscore) Under Non-DDA and DDA Conditions (Error Bars: ±1 Standard Error)

For playtime, the DDA condition had a mean of 164.70 seconds (SD = 50.10), while the non-DDA condition had a mean of 197.35 seconds (SD = 106.21) (Figure 6). The mean paired difference was −32.65 seconds (95% CI [−117.16, 51.86]). The difference was not statistically significant (t(9) = -0.874; one-tailed p = 0.798; d = -0.28). H2, the hypothesis that playtime in the DDA condition was longer than in the non-DDA condition, was not supported; the observed difference was in the opposite direction.

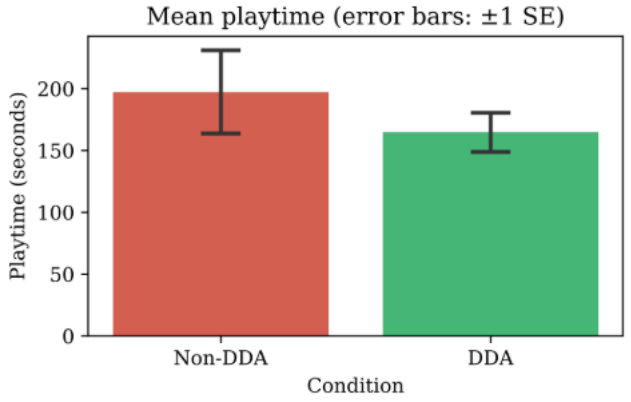


Figure 6. Mean Playtime Under Non-DDA and DDA Conditions (Error Bars: ±1 Standard Error)

D. Discussion

Integration of AlphaDDA into the AlphaZero architecture. As shown in Figure 1 (right), AlphaDDA requires no retraining of the AlphaZero model. The trained policy–value network is reused unchanged as the evaluator inside MCTS, and the AlphaDDA module is inserted between the canonical state and the search. It reads the value estimate of the current state (Nh = 1), converts it into a simulation budget Nsim bounded by Nmax = 300, and passes it to MCTS, which then selects the best move for that turn. The value head, which in AlphaZero only guides the search, thereby also serves as an estimate of which side is currently ahead, and difficulty is changed solely through search effort.

The significance of Δscore can be explained by the AlphaDDA mechanism. When the value indicates that the AI is gaining an advantage, the number of MCTS simulations is reduced, causing the quality of the search to decline and the agent to tend to choose more suboptimal moves. Conversely, when the AI falls behind, the number of simulations increases toward Nmax, causing its playing strength to return to that of the baseline model. This mechanism minimizes the difference in effective ability between the AI and its opponent from turn to turn.

The lower descriptive variability of Δscore in the DDA condition (SD = 4.71) compared with the non-DDA condition (SD = 15.37) is consistent with a more stable balancing effect across the tested participants. These findings support the use of score differences as an indicator of game balance in competitive games [18], [19].

The lack of significance in playtime does not mean that the AlphaDDA mechanism has failed. AlphaDDA directly manipulates Objective Game Difficulty (OGD) through the quality of the AI's decisions, whereas playtime—as a proxy for engagement—is more closely tied to players' cognitive processes and is not directly manipulated. Therefore, the effect on Δscore can be significant without accompanying changes in playtime [5], [17].

Playtime must be interpreted cautiously as an engagement indicator. Accumulated think time is an unobtrusive behavioural measure, but it is ambiguous: longer deliberation may reflect interest and effort, but also confusion or excessive difficulty, whereas shorter deliberation may reflect fluency as well as boredom. It does not capture enjoyment, immersion, or motivation, which require self-report instruments or richer behavioural logs [15], [17]. The observed mean difference was −32.65 seconds (95% CI [−117.16, 51.86]), opposite to the direction of H2, and the interval is wide enough to include both considerably shorter and longer playtimes under DDA. With n = 10, only effects of d ≥ 0.85 could be detected with a power of 0.80, whereas the observed effect was d = −0.28, for which the power is roughly 0.20. The playtime result should therefore be read as an absence of evidence for an engagement effect, not as evidence that AlphaDDA leaves engagement unchanged.

E. Limitations

Several limitations should be considered when interpreting these results. First, the sample is small (n = 10), was recruited by convenience sampling, and is homogeneous (all male, aged 21–22 years); the results cannot be generalized to women, other age groups, or players with other levels of Congklak experience, and the five experienced and five inexperienced players could not be analysed separately. No a priori power analysis was performed, and the sensitivity analysis shows that only large effects (d ≥ 0.85) were detectable; the Δscore effect

DOI: 10.34148/teknika.v15i3.1562
TEKNIKA, Volume 15(3), November 2026, pp. 497-503
ISSN 2549-8037, EISSN 2549-8045

(d = -0.87) lies just above this threshold, so its magnitude may be overestimated and should be considered preliminary. The effect remains significant under a two-tailed test (p = 0.022). Second, the evaluation was limited to a single session per participant, so long-term effects on balance and engagement were not examined. Third, engagement was measured only through playtime, without subjective measures such as questionnaires.

Fourth, the AlphaDDA parameters (Asim = 1.5, X0 = -1.75, Nmax = 300) were determined through manual tuning, without a grid search or a robustness analysis. Because these parameters determine how strongly the simulation budget responds to the value estimate, the reported balancing effect applies to this configuration only, and the sensitivity of the results to different parameter values remains unknown. Nmax was also set with the computing limits of mobile devices in mind, which may restrict the maximum strength of the agent. Future studies should therefore use larger and more diverse samples with an a priori power analysis, systematically explore the AlphaDDA parameters, and combine telemetry with validated engagement questionnaires.

IV. CONCLUSION

The AlphaDDA method was integrated into the AlphaZero architecture for the game of Congklak by dynamically adjusting the number of MCTS simulations based on the evaluation values generated by the value head (Figure 1). The base model was obtained through 75 iterations of self-play training, which proceeded stably until reaching a final policy loss of 0.8120 and a value loss of 0.0589. In an Elo-based evaluation of playing strength, the AlphaZero Congklak agent achieved an Elo rating of 1729.31, which was competitive with the Minimax agent (1738.73 Elo, a difference of 9.42 points) and outperformed a pure MCTS without a neural network (1422.09 Elo) by 307.22 points. The AlphaDDA module uses the following configuration, determined through manual tuning: Asim = 1.5, X0 = -1.75, Nmax = 300, and Nh = 1. In this sample of ten participants, AlphaDDA was effective for game balancing in the Congklak board game: a paired samples t-test showed that it significantly narrowed the final score gap ($t(9) = -2.762$; one-tailed $p = 0.011$; $d = -0.87$), from an average of 26.00 points in the non-DDA condition to 11.80 points in the DDA condition, representing a large effect size. Playtime did not show a statistically significant difference in means ($t(9) = -0.874$; $p = 0.798$; $d = -0.28$); given the limited statistical power and the fact that playtime is only a behavioural proxy, this indicates no evidence of an engagement effect rather than evidence that engagement is unaffected. The findings are preliminary because of the small, homogeneous convenience sample and the manually tuned parameters. Future research is recommended to (1) replicate the experiment with larger and more diverse samples, (2) analyse the robustness of AlphaDDA to its parameters (Asim, X0, and Nmax), (3) measure engagement with validated questionnaires in addition to telemetry, and (4) optimize the computational efficiency of MCTS on mobile platforms; Nmax = 300 in this study was found to be a safe limit to prevent devices from experiencing a drop in CPU performance (frame drops) while the AI agent is thinking.

REFERENCES

[1] S. Russell and P. Norvig, "Artificial Intelligence: A Modern Approach, Global Edition 4th," *Foundations*, vol. 19, 2021.

[2] D. Silver *et al.*, "A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play," *Science (1979).*, vol. 362, no. 6419, pp. 1140–1144, Dec. 2018, doi: 10.1126/science.aar6404.

[3] G. N. Yannakakis and J. Togelius, "Artificial Intelligence and Games," *Artificial Intelligence and Games*, pp. 1–525, Jan. 2025, doi: 10.1007/978-3-031-83347-2.

[4] K. Fujita, "AlphaDDA: strategies for adjusting the playing strength of a fully trained AlphaZero system to a suitable human training partner," *PeerJ Comput. Sci.*, vol. 8, 2022, doi: 10.7717/PEERJ-CS.1123.

[5] Z. Guo, R. Thawonmas, and X. Ren, "Rethinking dynamic difficulty adjustment for video game design," *Entertain. Comput.*, vol. 50, p. 100663, May 2024, doi: 10.1016/j.entcom.2024.100663.

[6] T. Fullerton, "Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Fifth Edition," *Game Design Workshop: A Playcentric Approach to Creating Innovative Games, Fifth Edition*, pp. 1–554, Jan. 2024, doi: 10.1201/9781003460268.

[7] A. Becker and D. Görlich, "What is Game Balancing? - An Examination of Concepts," *ParadigmPlus*, vol. 1, no. 1, pp. 22–41, Apr. 2020, doi: 10.55969/PARADIGMPLUS.V1N1A2.

[8] R. Suharti, K. P. Bhatt, O. Y. Tolulope, and Y. A. Sambo, "Exploration Ethnomathematics in Traditional Games of Mancala in Africa and Congklak in Indonesia," *Interval: Indonesian Journal of Mathematical Education*, vol. 3, no. 1, pp. 45–53, May 2025, doi: 10.37251/IJOME.V3I1.1594.

[9] K. D. Ramadani, R. Agustina, R. Sulistyowati, and S. W. Nugroho, "Statistik Sosial Budaya: Hasil Survei Sosial Ekonomi Nasional Modul Sosial Budaya dan Pendidikan (Susenas MSBP) 2024," *Badan Pusat Statistik*, vol. 13, no., p. 256, 2025, Accessed: Jan. 27, 2026. [Online]. Available: https://www.bps.go.id/id/publication/2025/05/28/1b5cdf50f49c754c3b581bd2/statistik-sosial-budaya-2024.html

[10] K. Peffers *et al.*, "Design Science Research Process: A Model for Producing and Presenting Information Systems Research," Jun. 2020, Accessed: Jul. 05, 2026. [Online]. Available: https://arxiv.org/pdf/2006.02763

[11] P. M. J. Delport, R. Von Solms, and M. Gerber, "Methodological Guidelines for Design Science Research," *Procedia Comput. Sci.*, vol. 237, pp. 195–203, Jan. 2024, doi: 10.1016/J.PROCS.2024.05.096.

[12] G. Charness, U. Gneezy, and M. A. Kuhn, "Experimental methods: Between-subject and within-subject design," *J. Econ. Behav. Organ.*, vol. 81, no. 1, pp. 1–8, Jan. 2012, doi: 10.1016/J.JEBO.2011.08.009.
[13] J. Ho and J. Min, "Causal Inference in Counterbalanced Within-Subjects Designs," 2025.
[14] E. Pagalyte, M. Mancini, and L. Climent, "Go with the Flow," in *Proceedings of the 20th ACM International Conference on Intelligent Virtual Agents*, New York, NY, USA: ACM, Oct. 2020, pp. 1–8. doi: 10.1145/3383652.3423868.
[15] A. Drachen, P. Mirza-Babaei, and L. E. Nacke, "Games User Research," *Games User Research*, pp. 1–524, Mar. 2018, doi: 10.1093/oso/9780198794844.001.0001.
[16] C. Wohlin and A. Aurum, "Towards a decision-making structure for selecting a research design in empirical software engineering," *Empirical Software Engineering 2014 20:6*, vol. 20, no. 6, pp. 1427–1455, May 2014, doi: 10.1007/s10664-014-9319-7.
[17] M. S. El-Nasr, T. H. Nguyen Dinh, A. Canossa, and A. Drachen, "GAME DATA SCIENCE," *Game Data Science*, pp. 1–397, Jan. 2021, doi: 10.1093/oso/9780192897879.001.0001.
[18] A. Darzi, S. M. McCrea, and D. Novak, "User Experience With Dynamic Difficulty Adjustment Methods for an Affective Exergame: Comparative Laboratory-Based Study," *JMIR Serious Games*, vol. 9, no. 2, p. e25771, Apr. 2021, doi: 10.2196/25771.
[19] A. Stein, Y. Yotam, R. Puzis, G. Shani, and M. Taieb-Maimon, "EEG-triggered dynamic difficulty adjustment for multiplayer games," *Entertain. Comput.*, vol. 25, pp. 14–25, Mar. 2018, doi: 10.1016/j.entcom.2017.11.003.

DOI: 10.34148/teknika.v15i3.1562
CC BY SA