Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (101-110)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Adaptive Hint Generation for Educational Games Using Fuzzy Logic

Anggina Primanita[1], Hadipurnawan Satria[2], Muhammad Qurhanul Rizqie[3], Ananda Haykel Iskandar[4], Wibisena Nugraha[5]

[1,2,3,4,5]Informatics Department, Computer Science Faculty, Sriwijaya University
[1,2,3,4,5]Jalan Raya Palembang-Prabumulih KM. 32 Indralaya, Ogan Ilir, Sumatera Selatan, Indonesia

Article:
Accepted: January 15, 2025
Revised: September 12, 2024
Issued: April 30, 2025

***Correspondence Address:**
anggina.primanita@ilkom.unsri.ac.id

ABSTRACT

The increasing interest in programming education has led to a wide variety of learner abilities. However, existing learning media often remain fragmented, necessitating the development of adaptive tools to cater to learners of varying skill levels. This study employs fuzzy logic to generate dynamic hints for players struggling to solve programming challenges in an educational game. The effectiveness of the system was evaluated through both simulation and real-world experiments. Simulation results indicate that the fuzzy logic system successfully generates personalized hints, with the highest frequency of hints provided to beginner players. Real-world testing using the GUESS-18 framework demonstrated high playability and excellent usability scores for the game.

Keywords : *educational game; adaptive game; programming logic; computational thinking;fuzzy logic.*

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (101-110)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

1. INTRODUCTION

Programming is the process of designing and writing code to solve specific problems. As a fundamental skill for computer science professionals [1], programming skills are becoming increasingly essential in various fields. In addition to practitioners in informatics, individuals from diverse disciplines now require logical thinking abilities to address complex problems.

As demand for programming education grows, there is a corresponding increase in the variety of learner abilities. To address this, video games—once primarily a source of entertainment—have increasingly been adopted in educational contexts, including subjects such as mathematics [2], biology[3], and programming[4][5]. Educational games cater to learners across age groups, from early childhood [6] to adulthood[7], [8].

Silva et al. highlight the positive effects of educational games, though challenges remain, particularly in adapting games to meet diverse pedagogical and cultural needs [5]. Educational games often offer benefits only within specific learning contexts, but adaptive games—games that adjust to players' abilities—have emerged as a solution to this issue.

For programming education, adaptive games have been developed with varying adaptive elements, one of which is the provision of hints. McBroom, Koprinska, and Yacef's HINTS framework [9] outlines four core processes for generating adaptive assistance in games: input, processing, data set augmentation, and output.

While adaptive educational game aids are typically available in English, this limits accessibility for learners in non-English-speaking regions. Providing hints in Bahasa Indonesia is anticipated to make programming education more accessible to Indonesian learners.

To develop the game, we first designed a framework that integrates the HINTS Framework with a Fuzzy Logic System. This framework serves as the foundation for Fuzzy LightBot, a programming game designed to provide adaptive hints in Bahasa Indonesia, tailored to players' skill levels. To assess the game's effectiveness, we conducted two evaluations. First, we simulated the model's output to analyze its adaptive hint generation. Second, we gathered player feedback using a questionnaire based on the GUESS-18 framework, measuring user experience and satisfaction.

This research seeks to answer two key questions:

1. How can adaptive hints in Bahasa Indonesia be generated for educational programming games?
2. How do the generated hints influence players' understanding of programming concepts?

The novelty of this research lies in the the **creation of the integrated fuzzy model, called the Fuzzy Adaptive Hint Generator (FAHG), to provide adaptive hint in Bahasa Indonesia based on players' skill models,** designed to enhance programming education through games.

This manuscript is the original result of our research using our own idea.

This section explores relevant literature on serious games, HINTS Framework, and the evaluation framework employed in this research.

1.1. Serious Games

Serious games are defined as games designed for a primary purpose other than entertainment [10]. Wilkinson divides the purpose of play, especially for children, into three main groups: educational, therapeutic, and social control [11]. Education through play is a play activity whose main purpose is to learn something previously unknown or not yet mastered. Play as therapy means that player is expected to feel calm when playing. Finally, play activities can be used as a place to convey social messages that will then be understood and absorbed by the player.

Education through games has been done to enhance players' understanding of a specific topic, such as health-related topics [12] or programming [4] [5]. The demand for education through serious games comes from the necessity of delivering the most effective learning strategies for players. Innovations are made to ensure that player preferences are taken into account and players can still have fun while playing[11].

1.2. HINTS Framework

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (101-110)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

Automated tutoring systems offer the flexibility and scalability needed to facilitate the provision of high-quality programming education and make it accessible to more people. The number of people accessing an automated system increases the variation in the capabilities of its users. For this, McBroom, Koprinska, and Yacef introduced a help system framework that is adaptive to the needs of its users [9].

The "Hint Iteration by Narrow-down and Transformation Steps (HINTS)" frame work is a framework that can be used to generate adaptive help for users. This framework in general can be used in all educational systems, whether it is a tutoring systems, gamified systems, or games in general.

HINTS framework is divided into 4 main steps, namely:

1. Input: The input of the process is a set of data in the form of help (hint pool) that will be provided to the user.
2. Process: The hint data is processed based on two primary techniques: narrowing down with relevance criteria and transforming the existing representation in the hint pool into new representation.
3. Addition: Newly generated hint data is incorporated into the hint pool.
4. Output: One or more hints are presented to the user.

1.3. Fuzzy Logic

Fuzzy logic is the concept of set where an object has a truth value between 0 and 1. Fuzzy logic can be used for decision making in various fields, including games. The decision making process in fuzzy logic is carried out through three main stages, namely fuzzification, inference, and defuzzification [13].

Fuzzification is the process of mapping an input in form of a crisp value into a membership degree value in a fuzzy set. This mapping is done based on the membership function of the fuzzy set. There are several forms of membership functions that are often used, namely, triangles, trapezoids, and Gaussian curves [14].

Rules in fuzzy logic are used to model human thinking and judgment. Generally, the rules in a fuzzy logic system follows the form of "if-then". For example: "if x is A then y is B" where A and B are predefined linguistic values. In the previous rule, the "x is A" is called the antecedent, while "y is B" is called the consequent.

Fuzzy inference is the process of inference from the collection and correlation between rules in a fuzzy logic system. In this phase, fuzzy logic rules are applied based on the membership values obtained in the fuzzification phase [13].

The Mamdani inference system consists of four stages, namely fuzzification, rule evaluation, output aggregation, and defuzzification. The fuzzification process is carried out by determining the membership degree of the inference system input in the form of a firm (crisp) value. The next step is rule evaluation, where the membership degree value is applied to the antecedent. If a fuzzy rule has more than one antecendents, the rules are combined with the fuzzy AND or OR operators. The final value of this process is called the truth value. The truth value is then applied to the consequent membership degree. Each consequent value is then aggregated into a field which will then go through the defuzzification stage.

1.4. Game User Experience Satisfaction Scale-18 (GUESS-18)

Playtesting is a testing activity conducted to help game developers improve or create a good quality game. However, quality feedback is not easily obtained from playtesting sessions that do not use good assessment tools. The Game User Experience Satisfaction Scale (GUESS) is an instrument developed to meet the needs of such assessment tools [15]. It was developed based on the System Usability Scale (SUS).

In its development, GUESS, which has 55 questions is considered too complex. For this reason, GUESS-18 has been developed, which is a game experience assessment instrument consisting of only 18 questions.

Each question on the GUESS-18 has a closed response on a Likert scale with the lowest value of 1 and the highest value of 7.

The questions in GUESS-18 can be randomized or presented according to Table 1. The value of each category is averaged to get the score of each category which can then be summed to get a combined score. There is one question that has a REVERSE CODE, and this question is calculated in reverse (9-value Likert scale).

Table 1. *GUESS-18 questions*

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (101-110)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

Construct	Question
Usability/Playability	*I find the controls of the game to be straightforward.*
	I find the game's interface to be easy to navigate.
Narratives	*I am captivated by the game's story from the beginning.*
	I enjoy the fantasy or story provided by the game.
Play Engrossment	*I feel detached from the outside world while playing the game.*
	I do not care to check events that are happening in the real world during the game.
Enjoyment	*I think the game is fun.*
	I feel bored while playing the game. (REVERSE CODE)
Creative Freedom	*I feel the game allows me to be imaginative.*
	I feel creative while playing the game
Audio Aesthetics	*I enjoy the sound effects in the game.*
	I feel the game's audio (e.g., sound effects, music) enhances my gaming experience.
Personal Gratification	*I am very focused on my own performance while playing the game.*
	I want to do as well as possible during the game.
Social Connectivity	*I find the game supports social interaction (e.g., chat) between players.*
	I like to play this game with other players.
Visual Aesthetics	*I enjoy the game's graphics.*
	I think the game is visually appealing.

According to research by Keebler et al., GUESS-18 is a validated instrument and serves as an effective model for measuring player satisfaction[16]. Compared to the original GUESS, GUESS-18 only takes 3-5 minutes to complete and can be used as a companion instrument if there are other measurements at the same time.

2. METHODS

This section details the methodologies employed in the development of the educational game and the framework used to generate adaptive hints.

The research, as well as the development of the educational game is carried out following the Game Development Life Cycle (GDLC). The phase of GDLC is shown in Figure 1.

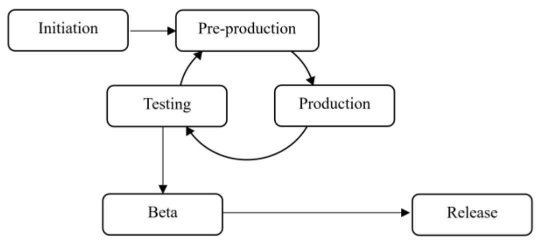


Figure 1. *Research methodology*

The development process began with a literature review, which informed the decision to create a game designed to assist players through adaptive hints.

In the pre-production phase, the Fuzzy Adaptive Hint Generator (FAHG) model was developed, and the game was designed.

The production phase focused on implementing and developing the Fuzzy LightBot Game. The effectiveness of the design was evaluated through three distinct assessments. First, a simulation test was conducted to analyze the performance of the hint generator. Second, a black box testing approach was used to assess the functionality of the Fuzzy LightBot software. Finally, a user evaluation was carried out using the GUESS-18 questionnaire, which is based on the System Usability Scale (SUS), to measure how effectively the generated hints supported player learning.

2.1. Fuzzy Adaptive Hint Generator

To evaluate the concept of fuzzy logic being an adaptive generator, a software has been developed based on the HINTS frame work. We modify the framework to fit the fuzzy inference system in the processing part. The framework of FAHG displayed in Figure 2 shows the four main steps of generating hint in the game.

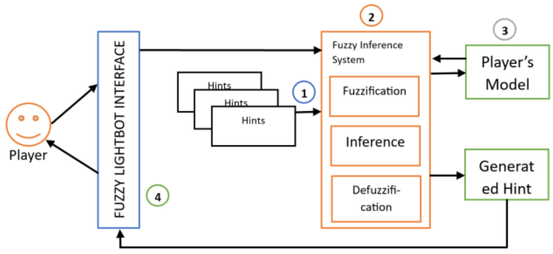


Figure 2. *Fuzzy adaptive hint generator (FAHG) framework*

The first step is Input, which consists of the player's in-game interactions and a categorized hint data pool designed to provide tailored assistance based on players' varying skill levels.

The second step in FAHG is Processing. In this stage, the system utilizes a Fuzzy Inference System (FIS) to process the hint data, using the inputs from the first step. This processing generates both the output and an updated player model.

The third step is Addition. Here, the player model produced by the fuzzy inference

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

system is stored and used as a basis for refining and enhancing the inference system over time.

The fourth step is Output, where the system presents the generated hint to players who fail to solve the given problem.

2.2. Fuzzy LightBot Design

A software application based on the Fuzzy Adaptive Hint Generator, named Fuzzy LightBot, was developed using the open-source framework Another-LightBot [17]. Additional features include fuzzy logic, integrated hints, and automatic player data storage.

The Use Case of this software is displayed in Figure 3.

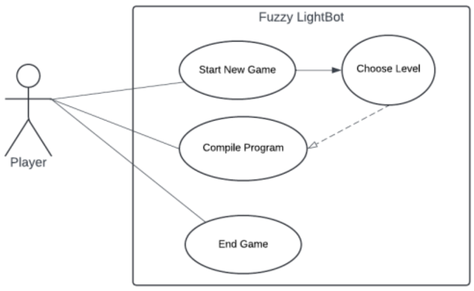


Figure 3. *Use case diagram of fuzzy lightbot*

The description of each Use Case is described in Table 2.

Table 2. *Description of fuzzy lightbot use case*

Id	Use Case	Description
UC-1	Start New Game	Player starts a new game and enters their username. System will then show the level choice screen.
UC-2	Choose Level	Player chooses the level that they will play. There are currently 5 levels available. System will show puzzle corresponds to the player's choice.
UC-3	Compile Program	Player build a program by entering the operation that will be followed by the robot. If all the target platform has been light up, the player can continue to the next level. If not, a hint will be displayed based on Players' model
UC-4	End Game	Player closes and ends the game.

The HFLS was developed to determine hints using three antecedents (average stars, completion, time) and one consequent (player's ability). The antecedents of HFLS are average stars, completion, and time. These three inputs are taken because they represent the progress of the Player in the LightBot game and thus represent the player's model.

The average star antecedent is the measurement result of the program that is built by the player. The fewer operations needed to complete the level, the higher the star the player gets at a level. Average stars are the average stars that players get at the previous level. Completion is the percentage of the player's progress compared to the most optimal solution of a level. Time is the time elapsed between when the level starts or the player resets the level until the player compiles the wrong solution.

The HFLS has one consequent which is the measure of player expertise. Players are divided into 3 skill levels namely newbie, intermediate, and expert. Membership functions are shown in Figure 4.

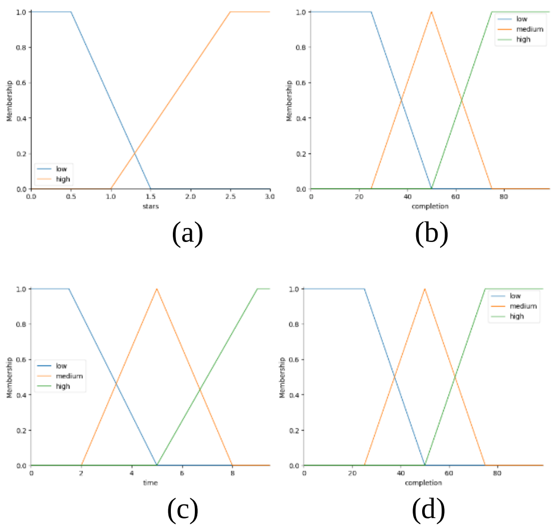


Figure 4. *Membership function of (a) average stars. (b) completion percentage, (c) time elapsed, and (d) players' ability*

After determining the membership function of the Antecedent and Consequent HFLS, the rules of HFLS are determined based on the player's ability. Eighteen rules were defined linking input levels (low, medium, high) to the player's ability (newbie, intermediate, expert.

3. RESULTS AND DISCUSSION

In this section, we present the results of the developed software and the evaluation of the implemented framework.

3.1. Hint Simulation Result

Simulations (1,000 iterations per level, 5,000 total) showed that Level 1 generated 881 newbie and 119 intermediate hints, with no expert hints as seen in Figure 5.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (101-110)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

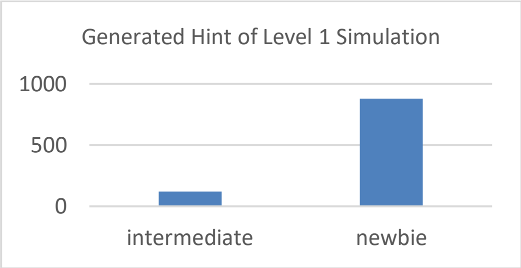


Figure 5. *Histogram of generated hint in level 1*

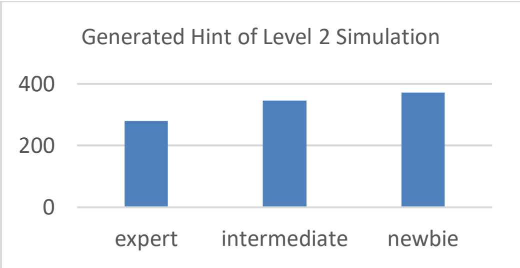


Figure 6. *Histogram of generated hint in level 2*

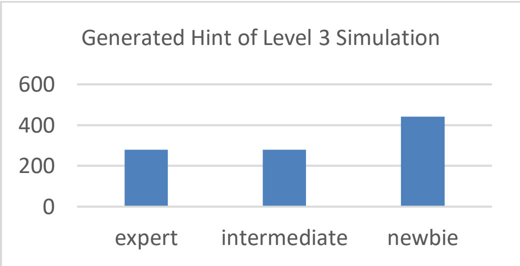


Figure 7. *Histogram of generated hint in level 3*

Levels 3–5 had varied outputs—with newbie hints most frequent overall—confirming that lower-ability players receive more assistance (see Figures 6–8).

To show the trend of generated hints, we combine all of the numbers into Figure 13. It can be seen that overall, the highest count of hints is generated for newbie player while the lowest count is the expert hint.

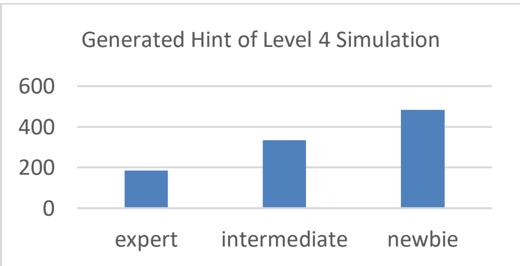


Figure 8. *Histogram of generated hint in level 4*

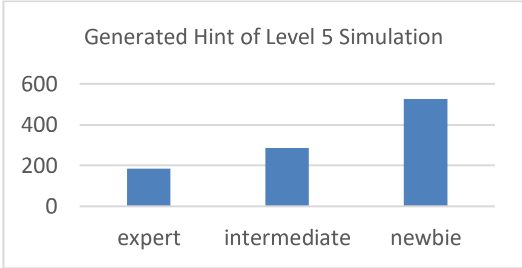


Figure 9. *Histogram of generated hint in level 5*

This in in line with the aim of the system. In which, newbie player will require the most assistance and given the most, while expert players will be able to play the game with least assistance and given the least. It can also be seen that hint generated for intermediate player maintained its numbers for level 2 to 4, which means that the intermediate player can utilize the hint when needed, but will not get too much hint that they will not be able to use their programming logics.

3.2. Fuzzy LightBot Software

Fuzzy Lightbot is a software that was developed as an implementation of the HFLS framework. The software is developed using Unity IDE and designed to run in Windows Platform.

The game starts with the player entering a username, which also serves as their id, then selecting a level where puzzles increase in complexity (see Figure 10).

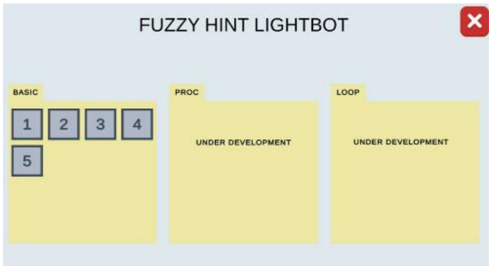
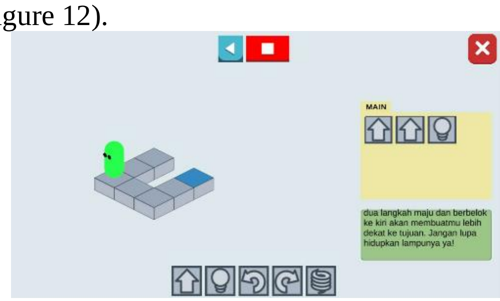


Figure 10. *Level selection screen*

An incorrect solution triggers hint generation based on the player's ability (Figure 11); a correct solution allows progression (see Figure 12).



Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (101-110)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.41893

Figure 11. *Generated Hints in Fuzzy LightBot*

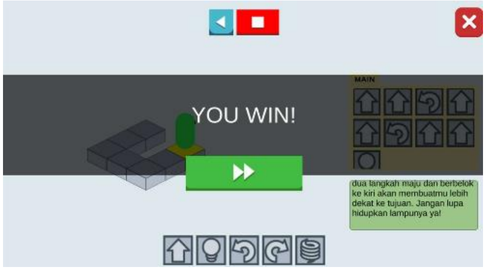


Figure 12. *Level progression screen*

The game ends when the player exits the application. When the game ended, the software will export the player's data into a comma separated value (.csv) file that can be used to analyze their performance.

Black Box testing was employed to evaluate the software. There were 4 categories of testing that cover each of the use case. Based on the evaluation, the software has fulfill all of the requirements.

3.3. GUESS-18 Result

A playtest with 17 students (aged 17–20) involved a preliminary questionnaire, game play testing from levels 1–5, and a post-playtest GUESS-18 survey was held in a laboratory setting.

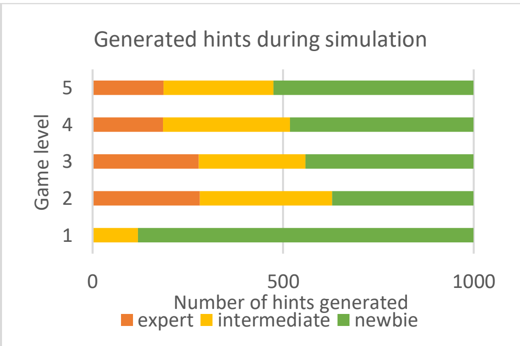


Figure 13. *Histogram of generated hints of all levels*

The result from the preliminary questionnaire is the demography and the ability of the students to grasp programming concepts. Based on the questionnaire, it was found that only 1 (5.9%) of the students answered 7 on a scale of 1-7. This means that only one student is very confident in their programming skill. The rest of the students answered between 2-5, with most students answering 3. This means that they perceive their programming ability as being less than average. The questionnaire also contains a question about the students' opinion of the programming class. The students' answers vary between 3 to 7 on a scale of 7, with most answering 5 and 7. This shows that the students are interested in learning programming.

Following the first session, the main playtesting session is conducted. In this session, we were able to obtain some data related to the students performance in the game. Based on the data we collected, we found that 10 out of 17 students (58.89%) needed hints in various levels to finish the game. The result of the hint generated for level 1 to level 5 is displayed in Figure 14.

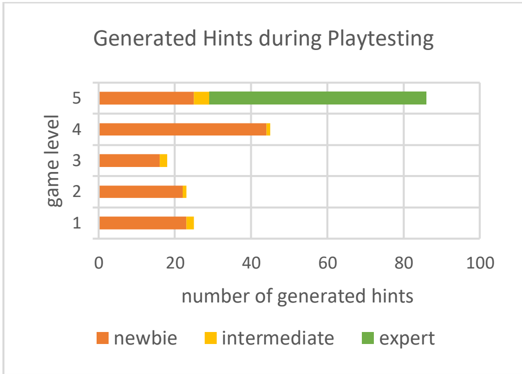


Figure 14. *Hint generated during playtesting session*

Results from the playtesting session show that most of the students are getting hint generated for newbie. This happened the most during the first four levels. In level 5, however, the students are getting significantly more expert hints. According to the obtained data, this is due to the higher value of the completion rate when the students are submitting their solutions. This shows that students are getting better at composing their solutions.

Results showed that most students required newbie hints initially, with Level 5 generating more expert hints. The overall GUESS-18 score was 49.206/63, rating the game's playability as Good (see Figure 15).

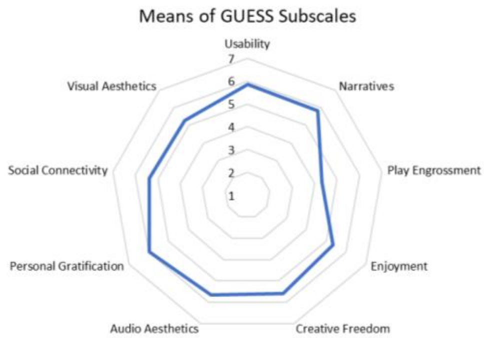


Figure 15. *Means of GUESS-18 subscales based on playtesting session*

During the last session, we also ask some questions specific to programming experience and hints while playing the game. The first question is asking about whether the hint is helping in finishing and understanding the puzzle. In this question, 7 out of 10 students who had to use the hint answered that they very agree that the hint helped them.

3.4. Discussion

A simulation test done to evaluate the Hint Fuzzy Logic System (HFLS) has been conducted. To test the variation of the output in HFLS, simulations were carried out at each level in the software and for each level, 1000 simulations were conducted.

At the first level, the simulation results show that HFLS provides only two types of input, namely newbie and intermediate hints. This is due to the absence of variation in the input. At this stage, the player has never played the game so the average stars value is always equal to 0.

Results from simulation in the second to fourth levels provide more varied outputs with different distributions. At these levels, expert output can already be generated by HFLS because there is already input in the form of average stars obtained at the previous level. At this level, the output with the largest percentage is the hint generated for newbie players.

At level five, the simulation provides the same variety of outputs, but with different distributions. The simulation results at level five have the largest distribution of hints for newbies compared to levels 2-4. Since level five is the most complex level, players with the lowest ability will need hint more often than players with higher ability levels. This is in line with the general demographics of players, where newbie players will ask for help more often than intermediate or expert players.

A playtesting session was conducted to evaluate real player reactions to the game and its hint feature. The results from playtesting differed from those observed in the simulation. While player behavior in the first four levels was similar to the simulation, a notable difference emerged in the fifth level. During playtesting, players required more expert-level hints, whereas the simulation had predicted a need for newbie-level hints. This discrepancy arose because, unlike the simulation, human players were able to learn and adapt throughout the gameplay. As a result, they leveraged their accumulated knowledge and only needed assistance to solve the final part of the puzzle rather than requiring hints for the entire level.

The simulations confirmed that HFLS adapts outputs based on player inputs, with newbie hints dominating especially in complex levels. In contrast, playtesting revealed more expert hints at Level 5, suggesting that human players leverage prior knowledge. Overall, players found the game playable (GUESS-18 rated Good), though some viewed it more as a learning tool. The students may still see the game as a learning platform instead of a pure gaming experience. In the last questionnaire, it was also found that the students find the hint to be helpful and match their ability level.

CONCLUSION

Fuzzy LightBot was developed to adapt to varying player abilities using a fuzzy logic hint system. Simulations demonstrated adaptive hint generation, and playtesting confirmed the game’s playability and hint effectiveness. Future enhancements direction is towards boosting engagement. One of the technology that can be utilized to increase engagement is Augmented Reality[18][19]. As such, in the future, we plan to include addition of features like an Augmented Reality play. Another direction is to add more levels to the game with higher complexity to give player more challenges.

ACKNOWLEDGEMENT

The research of this article was funded by DIPA of Public Service Agency of Universitas Sriwijaya 2024 Nomor SP DIPA 023.17.2.677515/2024, On November 24, 2023. In accordance with the Rector’s Decree Number: 0013/UN9/LP2M.PT/2024, On May 20, 2024

REFERENCES

[1] A. Arief, M. Muhammad, and F. Amin, "Development Of Educational Game Using GDLC Method: A Case Study Of Algorithm And Data Structures Course," *IJIS Indonesian Journal on Information System*, vol. 8, no. 2, pp. 120–125, 2023.

[2] J. Y. Arosquipa Lopez, R. N. Nuñoncca Huaycho, F. I. Yallercco Santos, F. T. Mendoza, and F. H. Rucano Paucar, "The Impact of Serious Games on Learning in Primary Education: A Systematic Literature Review," *International Journal of Learning, Teaching and Educational Research*, vol. 22, no. 3, pp. 379–395, Mar. 2023, doi: 10.26803/ijlter.22.3.23.

[3] A. Christopoulos, S. Mystakidis, E. Cachafeiro, and M. J. Laakso, "Escaping the cell: virtual reality escape rooms in biology education," *Behaviour and Information Technology*, vol. 42, no. 9, pp. 1434–1451, 2023, doi: 10.1080/0144929X.2022.2079560.

[4] M. A. Miljanovic, J. Bradbury, M. A. Miljanovic, and J. S. Bradbury, "A Review of Serious Games for Programming," in *Serious Games: 4th Joint International Conference*, 2018. [Online]. Available: https://www.researchgate.net/publication/327663695

[5] J. P. da Silva and I. F. Silveira, "A systematic review on open educational games for programming learning and teaching," *International Journal of Emerging Technologies in Learning*, vol. 15, no. 9, pp. 156–172, 2020, doi: 10.3991/ijet.v15i09.12437.

[6] N. Behnamnia, A. Kamsin, M. A. B. Ismail, and S. A. Hayati, "A review of using digital game-based learning for preschoolers," *Journal of Computers in Education*, vol. 10, no. 4, pp. 603–636, Dec. 2023, doi: 10.1007/s40692-022-00240-0.

[7] S. Xinogalos and S. Eleftheriadis, "Office Madness: Investigating the impact of a game using a real life job and programming scenario on player experience and perceived short-term learning," *Entertain Comput*, vol. 44, p. 100521, Jan. 2023, doi: 10.1016/j.entcom.2022.100521.

[8] K. Chrysafiadi, M. Kamitsios, and M. Virvou, "Fuzzy-based dynamic difficulty adjustment of an educational 3D-game," *Multimed Tools Appl*, vol. 82, no. 18, pp. 27525–27549, Jul. 2023, doi: 10.1007/s11042-023-14515-w.

[9] J. McBroom, I. Koprinska, and K. Yacef, "A Survey of Automated Programming Hint Generation -- The HINTS Framework," *ACM Computing Surveys (CSUR)* , vol. 54, no. 8, pp. 1–27, Aug. 2021, doi: 10.1145/3469885.

[10] R. Damaševičius, R. Maskeliūnas, and T. Blažauskas, "Serious Games and Gamification in Healthcare: A Meta-Review," Feb. 01, 2023, *MDPI*. doi: 10.3390/info14020105.

[11] P. Wilkinson, "Brief history of serious games," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, Springer Verlag, 2016, pp. 17–41. doi: 10.1007/978-3-319-46152-6_2.

[12] A. Kaffi, M. Rizqi, G. A. Laksono, A. Julian, and A. B. Gumelar, "The Development of Interactive Games for Covid-19 Prevention Using Indonesian Health Protocols," *JURNAL TEKNIK INFORMATIKA*, vol. 15, no. 2, pp. 101–109, Dec. 2022, doi: 10.15408/jti.v15i2.25649.

[13] I. Millington, *AI for Games*. CRC Press, 2019.

[14] S. Mitaim and B. Kosko, "What is the best shape for a fuzzy set in function approximation?," in *Proceedings of IEEE 5th International Fuzzy Systems*, IEEE, 1996, pp. 1237–1243. doi: 10.1109/FUZZY.1996.552354.

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

[16] J. R. Keebler, W. J. Shelstad, D. C. S. Google, B. S. Chaparro, and M. H. Phan, "Validation of the GUESS-18: A Short Version of the Game User Experience Satisfaction Scale (GUESS)," *J Usability*

Stud, vol. 16, no. 1, 2020, [Online]. Available: http://uxpajournal.org.

[17] "GitHub - ehsan-mohammadi/Another-LightBot: A clone of LightBot game made with Unity3D." Accessed: Oct. 10, 2024. [Online]. Available: https://github.com/ehsan-mohammadi/Another-LightBot

[18] G. Lampropoulos, E. Keramopoulos, K. Diamantaras, and G. Evangelidis, "Augmented Reality and Gamification in Education: A Systematic Literature Review of Research, Applications, and Empirical Studies," Jul. 01, 2022, *MDPI*. doi: 10.3390/app12136809.

[19] S. S. Farooq, H. Rahman, S. A. N. Raza, M. Raees, and S. K. Jung, "Designing Gamified Application: An Effective Integration of Augmented Reality to Support Learning," *IEEE Access*, vol. 10, pp. 121385–121394, 2022, doi: 10.1109/ACCESS.2022.3221473.