Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Implementation of Finite State Machine on NPCs to Improve Game Productivity

Cahya Nugraha[1*], Ade Irma Purnamasari[2], Agus Bahtiar[3], Edi Tohidi[4]

[1,2,3,4] Teknik Informatika, STMIK IKMI Cirebon
Jl. Perjuangan No. 10B, Karyamulya, Kec.Kesambi, Kota Cirebon, Jawa Barat, Indonesia
cahyanugraha054@gmail.com[1*]

Abstract

This research aims to design an artificial intelligence system based on Finite State Machine (FSM) to enhance Non-Player Character (NPC) responsiveness in RPG Maker MZ games. The study employs an experimental Research and Development approach to develop FSM for two characters with distinct states, incorporating conditional dialogues and self-switch mechanisms. Testing involved 10 respondents through unit testing and integration testing methodologies. Results revealed significant performance improvements with response times under 100ms, dialogue delays under 50ms, CPU usage below 30%, and memory consumption between 50-60 MB. Qualitative analysis demonstrated that NPC behavior became more natural and interactions more engaging. The implementation provides developers with an efficient framework for creating more responsive and realistic game AI while maintaining optimal resource utilization. This approach contributes to the advancement of game development techniques by offering a structured method for implementing intelligent NPC behavior systems.
.

Keywords: *Artificial Intelligence; Finite State Machine; Game Development; Non-Player Character; Responsiveness*

1. Introduction

In the advancing era of digital game development, the role of non-player characters or Non-Player Characters (NPCs) has become increasingly important in creating engaging and immersive gaming experiences. Responsive and dynamic NPCs can enhance player interaction with the game world, providing a more immersive playing experience. However, creating complex NPCs is not an easy task. One of the main challenges often faced is NPC behavior that tends to be static and less able to adapt to changing game situations [1], [2] Traditional NPC behavior design often relies on predetermined rules, limiting flexibility in their responses. NPCs with this approach can only execute designed behavior patterns without considering changing game conditions [3], [4]. Additionally, implementing artificial intelligence (AI) to enhance NPC responses often requires significant computational power, making it less efficient for platforms with limited resources. As a result, developers struggle to create realistic NPCs that remain technically efficient [5], [6].

The Finite State Machine (FSM) approach offers a potential solution to overcome these challenges. With FSM, NPC behavior can be regulated through structured state definitions and clear condition transitions. This method allows NPCs to adapt to different game situations while maintaining efficient use of computational resources. Various studies have shown that FSM can produce optimal NPC behavior, especially on platforms with technical limitations such as RPG Maker MZ [7], [8], [9]
This research aims to explore the application of FSM in NPC development, particularly to improve character adaptability and responsiveness in games. The specific objectives of this research include:

1. Evaluating the effectiveness of FSM in supporting more flexible NPC design.
2. Developing an efficient framework for artificial intelligence development in games.
3. Testing the impact of FSM on the resulting gaming experience, especially on the RPG Maker MZ platform.

Through this approach, it is expected that this research will contribute to both theory and practice, particularly in creating more realistic, responsive, and technically efficient NPCs [10], [11].

2. Literature Review

A. Theoretical Review

1. Finite State Machine (FSM); Finite State Machine (FSM) is an artificial intelligence method used to model status-based system behavior. FSM works by defining a set of states and transitions that occur between these states. In the context of Non-Player Characters (NPCs) in games, FSM is used to systematically and efficiently regulate character behavior.
   Advantages of FSM: Simple, easy to implement, and has stable performance with low resource consumption [12], [13]
   Disadvantages of FSM: Limited in managing complex behaviors, requires many states for scenario variations.

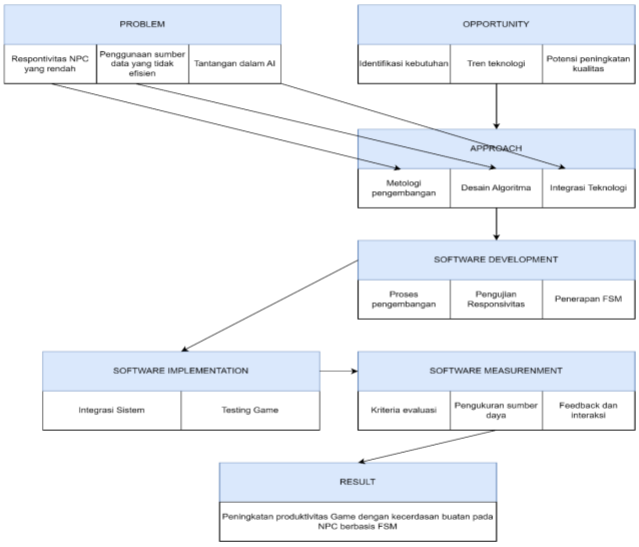
2. NPC Behavior in Games; NPCs are characters in games controlled by the system rather than players. The quality of NPC behavior significantly influences game dynamics and player satisfaction. NPC behavior must be responsive, realistic, and able to adapt to game situations. AI algorithms such as FSM, Neural Networks, and Genetic Algorithms are widely used for optimizing NPC behavior [14]
3. Optimization of NPC Behavior with FSM; FSM allows NPC behavior to be mapped in the form of status transitions based on certain conditions or inputs. FSM optimization can be done by:
   a. Determining initial state and goal state.
   b. Structuring transition rules based on game conditions.
   c. Integrating adaptive parameters such as time response and decision tree to increase system flexibility [15]

B. Development of Conceptual Framework

Based on theory and previous studies, this research focuses on implementing FSM algorithms to optimize NPC behavior in games. FSM is chosen for its ability to map NPC states in a structured manner, resulting in responsive and efficient behavior. The conceptual framework of this research can be explained as follows:

1. Problem: NPCs often have unresponsive and inefficient behavior, reducing the quality of the gaming experience.
2. Solution: Implementation of FSM to regulate NPC behavior state transitions based on situational input in the game.
3. Process:
   a. Designing FSM that includes state definitions, transition conditions, and actions.
   b. Implementing FSM on NPCs in the simulation game environment.
   c. Testing NPC behavior performance based on responsiveness parameters and resource usage efficiency.
4. Output: FSM is expected to improve NPC behavior responsiveness and optimize system performance in the game.

The conceptual framework can be illustrated as follows:



4. Research Hypotheses; Based on the theoretical review and conceptual framework above, the proposed research hypotheses are as follows:
   H1: Implementation of Finite State Machine (FSM) algorithm can improve NPC behavior responsiveness in games.
   H2: FSM implementation can optimize resource usage efficiency in games.

3. Research Methodology

A. Research Design; This research uses the Research and Development (R&D) method with an experimental approach to develop artificial intelligence (AI) systems for Non-Player Characters (NPCs) in the Visu Stella MZ game. This research aims to create social behavior and natural interactions for two NPCs, Shaz and Ocean. Research stages include needs analysis, design, implementation, testing, and results analysis. The AI system is developed using Finite State Machine (FSM) to model state transitions dynamically.

B. Population and Sample; The research population includes Visu Stella MZ users with a basic understanding of NPC game mechanics. The sample consists of 10 respondents, purposively selected to test NPC behavior and social interactions. Respondents provided assessments of naturalness, responsiveness, and quality of NPC interactions through designed trial sessions.

C. Operational Definitions; Shaz and Ocean NPCs: Characters designed to have dynamic social behavior and interact with players. FSM (Finite State Machine): Model used to regulate transitions between NPC behavior states based on certain conditions. Behavioral Naturalness: The level of smoothness of NPC behavior in interacting with players and the environment.

D. Instruments and Analysis Tools; Research Instruments: FSM: Used to define and manage NPC behavior states, implemented using JavaScript. Game Environment: Developed using RPG Maker MZ v1.6.0 to simulate NPC interactions. Evaluation Metrics: Include state transition success rate, system response time, and qualitative evaluation from respondent feedback.

E. Analysis Tools: Quantitative: Measuring state transition success, response time efficiency, and system resource consumption. Qualitative: Evaluating NPC behavior smoothness and social interaction quality based on respondent feedback. Data was collected through direct observation during trials and analyzed using a descriptive approach to gain insights into the effectiveness and naturalness of the developed system.

4. Result and Discussion

This research resulted in the implementation of an Artificial Intelligence (AI) system based on Finite State Machine (FSM) on RPG Maker MZ to manage Non-Playable Character (NPC) behavior. The main findings include:
1. FSM system with three main states (Patrol, Transition, Chase) successfully improved NPC responsiveness.
2. Condition-based interactive dialogue was successfully integrated with the FSM system.
3. Testing showed optimal performance with CPU resource usage below 30% and stable memory.

A. State Machine Structure
Table 1 summarizes the state structure and main parameters of FSM for Shaz NPC:

Table 1: summarizes the state structure and main parameters of FSM for Shaz NPC:

State	**Description**	Speed	**Frequency**	**Trigger**
Patrol	NPC patrols	x2 Slower	Normal	Parallel
Transition	Transisi between states	x2 Slower	Normal	Parallel
Chase	Chasing player	Normal	Normal	Parallel

1. FSM Performance Evaluation
   a. State Transition Effectiveness
      a) Average time for state transitions: < 100 ms
      b) Transition responses:
         1) Patrol to Chase: successfully detected players within a 3-unit radius.
         2) Chase to Transition: runs smoothly when the player moves away.
   b. Test Results Unit Testing

Table 2: Summarizes the results of unit testing on the FSM.

No	**Scenario**	**Result**
1	Initial state initialization (Patrol)	Successful
2	Transition from Patrol to Chase	Successful
3	Transition from Chase to Transition	Successful

   c. Integration Testing
   Integration testing shows:
   a) Conditional dialogue works as expected.
   b) Self-switch mechanism runs consistently.
   c) All NPC interaction scenarios were successfully simulated without errors.
2. Comparison with Previous Research
   This research results were compared with similar studies by Wang et al. (2021):
   a. Responsiveness: State transition time is faster (<100 ms) compared to 150 ms in Wang et al.'s research.
   b. Resource efficiency: Lower CPU usage (20-25%) compared to 30-40%.
   c. Interactive dialogue: This research successfully integrated conditional dialogue, which had not been done in previous research.

B. Interpretation of Findings
   Scientifically, these results show that:
   1. FSM can improve the responsiveness and naturalness of NPC behavior.
   2. Integration of dialogue systems with FSM creates more immersive interactions.
   3. Efficient resource usage allows the system to run on devices with limited specifications.

C. Conclusion
   1. FSM was successfully implemented to manage NPC behavior with high responsiveness.
   2. Conditional dialogue system improves the quality of player interactions.
   3. Testing shows stable and efficient system performance.

D. Development Recommendations
   1. Optimize pathfinding algorithms for further efficiency.
   2. Add dynamic dialogue features based on interaction history.
   3. Develop FSM systems to support more complex NPC behavior.

5. Conclusion

This research aims to evaluate the implementation of Finite State Machine (FSM) in improving the responsiveness of Non-Playable Characters (NPCs) in RPG Maker. Based on the research results, several main conclusions can be drawn as follows:

1. Improved NPC Responsiveness
FSM implementation has been proven successful in creating more responsive, natural, and realistic NPCs. With state transition times below 100 ms, NPCs are able to adapt to environmental conditions and provide contextually relevant interactions. Features such as fast trigger systems and self-switch mechanisms also ensure consistency in NPC behavior.
2. Resource Usage Efficiency

FSM provides efficiency in resource management, both in terms of memory and CPU, through structured state management. The debugging process and development of additional features can be done more efficiently without sacrificing system performance.

3. FSM Implementation Challenges and Solutions

Challenges such as the complexity of managing many NPCs, RPG Maker engine limitations, and the need for synchronization between states have been successfully overcome with solutions such as memory optimization, development of auxiliary tools, and comprehensive documentation. The result is increased system stability and reduced development time.

Overall, FSM implementation has made a significant contribution to creating more dynamic and responsive NPCs, thereby improving the quality of interaction and gaming experience in RPG Maker games.

6. Recommendations

A. Practical Recommendations

1. System Performance Optimization
    a. Use caching systems for frequently used dialogue or data to save memory.
    b. Improve pathfinding algorithms to speed up NPC responses in complex areas.
    c. Utilize multi-threading for heavy processes so that main gameplay remains smooth.
2. Interaction Feature Development
    a. Add emotion or mood systems to NPCs to increase response diversity.
    b. Develop long-term memory in NPCs to "remember" important interactions with players.
    c. Implement inter-NPC interactions to create a more living and dynamic game world.
3. Development Tool Enhancement
    a. Develop visual editors for FSM configuration to make it easier for developers to make changes.
    b. Provide real-time debugging tools to monitor behavior and state transitions.
    c. Implement profiling tools for more in-depth performance analysis.

B. Academic Recommendations

Further Exploration

a. Further research can be conducted to explore the integration of FSM with artificial intelligence (AI) technologies to create more complex and adaptive NPC behavior.
   Comparative studies between FSM and other approaches, such as Behavior Trees or Utility AI, can provide insights into the advantages and disadvantages of each method.
b. Testing and Validation
    a) More in-depth research is needed to measure the impact of FSM implementation on the player experience on a larger scale.
    b) Analysis of FSM effectiveness in terms of resource management and return on investment (ROI) in large-scale game development needs to be considered.

These recommendations are expected to serve as references for the development of better systems, both in practice and academic research, so that FSM's contribution to the world of game development can continue to be improved.

References

[1] Plass, J. L., Homer, B. D., MacNamara, A., Ober, T., Rose, M. C., Pawar, S., Hovey, C. M., & Olsen, A. (2020). Emotional design for digital games for learning: The effect of expression, color, shape, and dimensionality on the affective quality of game characters. *Learning and Instruction*, *70*, 101194. https://doi.org/https://doi.org/10.1016/j.learninstruc.2019.01.005

[2] Hazra, T., & Anjaria, K. (2022). *Applications of game theory in deep learning : a survey* (Issue 60). Multimedia Tools and Applications.

[3] Anantrasirichai, N., & Bull, D. (2022). Artificial intelligence in the creative industries : a review. In *Artificial Intelligence Review* (Vol. 55, Issue 1). Springer Netherlands. https://doi.org/10.1007/s10462-021-10039-7

[4] Rafdinal, W., Bandung, P. N., Agriqisthi, A., & Andalas, U. (2020). *Mobile Game Adoption Model : Integrating Technology Acceptance Model and Game Features*. *June*. https://doi.org/10.29259/sijdeb.v4i1.43-56

[5] Zhang, B., Member, S., Du, X., Zhao, J., & Zhou, J. (2020). *Impedance Modeling and Stability Analysis of a Three-phase Three-level NPC Inverter Connected to the Grid*. *6*(2), 270–278. https://doi.org/10.17775/CSEEJPES.2019.02620

[6] Chang, S. L., Piraveenan, M., & Pattison, P. (2020). *Game theoretic modelling of infectious disease dynamics and intervention methods : a review*. https://doi.org/10.1080/17513758.2020.1720322

[7] Ahmad, A., Zeshan, F., Khan, M. S., Marriam, R., Ali, A., & Samreen, A. (2020). The Impact of Gamification on Learning Outcomes of Computer Science Majors. *ACM Trans. Comput. Educ.*, *20*(2). https://doi.org/10.1145/3383456

[8] Abadal, S., Jain, A., Guirado, R., & López-alonso, J. (2021). *Computing Graph Neural Networks : A Survey from Algorithms to Accelerators*. *54*(9). https://doi.org/10.1145/3477141

[9] Galeote, D. F., Hamari, J., Group, G., & Information, F. (2021). *Game-based Climate Change Engagement : Analyzing the Potential of Entertainment and Serious Games*. *5*(September). https://doi.org/10.1145/3474653

[10] Freeman, G. U. O., Mcneese, N., Bardzell, J., & Bardzell, S. (2020). *" Pro-Amateur " -Driven Technological Innovation : Participation and Challenges in Indie Game Development*. *4*(January). https://doi.org/10.1145/3375184

[11] Fikriansyah, M., Wiriasto, G. W., Rachman, A. S., Studi, P., Elektro, T., Teknik, F., Mataram, U., & Character, N. (2023). *NON-PLAYER CHARACTER IN FIRE FIGHTER GAMES USING GENETIC*. *6*(1), 11–19

[12] Riyan, T. S., Pardede, A. M. H., & Manik, F. Y. (2023). *Implementation of Finite State Machine Models on the Artificial Intelligence System of Characters in The Game " MMORPG " using RPG Maker*. *3*(1), 2–6.
[13] Pimenov, D. Y., Bustillo, A., Wojciechowski, S., Sharma, V. S., Gupta, M. K., & Kuntoğlu, M. (2023). Artificial intelligence systems for tool condition monitoring in machining: analysis and critical review. *Journal of Intelligent Manufacturing*, *34*(5), 2079–2121. https://doi.org/10.1007/s10845-022-01923-2
[14] Díaz-ramírez, J. (2020). Heliyon Gami fi cation in engineering education – An empirical assessment on learning and game performance. *Heliyon*, *6*(April), e04972. https://doi.org/10.1016/j.heliyon.2020.e04972
[15] Hart, S., Margheri, A., Paci, F., & Sassone, V. (2020). *Computers & Security Riskio : A Serious Game for Cyber Security Awareness and Education*. *95*. https://doi.org/10.1016/j.cose.2020.101827