TSABIT
Journal of Computer Science
https://journal.ilmubersama.com/index.php/tsabit

Click here and write your Article Category

Development of an Android-Based Plant Care Simulation Game Using the Finite State Machine Method and Time-Based Growth Algorithm

M Rayyan Al fariz Prasetya [1], Al-Khowarizmi [2]

[1,2] Department of Information Technology, Faculty of Computer Science and Information Technology, Universitas Muhammadiyah Sumatera Utara, Medan, 20238, North Sumatra, Indonesia

ARTICLE INFORMATION

Received: February 00, 00
Revised: March 00, 00
Available Online: April 00, 00

KEYWORDS

Educational Game; Simulation; Finite State Machine; Time-Based Growth; Android.

CORRESPONDENCE

Phone: +6282164906878

E-mail: mrayyanalfarizprasetya@gmail.com

A B S T R A C T

The rapid advancement of digital technology has created new opportunities for the development of interactive educational media, particularly for learning topics that are difficult to understand through conventional methods, such as plant growth cycles and plant care processes. Traditional learning approaches often rely on theoretical explanations without providing interactive visualization, which may reduce student engagement and comprehension. Therefore, this study aims to develop an Android-based plant care simulation game utilizing the Finite State Machine (FSM) method and the Time-Based Growth algorithm as an interactive medium for both education and entertainment. The FSM method is implemented to manage structured transitions between plant growth states, including seed, sprout, mature plant, and death phases, based on predefined conditions and user interactions, while the Time-Based Growth algorithm regulates plant development according to time progression and in-game activities such as watering and fertilizing to create a more realistic simulation experience. The system also incorporates supporting mechanics, including plant health points (HP), water level monitoring, fertilizer-based growth acceleration, and a delay-death mechanism that allows plants to survive temporarily under unfavorable conditions. The application development process follows the Multimedia Development Life Cycle (MDLC) methodology, consisting of concept, design, material collecting, assembly, testing, and distribution stages. The implementation results demonstrate that the integration of FSM and Time-Based Growth successfully enables dynamic and adaptive plant growth simulations within the game environment. Furthermore, functional testing indicates that all features operate according to the designed specifications and provide an interactive and engaging learning experience that improves user understanding of plant care and growth cycles.

INTRODUCTION

The development of digital technology, particularly in the field of gaming, has created new opportunities to deliver educational content in a more interactive and engaging manner. One topic that is often difficult for students to understand conceptually is the plant growth cycle in basic biology. This material is typically presented through text or limited demonstrations, which restricts learners' ability to fully grasp the dynamic nature of plant growth and the effects of external actions on it. Research by Purba et al. [14] indicates that simulation-based games can integrate biological process visualization with real-time interaction, enabling users to learn through direct experience and observation without requiring laboratory facilities.

The main issue in learning plant growth lies in the lack of interactive media that supports exploration, repetition, and gradual observation of changes. Traditional teaching methods tend to be less engaging and do not provide immediate feedback. Therefore, a solution is needed in the form of an educational simulation game that can dynamically represent plant growth processes. In this context, the Finite State Machine (FSM) model is used to structure growth stages and state transitions systematically, while a time-based growth algorithm is implemented to simulate growth duration and the impact of user interactions on plant development [9], [10].

Traditional teaching methods regarding plant growth often appear less engaging. Several studies indicate that digitalization in learning can increase enthusiasm and conceptual understanding of the material presented, as long as the game offers mechanisms that support exploration and provide direct feedback. In order for the game to be used as an Based on these problems, this study aims to develop an Android-based plant care simulation game using Unity, implement the FSM method to manage plant growth states and transitions, and apply a time-based growth algorithm to simulate realistic growth processes. The scope of this study is limited to the Android platform, specific plant types, and a state-based system rather than a stage-based difficulty system.

The expected contributions of this research include providing an academic reference for future studies on simulation-based educational games, particularly those utilizing FSM and time-based growth models, as well as producing an interactive learning medium that can enhance users' understanding and motivation in learning plant growth concepts.

METHOD

This study uses a combination of the Research and Development (R&D) method and the Multimedia Development Life Cycle (MDLC). The Research and Development method is a research method that aims to produce a product and test the level of effectiveness of the product developed. In this study, the R&D method is used as a framework in the process of developing an Android-based plant care simulation game.

The Multimedia Development Life Cycle (MDLC) method is used as a system development method because it is suitable for multimedia application development such as games. MDLC consists of six main stages, namely concept, design, material collecting, assembly, testing, and distribution.

Finite State Machine (FSM)

Finite State Machine (FSM) is a computational model that describes a system through several different states and transitions between states that are triggered by certain inputs or conditions. In this study, FSM is applied to model the stages of plant growth and the behavior of pests in the game.

Each plant growth phase is represented as a state, such as seed, sprout, young plant, flowering, fruiting, and harvest. Transitions between states occur when certain conditions are met, such as the completion of time duration and the condition of plant health and water levels. If the conditions are not met, the plant can enter a delay phase before transitioning to the dead state.

FSM is also applied to pest behavior, where pests have states such as roaming, attacking, and escaping. Transitions between these states occur based on player interaction and game events. This implementation allows the system to manage object behavior in a structured and organized manner.

Time-Based Growth Algorithm

The Time-Based Growth algorithm is used to regulate plant growth based on time. Each growth phase has a predetermined duration that must be completed before transitioning to the next phase. The system calculates the elapsed time since the plant enters a state and resets the timer when a new state is reached.

Player interactions such as watering and fertilizing can influence the duration of plant growth. Fertilizer can accelerate the growth process by reducing the time required for each phase, while lack of water and health can delay or stop the growth process. If the plant's health and water reach critical levels, the system will trigger a delay phase before the plant transitions to the dead state.

This algorithm allows the simulation to represent plant growth dynamically and provides feedback based on player actions.

System Development Process

At the concept stage, the objectives of the application, target users, and main features are determined. The application developed is an Android-based plant care simulation game that allows users to perform activities such as watering plants, applying fertilizer, and handling pests. This stage also defines the basic gameplay and interaction mechanisms within the system.

The design stage includes system architecture design, user interface design, and game mechanics design. The material collecting stage involves gathering multimedia assets such as images, animations, and audio used in the game.

The assembly stage is the implementation phase where all components are integrated using Unity as the game engine. At this stage, FSM controls state transitions, while the Time-Based Growth algorithm manages the timing of plant growth.

The testing stage is conducted to ensure that all system functions operate according to the design using BlackBox Testing. This includes testing user interactions, FSM state transitions, and the Time-Based Growth mechanism. The final stage is distribution, where the application is packaged into an Android APK format and deployed for user use.

RESULTS AND DISCUSSION

The implementation of the system is carried out based on the design that has been developed using the Unity game engine. The results of this study include the realization of an Android-based plant care simulation game that integrates the Finite State Machine (FSM) method and the Time-Based Growth algorithm.

System Implementation

The system consists of several main interfaces, including the main menu, tutorial, gameplay, shop, and inventory. The gameplay interface displays the plant as the main object, along with supporting elements such as the timer, water status, and health bar.

Players can interact with the system through several actions, such as watering plants, applying fertilizer, and handling pests. These interactions directly affect the plant growth process and system behavior.

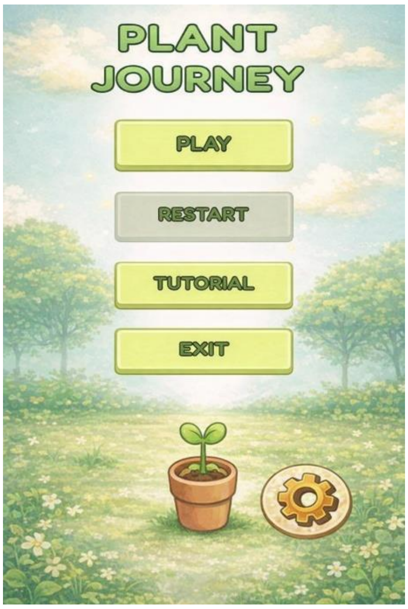

Figure 1. Main Menu Scene View (New Game)

 ISSN 3062-8504 (Online)

The main menu interface is the initial screen displayed to users when the application is launched. This interface functions as the central navigation hub for accessing all main features of the game.

In the center of the screen, several main buttons are arranged vertically, namely the **Play**, **Restart**, **Tutorial**, and **Exit** buttons. In addition, there is also a **Settings** button, typically represented by a gear icon located at the bottom of the screen.

The **Play** button only appears when the game is played for the first time or after the game has been reset. This button is used to start the game from the beginning. Meanwhile, the **Continue** button is used to resume the previously saved game progress.

The **Restart** button functions to reset the saved game data and restart the game from the beginning. This feature allows users to replay the game without being affected by previous progress.

The **Tutorial** button is used to open the tutorial scene, which contains information about how to play the game and its mechanics. This helps users understand the system before starting the game.

The **Exit** button is used to close the game application.
Additionally, the **Settings** button allows users to access the settings panel, such as adjusting the background music volume and sound effects in the game.

Overall, the main menu interface is designed to be simple and user-friendly, enabling users to quickly access the desired features.

Figure 2. Scene View Tutorial

The tutorial scene is displayed when the player presses the **"Tutorial"** button on the main menu. After transitioning to this scene, the player is presented with various information related to the game.

This includes explanations of button functions, icons, plant states, and other gameplay elements. The tutorial scene is designed to guide players in understanding how to play *Plant Journey* effectively.

The presence of this tutorial is important because many players often experience difficulties when playing a new game without clear instructions. This can lead to misunderstandings and ultimately result in failure during gameplay.

Therefore, the tutorial scene serves as a supporting feature that helps players learn the game mechanics and reduces confusion, allowing them to play more confidently and effectively.

 https://doi.org/10.12345/xxxxx

Figure 3. Initial Gameplay View

The initial gameplay interface can be seen in Figure 3. This interface represents the starting point of the game when the player begins for the first time. There are several buttons that can be used by the player to progress through the game. Detailed explanations of each button can be found in the tutorial scene.

Figure 4. Settings Panel Display

The settings panel can be opened by pressing the settings button with a gear icon. This panel functions to adjust the volume of the background music as well as the sound effects in the game.

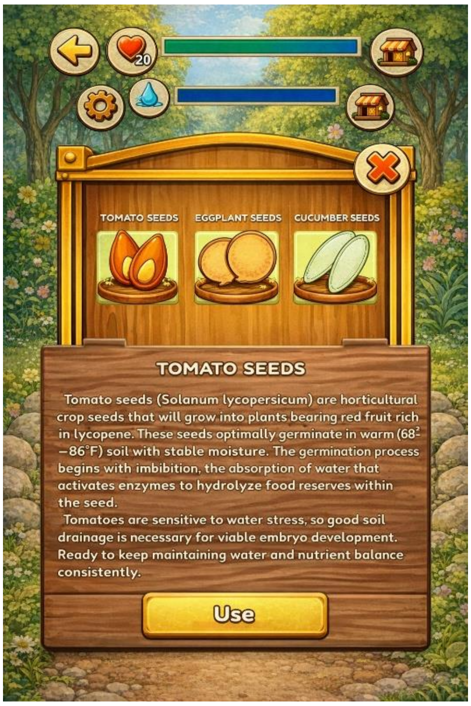

Figure 5. Storage Panel View (Warehouse/Inventory)

The storage panel can be accessed through the **“Storage”** button located at the top-right corner of the screen. In this panel, players can view the items they have obtained, such as plant seeds as shown in Figure 4.6. Players can tap an item to open an information panel about that item, and then press the **“Use”** button to use it. Items will only be available after they have been purchased from the shop.

Figure 6. Shop Menu Display

The shop menu can be accessed by pressing the **“Shop”** button located below the **“Storage”** button. In this menu, the player’s current coin balance is displayed, with an initial amount of 100 coins. Coins can be used to purchase items in the shop. Purchased plant seed items can be used through the storage panel, while fertilizer items can be bought to reset the cooldown of the **“Fertilizer”** button.

Finite State Machine (FSM) Implementation

The FSM method is successfully implemented to control the growth stages of plants and the behavior of pests. Each plant growth phase is represented as a state, including seed, sprout, young plant, flowering, fruiting, and harvest.

The test results show that state transitions occur according to the defined conditions. If the water and health values are greater than zero and the time duration is completed, the plant transitions to the next growth phase. Conversely, if the conditions are not met, the plant enters a delay state.

The delay state provides an opportunity for the player to recover the plant condition. If no action is taken, the plant transitions to the dead state, resulting in the end of the game.

Time-Based Growth Implementation

The Time-Based Growth algorithm is implemented using a timer mechanism that controls the duration of each plant growth phase. The system calculates the elapsed time and automatically updates the plant state when the duration is reached.

The test results indicate that the growth duration can be influenced by player interaction. The use of fertilizer reduces the time required for growth, while insufficient water and health can delay the process.

System Testing

Testing is carried out using the BlackBox Testing method to evaluate system functionality. The results show that all features operate according to the design.

The FSM testing confirms that all state transitions occur correctly based on defined conditions. The Time-Based Growth testing shows that the timer functions properly and responds to player interaction.

In addition, testing of plant status parameters such as water and health demonstrates that these variables significantly affect the growth process and game outcome.

CONCLUSION

Based on the results of the research and implementation testing, it can be concluded that the application of the Finite State Machine (FSM) method and the Time-Based Growth algorithm in the Android-based plant care simulation game has been successfully implemented in accordance with the system design and research objectives. The FSM method effectively manages structured transitions between plant and pest states, while the Time-Based Growth algorithm enables dynamic plant development based on time progression and user interactions, such as watering and fertilizing. In addition, supporting features including plant HP reduction, water status monitoring, delay-death mechanisms, fertilizer-based growth acceleration, and pest disturbances function logically and contribute to creating a more realistic plant care simulation experience. The testing results demonstrate that the system is capable of representing adaptive and responsive plant growth behavior within the game environment. Furthermore, evaluation and trial testing conducted by users indicate that the developed application provides an interactive and engaging simulation experience that can improve users' understanding of plant growth cycles and plant care concepts. Therefore, this research shows that the integration of FSM and Time-Based Growth methods can serve as an effective approach in developing educational simulation games that combine entertainment elements with interactive learning experiences.

REFERENCES

[1] J. Amacker, T. Kleiven, M. Grigore, P. Albrecht, and C. Horn, "A Learned Simulation Environment to Model Plant Growth in Indoor Farming," *arXiv preprint arXiv:2212.03155*, pp. 1–8, 2022.

[2] J. Arjoranta, "How to Define Games and Why We Need to," *The Computer Games Journal*, vol. 8, no. 3–4, pp. 109–120, 2019, doi: 10.1007/s40869-019-00080-6.

[3] M. S. Assidiq, "Game Edukasi Pengenalan Pertumbuhan Hewan dan Tumbuhan pada Anak Tunagrahita Ringan di SLB YPASP Gondangrejo," *Komunikasi dan Informatika*, pp. 1–19, 2020.

[4] F. Azizah and M. Salehudin, "Media Game Edukasi di Gadget: Studi Literatur Manfaat dan Dampaknya Terhadap Perkembangan Anak Usia Dini," *Journal of Instructional and Development Researches (JIDeR)*, vol. 3, no. 6, pp. 264–271, 2023, doi: 10.53621/jider.v3i6.265.

[5] B. C. R. Bustare, Wahyudin, A. H., and N. S. Putri, "Pengembangan dan Evaluasi Game Edukasi Berbasis Android Dengan Unity 3D Untuk Meningkatkan Pemahaman Konsep Persamaan Linear Satu Variabel" vol. 11, no. 3, pp. 131–143, 2024.

[6] L. P. Dewi, S. D. Putra, and E. Subyantoro, "Gamifikasi untuk Meningkatkan Kesadaran Lingkungan pada Anak Usia Dini: Studi Kasus pada Game Simulasi Merawat Tanaman," *ROUTERS: Jurnal Sistem dan Teknologi Informasi*, vol. 1, no. 2, pp. 103–108, 2023, doi: 10.25181/rt.v1i2.3119.

[7] W. Goldstone, *Unity 3.x Game Development Essentials: Game Development with C# and JavaScript*. Birmingham, U.K.: Packt Publishing, 2011.

[8] R. E. Hidayah, Murtono, S. Utaminingsih, and H. Pratama, "The Effectiveness of Game-Based Learning in the Media Android-Based Artificial Vegetative Plant Propagation Based on Natural Science Subject," vol. 1, no. 15, pp. 61–67, 2022.

[9] Y. Hitti, I. Buzatu, M. Del Verme, M. Lefsrud, F. Golemo, and A. Durand, "GrowSpace: Learning How to Shape Plants," *arXiv preprint arXiv:2110.08307*, 2021.

[10] D. Jagdale, "Finite State Machine in Game Development," *International Journal of Advanced Research in Science, Communication and Technology*, vol. 10, no. 1, pp. 384–390, 2021, doi: 10.48175/ijarsct-2062.

[11] F. Jasmin, L. Salsabila, and W. Tjandra, "Simulation and Game-Based Learning sebagai Strategi Pembentukan Karakter dalam Pembelajaran Pendidikan Pancasila," *Jurnal Ilmiah Mimbar Demokrasi*, vol. 24, no. 2, pp. 507–513, 2025, doi: 10.21009/jimd.v24i2.54661.

[12] V. Kuznetsov, M. Moiseienko, N. Moiseienko, B. Rostalny, and A. Kiv, "Using Unity to Teach Game Development," in *Proc. AET 2020*, vol. 2, pp. 506–515, 2022, doi: 10.5220/0010933400003364.

[13] H. Nonogaki, "Seed dormancy and germination—emerging mechanisms and new hypotheses," *Frontiers in Plant Science*, vol. 5, no. May, pp. 1–14, 2014, doi: 10.3389/fpls.2014.00233.

[14] A. S. Purba, A. Hufad, B. K. Amal, A. Ahyani, and N. Sutarni, "Development of games instruction within plant growth concept," *Journal of Engineering Science and Technology*, vol. 15, pp. 1–10, 2020.

[15] A. N. Puspitasari, D. U. Pribadi, and H. Suhardjono, "Growth and Yield of Tomato (*Lycopersicum esculentum* Mill.) Due to Application of Hormonic PGR and NPK Fertilizer," vol. 12, no. 4, pp. 948–956, 2023.

[16] S. Rabin, *Collected Wisdom of Game AI Professionals*. Boca Raton, FL, USA: CRC Press, 2009.

[17] D. N. Ramadhan and R. R. Adhisa, "Pengembangan Game Edukasi Materi Sistem Tata Surya Sekolah Dasar Menggunakan Aplikasi Construct 3," *Jurnal Informatika Universitas Muhammadiyah Tangerang*, vol. 8, no. 4, 2024.

[18] D. A. Sari, Yulianto, and N. R. Fadhliana, " 'Kebunku' Educational Game Using Construct 2," pp. 84–91, 2015.

[19] E. Satria, Y. Septiana, and R. Ramadhan, "Rancang Bangun Game Edukasi Pengenalan Bagian-Bagian Tumbuhan untuk Siswa Sekolah Dasar Berbasis Android," *Jurnal Algoritma*, vol. 18, no. 2, pp. 633–641, 2022, doi: 10.33364/algoritma/v.18-2.984.

[20] Savitri, D. E. Puspita, Usnawiyah, and J. Akmal, "Cucumber (*Cucumis sativus* L.) Growth and Productivity Optimization through the Application of Various Types of Organic Fertilizers," *Serambi Journal of Agricultural Technology*, vol. 7, no. 2, pp. 264–274, 2025.

[21] R. Summerley, G. Rubino, E. Lelievre, and T. Phillips, "Creation of a meaningful gardening game: Increasing players' nature awareness through soil management and plant recognition mechanics," *Journal of Environmental Media*, vol. 1, no. 2, pp. 185–207, 2020.

[22] L. Taiz and E. Zeiger, *Plant Physiology*. Cambridge, U.K.: Cambridge University Press, 2010.

[23] A. M. Wa'dan, "Pengembangan Game Pertanian Menggunakan Finite State," *Journal Sains dan Teknologi*, 2023.

[24] U. M. Wahyuni, J. Rahmadoni, A. D. Kartika, and H. Hanim, "The Combination of R&D and MDLC Models in Web-Based Interactive Learning Media," *Sistemasi*, vol. 11, no. 3, p. 713, 2022, doi: 10.32520/stmsi.v11i3.2085.