

PLAYJAVA: An Android-based Tutorial Game for Learning Basic Java Programming

Ann Valare P. Alcantara, JV R. Galasan, Regine J. Remojo & Jovi Rose P. Camacho

Computer programming surrounds daily life, enabling activities such as ordering takeout, streaming movies, and accessing online services. Technology companies now extend beyond software to offer services like personal training, meal delivery, taxi assistance, and even influence political outcomes. At its core, computer programming involves creating code to perform specific operations on computers, applications, or software programs, providing instructions for execution (Adams & Truong, 2022).

Although programming consists of various methods and practices, academic courses typically focus on essential, practical skills that allow learners to read and write code replicating real-world objects and interactions (Cote, 2022). Computer programmers, as specialists, write and test code to ensure smooth software and application performance. As the internet and the Information Technology (IT) sector continue to expand, programming languages have become the backbone of the industry, requiring IT professionals to master multiple coding languages to keep pace with evolving software demands (Torres, 2022).

Programming languages vary in difficulty and application. Beginner-friendly options include HTML, JavaScript, C, Python, and Java, while more complex ones, such as C++, Prolog, LISP, Haskell, and Malbolge, demand extensive practice and motivation (Damera & Chiampo, 2022). Java, in particular, is a robust and versatile language suited for complex, large-scale applications. As a compiled language, it is generally faster and more efficient than Python.

The increasing demand for computer use in a globalized, technology-driven society raises questions about whether students are adequately prepared for programming careers. Programming is widely acknowledged as complex, and research shows students often struggle in the early stages of learning. In the study of Derus and Ali (2012), factors such as computer background, prior experience, and ineffective teaching methods were identified as contributors to poor performance. Students reported challenges in building accurate mental models of memory processes, and difficulties were categorized into four areas: weak problem-solving skills, ineffective use of representation techniques, poor instructional strategies, and struggles with syntax and constructs (Ahmad & Abai, 2019).

Gamification has been proposed as a solution to improve motivation and engagement. Prior research demonstrated that applying game elements in non-gaming contexts enhances learning experiences (Hong et al., 2024; Christopoulos & Mystakidis, 2023; Khoshnoodifar et al., 2023; Triantafyllou et al., 2025). A gamified website designed for programming students showed potential to increase motivation and address common challenges (Ahmad & Abai, 2019; Polito & Temperini, 2021; Hellín et al., 2023; Kaya & Ercag, 2023; Ratinho & Martins, 2023).

Meanwhile, the widespread use of smartphones and mobile gaming presents both opportunities and challenges. While mobile phones support communication, advertising, and leisure, excessive gaming can lead to addiction and declining academic performance. A study on senior high school students revealed that popular games like Mobile Legends, Clash of Clans, and Rules of Survival negatively impacted grades. To address this issue, a researcher proposed developing a mobile game to aid in learning programming subjects (Rodica, 2019).

The global demand for software developers further underscores the urgency of preparing students. In 2021, there were 24.3 million active developers worldwide, with 61% under the age of 35 (Kinsta, 2023). Yet, many students still encounter the same challenges identified earlier: weak problem-solving, ineffective strategies, and difficulty mastering programming syntax (Ahmad & Abai, 2019).

To address these gaps, this study proposes PlayJava, an android-based tutorial game for learning basic java programming. The system was developed using GDevelop, a free, open-source, no-code 2D game engine that employs visual coding and scripting (Rival & Luna, 2022). The game integrates Java lessons, covering topics such as Introduction, Conditionals, Operators, Loops, and Arrays, with interactive exercises. After completing a lesson, students will engage in a mini-game, such as card-pairing with 25 programming-related questions. The goal of PlayJava is to enhance students' coding practice, particularly for those with limited programming exposure or those who struggled during online learning. By combining tutorials with interactive gameplay, the system seeks to make programming education more engaging and accessible, enabling learners to acquire fundamental coding skills while enjoying the process. The study's primary objective is to evaluate the functionality and effectiveness of PlayJava in

helping students familiarize themselves with the basic structure of Java code.

Theoretical Framework

Play and Mobile Game-Based Learning

The use of play in education is not a recent concept, but the rise of mobile games as popular entertainment has raised questions about their instructional potential. Despite this, mobile game-based learning (MGBL) offers significant benefits for both formal and informal education. For instance, Rekhawi and Naser (2018) created a web-based intelligent tutoring system for Android application development. The system provided lectures, user interface design fundamentals, and evaluations. However, it did not adequately support learning to code, highlighting the need for more interactive, code-oriented approaches.

Programming Challenges

Programming languages such as Java rely heavily on precise syntax. Students must carefully type commands, enclose structures correctly, and maintain logical organization. These demands often distract learners from programming's primary goal: problem-solving. Students can only progress efficiently once they master syntax alongside problem-solving techniques.

Research has emphasized the importance of integrating theory and practice in programming instruction. A study on computer laboratory contexts revealed that successful learning requires balancing both, rather than focusing exclusively on either (Kumari & Gupta, 2024). Phenomenography and variation theory suggest that understanding a concept requires learners to recognize significant aspects and their

variations (Han & Ellis, 2019). For instance, computer programs differ by application area, much like circles differ by diameter size. This framework highlights the diverse ways programming knowledge can be contextualized.

Digital Game-Based Learning (DGBL)

Digital game-based learning (DGBL) blends educational content with game elements such as storyline, challenge, role-playing, goals, feedback, and social interaction. These features foster engagement and motivation. Competition, in particular, encourages learners to progress, though its effects vary. Chen et al. (2018) found that peer competition enhanced interest and reduced anxiety in science lessons, while individual learning was less effective. However, Acquah and Katz (2020) noted that excessive competition can discourage weaker learners and reduce motivation.

Chang and Hwang (2019), in their review of MGBL studies (2007–2016), identified four key findings: (1) smartphones and tablets were the primary devices; (2) disciplines included science, social studies, and languages; (3) instructional methods often combined project-based learning, contextual learning, and competitions; and (4) studies explored multiple perspectives, including cognition, affective states, and learning outcomes.

Game-Based Learning in Programming

Educational games have proven effective in enhancing student engagement and outcomes. De Freitas (2018) emphasized that high-quality design elements in games positively impact learning, particularly when combined with collaboration. Liao et al. (2019) found that teamwork in DGBL significantly improved student motivation, especially when paired

with group projects. Examples of programming-focused educational games further support these findings. Battistella et al. (2017) designed games to teach Quicksort and Heapsort, improving motivation and cooperation. Malik et al. (2017) developed PROBSOL, a web-based and mobile tool for fostering problem-solving through pseudocode, which reduced attrition rates in introductory programming courses. Similarly, Sohail et al. (2019) reported that students preferred mobile learning (M-Learning) over traditional e-learning formats.

Text-Based and Coding Games

Text-based programming games provide another approach to learning. Côté (2018) introduced Rise of the Java Emperor, a text-based game designed to teach Java fundamentals through storytelling and problem-solving. Schildgen's (2014) SQL Island used a text-based adventure format to teach SQL and won recognition at the European Conference on Games-Based Learning. Other coding games like Clash of Codes allow multiplayer coding contests in languages such as Java and Python, promoting speed and competition. Compared to these, the proposed system PlayJava focuses on individual practice with structured Java lessons and exercises. While not multiplayer, it integrates interactive tutorials and mini-games to make learning both enjoyable and educational.

Research Framework

Data

The study utilized a close-ended questionnaire in which respondents rated or evaluated the system on a numeric scale ranging from 1 (lowest) to 4 (highest). Respondents selected the option that best represented their

opinion. The questionnaire was designed based on ISO 25010 quality attributes: reliability, performance efficiency, compatibility, and usability. The survey was conducted with a total of 86 respondents participated, consisting of 1st- and 2nd-year IT students.

Experimental Design

The experimental design was developed to test the project's independent and dependent variables.

Independent Variable: The scores given to the proposed system. These scores are independent as they can vary depending on the evaluations of students and other beneficiaries.

Dependent Variable: The ranking of the proposed system. Ranking serves as the dependent variable, as it is influenced by the variations in the independent variable.

Procedures of the Different Phases

The proposed system is designed to run on Android mobile devices and can be accessed anytime by students and other users. Upon launching the application, the main menu is displayed, presenting five selected topics in Java. After selecting the Play button, the system provides game instructions, followed by the display of basic Java syntax structures. Learners then complete programming exercises in a fill-in-the-blanks format, consisting of 25 randomized questions.

Upon completion, the system generates a dashboard showing a summary of answers. Scores are displayed and automatically recorded in the game's leaderboard for tracking and comparison.

Modelling

Figure 1

System flow modeling

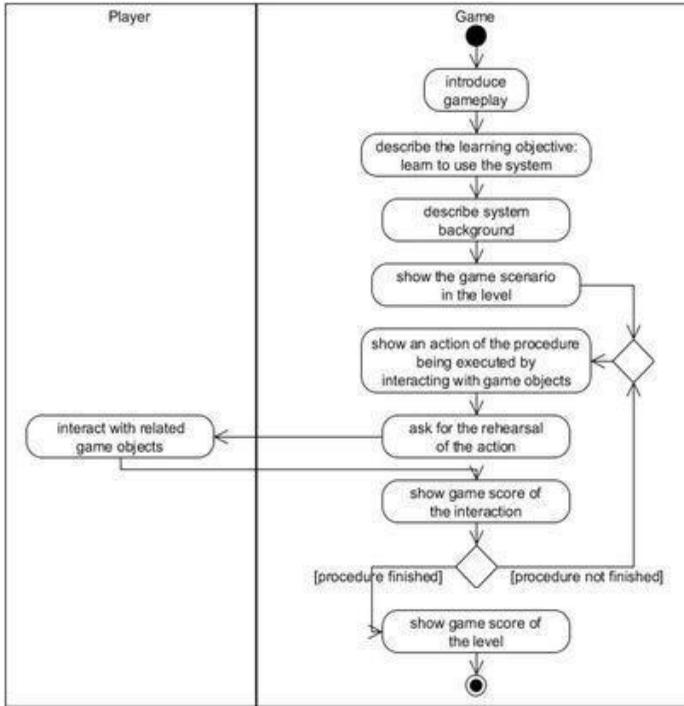
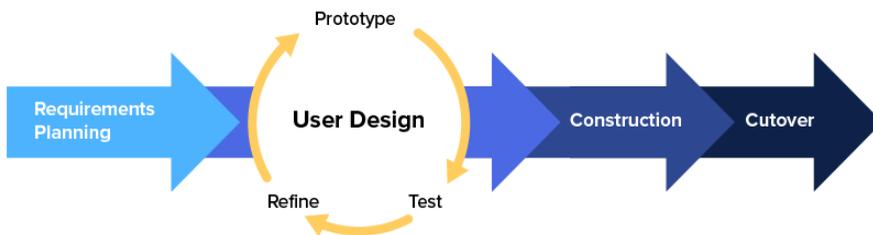


Figure 2

Rapid Application Development (RAD)



System Development Life Cycle (SDLC) Phases

Requirements planning. This phase combines the planning and analysis stages of the SDLC. Through observation and data collection, the

researchers identified problems and set objectives for potential solutions. By examining system requirements, including the necessary materials and their specifications, the researchers determined the actions needed to address issues in the existing system. Key activities included defining the problem, gathering information, setting goals, and identifying and evaluating software requirements. The outcome of this phase was a plan for the subsequent user design stage.

User design. In this phase, user-developed models representing system operations, inputs, and outputs were created. Through continuous interaction, users could understand, modify, and approve a functional model that meets their needs. The researchers focused on realizing the logical design of the proposed system, addressing operating system, programming, and security concerns. Plans also included the user interface for system development and management.

Construction. This phase involved finalizing the system design, constructing the software, and testing its performance. The goal was to develop a system that functioned at an acceptable level while preparing essential documentation for operation. Transitional software was designed, developed, and tested, along with the procedures required to prepare the system for conversion into production.

Cutover. The cutover phase consisted of the final implementation tasks. Activities included system setup, user training, and completion of all necessary documentation to ensure smooth deployment and operation.

Deployment and Maintenance

Preparation: During this phase, researchers gather all deployed code, libraries, configuration files, and resources required for the application. These components are packaged as a single software release.

Developers also ensure that the host server is correctly configured and operating without issues.

Testing: Updates are first deployed to a test server, where they undergo pre-configured automated tests. Results are evaluated, and any identified faults or errors are corrected before the update is released to the production environment.

Deployment: After thorough testing, updates are pushed to the live environment. Necessary scripts may be executed to update relevant databases. The final stage involves monitoring the live server to identify and resolve any remaining errors, ensuring an optimal user experience.

Ethical Considerations

Developing a mobile-based tutorial game for learning Java programming entails several ethical considerations:

Privacy: The system collects and stores players' names in the game leaderboard. All information is secured and used solely for scoring purposes.

Accessibility: The game is designed to be accessible to all users, regardless of prior programming experience, by introducing basic Java lessons.

Fairness: The system ensures fair and equal access to all players.

Responsibility: The developers are responsible for maintaining the security of the system and protecting all users' personal information.

Technical Framework

The study used various tools and software to design, develop, and document the proposed system.

Software

Table 1

Software specification and software platforms

System Type	X64-based PC
Database	GDevelop
Game Engine	GDevelop
Operating System	Windows 11

One of the primary tools utilized was GDevelop, a free, open-source, and no-code 2D game engine. Its accessibility and cost-free nature make it ideal for educational purposes, regardless of the number of users. GDevelop was employed in developing the proposed game, particularly in creating its functionalities, dashboard, and leaderboard features. In addition, Microsoft Word was used as a documentation tool. As a widely recognized word processing software developed by Microsoft, it enabled the researchers to prepare, edit, and organize project documentation such as reports, user manuals, and other written outputs necessary for the system's development.

Hardware

The implementation of the proposed system requires an Android phone with at least 2.00 GB of RAM or higher to ensure optimal performance. The specifications presented above illustrate the compatibility of the developed system with different Android phone models that meet or exceed these requirements.

Table 2*Hardware requirements*

Device Name:	Oppo A5s
Model:	CPH1909
Android Version:	8.1.0
Processor:	Octa – Core
RAM:	3.00 GB
Device Storage:	32.0 GB
Version:	CPH1909EX_11_A.46
Device Name:	Infinix Smart 5
Model:	X657C
Android Version:	10
Processor:	Helio H20
RAM:	2.00 GB
Device Storage:	32.0 GB
Version:	XOS v6.2.0 (Q-P89-2006i8)
Device Name:	Infinix Note 7
Model:	Infinix X690B
Android Version:	10
Processor:	Helio G70
RAM:	6.00 GB
Device Storage:	128.00 GB
Version:	XOS v6.1.0 (Q-P69-200225)
Device Name:	RedMI 9
Model:	M2004J19G
Android Version:	11
Processor:	Octa – Core Max 2.00 GHz
RAM:	4.00 GB
Device Storage:	64 GB
Version:	11 RPIA.200720.011
Device Name:	Vivo 1906
Model:	Vivo 1906
Android Version:	11
Processor:	1.95 GHz Snapdragon 439 Octa – Core
RAM:	3.00 GB
Device Storage:	32.0 GB
Version:	PD1930CF_EX_A_6.8.59

System Design

Main page. The first screen of the system serves as the introductory interface, displaying the game logo and the title Play Java. It contains two interactive buttons: Main Menu and Exit, which allow the user to either proceed to the main content of the application or close it.

Lesson screen. Upon selecting the Main Menu, the user is directed to the Lesson Screen, which presents the key topics covered in the Java tutorial. These include Java Introduction, Java Conditionals, Java Operators, Java Looping, and Java Array. This screen functions as the starting point for users to navigate and explore the lessons provided by the system.

Figure 3

Main page and lesson screen



First topic selection. This screen displays the first lesson, titled “Java Introduction,” providing users with an overview of the foundational concepts of the Java programming language.

First topic: Java introduction. This screen presents introductory information about the Java programming language, providing users with a basic understanding of its purpose, features, and applications.

Figure 4

The first topic



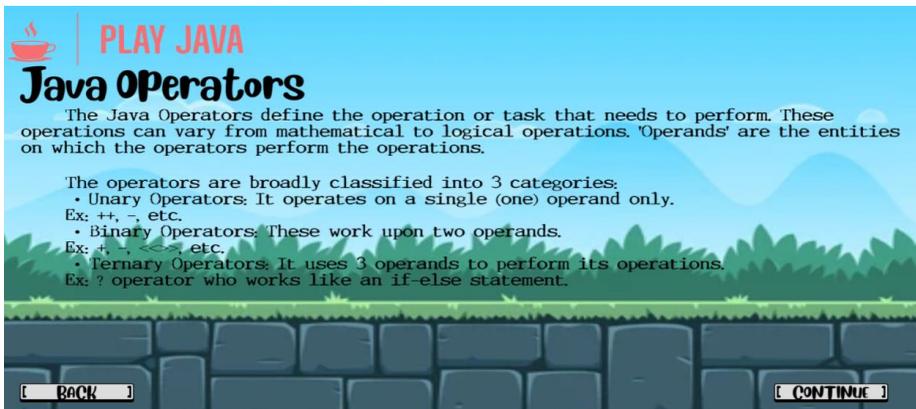
Selecting second topic (Java Conditionals). This screen presents the lesson on “Java Conditionals,” which is the second topic in the tutorial. It provides users with information on conditional statements in Java, explaining how decisions are made in programs based on certain conditions.

Selecting third topic (Java Operators). This screen provides information on Java Operators, introducing users to the symbols and expressions used to perform operations on variables and values in Java programs.

Java operators. This screen presents the lesson on Java Operators, including explanations and examples that demonstrate how operators are used to perform calculations and manipulate data within Java programs.

Figure 5

Third topic

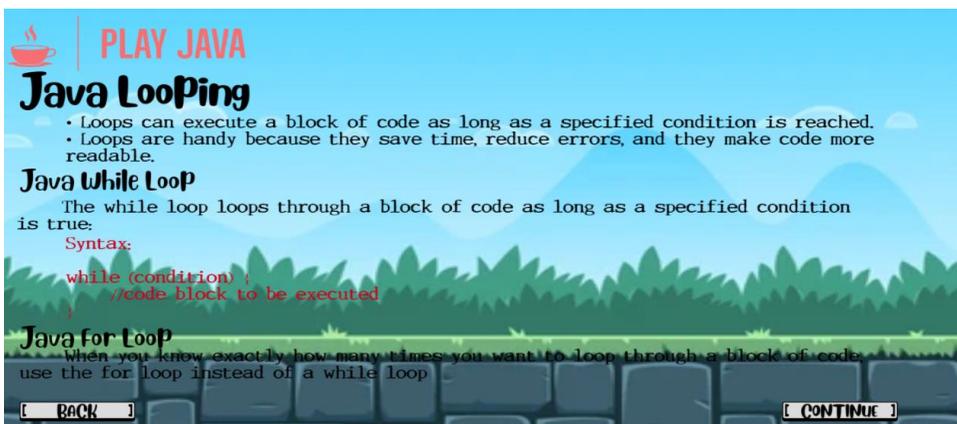


Selecting fourth topic. This screen presents the lesson on Java Looping, including explanations and examples that demonstrate how loops are used to execute repeated actions and control the flow of a Java program.

Java looping. Description: This screen displays the lesson on Java Looping, providing explanations and examples that illustrate how loops are used to perform repetitive tasks and manage program flow in Java.

Figure 6

Fourth topic

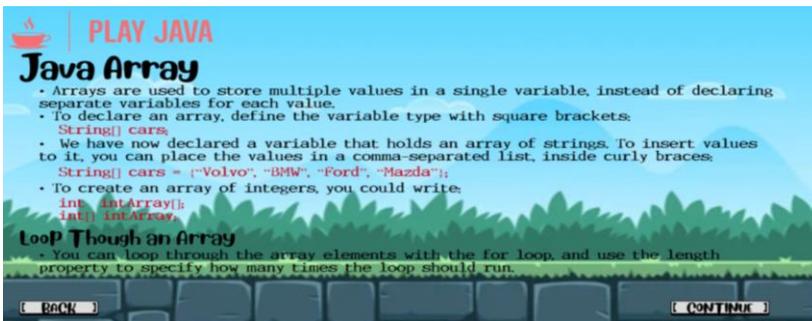


Selecting fifth topic. This screen presents the lesson on Java Arrays, including explanations and examples that demonstrate how arrays are used to store, access, and manipulate multiple values within a Java program.

Java array. This screen displays the lesson on Java Arrays, providing explanations and examples that illustrate how arrays are used to store and manipulate multiple data elements in a Java program.

Figure 7

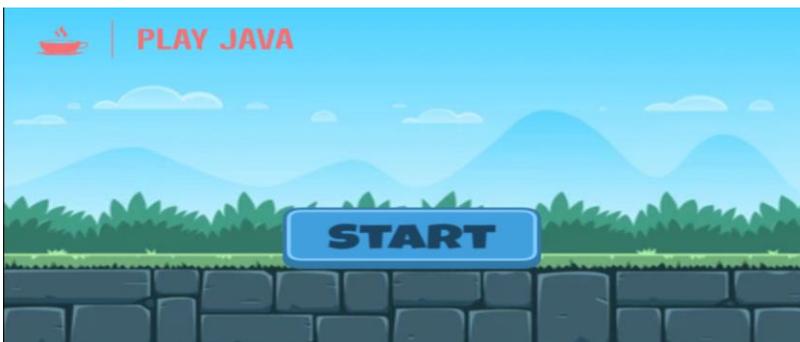
Fifth topic



Main screen of the game. This screen displays the Start button, which directs the user to the game section of the system, initiating the interactive learning activities.

Figure 8

Game screen



Pairing cards mini game. This screen presents the Java programming exercises, featuring a pairing card activity where users match answers to complete fill-in-the-blank questions, reinforcing their understanding of the lesson concepts.

Figure 9

The mini game



Wrong answer. This screen appears when a user provides an incorrect answer, offering feedback to indicate the error and guide the user toward the correct response.

Figure 10

Wrong answer screen



Correct answer. This screen appears when a user provides a correct answer, confirming the response and reinforcing the learning outcome.

Figure 11

Correct answer screen



Input player's name. This screen prompts the player to enter their name after completing all twenty-five questions, allowing the system to record and display their results on the leaderboard.

Figure 12

Screen to input player name



Dashboard. This screen displays a summary of the user's responses, showing both correct and incorrect answers to provide feedback on their performance.

Figure 13

Summary of answers screen



Figure 14

*Activity diagram for PlayJava
(Mini Game)*

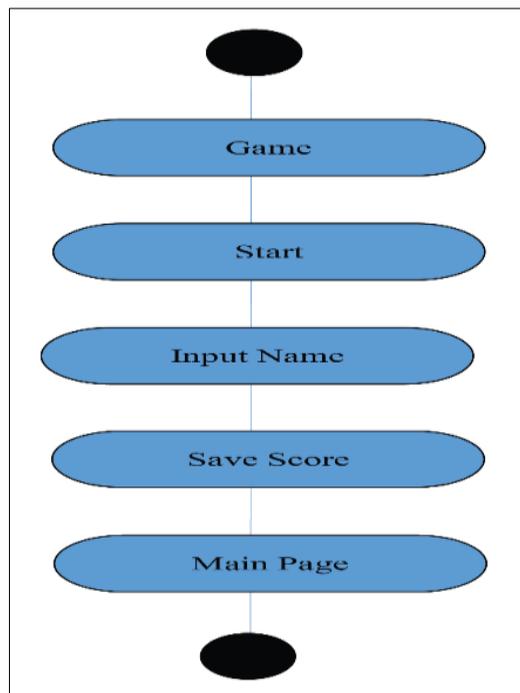


Figure 14 illustrates the process flow of the developed system. Users begin by entering the game and answering all randomized questions based on the selected topics. Upon completing the exercises, users input their name, after which the system provides a summary of correct and incorrect answers.

Figure 15

Activity diagram for PlayJava (Java Introduction)

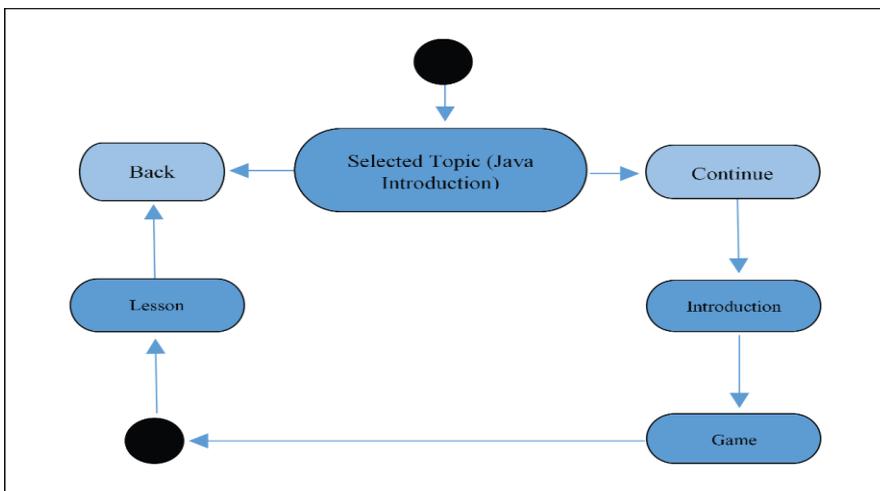


Figure 15 illustrates the process that occurs when the user clicks the “Java Introduction” button on the main page. The system immediately displays the corresponding lessons and examples, allowing the user to explore the foundational concepts of Java programming.

Figure 16 illustrates the process that occurs when the user clicks the “Java Conditional” button on the main page. The system immediately displays the corresponding lessons and examples, allowing the user to learn about conditional statements in Java.

Figure 16

Activity diagram for PlayJava (Java Conditionals)

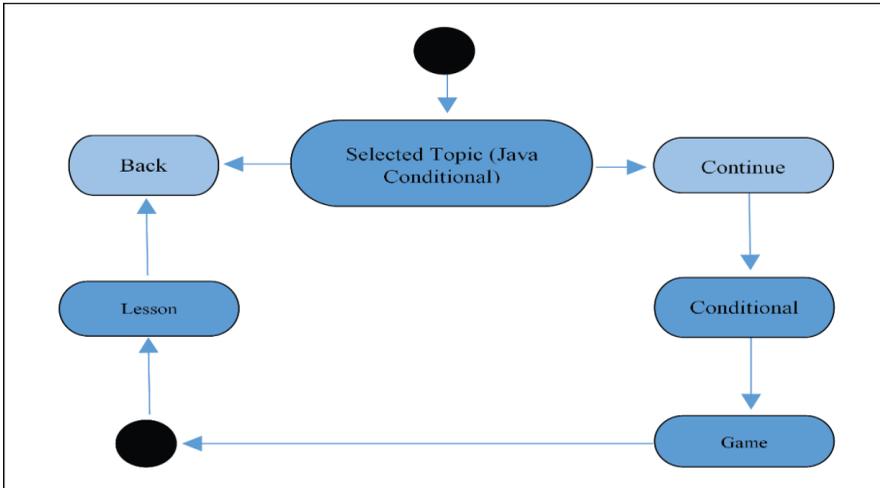


Figure 17

Activity diagram for PlayJava (Java Operators)

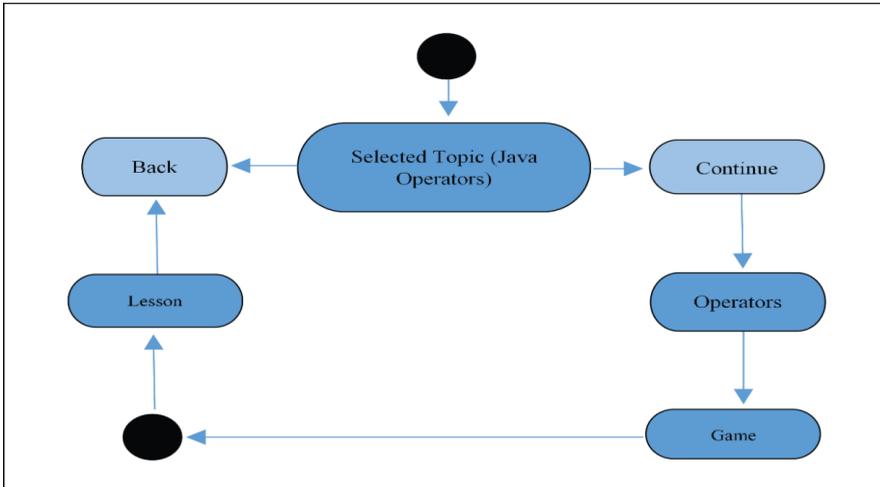


Figure 17 illustrates the process that occurs when the user clicks the “Java Operators” button on the main page. The system immediately displays the corresponding lessons and examples, enabling the user to learn how operators are used in Java programming.

Figure 18

Activity diagram for PlayJava (Java Looping)

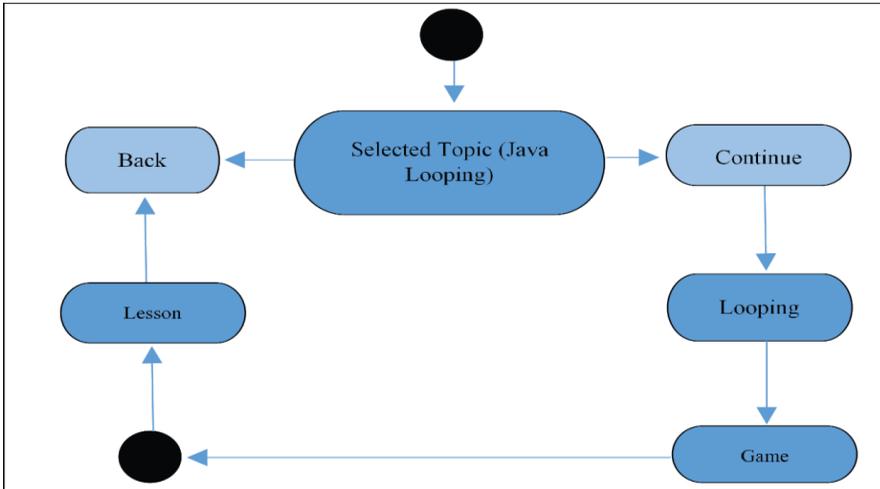


Figure 18 illustrates the process that occurs when the user clicks the “Java Looping” button on the main page. The system immediately displays the corresponding lessons and examples, allowing the user to understand and practice loop structures in Java programming.

Figure 19

Activity diagram for PlayJava (Java Array)

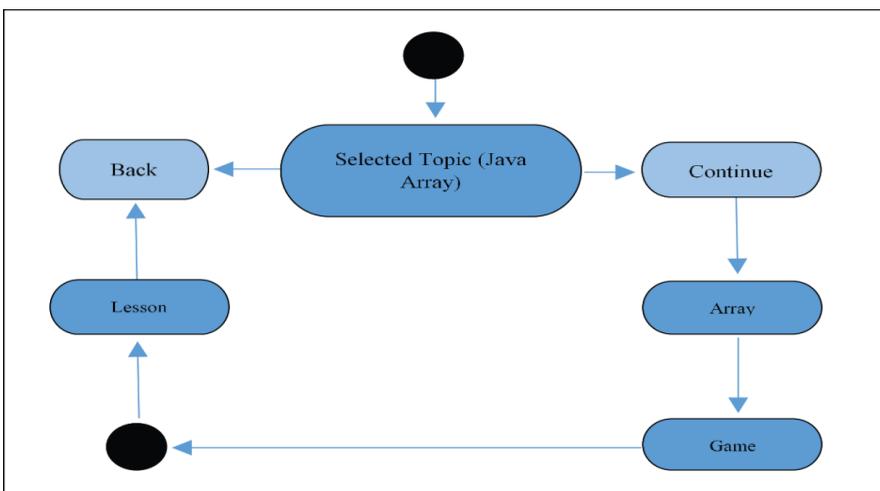


Figure 19 illustrates the process that occurs when the user clicks the “Java Array” button on the main page. The system immediately displays the corresponding lessons and examples, enabling the user to learn how arrays are used to store and manipulate data in Java programs.

Figure 20

USE case diagram of the proposed system

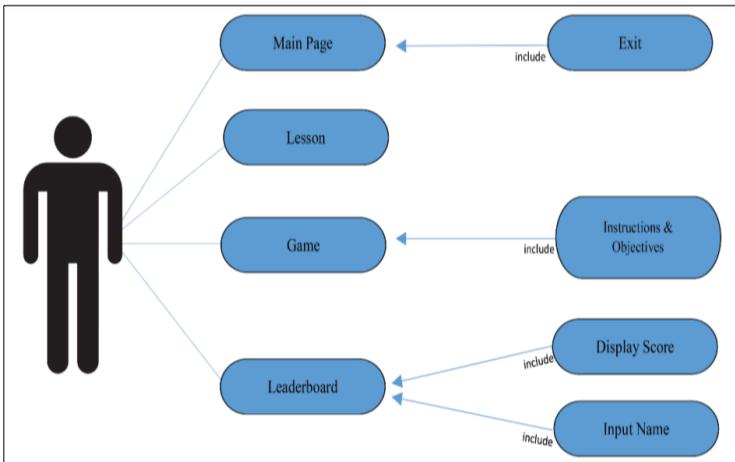


Figure 20 illustrates the process for using the system from the player’s perspective. Each stage of system access is represented, beginning with the player clicking the Start button to launch the game. Players must complete each assignment sequentially before progressing to the next level, and they cannot choose which mini-game to play.

Development and Maintenance

The development and maintenance of an Android game involve several stages. During development, game designers collaborate to create the game concept, including mechanics, graphics, sound, and other features. The development team then uses GDevelop, a no-code 2D game engine, to

implement the necessary code and functionalities. Following development, the game undergoes testing to identify and resolve any bugs or glitches. Once tested, the game is deployed on Android devices.

During the maintenance phase, developers monitor game performance, collect player feedback, and address any issues that arise. Maintenance may also include introducing new features, updating the game to remain compatible with new operating system versions, and resolving hardware or software compatibility issues. Effective development and maintenance ensure that the game remains functional, engaging, and relevant to its users.

Testing

The study adopts a proactive test approach, initiating the test design process as early as possible to detect and correct bugs before the system build is completed.

Table 3

Integration testing

Item No.	Test Case Objectives	Expected Result	Passed	Failed	Change Control #
1	Test the functionality of the game main menu	The game can be launched	/		1
2	Test the movement of the flipping cards	Flipping cards are accurate based on the command	/		2
3	Test the leaderboard	Player's name and summarization of the answer will stored in the leaderboard	/		3
4	Test the accuracy of lesson	The accuracy of the lesson is correct based on the given topics.	/		4
5	Test the sounds and design	The game sound and design is accurate	/		5
Test Completed by: Ann Valare P. Alcantara, JV R. Galasan, Regine J. Remojo & Jovi Rose P. Camacho			Date Completed: June 11, 2023		
Test Case Outcome			Accepted		

The application name corresponds to the system name, while the test case version represents the series of tests conducted. Each test identifies a specific procedure, with the objective and expected result defined according to the system process. Any issues identified during testing are recorded with a corresponding change control number. The integration test focuses primarily on the interfaces and the flow of data between modules, ensuring seamless interaction. The test form is used to document all required data systematically.

System Evaluation

This section presents the results of the survey with a total of 86 respondents. The sample size was determined using the Raosoft formula, and respondents were selected through a random sampling method. The questionnaire was designed based on ISO 25010 quality characteristics, including usability, reliability, performance efficiency, and compatibility. Data collected from the survey were analyzed using the weighted mean formula. A four-point Likert scale was employed, with the response options: strongly agree, agree, disagree, and strongly disagree. Each scale corresponds to a defined mean range, which was used to interpret and evaluate the respondents' feedback.

Table 4

Weighted mean distribution of criteria for developed system

Criteria	Weighted Mean	Remarks
Usability	3.49	Strongly Agree
Reliability	3.52	Strongly Agree
Performance Efficiency	3.32	Strongly Agree
Compatibility	3.52	Strongly Agree
Average	3.42	Strongly Agree

The overall average weighted mean of the survey is 3.42, indicating that respondents rated the system as outstanding in terms of reliability, compatibility, usability, and performance efficiency. This suggests that the system is capable of performing its intended functions effectively. Positive feedback from respondents highlighted that the system design is visually comfortable and the interface is intuitive, allowing users to understand and navigate the system easily.

Specifically, the reliability of the system received an average weighted mean of 3.52, demonstrating that the system performs its intended functions consistently without glitches or errors. For performance efficiency, the average weighted mean is 3.32, indicating that the system operates efficiently during use. The compatibility dimension also scored 3.52, showing that the system runs correctly across different devices and models. Finally, the summary of the weighted mean results confirms that the system meets the quality standards outlined in ISO 25010 for usability, reliability, performance efficiency, and compatibility.

Economic feasibility. An economic feasibility assessment was conducted through a cost-benefit analysis to determine whether the projected benefits of the Android-based application outweigh its development costs. This analysis considered the system's advantages and disadvantages and evaluated its applicability. Feedback on the system's functionality and economic feasibility was gathered through the ISO 25010 questionnaire-based survey, providing insight into its value relative to cost.

Technological feasibility. The system was found to be technologically feasible. Respondents noted that it is helpful for learning the basic syntax of Java, user-friendly, and compatible with various Android phone brands and models.

Operational feasibility. The system is operationally feasible, as it is easy to understand and operate. Survey responses indicated that users could quickly grasp its functionality after a brief demonstration, confirming that it meets operational expectations.

Conclusion

PlayJava, an Android-based tutorial game for learning basic java programming, allows Android users to engage with Java lessons while practicing code analysis in a gamified format. Users must complete each lesson before accessing the mini-game, enabling them to learn programming fundamentals while enjoying the experience. The testing and evaluation of PlayJava demonstrated that it meets ISO 25010 standards for performance efficiency, reliability, usability, and compatibility. These results confirm that the system is a viable alternative for learning basic Java programming through mobile-based, gamified instruction.

Bibliography

- Adams, R., & Truong, J. M. (2022, December 15). *What is programming*. Codecademy. <https://www.codecademy.com/article/what-is-programming>
- Ahmad, A. L., & Abai, A. (2019). What is difficult in learning programming language based on problem-solving skills. *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 18–22. <https://doi.org/10.1109/ICOASE.2019.8723740>
- Attard, L., & Busuttil, L. (2020). Teacher perspectives on introducing programming constructs through coding mobile-based games to secondary school students. *Informatics in Education*, 19(4), 543–568. <https://doi.org/10.15388/infedu.2020.19.4.03>
- Christopoulos, A., & Mystakidis, S. (2023). Gamification in education. *Encyclopedia*, 3(4), 1223–1243. <https://doi.org/10.3390/encyclopedia3040089>
- Cote, J. (2022, September 6). *What is computer programming*. Southern New Hampshire University. <https://www.snhu.edu/about-us/newsroom/stem/what-is-computer-programming>
- Damera, A., & Chiampo, A. (2022, December 15). *Coding career: Coding languages list for 2022*. Emeritus. <https://emeritus.org/blog/coding-career-coding-languages-list-for-2022/>
- Derus, S. R. M., & Mohamad Ali, A. Z. (2012, September). *Difficulties in learning programming: Views of students*. Paper presented at the 1st International Conference on Current Issues in Education (ICCIE 2012), Yogyakarta, Indonesia. <https://doi.org/10.13140/2.1.1055.7441>
- Han, F., & Ellis, R. A. (2019). Using phenomenography to tackle key challenges in science education. *Frontiers in Psychology*, 10, 1414. <https://doi.org/10.3389/fpsyg.2019.01414>
- Hellín, C. J., Calles-Esteban, F., Valledor, A., Gómez, J., Otón-Tortosa, S., & Tayebi, A. (2023). Enhancing student motivation and engagement through a gamified learning environment. *Sustainability*, 15(19), 14119. <https://doi.org/10.3390/su151914119>
- Holguin, A., & Torres, G. (2022, December 15). *Introduction to computer programming*. GCFGlobal. <https://edu.gcfglobal.org/en/computer-programming-basics/introduction-to-computer-programming/1/>
- Hong, Y., Saab, N., & Admiraal, W. (2024). Approaches and game elements used to tailor digital gamification for learning: A systematic literature review. *Computers & Education*, 212, 105000. <https://doi.org/10.1016/j.compedu.2024.105000>

- Kaya, O. S., & Ercag, E. (2023). The impact of applying challenge-based gamification program on students' learning outcomes: Academic achievement, motivation and flow. *Education and Information Technologies*, 28, 10053–10078. <https://doi.org/10.1007/s10639-023-11585-z>
- Khoshnoodifar, M., Ashouri, A., & Taheri, M. (2023). Effectiveness of gamification in enhancing learning and attitudes: A study of statistics education for health school students. *Journal of Advances in Medical Education & Professionalism*, 11(4), 230–239. <https://doi.org/10.30476/JAMP.2023.98953.1817>
- Kumari, A., & Gupta, A. (2024). Correlating Computer Laboratory Environments with Student Attitudes and Efficacy: A Study in Indian Secondary Schools. *Social Education Research*, 6(1), 20–38. <https://doi.org/10.37256/ser.6120255285>
- Liu, Y. J., Zhou, Y. G., Li, Q. L., & Dong, Y. (2022). Impact study of the learning effects and motivation of competitive modes in gamified learning. *Sustainability in Education Gamification*.
- Matthew, R., Malik, S. I., & Tawafak, R. M. (2019). Teaching problem-solving skills using an educational game in a computer programming course. *Informatics in Education*, 18(3), 359–373. <https://doi.org/10.15388/infedu.2019.18.3.06>
- Nhat Pham, L. L., Nguyen, H. T., & Le, V. T. K. (2018). Triggering students' learning autonomy using the combination of M-learning and gamification: A case study at Nguyen Tat Thanh University. *Teaching English with Technology*, 18(3), 66–91.
- Polito, G., & Temperini, M. (2021). A gamified web-based system for computer programming learning. *Computers and Education: Artificial Intelligence*, 2, 100029. <https://doi.org/10.1016/j.caeai.2021.100029>
- Ratinho, E., & Martins, C. (2023). The role of gamified learning strategies in student motivation in high school and higher education: A systematic review. *Heliyon*, 9(8), e19033. <https://doi.org/10.1016/j.heliyon.2023.e19033>
- Rekhawi, A., & Naser, H. (2018). *Web-based intelligent tutoring system for teaching Android application development* [Unpublished manuscript].
- Rival, F., & Luna, E. (2022, December 15). *GDevelop for education*. GDevelop. <https://gdevelop.io/page/education>
- Rodica, P. D. (2019). *Effects of online games in academic performance among senior high school students*. Unpublished Thesis. Department of Science, Technology, Engineering and Mathematics,

Mount Carmel School of Maria Aurora (MCSMA).

- Thume, M., & Ecderkal, A. (2018). Analysis of students' learning of computer programming in a computer laboratory context. *European Journal of Engineering Education*, 43(6), 769–786. <https://doi.org/10.1080/03043797.2018.1442679>
- Triantafyllou, S. A., Georgiadis, C., & Sapounidis, T. (2025). Gamification in education and training: A literature review. *International Review of Education*, 71, 483–517. <https://doi.org/10.1007/s11159-024-10111-8>
- Yang, D., & Kopcha, T. J. (2022). Developing board games for beginning block-based programmers. *International Journal of Designs for Learning*, 13(2). <https://doi.org/10.14434/ijdl.v13i2.34567>