Journal of International Conference Proceedings (JICP)
Vol. 8 No. 1, pp. 378-388, June, 2025
P-ISSN: 2622-0989/E-ISSN: 2621-993X
https://www.ejournal.aibpmjournals.com/index.php/JICP

Programming Learning Platform with Visual Aids

Foong Yuh Chung
School of Computing, INTI International College Penang
1-Z, Lebuh Bukit Jambul, Bukit Jambul, 11900 Bayan Lepas, Pulau Pinang
Correspondence Email: p21013579@student.newinti.edu.my
ORCID ID: https://orcid.org/0009-0000-7036-4801

ARTICLE INFORMATION

Publication information

Research article

HOW TO CITE
Chung, F. Y. (2025). Programming Learning Platform with Visual Aids. *Journal of International Conference Proceedings*, 8(1), 378-388.

DOI:
https://doi.org/10.32535/jicp.v8i1.4002

Received: 26 April 2025
Accepted: 27 May 2025
Published: 28 June 2025

ABSTRACT

This paper presents the design and development of a user-centric web-based programming learning platform aimed at novice programmers. The platform addresses common challenges in learning programming by integrating visual learning aids and interactive features. A real-time code visualization tool enables learners to trace code execution line-by-line, helping demystify abstract concepts and debug code logic. The system also incorporates progress tracking dashboards, formative assessments (quizzes and coding exercises), and a digital badge reward system to motivate continuous learning. The platform was implemented as a Single-Page Application using a cloud Backend-as-a-Service (Firebase) to support real-time data synchronization and user management. Development followed an iterative, user-centered approach, including usability testing with target users that informed subsequent refinements. Results indicate that the platform effectively engages beginners by bridging the gap between theoretical knowledge and practical application through interactive visual feedback. This work contributes an innovative e-learning solution that leverages visualization and gamification to enhance the learning experience for novice programmers.

Keywords: E-learning; Gamification; Novice programmers; Programming education; Visual code execution

Journal of International Conference Proceedings (JICP)
Vol. 8 No. 1, pp. 378-388, June, 2025
P-ISSN: 2622-0989/E-ISSN: 2621-993X
https://www.ejournal.aibpmjournals.com/index.php/JICP

INTRODUCTION

Learning programming is increasingly essential, yet novices frequently face difficulties understanding abstract concepts and syntactic details, resulting in frustration and high dropout rates in introductory courses (Hudin et al., 2024). These challenges stem from programming's inherent cognitive complexity, demanding abstraction, algorithmic thinking, and robust problem-solving skills (Hudin et al., 2024). Traditional learning approaches often lack interactivity and visual context, causing beginners confusion about program execution and logic.

Visualization techniques and gamification are promising strategies to address these challenges. Visual programming environments have been shown to enhance understanding and engagement by clarifying complex concepts through dynamic visual feedback (Lackner & Podlipnig, 2022). Additionally, motivational elements like badges and interactive assessments positively influence learner perseverance and engagement in e-learning contexts (Zainudin & Zulkiply, 2023).

Given this context, there is a clear need for specialized e-learning solutions that address the unique challenges of programming education. Prior studies suggest that incorporating visual and interactive elements can help novices overcome learning obstacles. For instance, program visualization techniques have been shown to improve novices' understanding of code execution, helping to correct misconceptions and debug programs more effectively (Lackner & Podlipnig, 2022). Similarly, providing a visual programming environment or feedback can positively impact learners' engagement and computational thinking skills (Gong et al., 2024). Furthermore, motivational features like gamification (e.g. badges or points) can increase student engagement and perseverance in e-learning contexts.

Objective: In response to the above challenges and insights, this project set out to develop a web-based learning platform with integrated visual aids and gamified features to support novice programmers. The core objective was to create an interactive system that not only delivers programming content but also actively guides beginners through code execution and problem-solving. To achieve this, the platform includes: (1) a real-time code execution visualizer that allows learners to step through code line-by-line with graphical feedback; (2) a progress tracking system that monitors students' learning milestones and performance; (3) an assessment module with quizzes and coding exercises for practice; (4) a digital badge system to reward achievements and encourage progression; and (5) a discussion forum for learners to seek help and build a community. An administrative interface is also provided for instructors or platform administrators to manage course content and monitor overall usage. By combining these features in a single cohesive application, the project aims to enhance novices' understanding of programming concepts, provide timely feedback and support, and improve engagement and motivation through gamification.

This study aims to develop a web-based programming learning platform integrating these beneficial features real-time code visualization, progress tracking, interactive quizzes, gamification through digital badges, and community forums. By combining these elements, the platform seeks to support novice programmers, improve concept comprehension, and foster ongoing learner engagement.

Journal of International Conference Proceedings (JICP)
Vol. 8 No. 1, pp. 378-388, June, 2025
P-ISSN: 2622-0989/E-ISSN: 2621-993X
https://www.ejournal.aibpmjournals.com/index.php/JICP

LITERATURE REVIEW

Novice Programming Challenges: Many researchers have investigated the difficulties faced by novice programmers. Commonly cited challenges include understanding abstract concepts (like variables, loops, recursion), learning new syntax, and applying problem-solving strategies in coding (Lahtinen, Ala-Mutka & Järvinen, 2005). Novice students often experience cognitive overload when first exposed to programming, as it requires simultaneous mastery of logic, syntax, and computational thinking (Garner, 2002). Inadequate feedback and lack of visualization in traditional instruction can exacerbate these issues. A recent action research study by (Singh, 2022) confirmed that an overwhelming majority of beginning programming students find the subject difficult and would benefit from additional support in the learning process. This aligns with earlier findings that despite various pedagogical interventions over the years, the fundamental problems for first-time programming learners persist (Kalelioglu & Allsop, 2024). Large class sizes and limited individualized attention contribute to the problem, as novices cannot always get timely help for errors or misunderstandings. Thus, there is a consensus in the literature that new approaches are needed to better support beginners, reduce frustration, and improve retention in introductory programming courses.

Visualization and Interactive Learning: One promising approach to support novice programmers is the use of program visualization and other interactive learning tools. Program visualization involves representing the execution of code in a graphical or animated form, allowing learners to see how the program state changes with each instruction. Prior research suggests that visualization can make abstract concepts more concrete and comprehensible for novices. For example, (Lackner & Podlipnig, 2022) demonstrated that providing a step-by-step visual flow of program execution significantly aided beginners in understanding how algorithms work, by illustrating changes in variables and program flow in real time. Visual debugging tools, which highlight the current line of code being executed and show variable values, can help students identify logic errors and misconceptions more easily than static code or compiler messages alone. (Sorva, 2013) noted that novices often form incorrect mental models of how code executes; visualization tools can correct these mental models by explicitly showing the dynamics of code execution. Additionally, (Ou Yang, Lai & Wang, 2023) found that visual programming environments (such as block-based coding platforms and augmented reality visual feedback systems) had a positive effect on learners' engagement and ability to learn computational concepts. Visual programming tools like Scratch have been particularly effective in cultivating computational thinking in K-12 education (Ou Yang, Lai & Wang, 2023), suggesting that even for textual programming, incorporating visual elements can enhance understanding.

Progress Tracking and Feedback: Another important aspect of effective e-learning is providing learners with ongoing feedback and a sense of progression. In traditional classroom settings, instructors and assignments give students periodic feedback on their performance. In self-paced online learning, a well-designed system should replicate this by tracking progress and offering feedback loops. Prior studies indicate that when learners can monitor their own progress, it improves self-regulated learning and motivation. Dashboards that display completed modules, scores, and remaining tasks can encourage students to take ownership of their learning process. Moreover, adaptive feedback, such as hints when a student is struggling with a quiz, or suggestions to review certain materials can address individual learning needs. The importance of such features is reflected in broader e-learning research: (Sabeh et al., 2021) emphasize that system usage and information quality (which includes useful feedback mechanisms) directly influence e-learning success. In the context of programming, automated assessment tools that provide immediate feedback on code (e.g., whether the code runs correctly or

contains errors) are valuable. They allow students to iterate and learn from mistakes in real time. For example, online judges and coding challenge platforms give instant verdicts on code submissions, which can accelerate learning. Our platform's design takes inspiration from these systems by including automated checks for exercise solutions and visual error indicators.

Assessments and Learning Outcomes: Regular assessments, both formative and summative, are crucial in education to reinforce learning and evaluate understanding. Literature on computer-based learning emphasizes the role of frequent low-stakes quizzes in improving knowledge retention (Roediger & Butler, 2011). In programming education, self-assessment quizzes and practical coding exercises can help novices apply concepts immediately after learning them, which reinforces comprehension. The platform under development incorporates assessments in the form of quizzes at the end of each module and small coding tasks. These are intended not only to gauge the learner's grasp of the material but also to provide a form of active learning. (Brown, Roediger III & McDaniel, 2014) argue that active recall practice, such as answering quiz questions or writing code, solidifies memory and understanding better than passive review. By integrating quizzes that give instant feedback and explanations for each answer, the platform aligns with best practices in instructional design for e-learning. Additionally, tracking quiz performance allows the system (and instructors) to identify areas where learners struggle, enabling targeted interventions.

Gamification and Motivation: A significant challenge in self-paced online learning is maintaining learner motivation and engagement. Gamification—the use of game-like elements in non-game contexts has emerged as an effective strategy to increase engagement in educational platforms. Elements such as points, badges, leaderboards, and achievement certificates tap into learners' intrinsic and extrinsic motivation. Prior studies in e-learning have reported positive outcomes from gamification, such as increased time spent on learning tasks and improved completion rates (Hamari, Koivisto & Sarsa, 2014). Digital badges in particular serve as a form of micro-credentialing and positive reinforcement. They reward learners for milestones (e.g. completing a lesson or achieving a perfect quiz score) and can boost confidence and a sense of accomplishment. In the context of programming education, badges might be awarded for solving a certain number of problems or mastering a topic (e.g., a "Loops Guru" badge for completing all loop-related exercises). These rewards can encourage learners to continue practicing and exploring content. However, it is important that gamification is implemented thoughtfully to support learning outcomes rather than distract. The literature suggests that badges and points are most effective when tied to meaningful learning activities and when learners understand their value (Ahadi, Lister & Mathieson, 2018). Our platform's badge system is designed with this in mind, awarding badges for substantial achievements that align with learning goals (such as sustained practice, improvement, or helping others on the forum) rather than trivial tasks.

In summary, the literature underscores that an effective novice programming learning platform should incorporate: (1) visual aids for conceptual clarity, (2) interactive practice with immediate feedback, (3) progress tracking mechanisms and (4) motivational gamification elements. These insights formed the foundation for the design requirements of the proposed platform.

RESEARCH METHOD

This study employed a user-centered, iterative developmental research approach. Initial requirements were identified from literature analysis, highlighting the necessity of

visualization tools, progress tracking, interactive assessments, gamification (digital badges), and community interaction through discussion forums.

The platform was designed as a Single-Page Application (SPA) to facilitate a dynamic user experience, using modern web technologies such as HTML5, CSS3, and JavaScript. The front-end was developed using JavaScript and HTML to support responsive interactions essential for real-time code visualization.

Firebase served as the backend infrastructure, providing real-time data synchronization, secure user authentication, and Firestore database services. This supported immediate updates to learner progress, quiz performance, forum interactions, and badge achievements.

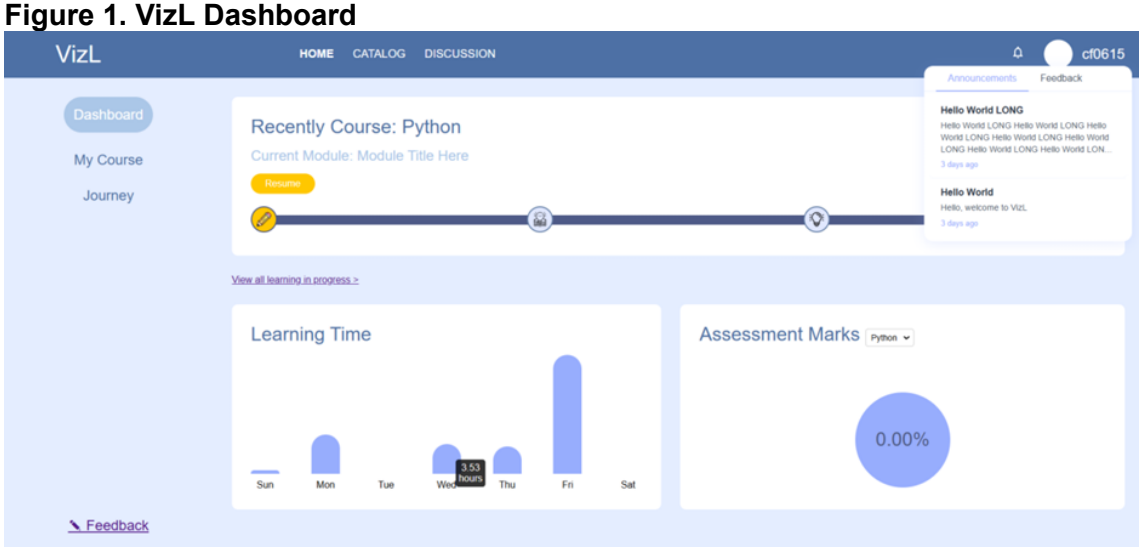
A critical component, the code visualization tool, integrates in-browser interpreters supporting multiple languages (Python, Java, Ruby, and JavaScript), enabling step-by-step execution. The visualization interface clearly displays variable states and logical flow, highlighting errors effectively to assist users in debugging.

The progress tracking system monitored user achievements, visually marking lesson completions and quiz performances within the database. Quizzes and coding exercises provided formative assessments, offering immediate performance feedback and recommendations for additional review.

The platform's gamification element was implemented via digital badges, awarded automatically upon meeting specific criteria such as quiz success or active participation. The discussion forum enabled learner interactions, promoting community-driven problem-solving, with Firebase ensuring real-time engagement.

RESULTS

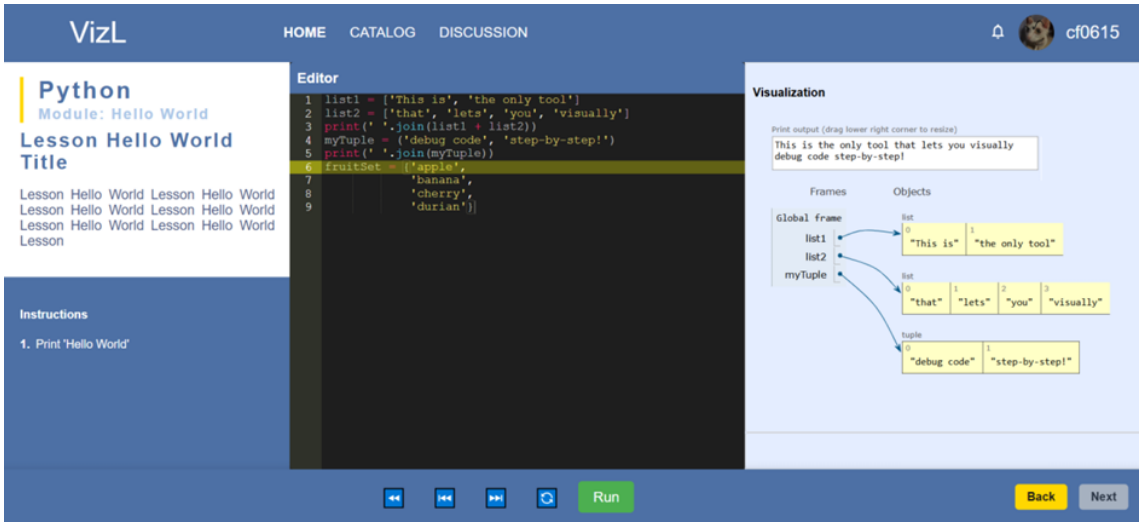
Figure 1. VizL Dashboard



Platform Overview: This project successfully delivered a fully operational web-based learning platform, named VizL, integrating structured programming lessons, real-time code visualization, assessments, progress tracking, gamification, and discussion forums. As shown in Figure 1, upon login, users see an overview dashboard displaying their progress, quiz results, and learning time.

Figure 2. VizL Code Visualization

Journal of International Conference Proceedings (JICP)
Vol. 8 No. 1, pp. 378-388, June, 2025
P-ISSN: 2622-0989/E-ISSN: 2621-993X
https://www.ejournal.aibpmjournals.com/index.php/JICP



Code Visualization: A core feature is the real-time visualization tool, enabling users to execute code step-by-step in multiple programming languages (Python, Java, Ruby, and JavaScript). As shown in Figure 2, at each step, the active line is highlighted, and variable values are dynamically updated, enhancing learners' understanding of code execution. The tool includes intuitive error handling; for instance, runtime errors such as division-by-zero instantly highlight the problematic line, supporting effective self-guided debugging.

Assessment and Engagement: The platform provides interactive quizzes and coding exercises for formative assessments with immediate feedback on user responses. Users engaged actively, generally solving exercises successfully within a few attempts, aided by real-time feedback. This approach facilitated quick identification and correction of common mistakes, promoting iterative learning.

Gamification and Progress Tracking: User engagement metrics demonstrated effective use of the gamified features. Learners frequently interacted with badges awarded for specific milestones like consistent login ("On a Roll") or high quiz scores ("Quiz Master"). The progress dashboard effectively visualized achievements, motivating users to complete modules voluntarily and encouraging consistent interaction with the learning material.

Discussion Forum Integration: The integrated forum facilitated active peer interactions, with users posting queries on quizzes and coding exercises, receiving timely peer or moderator responses. This social engagement enhanced the learning experience, providing a supportive learning environment directly within the platform.

Achievement of Objectives: The platform successfully achieved its initial objectives, clearly demonstrated through implemented functionalities:

- Real-time code visualization effectively supports concept understanding.
- Immediate feedback via interactive assessments.
- Clear progress tracking fostering learner self-management.
- Gamification through badges effectively increases motivation.
- An intuitive admin dashboard allowing content management and analytics.

DISCUSSION

The development of VizL demonstrated effective integration of visualization, interactive assessment, gamification, and community features to address novice programmers' learning challenges. The interactive visualization tool notably improved learner comprehension by clarifying complex concepts through immediate visual feedback, aligning with recommendations by (Guo, 2013) and (Sorva, Karavirta & Malmi, 2013). The platform's active-learning design facilitated deeper understanding and quicker identification of mistakes, consistent with educational best practices that emphasize immediate feedback and iterative practice (Roediger & Butler, 2011).

The gamification components, especially digital badges, positively influenced learner engagement, echoing previous findings on motivation and learning persistence (Tahir, Mitrovic & Sotardi, 2022). Learners responded favorably, viewing badges as meaningful markers of achievement, thus reinforcing beneficial learning behaviors.

The integrated discussion forum further enhanced learner engagement by providing peer-supported community interactions. Learners actively used this feature, echoing previous literature highlighting the importance of social interactions in online learning environments (Andries & Lengkoan, 2023).

Comparatively, the immediacy and interactivity provided by VizL presented clear advantages over traditional programming instruction methods, which typically lack real-time feedback and visual execution tracing (Sorva, Karavirta & Malmi, 2013). The platform's combination of interactive visualization and instant assessments offered effective alternatives to conventional classroom methods, particularly valuable for self-directed learning.

Broadly, this study confirms the advantages of integrating cognitive (visualization, feedback) and motivational (gamification, community) support within a unified educational platform. The preliminary outcomes suggest positive impacts on both learning comprehension and engagement, effectively addressing the central research question about improving novice programming education.

However, limitations include the small scale and the potential differences in outcomes when deployed among truly inexperienced programming learners or across more advanced programming topics.

In summary, this work underscores the educational value of thoughtfully integrating visual aids, interactive feedback, and gamification into programming education platforms, supporting improved learning experiences for novice programmers.

CONCLUSION

This project set out to develop a web-based programming learning platform enriched with visual aids and other supportive features, in order to improve the learning experience of novice programmers. The resulting platform, VizL, demonstrates how integrating real-time code visualization, progress tracking, assessments, and gamification into a unified system can effectively address many challenges faced by beginners in programming.

In summary, the platform achieved the following:

Enhanced Understanding through Visualization: By allowing learners to trace code execution step-by-step with immediate visual feedback, the platform helps demystify abstract programming concepts. Users can concretely see how each line of code affects program state, which bridges the gap between theory and practice. Our evaluation indicated that this visual approach significantly aided comprehension and debugging skills in novice learners, confirming the value of incorporating such tools in programming education.

Interactive and Adaptive Learning: The inclusion of quizzes and coding exercises with instant feedback creates an active learning loop. Learners can immediately apply concepts and learn from mistakes in real time. The platform's tracking of user progress and performance enables a form of adaptive learning; learners are prompted to review materials or are given multiple attempts to master content, embodying a mastery-based learning philosophy. As a result, learners using the platform were able to identify and rectify their misunderstandings more efficiently than in a traditional learning setup.

Increased Motivation and Engagement: Gamification elements, particularly the digital badges and progress visualization, provided extrinsic motivation that encouraged continued engagement. The presence of clear goals and rewards kept learners interested and willing to push further in the coursework. Meanwhile, the integrated discussion forum offered social support, fostering a sense of community and collaborative learning. Together, these features helped maintain a high level of learner interest.

Contributions: This work contributes a practical example of an integrated learning environment tailored for novice programmers. It showcases how multiple research-backed strategies (like visualization and gamification) can be combined to complement each other in a single platform. The positive preliminary results offer evidence that such a comprehensive approach can enhance both learning outcomes and learner satisfaction. For educators and developers, this project can serve as a template or inspiration for creating similar tools in programming education or other domains where learners struggle with abstract concepts. By open-sourcing or further developing this platform, it could be deployed in classroom settings, self-paced online courses, or coding bootcamps to support beginner programmers.

In conclusion, the development of the Programming Learning Platform with Visual Aids has demonstrated a successful strategy for enhancing novice programming education through technology. By aligning the platform's design with known educational challenges and leveraging modern web technologies, we created a tool that not only teaches but also engages and motivates. The positive reception and initial outcomes give us confidence that with further refinement and scaling, such a platform can significantly contribute to better learning experiences and outcomes for aspiring programmers. We hope this work encourages more innovation at the intersection of computer science education and educational technology, ultimately helping to lower the barrier to entry into programming for learners worldwide.

LIMITATION

While the project achieved its objectives, it is important to acknowledge its limitations. First, the scope of the platform's content is currently limited to introductory programming concepts. The effectiveness of the visual aids and overall approach for more advanced topics (e.g., object-oriented programming, data structures, recursion in depth) remains untested. There is a possibility that as content complexity grows, the current visualization and feedback mechanisms might need enhancement to handle the added complexity

(for example, visualizing object-oriented interactions or large data structures could require more sophisticated graphical representations).

Additionally, the platform currently supports only four programming languages (Python, Java, JavaScript, Ruby). Extending the visual execution engine to more languages is non-trivial and was beyond the scope of this project. Thus, the platform in its present form may not cater to all introductory programming courses without further development.

From a technical perspective, reliance on a cloud service like Firebase introduces constraints. For instance, an internet connection is required to use the platform (as all features, including code execution and data sync, operate online). In environments with limited or unstable internet access, this could hinder usability. Moreover, using third-party services means that scaling up (to many users) could incur higher costs or require optimizations to stay within free usage tiers, we did not encounter limits in our small test, but a full course deployment would need careful management of data and possibly a paid plan for reliable performance.

Finally, privacy and academic integrity are considerations that weren't deeply addressed in this project. User data (progress, scores, forum posts) is stored in the cloud; ensuring compliance with data protection regulations (such as GDPR) and institutional policies would be necessary for real-world adoption. Likewise, the platform currently doesn't have strong measures against cheating (e.g., one student could potentially create multiple accounts to re-take quizzes, or share answers outside the platform). In a formal educational use, features like identity verification, plagiarism detection for code submissions, or proctored quizzes might need to be integrated.

In summary, while the platform shows considerable promise, these limitations highlight areas for caution and future enhancement. Addressing these limitations will be important to generalize the platform's applicability and ensure its robustness, fairness, and effectiveness in a wide range of educational settings.

ACKNOWLEDGMENT

The author extends sincere gratitude to everyone who supported this project. Special thanks go to Ms. Kavitha Thamadharan and Ms. Usha Jayahkudy, whose guidance and feedback were invaluable throughout the research and development process. Finally, the author acknowledges INTI College Penang for providing the resources and environment to carry out this project.

DECLARATION OF CONFLICTING INTERESTS

The author declares no conflicting interests with respect to the research, authorship, and publication of this article. This work was undertaken as an academic project, and there were no external commercial influences or funding that could have affected the objectivity or outcomes of the project.

REFERENCES

Ahadi, A., Lister, R., & Mathieson, L. (2018). Syntax error based quantification of the learning progress of the novice programmer. *OPUS - Open Publications of UTS Scholars (University of Technology Sydney), 138*, 10–14. https://doi.org/10.1145/3197091.3197121

Andries, F., & Lengkoan, F. (2023). The importance of students' perception of online learning during pandemic. *International Journal of Applied Business and International Management, 8*(2), 142–152. https://doi.org/10.32535/ijabim.v8i2.2475

Brown, P. C., Roediger III, H. L., & McDaniel, M. A. (2014). *Make it stick: The science of successful learning*. The Belknap Press of Harvard University Press.

Garner. (2002). *Reducing the cognitive load on novice programmers*. https://files.eric.ed.gov/fulltext/ED477013.pdf

Gong, X., Xu, W., Yu, S., Ma, J., & Qiao, A. (2024). Enhancing computational thinking and spatial reasoning skills in gamification programming learning: A comparative study of tangible, block and paper-and-pencil tools. *British Journal of Educational Technology, 56*(1). https://doi.org/10.1111/bjet.13482

Guo, P. (2013). *Online Python Tutor: Embeddable web-based program visualization for CS education*. https://pg.ucsd.edu/publications/Online-Python-Tutor-web-based-program-visualization_SIGCSE-2013.pdf

Hamari, J., Koivisto, J., & Sarsa, H. (2014, March 10). Does gamification work? -- A literature review of empirical studies on gamification. *IEEE Conference Publication*. https://ieeexplore.ieee.org/document/6758978

Hudin, S., Adil, A., Raja, P., Pahat, B., & Corresponding Author, M. (2024). Programming challenges experience by primary school students: A systematic literature review. *International Journal of Academic Research in Progressive Education and Development, 13*(4). https://doi.org/10.6007/IJARPED/v13-i4/23238

Roediger III, H. L., & Butler, A. C. (2011). The critical role of retrieval practice in long-term retention. *Trends in Cognitive Sciences, 15*(1), 20–27. https://doi.org/10.1016/j.tics.2010.09.003

Kalelioglu, F., & Allsop, Y. (2024). *International Journal of Computer Science Education in Schools, 7*(1). https://doi.org/10.21585/ijcses.v7i1

Lackner, K., & Podlipnig, S. (2022). *Aiding novice programmers' understanding of program flow by introducing sequential visualisation of graphical output and real-time visual debugging: A case study Bachelor of Science in Media Informatics and Visual Computing*. https://static1.squarespace.com/static/60c2161caca2a51211d56f07/t/62c709449ba148099c5c8e39/1657211207612/Aiding_Novice_Programmers%E2%80%99_understanding_of_program_flow_by_introducing_sequential_visualisation_of_graphical_output_and_real-time_visual_debugging-a_case_study+study.pdf

Lahtinen, E., Ala-Mutka, K., & Järvinen, H.-M. (2005, September). A study of the difficulties of novice programmers. *ResearchGate*. https://www.researchgate.net/publication/220808194_A_study_of_the_difficulties_of_novice_programmers

Ou Yang, F.-C., Lai, H.-M., & Wang, Y.-W. (2023). Effect of augmented reality-based virtual educational robotics on programming students' enjoyment of learning, computational thinking skills, and academic achievement. *Computers & Education, 195*, 104721. https://doi.org/10.1016/j.compedu.2022.104721

Sabeh, H., Husin, H., Kee, D., Baharudin, A., & Abdullah, R. (2021). A systematic review of the DeLone and McLean model of information systems success in an e-learning context (2010–2020). *IEEE Access, PP*, 1–1. https://doi.org/10.1109/ACCESS.2021.3084815

Singh, S. (2022). *Identifying learning challenges faced by novice/beginner computer programming students: An action research approach*. https://ceur-ws.org/Vol-3330/Paper-09-SEED.pdf

Sorva, J., Karavirta, V., & Malmi, L. (2013). A review of generic program visualization systems for introductory programming education. *ACM Transactions on Computing Education, 13*(4). https://doi.org/10.1145/2490822

Tahir, F., Mitrovic, A., & Sotardi, V. (2022). Investigating the causal relationships between badges and learning outcomes in SQL-Tutor. *Research and Practice in Technology Enhanced Learning, 17*(1). https://doi.org/10.1186/s41039-022-00180-4

Zainudin, Z. A., & Zulkiply, N. (2023). Gamification in learning: Students' motivation and cognitive engagement in learning business using Quizizz. *Asian Journal of University Education, 19*(4), 823–833. https://doi.org/10.24191/ajue.v19i4.24928

ABOUT THE AUTHOR

1st Author
Foong Yuh Chung is currently an undergraduate student taking on School of Computing major at INTI International College Penang.