CODE QUEST: DEVELOPMENT OF A GAMIFIED CODE CHALLENGE SYSTEM FOR PROGRAMMING LEARNERS

DELWIN N. CALICA

HEROBIN A. MARIÑASxsdsdsd

VIEN IVORY R. MARZAN

DON MARIANO MARCOS MEMORIAL STATE UNIVERSITY NORTH LA UNION CAMPUS COLLEGE OF INFORMATION SYSTEMS

BACNOTAN LA UNION

BACHELOR OF SCIENCE IN INFORMATION SYSTEMS

JUNE 2025

1

Chapter 1

INTRODUCTION

Situation Analysis

Learning how to program, or “coding,” is like learning a new language but instead of communicating with people, you’re communicating with machines. As digital devices become integral to everyday life, programming has become an essential skill, comparable to reading or math. Yet, for many students, learning to code remains an intimidating and frustrating experience. This challenge is not limited to one country it’s a global concern, especially pressing in developing nations like the Philippines.

Globally, the importance of programming and computational thinking is well recognized. Countries that prioritize digital skills development tend to have stronger economies and greater innovation potential. As automation and artificial intelligence transform the workforce, digital fluency, not just digital literacy is now essential. This means understanding not just how to use technology, but how it works at a structural level, including logic, algorithms, and programming fundamentals (Mark Guzdial, 2021).

To respond to this shift, many nations have embedded Computational Thinking (CT) into their K–12 curricula. CT defined as a problem-solving approach foundational to computer science is increasingly regarded as a basic literacy skill (Wing, 2021). Recent studies (e.g., Bocconi et al., 2022; Kite & Park, 2023) continue to demonstrate that countries with strong CT integration often introduce programming in primary school and offer more advanced computing tracks in secondary education.

2

A study by Montiel & Gomez-Zermeño (2021) conducted a systematic literature review focusing on the use of Scratch, a block-based programming language, in fostering computational thinking among K–12 students. The review identified several challenges in implementing CT education, including limited teacher training, insufficient integration of CT across various subjects, and a predominant focus on tool usage over conceptual understanding. These findings highlight the necessity for comprehensive professional development programs and curriculum designs that embed CT principles across disciplines, ensuring a more holistic approach to CT education.

Singapore exemplifies a forward-thinking model. Through its Infocom Media 2025 masterplan, the country promotes scaffolded, personalized learning and integrates CT across subjects (Teo, 2021). Teacher training and adaptive learning tools are central to this strategy. Similarly, Finland introduces algorithmic thinking from Grade 1 as part of its cross-curricular approach, aiming to build logical reasoning skills early (Jo Earp, 2021).

Advanced educational frameworks support these global strategies. Cognitive Load Theory (CLT) suggests that students learn better when instructional design accounts for mental processing limits. Programming, being cognitively demanding, benefits from strategies that manage complexity, reduce distractions, and support deep learning (Cognitive Load Theory: Implications for Instructional Design in Digital Classrooms). Scaffolded instruction breaking content into smaller steps with guided support helps learners build confidence and understanding over time (Becker & Quille, 2021). Adaptive learning systems, powered by AI, can dynamically adjust difficulty levels to suit individual learners (llie Gligorea, 2023).

3

In addition to individualization, collaborative learning approaches are gaining traction. (Sol et al., 2021) conducted a systematic review and found that computersupported collaborative learning (CSCL) significantly enhances programming education. Their analysis of twelve studies revealed that CSCL not only improves students’ programming skills but also fosters engagement and motivation. The review highlighted that object-oriented programming languages are frequently used in CSCL environments and that such collaborative approaches are effective in diverse educational settings.

Recent advancements in AI have introduced innovative approaches to collaborative learning in programming education. (Fan et al., 2025) conducted a comparative study examining AI-assisted pair programming’s effects on undergraduate students’ intrinsic motivation, programming anxiety, and performance. The study revealed that AI-assisted pair programming significantly increased intrinsic motivation and reduced programming anxiety compared to individual programming. Moreover, students in the AI-assisted groups outperformed those in both individual and traditional human–human pair programming settings. These findings suggest that integrating AI tools can enhance learning outcomes by providing personalized support and reducing the cognitive load associated with programming tasks.

Constructivist approaches, like Project-Based Learning (PBL), are also effective. PBL encourages students to build knowledge through real-world problem-solving, enhancing engagement and practical skill development. Research supports its impact on motivation, collaboration, and critical thinking (Moldez et al., 2024).

Psychological barriers also play a role. Impostor syndrome, low self-confidence, and anxiety often deter beginners. Peer mentorship and community support have proven

4

effective in improving self-efficacy and reducing dropout rates (Jennifer Teshera-Levye, 2024). Gamification, when combined with real-time feedback and recognition, can also help students persist through challenges (Koukopoulos & Karampelas, 2022).

Another emerging strategy in addressing learning gaps is the use of Intelligent Tutoring Systems (ITS), which combine artificial intelligence with personalized instruction. (Ba & Hu, 2023) demonstrate that ITS platforms significantly improve student outcomes in programming education by providing adaptive content, immediate feedback, and personalized learning pathways. Their study highlights tools like AutoTutor and CodeWorkout, which help simulate individualized instruction and support students in developing core coding skills. These systems are particularly useful in resource-constrained environments, such as many schools in the Philippines, where teacher availability and student engagement are persistent challenges.

Another key issue is the mismatch between teaching pace and student learning styles. Some students pick up programming quickly; others need more time and support. A uniform teaching method often leaves many behind. In the Philippines, the Educational Research and Innovations Center (Educational Research and Innovations Center (ERIC), 2022) highlights that schools often lack trained ICT teachers and have inconsistent curriculum delivery. As a result, up to 40% of students report falling behind or feeling disengaged (Krajcik & Blumenfeld, 2022).

In the Philippines, efforts to improve programming education face several structural barriers. While national programs like DICT’s Coding for Kids and DepEd’s K–12 ICT tracks exist, implementation remains uneven. One of the most significant issues is limited technological infrastructure. Many rural schools face intermittent internet access, outdated

5

hardware, and poor computer-to-student ratios (Erik Kormos, 2021). Students often rely on smartphones, which are poorly suited for complex coding tasks.

Another critical bottleneck is teacher training. Schools often lack qualified ICT instructors, and even those who teach programming may lack experience with modern tools or methods (Educational Research and Innovations Center (ERIC), 2022). Professional development opportunities are rare, and large class sizes make personalized support difficult.

Cultural and societal factors also shape learning experiences. Programming is often viewed as overly technical or only for “geniuses,” discouraging many students especially girls from pursuing it (Sofía González Gallego, 2025). Fear of failure and high academic pressure can suppress creativity and risk-taking, both of which are crucial in programming. Moreover, employers report that graduates often lack soft skills like teamwork and communication, which are essential in the tech industry.

A study by Elsawah & Hill (2023) highlighted that both students and faculty members were not fully prepared to undergo full online learning. The study identified key barriers such as difficulty in clarifying topics with professors, lack of suitable study environments, and poor internet connectivity, all of which hinder effective programming education in the online setting.

Despite the challenges facing programming education in the Philippines, there are promising solutions emerging. One key problem is the lack of hands-on practice.

Many programming classes emphasize theory over real coding experience. As Zhang & Ma (2023) note, insufficient opportunities for students to write code lead to a disconnect between understanding concepts and applying them. To address this, local platforms like

6

CodeChum provide an integrated development environment (IDE) for C programming with instant feedback, problem stories, and leaderboards—aligning well with best practices in gamified and adaptive learning (ERIC, 2022).

Another challenge is the lack of timely support. Programming requires frequent debugging and problem-solving, and without immediate guidance, students may feel lost and demotivated. (Pursel & Zhang, 2022) found that nearly 30% of students drop out of introductory programming due to feeling isolated or overwhelmed. To respond, organizations like EskwelaCode offer peer-supported bootcamps, while some educators have turned to flipped classroom approaches (Joaquin VII & Andal, 2023) that emphasize student-centered learning and in-class collaboration, giving learners more support where it’s most needed.

Students also struggle with low engagement and motivation. Programming is often presented in abstract ways, making it hard to relate to or enjoy. Poor instructional design can reduce student engagement by up to 50% (Koukopoulos & Karampelas, 2022; Qian Fu, 2023). To solve this, some educators are using visual programming languages like Scratch and Blockly, which simplify complex concepts for beginners (Meta-analysis, 2021). Meanwhile, gamified environments like CodeChum use game elements such as points and challenges to make coding fun and interactive, increasing motivation and persistence (Cuervo-Cely et al., 2022).

Rigid instruction is another issue. Many courses follow fixed schedules, limiting flexibility and causing some students to fall behind. This inflexible pacing doesn’t accommodate different learning speeds or backgrounds. In response, online platforms are offering more adaptive and flexible course structures. A study by Dela Rosa (2023) at

7

Bulacan State University showed that an online Python course with well-planned content, time flexibility, and active communication with instructors and peers was perceived as highly effective by students.

Finally, there is a clear gap between education and industry requirements. Even those who finish programming courses often graduate lacking practical skills like debugging, version control, and teamwork. This mismatch weakens the country’s ability to innovate and compete globally (Global Recognition of Computational Thinking and Educational Strategies, 2021–2025). Regional programs such as “Hack for Change” hackathons in Northern Luzon address this by promoting collaborative, communitybased project learning, helping students gain real-world experience and build practical skills (UN-Habitat, 2021; Hackathon Hub Philippines, 2023).

At the Don Mariano Marcos Memorial State University – North La Union Campus (DMMMSU–NLUC), students in the Bachelor of Science in Information System (BSIS) program continue to face challenges in learning programming. Although the department has its own computer laboratories, students still experience difficulties. In some programming classes, lessons are taught through lectures where teachers explain topics and students listen. Because of this, many students struggle to apply what they’ve learned in actual programming exercises. While a few students are already comfortable with programming, most are still having a hard time keeping up. Students learn at different speeds, and some lack confidence or experience when using programming tools. Many students also feel shy or hesitant to ask questions when they don’t understand the topic, which leads to confusion and gaps in their learning.

8

In addition to these issues, not all students own laptops or computers, limiting their ability to practice programming outside school hours. To help address this, the university provides desktop computers that students may use during the day at the SAS office and in the library. However, since these devices cannot be brought home, the accessibility remains limited after school hours.

Even when labs are available, insufficient practice time due to limited lab schedules remains a concern. To address this, instructors can introduce online platforms and mobile-friendly compiler like W3Schools, allowing students to code using smartphones or shared devices. Assigning modular, self-paced activities also provides flexibility and more chances for independent learning.

Furthermore, the heavy cognitive load during early programming courses overwhelms many beginners. To lessen this, teachers can apply scaffolded instruction— starting with basic concepts and gradually introducing complexity. Using mini-projects, visual aids, and chunked lessons can help students build confidence and understanding. Peer tutoring and pair programming are also effective in promoting collaborative learning.

Lastly, the fear of making mistakes discourages many students from experimenting or asking questions, which is essential for developing coding skills. Creating a safe, encouraging classroom environment where errors are viewed as learning opportunities can help. Teachers are encouraged to cultivate a growth mindset, include debugging exercises, and openly discuss common coding challenges to normalize trial and error in the learning process.

9

This paper aims to give a simple and helpful solution to the problems students face when learning programming. The researchers created CodeQuest, a system that makes coding easier and more fun. Instead of relying only on classroom lessons or limited computer access, students can now learn using their phones or any device. CodeQuest lets them practice coding anytime, do activities at their own pace, and get instant feedback. It also has a forum where they can ask questions and help each other. With this system, learning programming becomes less stressful and more exciting. It saves time, helps students learn better, and makes the whole experience more interactive and flexible.

As shown in Figure. 1, the study used I-P-O (Input – Process – Output) format. The input box includes the following: (1) data requirements which include Name, Age, Section, Course, Login Credential and User Profile Information; (2) document requirements which include Modules, Exam, Exercises, Challenges and Video Tutorials. The process box includes the following: (1) System development using Rational Unified Process methodology (2) Evaluate the vulnerability test of the system (3) Evaluate the technical performances of the system (4) Determine the usability of the developed system .The output box consists of the developed CodeQuest a platform a platform both a mobile and web application designed to help students learn programming languages more effectively. The broken line denotes the feedback or comments coming from the expert evaluators and users which will be considered for improvement of the system.

10

Figure 1. Research Paradigm

Statement of Objectives

The main objective of this study is to develop CodeQuest, a platform both a mobile and web application designed to help students learn programming languages more effectively. This platform aims to address common challenges such as insufficient practice, varying learning speeds, limited support, and low motivation among students.

Specifically, this study will address the following:

assess the readiness of the students and faculty of the College of Information System (CIS) in Sapilang, Bacnotan, La Union, DMMMSU North La Union

Campus towards the implementation of the developed system.

evaluate the performance of the developed system as to: a. response time;

11

throughput; c. error rate; and

the vulnerability of the developed system.;

determine the usability of the developed system.

Time and Place of the Study

To finish the study, the researchers will conduct a series of activities presented in Figure 2 on page 12. The initial part of this project includes concept evaluation and series interviews that took place at College of Information Systems, DMMMSU-NLUC. Title defense was conducted at the College of Information Systems in the presence of OREC members. All their comments and suggestions were carefully compiled and incorporated for the improvement of the study. Furthermore, a series of interviews were conducted to gather data to be used in the design of the system.

Figure 2. Schedule of Activities

12

Definition of Terms

CodeQuest:Development of Of Gamified Code Challenge System for Programming Learners refers to the system developed in this study. It is a mobile and web app that helps students learn programming in a fun and interactive way.

Document Requirements refers to the learning materials such as modules, exams, exercises, challenges, and video tutorials that the system needs to help users learn programming.

Error Rate refers to how often mistakes or problems happen in a system during operation.

Feedback refers to comments and suggestions from users to help improve the system.

Functional Requirements refers to the actions the system should be able to do, like logging in, answering quizzes, and viewing scores.

Gamified Code Challenge System refers to a learning tool that uses game elements such as points and levels to make learning programming more fun and engaging.

Non-Functional Requirements refers to how the system performs in terms of speed, safety, and ease of use.

Readiness refers to how prepared students and teachers are to use the system, including their access to technology and willingness to learn.

Response Time refers to how fast a system reacts after a user makes a request. System Development refers to the process of designing and building the CodeQuest

system.

13

Usability refers to how easy and helpful the system is for its users, especially students and teachers.

Vulnerability refers to how safe the system is from hacking, errors, or other security issues.

Throughput refers to how much data or how many tasks a system can handle in a certain amount of time.

Vulnerability refers to a test conducted to check if the system is protected from hacking or technical problems.

14

Chapter 2

METHODOLOGY

Research Design

This study will use Applied Research. Applied research studies are widely used in finding practical solutions to real-world problems. Instead of just studying something for knowledge's sake (which is what basic research does), applied research focuses on solving specific issues, improving processes, and creating new innovations (Peter Joore, Guido Stompff, 2022) .

Applied research aims to produce answers that can be directly used to enhance processes, goods, or services. It crosses numerous disciplines and frequently entails experimenting, testing, and the creation of new procedures or technologies that promote social growth and innovation. Applied research is critical in sectors such as healthcare, engineering, environmental sciences, and technology, because its conclusions lead to practical benefits and improvements (von Larcher, T., Poole, C. & Sahraie, 2024)

In this study, applied research will be used in the development of Code Quest: Development of a Gamified Code Challenge System for Programming Learners for the College of Information Systems to identify specific issues that a system aims to address, ensuring that development focuses on solving real-world problems. By collecting data through surveys, observations, and user testing, researchers can refine the system’s features to improve its functionality.

15

Materials and Procedures

In order to complete this study, the following materials and procedures will be used and followed:

Materials

Software

In developing the system, this study will utilize Firebase as a real-time server during system development. Flutter enables the creation of cross-platform mobile applications, while Android Studio offers powerful emulators, debugging tools, and performance analysis to refine the app throughout development. It also functions as a text editor. Additionally, Dart is employed for both mobile and web applications.

Hardware

In the system development, the researchers will utilize an Intel(R) Core (TM) i31205U CPU 1.60 GHz (up to 1.6 GHz), with 8.00 GB of RAM (7.84 GB usable) and a 64bit operating system running on an x64-based processor. Additionally, this study will use smartphones that operate on Android 7 or higher. These devices must have a minimum of 4 GB of RAM and at least 64/128 GB of internal storage to accommodate app installations. Furthermore, they should support Full HD (1920x1080) resolution or higher for optimal visual clarity. These devices will be employed to install and test the app, ensuring it runs smoothly on actual hardware.

Procedures

The concept of the system (see Figure 3, page 17). As presented in the Figure, the system is designed for the College of Information Systems to help students to enhance their learning skills and knowledge in learning programming languages. The process begins

16

when the student logs in using their authorized credentials. Upon successful login, the system securely stores and accesses the student's profile information. The student can enroll in a course, take a challenge, watch tutorials, and post in a forum. The teacher has the ability to add other teachers as collaborators in the course, publish challenges, video tutorials, and courses, and approve or deny student enrollment. On the other hand, the other teacher will receive a notification as a collaborator. Once a decision is made, the student receives quick notice from the system confirming their enrollment and allowing them to take the course.

This study followed the Rational Unified Process methodology (see Figure 4, page

with four (4) phases that consisted of Inception, Elaboration, Construction, and Transition shown in Figure 4. This methodology was used in this study because it focuses on the project objectives from start to finish. It also focuses on the importance of early planning to avoid risks, increase productivity and meet customer satisfaction.

In phase 1 (Inception), the scope of the system and the key requirements are identified to gain a clear understanding of the project. The researchers will be conducted a series of interviews to determine the essential requirements for developing the system. This phase also focused on assessing feasibility, defining the initial system vision, and establishing the necessary documents, data, and permissions.

In phase 2 (Elaboration), the researchers will transform the agreed specification and scope into a use-case diagram see figure 5 to outline the roles of each user and the key functionalities. In this phase, it will be decided if the project is a “go” or “no go”.

17

Figure 3. Concept of The Study

In phase 3 (Construction), the researchers will convert the presented use case diagram into source code. A series of testing will be conducted in this phase to ensure the quality of the system.

Figure 4. Rational Unified Process Diagram

In phase 4 (Transition), the system will be deployed for use by College of Information Systems. Sample login credentials will be provided to Administrators,

18

Teachers, and Students to test and operate the live system. Acceptabilit assessment using a SUS-BASED questionnaire and beta testing was simultaneously conducted. Any feedback from the client was implemented in the system, and defects will be fixed in this phase.

Post activity involved production of the final project. In this phase, training for the clients, maintenance and updates will be conducted.

Fig. 5 Use Case Diagram of the Study

19

Data to be Gathered

The study will be conducted at the College of Information Systems at Don Mariano Marcos State University - North La Union Campus. In this study, there are three (3) types of respondents, IT Experts, Faculty and Students. Overall, there are a total of 120 respondents. In selecting the sample for IT experts, Purposive Sampling will be used. All faculty will serve as respondents. Moreover, quota sampling will be used for students.

Questionnaires, interviews, and checklists will be used in this study as the main tool in gathering data. An interview guide (see Appendix A on page 27) will be used to determine the current situations in learning programming at the College of Information Systems. For Readiness, Frequency and Percentage will be used to assess if they are ready (see Appendix B on page 28) of the College of Information Systems towards the implementation of Code Quest. Its purpose is to determine if the respondents are prepared for the implementation of technology; if not, appropriate training will be offered.

SUS questionnaire (see Appendix C on page 33) will be used to determine the usability of the developed system in the perspective of the Students, Faculty and IT Experts.

Table 1. Respondents of the Study

Data Analysis

This study will be using different statistical tools in treating the gathered data using the method mentioned above.

20

The criteria for the readiness of the developed systems are only answered by “YES” or “NO”. The developed system was deemed “Acceptable” if the overall rating in each criterion is within 51-100% and was “Not Acceptable” if the overall rating in each criterion is between 1-50% as seen in the statistical below:

The collected data will be analyzed, for questions 1, 3, 5, 7, and 9, the score distribution will be “Strongly Disagree = 1”, “Disagree = 2”, “Neither Agree nor Disagree

3”, “Agree = 4”, and “Strongly Agree = 5”. As for questions 2, 4, 6, 8, and 10, the score distribution will be “Strongly Disagree = 5”, “Disagree = 4”, “Neither Agree nor Disagree

3”, “Agree = 2”, and “Strongly Agree = 1”. After add values and multiply the total by

2.5. Calculate the mean to find the score. The total score should end up with a range between 0 and 100. The higher the score the more usable the website is. Values above 68 are considered a good usability score.

In terms of vulnerability testing using the OWASP ZAP tool there are four (4) categories of risks: high, medium, low, and informational.

21

LITERATURE CITED

Ba, S., & Hu, X. (2023). Measuring emotions in education using wearable devices: A

systematic review. Computers & Education, 200, 104797.

https://doi.org/10.1016/j.compedu.2023.104797

Becker, K., & Quille, K. (2021). Experiences of early assessment to teach functional

programming. Journal of Functional Programming. https://www.cambridge.org/core/journals/journal-offunctionalprogramming/article/experiences-of-early-assessment-to-teachfunctionalprogramming/667F18D3611D4A3A7C7C6A2A2FA4FB3C

Bocconi, S., Chioccariello, A., Kampylis, P., Dagienė, V., Wastiau, P., Engelhardt, K., …

Punie, Y. (2022). Reviewing computational thinking in compulsory education: State of play and practices from computing education (JRC Report JRC128347).

Publications Office of the European Union. https://publications.jrc.ec.europa.eu/repository/handle/JRC128347

Cuervo-Cely, K. D., Restrepo-Calle, F., & Ramírez-Echeverry, J. J. (2022). EFFECT OF GAMIFICATION ON THE MOTIVATION OF COMPUTER PROGRAMMING STUDENTS. Journal of Information Technology Education: Research, 21. https://doi.org/10.28945/4917

Dela Rosa, A. P. (2023). Effectiveness of an Online Course in Programming in a State University in the Philippines. International Journal of Computing Sciences Research, 7, 1685–1698. https://doi.org/10.25147/ijcsr.2017.001.1.127

Educational Research and Innovations Center (ERIC). (2022). Gamification as an Innovative Strategy to Improve Learners’ Writing Skills. Sustainability.

Elsawah, W., & Hill, C. (2023). Barriers to programming education in UAE primary schools: a qualitative review from ICT teachers’ perspectives. Discover Education, 2(1). https://doi.org/10.1007/s44217-023-00043-0

Erik Kormos. (2021).Rural Schools and the Digital Divide. Https://doi.org/10.3776/TPRE.2021.V11N1P25-39

22

Fan, G., Liu, D., Zhang, R., & Pan, L. (2025). The impact of AI-assisted pair programming on student motivation, programming anxiety, collaborative learning, and programming performance: a comparative study with traditional pair

programming and individual approaches. International Journal of

STEM Education, 12(1), 16.

https://doi.org/10.1186/s40594-025-00537-3

Jennifer Teshera-Levye. (2024). Peer mentorship and academic supports build sense of community and improve outcomes for transfer students.

Joaquin VII, A. R., & Andal, E. (2023). Flipped Classroom Approach and Motivation in the Acquisition of Practical Skills. International Journal of Educational

Management and Development Studies, 4(3), 16–38. https://doi.org/10.53378/352997

Pursel, B. K., & Zhang, L. (2022). Student Disengagement and Dropout Factors in Introductory Programming Courses. Journal of Computer Assisted Learning, 32(3), 202–217. https://doi.org/10.1111/jcal.12131

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Jo Earp. (2021). Integrating algorithm tasks into early years teaching. https://www.teachermagazine.com/sea\_en/articles/integrating-algorithm-tasksinto-earlyyears-teaching

Koukopoulos, K., & Karampelas, A. (2022). Enhancing Student Motivation and Engagement through a Gamified Learning Environment. Sustainability, 15(19), 14119. https://doi.org/10.3390/su151914119

Krajcik, J. S., & Blumenfeld, P. C. (2022). IF Science AND Making AND Computing:

Insights for Project-Based Learning and Primary Science Curriculum Design.

Journal of Education for Teaching, 50(2), 244–257. https://doi.org/10.1080/02607476.2022.2141503 llie Gligorea. (2023). Adaptive Learning Using Artificial Intelligence in e-Learning: A Literature Review.

Mark Guzdial. (2021). A Digital Fluency Framework to Support 21st-Century Skills.

23

Moldez, C., Crisanto, M. A., Gian, M., Cerdeña, R., Maranan, D. S., & Figueroa, R. (2024). ASEAN Journal of Open and Distance Learning (AJODL) https://ajodl Innovation in Education: Developing and Assessing Gamification in the University of the Philippines Open University Massive Open Online Courses (Vol. 16, Issue 1). https://ajodl.oum.edu.my/

Montiel, H., & Gomez-Zermeño, M. G. (2021). Educational Challenges for Computational Thinking in K–12 Education: A Systematic Literature Review of “Scratch” as an

Innovative Programming Tool. Computers, 10(6), 69.

https://doi.org/10.3390/computers10060069

(PDF) Cognitive Load Theory\_ Implications for Instructional Design in Digital Classrooms (1). (n.d.).

Peter Joore, Guido Stompff, J. van den E. (2022). Applied Design Research A Mosaic of 22 Examples, Experiences and Interpretations Focussing on Bridging the Gap between Practice and Academics.

Joaquin VII, A. R., & Andal, E. (2023). Flipped Classroom Approach and Motivation in the Acquisition of Practical Skills. International Journal of Educational

Management and Development Studies, 4(3), 16–38. https://doi.org/10.53378/352997

Pursel, B. K., & Zhang, L. (2022). Student Disengagement and Dropout Factors in Introductory Programming Courses. Journal of Computer Assisted Learning, 32(3), 202–217. https://doi.org/10.1111/jcal.12131

Wing, J. M. (2006). Computational thinking. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215

Qian Fu, Y. Z. (2023). Effects of different feedback strategies on academic achievements, learning motivations, and self-efficacy for novice programmers.

Sofía González Gallego. (2025). A critical examination of the underlying causes of the gender gap in STEM and the influence of computational thinking projects applied in secondary school on STEM Higher Education.

24

Sol, R., Santos, E., Reis, M. & Pereira, L. (2021). Computer Supported Collaborative Learning for Programming: A Systematic Review. Proceedings of the 13th International Conference on Computer Supported Education (CSEDU), pp. 184– 191. DOI: 10.5220/001040700184019

Teo, T. W. (2021). Infocomm Media 2025 masterplan. https://www.imda.gov.sg//media/imda/files/infocomm-media-2025-fulreport.pdf Zhang, L., & Ma, Y. (2023). A study of the impact of project-based learning on student learning effects: a meta-analysis study. In Frontiers in Psychology (Vol. 14).

Frontiers Media SA. https://doi.org/10.3389/fpsyg.2023.1202728

25

APPENDICES

26

APPENDIX A

INTERVIEW GUIDE

Name of Interviewee: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Address of Establishment: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Name of Interviewer: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

A. Nature of Work (Service Rendered)

Q1: What do you do to help students learn programming?

Q2: What problem do you usually experience teaching programming?

Q3: What techniques or materials do you use teaching programming?

Q4: What difficulties have you faced when trying new ways of teaching programming?

B. Practice in Teaching

Q1: What strategies do you currently use to teach programming effectively?

Q2: How do you ensure student engagement and participation in programming lessons?

Q3: How do you assess student progress and comprehension in programming subjects?

Q4: What challenges have you faced in teaching programming, and how have you addressed them?

Q5: What types of instructional or technological support do you require to enhance programming education?

Q6: Based on your experience, what improvements or additional resources would help improve programming instruction?

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(Signature over Printed Name)

Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

27

APPENDIX B

READINESS QUESTIONNAIRE

Dear Respondents,

We are students of Don Mariano Marcos Memorial State University-North La

Union Campus. Currently, we are working on a research paper entitled “Code Quest”. In this regard, we are seeking your assistance to share your perceptions regarding the extent of readiness towards the implementation of the developed system as specified in the survey questionnaire.

Rest assured that your responses would be kept in strict confidence and will be used solely for academic purposes.

Thank you very much.

Very truly yours,

DELWIN N. CALICA HEROBIN MARIÑAS VIEN IVORY R. MARZAN Researchers

Part I. Profile of the Respondents

Name (Optional) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Sex: \_\_\_\_\_\_\_ Age: \_\_\_\_\_\_\_\_

Designation: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Part II. Extent of Readiness

A. Technology Access

Direction: Please indicate a check (√) mark on the box if you have accessed the following technologies

I own a computer (pc, laptop) / smartphone

I have access to reliable internet connection in my computer/smartphone

28

I have heard about the following technology:

computer

smartphone

E-learning platform

I am familiar how the following hardware component works:

computer

smartphone

B. Technological Confidence towards the implementation of Code Quest for CIS Direction: Below are the indicators in measuring your technological confidence. Please rate the extent to which you agree or disagree with the following statements using the scale below:

29

smartphone

E-learning platform

I know how to open an application in my computer/mobile

I know how to use applications such as google classroom, Microsoft teams etc.

Attitudes towards the implementation of Code Quest for CIS

Direction: Below are the indicators in measuring your attitude towards the implementation of Code Quest for CIS. Please rate the extent to which you agree or disagree with the following statements using the scale below:

30

I am motivated to the implementation of Code Quest for CIS because it is helpful

31

Supporting data

1. Do you have any reservations/concern about e-learning/digital platforms?

Please

specify :\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

How soon can you implement a Code Quest for CIS in the institution? Please check your answer from the options below

Within the next month

Within the next six months

Within the next 12month

Comments:

Beyond 1 year from now

Never, please specify the

reason: \_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of the Evaluator

32

APPENDIX C

33

34

Appendix D

Activity Diagram (Code Quest)

35

CURRICULUM VITAE

36

37

38

39

40

41

42

43

44

45