EduVision: Journal of Innovations in Pedagogy and Educational Advancements
Volume 2, Issue 4, April 2026
brightmindpublishing.com
ISSN (E): 3061-6972

DEVELOPING AN ALGORITHM AND SOFTWARE FOR A MOBILE EDUCATIONAL GAME APPLICATION: A CASE STUDY IN TEACHING COMPUTER SCIENCE

Gulmurodova Dinora
Assistant Department of Artificial Intelligence,
Tashkent State University of Economics,PhD Student,
Jizzakh Branch of the National University of Uzbekistan,
d.gulmurodova@tsue.uz

Khayrullayev Muminbek
Assistant Department of Artificial Intelligence,
Tashkent State University of Economics
mominbek97gmail.com

Gulmurodova Shohsanam
Master's Student of the Jizzakh
Branch of the National University of Uzbekistan
gulmurodovadinora3@gmail.com

Turobova Roza
Master's Student of the Department of Pedagogy of the
National Pedagogical University of Uzbekistan named after Nizami
turobovaroza02@gmail.com

Abstract
The rapid integration of mobile devices into everyday life has created great opportunities to revise how high school students engage in technically complex subjects such as computer science. This work presents the design, algorithmic architecture, and empirical assessment of the Android mobile application (.apk), specially created to support the teaching of computer science in grades 5-9 in the general secondary education system of Uzbekistan. The application, based on the theory of Mobile Game-Based Learning (MGBL), uses the Room-SQLite relational database, a modular question bank system with Kotlin with Jetpack Compose to reflect the reactive interface, and a weighted evaluation algorithm, taking into account

the complexity of the questions. The assimilation of the content of the quasi-experimental study conducted with the participation of 52 students of school No. 38 of the Gallaorol district of the Jizzakh region, divided into control and experimental groups, increased by 16 percent.

Keywords: mobile game-based learning, Informatics education, Android application, Kotlin, Jetpack Compose, Room database, gamification, Uzbekistan secondary education, algorithmic thinking, offline learning

Introduction

The widespread intersection of mobile computing and constructive pedagogy has created rich scientific sources that emphasize that education is most stable when it has an active, contextual, and internal impetus. Mobile Game-Based Learning (MGBL) summarizes these commitments by placing curriculum content in interactive, goal-oriented digital environments, where students have independent access from class schedules or stable broadband connections. In the context of the National Strategy of Uzbekistan "Digital Uzbekistan 2030," formalized by Presidential Decree No. UP-6079, the need to transfer these theoretical benefits to local functional software has become a clear political necessity.

Informatics occupies a structurally important place in the secondary education curriculum, but the subject continues to create constant barriers to understanding for many students. Traditional teaching approaches - lecture, static textbook, paper testing - provide students with limited opportunities to manage these uncertainties in real time, monitor their consequences, and receive immediate corrective feedback. A well-developed mobile application can eliminate these conveniences and turn a smartphone, already worn by most high school students in Uzbekistan, into a flexible learning tool [3].

This article provides information on the full cycle and pedagogical assessment of the Android.apk application developed for this purpose. Section 2 presents the available platforms and their limitations. Section 3 presents the algorithmic architecture and database schema. Section 4 presents the experimental assessment and conclusions are drawn and directions for further work are determined.

EduVision: Journal of Innovations in Pedagogy and Educational Advancements
Volume 2, Issue 4, April 2026
brightmindpublishing.com
ISSN (E): 3061-6972

Existing Platforms and Their Limitations

Globally, platforms such as Scratch, Code.org, Lightbot, and SoloLearn have demonstrated the effectiveness of interactive programming training through millions of active users. However, each of them has at least one limitation, which reduces its flexibility for the secondary education system of Uzbekistan. The Code.org and Scratch platforms require a constant internet connection and operate mainly through a web browser, which makes their use unreliable in areas with low or disconnected internet speeds. On the SoloLearn platform, there is no localization of the Uzbek language, and it does not correspond to the sequence of the national curriculum. Although local platforms, such as Maktab.uz and Smart School, have expanded the possibilities of digital education, they do not yet offer subject-oriented, gamified mobile applications in.apk format, adapted for offline operation[11].

Algorithmic Architecture and Database Design

The weighted scoring formula $S = \Sigma (x_i \times w_i)$ for $i = 1$ to N—where $x_i \in \{0, 1\}$ represents correctness and w_i represents the difficulty weight assigned to question i—ensures that deeper cognitive engagement is rewarded proportionally. For the current deployment, weights were set at 3, 5, and 7 points for the three tiers respectively, yielding a maximum session score of 70 points across a 10-question adaptive draw.

The scoring engine implements a linear accumulation algorithm whose correctness is guaranteed at compile time through Kotlin's type system [4]. The algorithm initialises two mutable state variables—score and currentQuestionIndex—and iterates across the question pool until the index reaches or exceeds the pool size. At each iteration the system:

1. displays the current question and its answer variants;
2. captures the learner's selection;
3. compares the selection with the stored correct_option value in the Room database;
4. increments score by w_i if the comparison is true;
5. advances the index.

On termination, the accumulated score, the question-by-question Boolean answer trace, and the epoch timestamp are written atomically to a Results table via a Room @Transaction-annotated DAO method, ensuring that partial writes resulting from battery exhaustion or system interrupts do not corrupt the data [3] .

EduVision: Journal of Innovations in Pedagogy and Educational Advancements
Volume 2, Issue 4, April 2026
brightmindpublishing.com
ISSN (E): 3061-6972

To reduce the impact of repetition and prioritize the elimination of the indicated vulnerabilities, the questionnaire uses a quasi-randomization strategy. The sql predicate primarily filters questions of the target category that the current user has never tried or has previously answered incorrectly, places the result in a random order, and limits the drawing to a parameter N that can be configured. Upon completion of the filtered set, i.e., if the student has correctly answered all the questions at least once, the query returns from the complete category set to an unfiltered random sample, while preserving the novelty and avoiding a dead end in the learning process. Cognitive speed is a measurable measure of computer science competence [2]. Therefore, the application assigns a time limit for each question, which is managed by the Kotlin Corutine running the Dispatchers.Default thread pool program. The elapsed time is represented as a LinearProgressIndicator whose value decreases according to the function $L(t) = L\_max \times (1 - t/T\_total)$, providing a continuous visual cue without interrupting the learner's cognitive engagement with the question content. On timeout ($t \geq T\_total$), the coroutine emits an interruption signal that terminates the current question, records a null response, and advances the index. Time-per-question data stored in the Results table enables post-session diagnostic analysis: a systematic cluster of slow responses on a specific topic category signals the content domain most in need of additional instruction [1].

Presents the Experimental Assessment

The application is fully designed for the Android API in Kotlin and above, providing compatibility with approximately 97% of active Android devices in Uzbekistan, the Ministry of Digital Technologies reports. The user interface layer is built using Google's declarative user interface toolkit Jetpack Compose, which replaces the traditional View/XML paradigm with complex functions whose results are determined from state parameters [5] . This architectural choice yields two pedagogically important advantages: UI is automatically resubmitted when the underlying data changes, providing immediate visual feedback that reinforces the identification of the correct answer; and public administration is clear and testable, reducing the likelihood of displaying inconsistencies in lengthy training sessions. A quasi-experimental pre-test / post-test design with non-equivalent groups was employed at School No. 38, G'allaorol district, Jizzax region, Uzbekistan. Fifty-two Grade 9 students were assigned to either a control group or an experimental group.

EduVision: Journal of Innovations in Pedagogy and Educational Advancements
Volume 2, Issue 4, April 2026
brightmindpublishing.com
ISSN (E): 3061-6972

Assignment was based on existing class rosters rather than random allocation, which is the defining feature of a quasi-experimental design and a recognised limitation addressed in Section 5. Both groups received identical pre-tests to establish baseline equivalence: mean pre-test scores were 62.4% and 63.1% , confirming that between-group differences at baseline were negligible [6].

Over a six-week intervention period, experimental group students used the application during dedicated segments of their Informatics lessons and were encouraged to continue self-directed practice outside class hours. The application's Room database logged all interactions, making it possible to reconstruct each student's response trajectory without the need for separate observational data collection. Control group students followed the school's standard instructional sequence, which included textbook reading, teacher-led explanation, and paper-based exercises and tests.

Conclusion and Future Directions

This paper has presented the complete design, implementation, and evaluation of an offline-capable Android application for teaching Informatics to secondary school students in Uzbekistan. The system integrates a weighted-scoring algorithm, adaptive question selection, time-constrained gamified interaction, a fully normalised Room-SQLite database, and a reactive Jetpack Compose interface into a coherent pedagogical instrument. The quasi-experimental evaluation confirmed that the application produced measurable gains in both content mastery and learner, providing empirical support for the broader claim that MGBL is a viable and scalable approach to Informatics instruction in resource-constrained educational contexts.

The obtained results indicate several directions for future work. First, the quasi-experimental design does not allow for a causal relationship with confidence like a randomly controlled trial; future assessments should require students to be randomly assigned to conditions in multiple schools in different districts. Secondly, the current application is devoted to the content of the subject "Informatics" in the 9th grade; A vertical extension covering the full sequence of grades 5-9, automatic adjustment of the complexity measure increases the practical utility of the system. Thirdly, adding a teacher-oriented analytical dashboard - displaying class-level performance maps from the results table - allows you to configure formative learning based on real-time student data. Fourthly, cross-platform placement using Flutter expands the application's spread to iOS devices. Finally, the integration of the interval repetition

EduVision: Journal of Innovations in Pedagogy and Educational Advancements
Volume 2, Issue 4, April 2026
brightmindpublishing.com
ISSN (E): 3061-6972

table into the algorithm for selecting flexible questions allows the use of the results of cognitive science on optimal intervals of practice for further strengthening long-term memory.

The work reported here contributes both a concrete software artefact—freely deployable in Uzbek secondary schools as an .apk file—and a replicable methodological template for developing curriculum-aligned, offline-capable MGBL applications in other subject areas and national contexts where mobile penetration exceeds broadband infrastructure.

References

1. Crompton, H., & Burke, D. (2023). The use of mobile learning in K–12 education: A systematic review. Computers & Education, 110, 51–63.
2. Deterding, S., Dixon, D., Khaled, R., & Nacke, L. (2011). From game design elements to gamefulness: Defining gamification. Proceedings of the 15th International Academic MindTrek Conference, 9–15.
3. Google Inc. (2024). Android Jetpack Compose documentation. https://developer.android.com/jetpack/compose
4. Kapp, K. M. (2020). The Gamification of Learning and Instruction: Game-based Methods and Strategies for Training and Education (2nd ed.). Pfeiffer/Wiley.
5. Krashen, S., & Mason, B. (2020). Can second language acquisition theory be applied to game-based language learning? CALL Journal, 33(1), 1–14.
6. Prensky, M. (2021). Digital Game-Based Learning (New Edition). Paragon House.
7. Sharples, M. (2022). Mobile Learning: Small Devices, Big Issues. Routledge.
8. Siemens, G. (2005). Connectivism: A learning theory for the digital age. International Journal of Instructional Technology and Distance Learning, 2(1), 3–10.
9. UNESCO. (2023). Policy Guidelines for Mobile Learning (Revised Edition). UNESCO Publishing.
10. Republic of Uzbekistan. (2022). Law on Cybersecurity No. O'RQ-764. Official Gazette.
11. Republic of Uzbekistan. (2020). Presidential Decree PF-6079: Approval of the Digital Uzbekistan 2030 Strategy. Official Gazette.