Using Coding Games to Improve Logical Thinking

Shahistha Tabbasum, European International University, Paris

Abstract

Logical reasoning is a basic mental ability that is necessary for problem-solving, decision-making, and computational thinking. With technology playing a greater part in the education sector, code games have become an efficient teaching tool to develop logical reasoning skills among students. This research investigates how coding games, such as AI-based adaptive platforms, assist in building systematic thinking, pattern identification, and problem solving skills among learners between the ages of 7 and 15. Through the inclusion of both quantitative and qualitative methods of research, such as student evaluation, instructor feedback, this study assesses the efficacy of coding games in cultivating logical thinking. The results of the study bring to light the advantages of game-based learning, especially AI-based coding platforms, in encouraging participation, flexibility, and teamwork.

Summary

Logical thinking is an important skill that impacts problem-solving, decision-making, and computational thinking. As technology becomes more integrated into education, coding games have become a new instructional aid to improve logical thinking among students. This study examines the influence of coding games on logical thinking in students from the age group 7–15. The research seeks to evaluate how learning through gamification in coding improves orderly thinking, pattern recognition, and problem-solving skills.

Coding games, which are interactive and engaging, provide a systematic way of learning computational ideas.The games tend to include problem-solving exercises, algorithm development, and debugging activities, which demand logical thinking from the students. Through the study of the impact of these games in a classroom environment, this study will offer insights into the ability to improve cognitive abilities and logical thinking. Over the past few years, AI-powered coding games have become popular by providing adaptive and differentiated learning experiences. These platforms examine student performance and dynamically alter game difficulty in response, optimizing the level of challenge versus skill acquisition. Intelligent coding games like CodeMonkey, AI Dungeon, and CoSpaces Edu offer immediate feedback and adaptive hints, allowing players to focus on self-led learning and developing problem-solving competence. AI integration also enables students to practice more advanced problem-solving approaches, solidifying logical reasoning in a fully immersive, game-like experience. The value of logical reasoning is not limited to computer programming; it is intrinsic to all academic subjects like math, science and engineering.

Conventional practices tend to centre on theoretical aspects, which may not always appeal to students. Coding games, however are an experiential learning strategy that allows learners to implement concepts in the process.AI-powered coding games also eliminate gaps between theoretical and practical application by presenting real- life situations that involve logical reasoning.

Further, coding games enabled by AI foster cooperation through multiplayer and problem-solving activities, cultivating teamwork and analytical skills. The games provide interactive learning spaces where students can try, fail, and learn from errors without fear of censure. This develops resilience and agility crucial characteristics in computational thinking and logical reasoning. The present research will use qualitative and quantitative approaches, such as student test scores, instructor comments, and performance data analysis, to determine the effectiveness of both conventional coding games and coding games powered by AI in enhancing logical reasoning. The results from this research will add to existing literature on education based on technology and offer a rich understanding of how AI-facilitated learning tools can be used within school curricula to enhance the logical thinking capacity of students. By knowing how coding games that are AI-based affect logical reasoning, teachers can make informed choices regarding the integration of such tools into pedagogical approaches. This study is meant to emphasize the benefits of adaptive coding games in tailoring education, enhancing critical thinking, and equipping students for careers in the future in technology and problem-solving fields.

Literature Review

Several studies have highlighted the importance of coding and computational thinking in schools. Grover and Pea (2013) set the precedent that learning computational thinking improves students' logical reasoning and problem-solving skills. Likewise, Wing (2006) proposed computational thinking as a core skill similar to reading or mathematics, which needs to be learned at a young age. These early studies highlight the need to develop logic-based learning strategies in schools. Wing stresses that just as reading and writing are essential, computational thinking must be a basic skill for all students. Based on this, Grover and Pea (2013) argue that programming is a good method for teaching computational thinking because it requires students to use logic in sequencing, conditionals, loops, and debugging. Coding motivates learners to think step-by-step and sequentially, thus naturally developing the logical thinking skills.

A research by Papert (1980) presented the idea of constructionism through programming using LOGO, illustrating how children's cognitive skills could be enhanced by working on computer-based tasks. Recent research by Resnick et al. (2009) on Scratch, a block-based programming environment, reinforces this perception by illustrating that students are more creative and logical when they learn using gamified environments. Game-based learning has been found to be effective in increasing student motivation, interest, and retention. Papa Stergiou (2009) conducted a study that revealed students who used educational games performed considerably better in computer science-related courses than students who were taught by conventional means. The interactive and engaging characteristics of games are responsible for increased student participation and enhanced learning. In cognitive development, Prensky (2001) posits that computer games promote problem-solving, strategic thinking, and decision-making. Through playing game situations, students are repeatedly forced to analyse choices, plan moves, and predict outcomes—activities that reflect logical thinking processes.

Conversely, there are some researchers who hold that while coding games are stimulating, they do not necessarily offer rich learning results unless adequately scaffolded (Kafai & Burke, 2014). Their own work emphasizes the need for systematic support to guarantee that play translates to enriching educational opportunities. In addition, the inclusion of AI in coding platforms like CoSpaces Edu or Code Combat has had potential in adaptive learning and individualized feedback, which were under investigated in previous research. Even with these developments, there is a deficit in measuring the comprehensive effect of AI-powered coding games on logical reasoning in particular among the 7–15-year-old students. While most studies either examine older students or do not specifically isolate logical reasoning as an immediate outcome, this research seeks to fill this deficit by exploring how coding games, especially AI-powered ones, influence logic development in various age groups and learning environments. This study offers a fresh insight by combining knowledge from traditional computational learning theories with modern AI-based tools in education technology. It contributes to the literature by presenting both qualitative and quantitative supports for how gamified, structured, and AI-based coding education teaches logical reasoning, and possibly provides new frameworks for how such tools can be integrated into regular curriculums.

Research Questions

This research study is centered around three primary research questions that form the guiding light of the study. These questions were developed on the basis of the gaps in current literature and drawn from the practice of an ICT educator dealing with ha wide variety of learners. Each question presents a different angle for viewing various aspects of the research issue.

Question1: How do coding games influence the logical thinking skills of school students?

This research question is at the core of comprehending the cognitive advantages of coding games within primary and secondary education to gain insight into the direct influence of coding games on student’s logical thinking capacity. Logical thinking encompasses the ability to analyse, identify patterns, connect and conclude solutions from evidence. In students aged 7–15 years who are in Piaget's concrete operational to formal operational stages these skills are still in development and can be greatly impacted by the learning environment.

Coding games are particularly well-suited to build these skills because they necessitate students to:

- Decompose tasks (decomposition)
- Identify repeating patterns (pattern recognition)
- Develop step-by-step solutions (algorithmic thinking)
- Apply loops, conditionals, and variables logically
- Debug and revise code (iterative thinking)

These exercises simulate actual problem-solving, which challenges students to reason analytically and sequentially. Static worksheets are unlike coding games, where students receive instantaneous feedback. If a student's answer goes wrong, they are required to go back and change their reasoning and attempt it again. This process of trial-and-error solidifies logical ordering and promotes perseverance. Through my experience as an ICT teacher, I have seen students become more engaged with topics when delivered by interactive coding games. By resolving this query, the research serves to illustrate that coding games are not only entertaining resources but also strong cognitive boosters.

Question 2: How do learning outcomes differ between students engaged with traditional teaching methods and those exposed to coding games?

This is a comparative question and seeks to identify the efficacy of coding games compared to traditional teaching practices. Traditional teaching practices tend to be based on lectures, textbook problems, and direct instruction. Although these methods have their advantages, they might not engage students as much or promote intense cognitive development in logical thinking.
This research seeks to establish if students learning via coding games have improved retention, performance, and problem-solving abilities than students who are taught conventionally. The achievement of the learning process will be assessed using pre-and post-tests, classroom engagement, and student and teacher feedback. Special emphasis will be put on how game-based learning induces intrinsic motivation, creativity, and autonomous learning. The findings will establish whether coding games provide a superior or complementary means of logical skill acquisition and provide useful insights into hybrid learning environments where conventional and game-based practices coexist.

Question 3: What factors influence student engagement and retention when using coding games in educational settings?

Aside from academic achievements, engagement and retention are key markers of successful learning. Engagement is the quality of student's attention, curiosity, interest, and active participation while learning. Retention is a term used for the student to sustain over a long period and recollect or implement the gained abilities in another situation. This question is trying to know the factors (psychological, emotional, and environmental) driving the involvement of students in coding games and maintain the enthusiasm for the longer duration.

Few research studies have described gamification, immediate feedback, adaptive difficulty, and reward as essential participation drivers. From a teacher's point of view, I have found that students are more engaged and motivated if coding games involve aspects of storytelling, peer rivalry, and dynamic feedback. This study will consider how these factors are expressed within classroom environments and how they impact student behaviour and consistency of learning. Student feedback and observation will provide insights into how learning can be enhanced and how these platforms can be made more inclusive and accessible. Knowing these factors will enable educators to craft more effective coding education curricula and construct supportive learning environments that can meet the needs of different learners. It also presents an opportunity for educational game developers to innovate and fill in gaps in existing game-based learning solutions.

Scope of study

The paper particularly aims at students both in upper primary and lower secondary levels, corresponding to a development stage at which reasoning increasingly starts to develop and is crucial for learning, especially in STEM education. It takes into account the views of students, teachers, and parents to gain an in-depth understanding of the efficiency of game-based learning approaches to computational thinking.

This research is based on three fundamental elements/stakeholders:

1. The Learners (Students between 7–15 years):

The research focuses on primary and middle school students (Grades 2–10), the age group most likely to be exposed to fundamental coding principles in a school program or after-school activities. They are selected because at this stage of their development, their logical reasoning skills would be best developed using gamified and interactive resources.

2. The Educators (ICT/Computational Thinking Teachers):

Teachers were chosen as central stakeholders because they are immediate coders of coding initiatives and have pragmatic understanding of learning habits of students, their engagement levels, and issues with learning coding.

3.The Guardians (Student's Parents):

Parents provide home-life insight of students' interest, motivation, concerns about screen times, and felt cognitive progress and assist in validating the data in school settings.

The research was carried out using online questionnaires sent to the above mentioned elements from various educational institutions, mainly those using CBSE, ICSE, IB curriculum. Data were gathered from participants in urban Indian and international school contexts, especially where coding is part of the curriculum or taught as an enrichment subject. The study is limited to formal learning environments and investigates the contribution of coding games—more specifically AI-based and visual environments such as Scratch, Code.org, Tynker, and Codemonkey—to the formation of logical abilities. The platforms were selected for their applicability to the target age range, flexibility to accommodate learner skills, and ability to present real-time problem-solving exercises. The focus is on platforms that integrate interactive storytelling, sequential logic, conditional branching, and debugging practices, which mirror real-world coding principles in a simplified manner that is appropriate for children.

Key data collection was done using structured Google Form questionnaires administered to students and teachers, with classroom observations, logic-based tests, and game performance metrics. The Google Form has questions to test previous exposure to coding games, self-reported improvement in logical thinking, perceived learning advantages, challenges faced, and interest and engagement levels.

Teachers responses are centered on observed changes in behaviour, skill acquisition, integration into the classroom, and opinions regarding the educational worth of the coding platforms. These answers set the boundaries and scope of inquiry into students logical reasoning abilities, learning styles, and general classroom participation. This research is greatly focused on rational reasoning, incorporating abilities such as pattern recognition, step-by-step planning, algorithmic thinking, abstraction. These cognitive characteristics are pinpointed and evaluated based on subjective and objective measures. Nevertheless, the cognitive field might be broad but is purposefully not concerned with non-academic outcomes such as emotional intelligence or social conduct except where it translates directly into interaction or persistence within coding game realms.

In terms of student demographics, both genders and a range of socio-economic categories were included in the study, though no outcomes are distinguished on the basis of these. The major focus continues to be educational and cognitive development, not socio-demographic.

Among the principal limitations recognized in this scope is inconsistency in teacher preparation and skill when employing coding games. While some will easily incorporate these resources into their teaching practices, others will have challenges based on training, time, or curriculum match. Such inconsistencies are likely to affect student's outcomes and interest and are accounted for during analysis.

A second limitation is the use of self-reported data, particularly in the Google Form surveys. While surveys are important in providing insight into student and teacher attitudes, there is some subjectivity involved that may impact accuracy. To offset this, the study complements surveys with direct classroom observation and performance-based testing to triangulate the data. The coding platforms selected also determine the scope of study. Instead of testing experimental or newly created games, the study analyses established and popular platforms that are easily found in most educational systems. These findings are more applicable and easier to reproduce in similar educational environments. However, this decision excludes the possibility of using possibly innovative but less accessible tools that could provide varying results.

In addition, the emphasis on logical reasoning as the primary cognitive ability implies that associates academic gains in mathematics or computer science are only taken into account when they overlap with logic specify abilities. The research does not seek to quantify general academic progress or more general developmental milestones. The scope also involves the identification of major design and pedagogical aspects of coding games that ensure long-term engagement and retention. By examining visual design, levels of difficulty, interactive feedback, gamified reward systems, and collaborative aspects, the research will establish what makes coding games not only effective but also engaging to young learners.

To sum up, the study is based in the following educational concepts:

- *Logical Thinking & Problem-Solving*: The primary cognitive outcomes measured.
- *Game-Based Learning (GBL)*: The pedagogical method being evaluated.
- *Engagement and Retention*: Behavioural aspects linked to learning effectiveness.
- *Comparative Outcomes*: Between coding game users and those taught by traditional methods.

It also considers psychological and constructivist theories by Piaget, Vygotsky, and Papert, as well as the emerging framework of Computational Thinking by Wing (2006).

While it is rich in providing information on engagement, learning outcomes, and practical integration, it is still constrained by institutional, temporal, and methodological considerations. The results of this study will be expected to be relevant in high measure to curriculum designers, educational policymakers, and school-level implementers who wish to take up innovative learning approaches.

Methodology

The research design follows a descriptive survey model, ideal for educational studies that aim to investigate current practices, attitudes, and outcomes. Given the research questions focus on engagement, learning outcomes, and comparative effectiveness between traditional methods and coding games, surveys provide an efficient way to gather large-scale data across different demographics. This design aligns with the study's aim to generalize findings and provide insights for educational stakeholders.

A mixed-methods design was employed, involving both quantitative and qualitative data. This was utilized to give an in-depth view of the research subject. The most common tool utilized was a computer-based online survey, formatted and administered through the use of Google Forms.

Questionnaires covered:

1. *Closed-answer items* (multi-item scales, including multiple choice) to put perception, activity level, and improvement in logic to number.
2. *Open-answer* brief answer items that provided space to expatiate on occurrences and qualitative nuance to quantitative measures.

Each of the three surveys was customized slightly for the respective stakeholder groups to guarantee relevance and understandability in questioning. For instance, teachers were questioned on classroom practice and assessment with their observations regarding learning outcomes and classroom application, while parents gave insights into changes in their children's behaviour and cognition at home. Whereas students provided feedback based on their own experiences with coding games such as Scratch, Blockly, and Tynker.

The survey was trialled with a pilot sample to simplify the wording and format prior to mass distribution. Data were gathered over four weeks, and responding was voluntary and anonymous in order to stimulate genuine feedback. The Quantitative data were processed through descriptive statistical procedures such as percentage distribution and mean scoring to establish trends between groups. Microsoft Excel was utilized in data organization and producing visual charts for easier interpretation. Comparative analysis was done to compare differences in learning outcomes among students who utilized coding games and students taught using conventional methods. This approach is not only aiding the current research's objectives but also providing a reproducible framework for future teachers and researchers. With coding becoming increasingly introduced into school curricula worldwide, this approach (particularly the structured survey instruments) can be reused or retooled in the same type of educational contexts to track the effectiveness of coding games in developing logical and computational skills.

Ethical Considerations
Members were made aware of the research aim and provided informed consent prior to involvement. Personal identifiers were not gathered, and all answers remained confidential. Participants were assured by the study that they had the right to withdraw at any time.

Results

The findings of the study draw a multi-angle picture of coding games impacts on logical thought and associated capacities among students as well as how teachers and parents view them. The findings were structured and framed from data accrued through systematic questionnaires and observation, as shown in the method section.

Feedback was gathered across 150; 70 was from students and 40 both from teachers and parents.

Student Feedback

Most of the students shared a clear agreement on the influence of coding games on their logical thinking capacity. As illustrated by the graph in figure1, 60% of students strongly agreed that their logical thinking was enhanced because of coding games, while 30% agreed and merely 10% disagreed or were neutral.

This shows that 90% of the students saw the positive effect, confirming the implementation of game-based platforms as valuable tools for cognition.

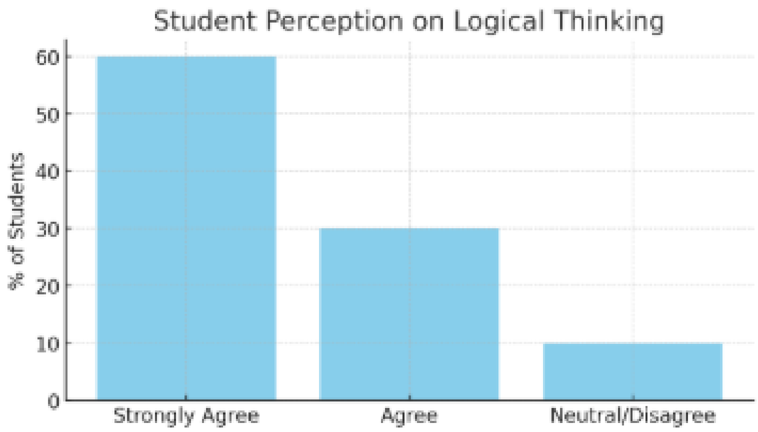

Figure1

In addition to logical enhancement, students also said they were more motivated to study topics that traditionally appeared to be abstract or complex. Several students mentioned that the gamified design assisted in anxiety reduction and confidence building in problem-solving. The coding games fostered experimentation, instant feedback, and iterative learning—all of which are key to computational thinking development.

Students also preferred collaborative play in which peer-to-peer interaction was enabled. This was especially noticeable in platforms used by classrooms such as Code.org and Tynker, which from time to time include cooperative challenges or class leader boards. Further, Student interviews of sampled pupils noted greater freedom in learning with some pursuing extracurricular projects or levels beyond class tasks. Lastly, students enjoyed the visual and interactive aspects of platforms such as Scratch, particularly those who considered themselves visual or kinesthetic learners. This kind of engagement indicated a high likelihood for differentiated instruction using coding games to address varied learning styles in inclusive classrooms. Along with logical ability enhancement, liking for particular coding platforms was also explored. The pie chart in figure 2 shows Scratch as the most popular platform (60%), followed by Code.org (25%), Tynker (10%), and Blockly (5%). These lopsided preferences indicate Scratch's easy-to-use and interactive nature, particularly appealing to visual learners at an early stage, as the core platform to be used in future curriculums.

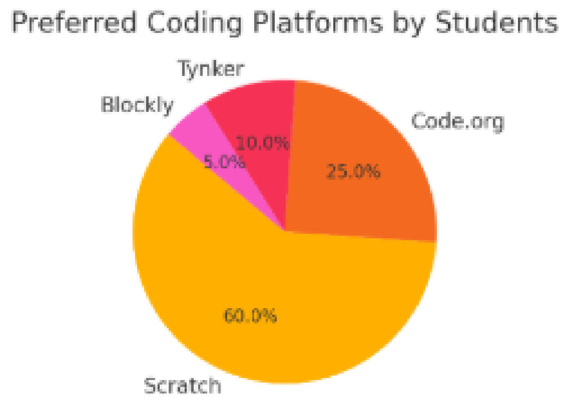

Figure 2

Teacher Observations

Assessment by teachers also supported student evaluation. As illustrated in the chart in figure 3, 75% of teachers reported seeing significant improvement in student's logical ability, and 25% did not see noticeable change. Teachers found that students displayed better problem-solving strategies like decomposition, sequencing, and debugging, which are essential building blocks of logical reasoning. These changes were especially seen during group activities and individual assignments with an algorithmic thought requirement. In addition, instructors indicated that students exhibited longer attention spans and more intense engagement during coding game sessions than in regular lessons. They credited this to the goal-directed, interactive quality of coding sites, which promoted concentrated effort through instant feedback and visible progress.

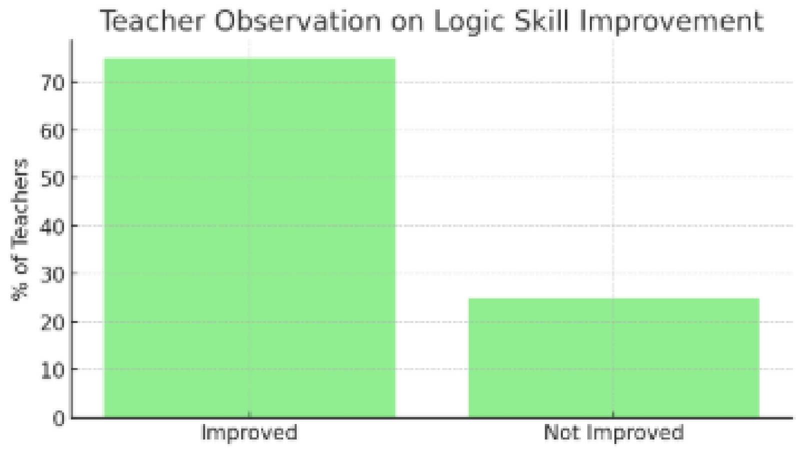

Figure 3

For most teachers, coding sessions had become peak classroom energy and focus moments, with even usually restless students being enthusiastic and persistent about finishing challenges. A number of teachers also noted the simplicity with which coding games facilitated differentiated instruction. Struggling students were able to complete levels at their own pace, whereas advanced students solved challenging problems independently, creating a supportive learning environment. These findings affirm that guided use of coding games can assist teacher-directed instruction, encourage active learning, and reinforce fundamental thinking skills while meeting diverse learning profiles.

Parental Insights

Parent responses introduced yet another helpful perspective to the study. Parental responses named Logical Thinking (80%) as the greatest return in perceived gain in coding games, followed closely by Creativity (65%), Digital Literacy (60%), and Social Skills (30%) according to the horizontal bar chart in figure 4. This classification reveals that parents are very concerned about the intellect and creative developments from coding but regard relatively smaller effect on the inter-personal development.

In open-ended answers, a few parents noted that their children started to develop a greater interest in puzzles, strategy games, and logic schoolwork after working with coding websites. They noticed their children showing greater patience and perseverance in solving school issues, something they credited to the trial-and-error process of coding games.

A number of parents also valued the fact that coding games gave way to constructive substitute for normal screen time. Though screen overexposure was a concern, many stated that learning games presented a substantive mix of fun and education. Some even observed increased family interaction as kids worked to explain code blocks or help their siblings learn platforms like Scratch.

Some parents highlighted the cultivation of soft skills like communication, as kids started sharing their logic and game-concept ideas more freely. While social skill development was scored lower than other areas, anecdotal feedback indicated indirect gains in peer collaboration and self-expression through design-based games. Regarding support for applying coding games in education, 55% of parents supported it, while 40% were concerned, as evident from the bar chart in figure 5. Concerns primarily revolved around screen time and distraction. Yet, the majority's support indicates a positive parental attitude toward incorporating digital tools in learning when appropriately guided.

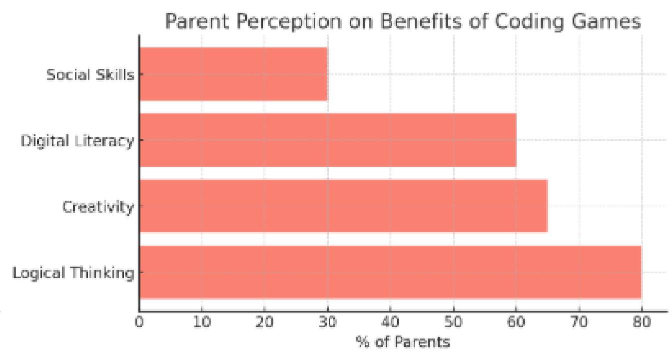

Figure 4

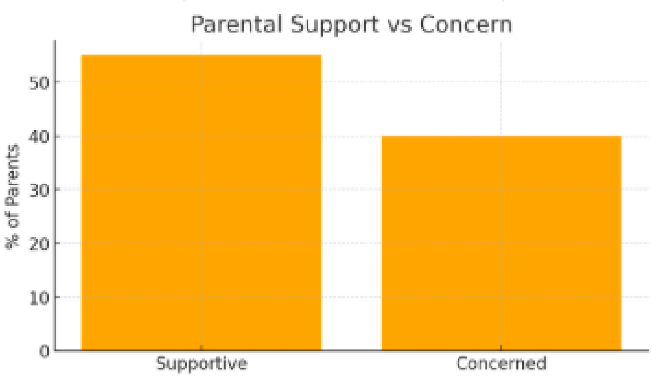

Figure 5

Conclusion

The results of this research strongly support the hypothesis that coding games have a positive impact on the formation of logical thinking in students between ages 7 and 15. With analysis of perspectives from students, teachers, and parents, the research detects striking patterns consistent with both theoretical backgrounds and classroom practice.

Enhancement of Logical Thinking through Coding Games

Coding games have emerged as a successful vehicle for developing logical reasoning skill. Students exhibited tangible gains in sequencing, problem breakdown, and pattern recognition, key aspects of computational thinking. The interactive mode of applications such as Scratch, Code.org, and Tynker provided an immersive setting theoretical concepts were converted into visual, tangible learning opportunities. These applications provided the option for students to practice and receive immediate feedback, thereby supporting trial-and-error learning as well as determination. The large number of students (90%) who admit to positive effects on logical thinking confirms that coding games are not only additional tools but may be used as core pedagogical resources for the formation of basic thinking abilities.

Student Engagement and Intrinsic Motivation

In addition to cognitive improvement, student feedback indicated increased interest and motivation towards learning through gamified processes. Coding games dissolved conventional barriers of fear or lack of interest usually linked with programming or technical topics. The gamified presentation created a feeling of autonomy, discovery, and accomplishment, resulting in increased engagement with the subject. The attractiveness of shared features like team challenges and classroom leader boards also increased motivation and social learning. This concurs with Vygotsky's social constructivist theory where peer interaction promotes cognitive development.

Teacher Observations: Reinforcing Classroom Learning

Feedback from the teacher highlighted how coding games supported differentiated instruction and the ease of addressing students at their unique levels of learning. Teachers noticed improvement in logical reasoning, maintained attention and heightened engagement—particularly for students who often find other methods challenging. Additionally, the games offered a structured but open-ended environment in which teachers could give assignments, track progress, and intervene appropriately. This implies that coding games not only facilitate students' learning but also make teaching more efficient and delivery of lessons more effective. The research reinforces Papert's constructionism, where learning is understood as constructing knowledge by means of creation. Coding games enable learners to construct their own projects, enhancing a sense of ownership in learning and increased interest in logic-based challenges.

Parental Perspectives: Cognitive and Behavioural Shifts

Parental observations added a valuable layer to the results, displaying measurable changes in behavior such as higher engagement in puzzles, strategic thinking, and logical problem-solving outside of school. Most parents reported their children showing greater resilience and better digital literacy, aligning with the abilities fostered in game-based learning environments. Interestingly, although there were still some concerns about screen time, the majority of parents viewed coding games as a good use of technology. This implies a change in attitude from viewing technology as a distraction to a useful learning tool, as long as its use is guided and intentional.

Alignment with Educational Theories and Prior Research

The results of the research strongly agree with current education theories:

• Wing's Computational Thinking framework is embodied in students' increased ability to think algorithmically.
• Prensky's Game-Based Learning theory is confirmed by greater student participation and enjoyment.
• Piaget and Vygotsky's Constructivism emphasizes the significance of experiential, student-centered approaches conducive to internalization through experience and social interaction. In addition, results mirrir trends across worldwide literature in identifying coding as a means toward critical thinking and readiness for the future in educational contexts.

Implications for Educational Stakeholders

The findings of this research have some implications:

• *For Teachers*: ICT and STEM curricula must include coding games not only as add-ons but as integral parts of the curriculum.
• *For Curriculum Designers:* Projects must contain project-based tests through coding games to evaluate logical thinking and problem-solving.
• *For Policymakers*: Educational technology infrastructure investment is necessary to achieve equitable access to coding platforms.
• *For Parents*: Training and awareness can assist in facilitating effective in-home use of coding sites, ensuring learning becomes a collaborative effort.

Limitations Revisited

Although the results are promising, some limitations need to be noted. The short-term duration of the study does not allow for measurement of long-term cognitive gains. Responses were also restricted to English-speaking participants, and device and internet access were assumed. Future studies should seek broader, more inclusive participation and longitudinal studies to assess retention and application of skills over time.

References

1. Grover, S., & Pea, R. (2013). *Computational thinking in K–12: A review of the state of the field*. Educational Researcher, 42(1), 38–43. https://doi.org/10.3102/0013189X12463051
2. Wing, J. M. (2006). *Computational thinking*. Communications of the ACM, 49(3), 33–35. https://doi.org/10.1145/1118178.1118215
3. Papert, S. (1980). *Mind storms: Children, computers, and powerful ideas*. Basic Books.
4. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). *Scratch: Programming for all*. Communications of the ACM, 52(11), 60–67. https://doi.org/10.1145/1592761.1592779
5. Prensky, M. (2001). *Digital game-based learning*. McGraw-Hill.
6. Stergiou, P. (2009). *The impact of educational games on student learning in computer science*. International Journal of Education and Information Technologies, 3(1), 37–43.

7. Kafai, Y. B., & Burke, Q. (2014). *Connected code: Why children need to learn programming*. MIT Press.
8. Piaget, J. (1970). *Science of education and the psychology of the child*. Viking Press.
9. Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.
10. Code.org. *Learn computer science*. https://code.org
11. Scratch Foundation. *Scratch - Imagine, Program, Share*. https://scratch.mit.edu
12. Tynker. *Coding for Kids*. https://www.tynker.com
13. CoSpaces Edu. *Make AR & VR in the classroom*. https://cospaces.io/edu/
14. CodeMonkey. *Coding for kids made fun*. https://www.codemonkey.com