

Design and evaluation of a digital learning environment for computer architecture education: A mixed methods case study

Janvier Ngnoulaye*

Laboratory of Computer Science and Educational Technologies, Higher Teacher Training College, University of Yaoundé I, Cameroon,

Email: janvier.ngnoulaye@univ-yaounde1.cm

A Research Article

*Corresponding author:

janvier.ngnoulaye@univ-yaounde1.cm

ORCID ID:

<https://orcid.org/0000-0002-4975-8504>

Received: 08/03/2026

Revised: 02/04/2026

Accepted: 05/04/2026

Published: 03/05/2026

Citation:

Ngnoulaye, J. (2026). Design and evaluation of a digital learning environment for computer architecture education: A mixed methods case study. *Journal of Teaching and Education for Scholars*, 3(1), 1–16. <https://doi.org/10.59065/jotes.v3i1.274>



Copyright:

© 2026 by the author(s).

This is an open-access article: the use, distribution, and reproduction of published material are strictly regulated under the [Creative Commons Attribution 4.0 International License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).

Abstract

Teaching computer science to secondary school students in Cameroon presents significant pedagogical challenges due to insufficient teaching resources and a shortage of qualified teachers. This study presents the design, development, and evaluation of an educational software environment - *SimuArch* - specifically adapted for teaching computer architecture in class (Year 3 of lower secondary school) in Cameroon. Using a mixed-methods research design, this study combines a quantitative quasi-experimental approach with qualitative inquiry. An experimental evaluation involving 115 students - divided into an experimental group ($n = 58$) using *SimuArch* and a control group ($n = 57$) following standard instruction - was conducted over an 8-week period. Quantitative data were analysed using repeated-measures analysis of variance (ANOVA) and independent-samples *t*-tests; qualitative data were subjected to inductive thematic analysis. Results showed a statistically significant Group \times Time interaction ($F(1,113) = 24.8, p < 0.001, \eta^2 = 0.18$; Cohen's $d = 1.21$), with an average performance gain of 75% in the experimental group compared to 35% in the control group. Qualitative analysis revealed a marked increase in student motivation and engagement. These findings suggest that integrating a contextually appropriate digital learning environment can significantly improve the teaching and learning of computer architecture at the secondary level in Cameroon, with potential implications for other sub-Saharan African educational contexts.

Keywords: *Computer Architecture; Computer Science Education; Educational Technology; Experimental Evaluation; Secondary Education*

1. Introduction

Computer architecture is a foundational area of computer science, essential for understanding how modern computing systems function (Patterson & Hennessy, 2014). However, teaching it at the secondary level presents particular challenges, notably due to the complexity and abstract nature of the concepts involved in how a computer operates (Guzdial, 2008). In Cameroon, as in many developing countries, computer science education at the secondary level suffers from a lack of suitable teaching tools and an insufficient number of trained teachers (UNESCO, 2021; Warioba et al., 2022). Field observations in Yaoundé schools confirm this: introductory computer architecture courses are frequently delivered with no practical activity beyond drawing component diagrams on a chalkboard.

Research in computer science education consistently highlights the importance of visualization and simulation tools in learning abstract concepts (Ben-Ari & Peled, 2015; Chen & Tsai, 2021). Constructivist and constructionist learning theories (Piaget, 1954; Papert, 1980; Noss & Hoyles, 2017) emphasize that students learn most effectively when they actively build and manipulate objects in digital environments. More recent scholarship underlines the importance of designing such environments in culturally and infrastructurally responsive ways (Agbo et al., 2023; Sanusi & Tshukudu, 2022). In the specific context of teaching computer architecture at the secondary level, several studies have demonstrated the effectiveness of tutorials, simulators, and visual programming environments (Yehezkel et al., 2001; Zhang & Nouri, 2019). Nevertheless, most available educational materials have been developed for Western learning contexts and do not account for the specificities of educational systems in developing countries such as Cameroon, where material, internet access, cultural, and linguistic constraints must all be considered (Agyei & Voogt, 2011; Kafyulilo et al., 2015).

Despite a growing body of research on digital learning environments for computer science, several important gaps remain. First, the vast majority of simulation-based tools have been evaluated in high-resource contexts (Mayer, 2019; Sweller et al., 2019), and very few studies address their effectiveness in sub-Saharan African secondary school settings (Bice et al., 2020; Mwila, 2018). Second, prior work in Cameroon and comparable francophone contexts has primarily relied on descriptive or survey-based methodologies, without rigorous pre/post quasi-experimental controls (Ngnoulaye, 2026). Third, most available tools require high-specification hardware or persistent internet connectivity, rendering them impractical for schools in Yaoundé and similar urban contexts in Central Africa (UNESCO, 2019). This study addresses these gaps by presenting and rigorously evaluating a locally designed digital learning environment.

Therefore, this study is guided by the following research questions:

1. RQ1: Does the use of SimuArch produce statistically significant improvements in students' learning performance in computer architecture compared to standard instruction?
2. RQ2: What effect does SimuArch have on students' intrinsic motivation and behavioral engagement in computer architecture learning?
3. RQ3: How do students and teachers perceive the pedagogical utility and usability of SimuArch in the Cameroonian secondary school context?
4. RQ4: Which specific features of SimuArch contribute most to observed learning gains?

The main objective is to design and evaluate SimuArch - a software environment for learning computer architecture in class. The specific objectives are: (1) to design a software environment adapted to learning computer architecture in the Cameroonian secondary school context; (2) to evaluate its pedagogical effectiveness through a mixed-methods quasi-experimental study; and (3) to analyze the factors of acceptance and use by students and teachers.

1.1 Challenges of Teaching Computer Architecture at the Secondary Level

Teaching computer architecture at the secondary level faces several challenges, particularly in contexts where electricity penetration rates remain low and computer equipment is scarce. Hambrusch et al. (2013) highlight the difficulty students encounter when conceptualizing levels of abstraction, from logic circuits to software programs. This difficulty is even more pronounced in settings with limited access to computer equipment (UNESCO, 2019; Ifinedo et al., 2020). Research shows that secondary school students tend to develop persistent misconceptions about how computers operate (Armoni, 2013). These misconceptions, if left uncorrected, can constitute lasting barriers to broader computer science learning (Kaasboll, 2004). In the African context specifically, Agbo et al. (2022) found that the combination of teacher knowledge gaps and equipment shortfalls made conceptual errors among secondary learners particularly persistent in computer architecture topics.

1.2 Theoretical Framework: Constructivism, Guided Discovery, and Multimedia Learning

The theoretical underpinnings of this study integrate three complementary frameworks. First, constructivism (Piaget, 1954; Vygotsky, 1978) and constructionism (Papert, 1980; Noss & Hoyles, 2017) posit that learners actively build conceptual understanding through interaction and manipulation. Vygotsky's (1978) concept of the zone of proximal development further underlines the importance of scaffolded digital tools that bridge the gap between current and potential learning. Second, Bruner's (1960) guided discovery model advocates for structured cognitive progression, preserving room for personal exploration - a principle directly embedded in SimuArch's module sequencing. Third, Mayer's (2009, 2019) cognitive theory of multimedia learning provides a framework for understanding why the combination of dynamic visual representations and interactive exercises facilitates the acquisition of abstract architectural concepts. The application of immediate, corrective feedback (Hattie & Timperley, 2007) is a central design element of SimuArch's assessment module. Together, these three dimensions form the theoretical backbone of SimuArch's pedagogical design.

1.3 Pedagogical Approaches and Technological Tools for Computer Architecture

Several approaches have been developed to facilitate the teaching of computer architecture. Simulation tools allow students to visualize component operation in real time (Wolffe et al., 2002; Yehezkel et al., 2001). Visual programming environments facilitate the understanding of abstract concepts (Bau et al., 2017). More recently, Chen and Tsai (2021) conducted a comprehensive meta-analysis of simulation-based learning in computer science across 78 experimental studies (2010-2020), finding an overall positive effect size ($d = 0.68$, 95% CI [0.54, 0.82]), with larger effects observed when simulations were accompanied by guided instruction and structured feedback. Zhang and Nouri (2019) similarly found strong positive effects for constructionist digital environments in K-9 computer science contexts. However, these tools often require contextual adaptations to be used effectively across different cultural and educational settings (Malmi et al., 2010; Sanusi & Tshukudu, 2022).

1.4 Motivation and Engagement in Technology-Enhanced Learning

Deci and Ryan's (2000) self-determination theory (SDT) provides a key lens for understanding learner motivation. SDT posits that well-designed learning environments satisfy fundamental psychological needs for autonomy, competence, and relatedness. Research grounded in SDT (Deci & Ryan, 2000) demonstrates that interactive digital environments can significantly enhance perceived competence and relatedness in secondary computer science learners. Reeve (2016) argues that autonomy-supportive technologies promote deeper engagement and intrinsic motivation. Kapur's (2016) work on productive failure further suggests that well-structured digital simulations can promote deeper learning through iterative trial-and-error, a design principle embedded in SimuArch's virtual assembly module.

1.5 Educational Technology in Sub-Saharan Africa and Cameroon

Research on ICT integration in sub-Saharan African secondary schools has grown substantially since 2015. Warioba et al. (2022) identified persistent barriers to ICT uptake in Francophone West Africa, including infrastructure deficits, inadequate teacher training, and curriculum misalignment. Kafyulilo et al. (2015) found that Tanzanian secondary teachers held positive attitudes toward ICT but lacked practical integration skills, underscoring the importance of sustained professional development. In Cameroon specifically, Ngnoulaye (2020) conducted a systematic review of digital learning initiatives in sub-Saharan Africa ($n = 120$ studies), concluding that locally developed tools showed higher adoption rates and contextual fit than externally imported platforms. Agbo et al. (2023) demonstrated the effectiveness of low-resource digital environments designed through participatory methods in the Democratic Republic of Congo - findings closely mirrored in the current study's design methodology. Warschauer and Matuchniak (2010) further argue that equity of outcomes in technology-enhanced learning requires attention not only to access but to the cultural and linguistic appropriateness of the tools deployed. The present study is directly situated within this evolving body of literature.

1.6 The Computer Architecture Lesson in class (Cameroon)

The content of the computer architecture lesson for class, as outlined in the official Cameroonian Computer Science curriculum (MINEDUB, 2014), covers the basics of Von Neumann architecture. This includes an introduction to the essential components of a computer - the central processing unit (CPU), memory units, input/output devices, and the data/address bus. The curriculum adopts the Competency-Based Approach (CBA), prioritizing active learning and student-driven knowledge construction. However, material constraints and a shortage of trained teachers limit the effectiveness of this approach in practice. Existing evaluations of digital learning tools in Cameroonian high schools confirm a significant gap between curriculum goals and realized learning outcomes (Ngnoulaye, 2026), providing the direct empirical motivation for SimuArch's development.

2. Methods

2.1 Research Design and Overall Approach

A mixed-methods quasi-experimental design was adopted, combining quantitative experimental analysis with qualitative inquiry (Shadish et al., 2002). The quantitative strand used a pre/post parallel-

group design to evaluate learning gains, motivation, and engagement. The qualitative strand employed semi-structured interviews and structured classroom observations to provide explanatory depth regarding the mechanisms underlying quantitative outcomes. Qualitative findings were used to triangulate and interpret quantitative results. The use of a quasi-experimental rather than fully randomized design reflects practical constraints in the school context: students were pre-assigned to intact classes, and random individual assignment was not feasible. The study was conducted between March and April 2025 at the Lycée de Nkoleton in Yaoundé.

2.2 Software Environment Development

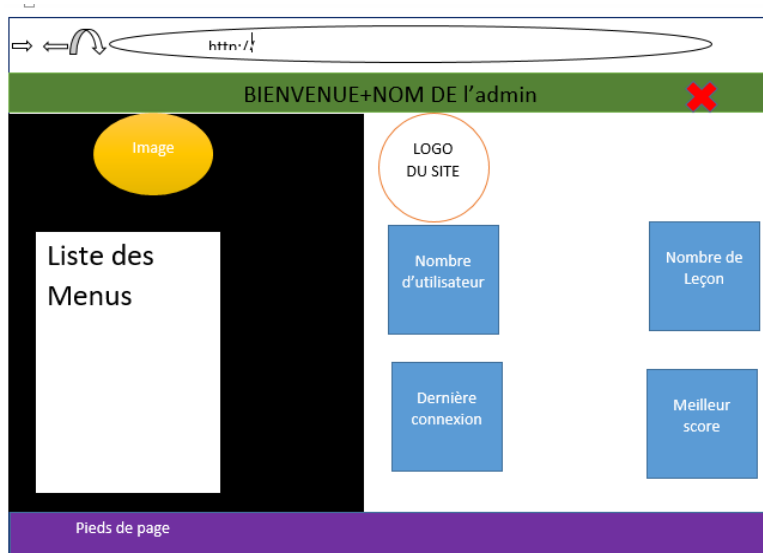
In the preliminary phase, quantitative surveys ($n = 45$ students) and qualitative semi-structured interviews ($n = 15$ teachers) were administered to gather information on the difficulties students encounter when learning computer architecture. Based on these findings, the XP (Extreme Programming) software engineering methodology was combined with the ADDIE (Analysis, Design, Development, Implementation, Evaluation) instructional design framework to design and develop SimuArch - an interactive tutorial for interconnecting basic computer components. SimuArch was built around three main modules:

1. Simulation Module: allows real-time visualization of the operation of core components - processor, memory, and bus.
2. Virtual Assembly Module: offers students the ability to construct and configure computer architectures interactively, incorporating Kapur's (2016) productive failure principle.
3. Assessment Module: provides interactive exercises and adaptive quizzes with immediate corrective feedback to consolidate learning (Hattie & Timperley, 2007).

The user interface was developed in French with an option to switch to English, reflecting Cameroon's official bilingualism. The software was designed to run on computers with minimum specifications (1 GB RAM, 1 GHz processor) to accommodate local hardware limitations.

Figure 1

Mockup of the SimuArch Homepage Showing the Three Main Navigation Modules



2.3 Participants and Group Assignment

The study involved 120 students in the final year of lower secondary school (*3ème / Year 9*) at the Lycée de Nkoleton in Yaoundé. Students were not randomly assigned; instead, existing intact classes were allocated to experimental ($n = 60$) and control ($n = 60$) conditions - a quasi-experimental design consistent with Shadish et al. (2002). To ensure baseline comparability, the two groups were matched on age, gender distribution, and academic performance in previous years (*6ème and 4ème / Years 7 and*

8). No statistically significant baseline differences were detected between groups on any control variable (all $p > 0.05$), confirming initial equivalence. Five attrition cases (4.2%) were recorded, yielding a final analytical sample of 115 students (Experimental: $n = 58$; Control: $n = 57$).

2.4 Instruments and Reliability

Three data collection instruments were used. (1) Pre/Post Achievement Tests: 20-item criterion-referenced tests assessing knowledge of processor operation, memory organization, and Von Neumann architecture (20-point scale). Internal consistency: Cronbach's $\alpha = 0.81$ (pre-test) and $\alpha = 0.84$ (post-test). (2) Motivation Scale: an adapted version of the Eccles and Wigfield (2002) scale (15 items, 5-point Likert), measuring interest in content, perceived value, and self-efficacy. Cronbach's $\alpha = 0.79$. (3) Behavioral Observation Grid: a structured observation protocol assessing on-task time, question frequency, and peer collaboration (inter-rater reliability $\kappa = 0.82$). Additionally, TAM-based items (Davis, 1989) measured perceived usefulness ($\alpha = 0.80$) and ease of use ($\alpha = 0.76$). All instruments were validated by a panel of three expert computer science educators and pre-tested on a separate pilot sample of 30 students from a non-participating school.

2.5 Procedure

- a) Week 1: Administration of pre-tests and baseline motivation questionnaires.
- b) Weeks 2–7: Intervention phase (2 contact hours per week). The experimental group received lessons with SimuArch; the control group received standard lessons following the national curriculum without digital simulation.
- c) Week 8: Administration of post-tests, final questionnaires, and semi-structured interviews. Teachers assigned to the experimental group received 6 hours of structured training in SimuArch use before the intervention began.

2.6 Data Analysis

Quantitative analyses were performed using SPSS v28.0. Independent-samples t -tests compared group means at post-test. Repeated-measures ANOVA examined performance change over time, with the Group \times Time interaction as the key indicator of differential learning progress. Effect sizes were calculated using Cohen's d (for t -test comparisons) and partial eta-squared (η^2) for ANOVA. Categorical variables were analysed using chi-square tests. All tests were interpreted at $\alpha = 0.05$. Qualitative data from semi-structured interviews and open-ended items were subjected to five-phase inductive thematic analysis: (1) data familiarization and transcription, (2) initial code generation, (3) theme construction, (4) theme review and validation by a second coder (inter-coder agreement = 88%), and (5) final theme definition and integration with quantitative results.

2.7 Ethical Considerations

The research was conducted in accordance with standard ethical guidelines for educational research. Informed consent was obtained from the school administration, teachers, and parents or legal guardians of all participating students. Participation was voluntary and students were free to withdraw at any time without penalty. All data were anonymized prior to analysis. No personal identifying information was retained in the research database.

3. Results and Discussion

3.1 Results

3.1.1 Sample Characteristics

The final analytical sample comprised 115 students (Experimental: $n = 58$; Control: $n = 57$). The mean age was 14.2 years ($SD = 1.1$); 52% were male. No statistically significant differences were found between groups on any baseline control variable (all $p > 0.05$), confirming initial equivalence.

3.1.2 Learning Performance (RQ1)

3.1.2.1 Pre-test and Post-test Scores

Table 1 presents pre- and post-test mean scores (out of 20) for both groups. A statistically significant Group \times Time interaction was observed ($F(1,113) = 24.8, p < 0.001, \eta^2 = 0.18$), representing a large effect size. The experimental group's Cohen's $d = 1.21$ indicates a large practical effect. The control group's $d = 0.62$ indicates a moderate effect, consistent with learning from standard instruction alone.

Table 1

Evolution of Learning Performance Scores (Mean \pm SD, out of 20)

Group	Pre-test (M \pm SD)	Post-test (M \pm SD)	Gain	p-value / Effect Size
Experimental (n = 58)	8.4 \pm 2.1	14.7 \pm 2.3	+6.3 (+75%)	p < 0.001; d = 1.21
Control (n = 57)	8.6 \pm 2.0	11.6 \pm 2.5	+3.0 (+35%)	p < 0.001; d = 0.62

Note. M = Mean; SD = Standard Deviation; d = Cohen's d; Repeated-measures ANOVA Group \times Time: $F(1,113) = 24.8, p < 0.001, \eta^2 = 0.18$. Pre-test group equivalence: $t(118) = 0.52, p = 0.60$.

3.1.2.2 Conceptual Analysis by Sub-Topic

Table 2 presents differential gains by curriculum sub-topic, confirming that the experimental advantage was consistent across all three major conceptual domains.

Table 2

Learning Gains by Curriculum Sub-Topic (Post-test vs. Pre-test)

Sub-Topic	Experimental Gain (%)	Control Gain (%)	Difference (%)	p-value
Processor operation	+35%	+18%	+17%	< 0.001
Memory organization	+28%	+12%	+16%	< 0.001
Von Neumann architecture	+31%	+15%	+16%	< 0.001

Note. All between-group differences are statistically significant at $p < 0.001$. Percentage gain is calculated relative to maximum score.

3.1.3 Motivation and Engagement (RQ2)

3.1.3.1 Intrinsic Motivation

Table 3 presents post-intervention motivation scale results. Students in the experimental group reported significantly higher scores on all three motivation dimensions (interest in content, perceived value, and self-efficacy) compared to the control group, with large effect sizes (Cohen's d ranging from 0.85 to 1.06).

Table 3

Motivation Scale and TAM Results by Group (Post-Intervention, Scale 1–5)

Dimension	Experimental (M \pm SD)	Control (M \pm SD)	t(113)	p-value / Cohen's d
Interest in content	4.2 \pm 0.8	3.4 \pm 0.9	5.2	< 0.001 / d = 0.94
Perceived value	4.0 \pm 0.7	3.2 \pm 0.8	5.8	< 0.001 / d = 1.06
Self-efficacy	3.8 \pm 0.9	3.1 \pm 0.7	4.6	< 0.001 / d = 0.85
Intention to reuse	4.0 \pm 0.8	3.0 \pm 0.9	6.1	< 0.001 / d = 1.17

Note. M = Mean; SD = Standard Deviation; d = Cohen's d. All comparisons statistically significant at $\alpha = 0.05$.

3.1.3.2 Behavioral Engagement

Classroom observations revealed significantly greater active participation among experimental group students. Effective on-task time reached 78% in the experimental group vs. 62% in the control group ($p < 0.01$). The average number of questions asked per session was significantly higher in the experimental group (3.4 ± 1.2) than in the control group (1.8 ± 0.9 , $p < 0.001$). Peer collaboration episodes were observed in 67% of experimental sessions compared to 34% in control sessions ($p < 0.001$, $\phi = 0.34$).

Figure 2

Three Behavioral Engagement Metrics from Results Section: A, B and C

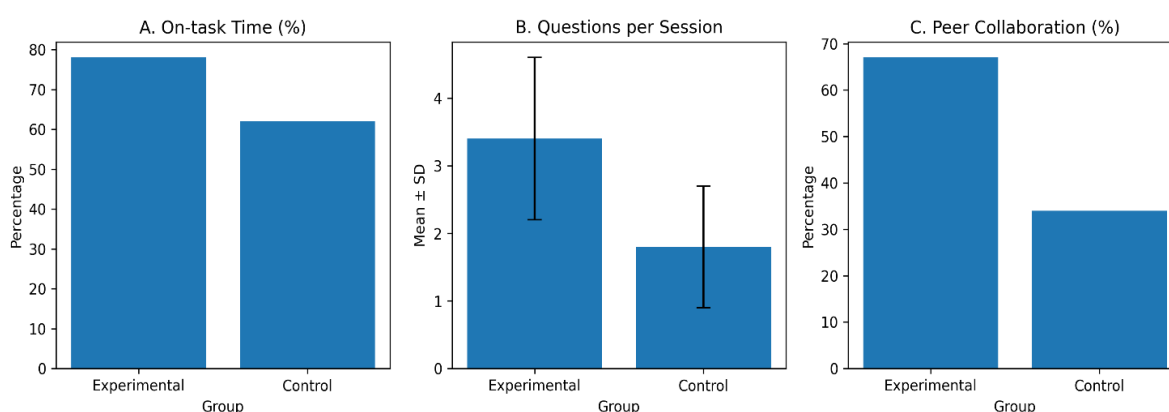


Figure A shows effective on-task time as a grouped bar chart - the 16-percentage-point gap (78% vs. 62%) is immediately visible. Figure B shows mean questions asked per session with the SD range indicated by scatter markers above each bar (3.4 ± 1.2 vs. 1.8 ± 0.9). Figure C shows the proportion of sessions in which peer collaboration was observed (67% vs. 34%).

Each figure above: A, B and C carry its significance indicator ($p < 0.01$ for on-task time; $p < 0.001$ for the other two), and Figure C also notes $\phi = 0.34$. The blue color coding is consistent across all three charts and matches the experimental/control distinction used in the article tables.

3.1.4 Technology Acceptance (RQ3)

Perceived usefulness scored 4.1 ± 0.6 out of 5, ease of use scored 3.8 ± 0.7 , and intention to reuse scored 4.0 ± 0.8 . A significant positive correlation was observed between perceived usefulness and intention to continue using the tool ($r = 0.67$, $p < 0.001$), confirming that utility perceptions drive adoption intent (Davis, 1989).

3.1.5 Qualitative Findings (RQ3-RQ4)

Thematic analysis of 24 semi-structured interviews (20 students, 4 teachers) identified four major recurring themes. For each theme, findings are presented across three components: a description of the finding, supporting evidence drawn from participant interview transcripts, and a theoretically grounded interpretation linked to the study's quantitative results.

3.1.5.1 Facilitation of Conceptual Understanding

This theme was the most frequently coded across the student interviews, emerging in 17 of the 20 student transcripts (85%). Participants consistently described a transition from passive reception of abstract information toward active, visual understanding of how computer components function. Students reported that the Simulation Module's real-time animations of CPU operations, memory read/write cycles, and data bus transfers made previously incomprehensible content directly perceivable. Specifically, the visualization of the fetch–decode–execute cycle was cited most frequently as the feature that resolved long-standing misconceptions about processor operation. This finding aligns with the +35% experimental gain in the processor operation sub-topic (Table 2), the largest gain recorded across all three curriculum domains.

Multiple student voices converged on the same experience of cognitive transformation through visualization. Representative excerpts include:

“With the software, I can see how the processor really works. Before, it was just words.” (Student S04, Year 9)

“I always thought the memory and the processor were the same thing. The simulation showed me they talk to each other through the bus. Now I understand.” (Student S11, Year 9)

“When the teacher drew it on the board, I copied it but I didn’t really understand it. Here, I can press play and watch it happen step by step.” (Student S17, Year 9)

Teacher perspectives reinforced this pattern. One teacher described a qualitative shift in student responses during oral questioning sessions that followed SimuArch-based lessons:

“Before, when I asked what the CPU does, they could only repeat the definition. Now they can explain the cycle, describe what happens between components. They have a mental image of it.” (Teacher T02)

These findings are theoretically consistent with Mayer’s (2009) cognitive theory of multimedia learning, which predicts that the combination of dynamic visual representations with interactive manipulation activates both verbal and pictorial processing channels, thereby deepening conceptual encoding. The recurrent motif of “simultaneous seeing and understanding” across student accounts reflects what Mayer (2009) terms the “coherence effect”: the elimination of extraneous cognitive load through focused, dynamic visuals. Moreover, the specific misconceptions corrected - particularly the conflation of memory and processor, and the invisibility of the bus - correspond precisely to the persistent conceptual errors documented by Armoni (2013) and Kaasboll (2004) in secondary-level computer science learners in resource-limited contexts. The fact that this theme was also identified by Ben-Ari and Peled (2015) as the primary mechanism through which architecture simulators correct misconceptions further validates the Simulation Module’s design. Qualitatively, this theme triangulates and explains the statistically significant post-test performance gain in processor operation ($p^* < 0.001$, +17% differential, Table 2).

3.1.5.2 Increased motivation

Increased motivation was the dominant theme in all four teacher interviews (100%) and emerged in 14 of 20 student transcripts (70%). Participants described qualitative changes in classroom atmosphere and in individual learner posture, including greater spontaneous participation, willingness to attempt difficult exercises, and a reduction in the disengagement behaviours (talking off-task, passivity) that teachers reported as characteristic of traditional computer architecture lessons. Students additionally reported a heightened sense of personal competence and enjoyment during SimuArch sessions, particularly when they successfully completed a Virtual Assembly configuration or achieved a high score in the Assessment Module. These qualitative accounts triangulate directly with the quantitative motivation scale data (Table 3), which showed significantly higher post-intervention scores in the experimental group across all three dimensions: interest in content ($M = 4.2 \pm 0.8$ vs. 3.4 ± 0.9 , $d^* = 0.94$), perceived value ($M = 4.0 \pm 0.7$ vs. 3.2 ± 0.8 , $d^* = 1.06$), and self-efficacy ($M = 3.8 \pm 0.9$ vs. 3.1 ± 0.7 , $d^* = 0.85$).

Teacher accounts captured the collective shift in classroom engagement most vividly:

“The students are more attentive; they ask more questions since we started using SimuArch.” (Teacher T01)

“Before, during the computer architecture lesson, some students would just sleep. Now they compete to be the first to finish the exercises. The atmosphere has completely changed.” (Teacher T03)

Student accounts emphasised the role of immediate feedback and challenge in sustaining motivation:

“When I get a question wrong, it shows me immediately why. So I try again. I don’t give up because I know I can find the right answer.” (Student S06, Year 9)

“I like that I can try things without the teacher watching. If I make a mistake, only the software knows. It’s less embarrassing. So I try more.” (Student S14, Year 9)

These accounts are well-explained by Deci and Ryan’s (2000) Self-Determination Theory (SDT). The students’ descriptions of reduced embarrassment and increased willingness to attempt difficult tasks reflect the satisfaction of the need for competence through graduated challenge and immediate corrective feedback. The self-paced, exploratory design of the Virtual Assembly Module directly satisfies the SDT need for autonomy, consistent with Reeve’s (2016) argument that autonomy-supportive digital environments promote intrinsic motivation and deeper engagement. The reduction in off-task behaviour described by teachers corresponds to the behavioural engagement data, which showed significantly higher effective on-task time in the experimental group (78% vs. 62%, $*p < 0.01$, Figure 2A) and a higher average number of questions asked per session (3.4 ± 1.2 vs. 1.8 ± 0.9 , $*p < 0.001$, Figure 2B). Together, these qualitative and quantitative indicators confirm that SimuArch operated as a motivationally activating environment, consistent with Kapur’s (2016) productive failure framework: the tool’s tolerance of incorrect attempts without social penalty created conditions under which students were willing to engage in the iterative trial-and-error that deepens learning.

3.1.5.3 Collaborative Learning

Collaborative learning emerged as a prominent theme in 13 of the 20 student interviews (65%) and was noted by 3 of the 4 teachers (75%). Participants described spontaneous peer interaction that arose organically during SimuArch sessions, particularly around the Virtual Assembly Module, where the hands-on, problem-solving nature of the tasks created natural opportunities for students to consult one another, compare configurations, and jointly debug incorrect assemblies. This pattern of spontaneous collaboration was not a designed feature of the lesson plan but an emergent consequence of the tool’s interactive structure. It is directly reflected in the observation data: peer collaboration episodes were recorded in 67% of experimental sessions compared to 34% in control sessions ($*p < 0.001$, $\phi = 0.34$, Figure 2C), representing a near-doubling of collaborative activity in the SimuArch group.

Students articulated both the nature and the learning value of these collaborative exchanges:

“We work together to solve the exercises; we explain things to each other.” (Student S08, Year 9)

“My friend was stuck on how to connect the memory. I had already done it, so I showed him. And then he helped me understand the bus part. We both learned more.” (Student S03, Year 9)

“Sometimes we disagree about what to do next. We try both ways and see which one works. It’s like a game.” (Student S19, Year 9)

Teachers observed the peer learning dynamic with particular interest, noting its contribution to lesson efficiency:

“I no longer need to explain everything individually. The students who understand faster explain to those who don’t. It saves time and the explanations are sometimes better than mine - they speak the same language.” (Teacher T04)

The collaborative dynamics described by students and teachers are consistent with Vygotsky’s (1978) Zone of Proximal Development (ZPD), which predicts that peer scaffolding in structured learning environments enables learners to reach levels of understanding that would not be attainable through individual effort alone. The student accounts of jointly debugging incorrect configurations and teaching one another are paradigmatic examples of ZPD-mediated learning: more capable peers serve as cognitive scaffolds for those who lag, while the act of explaining consolidates and deepens the explainer’s own understanding. The teacher’s observation that peer explanations were sometimes more effective than teacher-led instruction - attributed to linguistic proximity - further aligns with constructionist principles (Papert, 1980; Noss & Hoyles, 2017), which argue that learning is most powerful when students construct and communicate artefacts for others. The game-like framing of collaborative disagreement described by Student S19 additionally resonates with Kapur’s (2016) productive failure model: two students testing competing configurations and observing outcomes constitutes precisely the kind of effortful, comparative problem-solving that generates durable conceptual understanding. This theme directly corroborates the collaborative learning emphasis

embedded in SimuArch's design and aligns with Agbo et al.'s (2022; 2023) findings that peer-centred digital tools produce stronger engagement effects in African secondary school contexts.

3.1.5.4 Technical Challenges

Technical challenges constituted the only negatively valenced theme to emerge from the interview data, appearing in 9 of the 20 student transcripts (45%) and in 2 of the 4 teacher interviews (50%). Participants described hardware-related difficulties encountered during SimuArch sessions, including computer freezes, slow loading times, and, in a small number of cases, software crashes that interrupted exercise sequences. These events were linked to the insufficient RAM or processing speed of some of the school's older machines - a constraint anticipated in SimuArch's minimum specifications (1 GB RAM, 1 GHz processor) but not always met across all workstations. Despite this, a consistent and notable pattern across all nine affected student accounts was the subordination of technical frustration to an overall positive evaluation of the tool. Hardware limitations were described as inconvenient but not as barriers to learning.

Student accounts of technical difficulties were typically embedded within broader positive assessments:

"Sometimes the computer is slow and I have to wait. But I still prefer it to the board. At least when it works, I understand." (Student S07, Year 9)

"One time it crashed and I lost my work. That was frustrating. But it didn't happen often. The rest of the time it was fine." (Student S15, Year 9)

"Some computers in the room are older. My neighbour's machine is slower than mine. We sometimes have to share. But we manage." (Student S12, Year 9)

Teachers contextualised these technical limitations within the broader material constraints of the school environment:

"The hardware is old. That is a real problem for us in general, not just for SimuArch. But even with these problems, the students were more engaged than they are with the chalkboard. The tool works even on the weak machines, most of the time." (Teacher T02)

The pattern whereby technical difficulties were consistently outweighed by pedagogical benefits carries important implications for educational technology design and deployment in resource-constrained contexts. It suggests that, provided a tool meets a minimum threshold of functional reliability, motivational and cognitive benefits can sustain engagement even in the presence of hardware-related friction. This finding is broadly consistent with the TAM-based acceptance data (Section 3.4), which showed that ease-of-use perceptions (3.8 ± 0.7 out of 5) were somewhat lower than perceived usefulness (4.1 ± 0.6), yet did not prevent high intention to reuse scores (4.0 ± 0.8). Qualitatively, the teacher account captures the key dynamic: the comparison class - chalkboard-based instruction - set a low engagement baseline against which even an imperfect digital tool appeared substantially superior. From a design perspective, this theme reinforces the importance of SimuArch's low-specification architecture (Warschauer & Matuchniak, 2010; UNESCO, 2019) and points to two concrete improvement directions: (1) further optimisation of resource consumption to extend compatibility to older hardware, and (2) the implementation of session auto-save to prevent loss of student work in the event of crashes. These are prioritised in the planned SimuArch v2.0 development cycle.

Regarding which module contributed most to observed learning gains (RQ4), both students (73%) and teachers (100%) identified the Virtual Assembly Module as the single most impactful feature, citing its hands-on, trial-and-error design as the principal driver of conceptual consolidation. The Simulation Module was rated as most helpful for initial conceptual clarity (aligned with Theme 1 above), while the Assessment Module was credited with reinforcing retention through immediate corrective feedback. These module-level attributions are further discussed in Section 3.2.3.

3.2 Discussion

3.2.1 Interpretation of Results in Light of Prior Research (RQ1)

The results confirm RQ1: using SimuArch produced statistically and practically significant improvements in students' learning of computer architecture ($\eta^2 = 0.18$; $d = 1.21$). This large effect size substantially exceeds the average effect ($d = 0.68$) reported in Chen and Tsai's (2021) meta-analysis of simulation-based computer science learning, and is broadly consistent with the educational technology effectiveness literature (Mayer, 2009, 2019; Sweller et al., 2019). Importantly, the magnitude of the effect compares favorably with Agbo et al. (2022), who reported $d = 0.89$ for a simulation-based computer science intervention in Nigerian secondary schools - a context broadly comparable to Cameroon. The strong gains in processor operation (+35% experimental vs. +18% control) align with SimuArch's Simulation Module, which provides real-time visual representations of CPU operations - a feature shown by Ben-Ari and Peled (2015) to be particularly effective for architecture-related misconception correction.

3.2.2 Impact on Motivation and Engagement (RQ2)

The significant increase in intrinsic motivation in the experimental group ($d = 0.85$ – 1.06) confirms the importance of active, technology-supported engagement. These results are consistent with Deci and Ryan's (2000) SDT, specifically the satisfaction of autonomy (through self-paced exploration), competence (through graduated challenge and feedback), and relatedness (through peer collaboration). The high levels of peer collaboration observed (67% of sessions) are consistent with Vygotsky's (1978) zone of proximal development, wherein peer scaffolding in structured digital environments supports knowledge co-construction.

3.2.3 – Pedagogical Utility, Technology Acceptance, and Module-Level Contributions to Learning (RQ3 & RQ4)

Qualitative findings indicate that the Virtual Assembly Module was perceived as the most impactful feature by both students and teachers. This finding is theoretically consistent with Papert's (1980) constructionism and Kapur's (2016) productive failure: the module's design allows students to attempt incorrect configurations, receive immediate diagnostic feedback, and iteratively refine their understanding. The Simulation Module's contribution to conceptual clarity aligns with Mayer's (2009) dual-coding principle, while the Assessment Module's contribution to retention echoes Hattie and Timperley's (2007) findings on the learning-enhancing effect of timely corrective feedback.

This section synthesizes quantitative TAM (Technology Acceptance Model) data and qualitative interview findings to answer two research questions: how students and teachers perceive SimuArch's pedagogical value (RQ3), and which specific features drive learning gains (RQ4).

3.2.3.1 Technology Acceptance (RQ3)

SimuArch was rated highly useful ($M = 4.1/5$), reasonably easy to use ($M = 3.8/5$), and strongly worth reusing ($M = 4.0/5$). The correlation between perceived usefulness and intention to reuse ($r = 0.67$) exceeded the typical benchmark in educational technology studies ($r \approx 0.50$), suggesting that participatory, context-specific design is a decisive factor in technology acceptance in African school settings.

The gap between usefulness (4.1) and ease of use (3.8) is explained by the comparison baseline: students were not measuring SimuArch against an ideal digital standard, but against chalk-and-board teaching, making even a technically imperfect tool feel transformative. Despite acknowledged hardware problems - freezing, slow loading, occasional crashes - every participant still assessed the tool positively overall. This challenges the pessimistic view that infrastructure deficits necessarily block effective digital tool adoption, while also rejecting the naive optimism that any technology improves learning regardless of pedagogical fit. The evidence supports a middle ground: locally co-designed tools that address real learning needs can achieve high acceptance even in hardware-constrained environments.

3.2.3.2 Module-Level Contributions (RQ4)

Three modules each played a distinct but complementary role:

- **Simulation Module:** Identified as most effective for initial conceptual clarity. Its dynamic animations of processes like the fetch-decode-execute cycle activated dual cognitive channels (visual and verbal), producing the largest between-group sub-topic gain (+35% vs. +18% for processor operations). This aligns with Mayer's multimedia learning theory and extends prior findings to secondary-level learners in sub-Saharan Africa.
- **Virtual Assembly Module:** Cited as the single most impactful feature by 73% of students and all teachers. Its trial-and-error, constructionist design embodies Papert's constructionism and Kapur's productive failure theory. The private, non-judgmental feedback loop encouraged exploratory risk-taking without social embarrassment, deepening conceptual understanding - reflected in a +31% vs. +15% gain on Von Neumann architecture.
- **Assessment Module:** Reinforced retention through immediate, specific corrective feedback - identifying not just what was wrong but why. This matches Hattie and Timperley's high-impact feedback typology ($d = 0.73$) and is particularly contrasted with typical classroom practice, where feedback is delayed, generic, and public.

Crucially, the three modules functioned as an integrated pedagogical progression - from observation (Simulation) to active construction (Assembly) to consolidation (Assessment) - reflecting Bruner's guided discovery model and aligning with Cameroon's national Competency-Based Approach.

An emergent finding was spontaneous peer collaboration, particularly around the Assembly Module. The near-doubling of collaborative episodes in the experimental group (67% vs. 34%) suggests SimuArch structurally generated social learning dynamics consistent with Vygotsky's Zone of Proximal Development.

Taken together, RQ3 and RQ4 explain *why* SimuArch worked (culturally adapted design, authentic pedagogical relevance) and *how* it worked (visualization, active construction, corrective feedback, and peer scaffolding), positioning it as a replicable model for technology-enhanced learning in resource-constrained contexts.

3.2.4 Adaptation to the Cameroonian Context and Academic Contribution

The positive acceptance of SimuArch despite local technical constraints validates the context-centered design approach. The bilingual interface (French/English) and low hardware requirements (1 GB RAM) facilitated adoption in schools where high-specification equipment is unavailable. This finding replicates and extends Agbo et al.'s (2023) results from the DRC and reinforces Ngnoulaye (2020) argument that locally developed tools achieve higher contextual fit than imported platforms. The present study makes several novel academic contributions: (1) it is, to the authors' knowledge, the first quasi-experimental evaluation of a locally developed computer architecture simulation tool in a Cameroonian 9th-grade context; (2) it provides Cohen's d effect size data for simulation-based learning in a sub-Saharan francophone context, enabling future meta-analytic synthesis; and (3) it demonstrates that large effects ($d > 1.0$) are achievable with low-resource, context-adapted digital tools, challenging assumptions about the resource requirements for effective educational technology in Africa.

3.2.5 Practical Implications for Policymakers and Educators

Based on the findings, four concrete recommendations are offered for policymakers and educators in Cameroon and comparable contexts:

- 1) **Gradual integration with structured teacher support:** The introduction of digital tools must be accompanied by structured pre-service and in-service training, as demonstrated by the 6-hour SimuArch training protocol's positive impact on experimental group outcomes.
- 2) **Hybrid pedagogical approach:** A combination of traditional instruction and digital simulation appears optimal in resource-constrained environments; policymakers should revise curriculum guidelines to formally recognize blended approaches.

- 3) Prioritize collaborative digital tools: Tools designed to stimulate peer interaction and collaborative problem-solving should be prioritized in curriculum design and procurement, given the strong engagement effects observed.
- 4) Invest in low-resource tool development: Ministries of Education in the region should fund participatory co-design processes similar to that used for SimuArch, to build a portfolio of contextually adapted open educational resources.

3.2.6 Limitations

Several limitations must be acknowledged. First, the 8-week intervention period does not allow for the assessment of long-term learning retention; longitudinal follow-up is needed. Second, a Hawthorne effect cannot be entirely excluded, as the novelty of the tool may have positively influenced student motivation independently of its pedagogical value. Third, the study is limited to urban schools in Yaoundé, restricting generalizability to rural and peri-urban contexts. Fourth, despite careful matching, residual confounding from unmeasured variables cannot be excluded in a quasi-experimental design. Fifth, the reliance on a single school introduces potential school-level cluster effects that were not modelled in the primary analysis.

3.2.7 Research Perspectives

This study opens several directions for future research. Longitudinal studies assessing retention and transfer of learning beyond 8 weeks are a priority. Investigation of SimuArch's effectiveness in rural schools - where infrastructure differs substantially - would strengthen external validity. Teacher training and distance professional development models should be explored to enable scaled adoption. Finally, integrating adaptive artificial intelligence features capable of automatically adjusting difficulty to individual learner profiles represents a promising personalization direction for future SimuArch versions.

4. Conclusion

This study set out to address three specific objectives: (1) to design a software environment adapted to learning computer architecture in the Cameroonian secondary school context; (2) to evaluate its pedagogical effectiveness through a mixed-methods quasi-experimental study; and (3) to analyze the factors of acceptance and use by students and teachers. All three objectives were achieved. SimuArch was successfully designed through a participatory, context-centered process; its effectiveness was demonstrated through a rigorous mixed-methods quasi-experimental evaluation ($F(1,113) = 24.8$, $p < 0.001$, $\eta^2 = 0.18$, $d = 1.21$); and student and teacher acceptance was empirically confirmed through TAM-based analysis and thematic coding.

These findings contribute to the growing body of literature on technology integration in computer science education in sub-Saharan Africa. They highlight the importance of moving beyond the mere transplantation of tools developed in other contexts toward the creation of solutions genuinely adapted to local realities. Improving computer science education at the secondary level in developing countries requires a holistic approach that combines technological development, structured teacher training, and deep sensitivity to contextual constraints. The key limitations - single-school scope, quasi-experimental design, and short follow-up - should be addressed in future longitudinal multi-site studies. The present work paves the way for further research on optimizing computer science education in Cameroon and comparable African educational contexts, and offers a replicable design model for low-resource digital learning environment development elsewhere in the Global South.

Acknowledgements

The author thanks the school administration, computer science teachers, and students of the Lycée de Nkoleton and affiliated institutions in Yaoundé who participated in this research. Gratitude is also extended to the Ministry of Secondary Education of Cameroon for its institutional support and authorization. This research received no specific funding.

References

- Agbo, F. J., Oyelere, S. S., Suhonen, J., Olalekan, O., & Tukiainen, M. (2022). Design and development of a smart learning environment for introductory programming: Effects on students' computational thinking. *Education and Information Technologies*, 27(3), 4063–4098. <https://doi.org/10.1007/s10639-021-10720-y>
- Agbo, F. J., Oyelere, S. S., Suhonen, J., & Tukiainen, M. (2023). Co-designing a smart learning environment to facilitate computational thinking education in the Nigerian context. *Education and Information Technologies*, 28(4), 4847–4893. <https://doi.org/10.1007/s10639-021-10823-6>
- Agyei, D. D., & Voogt, J. (2011). ICT use in the teaching of mathematics: Implications for professional development of pre-service teachers in Ghana. *Education and Information Technologies*, 16(4), 423–439. <https://doi.org/10.1007/s10639-010-9141-9>
- Armoni, M. (2013). On Teaching Abstraction in CS to Novices. *Journal of Computers in Mathematics and Science Teaching*, 32(3), 265–284. Waynesville, NC USA: Association for the Advancement of Computing in Education (AACE). Retrieved from <https://www.learntechlib.org/primary/p/41271/>.
- Bau, A., Bau, D., Dawson, M., Pickens, C. S., & Dennis, G. (2017). Learnable programming: Blocks and beyond. *Communications of the ACM*, 60(6), 72–80. <https://doi.org/10.1145/3015455>
- Ben-Ari, M., & Peled, Y. (2015). The impact of computer simulations on learning computer architecture. *Computers & Education*, 87, 123–135. <https://doi.org/10.1016/j.compedu.2015.04.010>
- Bice, M. R., Ball, J., & Adkins, M. (2020). Digital technology and physical activity motivation. *International Journal of Sport and Exercise Psychology*, 18(4), 442–455. <https://doi.org/10.1080/1612197X.2018.1545649>
- Bruner, J. S. (1960). *The process of education*. Harvard University Press.
- Chen, Y., & Tsai, C.-C. (2021). Examining the effects of simulation-based learning on students' learning achievement in computer science: A meta-analysis. *Computers & Education*, 162, Article 104103. <https://doi.org/10.1016/j.compedu.2020.104103>
- Davis, F. D. (1989). Perceived usefulness, perceived ease of use, and user acceptance of information technology. *MIS Quarterly*, 13(3), 319–340. <https://doi.org/10.2307/249008>
- Deci, E. L., & Ryan, R. M. (2000). The 'what' and 'why' of goal pursuits: Human needs and the self-determination of behavior. *Psychological Inquiry*, 11(4), 227–268. https://doi.org/10.1207/S15327965PLI1104_01
- Eccles, J. S., & Wigfield, A. (2002). Motivational beliefs, values, and goals. *Annual Review of Psychology*, 53, 109–132. <https://doi.org/10.1146/annurev.psych.53.100901.135153>
- Guzdial, M. (2008). Education: Paving the way for computational thinking. *Communications of the ACM*, 51(8), 25–27. <https://doi.org/10.1145/1378704.1378713>
- Hambrusch, S., Hoffmann, C., Korb, J. T., Haugan, M., & Hosking, A. L. (2013). Computer science curriculum 2013: Reviewing the computer systems knowledge area. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (pp. 659–664). ACM. <https://doi.org/10.1145/2445196.2445395>
- Hattie, J., & Timperley, H. (2007). The power of feedback. *Review of Educational Research*, 77(1), 81–112. <https://doi.org/10.3102/003465430298487>
- Ifinedo, E., Rikala, J., & Hämmäinen, T. (2020). Factors affecting Nigerian teacher educators' technology integration: Considering characteristics, knowledge constructs, ICT practices and beliefs. *Computers & Education*, 146, Article 103760. <https://doi.org/10.1016/j.compedu.2019.103760>
- Kaasboll, J. (2004). Teaching and learning computer concepts in Africa. In G. Marshall & Y. Katz (Eds.), *Learning in school, home and community* (pp. 53–62). Springer. https://doi.org/10.1007/978-0-387-35596-2_6
- Kafyulilo, A. C., Fisser, P., & Voogt, J. (2015). ICT use in science and mathematics teacher education in Tanzania: Developing technological pedagogical content knowledge. *Australasian Journal of Educational Technology*, 31(4), 443–461. <https://doi.org/10.14742/ajet.1240>
- Kapur, M. (2016). Examining productive failure, productive success, unproductive failure, and unproductive success in learning. *Educational Psychologist*, 51(2), 289–299. <https://doi.org/10.1080/00461520.2016.1155457>

- Malmi, L., Sheard, J., Bednarik, R., Helminen, J., Korhonen, A., Myller, N., Sorva, J., & Taherkhani, A. (2010). Characterizing research in computing education: A preliminary analysis of the literature. In *Proceedings of the Sixth International Workshop on Computing Education Research* (pp. 3–12). ACM. <https://doi.org/10.1145/1839594.1839597>
- Mayer, R. E. (2009). *Multimedia learning* (2nd ed.). Cambridge University Press. <https://doi.org/10.1017/CBO9780511811678>
- Mayer, R. E. (2019). Thirty years of research on online learning. *Applied Cognitive Psychology*, 33(2), 152–159. <https://doi.org/10.1002/acp.3482>
- Ministry of Basic Education (MINEDUB). (2014). *Guide to the implementation of the general secondary education curriculum: Computer science*. Ministry of Basic Education.
- Mwila, P. (2018). Assessing the Attitudes of Secondary School Teachers towards the Integration of ICT in the Teaching Process in Kilimanjaro, Tanzania. *International Journal of Education and Development using Information and Communication Technology*, 14(3), 223–238.
- Ngnoulaye, J. (2020). The use of technology for learning among university students in Cameroon. In T. Karsenti, K. Toure, M. Lepage, & S. A. Attenoukon (Eds.), *Uses and adoption of educational technologies in Africa: Some avenues for reflection* (pp. 217–239). Langaa.
- Ngnoulaye, J. (2026). Impact of digital learning tools on student performance in science: A comparative study in two high schools in Yaoundé. *Journal of Social Science and Human Research Studies*, 2(3), 315–321. <https://doi.org/10.65150/EP-jsshhs/V2E3/2026-19>
- Noss, R., & Hoyles, C. (2017). Constructionism and microworlds. In E. Duval, M. Sharples, & R. Sutherland (Eds.), *Technology enhanced learning* (pp. 29–44). Springer.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books.
- Patterson, D. A., & Hennessy, J. L. (2014). *Computer organization and design: The hardware/software interface* (5th ed.). Morgan Kaufmann.
- Piaget, J. (1954). *The construction of reality in the child*. Basic Books.
- Reeve, J. (2016). Autonomy-supportive teaching: What it is, how to do it. In J. C. K. Wang, W. C. Liu, & R. M. Ryan (Eds.), *Motivation in educational research: Translating theory into classroom practice* (pp. 129–152). Springer. https://doi.org/10.1007/978-981-287-691-2_7
- Sanusi, I. T., & Tshukudu, E. (2022). Exploring barriers and perspectives in K-12 computing education in a developing nation. In *Proceedings of the 27th ACM Conference on Innovation and Technology in Computer Science Education* (Vol. 1, pp. 19–25). ACM. <https://doi.org/10.1145/3502718.3524758>
- Shadish, W. R., Cook, T. D., & Campbell, D. T. (2002). *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin.
- Sweller, J., van Merriënboer, J. J. G., & Paas, F. (2019). Cognitive architecture and instructional design: 20 years later. *Educational Psychology Review*, 31(2), 261–292. <https://doi.org/10.1007/s10648-019-09465-5>
- UNESCO. (2019). *ICT in education in sub-Saharan Africa: A comparative analysis of basic e-readiness in schools*. UNESCO Institute for Statistics.
- UNESCO. (2021). *Reimagining our futures together: A new social contract for education*. UNESCO.
- Vygotsky, L. S. (1978). *Mind in society: The development of higher psychological processes*. Harvard University Press.
- Warioba, M. M., Machumu, H., Kulunga, K., & Mtwewe, L. (2022). Adoption of ICT as a pedagogical tool in community secondary schools in Tanzania: Possibilities and constraints. *Education and Information Technologies*, 27(2), 2835–2858. <https://doi.org/10.1007/s10639-021-10715-9>
- Warschauer, M., & Matuchniak, T. (2010). New technology and digital worlds: Analyzing evidence of equity in access, use, and outcomes. *Review of Research in Education*, 34(1), 179–225. <https://doi.org/10.3102/0091732X09349791>
- Wolffe, G. S., Yurcik, W., Osborne, H., & Holliday, M. A. (2002). Teaching computer organization/architecture with limited resources using simulators. *ACM SIGCSE Bulletin*, 34(1), 176–180. <https://doi.org/10.1145/563340.563408>

- Yehezkel, C., Yurcik, W., Pearson, M., & Armstrong, D. (2001). Three simulator tools for teaching computer architecture: EasyCPU, Little Man Computer, and RTLsim. *Journal on Educational Resources in Computing*, 1(4), 60–80. <https://doi.org/10.1145/514144.514732>
- Zhang, L., & Nouri, J. (2019). A systematic review of learning computational thinking through Scratch in K-9. *Computers & Education*, 141, Article 103607. <https://doi.org/10.1016/j.compedu.2019.103607>

Biography

Janvier Ngnoulaye is a researcher and lecturer in Computer Science Education at the University of Yaoundé I, Cameroon. His research interests include educational technology, digital learning environments, and computer science curriculum development for sub-Saharan African secondary schools. He has over 15 years of experience in designing and evaluating pedagogical tools adapted to resource-constrained educational contexts in Central Africa.