



Research article

Impact of a Scratch programming intervention on student engagement in a Nigerian polytechnic first-year class: verdict from the observers

Oladele O. Campbell^{a,b,*}, Harrison I. Atagana^c^a Dept of Computer Science, Niger State Polytechnic, Zungeru, Nigeria^b Institute for Science and Technology Education, College of Science, Engineering and Technology, University of South Africa, Pretoria, South Africa^c Institute for Nanotechnology & Water Sustainability, College of Science, Engineering and Technology, University of South Africa, South Africa

ARTICLE INFO

Keywords:

Improving classroom teaching
 Post-secondary education
 Novice programming
 Student engagement
 Scratch
 Constructionism
 Class observation
 CS1

ABSTRACT

An engaging first programming class (CS1) often inspires students' passion for computer science (CS). However, the evidence in the literature suggests that the average CS1 classes are anything but engaging for many students. The performance of CS compared to other science, technology, engineering, and mathematics (STEM) courses in international student engagement surveys seems to substantiate CS1 failure, attrition rates, and lack of diversity in most CS classes. Meanwhile, for its simplicity in introducing programming to beginners, primary and secondary schools use *Scratch*, an educational programming environment developed by the Massachusetts Institute of Technology, USA. For the same reason, higher institutions now include some forms of *Scratch* instruction in CS1. The question remains, to what extent is *Scratch* engaging, especially for students in higher education? This study addressed this gap by observing college computer science students exposed to a constructionist *Scratch* programming pedagogy. We adopted a descriptive design based on quantitative observations. To observe the class during a weekly 2-hour session, we employed five CS educators, one observer per week. Each observer, employing a 20-item observation protocol, rated the extent of affective, behavioral, and cognitive engagement of first-year CS students in a polytechnic in North Central Nigeria. Most of the students were learning to program for the first time. Analysis of the data showed a significant agreement in the ratings of the five observers for overall student engagement, although the impact was moderate. However, while agreement in their ratings for affective engagement was significant, with a large effect, there was no significant concordance in their ratings for behavioral engagement. Observers also significantly agreed in their ratings for cognitive engagement; however, the impact was moderate. These findings suggest that employing *Scratch* in higher education can be engaging and useful, especially for students with no prior programming experience.

1. Introduction

1.1. Searching for an engaging CS1

Globally, learning to program is the core of the higher education computer science (CS) curricula. The ability to create programs comes with many benefits, but the complexity of this creative process often negatively impacts first-year students' engagement (Sharmin, 2021). While there is no consensus on defining student engagement (Moreira et al., 2020), in this paper, we define student engagement as the ability of an institution or instructor to capture student's interest and commitment toward a course, and the student's active use of various faculties toward learning the course. An engaging first programming class (CS1) often

inspires students' passion for CS. However, most introductory computer science classes are far from engaging (Higgins et al., 2019). Educational statistics from within institutions and those generated by external bodies from educational data collected from institutions in different parts of the world prove this assertion. For instance, the National Survey of Student Engagement, an annual survey of the first and final-year students in colleges and universities in the US, Canada, and other countries, shows that CS classes have always recorded one of the lowest levels of student engagement (Morgan et al., 2018). While some have raised questions about the validity or reliability of the NSSE for measuring student engagement in CS (Morgan et al., 2017), empirical and anecdotal evidence of students' learning outcomes from computer science educators or researchers paint no picture different from what NSSE suggests. Although

* Corresponding author.

E-mail address: 51898772@mylife.unisa.ac.za (O.O. Campbell).

the most recent CS1 failure rate is estimated to be 28% (Bennedsen and Caspersen, 2019), because the experiences of instructors in various parts of the world are far from this global average, the narrative of the high failure rate is not going away from the discourse. For instance, Liénardy et al. (2021) reported a failure rate of 70% for university CS1 students in Belgium. With such a level of failures, CS has also been known to record higher attrition than other STEM courses (Hermans, 2020). The lack of female or minority students in CS suggests another sign of how less engaging computer science is (Falkner and Sheard, 2019). From the foregoing, conventional CS1 programming instruction involving lectures and labs appears not to engage some groups of students.

Attrition of CS students who were once enthusiastic about studying CS probably suggests, among other potential factors, that these students had uninteresting CS1 courses. That was the case with average CS1 classes that got Guzdial and Soloway (2002) lamenting, “Why are we doing such a poor job of getting and keeping students in computer science?” (p. 17). Evidence in the literature suggests that the average CS1 classes are still anything but engaging for many students (Becker, 2019). Falkner and Sheard (2019) capture this dilemma for computer science educators in that seminal work, *The Cambridge Handbook of Computer Science Education* thus:

A recent challenge is the sharp increase in enrollments in many computing courses; at the same time, however, there are signs of decreasing student engagement. Although there has been considerable effort invested into adapting to students’ learning needs, there is an apparent mismatch between the pedagogic approaches we use to teach our students and how they want to learn (Falkner and Sheard, 2019, p. 445).

Medeiros et al. (2019) also found from a systematic review of introductory programming in higher education that motivating and engaging novices remain one of the main learning challenges, further suggesting a lack of engaging CS1.

In Nigeria, there is no rigorous K-12 computer science or programming education; most college students write computer programs for the first time in their higher education CS1 classes. In these programming classes, students learn to write programs using textual programming environments such as Visual Basic, C++, Java, and Python. The cognitive load for learning syntax and developing programs in these environments becomes so high that a significant number of these CS1 students fail or drop out of the course (Hermans, 2020). Therefore, there is a need to make the learning of programming interesting and engaging for these students to reduce the dropout rates in CS in Nigerian tertiary institutions (Mork et al., 2020).

One popular initiative is supporting programming learning with *Scratch* to support our novice students in having an engaging programming experience (Tijani et al., 2020). *Scratch* follows earlier programming languages in the spirit of constructionism - the educational philosophy propounded by Seymour Papert. Constructionism argues that rather than being spoon-fed with knowledge, young people learn better when they are provided with an atmosphere where they can express their creativity, build artifacts of their interests and share them with their peers (Alanazi, 2019). *Scratch* was originally a programming language for the young aged 8 to 16. *Scratch* has become a staple of K-12 programming classes worldwide because of its ease of introducing novices to programming (Rich et al., 2019; Szabo et al., 2019). For the same reason, higher education institutions now include some forms of *Scratch* instructions, either as a CS0 or CS1 course (Becker, 2019; Hijón-Neira et al., 2021).

1.2. Defining an engaging CS1

Before identifying what makes a novice student’s first programming class engaging, let us consider what we mean by student engagement. No consensus exists yet for the definition, forms, and measurements of the construct called *student engagement* (Bond et al., 2020; Moreira et al.,

2020). However, an excellent characterization of student engagement adopted in this study is given below:

Student engagement is the energy and effort that students employ within their learning community, observable via any number of behavioral, cognitive, or affective indicators across a continuum. It is shaped by a range of structural and internal influences, including the complex interplay of relationships, learning activities and the learning environment. The more students are engaged and empowered within their learning community, the more likely they are to channel that energy back into their learning, leading to a range of short and long term outcomes that can likewise further fuel engagement (Bond et al., 2020, p. 3).

From the above definition, we identify three widely acknowledged dimensions of engagement: affective, behavioral, and cognitive.

Simply put, affective engagement refers to a student’s attitude, interest, or motivation toward learning anything; behavioral engagement signifies the tangible efforts the student makes toward learning that thing; cognitive engagement represents the mental faculties the student employs in the learning of that thing.

How do we identify these dimensions of engagement in a class or school? Overlaps exist in the literature, where an indicator referring to a dimension in one study may be labelled an indicator for another dimension in another study. Also, an indicator can be indicated as a continuum, with a negative value indicating disengagement and a positive value showing a level of engagement. However, Bond et al. (2020) conducted a systematic review of studies (n = 243) investigating student engagement in higher education. They provided the top 5 indicators for each dimension of student engagement reported in these studies (See Table 1).

A synthesis from literature and anecdotal reports of those who got “hooked” to CS from their CS1 (Orbey, 2020; Pino-James et al., 2019; Ryoo, 2019; Sorensen, 2021) suggests the following common factors likely to be identified in an engaging CS1 class:

- Cordial supportive relationship between a teacher and the students.
- Collaborative active learning sessions with peers.
- Challenging computing lesson or exercise commensurate with students’ capacity or scaffolded to support students’ learning
- Contents tailored to learners’ interests and needs.
- Commitment to learning, not just grade.
- Curiosity and a can-do attitude toward computing.

1.3. The constructionist programming class

Constructionism is a variant of the Piagetian constructivist theory. Seymour Papert, the South-African born American mathematician and computer scientist, propounded the educational theory (or philosophy) called constructionism (Ellison, 2021). Papert, who worked with Jean Piaget, understood him well. He extended his brand of constructivism, positing that students do not learn only by active processes of *accommodating* and *assimilating* new knowledge depending on what they already know. They do that well in an environment where their creative ideas are allowed to flourish in collaboration with a teacher and their peers as they build artifacts of interest. That implies—unlike conventional educational environment—less structure but more agency of the learner is emphasized in such an environment. This study employed the constructionist approach in introducing novice students to programming using *Scratch*—a constructionist programming environment. In such a constructionist class, Rob and Rob (2018) identified the following as features that make meaningful learning happens:

1.3.1. Facilitator

The teacher facilitates constructionist learning. This is achieved not by spoon-feeding the student with the knowledge, but by providing the

environment that motivates the student's creative expressions in collaboration with peers. So, teachers not removed from a constructionist class, rather than act "as sage on the stage," they work as "a guide on the side."

1.3.2. Context

That is the environment or community in which learning takes place. It is characterized by a less curricular structure but more autonomy for learners.

1.3.3. Collaboration

This is key in a constructionist class: the collaboration between the teacher and the students, and between students and their peers. That interaction engenders sharing of ideas during the creative process.

1.3.4. Sharing

In a constructionist programming class, there is less fear or incidence of code plagiarism as sharing, remixing, or tinkering with other's codes are encouraged. That is captured in Scratch's slogan: *Imagine, Program, Share.*

1.3.5. Tools

These are physical materials such as programming environments, websites, laptops etc.

1.3.6. Product

The ideas and artifacts that students develop and share.

1.3.7. Media

Refers items in the context such as blackboard, whiteboard, or other platforms for displaying contents or demos for learners.

1.4. Related works

Several studies have reported on student engagement in Scratch. Working with Israeli middle school students, Meerbaum-Salant et al. (2013) found that students, apart from learning computer science concepts in a Scratch class, were highly emotionally engaged—although empirical evidence for engagement was not provided. Giannakos, Jacheri, and Leftheriotis (2014) introduced programming to 37 female Greek students aged 12–17. They found that happiness and anxiety predicted students' intentions to learn to program, while the enjoyment of the class did not. Pursuing a similar goal, Yildiz Durak (2020) studied the individual and comparative impacts of two popular block-based programming environments —Scratch and Alice — on Turkish 5th-grade pupils' problem-solving skills, engagement, and computational thinking skills with a focus on students with no prior programming experience. The study reported that while their pre-test engagement scores for both

Alice and Scratch classes were similar, and the post-test engagement scores suggest both classes were equally engaged, those in the Scratch recorded higher post-test engagement scores. The simplicity of the Scratch environment compared to Alice may be the reason for the higher engagement. In a study involving Greek university student teachers with no prior programming experience, Papadakis and Kalogiannakis (2019a) found that while the Scratch intervention from the quantitative data showed moderate cognitive impart, the qualitative data showed a strong affective impact. In another study of Scratch in university CS1, Hijón-Neira et al. (2021) investigated the impact of a guided Scratch Visual Execution Environment on some Spanish students programming learning. They found that the Scratch pedagogy employed led to significant learning gains in all programming concepts examined. However, no information was provided on how engaging the Scratch instruction was since the study did not measure any such affective outcome.

Past studies suggest Scratch can engage novice students who may find textual programming language less engaging (Papadakis and Kalogiannakis, 2019b). However, studies on post-secondary school introductory programming interventions provide anecdotal evidence of their impacts on student engagement (Mork et al., 2020; Santos et al., 2020; Thomas et al., 2018). Not satisfied by usual vague claims or anecdotal evidence for CS1 engagement, Kothiyal et al. (2013) opined, "there is a need for research-based evidence on the nature of this engagement" (pg. 137).

Evidence in the literature also shows that Scratch may not be engaging for university CS1 students, especially when they have prior programming experience or are given too many Scratch lessons. For instance, Martínez-Valdés et al. (2017) reported that most students at a Spanish university did not show motivation or engagement with Scratch, as they expected. Quille and Bergin (2016) also found that Scratch, in a higher education class, appeared to engage less than half of the class in an Irish college and did not lead to increased self-efficacy and students' achievement, compared to those not exposed to Scratch. This raises some questions or cautions for the computer science educators intending to introduce Scratch to higher education students.

It appears some disappointing experiences with Scratch may be attributed to pedagogical or methodological flaws. For instance, Martínez-Valdés et al. (2017) reported that students complained of not learning all the topics and needed more interesting examples than what they were offered, invariably feeling dissatisfied with being introduced to Scratch before Java, leading to them not covering all topics in the curriculum. However, Papadakis and Kalogiannakis (2019b) reported a positive experience with female Greek university pre-service students. After their exposure to 13 weeks of Scratch, using project-based learning, information from focus interviews suggests the students were engaged while learning and working with Scratch. Although, the level of engagement could not be substantiated since it was not measured directly in the study. Interestingly, while students' in Martínez-Valdés et al. (2017) complained of too many Scratch lessons, those of Papadakis and Kalogiannakis (2019b) were willing to learn more.

John Keller's Attention Relevance Confidence Satisfaction (ARCS) model seems to provide a theoretical explanation for the above contrasting experiences with university students introduced to programming using Scratch. According to John Keller's ARCS model, learners' attention is directly related to their satisfaction with learning contents, perceived relevance of what is being taught, and confidence with the instruction (Keller, 2016). When students lose satisfaction and belief in the relevance of what and how we teach, they lose confidence in the learning atmosphere, and therefore we cannot keep their attention or engagement within the class. Keller (2016) cautions that the "use of unnecessary or excessive motivational strategies will take away from valuable instructional time and annoy the students if they are already highly motivated to learn the material" (pg. 7). Thus, employing only Scratch is not the magic wand. We must customize our pedagogical offering to our students to gain or develop a positive level of relevance, confidence, and satisfaction

Table 1. A picture of top engagement indicators.

Rank	Cognitive SE	Behavioural SE	Affective SE
1	Learning from peers (n=54, 22%)	Participation/interaction/ involvement (n=118, 49%)	Positive interaction with teachers and peers (n=100, 41%)
2	Deep learning (n=45, 19%)	Achievement (n=106, 44%)	Enjoyment (n=55, 23%)
3	Self-regulation (n=39, 16%)	Confidence (n=37, 15%)	Positive attitude about learning Interest (n=37, 15%)
4	Positive self-perception (n=26, 11%)	Study habits (n=19, 8%)	Motivation (n=32, 13%)
5	Critical thinking (n=25, 10%)	Attention/focus Assume responsibility (n=17, 7%)	Enthusiasm (n=27, 11%)

Extract from (Bond et al., 2020, p. 14).

before we can engage them as the disappointing outcomes of some studies suggest (Kalelioglu & Gulbahar, 2014; Martínez-Valdés et al., 2017).

Since *Scratch* instruction is employed to motivate higher education students with no prior programming experience, and by drawing from Keller's (2016) pedagogical guide, this study explores how engaging constructionist *Scratch* programming can be for some Nigerian polytechnic CS1 students. Therefore, the research questions were as follows:

1. Is there a significant agreement among observers that a constructionist *Scratch* programming intervention engages novice CS1 students in a Nigerian polytechnic?
2. To what extent is the constructionist *Scratch* programming instruction engaging in a polytechnic CS1 class in terms of affective, behavioral, and cognitive engagement scores?

1.5. Research hypotheses

H₀1: There is no significant agreement among observers in their ratings for overall student engagement during a constructionist *Scratch* programming instruction in a polytechnic CS1 class.

H₀2: There is no significant agreement among observers in their ratings for students' affective engagement during a constructionist *Scratch* programming instruction in a polytechnic CS1 class.

H₀3: There is no significant agreement among observers of their ratings for students' behavioral engagement during a constructionist *Scratch* programming instruction in a polytechnic CS1 class.

H₀4: There is no significant agreement among observers of their ratings for students' cognitive engagement during a constructionist *Scratch* programming instruction in a polytechnic CS1 class.

1.6. Contributions of the study

Evidence of engaging programming pedagogy is likely to resonate with CS educators. Some have employed *Scratch* in their desire to engage higher education students. Meanwhile, research has found mixed, but mostly anecdotal, evidence of the impact of *Scratch* on student engagement in higher education CS1 class. As far we know, no study has provided wide-ranging fine-grained empirical evidence revealing the impact of *Scratch* on student engagement in higher education CS1 class. We contribute to this research gap by employing five CS educators who observed and rated the level of student engagement in a *Scratch* class. To examine their agreement level for each engagement dimension, we analyzed ratings from the observers using Kendall's W available in *vegan* - an R package. We used this R package rather than the classical test in SPSS for these reasons: one, *vegan* has a permutation test, providing higher power; two, the opportunity of conducting—apart from the global test of concordance of the observers for each dimension of student engagement—a posteriori test providing additional insight into concordance between an observer and other observers. Using this approach, we contribute strong evidence to the literature suggesting *Scratch* can engage higher education CS1 students, although the overall impact is moderate. The rest of the article is divided as follows. The next section presents details on the research methodology. Results from the data follow. The article ends with discussions, conclusions, limitations, and suggestions for further study.

Table 2. Reliability of research instruments.

Instrument	Construct	Items	Ordinal Alpha
CSPROQ	Academic background	3	0.72
	Programming Background	17	0.85
	Visual Art background	5	0.75
SCOP	Affective, behavioral, cognitive engagements	20	0.78

2. Methods

2.1. Research paradigm

The researchers took a positivist stance during the study. We realized this by involving independent observers while facilitating the learning sessions, thus keeping a distance so as not to interfere with students' behaviors or bias the data from the observations.

2.2. Research design

This study is a descriptive quantitative study of class sessions in a constructionist *Scratch* intervention using a structured observation method.

2.3. Population and sampling

The study involved students from a polytechnic in north-central Nigeria. We employed purposive sampling to select an entire class of CS1 students learning to program using *Scratch* programming. The interest is to measure the level of student engagement shown by novice computer science students in the *Scratch* class. We employed the same purposive sampling method to select five observers. We chose the observers because of their availability, experience, and being CS educators. However, we employed random assignment to allocate the observers so that each week, an observer recorded events during the 2-hour class lasting five weeks.

2.4. Setting and participants

The *Scratch* class comprised first-year National diploma in CS students at a north-central Nigerian polytechnic. A polytechnic tertiary institution in Nigeria runs mainly National and Higher National Diploma programs, equivalent to two-year and four-year college programs, respectively. The school is in a semi-rural area, with inhabitants majorly agrarian by occupation. Most students were from this region of Nigeria, although a few came from other regions of the country. Because of the lack of opportunities for learning to program in prior education, most of the students learnt to program for the first time during this intervention in the 2015/2016 academic session.

2.5. Instrumentation

We used two instruments in this study: CS1 Students Profile Questionnaire (CSPROQ) — adapted from an earlier study (Meerbaum-Salant et al., 2013) — and *Scratch* Class Observation Protocol (SCOP). CSPROQ was validated by pilot-testing with CS1 students who were not part of this study. SCOP was constructed after reviewing several class observation instruments available online. Two science educators assessed SCOP for its construct validity, and we made some amendments to the instrument following their recommendations. The reliability of CSPROQ and SCOP was calculated with an R package for computing ordinal alpha. The results are presented in Table 2. Research has shown that ordinal alpha provides more accurate reliability measurements for ordinal data (Gadermann et al., 2012). The first author provided the five observers with training on the use of SCOP to enhance agreement in the inter-observer rating of behavior and events during class sessions. The values for both instruments given in Table 2 suggest acceptable levels of reliability.

2.6. Procedure for data collection

2.6.1. Ethics statement

Both the first author's institute and the college of science ethical review committees at the University of South Africa assessed the research proposal and gave ethical approvals to conduct the study. The

polytechnic management also granted permission, and the subjects consented to participate by signing consent forms after the first author informed them about the study.

2.6.2. Data collection

During the first week of the first semester, activities included getting informed consent, administering the demographic questionnaire, preparing the computer lab, training the observers, giving an orientation to lab assistants, installing Scratch on the institution's, and a few students' personal computers.

Class sessions lasted for five weeks; each week's 2-hour session began with Scratch demos—a program addressing topical areas and aspects of the CS1 curricula written by the instructor (the first author) with the students watching on the whiteboard. The students sat in groups of five or six. Then, students working in groups developed their assigned class exercises. The first author, assisted by lab assistants, attended to groups' questions, following the constructionist educational philosophy. Weekly group projects were also assigned and submitted by the students to keep the momentum in the class and cover topics in the curriculum.

Another educational aspect (although not the focus) of this study was the introduction of the students to the online Scratch community during the first week of orientation. Constrained by a lack of resources, few students registered on the site. This gave them access to remix (i.e., tinker with) other members' Scratch codes. Some students also submitted their Scratch codes on the site.

The first author trained the observers, randomly scheduled the dates for their observations, and gave them a copy of the observation protocol.

By random assignment, each week had a different observer to observe the classes, which lasted for five weeks. Each observer did their observation within the 2-h Scratch programming sessions. They observed students for about 30 min to an hour with the observer sitting or standing in a corner of the room or walking around the class while they were working on their Scratch projects. Employing natural sampling of events identified in the class, the observer recorded what they observed as requested by the 20 Likert-type items in the SCOP. The observer filled out the protocol and submitted it to the researcher before leaving the class.

The instrument included seven items with negative wordings to mitigate bias in the observers' ratings. Therefore, we also reverse-coded those seven items before analyzing the data so that a higher score implies a higher engagement level.

Table 8 presents the profiles of the observers. The most senior computer science educator in the study is observer 2, then the Head of the Department. Next is observer 4, who oversaw examinations in the department. observer 1 is the section head in charge of practical labs in the department. According to her years of experience in that section, she is the most senior. While the rest of the observers are full-time staff at the polytechnic, observer 3 is a visiting lecturer. Finally, observer 5 is an instructor handling students' practical lab. She is also the departmental officer in charge of students' placements in industries for the compulsory students' internship called Students Industrial Works Experience Scheme (SIWES). Table 8 lists the observers in the order of their weekly observation; this implies observer 1, the first week; observer 2, the second week, etc. In the submitted SCOP data, two observers omitted two

ratings. We left the missing data since their absence does not threaten the integrity of the data.

2.7. Data analysis

Descriptive and inferential statistics were used to analyze research data. The descriptive statistics include percentage, mode, median and interquartile range (IQR). Nature of data determines descriptive statistics used: percentage for continuous data; mode, median, and IQR, for ordinal data. For the inferential statistics, we employed Kendall's W to test the hypotheses Kendall's W. Kendall's W is suited for answering a research question or testing a hypothesis aimed at determining the level of agreement among m observers rating n objects (in this study, the 20 items in SCOP). Hypotheses were tested at a p-value less than 0.05.

Kraska-Miller (2013) gives three formulas for calculating Kendall's W, one of which is Eq. (1):

$$W = 12S / m^2 n (n^2 - 1) \tag{1}$$

S is the sum of the squared deviation of each item's rating by the m observers from the mean rating for that item, m is the number of judges (in this study, observers), and n is the number of objects or items being rated.

Given that r_{ij} is the rating for item i by observer j, R_i given in Eq. (2) is the total ratings for item i by all observers.

$$R_i = \sum_{j=1}^m r_{ij} \tag{2}$$

Assume \bar{R} is the mean of R_i, i.e., the mean of rating for item i by the m observers, then.

S in Eq. (1) is given Eq. (3):

$$S = \sum_1^n (R_i - \bar{R})^2 \tag{3}$$

The other two equations in (Kraska-Miller, 2013) are Eqs. (4) and (5) below:

$$W = \chi^2 / m(n - 1) \tag{4}$$

where χ^2 is the Friedman chi-square value, m is the number of observers and n is the number of objects (items) being rated.

$$W = (n - 1)\bar{r}_s + 1/n \tag{5}$$

where \bar{r}_s is the mean rank of the Spearman rho correlation coefficient with no tied ranks, n is the number of observers.

The value of W from statistical analysis ranges from 0 to 1 (suggesting no agreement to perfect agreement, respectively). Interpreting the result of the test statistic requires we consider the value of W and the p-value. In this study, the p-value is 0.05. When the p-value is less than 0.05, we have a significant result. However, how significant W is we can determine by following a guideline from Kraska-Miller (2013): value

Table 3. Profiles of study participants.

Gender	N (%)	Age	N (%)	Academic Background	N (%)	Program Writing Background	N (%)	Visual Art Background	N (%)
Male	76 (79.2)	16–18	12 (12.5)	Low	80 (83.3)	None	86 (89.6)	None	36 (37.5)
Female	20 (20.8)	19–21	44 (45.8)	Average	14 (14.6)	Some	10 (10.4)	Some	60 (62.5)
		22–24	34 (35.4)	High	2 (2.1)				
		>24	5 (5.2)						
		Others	1 (1.0)						
Total (N)		96							

Table 4. Descriptive Statistics (Items' frequency of ratings, Median, Mode, and IQR).

Observed Item	Observers' Ratings					Statistics		
	Never (1) Freq. (%)	Sometimes (2) Freq. (%)	Often (3) Freq. (%)	Gen. (4) Freq. (%)	Always (5) Freq. (%)	Median	Mode	IQR
1. Show boredom (A)	4 (80)	1 (20)				1.0	1.0	1.0
2. Confused with the task. (C)	3 (60)	2 (40)				1.0	1.0	1.0
3. Confused with topic (C)	3 (60)	2 (40)				1.0	1.0	1.0
4. Delighted (A)				5 (100)		4.0	4.0	0
5. Show surprise (C)	1 (20)	2 (40)	2 (40)			2.0	2.0 ^a	2.0
6. Show frustration (A)	4 (80)	1 (20)				1.0	1.0	1.0
7. Immersed in class (C)			1 (20)	3 (60)	1 (20)	4.0	4.0	1.0
8. Motivated (A)				4 (80)	1 (20)	4.0	4.0	1.0
9. Self-confidence (C)			1 (20)	3 (60)	1 (20)	4.0	4.0	1.0
10. Show resentment (A)	5 (100)					1.0	1.0	0
11. Creativity (C)			1 (20)	4 (80)		4.0	4.0	1.0
12. 'Gaming the system' (C)	1 (20)		2 (40)	1 (20)	1 (20)	3.0	3.0	3.0
13. Correct designs (C)	2 (40)		1 (20)	1 (20)	1 (20)	3.0	1.0	4.0
14. Correct programs (C)		1 (25)		3 (75)		4.0	4.0	2.0
15. Agile in Scratch (B)		1 (20)		4 (80)		4.0	4.0	1.0
16. Turn in solutions quickly (B)		1 (25)	1 (25)	1 (25)	1 (25)	3.5	2.0 ^a	3.0
17. Inactive (B)	3 (60)	2 (40)				1.0	1.0	1.0
18. Interacting (B)				2 (40)	3 (60)	5.0	5.0	1.0
19. Ask colleagues (B)			1 (20)	3 (60)	1 (20)	4.0	4.0	1.0
20. Ask the teacher (B)		2 (40)		2 (20)	1 (20)	4.0	2.0 ^a	3.0

Note: a. Multiple modes exist. We show the smallest value. A – Affective. B– Behavioral. C–Cognitive.

Table 5. Engagement scores.

Observer	Affective Engagement (Out of 25)	Behavioral Engagement (Out of 30)	Cognitive Engagement (Out of 45)	Overall Engagement (Out of 100)
Observer 1	21.00	23.00	35.00	79.00
Observer 2	23.00	19.00	35.00	77.00
Observer 3	23.00	20.00	33.00	76.00
Observer 4	23.00	24.00	30.00	77.00
Observer 5	24.00	29.00	25.00	78.00

0.10–0.29, 0.30–0.49, and 0.50–1.0 shows weak, moderate, and strong agreement, respectively.

3. Results

The findings presented in this section include the demography of research participants. This is followed by descriptive statistics to make sense of observers' ratings. We conclude the section by presenting the results for hypotheses tests, looking at the extent of agreement between the five observers for various dimensions of student engagements considered in this study.

3.1. Demography of study participants

Table 3 reveals what has been a concern about most CS1 classes: fewer females, often male-populated. Most participants are also academically weak, considering their academic background and age. The academic background was derived from their Unified Tertiary Matriculation Examination (UTME) scores and their final secondary school exam grades in English, Math, and Physics—the three compulsory subjects for admission into CS. The entry age for higher education in Nigeria is 16 years, and we found few in that category, suggesting that many of these students probably could not meet the matriculation requirements when they finished secondary school education or were indigent students who

Table 6. Test of observers' agreement.

Engagement Dimension	Friedman's chi-square	F-statistic		Kendall's W	
		test value	p-value	test value	permutation p-value
Overall	42.56576	3.247172	0.0002*	0.448	0.0003*
Affective	14.462	10.444	0.0004*	0.723	0.002*
Behavioral	9.0244	2.260	0.095	0.361	0.881
Cognitive	17.308	3.051	0.013*	0.433	0.0145*

*Significant at p < 0.05.

lacked the means for enrolment. We also see that only one in ten students in that class have prior program writing experience. That reflects the state of K-12 CSE in that region and most parts of the country—the lack of opportunities for learning to program before higher education. However, more than half of the students have had experience using their creativity in visual art activities.

Table 4 gives a preliminary insight into observers' ratings during the 5-week Scratch sessions. We see that the five observers were unanimous in their ratings (IQR = 0) for two affective engagement items: students showing no resentment but delight in learning to program in Scratch. We find other items (1,2,3, 6,7,8,9, 11, 15, 17,18, 19) had IQR of 1.0, suggesting that they largely agreed with their ratings of these items. Thus, out of the twenty items, they agreed on their ratings for 14 items. This suggests that the five observers found overall student engagement during their observation of class sessions. Five of the fourteen items were affective engagement, showing that the class sessions were highly emotionally engaging for the students. Five out of the nine cognitive engagement items were included in those fourteen items, with a perfect or near-perfect agreement in the observers' ratings. This suggests that, compared to their ratings for affective engagement, they did not find a high level of cognitive engagement among students during their observations. 4 items recorded IQR = 4.0 or 3.0, which shows observers' ratings are polarized rather than agreed.

Table 7. A posteriori test of each observer concordance with the rest.

Engagement	Observer (W_j)				
	Observer 1	Observer 2	Observer 3	Observer 4	Observer 5
Overall	0.3096337	0.5232468*	0.4263924	0.5043026*	0.4446497
Affective	0.4949490	0.8041241	0.8041241	0.8041241	0.6174235
Behavioral	0.15736745 ^a	0.4337857	0.40004	0.3699019	0.3553573
Cognitive	0.2974812	0.5248991	0.3496053	0.4983750	0.4922164

*Significant at $p < 0.05$. W_j = partial concordance per observer. a = this observer has a negative Spearman correlation (see appendix).

3.2. Descriptive statistics: making sense of observers' ratings

Table 5 presents an initial sign of the extent of agreement among the five observers for each dimension of engagement. We computed scores after reverse-scoring negatively worded items in SCOP. The level of agreement was higher for affective engagement, less for cognitive engagement and overall engagement, and none for behavioral engagement.

3.3. Inferential statistics: making inference from observers' ratings

We analyzed ratings from the observers using Kendall's W available in *vegan* — an R package developed by Oksanen et al. (2020). We used this R package rather than the classical test in SPSS for two reasons: one, *vegan* has a permutation test, providing higher power; two, the opportunity of having, apart from the global test of concordance, an a posteriori test providing additional insight into concordance between an observer and other observers (Legendre, 2005).

3.3.1. Testing of observers' agreement for overall student engagement

H₀₁: There is no significant agreement among the five observers that there is an overall engagement of CS1 students in a constructionist *Scratch* programming class.

The test of the above hypothesis given in Table 6 is significant; therefore, we reject H₀. This implies a significant agreement in the ratings of the twenty items by the five computer science educators, although the effect was moderate, $W = 0.448, p = .001$.

3.3.2. Testing of observers' agreement for affective engagement

H₀₂: There is no significant agreement among the five observers that there is an overall affective engagement of CS1 students in a constructionist *Scratch* programming class.

The result of a test of observers' agreement for overall affective engagement in Table 6 is significant. Therefore, we reject H₀₂. This suggests that the five computer science educators agreed that they found the constructionist *Scratch* programming class emotionally engaging for the students with a large effect, $W = 0.723, p = 0.002$.

3.3.3. Testing of observers' agreement for behavioral engagement

H₀₃: There is no significant agreement among the five observers that there is an overall behavioral engagement of CS1 students in a constructionist *Scratch* programming class.

The result of a test of observers' agreement for overall behavioral engagement in Table 6 is not significant. Therefore, we cannot reject H₀₃. This suggests that the five computer science educators were more polarized and did not agree that they found overall behavioral engagement of students in a constructionist *Scratch* programming class.

3.3.4. Testing of observers' agreement for cognitive engagement

H₀₄: There is no significant agreement among the five observers that there is an overall cognitive engagement of CS1 students in a constructionist *Scratch* programming class.

The result of a test of observers' agreement for overall cognitive engagement in Table 6 is significant. Therefore, we reject H₀₄. This suggests that the five computer science educators agreed that the constructionist *Scratch* programming class was cognitively engaging for the students, although with a moderate effect, $W = 0.433, p = 0.01$.

While Table 5 gives a global concordance of all the observers for each engagement construct, Table 7 provides an insight into each observer's concordance with the rest. Table 7 shows that two observers (2 and 4) agreed significantly with one or several other observers in their ratings for overall engagement. Interestingly, Table 8 shows both observers had the most experience, probably suggesting that the more the experience, the better the quality of ratings by an observer. However, the table also shows that the agreement of each of the observers with the rest in their ratings for affective engagement of the students did not reach a significant level. Table 7 reveals that, although each of the observers' ratings for BE had some level of agreement with the rest, none reached a significant level. Note that observer 1 has a negative correlation coefficient (see appendix, Table 14), showing that the ratings for this observer are arbitrary compared to others. Looking at Table 7, the same observer consistently had the lowest values for the partial concordance. We assume two things: inadequate training of observers before the study, and this observer probably should not have been part of the observers for the study (Oksanen et al., 2020, p. 112). It could also mean much change in the class from the second week of the observation, and we should not have started our class observations from the first week. Table 7 also presents a picture of the level of agreement of each observer with the rest in their ratings for cognitive engagement of the students. None of the observers' ratings reached a level that showed significant concordance with the ratings of others.

Table 8. Profile of the observers.

TAG	GENDER	DESIGNATION	QUALIFICATION	EXPERIENCE
Observer 1	Female	Lab Instructor/HOS ^a	HND (CS), PGD(CS) ^c	7 years
Observer 2	Male	Lecturer/HOD	B. Sc.(CS) MSc (IT)	12 years
Observer 3	Male	Lecturer	B.Tech (Physics/CS)	8 years
Observer 4	Male	Lecturer/Exam Officer	B. Tech, MTech.	7 years
Observer 5	Female	Lab Instructor/SIWES Coordinator ^b	HND (CS)	6 years

NOTES: a – Head of Section b. SIWES – Student Industrial Work Experience Scheme. c - Postgraduate Diploma.

4. Conclusions

4.1. Discussion

This descriptive study explored the nature of student engagement in a constructionist *Scratch* programming class for higher education first-year CS students. Five computer science educators observed the class. We sought two key questions in this study. First, is there a significant agreement among the observers that the class was overall engaging for the students? Second, is there a significant agreement in their ratings for

affective, behavioral, and cognitive engagement of students? Analysis of ratings of 20-items in the observation protocol by the five observers shows significant agreement among them, suggesting that overall, they found *Scratch* to be engaging for the students, although the impact was moderate. This finding is consistent with evidence from past studies (Chang et al., 2017; Papadakis and Kalogiannakis, 2019b) that found *Scratch* to be engaging for university students, especially the low-achieving and those with no prior programming experience. With an average overall engagement score of 77.4%, the result of this study is consistent with a similar study employing an active learning approach, which recorded an overall engagement of 83% (Kothiyal et al., 2013). This study contrasts with results from a traditional CS1 class of Irish students learning software development, which recorded an overall low-level engagement score of 5.7 out of 12 (47.5%) (Higgins et al., 2019). This suggests active learning — like the one employed in this study — rather than the traditional approach, delivers a more engaging experience for CS1 students (Luxton-Reilly et al., 2018).

The study also revealed a significant agreement amongst the observers that the *Scratch* intervention had a high affective impact on the students. This result is consistent with previous studies (Almeida et al., 2019; Papadakis and Kalogiannakis, 2019a). However, this finding contrasts with that of Martínez-Valdés et al. (2017) who reported a lower level of motivation in a University CS1 *Scratch* class. The nature of students' educational needs or course design may explain the reason for the low affective engagement.

The results also showed that agreement amongst the observers in their ratings for behavioral engagement did not reach a significant level. This suggests, unlike what the observers found consistently for students' affective engagement during weekly classes, they witnessed a mixed bag of behavior leading to polarized ratings by the observers. A study by Adamopoulos (2017) found that students learning to program come with different perceived levels of the relevance of programming and they express different levels of behavioral engagement with the course. Those who come with a high level of perceived relevance of programming will display a higher level of engagement. The disparity in the observers' ratings on behavioral engagement might be due to problems with the items in the behavioral engagement construct. Another possible cause may be inadequate formal training of the observers, as the result suggests that experience affected the observers' ratings.

This study also found a significant agreement amongst observers that students were cognitively engaged during the *Scratch* instruction, although the impact was moderate. This result is consistent with Papadakis and Kalogiannakis (2019a), who found that a *Scratch* intervention had moderate cognitive impact on pre-service college student teachers.

The above findings might be that students were only enthusiastic about the *Scratch* programming environment without internalizing the programming concepts. Lessons learned from this study are that *Scratch* programming enabled students' engagement in learning to program. We, therefore, conjecture that by exposing them to *Scratch* programming in their CS1, the students' programming interest will increase, and we expect a smooth transition to text-based programming in their 2nd year (Chen et al., 2019).

4.2. Conclusion

The study aimed to examine the nature of engagement in a higher education CS1 class. The results from this study suggest that a constructionist *Scratch* programming intervention can engage first-year CS students in tertiary institutions, especially those without prior programming experience. Observers' agreement for students' affective engagement was significant with a large impact. That implies that students expressed more positive signs of affective engagement with *Scratch*, among other indicators of how engaging the intervention was. Although

their behavioral engagement was unclear, the impact of the intervention on cognitive engagement was moderate. We, therefore, conclude that introductory programming pedagogy using a block-based programming language like *Scratch* will heighten novice students' interest in programming, where direct exposure to text-based programming might hamper their interest.

This study suggests that *Scratch* provides engaging instruction for introducing novice college students to programming.

However, was it *Scratch* or the constructionist pedagogy that fostered the engagement witnessed in this study? An experiment looking at this question may employ two groups of CS1 students: one taught *Scratch* the conventional way, and the other taught *Scratch* the constructionist way. Another limitation of the study is the use of observations and the possibility of erroneous ratings because of the observer's bias, lack of experience, or formal training in using the observation method. We mitigated this concern by employing five CS educators with varying experience levels. In the future, we intend to conduct formal training for all observers by experts in classroom observations before they conduct their ratings and pilot testing the *Scratch* Observation Protocol, to achieve more valid and reliable ratings. Another way of addressing possible errors in the observation is to collect engagement data from students by employing a reliable and valid questionnaire. This will provide data for triangulating the observers' perceptions of student engagement in class. The sample employed in this study also limits the generalization of the result. More representative cohorts of CS1 students could be employed in future studies.

Nevertheless, the finding of this study implies that to engage CS1 students, educators can introduce a constructionist *Scratch* programming course, as a CS0 (an appreciation course) or CS1 in the higher education curriculum for first-year students, especially where students enroll with no prior programming experience. Computer science educators and researchers continue to search for empirically proven ways of engaging the teeming population of novice computer science students in higher education. This study contributes to that agenda.

Declarations

Author contribution statement

Oladele Oladunjoye Campbell: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Harrison I. Atagana: Conceived and designed the experiments.

Funding statement

This work was supported by the sponsorship of a PhD study through the Tertiary Education Trust Fund (TETFund) by the Nigerian government, and Masters and Doctoral Bursary grant provided by the University of South Africa.

Data availability statement

Data associated with this study has been deposited at <https://figshare.com/s/e8afbeaf5a40dca6dcf1>.

Declaration of interests statement

The authors declare no conflict of interest.

Additional information

No additional information is available for this paper.

Appendix

Table 9: Global observers' agreement for overall engagement. Overall test of W for the five Observers. H₀: There is no agreement between the five observers.

Kendall's W =	0.448	Permutation p value = 0.0003*	Reject H ₀
Friedman's chi-square =	42.56576		
F statistics =	3.247172	F distribution p value = 0.0002*	

Table 10: A posteriori test of each observer contribution to the overall W. A posteriori test of the five observers' contributions to the overall W. Ho: This observer is not in agreement with the other four.

	\bar{r}_j	W _j	p-Value	Corrected p-Value	Decision
Observer2	0.4040585	0.5232468	0.0027	0.0135*	Reject Ho
Observer4	0.3803782	0.5043026	0.0035	0.014*	Reject Ho
Observer5	0.3058122	0.4446497	0.0260	0.078	Retain Ho
Observer3	0.2829905	0.4263924	0.0388	0.078	Retain Ho
Observer1	0.1370422	0.3096337	0.2252	0.2252	Retain Ho

Notes: \bar{r}_j = mean of the Spearman correlations with the other observers; W_j = partial concordance per observer; p-value = permutational probability (9,999 random permutations); corrected p = Holm-corrected p-value; * = Reject H₀ at $\alpha < .05$.

Table 11: Overall Observers' agreement on Affective Engagement (AE). Overall test of W for the five observers' judgments on the Affective engagement seen during class sessions. Ho: There is no agreement between the five observers.

Kendall's W	0.723	Permutation p-value = 0.002*	Reject Ho
Friedman's chi-square =	14.462		
F statistics =	10.444	F distribution p-value = 0.0004	

Table 12: A posteriori test of each observer contribution to overall W AE. A posteriori test, of each observer's contribution to the overall W. Ho: This observer is not in agreement with the other four.

Observer	\bar{r}_j	W _j	p-Value	Corrected p-Value	Decision
Observer2	0.7551552	0.8041241	0.1027	0.4990	Retain Ho
Observer3	0.7551552	0.8041241	0.0998	0.4990	Retain Ho
Observer4	0.7551552	0.8041241	0.1050	0.4990	Retain Ho
Observer5	0.5217793	0.6174235	0.3983	0.7966	Retain Ho
Observer1	0.3686862	0.4949490	0.5954	0.7966	Retain Ho

Notes: \bar{r}_j = mean of the pairwise Spearman correlations of observer j with the other observers; W_j = partial concordance per observer; p-value = permutational probability (9,999 random permutations); corrected p = Holm-corrected p-value; * = Reject H₀ at $\alpha < .05$.

Table 13: Overall Observers' agreement on Behavioural Engagement (BE). Overall test of W for the five observers on the Behavioural engagement seen during class sessions. Ho: There is no agreement between the five observers.

Kendall's W	0.361	Permutation p-value = 0.0881	Retain Ho
Friedman's chi-square =	9.0244		
F statistics =	2.260	F distribution p-value = 0.095	

Table 14: A posteriori test of each observer contribution to overall W (BE). A posteriori test, of each observer's contribution to the overall W. Ho: This observer is not in agreement with the other four.

Observer	\bar{r}_j	W _j	p-Value	Corrected p-Value	Decision
Observer2	0.2922321	0.4337857	0.1664	0.832	Retain Ho
Observer3	0.25005	0.40004	0.2093	0.8372	Retain Ho
Observer4	0.2123773	0.3699019	0.2557	0.8372	Retain Ho
Observer5	0.1941967	0.3553573	0.5034	1.000	Retain Ho
Observer1	-0.05329069	0.15736745	0.6679	1.000	Retain Ho

Notes: \bar{r}_j = mean of the pairwise Spearman correlations of observer j with the other observers; W_j = partial concordance per observer; p-value = permutational probability (9,999 random permutations); corrected p = Holm-corrected p-value; * = Reject H₀ at $\alpha < .05$.

Table 15: Overall Observers' agreement on Cognitive engagement (CE). Overall test of W for the five observers on the cognitive engagement seen during class sessions. Ho: There is no agreement between the five observers.

Kendall's W	0.433	Permutation p-value = 0.0146*	Reject Ho
Friedman's chi-square =	17.308		
F statistics =	3.051	F distribution p-value = 0.013	

Table 16: A posteriori test of each observer contribution to overall W (CE). A posteriori test, of each observer's contribution to the overall W. Ho: This observer is not in agreement with the other four.

Observer	\bar{r}_j	W_j	p-Value	Corrected p-Value	Decision
Observer2	0.4061238	0.5248991	0.0352	0.176	Retain Ho
Observer4	0.3729687	0.4983750	0.0631	0.2444	Retain Ho
Observer5	0.3652705	0.4922164	0.0611	0.2444	Retain Ho
Observer3	0.1870066	0.3496053	0.2386	0.4772	Retain Ho
Observer1	0.1218515	0.2974812	0.3416	0.4772	Retain Ho

Notes: \bar{r}_j = mean of the pairwise Spearman correlations of observer j with the other observers; W_j = partial concordance per observer; p-value = permutational probability (9,999 random permutations); corrected p = Holm-corrected p-value; * = Reject H_0 at $\alpha < .05$.

References

- Adamopoulos, F.A., 2017. An Influence Model of the Experience of Learning Programming. RMIT University. <https://researchrepository.rmit.edu.au/esploro/outputs/doctoral/An-influence-model-of-the-experience-of-learning-programming/9921863789201341>.
- Alanazi, A., 2019. A Critical Review of Constructivist Theory and the Emergence of Constructionism. <https://www.researchgate.net/publication/331627180>.
- Almeida, R., Pessoa, T., Gomes, A., 2019. Learning to think like a trainer: bringing Scratch for Educational Sciences professional's formation. In: Proceedings - Frontiers in Education Conference, FIE, 2018-October.
- Becker, B.A., 2019. A survey of introductory programming courses in Ireland. In: Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education, pp. 58–64.
- Benedsen, J., Caspersen, M.E., 2019. Failure rates in introductory programming. *ACM Inroads* 10 (2), 30–35.
- Bond, M., Buntins, K., Bedenlier, S., Zawacki-Richter, O., Kerres, M., 2020. Mapping research in student engagement and educational technology in higher education: a systematic evidence map. *Int. J. Educ. Technol. Higher Educ.* 17 (1), 2.
- Chang, C.K., Yang, Y.F., Tsai, Y.T., 2017. Exploring the engagement effects of visual programming language for data structure courses. *Educ. Inf.* 33 (3), 187–200.
- Chen, C., Haduon, P., Brennan, K., Sonnet, G., Sadler, P., 2019. The effects of first programming language on college students' computing attitude and achievement: a comparison of graphical and textual languages. *Comput. Sci. Educ.* 29 (1), 23–48.
- Ellison, N., 2021. Seymour papert. In: *Encyclopedia Britannica*. <https://www.britannica.com/biography/Seymour-Papert>.
- Falkner, K., Sheard, J., 2019. Pedagogic approaches. In: Fincher, S.A., Robins, A.V. (Eds.), *The Cambridge Handbook of Computing Education Research*. Cambridge University Press, pp. 445–480.
- Gadermann, A.M., Guhn, M., Zumbo, B.D., 2012. Estimating ordinal reliability for Likert-type and ordinal item response data: a conceptual, empirical, and practical guide. *Practical Assess. Res. Eval.* 17 (1), 3.
- Giannakos, M.N., Jaccheri, L., Leftheriotis, I., 2014. Happy girls engaging with technology: assessing emotions and engagement related to programming activities. In: Zaphiris, P., Ioannou, A. (Eds.), *Learning and Collaboration Technologies. Designing and Developing Novel Learning Experiences*. Springer International Publishing, pp. 398–409.
- Guzdial, M., Soloway, E., 2002. Teaching the nintendo generation to program. *Commun. ACM* 45 (4), 17–21.
- Hermans, F., 2020. Hedy: a gradual language for programming education. In: *ICER 2020 - Proceedings of the 2020 ACM Conference on International Computing Education Research*, pp. 259–270.
- Higgins, C., O'Leary, C., McAvinia, C., Ryan, B., 2019. A study of first year undergraduate computing students' experience of learning software development in the absence of a software development process. In: *Proceedings of the 11th International Conference on Computer Supported Education*, pp. 231–240.
- Hijón-Neira, R., Connolly, C., Palacios-Alonso, D., Borrás-Gené, O., 2021. A guided Scratch visual execution environment to introduce programming concepts to CS1 students. *Information* 12 (9), 378.
- Kalelioglu, F., Güllübahar, Y., 2014. The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. *Inf. Educ.* 13 (1), 33–50.
- Keller, J.M., 2016. Motivation, learning, and technology: applying the ARCS-V motivation model. *Participat. Educ. Res.* 3 (2), 1–15.
- Kothiyal, A., Majumdar, R., Murthy, S., Iyer, S., 2013. Effect of think-pair-share in a large CS1 class. In: *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, pp. 137–144.
- Kraska-Miller, M., 2013. *Nonparametric Statistics for Social and Behavioral Sciences*. Taylor & Francis.
- Legendre, P., 2005. Species associations: the Kendall coefficient of concordance revisited. *J. Agric. Biol. Environ. Stat.* 10 (2), 226.
- Liénardy, S., Leduc, L., Donnet, B., 2021. Promoting engagement in a CS1 course with assessment for learning. *Student Success* 12 (1), 102–111.
- Luxton-Reilly, A., Simon, Albluwi, I., Becker, B.A., Giannakos, M., Kumar, A.N., Ott, L., Paterson, J., Scott, M.J., Sheard, J., Szabo, C., 2018. Introductory programming: a systematic literature review. In: *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pp. 55–106.
- Martínez-Valdés, J.A., Velázquez-Iturbide, J.Á., Hijón-Neira, R., 2017. A (relatively) unsatisfactory experience of use of Scratch in CS1. In: *Proceedings of the 5th International Conference on Technological Ecosystems for Enhancing Multiculturality - TEEM 2017*, pp. 1–7.
- Medeiros, R.P., Ramalho, G.L., Falcão, T.P., 2019. A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Trans. Educ.* 62 (2), 77–90.
- (Moti) Meerbaum-Salant, O., Armoni, M., Ben-Ari, M., 2013. Learning computer science concepts with Scratch. *Comput. Sci. Educ.* 23 (3), 239–264.
- Moreira, P., Cunha, D., Inman, R.A., 2020. An integration of multiple student engagement dimensions into a single measure and validity-based studies. *J. Psychoeduc. Assess.* 38 (5), 564–580.
- Morgan, M., Butler, M., Thota, N., Sinclair, J., 2018. How CS academics view student engagement. In: *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, pp. 284–289.
- Morgan, M., Sinclair, J.E., Butler, M., Thota, N., Fraser, J., Cross, G.W., Jacková, J., 2017. Understanding international benchmarks on student engagement: awareness and research alignment from a computer science perspective. In: Sheard, J., Korhonen, A. (Eds.), *Proceedings of the 2017 ITICSE Working Group Reports, ITICSE-WGR 2017*. ACM, Bologna, Italy, pp. 1–24. July 3–5, 2017.
- Mork, K., Migler, T., Wood, Z., 2020. Introducing computing to a cohort of incarcerated youth. In: *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pp. 234–240.
- Oksanen, J., Blanchet, F.G., Friendly, M., Kindt, R., Legendre, P., McGlenn, D., Minchin, P.R., O'Hara, R.B., Simpson, G.L., Solymos, P., Henry, M., Stevens, H., Szoecs, E., Wagner, H., 2020. *Vegan (Version 2.5-7)* [vegan: Community Ecology Package. R package] (2.5-7). <https://cran.r-project.org/web/packages/vegan/index.html>.
- Orbey, E., 2020, July. How Harvard's Star Computer-Science Professor Built a Distance-Learning Empire. *The New Yorker*. <https://www.newyorker.com>.
- Papadakis, S., Kalogiannakis, M., 2019a. Evaluating a course for teaching introductory programming with Scratch to pre-service kindergarten teachers. *Int. J. Technol. Enhanc. Learn. (IJTEL)* 11 (3), 231–246.
- Papadakis, S., Kalogiannakis, M., 2019b. Evaluating a course for teaching advanced programming concepts with Scratch to preservice kindergarten teachers: a case study in Greece. In: *Early Childhood Education*. IntechOpen.
- Pino-James, N., Shernoff, D.J., Bressler, D.M., Larson, S.C., Sinha, S., 2019. Instructional interventions that support student engagement: an international perspective. In: R, A.L., Jennifer, S.L.C., Fredricks, A. (Eds.), *Handbook of Student Engagement Interventions: Working with Disengaged Students*. Elsevier, pp. 103–119.
- Quille, K., Bergin, S., 2016. Does Scratch improve self-efficacy and performance when learning to program in C#? An empirical study. In: *International Conference on Enguaging Pedagogy (ICEP)*, December.
- Rich, P.J., Browning, S.F., Perkins, M.K., Shoop, T., Yoshikawa, E., Belikov, O.M., 2019. Coding in K-8: international trends in teaching elementary/primary computing. *TechTrends* 63 (3), 311–329.
- Rob, M., Rob, F., 2018. Dilemma between constructivism and constructionism: leading to the development of a teaching-learning framework for student engagement & learning. *J. Int. Educ. Bus.* 11, 0.

- Ryoo, J.J., 2019. Pedagogy that supports computer science for all. *J. Educ. Resour. Comput.* 19 (4), 1–23.
- Santos, S.C., Tedesco, P.A., Borba, M., Brito, M., 2020. Innovative approaches in teaching programming: a systematic literature review. In: *CSEDU 2020 - Proceedings of the 12th International Conference on Computer Supported Education*, vol. 1, pp. 205–214.
- Sharmin, S., 2021. Creativity in CS1: a literature review. *J. Educ. Resour. Comput.* 22 (2), 1–26.
- Sorensen, A., 2021. How music and programming led me to build digital microworlds. *Commun. ACM* 64 (9), 7.
- Szabo, C., Sheard, J., Luxton-Reilly, A., Simon, Becker, B.A., Ott, L., 2019. *Fifteen Years of Introductory Programming in Schools: A Global Overview of K-12 Initiatives*. PervasiveHealth: Pervasive Computing Technologies for Healthcare.
- Thomas, E., Mostéfaoui, S.K., Jefferis, H., 2018. Visualising the code: a study of student engagement with programming in a distance learning context. In: *Proceedings of the 11th International Conference on Networked Learning*, pp. 140–148. https://www.networkedlearningconference.org.uk/abstracts/papers/thomas_12.pdf.
- Tijani, F., Callaghan, R., de Villers, R., 2020. An investigation into pre-service teachers' experiences while transitioning from Scratch programming to procedural programming. *Afr. J. Res. Mathemat. Sci. Technol. Educ.*
- Yildiz Durak, H., 2020. The effects of using different tools in programming teaching of secondary school students on engagement, computational thinking and reflective thinking skills for problem solving. *Technol. Knowl. Learn.* 25 (1), 179–195.