



Quantifying the Corona Effect

How much the pandemic-induced switch from face-to-face to online teaching increased students' self-efficacy – a practical report.

Nicolas Fahrni

School of Education Northwestern Switzerland (PH FHNW), Windisch, Switzerland
nicolas.fahrni@fhnw.ch

Alexander Repenning

School of Education Northwestern Switzerland (PH FHNW), Windisch, Switzerland
University of Colorado
alexander.repenning@fhnw.ch

ABSTRACT

This practical report explores the impact of forced learning design changes due to the Corona pandemic. At the School of Education Northwestern Switzerland over 2000 K-6 pre-service elementary school teachers got educated in computer science and computer science education over the last five years employing a learning design evolved through a Design Based Implementation Research (DBIR) approach. Assessing efficacy of the course through effect sizes the 2019 courses have served as pre-Corona baseline. Changing hastily in 2020 to online learning dramatically shifted the learning design in ways not initially anticipated in the DBIR process. Collaborative face-to-face (f2f) learning activities got replaced with individual online learning. Employing effect sizes has allowed us to quantify a Corona effect by comparing self-efficacy measures before Corona and during Corona. While there were only small effect sizes ($0.5 > \text{Cohen's } d \geq 0.2$) all these small effects were positive suggesting that the individual/online seminar worked slightly better than the collaborative/f2f seminar. The report highlights the most important changes to the learning design and compares 2019 with 2020 using effect sizes. For the most part the report can only speculate about the most relevant factors in the design change resulting in the unexpected overall improvement of course efficacy. It could be the shift from collaborative to individual practices, the mandatory peer feedbacks, or the online learning situation.

CCS CONCEPTS

• **Applied computing** → Education; Distance learning; • **Human-centered computing** → Interaction design.

KEYWORDS

computer science education, online learning, collaborative learning, computational thinking

ACM Reference Format:

Nicolas Fahrni and Alexander Repenning. 2022. Quantifying the Corona Effect: How much the pandemic-induced switch from face-to-face to online

teaching increased students' self-efficacy – a practical report.. In *Proceedings of the 17th Workshop in Primary and Secondary Computing Education (WiPSCE '22)*, October 31–November 02, 2022, Morschach, Switzerland. ACM, New York, NY, USA, 6 pages. <https://doi.org/10.1145/3556787.3556865>

1 INTRODUCTION

The measures and the limitations due to Corona had a significant, but hard to quantify effect on teaching practices worldwide [18]. By and large, learning in classrooms has been replaced with online learning approaches. In computer science education the overall effect of this systemic shift is somewhat unclear. On the one hand, social learning practices, such as pair programming or discussions may have been replaced with more difficult to use online mechanisms. On the other hand, however, the pandemic has removed some of the resistance towards technology [5].

Scalable Game Design (SGD) is a K-12 computer science education strategy teaching Computational Thinking (CT) [22] by creating 2D and 3D games and simulations [21]. The strategy was used since 2017 in primary teacher education [17]. As an emergency measure Corona forced a change in the learning design. The face-to-face (f2f) teaching with many cooperative and collaborative interactions among students were replaced by online teaching practices. The SGD teaching practices has been developed and improved since 2017 using Design Based Implementation Research (DBIR). DBIR is an iterative process employing pre/post research instruments to assess the efficacy of the learning design [9] with respect to computational thinking learning outcomes. Over five years the learning design evolved to the point where it was able to provide evidence of efficacy quantifying cognitive as well as affective [19] learning gains as effect sizes [4, 13].

The Corona situation, forcing a significant shift in teaching practice, changed the learning design dramatically in a short period of time. This kind of change was not anticipated by the DBIR process. Instead of fine tuning the next iteration of a learning design based on previous efficacy measures this change was simply forced upon teaching practice for highly pragmatic concerns. Some of these changes included adjustments that, without the forcing function of the measures against the pandemic, we would not have contemplated. For instance, the shift to online learning resulted in a de-emphasis of social learning such as peer programming. We prepared to search for mitigation techniques to overcome negative impact onto the learning outcomes. However, it was a surprise that nearly in all aspects of learning evaluation, the results improved in distance learning.



This work is licensed under a Creative Commons Attribution-Share Alike International 4.0 License.

WiPSCE '22, October 31–November 02, 2022, Morschach, Switzerland
© 2022 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-9853-4/22/10.
<https://doi.org/10.1145/3556787.3556865>

This experience report attempts to quantify the Corona Effect by employing previous iterations of these courses as an efficacy baseline. The report first highlights the most salient features of the changes to the learning design, discusses related work, contrasts pre-Corona and during Corona conditions through effect sizes and discusses possible explanations that could have led to this result.

2 RELATED WORK

Design-Based Research (DBR) [1] and DBIR are relevant research methodologies because they were used to establish a baseline efficacy of a large scale computational thinking course. The idea of design-based research was first mentioned by Ann Brown in 1999 [3]. She was looking for a research approach that investigates learning phenomena not in the laboratory, but in real situations. “Design research is not defined by methodology. All sorts of methods may be employed. What defines design research is its purpose: sustained innovative development.” [2] DBIR grew out of the DBR approach to education. The difference between the two approaches is well described by Kevin Crowley [6] and by Fishman et al. [9]. In DBIR teachers develop and implement the design of an activity together with researchers. Based on the evaluated data, they create a Δ design at each iteration to enhance the desired learning outcomes.

Effect sizes [4] are suitable indicators to quantify the efficacy of educational interventions. Hattie for instance, in his seminal work on visible learning [12], employed effect sizes to rank the efficacy of a very large number of teaching practices. To help researchers with interpreting effect sizes, Hill developed a set of benchmarks establishing typical effect sizes derived from large scale standardized tests such as the improvement of reading comprehension of typical 1st graders in school [13].

To understand the roots of a potential Corona-effect, it's important to understand shifts in social learning practices typical in face-to-face versus online education. Kozar distinguishes between cooperation and collaboration [16]. Collaborative Learning can be defined [8] as “a situation in which two or more people learn or attempt to learn something together.” Research on collaborative learning, for instance [11], includes both “collaborate to learn” and “learn to collaborate.” Schneider et al. summarize in their paper approaches to measure the effects of collaborative learning [24]. They point out that there is neither a generally accepted definition of collaborative nor cooperative learning but different approaches to analyze the issue. Johnson/Johnson compared in their meta-study eight methods of cooperative learning [14]. They conclude that cooperative learning contributes to a small positive learning effect across all 194 studies reviewed. But what if students don't program together in a way of collaborative learning but simply split the work? This collaborative practice of splitting the work would typically be considered a cooperative teamwork [17].

Studies conducted during Corona suggest that under some circumstances collaborative learning methods are less effective than learning individually. Garrote et al. [10] showed that during the lockdown in Switzerland, learning gains in language or mathematics were higher than in a classroom situation. They attribute this increase in efficacy to predominantly individual work caused by the lockdown.

Beyond cognitive shifts there are also attitudinal ones. Crick et al. [5] found an affective impact specific to computer science

education. They found significantly more positive attitudes towards online learning in the UK educational workforce in CS compared to other disciplines.

3 Δ DESIGN – ADAPTION OF THE SCALABLE GAME DESIGN COURSE

Computer science education is mandatory for all primary pre-service teachers who are undergraduate students at the School of Education Northwestern Switzerland (PH FHNW). These pre-service teachers must take two courses in their first year. In the first semester they take a science seminar learning about computational thinking and programming. The second seminar focuses on didactics, but it is not considered in this report. The science seminar consists of three pillars:

- **Motivation and Learning Strategy: Scalable Game Design.** Motivation is at the center of this course. SGD describes the creation of games and simulations and their continuous adaptation to new ideas [22].
- **Tools designed specifically to teach and support Computational Thinking (CT) at the elementary school level.** Students create their projects using CT tools optimized for use with children and youth. They do not need to learn the syntax of a programming language and are supported in the process of abstraction, code creation and debugging [21]. In addition, they work with tools they can use directly in their future teaching.
- **The 7 big ideas of Computer Science from Computer Science Principles.** The mapping of the SGD strategy onto the computer science part of the Swiss curriculum called Lehrplan 21 [7] was straightforward. The three main Lehrplan 21 CS topics (data, algorithms, and systems) were identified as a subset of the 7 big ideas found in the Advanced Placement (AP) Computer Science Principles [20] framework: creativity, abstraction, data, algorithms, programming, networks and global impact.

Even before the pandemic in 2019, flipped classroom practices were used as part of the learning design. The theory was provided with videos or interactive presentations and the discussions on the texts took place in online discussions. The time in the classroom was used mainly for hands-on computational thinking and programming activities. Because of the pandemic the university was forced to switch completely to online education. Because of the previous use of flipped classroom practices the transition to online learning was possible with manageable effort.

Table 1 summarizes the shift in learning design from 2019 (before Corona) to 2020 (during Corona). The six sections below describe the shift in more detail.

- **Programming activities before Corona.** Students worked in small teams of 2 or 3 people to design, and program games. Students largely decided on their own how to organize the collaborative activities. Many of these games produced collaboratively were rooted in existing 2D 1980 arcade game play ideas such as Pac-Man, or Frogger. More sophisticated kinds of games were Tetris (including non-trivial geometric transformations of compound shapes). Other projects were 3D adventure games where a player needs to roam through

Table 1: Learning Design Shift

	before Corona / 2019, f2f	during Corona / 2020, online
Programming	pair programming	individual programming
Peer Review	informal	formal
Grading	individual test + team project	individual project + code walkthrough

3D world to find clues to make it to a final goal. During the f2f lecture students engaged in pair programming [17]. However, in many cases, during lecture or when working at home, the collaborative learning often devolved into cooperative work [16] where two students would simply split up design and programming tasks. Worse, often contributions of the team members were not balanced.

- **Programming activities during Corona:** In online learning, all students had to develop all projects themselves. This would lead to a much larger number of projects. Various tutorials and a weekly online office hour meeting were available for support.
- **Peer review before Corona.** Before Corona peer-review was used informally. While programming in the classroom, teachers periodically asked students to share and exchange their projects in the spirit of Resnick’s creativity spiral [23].
- **Peer review during Corona.** In online learning, peer review had to be formalized. Students had to test and evaluate three projects of fellow students. The assessment was criteria-based using a questionnaire. Through this measure, each student saw not just one - but at least three other projects.
- **Grading before Corona** consisted in the 2019 course a written exam and a collaborative or cooperative project. In the written test, students were asked about the theory, and they had to solve small programming tasks. Beside the evaluation of their competencies one important function of this exam was to motivate (extrinsic) students to engage with the content. After the written test, the students received an assignment to develop a computer game. They could work on this in teams of two or three. The game was presented to and tested by the students in the final session. Half of the grade was based on the exam and half on the project.
- **Grading during Corona:** In online learning, the written exam was cancelled for practical reasons. On the one hand this was because of the massively increased volume of projects which would at least double the project grading time. On the other hand, one purpose of the exam was to create individual grades and to catch students who did not sufficiently contribute to projects. With individual projects this was no longer necessary. The previous collaborative project presentation of the project (a computer game) at the end of the course was replaced by a code walk video. The code walk is a 5-minute video in which every student had to present his/her game and explain the written code.

4 METHOD

The DBIR was conducted at the School of Education Northwestern Switzerland (PH FHNW). To compare the two years, both the results

of the survey (table 3) and the quality of the final projects were considered.

In both years, nearly 400 students started this mandatory course in 16 different groups of approximately 25 students. The course was led by 4 different instructors and counts as 2 Credits (60 working hours). The students’ mean age is 22. 75% of them are female.

At the conclusion of the course, students were asked to complete a survey. The participation in the surveys was anonymous and voluntary. The course was taken by approximately 400 students (pre-service teachers) mostly in their first semester of their bachelor programs. The 2020 response rate described in table 2 was slightly lower than in 2019, but still above 50%.

The survey asked about self-efficacy in CS knowledge/skills questions. It used a four-point Likert scale (4 = strongly agree / 1 = strongly disagree). Only 4 scale levels were defined to avoid the tendency toward the middle [15].

The questionnaire of the online seminar was the same as the year before. Only two questions about the special situation were added. We asked them whether the computer or computer science had become more relevant for them or for schools in general. The questionnaires also included a qualitative course evaluation part, in which students could write an open-ended text response. Second, the instructors grading all projects were asked about the quality of the final projects. This was done informally in an open discussion after the final projects were evaluated. The analysis of these data is not part of this article but will be used to design the next iteration of the seminar. In the discussion section, a few quotes are used to support the conclusion.

5 FINDINGS

As mentioned in section 3 (Δ Design), the instructors were well prepared for teaching online. Therefore, it was hoped that the quality of teaching would only slightly decrease. But the self-efficacy scores increased. The magnitude in shift is captured by two measures. The first one is shift in self-efficacy described in Table 3. The second one is an increase in the project quality described in section 5.2.

5.1 The Corona-Effect

Table 3 describes the Corona effect. We were hoping to see only small undesirable effects but instead found an overall desirable effect. Desirable effects (dark green or light green) can be positive or negative effect sizes depending on how the question was asked. All the small effects (Cohen’s $|d| \geq 0.2$) were desirable. With an average of 0.15 (sign flipped for negatively worded questions) the overall effect is positive and with the largest negative effect being -0.05 (“I can explain the concept of computational thinking” and “I can teach computer science”). Statements “Computer science is difficult” and “Computer science is boring” also have negative

Table 2: Response to survey

	before Corona / 2019, f2f	during Corona / 2020, online
Total of students	370	381
Response to post survey	236	217
Response to post %	63.78%	56.96%

Table 3: The Corona Effect: Effect Sizes resulting from Before Corona to During Corona learning design.

		Before Corona		During Corona		StdDev Spooler	Cohen's d
		Mean	StdDev	Mean	StdDev		
1	I am good with computers	2.72	0.77	2.76	0.63	0.70	0.06
2	I can program	2.62	0.73	2.66	0.70	0.71	0.06
3	I can explain the concept of computational thinking	3.21	0.69	3.17	0.69	0.69	-0.05
4	I would like to learn to program (or improve my skill)	2.61	0.99	2.87	0.86	0.93	0.28
5	I would be excited to create a computer game	2.52	1.07	2.85	0.91	0.99	0.33
6	Computer science is difficult.	2.35	0.87	2.26	0.82	0.85	-0.10
7	Computer science is boring.	1.90	0.84	1.81	0.82	0.83	-0.11
8	I believe, CS is important for my profession as a primary school teacher	2.85	0.79	3.07	0.78	0.78	0.28
9	I believe, CS is important for the future of my students	3.26	0.76	3.49	0.71	0.74	0.31
10	Developing computer games is an excellent way to teach CS	3.08	0.73	3.35	0.69	0.71	0.37
11	Developing simulations is an excellent way to teach CS	2.99	0.69	3.19	0.67	0.68	0.30
12	By creating computer games, kids can learn a lot	3.17	0.75	3.31	0.73	0.74	0.18
13	By creating computer games, kids have a lot of fun	3.41	0.74	3.47	0.69	0.71	0.09
14	I am looking forward to teaching the kids computer science	2.68	0.90	2.93	0.88	0.89	0.28
15	I can teach computer science	2.55	0.79	2.53	0.75	0.77	-0.03
16	I can teach computational thinking	2.94	0.74	2.91	0.70	0.72	-0.05
17	I can teach how to create computer games	2.98	0.71	3.07	0.72	0.71	0.13
18	I can teach how to program simulations	2.83	0.74	2.85	0.71	0.72	0.03
Small effect ($d > 0.2$)							
positive trend ($d > 0.1$)							
no effect							

effect sizes but these questions are formulated negatively so that a negative effect size is interpreted as desirable.

Questions 1-3 show that the students assessed themselves as equally competent in computer science after the course as they did the year before. But they were clearly more motivated to learn programming or to develop a game (question 4 and 5). Question 14 also concerns the motivation of the students. The students were not merely more motivated to learn about computer science, but more excited about teaching computer science themselves in the future.

The strategy of SGD seems to convince the students. With a mean of 3.35, the students are very convinced that computer games are a good way to let children experience computational thinking (question 10). With an effect size of 0.37 this attitude could be improved. For simulations (question 11), the mean value is somewhat lower than for games. But the value is significantly better than the year before.

Further significant positive effects were found in questions 8 and 9. The students rate computer science as important for the primary

teacher and even more important for the future of their students. Both values were significantly higher than in the previous year.

No significant effects were found for the other items. We are especially pleased and surprised that no negative effects (below -0.1) could be found. The survey clearly shows that students considered themselves more competent and were more motivated to understand and teach computer science.

5.2 Quality of Artefacts

In addition to the questionnaire serving as a self-efficacy instrument we also looked at the quality of the final projects. Figure 1 and Figure 2 below show two exemplary games produced. The first one is a 2D adventure style game with 22 agent classes, 218 IF/THEN branches, and 51 methods. The second one is a 3D first person adventure game with 36 agent classes, 64 IF/THEN branches and 40 methods.

Instructors assessed the quality of games using the same criteria grid. All instructors (who had to grade a total of 331 projects) unanimously shared the impression that the level of sophistication of the projects was higher and the game design was better in the



Figure 1: Save your brother: 2D adventure game



Figure 2: Pharaohs Grave: 3D first person game.

online seminar. In both years, students had the same amount of time to create the projects. But they had to work alone and at the same time, by eliminating the written exam, the grade relevancy of the project increased.

5.3 Self-Assessed Corona Impact

The survey of the online course included two additional questions to have students self-assess the impact of the shift towards an online seminar. They agreed (mean value 2.8 of 4 with a StdDev of 0.85) that they invested more time in computer science because they had to do all the work on a computer online. And they also agreed (mean value 3.13 of 4 with a StdDev of 0.86) that the pandemic increased the relevance of computers and computer science for primary education.

6 DISCUSSION

Using the pre-Corona efficacy as a baseline the Corona effect can be quantified. However, while the data indicate an overall positive shift, we cannot attribute this shift to specific changes in the learning design. In this discussion section we speculate about the most salient aspects that could have caused the shift.

The discussion focuses on the elements that were changed in the online seminar: programming activities, peer review, and grading. As mentioned, it is likely that with the increased relevance of computers and computer science during the pandemic, students were generally more motivated and engaged. One could also explain the

improvement by the fact that creative work in computer science is more suited to online learning than in other disciplines such as music, design, or sports. There were various feedbacks from students indicating this:

“Huge compliments to the organization! Compared to other courses, this one was very well designed and structured. The provision of weekly office hours was also very helpful. The workload was sometimes extremely high :) but it was fun.”

“I found the course super structured, and it is one of the most motivating subjects, because you can do a lot by yourself. I find the certificate of achievement great but also time consuming and would have liked more time to be able to implement everything I wanted.”

6.1 Coding alone

Why did the quality of computer games created by students in 2020 improve compared with the projects of 2019? Maybe because students were convinced that developing computer games is a good approach to teach computer science and that Computer Science Education generally gained in importance during the pandemic?

Surprisingly, a potential reason for the positive effect onto self-efficacy could be the omission of cooperative and collaborative learning. The students had to master all their coding activities individually and could not take a back seat in a team. Programming is an activity that, like learning an instrument or a language, is not learned by insight alone but by continuing practice. The students in the seminar typically have very little experience with collaborative learning. Most of them come straight out of the very traditional K-12 school contexts. Lacking these collaborative learning skills, it is not surprising, that social learning approaches employed by preservice teachers are likely to devolve into cooperation, which means to simply split the task.

Being forced to work on their projects alone, students had to develop not only the necessary programming skills but also strategies to cope with challenging situations. For instance, one student described his/her strategy as follows:

“I have no comparison to before Corona, but for me this online course made sense. Because if it had been there, I would have asked you [the instructor] or fellow students much earlier if I hadn’t gotten any further. So, I had to acquire a lot myself and had to search and try out until I knew how to do it. I learned a lot that way.”

6.2 Coding activities with deadlines

Another aspect is closely related to programming activities. While in previous years the submission and assessment of the coding activities were optional, they were mandatory in the online seminar. Students had to submit their projects by a certain date and then had one week to test and assess three projects from their fellows. The necessity of completing all activities increased dramatically.

6.3 Inspiration trumps collaboration

Additionally, peer review provides insight into other projects. This could inspire creativity and the students receive solution hints if something did not work out. Students were inspired by design ideas embodied by other students' games. We started to notice common patterns of game design that some students figured out and others started to copy. So, they would learn from each other but through inspiration not collaboration. Particularly, when social learning degrades into cooperative practises social approaches essentially just become coping mechanisms to deal with workload. In cooperative game development game design is split up between group members similarly the way potluck parties consist of independent contributions. Somebody brings the potato salad, somebody else beer and drinks.

Comparing collaboration with inspiration it would seem for this particular use case involving inexperienced students' inspiration, formally encouraged through peer reviews, may have trumped collaboration.

6.4 Grading

The change in grading could be responsible for certain effects. The written exam, which in previous years took place in week 11, was omitted. Students had to pass a multiple-choice test after each topic to unlock the next one. But it was not part of the grade. Maybe this devalued the theory of computer science and put the focus on programming. On the other hand, students in previous years were often stressed by the exam. They put a lot of effort in memorising the theory, which could have a negative effect on their motivation on computational thinking and the subject of computer science. When the written exam was defined five years ago, this was mainly for pragmatic reasons. Usually at this university, a seminar is concluded by a written exam. The function of this exam was besides summative assessment, selection, and motivation. The students should deal with the learning content because they want to pass the exam at the end. The survey clearly indicates that the elimination of the test did not cause students to engage less with the activities.

6.5 Next steps

Future research should explore if the learning design can be improved benefitting from inspiration but, at the same time, advancing collaboration. One pragmatic challenge is to deal with shifts in workload for the instructors. On the one hand, online learning resources such as learning materials can scale better serving a larger number of students. On the other hand, grading becomes the new bottleneck. Instructors' workload for grading individual projects doubles compared to grading pair projects. Nonetheless, for the next iterations of our learning designs we plan to keep the online/individual strategy.

7 CONCLUSION

The impact of Corona onto courses teaching computer science to pre-service elementary school teachers was quantified through effect sizes. Replacing a collaborative f2f course, providing no scaffolding for successful collaborative practices, with a non-collaborative

online course had an overall small but positive effect onto self-efficacy and improved the quality of projects. Without proper scaffolding project-based group work is likely to devolve into cooperative instead of collaborative learning practice. Cooperative learning, the practice where students split a project up into separate chunks, can be inferior to non-collaborative learning in terms of self-efficacy and product quality.

REFERENCES

- [1] T. Anderson and J. Shattuck, "Design-Based Research," *Educational Researcher*, vol. 41, pp. 16-25, 02/03 2012, doi: 10.3102/0013189X11428813.
- [2] C. Bereiter, "Design research for sustained innovation," *Cognitive Studies: Bulletin of the Japanese Cognitive Science Society*, vol. 9, no. 3, pp. 321-327, 2002.
- [3] A. L. Brown, "Transforming schools into communities of thinking and learning about serious matters," *American psychologist*, vol. 52, no. 4, p. 399, 1997.
- [4] R. Coe, "It's the effect size, stupid," in *British Educational Research Association Annual Conference*, 2002, vol. 12, p. 14.
- [5] T. Crick, C. Knight, R. Watermeyer, and J. Goodall, "The impact of COVID-19 and "Emergency Remote Teaching" on the UK computer science education community," in *United Kingdom & Ireland Computing Education Research conference.*, 2020, pp. 31-37.
- [6] K. Crowley, "What's the Difference between Design-Based Research and Design-Based Implementation Research?" <https://www.informalscience.org/news-views/what's-difference-between-design-based-research-and-design-based-implementation-research> (accessed 30.05., 2022).
- [7] D-EDK, "Vorlage des Lehrplans 21," <https://v-fe.lehrplan.ch> (accessed 25.5.2022).
- [8] P. Dillenbourg, "What do you mean by collaborative learning?," ed: Citeseer, 1999.
- [9] B. J. Fishman, W. R. Peniel, A.-R. Allen, B. H. Cheng, and N. Sabelli, "Design-based implementation research: An emerging model for transforming the relationship of research and practice," *Teachers College Record*, vol. 115, no. 14, pp. 136-156, 2013.
- [10] A. Garrote *et al.*, "Fernunterricht während der Coronavirus-Pandemie," 2021.
- [11] H. Han, *Middle school students' quadrilateral learning: A comparison study*. University of Minnesota, 2007.
- [12] J. Hattie, *Visible learning: A synthesis of over 800 meta-analyses relating to achievement*. routledge, 2008.
- [13] C. J. Hill, H. S. Bloom, A. R. Black, and M. W. Lipsey, "Empirical benchmarks for interpreting effect sizes in research," *Child development perspectives*, vol. 2, no. 3, pp. 172-177, 2008.
- [14] D. W. Johnson, R. T. Johnson, and M. B. Stanne, "Cooperative learning methods: A meta-analysis," 2000.
- [15] A. K. Korman, *Industrial and Organizational Psychology* (Relations Industrielles-industrial Relations). New Jersey: Englewood Cliffs: Prentice Hall., 1971, pp. 513-513.
- [16] O. Kozar, "Towards Better Group Work: Seeing the Difference between Cooperation and Collaboration," in *English teaching forum*, 2010, vol. 48, no. 2: ERIC, pp. 16-23.
- [17] A. Lamprou and A. Repenning, "Teaching how to teach computational thinking," in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, pp. 69-74.
- [18] E. M. Onyema *et al.*, "Impact of Coronavirus pandemic on education," *Journal of Education and Practice*, vol. 11, no. 13, pp. 108-121, 2020.
- [19] R. W. Picard *et al.*, "Affective learning—a manifesto," *BT technology journal*, vol. 22, no. 4, pp. 253-269, 2004.
- [20] C. B. s. A. P. Program and (2016). AP Computer Science Principles, Course and Exam Description. Available: apcentral.collegeboard.com
- [21] A. Repenning, C. Smith, R. Owen, and N. Repenning, "Agentcubes: Enabling 3d creativity by addressing cognitive and affective programming challenges," in *Ed-Media+ Innovate Learning*, 2012: Association for the Advancement of Computing in Education (AACE), pp. 2762-2771.
- [22] A. Repenning, D. Webb, and A. Ioannidou, "Scalable game design and the development of a checklist for getting computational thinking into public schools," in *Proceedings of the 41st ACM technical symposium on Computer science education*, 2010, pp. 265-269.
- [23] M. Resnick, "Sowing the seeds for a more creative society," *Learning & Leading with Technology*, vol. 35, no. 4, pp. 18-22, 2008.
- [24] B. Schneider, N. Dowell, and K. Thompson, "Collaboration Analytics—Current State and Potential Futures," *Journal of Learning Analytics*, vol. 8, no. 1, pp. 1-12, 2021.