Designing activities and tools to support university students' creative and collaborative exploration of physical computing

Melissa Kaivo, David Cuartielles

Arduino AB, Malmö, Sweden, m.kaivo@arduino.cc, d.cuartielles@arduino.cc

Daniel Spikol

Department of Computer Science and Media Technology, Malmö University, daniel.spikol@mau.se

Abstract

Today's digital world requires students to gain skills in collaborative problem solving and digital literacy. One approach is to teach people how to design computational artefacts that require both electronics and programming. Physical computing platforms offer an endless amount of possible opportunities for people to design and develop technological artefacts. However, many times students are overwhelmed when trying to learn both software and hardware simultaneously. The students struggle to be innovative and creative in their projects. Also, they focus on mastering the tool and following instructions for existing projects rather than being able to creatively explore the tool and understand the process of designing and developing new artefacts. For that reason, we aimed to answer the question: What type of tools and activities can be developed to support university students creative exploration of physical computing? Programming and electronics are fundamental design disciplines in today's digital world, and therefore they should be taught through design activities rather than limiting education to textbook readings and laboratory exercises.

We introduce our process of designing activities combined with a supportive tool to ease these challenges. The activities and tools were developed iteratively in three phases with a series of workshops with 126 students and teachers. The tool consists of a set of paper cards that provide necessary details (hints) about the electronics and software and help provide structure for the students to conceptualise how their artefact interacts. We additionally, introduced a learning Jigsaw pattern (orchestration script) for the later intervention that enabled individual students in the groups to focus on design, hardware, or software. For evaluation, we used the Creativity Support Index (CSI), which is a psychometric survey designed to assess the support of the creative process. The instrument investigates collaboration, efforts worth the result, exploration, immersion, enjoyment, and expressiveness. The results between the phases showed improvement with the use of the refined versions of the cards and orchestration of the learning activity. This study has demonstrated that design activities can provide a more accessible approach for the introduction of physical computing to students from various majors. Moreover, learning physical computing through design activities allows the learner to develop computational and design thinking skills for collaboratively solving problems.

Keywords

Collaborative Problem Solving, Learning by Doing, and Creativity.

Introduction

Interactions with technology are exponentially increasing, and the need for understanding the digital world is of utmost importance, especially for young people (Heintz & Mannila, 2018). More than 20 billion connected devices exist in the world; however, less than 1 percent of people have the skills to understand and influence them (British Council, n.d.) To be able to understand digital artefacts and explore the world around them, people need to be introduced with computational thinking and programming. Similarly, design thinking is a significant skill when developing technology to respond to our needs. Furthermore, schools and universities are seeking ways to introduce these subjects to a diverse population of students to prepare them for the changing future (Trust, Maloy, & Edwards, 2018). One of the focal challenges is to develop methods for teaching students from various backgrounds.

In today’s world, computational thinking and programming are acknowledged to be fundamental skills like numeracy and literacy are (Bocconi, Chioccariello, & Earp, 2018). To teach these computational skills, schools and universities commonly use different physical computing platforms (like Arduino, Raspberry Pi, Micro:bit, and Scratch). These different platforms allow teachers to introduce technology, programming and problem solving to students by building tangible and fun projects. These interest-driven projects combine problem solving, engineering, and design. However, physical computing platforms require the learner to work simultaneously with both software and hardware.

Moreover, driving the learners’ focus towards understanding the process of wiring and programming of particular tasks and following step-by-step instruction, rather than gaining skills on computational thinking, designing new solutions, and understanding digital artefacts. Learning both software and hardware at the same time is often perceived overwhelming and students struggle to design new types of artefacts beyond the components, programming concepts, and instructions they have used before. To support students, new kinds of activities and tools are needed to support the creative exploration of physical computing

Additionally, these tools and processes connect the learners to communities of makers and tinkerers. We see this aligning with the broader notions of networked learning through the maker-space ethos that includes: openness in the educational process, self-determined learning, purpose in the cooperative process, supportive learning environment, and a focus on process (McConnell, Hodgson, & Dirckinck-Holmfeld, 2012). For that reason, we aimed to answer the question: What type of tools and activities can be further developed to support university students' creative exploration of physical computing? The aim is to allow students to focus on one aspect of physical computing at a time and design new solutions, not merely re-creating existing ones.

Background

Sentance and Csizmadia (2017) observed that in schools, programming is perceived to be the most challenging aspect of computer science: students have problems with connecting the theoretical concepts to the practical applications and thinking computationally, as well as breaking problems into smaller pieces was perceived difficult. Visual programming tools have been developed to help with these challenges and aim to make programming more accessible for non-technical students. However, the differences between visual programming and traditional text-based programming are considerably extensive. Visual programming languages simplify procedures and guide the user through the creation, for example, with blocks that snap together or use colour-coding. Vihavainen, Paksula and Luukkainen (2011) noticed that a common problem when teaching text-based programming languages is that the focus is too much on learning specific syntax or semantics rather than understanding the process. When students understand the process of coding, they can construct more meaningful programs. Also, Przybylla & Romeike (2014) assert that physical computing encourages learners to use their imagination and creativity, focusing more on ideas, not on technical limitation. Physical computing takes a hands-on approach to understand computational thinking, building tangible artefacts to visualise the abstract programming concepts.

Programming and electronics are fundamental design disciplines, and therefore they should be taught through design activities rather than limiting education to textbook readings and “cookbook” laboratory exercises (Buechley, Eisenberg, & Elumeze, 2007). Even if the outcome does not turn out to be as hoped, students can revise their ideas and create a new version. Iversen and colleagues (2016) argue that design-based activities with tangible digital artefacts provide learners with competencies that reach beyond STEM (Science, Technology, Engineering, and Mathematics) skills. New digital tools, such as 3D printers, laser cutters, and construction kits should expand the forms of learning in classrooms enabling children to learn through the processes of constructing and thinking rather than disabling their thinking by letting them merely carry out ready planned projects. As a matter of fact, design thinking, and computational thinking are both tools for problem-solving (Bowler, 2014).

Learning through design argues that students construct their knowledge by designing and creating meaningful projects and that learning is the most effective when pupils are engaged in creating a tangible artefact . Moreover, Kafai and Resnick (2012) state that activities involving design provide a rich context for learning as knowledge should not be something that is merely transmitted from teachers to students. As previous research state, physical computing can be a more natural way for learners to grasp on programming and computational thinking. Likewise, using familiar techniques and tools decrease the feeling of being overwhelmed and that way help learners to enter a mindset of collaborative creative exploration (Kafai & Resnick, 2012; Qi & Buechley, 2014). We see the practice of physical computing as a strong example of how Networked learning that brings together different aspects of information and communication technology (ICT) that promotes connections between learners and tutors and the larger community around physical computers from maker spaces to open source hardware (Hansen & Dohn, 2018). Koole and colleagues (2018) have argued for the socio-material approach of maker-spaces that combines cooperative process, supportive learning environment, and a focus on process with the hands-on physical, face-to-face-interaction relational style of learning that is augmented by online interactions with people and resources. Opportunities exist to use physical computing as a means to support developing skills in computational thinking and problem-solving through creativity. Developing a supportive tool in the form of paper cards allows students to work with a familiar concept and does not add any utterly new information to students creative exploration of physical computing.

Methodological Approach

We started from our experiences of teaching physical computing to diverse groups of students from K-12, university students across domains and informal and formal learning activities. The common theme across the different activities was that we observed that these diverse students were capable of following the direct and process-oriented workshops, for electronics and programming. The general approach uses a microcontroller, a small computer that users can program and plugin different sensors and outputs. In our case Arduino, which is an open-source hardware and software platform[1].

Arduino was originally created as a tool for artists and designers to have an easy way in the world of electronic artefac production. Participants in workshops are exposed to a combination of building, coding, and reflecting on how digital electronics work. Educational activities around these tools take many forms and are many times contextualized to the educator's knowledge, the learners' point of departure, and even the availability of electronic components or Internet connectivity. Arduino was the first open-source hardware educational tool for all ages. Arduino's design was generic enough for it to be adopted in many different contexts, from primary schools to universities.

From our education and product development backgrounds, we see that students struggle when it comes to collaborative problem solving when they were given a design or engineering task. In other words, how to create an idea and solution, design it, build it, wire up the electronics and program their concepts. Across the different groups of students, the frustration of putting together electronics, programming the software in a concept different than the workshop examples highlighted the challenges of physical computing and the benefits for creating creative learning. Our general approach to running physical computing workshops is to introduce the electronics and the microcontroller first and then begin with the programming. This introducation was accomplished by a series of hands-on workshops and activities where the participants are learning by doing with different semi-completed examples and some open-ended tasks for higher-performing groups.

Working from an iterative human-centred design approach (Bjögvinsson, Ehn, & Hillgren, 2012; Buxton, 2007; Ratto, 2011), we approached the above challenge through several cycles of investigating the needs of the different people (students and teachers) through observations, interviews, the design of different artefacts. We started with the notion that students needed some support and scaffolding to take the process knowledge of the hands-on demonstrations of hardware and software that they can follow and transfer those experiences into the creative collaborative problem-solving. Our initial hypothesis was that we needed to create some tangible like "cheat sheets" about the hardware and the software that would enable the students to carry forward their workshop knowledge into the creative phase of the learning activities.

We used Cherry and Latulipe's (2014) Creativity Support Index (CSI), which is a psychometric survey designed to assess the support the creative process of digital tools. Cherry and Latulipe (2014) developed the CSI as a way to investigate creativity and tasks and the starting point for the instrument was the shorting comings of instruments like the NASA Task Load Index (TLX), which is a standardised survey used to quantify workload (Hart & Staveland, 1988).

Interventions

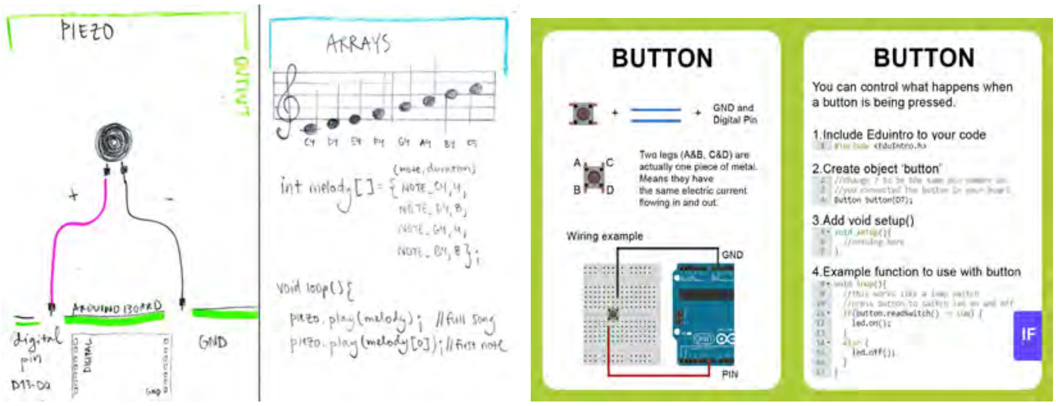
Our initial workshop started with 40 Interaction Design students at their first workshop with physical computing. From this group we worked with a group of 6 students where they used our hand-sketched paper cards (see Figure 1). We observed how the students used the cards, and we interviewed students, teachers, and teaching assistants to evaluate the cards. In these workshops, the participants learn the basics of physical

---

[1] See https://www.arduino.cc/

computing through the process of constructing their projects. Students knew how to design but not the knowledge of how to transform the idea to the electronics and programming. This similar problem was noticed when interviewing teachers and teacher assistants working in Arduino workshops. Students struggled to combine the electronic components and programming concepts as well as to take into considerations components outside of the workshop examples. They were limiting their ideas to the few components they had managed to wire and program properly before.

Fig. 1. First two versions of the paper cards: hand-sketched cards and two-sided cards to help with wiring and programming.



```
int melody[] = {NOTE_C4,4,
                NOTE_D4,8,
                NOTE_G4,4,
                NOTE_B4,8};

void loop(){
  piezo.play(melody);   //full song
  piezo.play(melody[0]); //first note
```

Two legs (A&B, C&D) are actually one piece of metal. Means they have the same electric current flowing in and out.

You can control what happens when a button is being pressed.

1. Include Eduintro to your code
2. Create object 'button'

```
Button button(D7);
```

3. Add void setup()
4. Example function to use with button

```
    led.on();
    led.off();
```

After Design Phase 1, we created sixteen cards, including tips on how to connect and program electronic components commonly used in Arduino workshops, such as LEDs, buttons, resistors, and different sensors. Additionally, we introduced a pre- and post-survey about their knowledge and used the Creativity Support Index for Design Phase 2. For this activity, we tested at a nearby university where the students worked for five days in groups of three. The goal was to see if participants could explore the given sensors and components more freely without being overwhelmed by the fact that they did not yet have the skills to work with them. This workshop had a total of 50 participants. The design activity and cards (tools) under observation seemed to be promising in enabling students to collaborate further, explore new components and concepts, and design and develop new kinds of solutions.

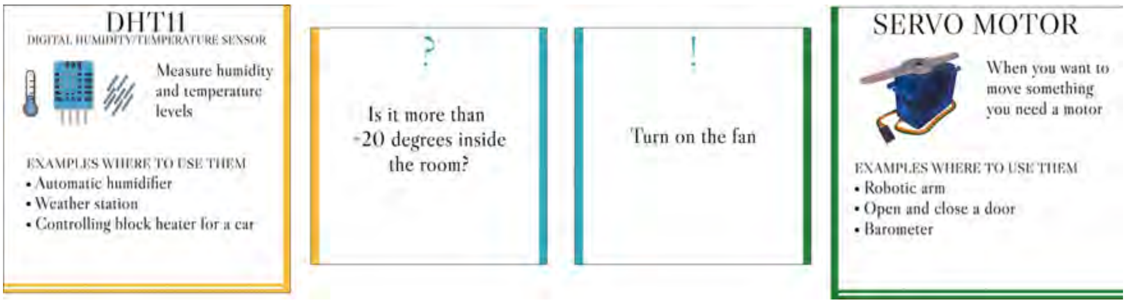
Table 1. Design Interventions across learners

Design Phase	Interventions	Artefacts	n
Interaction Design Students	Interviews, videos	Cards v01	n= 40
I.T. and Learning Students	Surveys interviews, video	Cards v02	n= 50
Engineering Students	Surveys interviews, video	Cards v03	n= 36

Through the feedback from Design Phase 2, we refined the cards (see Figure 2) to be more straightforward and more as a stepping stone to help them design and solve the design task. The cards were divided into three pages: design, electronics, and programming. The idea was that students would be able to create pseudocode with the first page of the cards that could be then transferred to an actual working project using the Arduino components and with the help of the information on the second and third page of the cards. The goal for Design Phase 3 was to see if this kind of activities would shift the focus from mastering everything at once to understanding one aspect at a time and through collaborative working design, assembly and creative program prototypes. This workshop had a total of 36 participants, and this time activity was tested by conducting six separate 70 minutes workshops. Each workshop had six students, divided into two groups of three.

We were inspired from the field of computer-supported learning and the design of learning activities. We chose to use a Jigsaw script, which is a set of instructions that specialise in group formation, distribution of resources, role assignment, and sequences of activities (Fischer, 2007). The Jigsaw method structures segments of the learning activity into expert groups that reform the workgroups to bring specific knowledge back. In our case, this was about electronic, programming the microcomputers, and the design of the task. In this workshop, we used Jigsaw method to see if that would help students to focus on one aspect of physical computing at a time and reduce the feeling of being overwhelmed (Pozzi, 2010).

Fig. 2. Refined set of cards for the third design phase to help with designing.



At the beginning of each workshop, two student groups had 10 minutes to get familiar with Arduino and an open-ended problem given to them. After 10 minutes, these two initial groups were divided again, now into three expert groups: hardware, software and design — each group having now two members, one member from both of the initial groups. Expert groups received a kit, including a set of cards and a specific task for each group and a half-constructed Arduino project. Hardware group received a piece of code in the IDE, and their task was to with the help of the cards to wire the components accordingly to work together with the ready-made code.

Additionally, the software group received components that were ready wired, and their task was to with the help of the cards to create a code to work together with these components. Design group's task was to explore input and output cards and create different types of solutions with the if-then conceptualising cards without needing any knowledge on how to wire and program these components. After working 20 minutes in the expert groups, the students went back to the initial groups and started to design a solution for the given problem. The idea was that students would work collaboratively sharing their knowledge gained from expert groups and developing a solution using the cards, Arduino IDE, microcontroller and a set of components.

Results

For evaluating the learning activities and tools (the cards), we used the Creativity Support Index (CSI). The CSI scoring system maps to educational grading systems, and researchers can use grades as a rule of thumb. For instance, a score above 90 is an "A," which indicates excellent support for creative work. A score below 50 is an "F," which indicates that the tool does not support creative work. The CSI also generates individual factor scores that can help a researcher understand how a tool supports various aspects of creative work. The CSI consists of two different types of ratings. The first section includes statements about the six factors (Collaboration, Results worth the Effort, Exploration, Immersion, Expressiveness, and Enjoyment) on a Likert scale where participants rate how well the cards matched with these factors (see Appendix A). Statements were about working with the set of cards, such as it was easy to work with others, explore different ideas, and be creative while using the tool. On the other section (see Appendix B), participants rate what factors they value the most when doing this kind of activity.

Table 2. Intervention Design Phase 2 The average CSI score was 67.70 (SD 12.32, n=50).

Factors	Avg. Factor Counts (SD)	Avg. Factor Score (SD)	Avg. Weighted Factor Score (SD)
Collaboration	2.67 (1.43)	15.35 (3.68)	41.47 (24.38)
Results worth Effort	2.27 (1.73)	13.39 (3.32)	28.33 (23.69)
Exploration	2.27 (1.35)	13.22 (3.76)	28.98 (18.46)
Immersion	2.16 (1.49)	10.53 (3.80)	22.04 (21.56)
Expressiveness	2.71 (1.71)	12.98 (2.89)	36.16 (23.02)
Enjoyment	2.88 (1.51)	14.53 (3.53)	41.71 (25.29)

The CSI is scored first by multiplying each factor score by its factor comparison count (the number of times it was chosen in the factor comparisons). Then, these are summed and divided by three for 0-100 index score (Carroll & Latulipe, 2009). The factor counts for any particular factor are on a range of 1 to 5, with 5 being the highest possible score. The factor score is the agreement statement responses for each which is on a scale of 1 to 20; therefore, the maximum score is 20. Weighted factor scores are calculated by multiplying a participants factor agreement scale score by the factor count, in order to make the weighted factor score more sensitive to the factors that are most important to the given task (Cherry & Latulipe, 2014).

If we look at CSI for Design Phase 2 (see Table 2), we see that the overall average score is 67.70, providing a satisfactory score. However, for Design Phase 3 (see Table 3), the overall score moved up to 82.4. If we compare the different factors, we can see that Enjoyment factor showed the most improvement between the different design phases. Also, with small improvements in all other factors (Collaboration, Results worth the Effort, and Expressiveness). However, the Immersion factor showed a decline between the two phases. In order to understand how the tools and the activity can impact the learning experience, we can investigate the different factors.

Collaboration

Between Phase 2 and 3 Collaboration, average factor counts are 2.67 and 2.44 suggesting with the first intervention that collaboration was more critical to Phase 2 while not as crucial for Phase 3. However, the average factor scores point towards value of collaboration being higher with a score of 17.61 compared to 15.35 with a lower SD (1.86 compared to 3.68) suggesting that students in Phase 2 felt more engaged with their collaboration.

Table 3. Intervention Design Phase 3 The average CSI score was 82.4. (SD 14.19, n=36)

Factors	Avg. Factor Counts (SD)	Avg. Factor Score (SD)	Avg. Weighted Factor Score (SD)
Collaboration	2.44 (1.30)	17.61 (1.86)	42.5 (22.24)
Results worth Effort	2.64 (1.44)	16.67 (4.01)	42.5 (24.14)
Exploration	2.08 (1.36)	16.11 (3.11)	33.61 (24.58)
Immersion	1.22 (1.73)	15.08 (4.37)	17.72 (25.04)
Expressiveness	2.42 (1.20)	15.17 (3.72)	37.03 (21.71)
Enjoyment	4.19 (0.98)	17.58 (3.33)	73.75 (23.36)

Results worth the Effort

The average factor count for Results Worth the Effort in Phase 2 was 2.27 and in Phase 3 was 2.64, suggesting some importance to the students. While the average weighted scores respectively were 13.39 and 16.67, suggesting that the students required more effort to get results. However, it is essential to note that the scenarios are different in time frame and situation. The students in Phase 3 had only limited time to complete a task while in Phase 2, the students had more time, over two days.

Exploration

The average factor count for Exploration was 2.27 in Phase 2 and 2.08 in Phase 3, suggesting low importance for the students. While the average factor score for Phase 2 was 13.22 and for Phase 3 was 16.11 suggesting that the students felt more comfortable exploring the different design and technical options of the assignment with the revised cards and learning activity.

Immersion

Between the two design phases, the average factor count for Immersion for Phase 2 was 10.53, and for Phase 3 it was 15.08, illustrating the perceived level of immersion by the students. However, the SD was large for this factor. Additionally, similar to the Exploration factor, students in Phase 2 felt that Immersion was easier while in Phase 3 the pressure to perform and the time constraints limited the feeling of Immersion with more disturbances and notifications.

Expressiveness

The average factor count of Expressiveness for Design Phase 2 was 2.71 and for Phase 3 was 2.42 illustrating that the students did not generally feel that the assignment allowed them to express their ideas from concept to prototype. While the average factor score for Phase 2 was 12.98 and for Phase 3 was 15.17 illustrating the fact that the students may have experienced more control of the design tasks through the Jigsaw and the cards allowing them to work on their concepts more effectively.

Enjoyment

The average factor count for Enjoyment between Phase 2 and Phase 3 are 2.88 and 4.19, showing that this factor was especially high for Phase 3 and somewhat high in Phase 2. While the average factor scores show that for Phase 2, the students enjoyed the activity slightly less than Collaboration. However, for Phase 3, the Enjoyment factor is significantly higher, giving us insight that supporting the design activity and the tasks with additional learning materials need further investigation.

Discussion

Using the Jigsaw script together with an open-ended problem and the cards allowed students to design creative solutions with components and concepts they had not used before. Having a real purpose in learning, creating a solution to an existing problem engages students with the learning process. After being introduced to one part of the process: hardware, software, or design students worked collaboratively, sharing their new knowledge and ideas, designed a solution to an existing problem and developed the final artefacts while simultaneously learning the skills required for the whole process. Students had the opportunity to share their expertise and encourage each other by doing and trying together. Our workshop results confirm that chopping the design activity to three parts by using the Jigsaw method students were able to focus on one thing at a time. Cards made it easy for students to still have enough information without knowing about the other aspects and then collaboratively combine those three aspects into functional projects.

In Design Phase 1 and 2, when developing these skills with Arduino, students were overwhelmed and mastering two things, software and hardware at the same time was complicated. Significant challenges were encountered when students were required to apply the knowledge gained from the step by step assignments to their projects. Because previously students used wiring and coding examples without more profound understanding, they were not able to modify these examples to work with their project. These lead students to limit their thinking to components that they managed to wire and program correctly with the previous assignments. Students were struggling to explore what can be done with different components, and many groups were using the outputs and inputs for the same purposes.

For the challenges that students encountered when working with their projects, the first version of the prototype was not enough. Students needed help on how to construct their project, how to think logically, and where to start the whole process. Moreover, this is where the design activity with the cards, tested in Design Phase 3, was found helpful. The support that was needed to explain the abstract concepts had to be something else than visualisations of each part of the process, such as tips on how to wire an LED correctly. Moreover, students required help with conceptualising and exploring creative ways to use technology.

Creating a design of the project with the input, output, and conceptualising cards were working as a bridge between building the understanding of technology and developing the skills to work with the technology. Students could take a problem they have witnessed in the real world and with the help of the cards break this problem into smaller, easier to handle tasks. Breaking down to what inputs are needed to detect the world and what outputs should be triggered. When they had chosen all the inputs and outputs, they could start thinking logically: if this is detected, then this should act accordingly. Students constructed their knowledge of design thinking, computational thinking, technologies, and different components work as well as they were immediately able to apply and test their new knowledge when building and programming designed projects.

We take the view that creativity support tools are tools (digital and physical) that can be used by people in the open-ended creation of new artefacts. Our aim, with the work, has been to investigate how to create tools and activities, can be further developed to support university students creative exploration of physical computing. We see from the results that both the process (the Jigsaw) and the support materials (the cards) support collaboration, that students felt that the creativity support tools helped them achieve what they wanted. Maybe, more importantly, that the students had higher enjoyment when we combined the Jigsaw and the cards.

Conclusion

Framing our work on learning networks, we see the creativity tools support the learning outcomes, activities and tasks. While the settings and the divisions of labour can be supporting with orchestration (scripts) they are emergent (Goodyear, Carvalho, & Dohn, 2014). We have tried to focus on the physical settings through the nature of the both the digital and the materials through the use of physical computing. The CSI is a relevant instrument that can support the design of networked learning and provide insight.

This study has demonstrated that design activities can provide a more accessible approach to introduce physical computing to students from various majors. Moreover, learning physical computing through design activities allows the learner to develop computational thinking and design thinking skills for collaboratively solving problems. It is essential to develop different approaches to teach computational thinking and programming to a diverse population of people. To allow everyone to understand, actively participate, and communicate with the digital world around us.

Next steps for future work involves more structured research design with more similar groups of learners and longer time frames. One issue not to overlook is that in Phase 2, the students were generally from a social science background while Phase 3 were engineering students who were generally more familiar with lab workshops and different aspects of technology. However, the structuring of the learning activities with the support of the cards provides needed orchestration for teaching creativity with technology.

References

Bjögvinsson, E., Ehn, P., & Hillgren, P.-A. (2012). Design Things and Design Thinking: Contemporary Participatory Design Challenges. *Design Issues*, *28*(3), 101–116. https://doi.org/10.1162/DESI_a_00165

Bocconi, S., Chioccariello, A., & Earp, J. (2018). The Nordic approach to introducing Computational Thinking and programming in compulsory education. *Report Prepared for the Nordic@ BETT2018 Steering Group. Doi: Https://Doi. Org/10.17471/54007.*

Bowler, L. (2014). Creativity through "maker" experiences and design thinking in the education of librarians. *Knowledge Quest*, *42*(5), 58–62. Retrieved from https://go.galegroup.com/ps/anonymous?id=GALE%7CA371688431&sid=googleScholar&v=2.1&it=r&linkaccess=abs&issn=10949046&p=AONE&sw=w

British Council. (n.d.). Creative computing in Bangladesh. Retrieved September 9, 2019, from https://www.britishcouncil.org/work/partner/coding-bangladesh

Buechley, L., Eisenberg, M., & Elumeze, N. (2007). Towards a curriculum for electronic textiles in the high school classroom. In *ITiCSE 2007: 12th Annual Conference on Innovation and Technology in Computer Science Education - Inclusive Education in Computer Science* (Vol. 39, pp. 28–32). New York, New York, USA: ACM Press. https://doi.org/10.1145/1268784.1268795

Buxton, W. (2007). *Sketching user experiences : getting the design right and the right design.* Amsterdam: Elsevier/Morgan Kaufmann.

Carroll, E. A., & Latulipe, C. (2009). The Creativity Support Index. In *Conference on Human Factors in Computing Systems - Proceedings* (pp. 4009–4014). https://doi.org/10.1145/1520340.1520609

Cherry, E., & Latulipe, C. (2014). Quantifying the creativity support of digital tools through the creativity support index. *ACM Transactions on Computer-Human Interaction*, *21*(4). https://doi.org/10.1145/2617588

Fischer, F. (2007). *Scripting computer-supported collaborative learning [Elektronisk resurs] cognitive, computational and educational perspectives.* Boston, MA: Springer Science+Business Media, LLC.

Goodyear, P., Carvalho, L., & Dohn, N. B. (2014). Design for networked learning: framing relations between participants' activities and the physical setting. In *Proceedings of the 9th international conference on networked learning* (pp. 137–144).

Hansen, J. J., & Dohn, N. B. (2018). Analysing learning designs of'learning through practice'as Networked Learning. In *11th International Conference on Networked LearningNetworked Learning Conference* (pp. 447–454).

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology*, *52*, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

Heintz, F., & Mannila, L. (2018). Computational Thinking for All. In *Proceedings of the 49th ACM Technical Symposium on Computer Science Education - SIGCSE '18* (pp. 137–142). New York, New York, USA: ACM Press. https://doi.org/10.1145/3159450.3159586

Iversen, O. S., Smith, R. C., Blikstein, P., Katterfeldt, E.-S., & Read, J. C. (2016). Accepted Manuscript Digital fabrication in education: Expanding the research towards design and reflective practices. *International Journal of Child-Computer Interaction*. https://doi.org/10.1016/j.ijcci.2016.01.001

Kafai, Y. B., & Resnick, M. (2012). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Routledge.

McConnell, D., Hodgson, V., & Dirckinck-Holmfeld, L. (2012). Networked learning: A brief history and new trends. In *Exploring the theory, pedagogy and practice of networked learning* (pp. 3–24). Springer.

Pozzi, F. (2010). Using Jigsaw and Case Study for supporting online collaborative learning. *Computers & Education*, *55*(1), 67–75. https://doi.org/10.1016/J.COMPEDU.2009.12.003

Przybylla, M., & Romeike, R. (2014). Physical computing and its scope - towards a constructionist computer science curriculum with physical computing. *Informatics in Education*, *13*(2), 241–254. https://doi.org/10.15388/infedu.2014.05

Qi, J., Buechley, L., Qi, J., & Buechley, L. (2014). Sketching in circuits. In *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (pp. 1713–1722). New York, New York, USA: ACM Press. https://doi.org/10.1145/2556288.2557391

Ratto, M. (2011). Critical Making: Conceptual and Material Studies in Technology and Social Life. *The Information Society*, *27*(4), 252–260. https://doi.org/10.1080/01972243.2011.583819

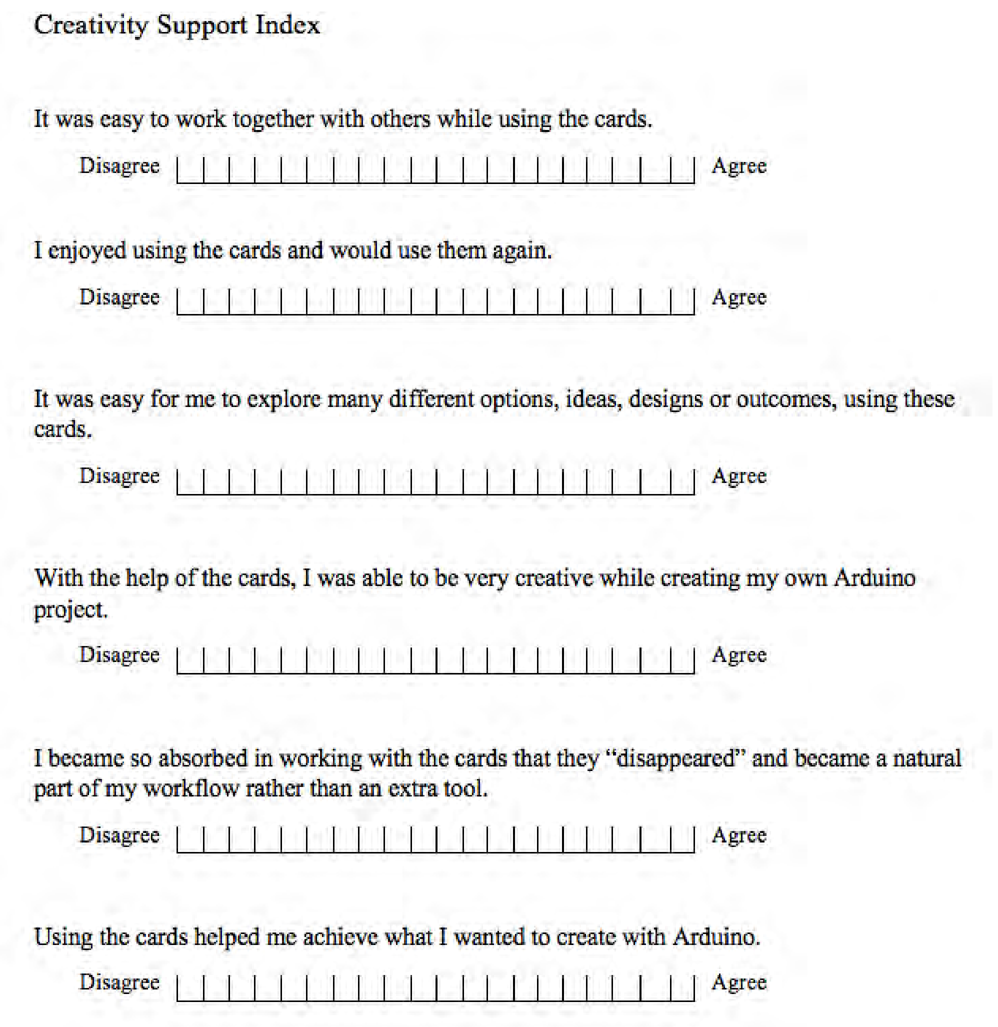
Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, *22*(2), 469–495. https://doi.org/10.1007/s10639-016-9482-0

Trust, T., Maloy, R. W., & Edwards, S. (2018). Learning through Making: Emerging and Expanding Designs for College Classes. *TechTrends*, *62*(1), 19–28. https://doi.org/10.1007/s11528-017-0214-0

Vihavainen, A., Paksula, M., & Luukkainen, M. (2011). Extreme apprenticeship method in teaching programming for beginners. In *Proceedings of the 42nd ACM technical symposium on Computer science education - SIGCSE '11* (p. 93). New York, New York, USA: ACM Press. https://doi.org/10.1145/1953163.1953196

Appendix

A. First part of the CSI



Creativity Support Index

It was easy to work together with others while using the cards.

Disagree |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Agree

I enjoyed using the cards and would use them again.

Disagree |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Agree

It was easy for me to explore many different options, ideas, designs or outcomes, using these cards.

Disagree |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Agree

With the help of the cards, I was able to be very creative while creating my own Arduino project.

Disagree |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Agree

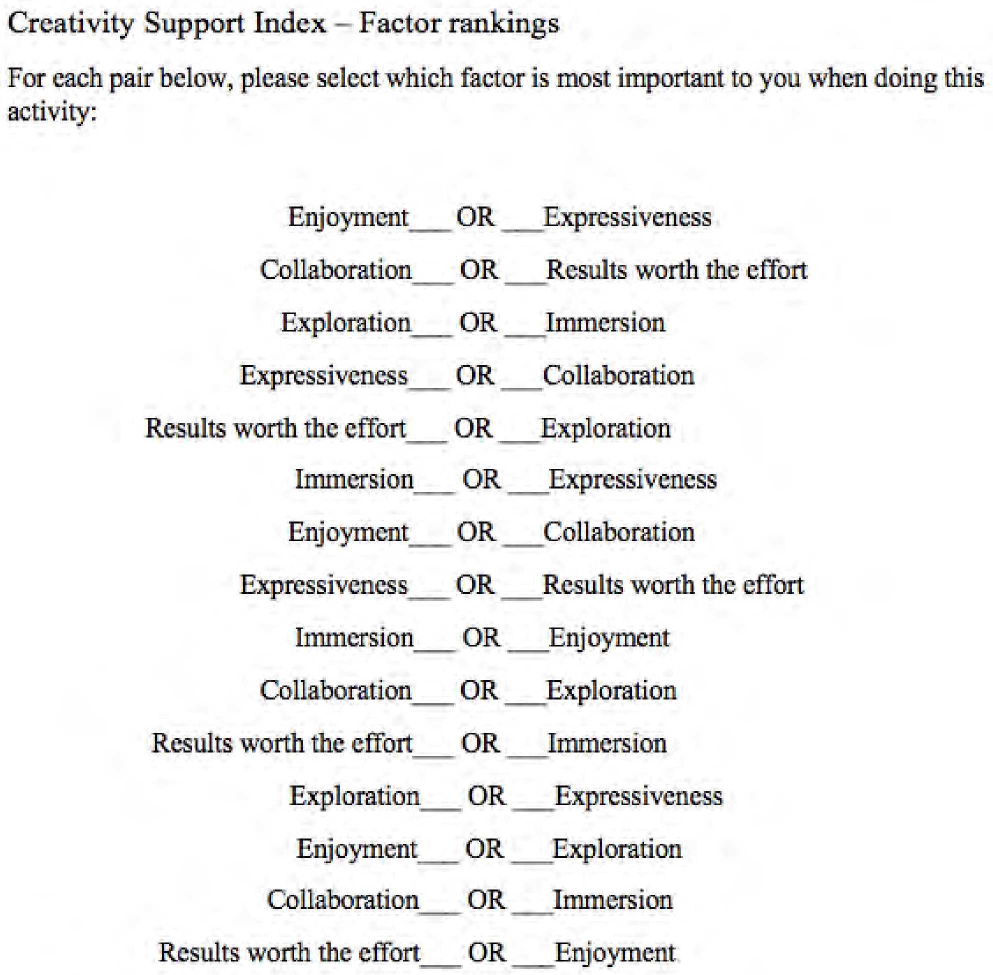
I became so absorbed in working with the cards that they "disappeared" and became a natural part of my workflow rather than an extra tool.

Disagree |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Agree

Using the cards helped me achieve what I wanted to create with Arduino.

Disagree |_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_|_| Agree

B. Second part of the CSI



Creativity Support Index – Factor rankings

For each pair below, please select which factor is most important to you when doing this activity:

Enjoyment___ OR ___Expressiveness
Collaboration___ OR ___Results worth the effort
Exploration___ OR ___Immersion
Expressiveness___ OR ___Collaboration
Results worth the effort___ OR ___Exploration
Immersion___ OR ___Expressiveness
Enjoyment___ OR ___Collaboration
Expressiveness___ OR ___Results worth the effort
Immersion___ OR ___Enjoyment
Collaboration___ OR ___Exploration
Results worth the effort___ OR ___Immersion
Exploration___ OR ___Expressiveness
Enjoyment___ OR ___Exploration
Collaboration___ OR ___Immersion
Results worth the effort___ OR ___Enjoyment