Scratching the surface of digital literacy... but we need to go deeper

Jon Bustillo
School of Education
University of the Basque Country
Vitoria-Gasteiz, Spain.
jon.bustillo@ehu.es

Pablo Garaizar
Deusto Institute of Technology (DeustoTech)
University of Deusto
Bilbao, Spain.
garaizar@deusto.es

***Abstract*—The popularization of digital educational devices with low barriers to entry has encouraged the development of many training activities oriented towards the incorporation of technology in schools. In some cases, the introduction of technology has led to the development of new educational practices that support the development of computational thinking. However, the supposed benefits of these approaches have not been properly assessed. Moreover, having taught over 30 workshops with the Scratch programming tool to teachers at different educational levels (primary, secondary, university), we found no evidence of subsequent methodological changes in schools. This study tries to understand the dissonance between the alleged success of initiatives around digital literacy and the lack of continuity in the use of user-friendly programming environments like Scratch. For this purpose, we analyzed the evolution of the grades of Scratch programming achieved by the students of the School of Education of Vitoria-Gasteiz and their engagement with Scratch. We also interviewed teachers from different schools who have participated in Scratch workshops with us. After this study, we identified some of the circumstances that facilitate and hinder the development of computational thinking. Since we consider Scratch as a resource that allows the development of new methodological approaches in the classroom as well as the acquisition of skills related to computational thinking, we propose a framework that will help to overcome the current status.**

Keywords*—computational thinking; digital literacy; education; programming; Scratch

I. Introduction

Despite the uncertainty caused by the dot-com crash in 2001, the transformative -and sometimes disruptor- power of the technology over is evident the past decade in many different areas. This technological tsunami has swept very well established industries such as the music industry (e.g., Spotify, iTunes), television (e.g., Youtube, Netflix), journalism (e.g., Twitter, The Huffington Post), or even the financial sector, where thousands of intelligent agents exchange millions of shares in milliseconds. This undeniable success of technology as a driver of change encouraged analysts, managers and politicians to apply the same recipe for success to other fields such as health or education, with less immediate results.

In regards of education, there are many and varied approaches that try to take advantage from technological advances to improve learning. Beyond the proposals with a strong emphasis on technology (e.g,, Mobile Learning [1]), as those who seek to exploit successful mechanical technology in other areas (e.g., Gamification [2]), there are those who advocate the need for a new digital literacy [3], with a dual purpose: 1) Better understand the technologized world in which we live, and 2) Build more and better technologies that improve learning.

Scratch is a project of the Lifelong Kindergarten Group at MIT Media Lab [4] that provides both a programming language and online community to create interactive stories, games, animations, and simulations that can be easily shared online. The goal of Scratch [5] is to promote to think creatively, reason systematically, and work collaboratively, while also learning important mathematical and computational ideas. Unlike other initiatives for teaching computational thinking such as Code.org or Codecademy, Scratch proposes an open and fully configurable environment in which the goal is to not only learn how to program, but to develop many other skills to achieve real digital literacy. Despite the growing popularity of Scratch [6] and alleged success of initiatives that promote digital literacy, we have noticed that teachers and students have a limited, immediate, and concrete vision of using Scratch (e.g., a step-by-step guide to program a video game during a semester), instead of realizing the cross-curricular potential of computational thinking (e.g., an unguided activity to design and develop a game that works on emotions).

II. Analysys

Through more than 30 training activities that we have carried out on Scratch, we have seen that most people who attended have a vague idea of what is Scratch and how can be integrated into classroom dynamics within schools. They tend to focus on the development of programs and not the potential for the development of basic competences by students.

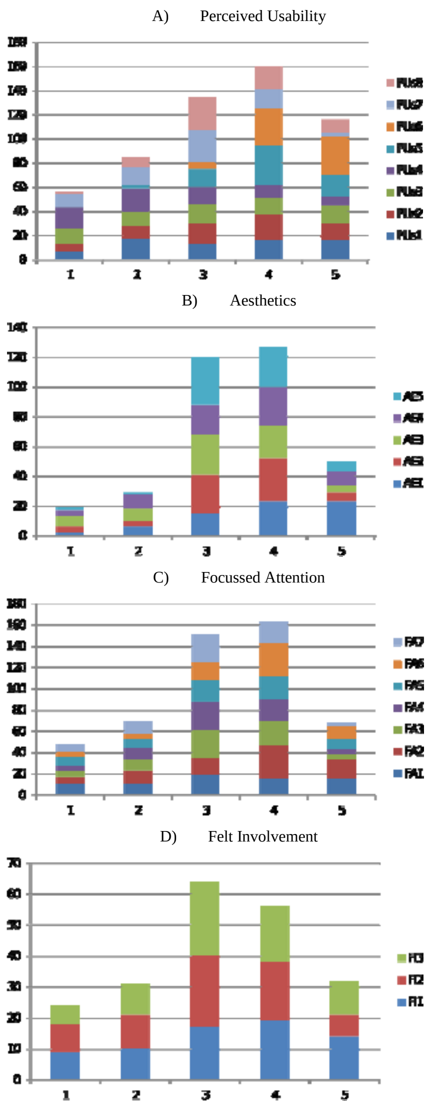
A. *Grades evolution and Scratch engagement at the School of Education*

During the first two courses where Scratch was taught as a mandatory activity at the School of Education of Vitoria-Gasteiz, the results were fair from acceptable. For this reason, we suggested to change the methodology to revert this situation. However, the results of the 2013-2014 school year did not improve as much as expected (see Table 1).

TABLE I. DISTRIBUTION OF GRADES OF THE STUDENTS OF THE SCHOOL OF EDUCATION OF VITORIA-GASTEIZ

School year	N	Grades			
		NR	C	B	A
2011-2012	93	42%	29%	19%	9%
2012-2013	76	30%	39%	27%	4%
2013-2014	109	36%	23%	37%	4%

The objective of the new training plan was to promote that students experience their ability to create their own games, letting them to define the topic and the level of complexity of their designs. To this end, we offered the students enough tools and examples to become familiar with the working environment and understand programming through the use of Scratch commands.



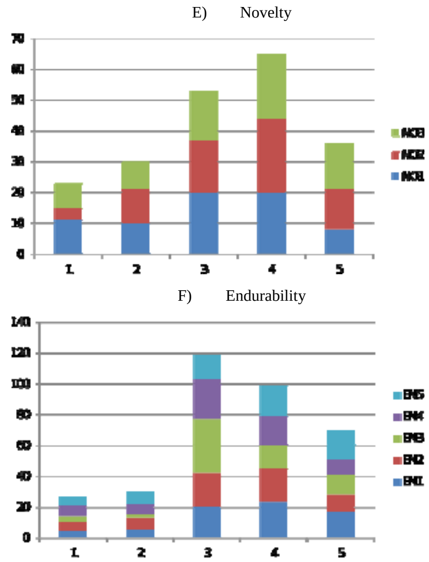


Fig. 1. Responses of the participants in the engagement scale (N=69). Focused Attention (FA): (1) I lost myself programming in Scratch; (2) I was so involved in my programming in Scratch task that I lost track of time; (3) I blocked out things around me when I was programming in Scratch; (4) When I was programming in Scratch, I lost track of the world around me; (5) The time I spent programming in Scratch just slipped away; (6) I was absorbed in my programming in Scratch task; (7) During this programming in Scratch experience I let myself go. Perceived Usability (PUs): (1) I felt frustrated while programming in Scratch; (2) I found programming in Scratch confusing to use; (3) I felt annoyed while programming in Scratch; (4) I felt discouraged while programming in Scratch; (5) Programming in Scratch was mentally taxing; (6) Programming in Scratch was demanding; (7) I felt in control of my programming in Scratch experience; (8) I could not do some of the things I needed to do programming in Scratch. Aesthetics (AE): (1) Scratch is attractive; (2) Scratch was aesthetically appealing; (3) I liked the graphics and images used on Scratch; (4) Scratch appealed to my visual senses; (5) The screen layout of Scratch was visually pleasing. Endurability (EN): (1) Programming in Scratch was worthwhile; (2) I consider my Scratch experience a success; (3) Programming in Scratch did not work out as I had planned; (4) Programming in Scratch was rewarding; (5) I would recommend programming in Scratch to my friends and family. Novelty (NO): (1) I continued to programm in Scratch out of curiosity; (2) The content of Scratch incited my curiosity; (3) I felt interested in programming in Scratch. Felt Involvement (FI): (1) I was really drawn into my programming in Scratch task; (2) I felt involved in this programming in Scratch task; (3) This programming in Scratch experience was fun.

The evaluation process was agreed initially with the students and was focused on the creative process, giving more weight to the effort to improve rather than the project completion. Its aim was to observe the ability of each student to address the development of her own productions guided by the initial idea, all within the spiral of the creative process described by Resnick [7]. In addition to this training plan, we evaluated their engagement with Scratch adapting a previously developed multidimensional scale to measure user engagement [8]. The responses of the 69 participants are summarized in Fig 1. We found no significant difference between participants of the 2013-2014 school year (N=34) and previous school years (N=35). We also organized a discussion group with 5 students of School of Education (ages: 20-25). As emerges from the views of participants in the discussion groups, previous educational references condition the process of learning to use

Scratch. In the case of the students of the School of Education of Vitoria-Gasteiz, these are basically three: 1) The experience as a student in a place where they had no opportunity to access to educational resources to learn how to program; 2) The traditional formative methodology received in the School of Education; and 3) The training practices carried out in schools. In none of these cases is common for these students to find references to Scratch or the impact Scratch may have on the development of basic competences.

In relation to their training as future teachers, they report that working with Scratch is a unique experience. As one student explains, *"In my opinion, we are not used [to this type of learning]; Scratch is something totally different. What I said before: we have several classes a day where they ask us all the same. Except for your [Scratch] class, where everything is different."* The methodology used for learning Scratch is totally different from the other disciplines that students in the School of Education have and this results in a lack of habit. They have to rely on the alleged virtues of Scratch as a teaching resource. Similarly, when these students take internships in schools, there are not examples to help them understand how resources like Scratch can be useful to promote the acquisition of basic competences. Students said it would be helpful to learn about best practices in classroom using Scratch: *"If we see a child doing something that improves her logic, if you tell us about it, we'll believe it, I guess. I think that way, yes. If you tell me that doing this or that is proven that children can develop some competences, I will believe it and see Scratch as a valuable resource."*

All students in the discussion group believe they may be able to acquire the necessary skills to use Scratch. However, they claim that it is necessary to go the extra mile, not easy without the necessary motivation (or need). Likewise, they think that a learning process that does not provide a clear and unambiguous work plan is not suitable. They prefer a learning process that is highly targeted and not open. A methodology that is more focused on the implementation and not the process: *"[...] but the problem is that we are not used to this. We have always had, since childhood, a teacher that said... you have to do this, this and this. As we have not worked autonomously, we find very difficult to get at the computer and... let's go... imagine and create a project, and start doing things without knowing anything. Therefore, we find it more difficult. Maybe if you work that way from childhood... then yes."* Not having a step by step guide involves being in a loop of imagine > define > test > program > share where the most important is the process, which it is very tiring for them. This differs from a model more targeted, focused and guided by the product in a read > understand > organize > schedule > check cycle, which is much more familiar for them. In this sense, the errors are not experienced by students as an opportunity to learn, but as something that slows the completion of their products: *"Complete the project in the shortest time possible ... in this case, each error is a pain because it causes us to spend more time without noticeable progress."*

It is remarkable that student teachers in the discussion group have not been able to identify what competences they worked using Scratch. Consequently, they are unable to understand or transfer the Scratch potential as a teaching resource for the development of basic competences: *"That is something I am not able to see. When I'm using Scratch, I look patiently to the computer and check if things move properly and if it worked. But I do not try to understand what I'm cognitively improving. We do not know what is improved using Scratch."* Although students of the School of Education do not see Scratch as a valuable teaching resource, and no one thinks about using it in the future, they all agreed about the attractiveness that digital devices have for children today. They also mentioned the need of a methodological change in schools. Despite their lack of confidence addressing that change, they are aware of the need of a new set of technologies, different from the ones they learned in their schools: *"Technologies have been introduced in the classroom, but they are used in the same way. For example, a digital whiteboard is used as a traditional whiteboard. Introducing technology in this way makes no sense."*

B. *Interviews with teachers*

Following a discussion meeting with 18 teachers (13 female / 5 male, ages 40 to 63) that currently work as counselors in the Center of ICT Innovation in Education of the Basque Country, 78% indicated that the presence of Scratch in the classroom was very low despite the several training workshops conducted. Similarly, the remaining 22% indicated not having records of its use in schools. During the discussion, the reasons for the lack of presence of Scratch in schools were grouped into three blocks: 1) Training, 2) Methodology, and 3) Institutional support. First, there is a need for more training for teachers to master the tool. Second, teachers need for a guide (textbook) to ensure proper use in the classroom as well as a justification of the competences developed by the students. Finally, teachers need support from the managers of the schools and the educational administration. Only 21% of participants of the discussion group believe that teachers are able to use Scratch properly in the classroom. Another 21% of them think it will depend on the particular circumstances of these teachers (e.g., technological proficiency, support from the manager of the school, and so on). In contrast, 58% of them do not believe that teachers will incorporate Scratch in their teaching in the short term.

III. A NEW FRAMEWORK FOR THE USE OF SCRATCH IN THE CLASSROOM

Through the various surveys, interviews and focus groups conducted with both teachers and students of the School of Education, we identified three vectors influencing the presence of Scratch in the classroom: 1) The acquisition of sufficient technical skills in using Scratch by teachers (i.e., knowing the capabilities of the tool and having a wide range of examples that could act as a guide); 2) The need for a trusted environment (a term already used by Computer Clubhouses [9]) that allows integrating educational resources such as Scratch in teaching (to do this, it would be helpful to have institutional support, time to develop materials, an implementation guide with examples of use in the classroom and evidence that working with Scratch promotes skills acquisition); and 3) The motivation and engagement of teachers to use teaching resources that allow students to create their own content. Currently, most of the training activities have been oriented primarily toward the first vector (Technical skills). The other two (Trusted environment and Engagement) have received less attention. This imbalance has resulted in a very low presence of Scratch in the classroom. Future training activities should include examples of good educational practices and guidelines using Scratch in the classroom. This would allow many teachers incorporate Scratch in teaching, overcoming fears and doubts described in the previous section.

IV. Discussion

The ultimate goal of promoting programming in schools is not getting more students into Computer Science, but enhancing the formal education through computational thinking. However, this is not a simple task and involves a variety of cognitive activities and mental representations related to program design, program understanding, modifying and debugging [10]. Despite their benefits in problem solving, creative thinking, logical reasoning and systematic experimentation [9], these digital skills tend to be forgotten by Schools of Education in their curricula.

Incorporating programming to K-12 classrooms is a very important challenge. As Jacobsen, Clifford & Friesen stated, it's not easy and "requires imagination, intellect, creativity and no small courage", especially in our dominant curriculum models [11]. The greatest difficulty lies not in learning how to use a particular tool [12], but finding experiences that encourage observing, question, interpreting and testing ideas in the classroom [13]. Most of the teachers interviewed evaluated Scratch positively [14-17]. However, they also noted that in addition to learning to use Scratch, it was equally important to know best practices of using Scratch at schools [18], having learning guides, and developing a new curriculum model. After this study, we identified some of the circumstances that can facilitate and hinder the development of computational thinking through Scratch. In agreement with previous works [19], we have found that when student teachers have the opportunity to see the incidence of Scratch in primary school children during their training periods, their perspective changed dramatically. Subsequently, they show enthusiasm and interest in introducing classroom teaching resources as Scratch, regardless of their level of competence in using them [20].

The training efforts for integrating Scratch in the classroom have traditionally focused on acquiring technical skills and provide a brief introduction to programming. Instead, teachers also need to have a wide set of best practices that explain how this type of learning resources can help their students acquire the skills covered in the curriculum. We believe that, without these basic premises, is very difficult to go deeper in the use of Scratch, which is currently focused on the production of video games through a guided model that leaves no room for the creative development of students. This change requires proficiency in competence acquisition and the evaluation of processes versus products, considering errors as learning enablers and not as an avoidable waste of time [21]. Scratch is a great educational tool for the development of basic skills. This is necessary to allow students to develop their own productions, which also serve teachers as indicators to assess the competence development of students [22].

References

[1] Kearney M. et al. 2012. *Viewing mobile learning from a pedagogical perspective*, Routledge, vol. 20, (3), pp. 1-17.

[2] Kapp, K. M. 2012. The gamification of learning and instruction: game-based methods and strategies for training and education. San Francisco: John Wiley & Sons.

[3] Casey, L. and Bruce, B. C. 2011. *The practice profile of inquiry: Connecting digital literacy and pedagogy*. E-Learning and Digital Media, vol. 8 (1), pp. 76-85.

[4] http://llk.media.mit.edu/ Accessed: 20 April 2014.

[5] Maloney, J. et al. 2004. *Scratch: a sneak preview.* Proceedings. Second International Conference on Creating, Connecting and Collaborating through Computing, pp. 104-109.

[6] Scratch statistics. http://scratch.mit.edu/statistics/ Accessed: 20 April 2014.

[7] Resnick, M. 2007. *Sowing the seeds for a more creative society*. Learning and Leading with Technology, vol. 35(4), pp. 18–22.

[8] O'Brien, H.L. & Toms, E.G. 2010. *The development and evaluation of a survey to measure user engagement*. Journal of the American Society for Information Science & Technology, vol. 6(1), pp. 50-69. doi: 10.1002/asi.21229.

[9] Kafai Y., Peppler K. and Chapman R. 2009. *The Computer Clubhouse*, New York: Teachers College Press.

[10] Rogalski, J., & Samurçay, R. 1990. *Acquisition of programming knowledge and skills*. Psychology of Programming, 18, 157–174.

[11] Jacobsen, M., Clifford, P. & Friesen, S. 2002. *Preparing Teachers for Technology Integration: Creating a Culture of Inquiry in the Context of Use.* Contemporary Issues in Technology and Teacher Education, vol. 2, no. 3, pp. 363–388. Norfolk, VA: AACE

[12] Ahern, T.C. 2009. *Bridging the gap: cognitive scaffolding to improve computer programming for middle school teachers.* Presented at the Frontiers in Education Conference, 2009. FIE'09. 39th IEEE, pp. 1–5.

[13] Robins, A. Rountree, J. & Rountree, N. 2003. *Learning and teaching programming: A review and discussion.* Computer Science Education, vol. 13, no. 2, pp. 137–172.

[14] Fesakis G, Serafeim K. 2009. *Influence of the familiarization with "scratch" on future teachers' opinions and attitudes about programming and ICT in education*. ACM SIGCSE Bulletin 25;41(3).

[15] Clark, J., Rogers, M. and Spradling, C. 2011. *Scratch the workshop and its implications on our world of computing.* Journal of Computing Sciences in Colleges, vol. 26, no. 5, pp. 235–243.

[16] Repenning, A. 2013. *Making Programming Accessible and Exciting.* IEEE Computer Society, vol. 46, no. 6, pp. 78–81.

[17] Wolz, U., Leitner, H.H., Malan, D.J. and J. Maloney, J. 2009. *Starting with scratch in CS 1.* Presented at the ACM SIGCSE Bulletin, vol. 41, pp. 2–3.

[18] O. Meerbaum-Salant, O., Armoni, M. and Ben-Ari, M. 2013. *Learning computer science concepts with scratch.* Computer Science Education, vol. 23, no. 3, pp. 239–264, 2013.

[19] Ertmer, P. A. & Ottenbreit-Leftwich, A. T. 2010. *Teacher Technology Change: How Knowledge, Confidence, Beliefs, and Culture Intersect.* Journal of Research on Technology in Education, vol. 42, no. 3, pp. 255–284.

[20] Ertmer, P.A. 2005. *Teacher pedagogical beliefs: The final frontier in our quest for technology integration?* Educational technology research and development, vol. 53, no. 4, pp. 25–39.

[21] Bustillo, J. 2014. Análisis del pensamiento creativo en internos del centro penitenciario de Álava que participan en experiencias formativas centradas en Scratch. Vitoria-Gasteiz. Ph. D. Dissertation. Basque Country University

[22] Brennan, K. A. 2013. *Best of both worlds: issues of structure and agency in computational creation, in and out of school*. Massachusetts. Ph. D. Dissertation: MIT Media Lab. Retrieved from http://web.media.mit.edu/~kbrennan/files/dissertation/Brennan_Dissertation.pdf