

Teaching Game Design and Basic Computing Concepts: a democratic experiment in a Socioeconomically Vulnerable Community

Ermelindo P. B. Schultz¹, Júlia S. B. Ortiz¹, Laura Sanchez García¹, Roberto Pereira¹

¹Departamento de Informática – Universidade Federal do Paraná (UFPR)
Curitiba – PR – Brasil

ermelindo.schultz@gmail.com

Abstract. *Computational Thinking has been promising for inclusive education in Computing. This paper presents a game programming course planned and conducted with Brazilian young people in a socioeconomically vulnerable community. In this non-formal course, nine workshops were designed and conducted considering participants' language and social context. Workshops were grounded on a structural framework based on Paulo Freire's method and their practices were inspired by techniques from Participatory Design. The paper presents the theoretical and methodological grounds adopted for the course, the structure designed for the workshops, the main observations and main learned lessons from the experience. Results suggest the course was successful to provide a meaningful education in basic computing concepts, engaging students throughout the workshops and reducing students dropout.*

1. Introduction

Algorithms, computers, programming, games and artificial intelligence have become everyday words in people's lives. People do not use computing, people live with computing. "Computers (and what "runs inside") shape everything in our society, from individual's social life to political and economic aspects"¹.

Many initiatives towards computing education for all have been taken to promote computational thinking, computer programming and computational literacy around the world. Recently, some countries, such as New Zealand, Denmark, and England, implemented computing curriculum at schools [Guzdial 2015].

A survey on Pre-College Computer Science Education carried out by Google and Stanford University identified equitable participation as one of the main challenges to promote computing education [Blikstein 2018]. The survey points out to equity and inclusiveness as critical issues to advancing both computer science education and teacher development. For equitable participation, the survey means improving participation of all students regardless background, motivations, preparations and abilities.

The survey [Blikstein 2018] highlights demand for computing skills is overgrowing not only for economic reasons but in all aspects of children's lives, and that preparing all students for the future requires institutions and mechanisms to provide learning opportunities for them. The survey claims that, if we are to overcome biases and structural

¹Sentence from a brainstorming activity with students engaged in participatory practices presented in this paper.

inequalities that prevent equitable participation, computer science should be mandatory content in public schools. Otherwise, computing will remain an elective or specialty subject, unequally present in public schools, and accessible (or suitable) for a narrower audience.

In a challenging context like the Brazilian one, structural inequalities directly affect formal education priorities and take the educational challenge to an extreme level of importance. Beyond curricula and technical resources for teaching and learning, other challenges from the social life come into play such as violence inside and outside schools, poverty, low wages and unhealthy work environments for education professionals, just to name a few. Beyond children education, people education must be the focus: from children to elderly, unequal access to education have produced social exclusion and underdevelopment conditions. Beyond formal education, informal and non-formal education must help to address the challenges formal education is not able to solve in isolation. Therefore, it is necessary to find means to provide people access to computing knowledge. Without an official agenda for computing education and having poor structured schools, non-formal initiatives are even more crucial to promoting computing education outside schools.

As a social service project in the Federal University of Paraná, we have planned workshops to teach basic computing concepts, open for all interested people, with practices that engage them and make sense to their social, cultural and economic realities. In one of the initiatives, we offered nine workshops intended to promote participants' understanding of basic programming concepts and computational thinking. The workshops were offered in an open course for game design in a socially vulnerable community. The course was conceived considering participants' knowledge, language and social context as primary values.

In this paper, we present the theoretical and methodological grounds adopted for the course, the structure designed for the workshops, the main observations and learned lessons from the practical experience. Results suggest the course was successful to provide a meaningful computing education, engaging students throughout the workshops and reducing dropout. This experience shows us that listening to people and acting with them, sharing experiences, learning and doing together, are useful mindsets and attributes for educational practices.

2. Context and Methodology

The course took place in a local government building called "Youth Center" (YC). YC's objective is to provide local youth, aged 12-29 and in socially vulnerable situation, access to culture, art, sports, and opportunities for professional, personal and political development [IPARDES 2011]. YC has a suitable structure, with sports court, computer lab, library, classrooms and auditorium. However, according to YC manager, courses and activities usually have high dropout rate.

The course had 18 registered students. In the first workshop, 13 students appeared. In the second workshop, 2 new students appeared, and other 2 students drop out. The 13 remaining students from the second workshop were considered for the analysis presented in this paper: 10 boys and 3 girls, aged between 11 and 20 (the average age is 15,6). Only one student had no previous experience with technology until the beginning of the course.

Nine students finished the course. Two students had to leave the course because of personal issues. Other two students left the course leaving no information about the possible reasons. The dropout was considered low by YC's staff, contradicting their expectation.

The course structure was inspired by Paulo Freire's method. Freire was a Brazilian philosopher and educator with a notable influence in critical pedagogy and participatory action research. He developed an adult literacy method while involved with workers who lived in poverty condition [Schugurensky 2014]. His method starts by considering learners vocabulary and reality to construct a contextualized literacy process. Freire's method informed the methodology adopted for the course, which recognized the philosophy, principles and objectives of the method. In this case, we did not explore the possible relationships between critical literacy and programming.

The three stages from Freire's method were adopted (investigation, thematization, problematization [Feitosa 1999]) and a fourth stage named "Construction" was included for structuring the course. The main stages were organized as follows:

Investigation stage: students, and facilitators talked about their preferences, visions, and customs. The objective was to learn about students realities and expectations by listening to them and planning with them.

Thematization stage: students and facilitators debated and chose problems from their reality for which a computational artifact could mediate in a positive way. Inspired by "generating words" in Freire's method, these problems are called "generating problems" in this paper.

Problematization stage: students and facilitators idealized a computational artifact that addressed one or more generating problems. Students and facilitators reflected on how the idealized artifact should be represented by a computer, the ways the idealized artifact could impact people's lives and about the generating problems themselves.

Construction stage: students built the computational artifact idealized in the previous stage, and it was expected students to develop computing knowledge. Papert's constructionism influenced this stage: when children are building something in the world, they construct knowledge simultaneously in their heads [Falbel 1993].

Workshops covered one or more stages, being designed inspired by Participatory Design (PD) methodological perspective. PD refers to a 40 years design tradition where things are made **with** people. PD practitioners developed a set of practices, tools, and methods to facilitate mutual learning and people participation in design processes [DiSalvo et al. 2017].

After each workshop, facilitators wrote a report based on results, materials, and commentaries of the students, serving as input for the preparation of the next workshops. When the course finished, we obtained students perceptions via an open feedback form and discussed with them. A "Closing Ceremony" with students' families, YC staff and local authorities was organized by the YC to socialize the course results.

3. Related Work

Understanding Computer Science as a cultural product and observing that computing education is almost eurocentric [Tedre et al. 2006], Culturally Responsive Computing Educa-

tion (CRCE) explores problems and solutions using traditional knowledge or practices of the group being educated [Eglash et al. 2013]. The results of CRCE studies have shown benefits of this perspective in computing education of underrepresented groups, such as indigenous people, women and other minorities ².

Buckley and his colleagues present the concept of Socially Relevant Computing (SRC), where Computer Science is presented as a discipline that empowers students to solve problems of their interests and from society in large-scale [Buckley et al. 2008]. To illustrate this concept, they describe two projects where students created software/hardware to solve societal issues. Results suggested the potential of this approach to improve engagement in undergraduate Computer Science courses.

Bringing the SRC perspective to middle-school, authors of [Ni et al. 2016] report the outcomes of a 5-day summer camp in which 72 students created apps to address needs of their community using the MIT AppInventor. The outcomes showed that the confidence about creating apps and the interest in learning computer science grew significantly among students after the experience.

Both approaches (SRC and CRCE) can be mapped into the one we designed for our project: when students worked on problems from their reality (problems they chose to face and work on), they worked on socially relevant problems - a SRC approach. The descriptions of [Buckley et al. 2008] and [Ni et al. 2016] do not mention how students could select problems they would work on, and do not present or discuss a structure or approach to engage students in reflecting about what they were building. We address the first issue conducting participatory practices with the youth and the second issue adopting a dialogical method of teaching for the entire course.

When workshops and activities were designed based on language and context, they were approaching problems considering students' knowledge and practices - a CRCE approach. A meaningful education has been discussed in literature, such as Freire, who investigated and worked with people in political and societal contexts. A Freirean approach allows students to reflect critically about the solutions constructed during the course and about interests underlying them.

4. Workshops

In a Participatory style, 9 workshops were planned and conducted from September to December 2017, covering one or more stages of Freire's method – table 1 shows an overview of these workshops. During the workshops, students sat down around facilitators (see Figure 1), then they interrupted facilitators at any time with any question or new idea. This structure was based in culture circles. Active learning strategies ³ were applied to motivate students' participation and critical thinking about the workshop subject matter.

Workshop 1 (W1): *Course presentation*. Facilitators and students debated about how computing affects society. Social media, location systems and games were some of the topics of this debate. In the following, an activity mixing a storytelling workshop practice (Practice 11.10.55 [Muller et al. 1997]) and a Brazilian Hot Potato game

²A list of selected publications with some of these results could be find at <https://csdt.rpi.edu/publications/>

³Resources about active learning at an open online course from MIT were used. The complete course could be find at <http://ocw.mit.edu/5-95JF15>

Workshop	Topic	Investigation	Thematization	Problematization	Construction
1	Course presentation	X	X		
2	Choosing themes for games	X	X		
3	Talk with a Game expert		X	X	
4	Decryption challenge			X	X
6	Variable concept and Bets			X	
5, 7, 8, 9	Hands-on			X	X

Table 1. Workshops overview.

was conducted with students. The objective was to engage students and facilitators in interaction as a cold-break activity. Students and facilitators talked about bad and good experiences with technology, as in storytelling practices. Before finishing the workshop, facilitators asked students about their expectations for the course. "Patience," "learning" and "learning something I will really use in my life" were their main answers.

Workshop 2 (W2): *Choosing themes for games*. As indicated by students in the previous workshop, they wanted to learn something useful for their lives. Thus, three videos were presented to students when starting W2. The videos provoked debates with students about three subjects: "is programming for anyone?"⁴, "programming as a path to professionalization"⁵, and "programming as a path to social change"⁶.

Students were grouped into four working groups (see Figure 2). Keeping in mind that designing games are a powerful way for social change, the primary objective of the workshop was each group to chose a social problem to work on during game design.



Figure 1. Dialogical Classroom setup in W1.



Figure 2. Groups distribution in W2.

The Group Elicitation Method (Practice 11.10.27 [Muller et al. 1997]) was adapted for the workshop. During 5 minutes, students wrote in a paper sheet all the problems they could think of. Then, each student discussed his/her motivations to chose each problem with all the colleagues from the same group. Students reached a consensus and decided the theme for their games: all groups chose violence, and one group chose laziness in addition to violence. After choosing generating problems, students idealized

⁴<https://youtu.be/dU1xS07N-FA>

⁵<https://bit.ly/2jR6UZr>

⁶<https://youtu.be/5U4sw6bUnwc>

their game, writing down their ideas. Finally, the Braindraw technique (Practice 11.10.5 [Muller et al. 1997]) was conducted with groups, and they could make their first draws for their storyboards. At the end of the workshop, each group presented their idealized game to the other groups and facilitators.

Workshop 3 (W3): *Talk with a Game expert*. A game designer who started creating games since his childhood was invited to talk with students in W3. He was invited because his history of life could inspire them. The Expert conducted the workshop, firstly talking about main elements in a game. Embodying course's philosophy, he kept dialogical classroom structure and asked for participants to find characteristics about games in the games they already played. For each characteristic told by a student, a discussion was made and the concept behind this characteristic was introduced. Finally, the expert taught the steps of game design process coming from his own experience.

Workshop 4 (W4): *Decryption challenge*. An university teacher, a Ph.D. student and a master's student from the facilitators' research group participated in the W4 activity. They have solid computer science background. Primarily, the concept of "algorithm" was built together with students, inspired by the expert's approach in W3. After, students and guests organized themselves into three teams (see Figure 3). Teams received blue, green or yellow ribbons as their representation colors. After solving two algorithm exercises, teams competed to decrypt two sentences: an easy one, like the exercises, and a hard one. Blue team won the easy challenge while the yellow team won the hard challenge.

Workshop 6 (W6): *Variable concept and Bets*. In W1, during the hot potato storytelling, one facilitator said he loved to play Bets (a game similar to Backyard Cricket Game) in his childhood, and students reacted positively and engaged in the discussion. Therefore, an activity based on the Bets game was prepared to introduce concepts of variables, after a hands-on activity in W5 where students were introduced to Scratch platform.

Bets match occurs with two teams with two players each. One team takes sticks and tries to hit the ball to score points while the other team throws the ball to hit an object in order to recover the sticks. The game has several rules that depend on neighborhood/district/city where the game is played. The game was played according to students' rules defined before the first round. Bets game could be used to introduce the variable concept due its score and gameplay style. In the applied rules there are three different kind of scores. In a hypothetical digital game, each score could be represented as an integer value. The values of each score change in different ways during the game. Such analogy was used to introduce the integer type, its purpose and applicability in programming and how to use it. Identifying whether a team holds or not the sticks was another example explored by the facilitators by explaining boolean variables.

Five students attended the workshop and five matches (or rounds) were played. Teams were randomly formed by facilitators and students. While two teams played, a facilitator introduced the variable concept to the team that was waiting, linking this content with Bets. At the end of the activity, students returned to the computer lab and applied recently learned concepts into their prototypes successfully.

Workshops 5, 7, 8 and 9 (W5 to W9): *Hands-on activities*. From W5 to W9, all workshops but W6 focused on game building activities. In W5, computer programming concepts and Scratch interface were presented. Then, students explored Scratch at com-



Figure 3. Participants engaged in the decrypt challenge.



Figure 4. Students and facilitators playing bets.

puter lab. In W7, W8, and W9 all workshop time was reserved for game programming, while in W6 one hour was reserved for this purpose after the Bets game.

Groups formed in W1 were dissolved, forming only two groups. Therefore, student's games themes changed and the new groups and facilitators decided to deliver a game's prototype by the ending of the the course. Students learned the concept of prototyping, practicing prototyping activities both in paper and in computing tools.

As results, a point-and-click game prototype was presented by the first group (see Figure 5), where a player must collect garbage correctly to fight pollution. Students decided to change the original game theme from "violence" to "pollution" because they saw point-and-click mechanics more appropriate for their ideas, and agreed the theme was socially relevant and important for their reality as well.

The other group presented a prototype of the adventure game "Agir" (in English, "to act"), see Figure 6. Students focused on the theme violence against women for the final prototype. Group 2 created their own characters' sprites and very detailed animations.

One week after the last workshop, students presented their final prototype to other students, parents, guests and YC's staff. Presentation format was a trailer created by one of the facilitators after students agreement ⁷.

In the last workshop and the final presentation, a questionnaire was applied to receive students perceptions about computing and about the course itself for further analysis. There were 5 open questions: Q1: In your opinion, why computing is important? Q2: Which workshop did you like the most? and Why? Q3: Cite three positive points in the course. Q4: Cite three negative points in the course. Q5: Cite something you have learned in the course.

Five students gave their feedback, as two participants did not come for the last workshop and final presentation, and other two participants came for the final presentation but had to leave before filling out the questionnaire. Students showed they know the importance of computing in society, but three of them could not put it in words on the

⁷Group 1 trailer: <https://youtu.be/dNxJRQnaWAo>.
Group 2 trailer: https://youtu.be/CyEcM_NcMNI



Figure 5. Group 1 prototype.



Figure 6. Group 2 prototype.

paper sheet. Students did not show consensus on which workshop they liked the most: two students chose practical classes, two students chose all the workshops, and one chose the Bets activity.

Q3 highlights that fun and new experiences were significant good points in the course for students and Q4 shows students' particularities. Despite a student stated fun, learning and experience as positive points, he suggested course content was hard to learn. Other student shared her disappointment with course dropouts, showing the importance of the efforts to reduce dropouts and reinforce the sense of belonging to a group. A student who had to move away from the city after course mentioned he would have liked to participate in more workshops.

In the last question, three students answered course's contents, and two left it blank. The two students that left the question blank were asked directly by facilitators and then they succeed in list some contents presented to them during the course, such as create games, variables and conditionals. The objective of this question was not to assess students knowledge, but identifying if they felt that they have learned something during the course.

5. Learned Lessons: A Reflection under Culture Circles Elements

In Schugurensky's biography about Freire [Schugurensky 2014] he describes the three elements of culture circles: facilitator, dialogical methodology and curriculum related to students' life. The results discussed in this paper considered these elements and are based both on dialogues registered during the workshops and the questionnaire applied to 5 participants.

Facilitator⁸ instead of a traditional teacher: In [Iversen et al. 2017], authors suggest a new role for children in PD practices: the protagonist role. Authors argue for children *as the main agents in driving the design process and thereby to develop skills to design and reflect on technology and its role in their life* in participatory practices [Iversen et al. 2017, p. 30].

In our approach, teachers acted as facilitators and students were the protagonists.

⁸Schugurensky's original word is *Coordinator*. *Facilitator* was used because it sounds less decentralized than the original word.

They chose their own games' topics and brought new tools and ideas without facilitators asking them for it. We could translate [Iversen et al. 2017] ideas to our context: students were the primary agents in driving the teaching-learning process. Additionally, they did not only learned about technology but reflected about its role in their lives as well.

Dialogical methodology instead of lecture-based classes: [Guzdial 2015, p. 36] suggests ways of valuing computing for students, such as *show students role models, show them they can be successful, show them how valuable it is and make it fun*. Instancing a dialogical method of teaching helped us to address all these aspects. For example, in the first workshop, we discovered that students want to learn something useful in their lives. Then, in the next workshop we brought videos showing three possible contributions and opportunities from programming valued in society and discussed each one with students. In the first workshop, we also discovered that Bets was a game that everyone had already played and had positive experiences with. Therefore, we designed a workshop associating computing concepts and this game, using the game experience not as a background or pretext to teach computing, but as the context where computing concepts made sense. Questionnaire answers and informal feedback during the workshops showed that the course was a fun experience, something that was possible only because we had discovered what is fun for students, listening to them and discussing with them, instead of starting the practices with a predefined set of practices and activities supposed to be fun and interesting for them.

As learned lessons from this experience, we identified the need for "socializing on-the-fly", where students must act together and share their results, discussing with each other. Moreover, we suggest adding "Evaluation" as a cross-cut stage to Investigation, Thematization, Problematization and Construction, where students evaluate their results and the results from their colleagues.

Curricula related to students' life instead of non contextualized and fragmented content: approaches presented in related work section already suggested the potentiality of contextualized approaches in computing education. Learners' needs, environments possibilities, and contents limitations were identified during the workshops and considered on-the-fly as the workshops took place. As results, a meaningful learning experience for students and facilitators was provided, were results from a workshop provided input and insights for another.

6. Conclusion

This paper presented and discussed the background, principles and the experience of organizing and conducting a non-formal game programming course. The course was designed in a way it allowed to understand participants language and context during the process, bringing elements and strategies relevant to their context of life, and making them part of the course planning and execution. PD methods, local's traditional games, and active learning strategies were mixed to help both students to create their prototypes and facilitators to understand students' reality. Inspired by the dialogical method of teaching, activities were conducted to promote more meaningful computing education with youth from a socially vulnerable community.

Young people from YC's context face many societal issues that influence the dropout rate at courses. According YC's manager, the dropout was considered low, sug-

gesting the course's approach could be a way to overcome the dropout rate issue. However, four dropouts out of twelve students is yet a troubling situation. Although two students reported personal reasons that no external action could have solved, the unknown reasons for the other two students should be taken into account for future experiences.

The approach adopted in this experience can inform computing educators from similar contexts to organize and put in practice culturally responsible and socially relevant activities in non-formal settings. Next steps in this project involve extending the course with new workshops to develop more complex concepts with the same participants, and offering new instances of the course for new groups of students, preparing a repertory of activities and practices to be investigated and developed with them.

References

- Blikstein, P. (2018). Pre-college computer science education: A survey of the field. Technical report, Mountain View, CA: Google LLC. Retrieved March 20, 2018 from <https://goo.gl/gmS1Vm>.
- Buckley, M., Nordlinger, J., and Subramanian, D. (2008). Socially relevant computing. In *ACM SIGCSE Bulletin*, volume 40, pages 347–351. ACM.
- DiSalvo, B., Yip, J., Bonsignore, E., and DiSalvo, C. (2017). *Participatory Design for Learning*. Routledge.
- Eglash, R., Gilbert, J. E., and Foster, E. (2013). Toward culturally responsive computing education. *Communications of the ACM*, 56(7):33–36.
- Falbel, A. (1993). Constructionism: Tools to build (and think) with. *LEGO Group, Denmark*.
- Feitosa, S. C. S. (1999). Método paulo freire-princípios e práticas de uma concepção popular de educação.
- Guzdial, M. (2015). Learner-centered design of computing education: Research on computing for everyone. *Synthesis Lectures on Human-Centered Informatics*, 8(6):1–165.
- IPARDES (2011). Programa centros da juventude : sumário executivo. Technical Report. Retrieved January 08, 2018 from www.ipardes.gov.br/biblioteca/docs/sumario_exec_centros_juventude.pdf.
- Iversen, O. S., Smith, R. C., and Dindler, C. (2017). Child as protagonist: Expanding the role of children in participatory design. In *Proceedings of the 2017 Conference on Interaction Design and Children*, pages 27–37. ACM.
- Muller, M. J., Haslwanter, J. H., and Dayton, T. (1997). Participatory practices in the software lifecycle. *Handbook of human-computer interaction*, 2:255–297.
- Ni, L., Schilder, D., Sherman, M., and Martin, F. (2016). Computing with a community focus: outcomes from an app inventor summer camp for middle school students. *Journal of Computing Sciences in Colleges*, 31(6):82–89.
- Schugurensky, D. (2014). *Paulo Freire*. Bloomsbury Publishing.
- Tedre, M., Sutinen, E., Kähkönen, E., and Kommers, P. (2006). Ethnocomputing: Ict in cultural and social context. *Communications of the ACM*, 49(1):126–130.