INCoD
Brazilian Institute for Digital Convergence

UNIVERSIDADE FEDERAL
DE SANTA CATARINA

ISSN 2236-5281

Technical Report
INCoD/GQS.04.2015.E

Design and Experiences with Teaching Physical Computing in Family Workshops

Authors:
Christiane Gresse von Wangenheim
Aldo von Wangenheim
Fernando S. Pacheco
Jean C. R. Hauck
Miriam Nathalie Fortuna Ferreira

Version 1.0
Status: Final
Distribution: External
December 2015

INCoD

Brazilian Institute for Digital Convergence

Federal University of Santa Catarina - UFSC
Campus Universitário João David Ferreira Lima - Trindade
Departamento de Informática e Estatística - Room 320
Florianópolis-SC - CEP 88040-970

Phone/FAX: +55 48 3721-9516 R.17
www.incod.ufsc.br

ISSN 2236-5281

Relatório Técnico do Instituto Nacional para Convergência Digital/ Departamento de Informática e Estatística, Centro Tecnológico, Universidade Federal de Santa Catarina. -- v.1, n.1 (2011).-- Florianópolis: INE, UFSC, 2011 -

Semestral
Resumo em português

ISSN 2236-5281

1. Convergência digital. 2. Tecnologia da informação. 3. Informática na saúde. 4. Mídia digital I. Universidade Federal de Santa Catarina. Departamento de Informática e Estatística.

Design and Experiences with Teaching Physical Computing in Family Workshops

Teaching computing to kids in workshops, camps, online tutorials or within their school context has become a trend in education. However it’s not only the children who should be taken into consideration. Parents, family and community members play significant roles in the lives of children and their involvement also in computing education is important for several reasons. Parents and families have the most direct and lasting impact on children's learning and education. When parents are informed and convinced of the importance of computing education, their motivation, assistance and investment (e.g., in books or computers) helps their children to achieve more, exhibit more positive attitudes and behavior and increase attendance and engagement. Involving parents in computer science education is even more important as they probably have little computing knowledge themselves. Therefore, computing programs for kids should also reach out to families in order to build the kind of relationships that inform and engage them as active partners in their children's education.

As part of our *Computing at School Initiative* (Computacaonaescola.ufsc.br, 2013), we run family workshops in which learning of basic computing capabilities is supplemented by parent-child activity time. During these workshops, children along with their parents learn physical computing by programming an interactive robot. Within the context of a story on an otter that needs to educate an ogre to stop polluting his lake, the workshop participants make to come alive a super hero robot by making him firing lasers through his eyes, throwing mud balls against a picture of an ogre and, once sensing the presence of the ogre, making a sound in order to draw its attention.

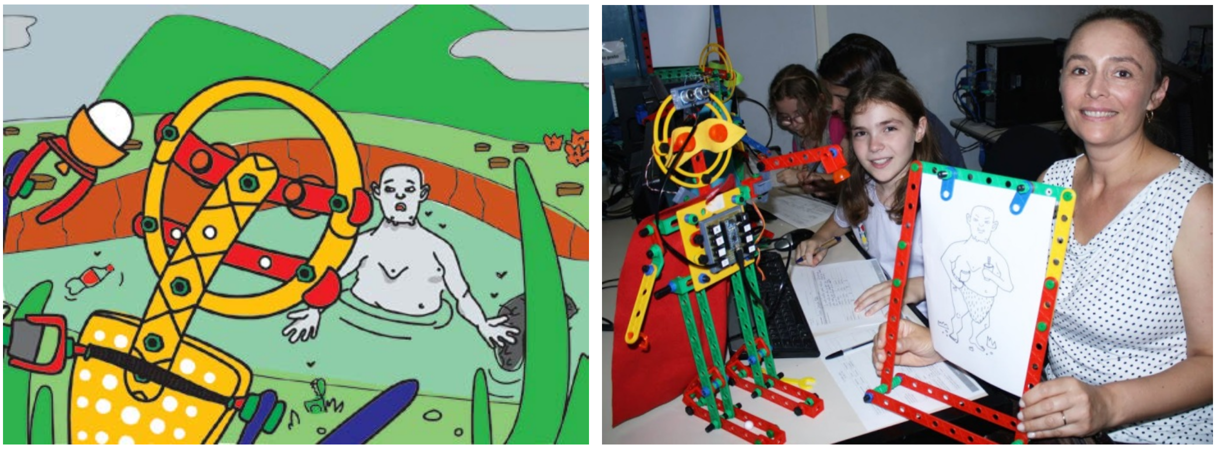
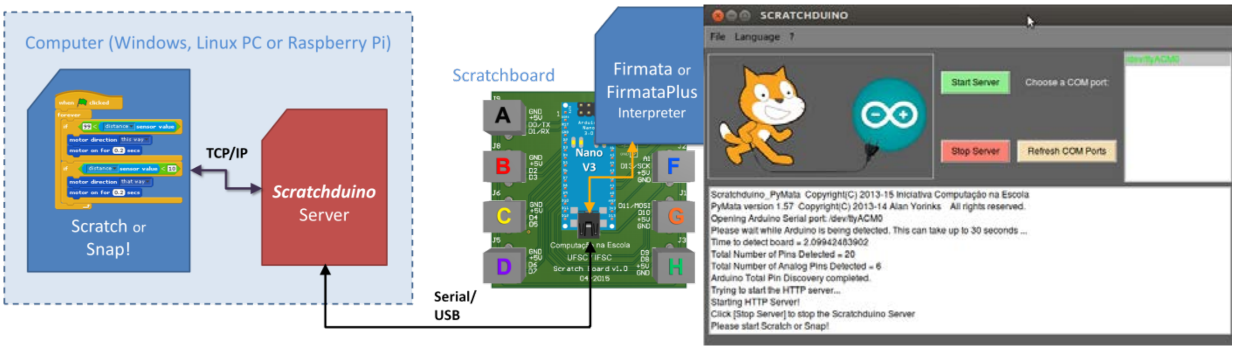


Figure 1. Scene form the “Ogre” narrative and kids enacting it with their programmed robots.

Therefore, we created a simple, low-cost, platform-independent, reliable and stable strategy integrating a microcontroller and a blocks-based visual programming language. The robot is automated by an Arduino Nano microcontroller and few hardware parts, such as a servo motor, an ultrasound sensor and two LEDs (with an approx. cost under US$45.00).

In order to facilitate programming, we use the Scratch blocks-oriented programming language (Malan and Leitner, 2007, Monroy-Hernández and Resnick. 2008), a free programming language developed at the MIT Media Lab especially for young people by simply using drag and drop blocks to perform different commands or actions. We support both most used flavors of Scratch: Scratch 2 (Scratch.mit.edu, 2007) and Snap! (Harvey and Mönig, 2015), which are chosen depending on the environment where the workshop is held. To enable a quick learning experience and not having kids to hassle with jumper wires, breadboards and command line-based microcontroller drivers, we developed a set of tools:

- **Scratchduino** (http://www.computacaonaescola.ufsc.br/scratchduino): a Scratch and Snap! to Arduino communication server. It is an easy to use, multiplatform GUI-based program that acts as a bidirectional protocol translator that provides interoperability between the Scratch Extension Protocol and the traditional Firmata microcontroller protocol. It allows driving microcontrollers using Scratch or Snap! programs in an intuitive, flexible and easy way.
- **Scratchboard** (http://www.computacaonaescola.ufsc.br/scratchboard): a low-cost and easy to use Arduino Nano break-out board that allows rapidly building physical devices using telephone cable jacks and helps visualizing the structure and logic of the connections.



```
Scratchduino_PyMata  Copyright(C) 2013-15 Iniciativa Computação na Escola
PyMata version 1.57  Copyright(C) 2013-14 Alan Yorinks    All rights reserved.
Opening Arduino Serial port: /dev/ttyACM0
Please wait while Arduino is being detected. This can take up to 30 seconds ...
Time to detect board = 2.09942483902
Total Number of Pins Detected = 20
Total Number of Analog Pins Detected = 6
Arduino Total Pin Discovery completed.
Trying to start the HTTP server...
Starting HTTP Server!
Click [Stop Server] to stop the Scratchduino Server
Please start Scratch or Snap!
```

Figure 2. Scratchduino/Scratchboard operating schema and Scratchduino GUI

Scratchduino is an extension of the *s2a_fm* command line Scratch to Firmata protocol translator originally developed by Yorinks (2013). In order to avoid the usage of an Unix-like command line application, we developed Scratchduino with a simplified GUI, usable by kids, in contact with both the original developers of s2a_fm and Snap! and also provided the Portuguese localization for the Scratch and Snap! Arduino Blocks and the Brazilian Portuguese localization for the whole Snap! programming language. This allowed us to employ Scratch and Snap! in physical computing family workshops in Brazil.

The Scratchboard was developed as a printed circuit board with a DIP socket for the Arduino Nano, and eight 4P4C jacks, where the automation devices can be connected, as shown in Fig.2. These can be analog and digital devices, from actuators and sensors to simple handmade analog devices.

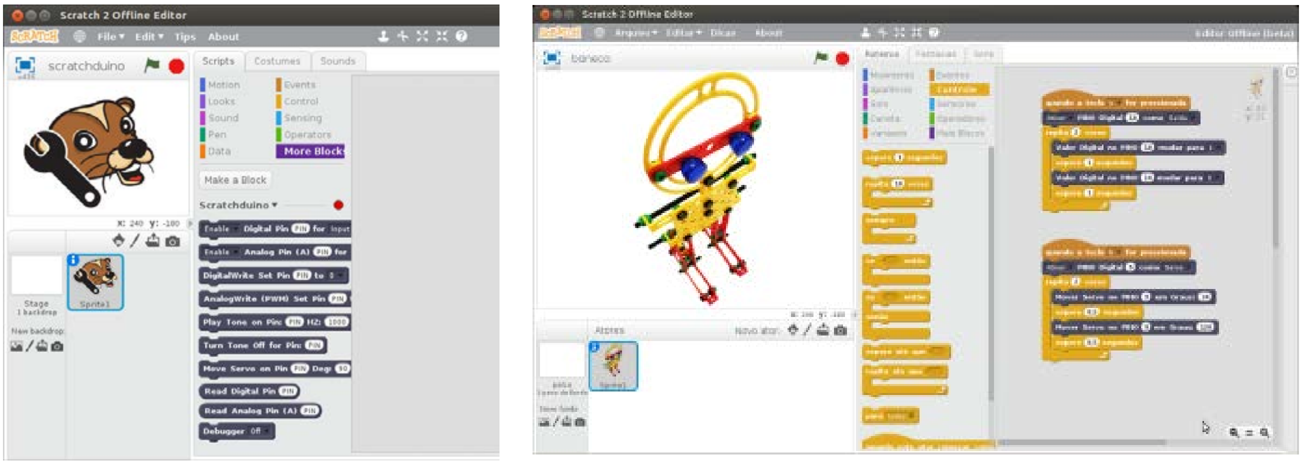


Figure 3. s2a_fm blocks used by Scratchduino and example of Scratch program for moving the robot’s arm

Running Family Workshops

In order to teach physical computing, we run family workshops either as part of school programs or independently. The workshops are aimed at children accompanied by a parent (or any kind of adult family member or friend).

During the 3 1/2-hours workshop participants learn step-by-step how to program the super hero robot. This involves learning the use of simple commands as well as events, conditionals, and loops using also sensing commands and operators. We also show how to create a function and how to use it. Implicitly are taught basic computing concepts such as an understanding of algorithmic problem-solving (problem statement, implementation and testing cycle), collaboration in form of pair programming as well as the understanding that a computer program is a set of step-by-step instructions to be acted out. The instructors are either professors or graduate students with the assistance of undergraduates of the Department of Informatics and Statistics at the Federal University of Santa Catarina (UFSC) in cooperation with the Federal Institute of Santa Catarina (IFSC).

So far we have run 5 family workshops. They took either place at the Federal University of Santa Catarina or in schools in the state of Santa Catarina/Brasil. In total, more than 65 families, with children ranging from 6 to 13 years, participated.

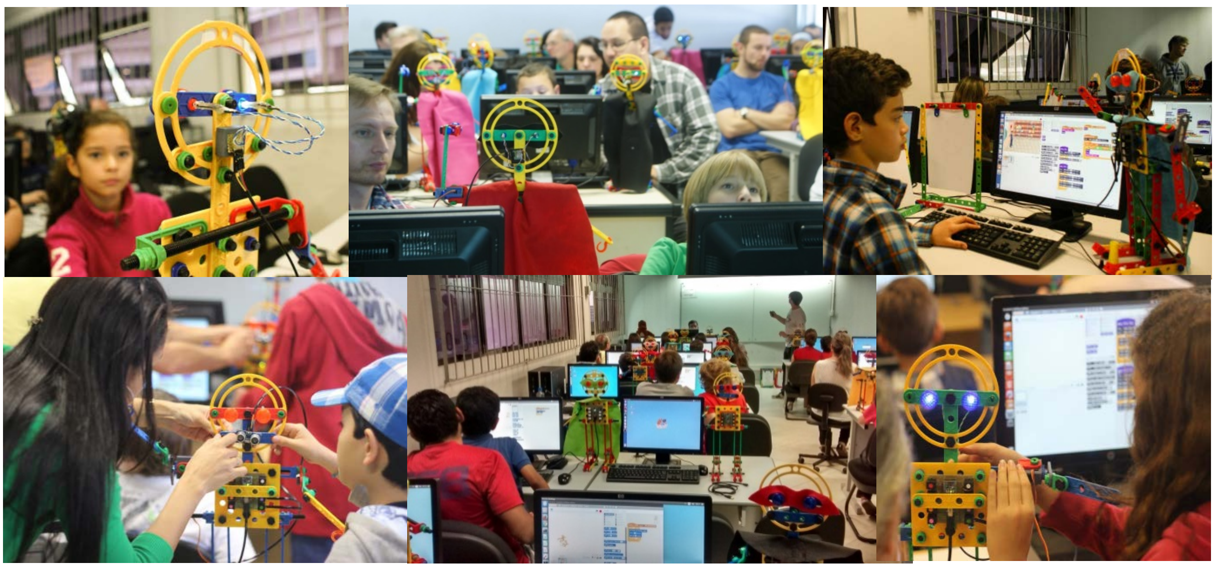


Figure 4. Scenes from the family workshops for physical computing

Logistics (registration, coffee-break and certificates) have been handled by administrative staff of our initiative. The workshops were offered for free funded by the Google Rise Award and governmental funding (CNPq).

Lessons Learned

All workshops have been very successful. We publicized the workshops via social networks and local press, observing an increased interest with each workshop with waiting lists already for the next ones.

During the workshops, participants actively take part and are able to follow the instruction. Especially when testing the steps, the kids express their enjoyment and satisfaction e.g., by moving the arm of the robot. Step-by-step instruction with immediate execution by the participants involving problem statement, implementation and testing, also allowed to implicitly introduce an understanding of algorithmic problem-solving. Using the scratchboard and the few-button interface communication server has also been shown to facilitate the learning process and enable a greater emphasis on programming aspects when compared with earlier concepts of the workshop using traditional breadboards and command line interfaces for establishing the communication. In the workshops, Scratch has also shown to be an extremely intuitive environment to teach programming.

Parents as well as the kids enjoy programming the super hero robot and we observed that they immerse into the story. One child for example had himself filmed by his mother re-telling the story while running the robot he programmed.

The children like the workshop a lot, expressing this in their comments: “Very cool and lot of fun”; “I loved the workshop”; and “Incredible – it’s magic” emphasizing especially that they liked make a robot move. As well as the parents who also demonstrated their contentment praising especially the didactic, dynamic and active format of the workshop in which it taught programming concepts in a creative and attractive way for the children. Several parents also cited as a strength the possibility of having this experience together with their child creating valuable parent-child activity time. This positive evaluation has also been confirmed by post-class surveys in which the workshop was highly rated: on a scale of 1- poor to 4 - excellent, the participant's average ratings were 4 (median) for the workshop in general. The majority of the children also considered the workshop easy and fun. At the end of the workshop the large majority (60 children) think that they can make computer programs and want to learn more about programming. The parents also expressed a very positive feedback indicating that they liked very much participating in the workshop and learning on how to make computer programs. The only improvement suggestions were related to offering more such workshops including further functionalities such as a walking robot. Comments from the parents also indicate that they recognize the importance of computing education as they expect it also to contribute to logical reasoning, creativity and concentration/persistence. Parents also commented that the workshop helped to demystify computing giving an example that computing can be easy and fun to learn.

At the end of each workshop, most participants (parents and kids) were eager to continue at home and/or through other workshops.

The positive potentials of such family workshops go well beyond computing education for children by involving the parents to learn along with the kids and providing also valuable parent-child activity time. The workshops yield benefits by contributing to the children's education through parental involvement, but also by reaching a different group - the adults – and, thus, providing computing education to a larger share of the community.

Due to the success we plan to continue these workshops not only as self-contained events but also as a part of the school programs. In order to widen their application we are also starting to develop teacher courses. Furthermore we are developing additional activities via online tutorials in order to enable participants to continue at home afterwards.

Acknowledgements

Scratch and the Scratch web site have been developed by the Lifelong Kindergarten Group at the MIT Media Lab. The development team included: John Maloney, Natalie Rusk, Evelyn Eastmond, Tammy Stern, Amon Millner, Jay Silver, Han Xu, Eric Rosenbaum, Karen Brennan, Brian Silverman, Andrés Monroy-Hernández and Mitchel Resnick. Snap! has been developed at the CS Department at UC Berkeley, mainly by Brian Harvey and Jens Mönig. Jens Mönig has been additionally supported by Miosoft and SAP. s2a_fm was developed and is supported by Alan Yorinks as a private project. We want to thank especially Alan Yorinks and Jens Mönig for fruitful discussions of different aspects of this project.

We want also to thank all students that helped during the workshops.

This work was supported by the Google Rise Award and the CNPq (*Conselho Nacional de Desenvolvimento Científico e Tecnológico – www.cnpq.br*), an entity of the Brazilian government focused on scientific and technological development.

References

Computacaonaescola.ufsc.br. 2013. Iniciativa Computação na Escola. Retrieved 30 November 2015, from http://www.computacaonaescola.ufsc.br

Malan, D. J. and Henry H. Leitner, H. H. Scratch for budding computer scientists. In Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education (SIGCSE). ACM, New York, NY, USA, 2007, 223-227.

Harvey, B. and Mönig, J. 2015. Snap! (Build Your Own Blocks) 4.0. Retrieved 30 November 2015, from http://snap.berkeley.edu/

Monroy-Hernández, A. and Resnick, M. FEATURE: Empowering kids to create and share programmable media. Interactions, 15(2), 2008), 50-53.

Scratch.mit.edu,. 2007. Scratch - Imagine, Program, Share. Retrieved 30 November 2015, from https://scratch.mit.edu/

Yorinks, A. 2013. MrYsLab/s2a_fm. Retrieved 30 November 2015, from https://github.com/MrYsLab/s2a_fm