A national experience in training teachers: Scratch and Robotics in Uruguay

Una experiencia nacional en la formación de profesores: Scratch y Robótica en Uruguay

Inés Friss de Kereki, André Fonseca de Oliveira

Abstract

In 2007, the Uruguayan government launched the "Basic Information Educational Program for Online Learning" ("CEIBAL"). The project's immediate objective is to provide all public primary school students and teachers with free laptop access. CEIBAL's longer-term objective is to promote social justice. It combines the distribution of computers with a program to train teachers. In 2011 a specific program was launched to train all Informatics teachers (1600 people) in Programming and Robotics. The school of Engineering of Universidad ORT Uruguay was selected to train 800 of those teachers. In this article, we describe the characteristics of the courses and participant's opinions.

Categories and Subject Descriptors: Computers and Education

Keywords: Programming, learning - teaching, experiences.

Resumen

En 2007, el gobierno uruguayo puso en marcha el "puso en marcha el plan "Conectividad Educativa de Informática Básica para el Aprendizaje en Línea" ("CEIBAL"). El objetivo inmediato del proyecto fue proveer a todos los estudiantes y maestros de escuelas primarias públicas de acceso a computadores portátiles gratis. El objetivo a largo plazo de CEIBAL es promover la justicia social. Combina la distribución de computadoras con un programa para capacitar a los maestros. En 2011 se puso en marcha un programa específico para formar a todos los profesores de Informática (1600 personas) en programación y robótica. La Escuela de Ingeniería de la Universidad ORT Uruguay fue seleccionada para formar 800 de esos profesores. En este artículo se describen las características de los cursos y opiniones de los participantes a manera de resultados.

Palabras Claves: Programación, experiencias, enseñanza.

Recibido: Noviembre 14 de 2013 Aprobado: Noviembre 25 de 2013

Tipo de artículo: Artículo de investigación científica y tecnológica terminada.

Afiliación Institucional de los autores: (Universidad ORT Uruguay). Cuareim 1451. Montevideo. Uruguay.

Los autores declaran que no tienen conflicto de interés.

Introduction

Uruguay is a small country in South America with 3.5 million inhabitants. In 2007, the government launched the "Basic Information Educational Program for Online Learning" ("CEIBAL") [7]. It is based on the ideas of the project "One Laptop per Child" (OLPC).

The project's immediate objective is to provide all public primary school students and teachers with free laptop access. CEIBAL's longer-term objective is to promote social justice by promoting equal access to information and communication tools for all our people [1]. It combines the distribution of computers with a program to train teachers in the cognitive skills needed to use information technology for maximum benefit [1].

In this context, in 2011 a specific program developed by CEIBAL was launched to train all Informatics teachers (1600 people) in Programming and Robotics. The School of Engineering of Universidad ORT Uruguay was selected to train 800 of those teachers in 2011 and 2012. In this article, we describe the main characteristics of both courses and the participant's opinions.

Organization of the courses

The requirements of CEIBAL included to teach Scratch and Robotics in 3 modules of 12 hours. We designed the detailed curricula. The training consisted of 3 modules: a) Programming (Scratch [14]) (12 hours), b) Basic robotics programming (NXT-G [12])(12 hours) and c) Robotics projects (NXT-G and Enchanting [8])(12 hours). The courses were free for the teachers. In each class group were at most 80 K7-K12 teachers, almost all of them with little or no previous programming, Scratch nor Robotics knowledge. Most of the teachers taught informatics, programming or personal computer maintenance. For instance, the ANEP-CES (National Administration of Public Education - Board of Secondary Education) official curricula of K7 informatics [2, 3] includes use of presentation, word processing and spreadsheet software, making web pages and using Internet tools. The curriculum of K8 besides includes a brief topic related to the use of working environments for learning programming (simple concepts and practical elements using Turtle Art, Etoys and/or Scratch). The K7 and K8 curriculum of Informatics of the ANEP–CETP (National Administration of Public Education - Council of Technical Vocational Education) [4, 5] are similar and include, besides other topics, the use of operating systems, word processing, presentations and spreadsheets software. It is encouraged to collaborate with teachers and projects of other subjects (for instance, Math or Physics).

We decided to include a lecturer and 10 teaching assistants (TA) to help the development of each course. Each topic was briefly presented by the lecturer and then the participants developed applications and gained knowledge and practice. The TA helped and solved particular questions or problems. They were advanced students of Computer Science and Electronics Engineering of our University.

In order to select the TA, we offered an open and free training for our Engineering students and choose more than 25 students.

Programming course: Scratch

The required programming language was Scratch. Each course consists of 12 hours of full training. We designed the course. It contains theory and a strong emphasis on exercises.

We divided the training in 3 days: a) main description of Scratch; control structures and variables (4 hours); b) image processing and events (4 hours); and c) games and sensors (4 hours). We describe here each day.

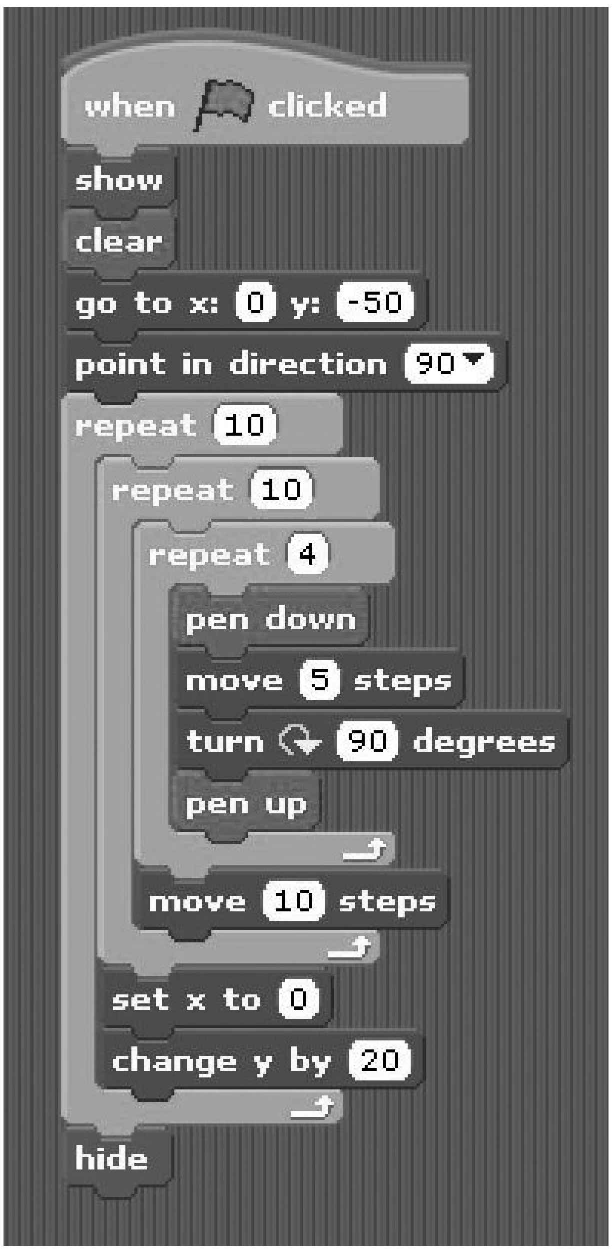
Module 1: First day

In the first class, we explained the basic programming concepts using daily life examples, like cooking pancakes or crossing a street watching the lights. The notions of what a program is, algorithm and control structures were discussed. We explained that programming is not only a way to solve problems, it is an activity that collaborates in developing creative thinking, team working abilities and systemic reasoning.

After that, we presented the Scratch interface and programmed simple exercises like drawing a square. We showed the basic elements of Scratch (blocks area, scripts area, stage, etc). We developed some introductory samples, like to draw a square using a pencil. We explained the use of the green flag and how to save and open a Scratch file.

We asked the students to solve some similar problems: to draw a rectangular grid (see Figure 1: Grid), to draw their initials and to draw regular shapes. The variable concept was presented using an analogy with a calculator. Some examples related to math were presented: sum of angles of a triangle, area of a rectangle and to simulate probability.

Figure 1. *Grid.*

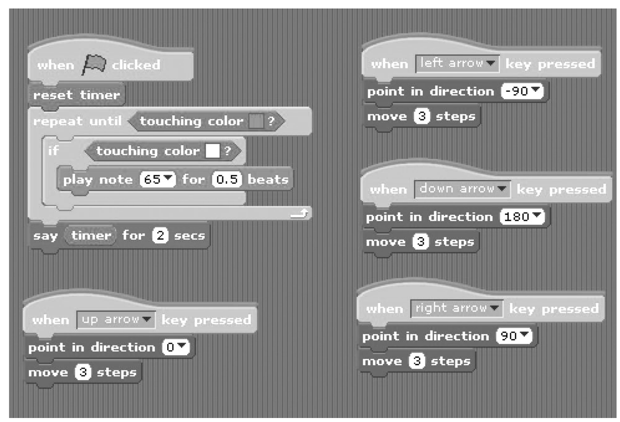


We strengthened the idea of incremental development and also to combine different media (for instance to include sounds). Also we analyzed different projects provided by Scratch like "Aquarium". We introduced some modifications to this animation, in order to empathize the ideas of collaboration, remixing projects and sharing programs.

Module 1: Second day

In the second day, related to image processing and events the proposed exercises included: to program a screen saver, to design a clock, to traverse a maze and to simulate a car race. (See Figure 2: Maze).

Figure 2. *Maze.*

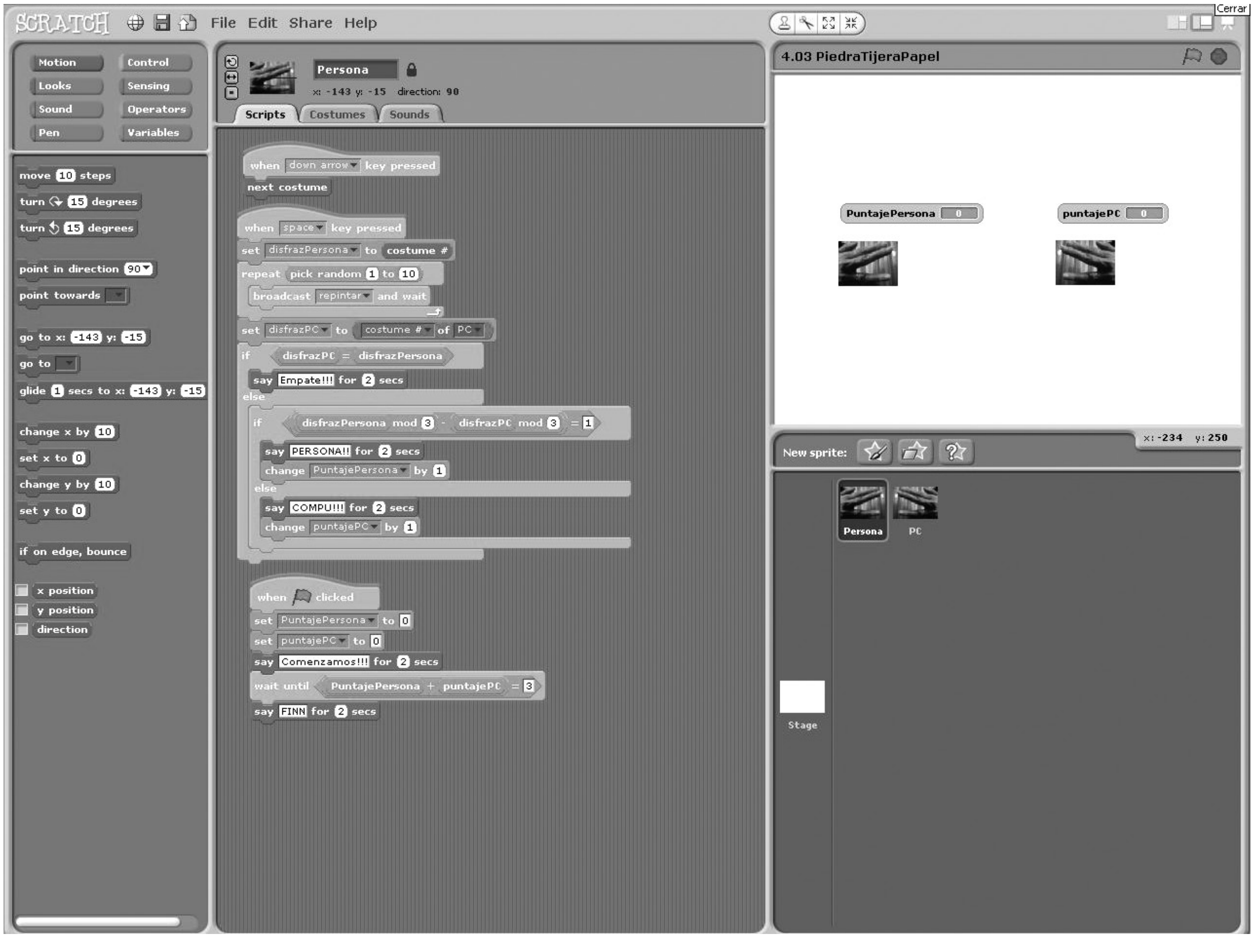


We explained in detail how to manage events and how to use different sprites. We also analyzed operators and conditionals. In each sample, we asked the participants for applications to their particular courses. In some cases, teachers proposed very interesting ideas, for instance related to simulating electric circuits in collaboration with Physics' teachers.

Module 1: Third day

In the last day, some games were developed. For instance, the hand game "rock-paper-scissors" was implemented. It included pictures (taken with the XO camera) and messages: sending messages to sprites. (See Figure 3: Rock-Paper-Scissors). Also some guides related to debugging and testing programs were shared. The iterative and incremental software development cycle was discussed.

Figure 3 Rock-Papers-Scissors.



Related to sensors, the most appreciated example was the simulation of a whirligig's movement (the wind is simulated by blowing into the sound sensor). It includes only 2 Scratch blocks: "forever" and "turn loudness degrees"! We discussed in detail the Paint Editor, and its options, like "set costume center". (See Figure 4 Whirligig (code) and Figure 5 Whirligig (complete)).

Figure 4 Whirligig (code).

It also was interesting to draw a route and to program an automatic walk of the route by the Scratch cat, target shooting and a slot machine. Tanks battle was also appreciated. (See Figure 6: Slot Machine)

Some programming concepts, like how to manage run time errors, the use of program comments, how to make compounds conditions with "and" and "or", and some other tips were discussed through the development of the course.

In particular, we discuss that Scratch is a good tool. It is extremely recommendable that the professor analyze and try himself or herself the solution of a particular example before proposing it to the students. Scratch is not applicable for all possible problems. Also, we emphasized that there is more than one possible solution to a problem and a good strategy is first to discuss general cases and after that, the particular details.

Figure 5 Whirligig (complete).

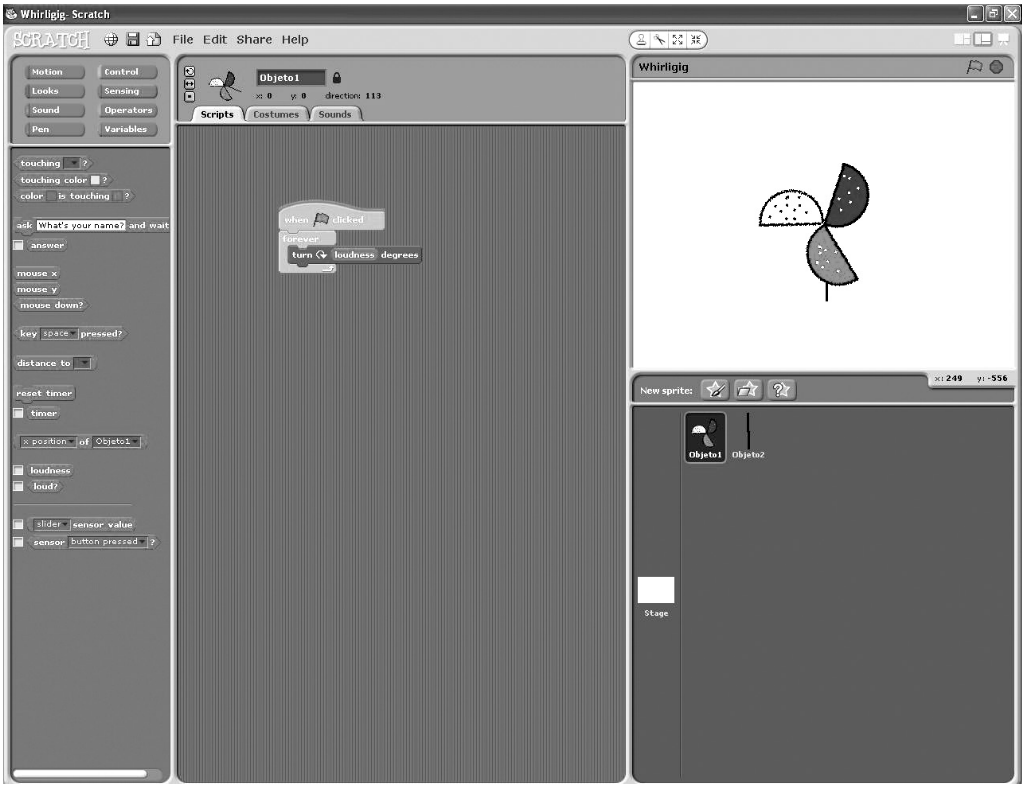
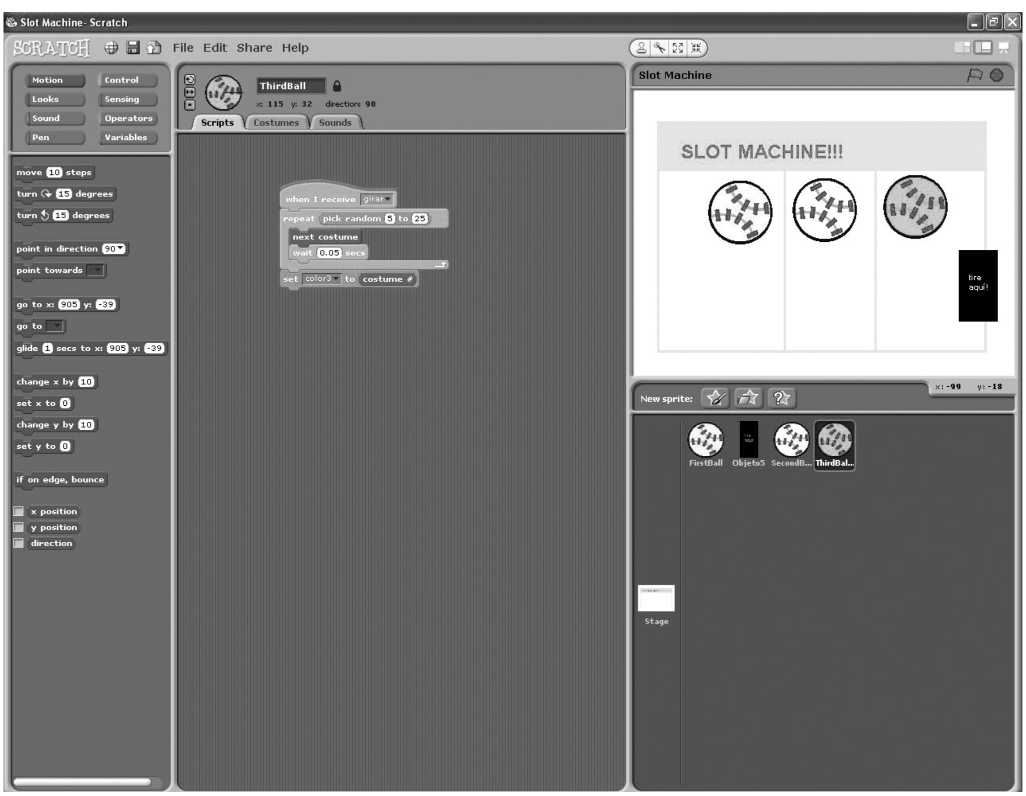


Figure 6 Slot Machine.



Robotics courses

Module 2: Robotics basic programming

With basic programming skills acquired in Module 1, the aim of the second module was to learn how to make programs that interact with the input signals (sensors) and outputs (motors). In this module the programming language was NXT-G: a graphical programming environment that comes bundled with the NXT [11].

This module consists of 12 hours of theory and exercises, divided in a 3 days training. During this module the participants worked in pairs with a pre-assembled robot model: this is because the focus is on programming the controller. Figure 7 Robot shows the model used (without sensors) [13].

Figure 7 Robot.

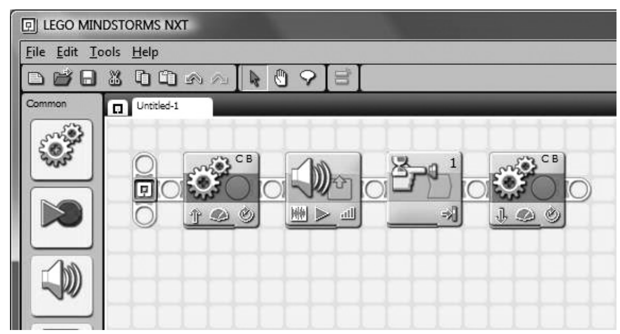
Module 2: First day.

This module consists of:

a. **Introduction to Robotics and LEGO NXT kit (2 hours).** In these first two hours we introduced some aspects of today's robotics, from the robot used in the exploration of Mars, to commercial applications as the robot smart cleaner. A quite interesting application is Asimo, the Honda humanoid robot [6]. As a motivational part we introduced some robots examples created with the NXT kit. Right away we introduce the NXT kit: the controller, the sensors and the motors. In this phase, participants learned how to operate the controller's menu, and make small programs (2 to 5 instructions programs).

b. **Introduction to NXT-G: a graphical programming environment (2 hours).** As the official programming environment by LEGO, NXT-G [12] is a simple language and has all the programming options for NXT. (See Figure 8 Environment) It turns out it is a must for a first contact with LEGO robotics. In these early practices participants learned how to start the programming environment and build a program, how to connect and monitor the controller. They made a first "Hello world" program, loaded and ran it on the controller. Finally they learned the basic programming blocks on NXT-G. All was accompanied by simple exercises.

Figure 8 Environment .



Module 2: Second day.

On this second day the focus was to learn how the NXT-G palette is organized and to learn how to use the blocks. In order to facilitate the comprehension, most of the exercises were based on modifying the examples provided.

The structure was:

a. **Basic Programming with NXT-G (2 hours).** As a first approach to programming with LEGO NXT, we introduced how the NXT-G environment is organized: code is made by connection of function blocks existing in three palettes (common, complete, custom). A brief description of each block was made, with emphasis in how to use Help to have a complete description of the block's functionality. Wiring between blocks to achieve data communication were used to build more complex codes.

b. **Special blocks and multitasking (2 hours).** After the basic programming blocks were introduced, some important concepts to improve algorithms were presented: use of constants and variables, and how to calibrate sensors. More complex code needs the ability to define user blocks (user functions). To achieve this, LEGO NXT-G allows the user to create macro blocks. Participants made some exercises where they created their own blocks. At the end, the concept of multitasking was introduced in examples where codes for acquiring sensors data and making movements were executed in parallel.

Module 2: Third day

The structure was:

a. **Advanced Concepts (2 hours).** It is important to know how to update the LEGO NXT controller. The firmware was explained, what it is, how to download a new version and how to update it. As an example of data communication it was shown how to use Bluetooth in programs. Finally, it was introduced the new "Data Logging" environment, used to capture data with sensors.

b. **Advanced exercises (2 hours).** As the last module exercise, the participants had to develop a "Line Follower", in a three step exercise:

   1. Make an algorithm to calibrate the light sensor and turn it into a "My block".
   2. Do a simple line follower with a "color detection"-"small movement" loop.
   3. Improve the line follower with multitask: color detection and movement at same time

Module 3: Robotics projects

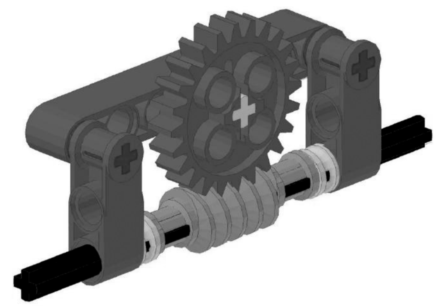
While the aim of the second module was the programming of robots, in the third we worked with projects and alternative environments. This module also consists of 12 hours divided in a 3 days training.

Module 3: First day

It was divided in:

a. **Knowing the mechanics of LEGO NXT (2 hours).** To achieve the ability to build mechanisms from scratch, the participants had to known the LEGO pieces available. Due to this, the LEGO NXT Mindstorm Education Set kit [10], was introduced and some basic techniques and tips for assembling structures with the kit were explained. Some simple exercises with gears and motors were proposed to achieve the comprehension of LEGO bricks mechanisms. (See Figure 9: Lego)

Figure 9 Lego



b. **Enchanting (2 hours).** As an alternate programming environment, Enchanting [8] was chosen for two main reasons: it is a Scratch based environment and runs in Windows and Linux. This time, as there was no need to learn programming structures, only a small introduction of how to load the new firmware and configure motors and sensors was made. Here, the main objective was to show the interaction, with variables and messages, between the Enchanting and Scratch environments. A few examples on how communicate the LEGO NXT and Scratch were presented. In the last exercise, participants built a joystick using a LEGO motor to control a cannon in a version of Space Invaders game programmed in Scratch. (See Figure 10 Space Invaders)

Figure 10 Space Invaders

Module 3: Second day

The plan was: *First project from scratch: a spin art* (4 hours). This proposal consisted in a simple base with a motor to turn around pieces of paper (See Figure 11 Spin art model and Figure 12 Spin art in action). With a colored pencil set the participants could generate drawings on paper. We proposed three levels of difficulty for the programming: a basic level with only turn on and off the motor, a second level with velocity control using the LEGO NXT buttons, and a third level controlling the velocity with the PC keyboard, using communication between Scratch and Enchanting.

Figure 11. *Spin art model.*

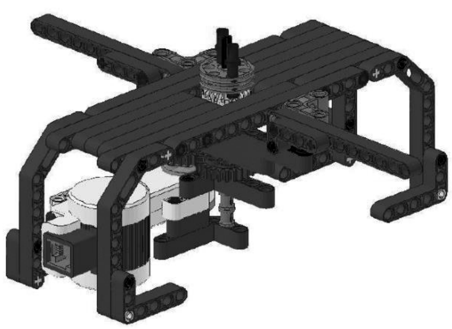
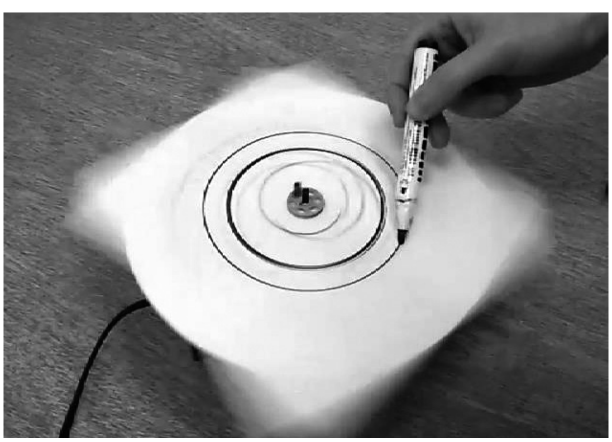


Figure 12. *Spin art in action.*



Module 3: Third day

The proposal was *Final project* (4 hours). On this last day the participants had to choose one of three possible projects: a frog, a reflex game an robot arm. (See Figure 13 Robotic Frog, Figure 14 Reflex Game and Figure 15 Robot Arm). Each project had different difficulties about mechanical construction and programming, classified from 1 (easier) to 3 (harder). Table 1 summarizes this classification (See Table 1: Projects). Thus, participants could choose the project that best fitted their expectations.

Figure 13. *Robotic frog.*

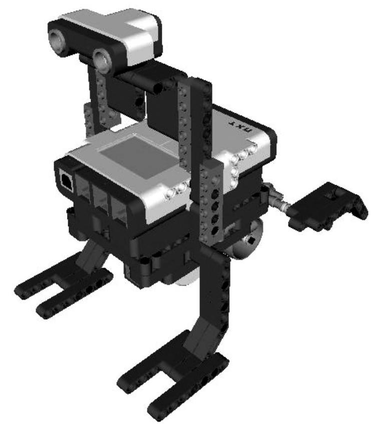


Figure 14. *Reflex game.*

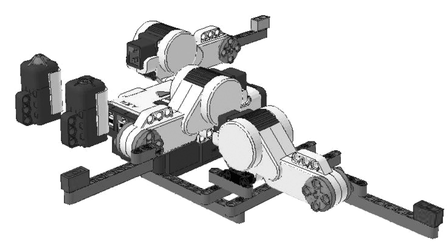


Figure 15. *Robot arm.*

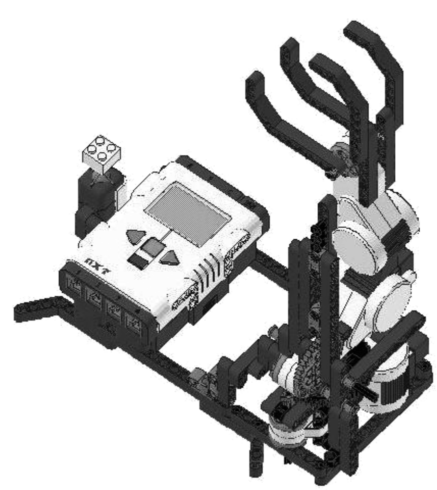


Table 1: Projects

Project	Mechanical complexity	Programming complexity
Frog	2	1
Reflex game	1	3
Robot arm	3	2

Surveys of courses

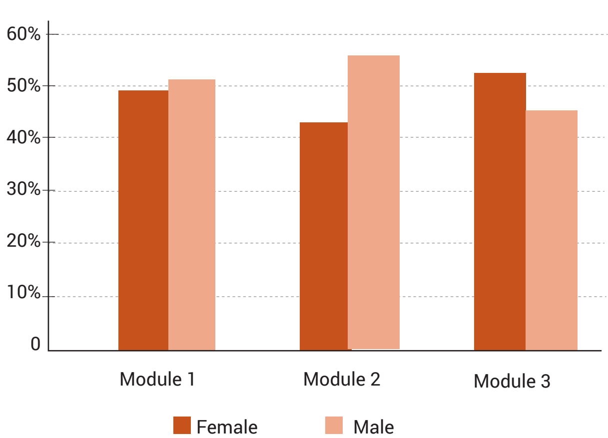
In order to know the participants opinions of the training, we conducted a non compulsory survey after each course. In some cases, the required information was not completely given. Table 2 summarizes the number of surveys returned by module, as well as the average, and teaching experience age of the participants. As can be seen, participants had mid age profile and experience, but with a high deviation.

Table 2. *Surveys*

	Returned surveys	Average age (stdv)	Average teaching experience in years (stdv)
Module 1	452	39.7 (9.9)	11.3 (8.1)
Module 2	541	41.5 (10.1)	12.8 (8.7)
Module 3	414	41.0 (9.6)	12.3 (8.3)

The gender composition was approximately 50% in all modules, with small variations (Figure 16 Gender composition of courses participants).

Figure 16. *Gender composition of courses participants*



Even though participants were informatics teachers, they actually have been teaching office programs. When asked about programming skills only 41% felt they had adequate knowledge (15% high and 26% average). (Figure 17 Programming skills) With respect to Scratch, about twenty percent of teachers indicated that they previously knew and used Scratch in their courses; fifty percent indicated that they knew but did not use Scratch and thirty percent said that they did not know Scratch. (See Figure 18 Knowledge and/or use of Scratch.) Related to attending previous courses of Scratch, 75% answered that they had not attended or received any specific course. Regarding the use of robotics kits in education (particularly the LEGO Mindstorm NXT), 80% replied that they did not know anything. (See Figure 19 Knowledge and/or use of Lego NXT).

Figure 17. *Programming skills of participants.*

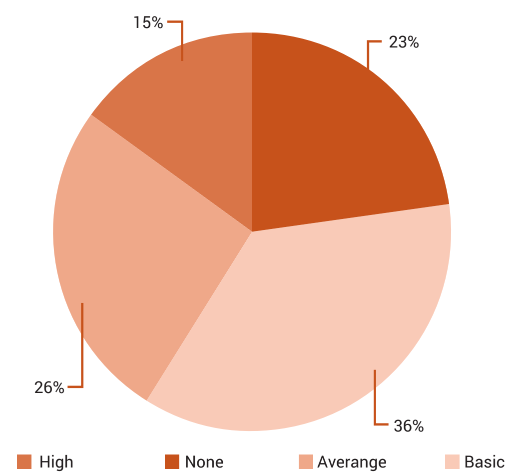


Figure 18. *Knowledge and/or use of Scratch.*

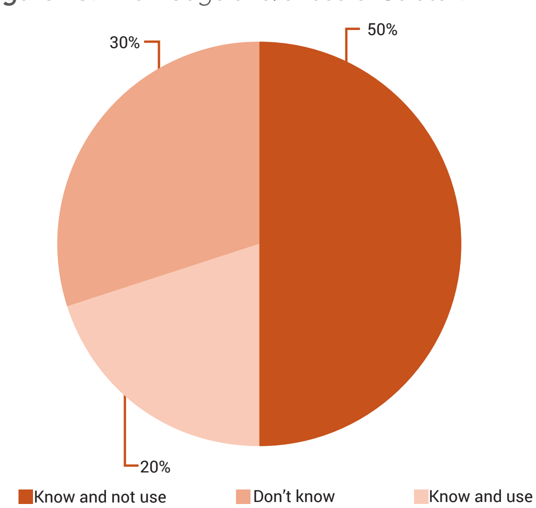
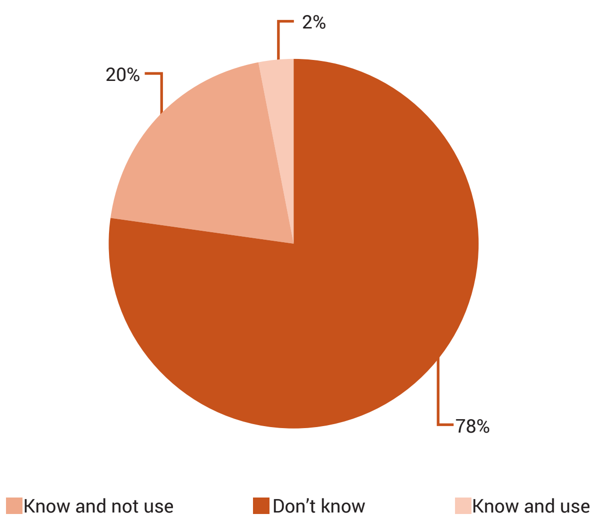


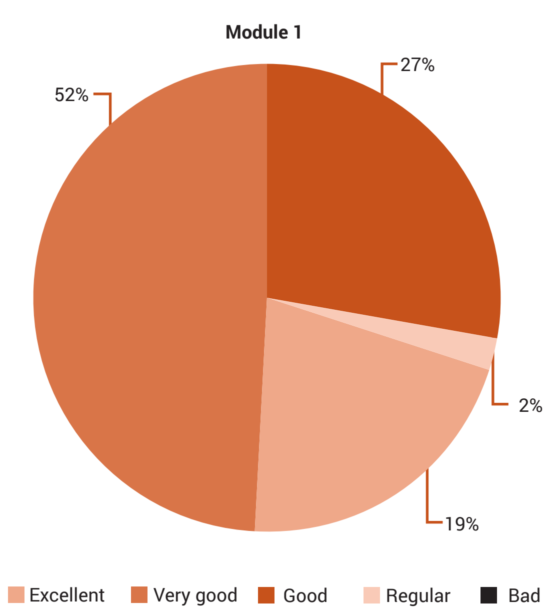
Figure 19. *Knowledge and/or use of Lego NXT.*

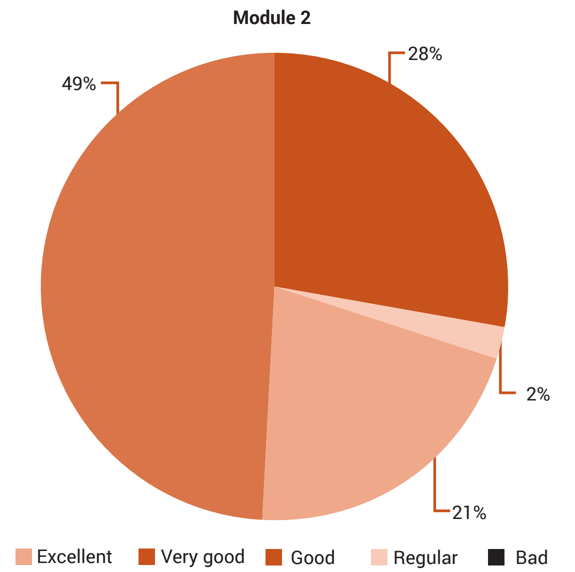


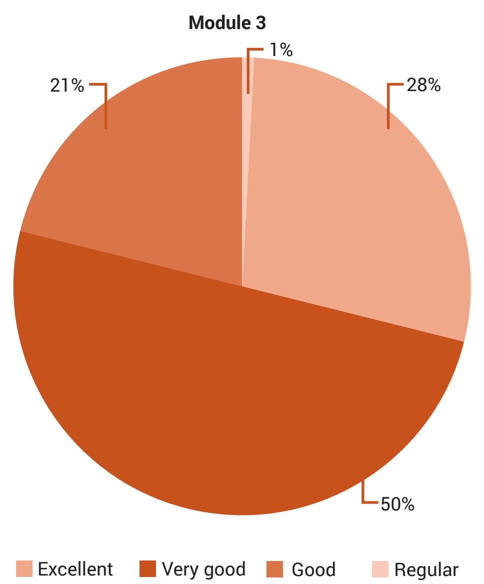
About the proposed activities, in Scratch module (Module 1) the preferred exercises where those related to sensors (example: whirligig), robotics (example: maze' simulation) and games (example: Rock-Paper-Scissors). In the robotics modules, the most preferred activity was work in projects.

The general opinion about all modules was really good. About ninety eight percent expressed that the courses were good, very good or excellent. See Figure 20 Opinion of participants about the courses.

Figure 20. *Opinion of participants about the courses.*







The participants pointed out that the best things of the course were the personalized attention, the practical approach and opportunities of improvement of their own classes.

Upon completion of all modules they were asked if they would use Scratch, Lego NXT or both in their own courses. Almost ninety percent (88%) said "Yes" for the use of Scratch. An important percentage said they intended to use Lego NXT in classes, if they had access to the kit (76%), and a 71% intended to use both. See Figure 21: Future intention of use in classes

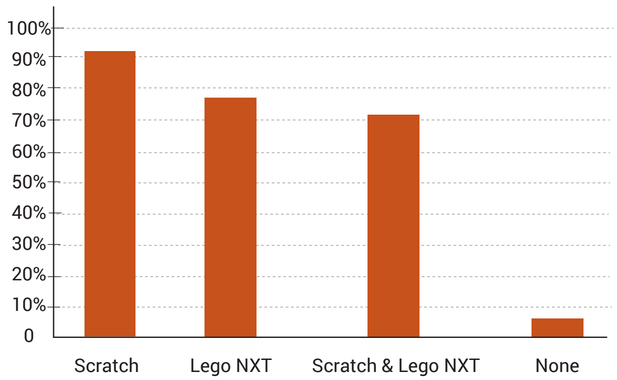


Figure 21. *Future intention of use in classes.*

We should point that Universidad ORT Uruguay was selected to design the courses and to train the teachers. At the moment, we do not have formal information about their follow up. Although the training is too recent to detect remarkable changes, some new experiences have been reported by the teachers [9, 15].

Conclusion

To sum up, based on the results of the experience we could infer that it was very successful. From an initial low quantity of teachers using Scratch, we obtained an a high number of teachers thinking on using it for their courses. The same occurs in the use of robotics to stimulate the learning of programming skills. A pending topic is to evaluate in the long term the impact of the training effectiveness.

As a whole, the participants' perception of the courses was mostly very good or excellent. It was also an interesting experience for all of the teaching assistants, as students it was a way to get more involved with the University, teaching and professors.

References

[1] Americas Quarterly, http://www.americasquarterly.org/node/370. Accessed 2013 March 22.

[2] ANEP – CES (National Administration of Public Education - Board of Secondary Education)– Informatics 1st year. http://www.ces.edu.uy/ces/images/stories/reformulacion06primerocb/informatica1cb22.pdf Accessed 2013 September 28

[3] ANEP – CES (National Administration of Public Education - Board of Secondary Education) – Informatics 2nd year. http://www.ces.edu.uy/ces/images/stories/reformulacion06segundocb/informatica2cb.pdf Accessed 2013 September 28

[4] ANEP – CETP (National Administration of Public Education - Council of Technical Vocational Education)– Informatics 1st year. http://www.utu.edu.uy/webnew/modulos/utu/Areas%20de%20Cursos/Cursos%20y%20Programas/Ciclo%20Basico/Tipo%20de%20Cursos/CBT%202007/PROGRAMAS_1ERO_CBTyCBTA_2007.pdf

[5] ANEP – CETP (National Administration of Public Education - Council of Technical Vocational Education) – Informatics 2nd year. http://www.utu.edu.uy/webnew/modulos/utu/Areas%20de%20Cursos/Cursos%20y%20Programas/Ciclo%20Basico/Tipo%20de%20Cursos/CBT%202007/PROGRAMAS_2DO_CBTyCBTA_2007.pdf

[6] Asimo, asimo.honda.com. Accessed 2012 June 25.

[7] CEIBAL, www.ceibal.org.uy. Accessed 2013 March 22.

[8] Enchanting, enchanting.robotclub.ab.ca. Accessed 2012 June 25.

[9] ExpoAprende 2012 Plan CEIBAL http://expoaprende.ceibal.edu.uy/ Accessed 2013 September 30

[10] LEGO NXT Mindstorm Education Set, http://www.legoeducation.us/eng/product/lego_mindstorms_education_nxt_base_set/2095. Accessed 2012 June 25.

[11] NXT, http://mindstorms.lego.com/en-us/whatisnxt/default.aspx. Accessed 2013 August 5

[12] [12] NXT-G, http://www.nxtprograms.com/help/learn.html. Accessed 2012 July .13

[13] Robot LEGO-CEIBAL, www.ceibal.edu.uy/Articulos/Paginas/Material para la utilización de los kits de Robótica.apx. Accessed 2012 July 13.

[14] Scratch, scratch.mit.edu. Accessed 2013 March 22.

[15] Scratch Day 2013 – Universidad ORT Uruguay http://fi.ort.edu.uy/scratchday. Accessed 2013 September 30.

Los Autores

André Luiz Fonseca De Oliveira

Doctor (Cand.) en Ingeniería Informática, Universidad Politécnica de Madrid. Master en Ingeniería Electrónica, Universidad de la República. Ingeniero Electricista. Catedrático de Electrónica Analógica y Control Automático. Facultad de Ingeniería de la Universidad ORT Uruguay.

Materias dictadas: Diseño de sistemas de control, Dispositivos electrónicos, Proyecto de tesis, Sistemas dinámicos, Taller de diseño, Teoría de circuitos.

Inés Patricia Friss de Kereki Guerrero

Doctora en Ingeniería en Informática, Universidad Politécnica de Madrid. Diploma en Educación, Universidad ORT Uruguay. Ingeniera en Computación. Premio a la Excelencia Docente 1998, Facultad de Ingeniería. Catedrática de Programación. Facultad de Ingeniería de la Universidad ORT Uruguay.

Materias dictadas: Ingeniería Programación 1, Programación 2.