Code2Bot, a social robot for the classroom

J.M. Haut[1], M.E. Paoletti[1] P. Bustos[1], and N. García[2]

[1] Robolab, University of Extremadura, Spain
juanmariohaut@unex.es,
homepage: http://robolab.unex.es
[2] Nodo Educativo, University of Extremadura, Spain
nuria@unex.es

Abstract. LearnBot is a social low cost robot that has been designed into the area of educational robotics for promoting the development of computational thinking in different educational stages, specifically through the learning of the Python language. It has being entirely built using a robotics framework developed by our Robotic Laboratory. But Learnbot is part of a more ambitious ecosystem, Code2Bot, which is being created as a facilitator environment to encourage current teachers without programming knowledge to learn the basics of Python, at least into a level where they can introduce their students into the digital world. Also Code2Bot is developed as a collaborative environment to facilitate the introduction of robotics programming in non technological areas of the curriculum. This paper describes the design and construction process of Learnbot, the conception of the web ecosystem, a concise theoretical and conceptual framework and an analysis of the results of an initial validation experiment.

Keywords: Educational Robotics, Programming, Social Robots, Computational thinking.

1 INTRODUCTION

The use of robots as tools to facilitate technological education inside the social robotics field is rapidly gaining momentum. The educational robotics allows students to experience situations that contribute to acquire cognitive strategies for solving, planning and execution real problems. Through real situations students can locate the main problem, segment its structure in a sequence of steps and imagine, formulate, construct and experience the possible solutions. Recent research conclude that the application of robotics in education improves learning when the applied methodologies are consistent with the different factors involved in the teaching-learning process [1].
Among the capabilities and values relevant for the teaching-learning process of educational robotics, we highlight those reported by [2]:

a) It integrates different areas of knowledge.

b) It operates with manipulative objects, helping in the transition from abstract to concrete.
c) It enables students the use of different languages (graphical, iconical, mathematical, etc.).
d) It operates synchronously with different variables
e) It encourages the development of a systemic and systematic thinking.
f) It contributes to the construction and evaluation of strategies for acquiring knowledge through educational guidance.
g) It creates learning environments by itself.
h) It applies and teaches the scientific process and the mathematical representation and modeling.
i) It establishes a playful and heuristic learning environment.

These elements remit us to the construction of computational thinking [3]. This term refers to a set of patterns of thought that involve systematically and efficiently the processing of information and accomplishment of simpler tasks, through abstraction and fragmentation of more complex tasks. [3] argues that this kind of analytic thinking shares ground with the mathematical thinking of problem solving, with the thinking developed in engineering, based on the design and evaluation of complex systems that operate within the constraints of the real world, and with the scientific on which the understanding of computability, intelligence, mind and human behavior are sustained [4]. The definition of computational thinking has evolved, its meaning becoming stronger and refined, in order to identify practices or methodological strategies that provide a framework of effective evaluation. Selby and Woollard [5] proposed a definition: 'the computational thinking is an activity, often product oriented, associated with, but not limited to, problem solving' and 'it is a cognitive process that reflects different skills as: think in abstractions, think in terms of decomposition, think algorithmically, think in terms of evaluations, and think in generalizations'.
The development of computational thinking is also associated with a number of essential attitudes such as self-confidence in managing complexity, persistence in working with difficult problems, frustration tolerance, communication skills and competence to work collaboratively with others to achieve a common goal.
Without being exhaustive, we review below some of the educational robots that are already being used in classrooms. For example, AIToy is a robot designed by AISoy Robotics Company, in collaboration with the Universidad Rey Juan Carlos of Madrid, as a toy with educational purposes in which the concept of gamification becomes important in an educational model denominated by the authors as 'neo natural education' [6]. Also, we found numerous educational experiences with robots both at the formal education sector as in the informal. In the last one the experiences with physical robots are proliferating, in particular based on the construction of robots with Lego and Lego Mindstorms. The research on the influence on student motivation that the use of educational robots can do is very interesting [7], and also the experiences gained in secondary education by building robots with Lego as an optional subject [8]. A clear example is Ipod-Lego, a robot developed at the University Ramón Llull that proposed

activities of social (e.g. communication skills) and academic area (the robot proposed various tasks of different curriculum subjects and memory exercises) [9] or IROBI, a domestic robot designed as an human-friendly Internet-connected home robot with e-Learning technology and as a learning tool-assistant at home [10].
Beyond educational robotics there is programming, the universal glue that links the digital world. The goal of this investigation is not oriented to make students build a physical robot, but to make them start with a textual programming language. Python has been chosen because its simplicity, its small learning curve, the variety of tools available in the form of a huge number of modules ready to be used, and an interactive shell easy to use. All these characteristics are essential for a novice programmer or for students that begin the programming path.
Through programming we want to reach two interrelated objectives: a) to promote the development of computational thinking by solving real problems and b) to stimulate the constructionist pedagogical aspect [11] of technology, understood not only as an information medium but also as construction, development and creation.

2 THE CODE2BOT ECOSYSTEM

In order to achieve these goals and to contribute to make changes happen in our current technological deprived educational system, we are bulding Code2Bot. Its main purpose is to build a learning community that supports, transfers and co-builds knowledge, sharing experiences and programming codes. LearnBot has been designed as the material tool of Code2Bot like an educational robot built to fulfill the following specific objectives:

- To structure the initiation into teaching-learning process of Python programming language.
- To provide LearnBot with the tools and strategies needed to be successfully incorporated into the classrooms at different educational levels.
- To make LearnBot a multidisciplinary collaborative tool valuable for the teaching-learning process.

All these without detriment of creating a low-cost robot of educational quality and versatility, that can be made available to any institution and educational center.
Code2Bot is composed of several pieces that fit together into a well designed, usable digital ecosystem. It is composed of a low cost robot, LearnBot; a module that is integrated into student programs and connects the desktop computer to the robot via WiFi (LearnPy); a robot simulator to accelerate programs development (LearnSim); a repository in the cloud for storage, access and sharing the programs and resources generated (LearnCloud); a module that translates Scratch programs into Python code to facilitate access to the platform to K1 students (S2Bot), and a forum hosted on the same site and where all Code2Bot users can consult and share experiences, solve problems, propose new projects, discuss

results and collaborate on the collective creation of programs (ForoBot). For example, two classrooms could develop a program that would allow two robots, physically located in different cities but living in an identical physical environment, to solve collaboratively the same problem. Figure 1 shows an schematic representation of the overall system.

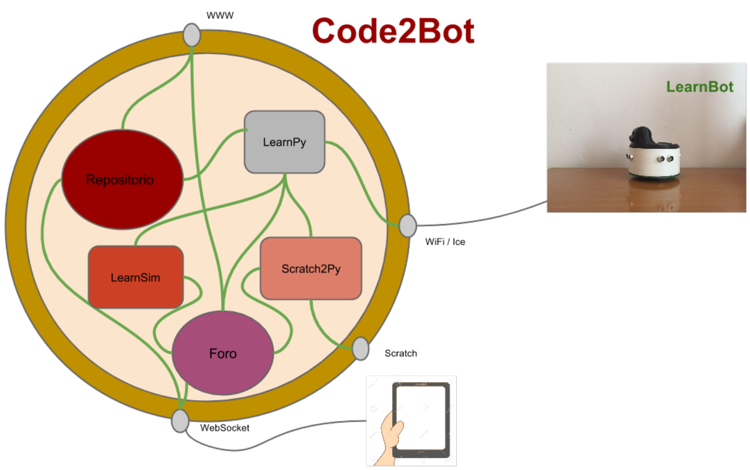

Fig. 1: Main parts of the Code2Bot ecosystem: the low cost robot (LearnBot), the bridge module between the robot and the internal ecosystem (LearnPy), the robot simulator (LearnSim), the translator module from Scratch to Python (S2Bot or Scratch2Py), the support of the forum (ForoBot) and the code repository (LearnCloud) on the cloud.

Code2Bot primarily seeks to facilitate and motivate the learning of textual programming in the final years of primary education (K4-K6), in Secondary Obligatory Education (ESO, K7-K10) and Baccalaureate (K11-K12). Finally, Code2Bot allows to share and store the experiences and programs created both by teachers and by the students, that can be organized according to contents, level of difficulty, goals, etc.

2.1 Learnbot

Although the overall aim of the Code2Bot system has remained unchanged since the beginning of the investigation, the design of LearnBot has evolved over time, always adapting to the available technology to produce an affordable, low cost, easy to program and manipulate, pleasant to use and with the services necessary to realize its function as an educational robot. Learnbot is an evolving idea of a low-cost robot that, in one way, facilitates the initiation of textual programming

(a) Generation I (b) Generation II (c) Generation III

Fig. 2: LearnBot design stages. See text for a detailed explanation

in Python and, in the other, facilitates the learning of curriculum contents by using a creative, hands on tool.

The unchanged requirements that are kept from the beginning are the cost, up to 100, and the use of a camera as the main sensor. So far, we have created three generations of LearnBots (Figure 2).

- The first generation is a rough prototype, without cover. The base platform is made of metal. The internal skeleton (plastic) supports a three-tier architecture:
  - First level has a 5V Lithium-ion Battery that supplied 10.000 mAh.
  - The second level has the master control unit (Raspberry Pi), responsible for running the high-level processes that shape the behavior of the robot.
  - In the third level there is the embedded low-level micro controller, Arduino Mega, which is responsible for controlling all robot sensors.

  Additionally, it had two wheels in differential configuration with encoders that were powered by a custom made H bridge and controlled by the Arduino. This first generation had four ultrasonic sensors, four LDR light sensors located in a small antenna and four CNY70 opto-reflective infrared sensors located at the bottom of the base.
- The second generation was designed to reduce the cost of production by eliminating the metal skeleton, replacing it with parts made using 3D printing technology. With this technology a protective plastic cover was added, that hides and protects the internal electronics of the robot composed by sensors (following the architecture of the first generation, LearnBot maintains four ultrasonic units and adds a camera on the front top of its upper side), a HardKernel ODROID-C1 as its embedded computer, a 5v, 11000mAh litium battery providing a total of 6h of autonomy and two power outputs, one supplying 1A that feeds the motors the drive the wheels, and another supplying 2A which feeds the control unit.
- The third generation is still in the design phase. It inherits from the second generation the idea of a 3D printed cover and chassis, but it has suffered a re-

modeling to add new features and make its appearance more attractive: The new design includes a 5” tactile LCD screen, a speaker and a microphone.

2.2 The software architecture of LearnBot

Robocomp is an open source robotic framework whose development started in 2005 [12] [13]. It is designed to be used in the field of robotics and is heavily based on the component-oriented programming (COP) paradigm. Robocomp uses as middleware Ice, the Internet Communications Engine by ZeroC [14] that provides the necessary technology to work in heterogeneous environments with different programming languages, networking technologies and operating systems.

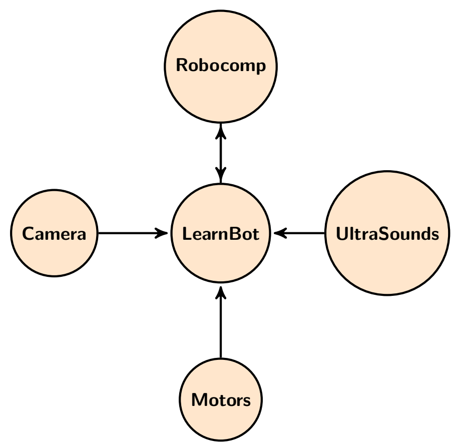

Fig. 3: RoboComp components in Learnbot-Core

Robocomp provides a component model with DSL based tools that generate code from high level specifications and a set of tools designed to reduce the programming time and the complexity of the deployed systems.

2.3 Python as the target programming language

Python has been chosen as the target programming language for some reasons:

– It has dynamic type resolution, which allows students to create programs agilely, without the necessity of stating the types of variables.
– It is an interpreted language that facilitates quick creation of code and provides a high degree of expressiveness.
– It is experimenting now a huge expansion all over the world in all disciplines and, specifically, in educational environments.
– The development community has generated many libraries that solve countless types of problems and allow connecting to a programming almost all types of web services and resources available on the Internet.

Also, Python provides an interactive shell where we run programs and lines of code without any prior installation. The Python interpreter can be obtained from the official website, and is integrated in most Linux distributions, so installation is not required. It offers features that can facilitate the learning process, allowing, for example, to design simple animations to build games.
Python is, therefore, a high level programming language, which offers a high degree of abstraction and expressiveness, efficient and portable to different current operating systems.

2.4 Connecting the robot with the PC

The python scripts that the robot executes are coded by students in their laptops and debugged using a 3D simulator. When students are ready to test the program, it can easily be redirected to the real robot. Only a matter of changing a configuration parameter. All these functionality is hidden in the LearnPy module, imported at the begining of students codes. LearnPy provides an instance of a class that is reading in the background the ultrasound measures from the robot and the video stream. Those data can be accessed through methods of the object. Another method allows the program to send velocity commands to the robot.
At start, the robot powers the embedded board and sets up a WiFi access point. It also starts the software components of the LearnBot, and waits for requests. Then, students connect their computers to the robot's access point, open a Python interpreter,import LearnPy and they can start coding. All low-level details are hidden and a very expressive environment, with access to the robot's video frames and to all libraries provided by Python, such as OpenCV, is at the hands of the users.

2.5 Programming Learnbot: first examples

Here are two simple examples which show what can be done using LearnPy and LearnBot:
Obstabcle Detect: The robot moves forward until it encounters an obstacle closer than 10 cm. Then stops.

```python
def my_code(self):
  distanceStop = 10
  self.runRobot()
  while (True):
    distance = self.getDistanceFront()
    if distance < distanceStop:
      self.stop()
      self.exit()
```

LineFollower: The robot follows a black line on the floor using three strategically positioned ROIs as samples of the whole image.

```python
def code(self):
  commands = (self.TurnRight(),
              self.straightAhead(),
              self.turnLeft())
  while(True):
    frame = self.getImage()
    rois = self.getROIS(frame)
    maxIndex = np.argmax(rois)
    commands(maxIndex)
```

The *getRois()* method extracts three ROISs from the image situated in a middle line, one at the left, other at the center and the last one at the right of the image. Note that this problem, requiring visual processing is not easy to explain to the children and to make help them come up with a simple solution. With the resources of the LearnBot environment and the expressiveness of Python, a simple and elegant solution can be achieved.

3 EXPERIMENTAL RESULTS

To test our ideas about LearnBot and to see in first hand how children reacted to its use as a learning-by-doing tool, we prepared a simple experiment involving eighteen children between 10 and 13 years. The subjects were selected in a summer camp organized by our University. An exercise was designed that consisted in the collaborative resolution in pairs of three problems of increasing difficulty. Sessions began by acquiring some information from the children through a quick questionnaire. They answered a Likert scale (1-7) where demographic data, such as age or completed school year, were collected and simple questions were raised concerning the daily use of technologies and their views on programming and robotics. Afterwards, we gave them a brief introduction to robotics and to the Python language commands that would be used to solve the proposed challenges. The result of the survey shows little knowledge of Robotics (table 1, graphic 1). However, when they were asked about what they think a robot is, they showed a fairly accurate idea. So when they say 'a robot does what it is told to' or 'you have to tell it everything', it is clear that they know or guess that a human must give orders to the robot in order for it to act. You have to program it. Along the same line, we find words related to the cognitive dimension of robot ('think', 'artificial intelligence', 'thinks for itself') which, unlike other machines, gives it the ability to interact with the environment and take actions with some autonomy. It also highlights the functionality of robots: 'to help or to supply labor', 'a machine to serve humans', 'multitasking'; it includes also an element directly related to the human-machine interaction from the social and emotional perspective, when a girl says 'give me company'.
The dominant trend in the spent time solving the challenges decreased as the session progressed. Thus, for example, the first couple took 15 minutes to solve the first problem and only five minutes to complete the second. Also the fifth

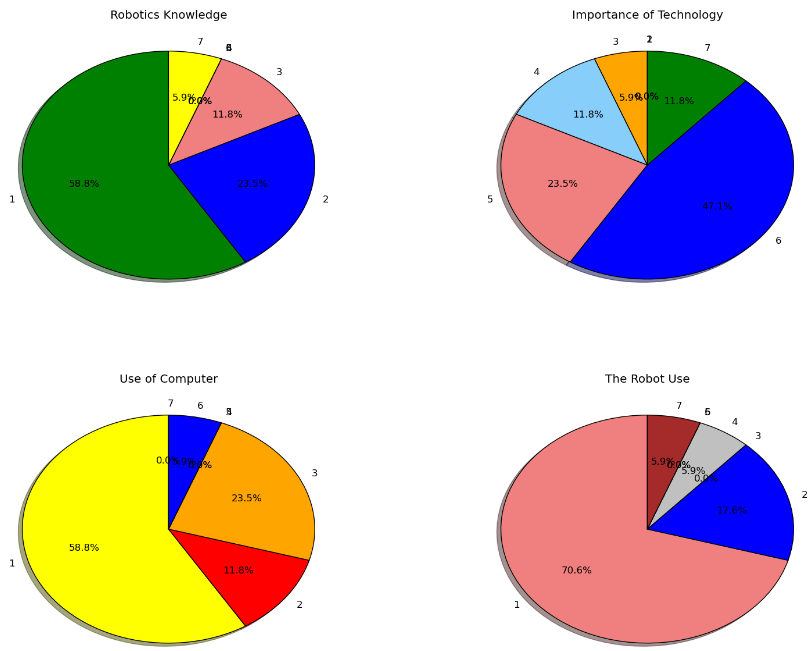

Table 1: In all graphics, 1 means 'I know nothing'; 7 means 'know a lot'. We can see in the first graphic that most of the children haven't had contact with robotics and they don't know exactly what robotics is. However, we can note in the second graphic that children often give much importance to technology. In the third and the fourth graphic we can observe the little computer use and no presence of educational robots in the classroom

couple, who invested 12 minutes in the first proposal and five minutes in the second. The most frequent errors were related to turning angle and the omission of the last command that stops the robot.
As noted, in general the students belief that technology will be a competence to consider in the development of society.

However, as reflected in Graphic 3 and 4 of Table 1, the computer is barely used inside classrooms and even less the educational robots. In contrast, 100% of students use some smart device to access the Internet and receive information electronically, but not to interact with technology.
One of the main consequences that can result from this situation is that students do not acquire all the skills tagged in the European Educational Laws. Moreover, authors such as [15], [16] or [17] defend that programming is the new literacy, because in the process of learning to program, in addition to grasp mathematical and computational concepts, strategies of problem solving, project design, col-

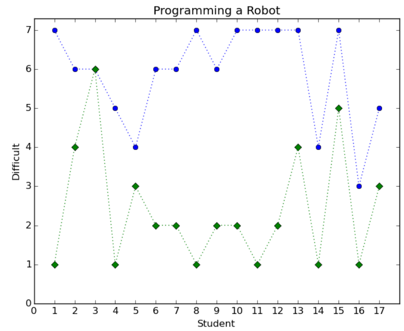

Table 2: Girls studying the problem before programming the solution and the Children difficulty perceived in handling robots, before the programming phase (in blue) and after programming (green)

laborative work and communication are developed, which makes programming be necessary for the citizen in am inexorable digital future.
Finally, it is pleasant to see how the perception of the difficulty of programming robots changes before and after working with Learnbot. Table 2 shows the difficulty estimated per student before and after the process.
The general attitude of the participants and interaction with LearnBot has been positive. Children have expressed interest and motivation, with satisfaction when the execution of the program is correct and they can view the results, although not all couples managed to work collaboratively.

4 CONCLUSIONS AND FURTHER WORK

In this first experience with LearnBot we noticed that the children were able to write some very simple programs with a real programming language in just a few minutes. They went a step further, becoming creators and not only 'consumers' of technology, and a socio-affective link was developed between them and the robot. Although with the data obtained it is not possible to speak of results that demonstrate the development of cognitive skills in the line of computational thinking, since that their acquisition is the result of a process with continuity in time, there has been very unexpected and motivating signs of some of the skills that define it: abstraction, thinking in terms of decomposition and thinking algorithmically.
Experiences with children has put us on the path to some improvements towards a more emotional robot. 'LearnBot' is for them a depersonalized name and they demanded a facial expression, closer and friendlier. These elements, which may seem trivial, are needed to consolidate the bond with the robot, which is essential to generate the motivation, involvement, interaction and socio-emotional relationship required in all teaching-learning process. And that is why the design of

next generation of LearnBot will be based on notes taken during this experience, such as some improvements in the software, changes in the design and colors of the outer cover or inclusion of elements that facilitate the HRI.
Now more experiences are needed in the classroom in order to improve and give validity to Code2Bot. In the mid term we expect to create a true collaborative ecosystem designed to stimulate and facilitate the learning of programming for the entire school community.

References

1. Fabiane Barreto Vavassori Benitti. Exploring the educational potential of robotics in schools: A systematic review. *Computers and Education An International Journal*, 2011.
2. García Méndez J.V. y Rosas Chávez L.A. Ruíz-Velasco Sánchez, E. Robótica pedagógica virtual para la inteligencia colectiva. In *Memoria congreso Virtual Educa.*, (2010).
3. J. M. Wing. Computational thinking. *Communications of the ACM*, 2006.
4. J. M. Wing. Computational thinking and thinking about computing. *Philosophycal Transaction of the Royal Society*, 2008.
5. John Woollard Cynthia C. Selby. Computational thinking: The developing definition. *ACM Special Interest Group on Computer Science Education*, 2010.
6. A. González Ledesma, A. y Redondo Hernández. Aitoy 1, un robot neo-educativo con emociones. *IE Comunicaciones: Revista Iberoamericana de Informática Educativa*, 2013.
7. M. M. McGill. Learning to program with personal robots: Influences on student motivation. *ACM Trans. Comput. Educ. 12, 1, Article 4*, 2012.
8. G. Ocaña Rebollo. Robótica como asignatura en enseanza secundaria. resultados de una experiencia educativa. *Cuadernos del Profesorado. Espiral*, 2012.
9. Albo-Canals J. y Garriga C. Alex Barco, A. Engagement based on a customization of an ipod-lego robot for a long-term interaction for an educational purpose., 2014.
10. Miheon Jo Jeonghye Han and Sungho Kim Sungju Park. The educational use of home robots for children. *IEEE International Workshop on Robots and Human Interactive Communication*, 2005.
11. S. Papert. *La máquina de los niñ os: replantearse la educación en la era de los ordenadores.* Barcelona: Paidós Contextos, 1995.
12. P. Bustos P. Núñez R. Cintas L. Manso, P. Bachiller and L. Calderita. Robocomp: a tool-based robotics framework. In *Proceedings, SIMPAR Second International Conference on Simulation, Modeling and Programming for Autonomous Robots.*, (2010).
13. Romero-Garcés A. Bustos P. Martínez J. Rodríguez, M. Progress in robocomp. *Journal of Physical Agents*, 2013.
14. M. Henning and M. Spruiell. *Distributed Programming with Ice.* 2005.
15. M. Prensky. Programming is the new literacy. *Edutopia*, 2008.
16. M. Resnick. Learn to code, code to learn. *edSurge*, 2013.
17. D. Rushkoff. *Program or be programmed. Ten commands for the Digital age.* New York: OR Books, 2010.