

New Tools and Methodologies for Programming Languages Learning using the Scribbler Robot and Alice

Sebastian Londoño Salcedo, Ana Maria Orozco Idrobo
Icesi University, slondono@icesi.edu.co, ana.orozco@correo.icesi.edu.co

Abstract - Today is often raised the question, what happened to engineering? Each day are less aspiring to such programs, and we need to find innovative ways to motivate students, encouraging learning fun, and that his time, guaranteeing a solid covering topics that are usually evaluated in the first programming courses, some of the tools that were used during the experience at the University Icesi for teaching programming and Alice were Scribble robot, the which allowed students to interact with the fast world of programming. The teaching methodology had proven that through it the student can interact with a friendly interface that let users explore and use the concepts seen in the programming courses, encouraging students to improve their capacity through algorithmic understanding. Working with new strategies in the teaching of computer courses, is a fundamental pillar in the education of future engineers, given that capacity building fosters algorithmic analysis and problem solving skills as required by engineers.

This paper will present the progress and results obtained using different programming interfaces and robots, as a complement to the basic algorithms and programming languages. It also will review the differences between traditional teachings vs. new tools for learning.

Index Terms - Scribbler and Alice, Lego NXT, ROBOTC, Microsoft Robotic Studio

INTRODUCTION

Learning of programming and development of algorithmic thinking plays a role in engineering education, especially in engineering careers systems, telematics, electronics, robotics and related fields. One of the biggest concerns by the faculties of engineering is the reluctance of students into programming language materials, and results in the drop by students in these courses or even by the undergraduate program, likewise, has seen a dramatic decrease in the number of students who enter such university programs, as shown in the following figure 1, in the case of the city of Cali, Colombia.

In the context of this problem has been the need to implement more attractive methods in the teaching of programming, because usually the learning process was

carried out by traditional software development exercises, readings, typically a solution to problem exposed, such as: payroll, accounting, grades, etc., as well as memorizing the syntax of specific language and conceptualization of subjects required in programming.

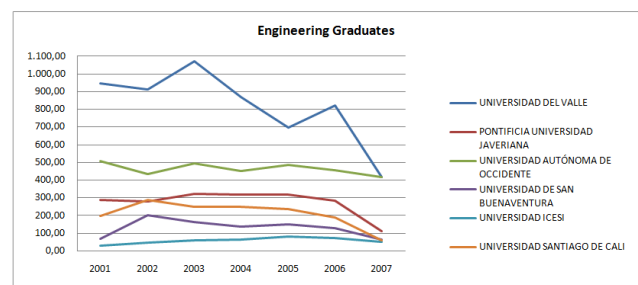


FIGURE 1
STUDENTS ENROLLED IN ENGINEERING PROGRAMS IN CALI
Source: Labour Centre - Ministry of Education [1]

ALICE AND SCRIBBLER AS TOOLS FOR TEACHING BASIC PROGRAMMING

Recent research has demonstrated the validity of using new methodologies in the teaching of programming [2] [3] through graphical interfaces, programmable hardware, robots and API's (Application Programmer's Interface) friendly that enable students to build and develop concepts with programming. The introduction of interactive tools in the teaching of programming as a first approach to languages, computer science, has shown an improvement in student performance and perception towards them. The statistics obtained in studies conducted at Ithaca College and Saint Joseph's University(United States) revealed that the notes increased significantly and retention of students increased from 47% to 88% [4].

The potential of teaching tools and adaptable to the curricula of the courses make it possible for the implementation and development of workshops with students as a first approach to programming world. Figure 2 shows an example of programming graphical interfaces, where students intuitively constructed algorithms and programs in a simple, visually appealing, the programming becomes an enjoyable task and also enables students to acquire knowledge basic programming.

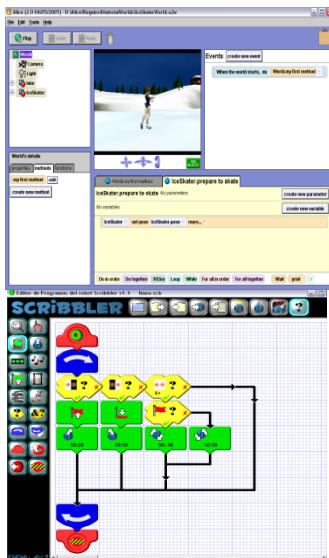


FIGURE 2
ALICE AND SCRIBBLER PROGRAMMING INTERFACE

ALICE

Alice is the result of the project, "Learning to Program with Alice" (learning to program with Alice) of Carnegie Mellon University [5] (United States) which aims to bring young people in the world programming, through a completely visual and animated in three dimensions. Programming with 3D animation is a high visual impact and change the programming paradigm as a monotonous and uninteresting. The concepts and syntax of programming languages are acquired through practical exercises and interaction with the tool. Students can view many of the fundamentals of object-oriented programming, such as full graphical representation of an 'object', the behavior of a repetitive statement, just to name a few, so that at times the concepts are learned intuitively through exercises developed in Alice. Alice interface enables the development of comprehensive programs by dragging items from one panel to another thereby generating the code for such methods, attributes and objects in general. Importantly, students will not have to deal with the frustration that can lead to coding errors and implementation, which is one of the factors that influence the motivation of students in programming courses. Figure 2 shows the graphical user interface programming Alice.

Experience with Alice

With respect to Alice, were designed at the University Icesi 15 lessons covering the fundamental issues of programming using Alice, for each lesson was designed a video-phone and a workshop, the videos are made by way of tutorials and cover specific topics Each video has 3 parts:

- **Theoretical Introduction:** concepts are explained clearly, are examples graphs, charts or diagrams.
- **Step by step exercise:** developing a program to apply the concepts studied, in which they explain each procedure to follow to get the result.

- **End result:** it makes a small demonstration of the completed and ready for testing.

Once the student has seen the video, you can go on to develop the proposed workshop. The workshops aim to strengthen the concepts seen in the tutorials, for that matter consists of a short questionnaire about the theory and a practical exercise was developed by applying the lesson. For the preparation of this material is taken as a reference for programming courses at the University Icesi: Algorithms and Languages I, Algorithms and Languages II and tutorials "Alice Programming Tutorial" by Richard G Baldwin. [6]

SCRIBBLER

Scribbler is a fully programmable robot with multiple sensors that allows interaction with people and objects around him, is able to navigate by himself, explore his surroundings and report the information obtained through the sensors available (lights, sounds, objects). Among the Scribbler robot components are: light sensors, object detectors, detectors of lines, independent motors, indicator lights, a speaker to reproduce a range of sounds and musical notes, a serial port for communication with the computer, etc. [7]. The Scribbler robot graphical interface allows programming in a completely visual and friendly environment; in addition the robot can be programmed through the traditional programming languages: Basic, Python, and Java, among others.

Experience with the Scribbler robot

The Scribbler interaction allows students to have a different learning from traditional programming, through the use of instructions and programming structures represented by icons, an algorithm is constructed by interconnecting the blocks of programming, allowing students to modify the behavior of the robot.

Another decisive factor for using this type of tool is the cost (100 USD), which allows it to be purchased by educational institutions as a teaching tool for students, without incurring high costs in implementing this teaching strategy [8].

NXT TECH VIRTUALS ROBOTC WORLDS

ROBOTC is an environment based on C programming for robotics developed by the robotics department of Carnegie Mellon University (USA) [5], ROBOTC is a premier educational programming language for robotics, with an interactive environment that allows development simulated by a 3D engine, robots, scenes and physical conditions quite close to the reality. Among the platforms supported are the most popular Lego Mindstorms NXT and Vex robots innovation. The environment has a debugging system that allows students to observe the robot's behavior both in the 3D simulation and in code, as the simulation environment will be developed step by step, pointing out some code that is running. The programming interface allows rapid adaptation to the C language to see that the syntax allow translates into visible events in the simulated robot [9].

Experience with ROBOTC

With respect to ROBOTC were designed at the University Icesi 6 lessons that address fundamental issues of programming using ROBOTC, were developed for each lesson videos, presentations and guidelines that were composed:

1. **Practice objectives.** It explains the concepts discussed in the section and issues seen in the lectures will be applied to practice.
2. **Transition.** It explains the concepts studied in class are translated into actions and events in the 3D simulation
3. **Exercise guidance.** It explains how to apply the concepts studied in class through a problem which is solved step by step showing how to develop it in its entirety.
4. **Exercise.** It places a specific problem the student must resolve through the learning session.

The exercises seek to strengthen the concepts seen in class; the student has a positive online learning as supplements in a practical and creative teaching the class [10].

LEGO NXT 2.0

The Lego NXT 2.0 is a kit used to build robots from Lego Mindstorms line. It contains 619 pieces and identification key number is 8547. Unlike its predecessor, this set is released RGB color sensor which allows the distinction of Red, Green and Blue (RGB). The IDE contains the Mindstorm has basic editing capabilities of sound, recording and playback, as well as basic editing of images to be displayed on the brick. Additionally, this version has included Lego Bluetooth connectivity allowing control of the robot from a computer. Contains two pressure sensors that act as an ultrasonic touch and let you know with a value of distance between him and an object.

Lego NXT Experience

The development of activities with the Lego NXT enables students to active learning to put theory from the classroom to practice in the computer room, for them to see materialized programming concepts like variables, conditional or repetitive structures generated a consolidation of concepts and an understanding that in the event that they only had the material covered in class [11].

In the case of Lego skills were developed that allowed the teamwork, the students formed working groups to build robots that in many cases takes time and dedication. Achievement is evident in the groups had a leader who was responsible for delegating tasks. In the case of humanoid construction of Lego, the team manager delegated to each student to build a tip to finish in less time the design and implementation of the robot.

MICROSOFT ROBOTIC DEVELOPER STUDIO 2008 R3

Microsoft Robotics Developer Studio 2008 R3 (RDS) is a windows based platform for academics, hobbyists and commercial developers to create robotics applications

(hardware and software) through a variety of predefined hardware.

Microsoft RDS provides a lot of supports for the development of robotic applications easy. Microsoft RDS includes a programming model that allows for the possible development of asynchronous applications and autonomous management. RDS provides a framework that can support a number of robots allowing editing of code between them is heterogeneous. Robotics Developer Studio has been known for programming in VPL (Visual Programming Language) that allows the full development of behavior and simulations based on rapid prototyping that is the union of graphic objects that are reflected in the syntax used in the 3D environment or hardware. Being based on the platform of Microsoft .Net uses part of the XNA graphics engine used for the generation of VSE (Visual Simulation Environment) which is what provides the ability to simulate and test robotic applications using a 3D physical tools. Students can create robotics applications without hardware, simulation and modeling of robots that already has the software can test applications on a variety of 3D scenes [12].

As an experience with Alice, the university conducted the workshops "Icesi BotChallenge" with the Scribbler robots, Lego and besides the simulations of ROBOTC and Robotic Studio, in the first sessions of these workshops (basic) addressed the key issues for the use of different platforms, in which students are familiar with the programming interface, in the following sessions (intermediate) deepened more complex programming issues and the formal structures of language programming (such as subroutines, loops, etc.). In addition to this, students designed algorithms to solve given problems. In the final sessions (advanced level) were carried out practices that included physical 3D simulations, asynchronous network programming and other issues that require a higher degree of knowledge. The courses were designed for students to have a line of learning according to their level of development in programming and knowledge [13].

From the experiences in the university based on the use of teaching tools (software and hardware), you can make a comparison between these and traditional teaching computer courses. See Table 1.

RESULTS

One of the results with students after the workshops of the different platforms was to raise the interest in programming, earmarked workshops were held with success since the environment of the classes allowed an adequate job. In the graphs of Figure 3 shows that the use of these tools was well received by the student population. From Graphic A in Figure 3, suggest that students reach their programming courses with a negative view towards the subject, above, 58% of respondents found it difficult to schedule, for 92% of respondents their experience with the lessons was good (Graphic C), compared with 75% (see Graphic B) that it considered fair or poor experience with other programming languages (Java and C++) [14].

FIGURE 3
SURVEY RESULTS SUMMARY

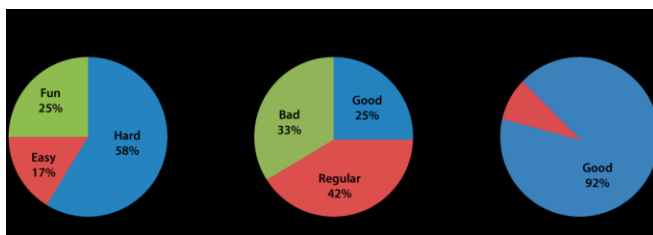
TABLE 1
COMPARISON OF TEACHING METHODS

Comparison Criteria	Traditional teaching	Teaching using didactic tools
Prerequisites for the development of programs	Must have knowledge of structures and programming languages by the students.	The graphical interface element allow easy recognition of basic programming structures, and in turn hides low level details to facilitate the use of tools.
Debugging and Error Handling	All elements of the program are developed and used by the user. You need to memorize sentences and syntax of programming languages.	Programming can be carried out 'drag and drop predefined commands with the mouse to create more complex programs.
Programming environment	Programming environments are mostly unattractive since they consist of code.	The graphics are visually appealing interfaces for students.
Documentation	There are many books in support for different languages, with different cost and complexity.	There are books of authors to support platforms. [8]
Motivation for programming languages	Low due to the complex development environments and unfriendly in programming.	High motivation and providing greater interaction and visible results, such as animation and 3D modeling and management of the robot.

As to the scope of the issues of programming, the tools allowed include the contents shown in Table 2:

TABLE 2
COMPARISON OF TOPICS COVERED.

Topics covered	Alice	Scribbler	LEGO NXT 2.0	ROBOTC	Robotic Studio R3
Basics					
Download and Installation Guide	✓	✓	✓	✓	✓
Object Concept	✓	✗	✓	✓	✓
Methods	✓	✗	✓	✓	✓
Attributes	✓	✓	✓	✓	✓
Variables and types of variables	✓	✓	✓	✓	✓
Class concept	✓	✗	✓	✓	✓
Inheritance	✓	✗	✓	✓	✓
Sequential structures	✓	✓	✓	✓	✓
Repetitive structures	✓	✓	✓	✓	✓
Selection structures	✓	✓	✓	✓	✓
Expressions and logical operations	✓	✓	✓	✓	✓
Arrangements and lists	✓	✗	✓	✓	✓
Events	✓	✓	✓	✓	✓
Routines and subroutines	✗	✓	✓	✓	✓
Programming and management of sensors	✗	✓	✓	✓	✓
Advanced					
3D simulation	✓	✗	✗	✓	✓
Distributed services	✗	✗	✗	✓	✓
Own framework SDK	✓	✗	✓	✓	✓
Simulation by network	✗	✗	✗	✓	✓
Parallelism	✗	✗	✗	✓	✓
Third-party frameworks	✓	✓	✗	✗	✓
Generation of physical models	✓	✗	✗	✗	✓
Online community	✓	✓	✓	✓	✓



The workshops were attended by students with some prior knowledge of programming in Java or C + +, to interact with a friendly interface and put aside problems of syntax in writing code, students moved into more complex issues and developed a lot more trouble developed based on the theoretical and the explanations given in the workshops. On the other hand, those students who had no knowledge of programming, the tools found in an understandable and

manageable environment that allowed the writing and display of its first programs [15].

Scheduled workshops, developed based on tools, provide an excellent approximation to the fundamental concepts of programming because they have been developed in a very detailed, elementary and concise so that young people can learn and / or strengthen its theoretical basis in a simple and fun.

CONCLUSIONS

One of the commitments they have the Faculty of Engineering is to contribute to the development of the country, which is only viable from highly qualified and competent staff that can respond to the challenges of the current situation, one way of ensuring this contribution is to contribute in changing the perception that today's youth toward engineering, providing tools that enable students to become familiar with the concepts of computer science in a friendly way and incremental [16].

It is possible to motivate and encourage engineering students to the 'world of programming' using visual tools, which allow easy understanding of theoretical concepts and above all have a pleasant experience and tangible results. The various programming tools offer great potential for teachers of programming. Tools like Alice can become an element for strengthening the theoretical concepts seen in the classrooms and labs students can do, whereas Lego Scribbler and students can view your code become actions that are executed by the robot [17].

ROBOTC and Microsoft Robotic Studio allows students to go beyond the examples and guidelines can understand complex issues typically have a long learning curve and turn them into interactive learning process to move forward and understand issues with applications in other areas. It should be noted that experience with those two tools shows that when the concepts are explained through interactive tools and platforms the student more easily understand concepts and be able to abstract the concept and apply it to other cases.

Both basic and intermediate tools and advanced can be implemented in all semesters at universities and in the upper grades for secondary education. Having learned the basic management tools, the student will understand the fundamental ideas of programming. So you can make the transition from any of the mentioned platforms to actual languages and programming environments, and involving elements such as the syntax of the language and especially the development of algorithmic thinking.

Advanced tools allow you to transition from basics to intermediate to advanced concepts such as network programming, 3D simulation, actual conditions, and distributed systems, among others. The really important thing of the latest tools to handle concepts is highly applicable to real life that allow a more transparent way for the student when it desires to both professional modeling tools, and programming and 3D simulation.

The workshops help to strengthen the concepts learned and the development of algorithmic thinking, which is reflected in the ability to analyze and design a solution to a problem.

With the support of intuitive graphical tools and manipulation of them by students, it is possible to capture and hold their attention longer than the same time having fun programming also obtained new foundations in the various algorithms and programming structures.

REFERENCES

- [1] Education Centre for Education. Colombia Ministry of National Education
<http://www.graduadoscolombia.edu.co/html/1732/articloe-144482.html>
- [2] Mansi Gupta, Marwa Nur Muhammad, Shikha Prashad. Robots Byte In: An Exploration of Computer Science Education in Middle Schools. Bryan Mawr College.http://www.roboteducation.org/files/CREU_Final_Report.pdf
- [3] Londoño Guillermo, Adolfo Paz Gustavo "Basic programming for adolescents", *book*. SyT Icesi University 2007.
- [4] M. Moskal, D. Lurie, and S. Cooper. Evaluating the Effectiveness of a New Instructional Approach.
- [5] DannStephen Wanda Cooper, Randy Pausch. Learning to Program with Alice. Carnegie Mellon University, www.alice.org
- [6] Richard G. Baldwin Learn to Program using Alice.<http://www.dickbaldwin.com/tocalice.htm>
- [7] Features Scribbler Robot.
<http://www.parallax.com/tabid/455/Default.aspx>
- [8] Learning Computing With Robots.
http://wiki.roboteducation.org/Introduction_to_Computer_Science_via_Robots
- [9] ROBOTC, a C Programming Language for Robotics. Carnegie Mellon University, www.robotc.net
- [10] ROBOTC Curriculum for the NXT and Tetrax
<http://www.robotc.net/education/curriculum/nxt/>
- [11] Floyd Kelly, James, "Lego Mindstorm NXT-G", Second Edition 2010
- [12] Morgan, Sara, "Programming Microsoft Robotic Studio", 2008, Microsoft Press
- [13] Tzafestas, Spyros G., "Web-Based Control and Robotic Education", 2009, Springer
- [14] Iskander, Magued, "Innovative techniques in instruction technology, e-learning, e-assessment, and education", 2008, Springer
- [15] Aggarwal, Anil, "Web-Based Education: Learning from experience", 2003, Information Science Publishing

- [16] Sande Carter, Sande, Warren, "Hello World!: computer Programming for kids (and other beginners)", 2009, Manning Publications
- [17] Kahn, Ken, "ToonTalk – Concurrent constraint programming for kids", 1995, Logic Programming, proceedings of the twelfth international conference on Logic Programming

AUTHOR INFORMATION

Sebastian Londoño Salcedo, Researcher, Icesi University, slondono@icesi.edu.co

Ana Maria Orozco Idrobo, Researcher, Icesi University, ana.orozco@correo.icesi.edu.co