

# Incorporation of "Kinesthetic Learning Activities" to Computer Science 1 course: Use and Results

**Inés P. Friss de Kereki**

ORT Uruguay University  
Cuareim 1451, Montevideo, 11100, Uruguay  
*kereki\_i@ort.edu.uy*

## **Abstract**

In this work, we present the systematic incorporation of Kinesthetic Learning Activities (activities with movements and sensations), into the Computer Science 1 course. We detail the different activities and their use in the course. We evaluate student results in relation to their final grades in the course. Moreover, we include opinions of students that participated in the experimentation. Based on students' results, we appreciate a higher degree of course achievement, that is, students who achieve an exceptional result (86% or more points) in comparison with traditional courses.

**Keywords:** Programming, Learning, Teaching, Object Oriented Programming, Kinesthetic Learning Activities.

## **1 INTRODUCTION**

This work intends to improve the processes of Programming teaching and learning, particularly Object Oriented Programming (OOP). In ORT Uruguay University (abbreviated to ORT in the following), the OOP paradigm has been taught since 1996 in Computer Science 1 (CS1) courses. A standard course consists of lectures and practical lab classes. Here we present the systematic incorporation of Kinesthetic Learning Activities (KLAs) into the lectures.

Firstly, we describe some aspects of the CS1 course and describe KLAs. After that, we present details of the CS1 course in ORT Uruguay and the incorporation of KLAs. We also include the experimentation and the analysis of final results of students' works and their perceptions (obtained through a personal survey). Finally, we offer some conclusions and ideas for future work.

## **2 COMPUTER SCIENCE 1 COURSES AND KLA**

Learning is an active process, not a passive one. Students, in order to gain optimal level of competence, should be motivated to be involved actively in the learning process, in activities which include, for instance: solving real problems, producing original written work, building objects and dialoguing with other people about important topics [1].

Programming is a complex activity. It is important that the CS1 course be correctly designed [2]. Robins, Rountree and Rountree [3] say that learning to program implies to gain a completely new knowledge and a series of strategies and practical abilities. Gries [4] says that it is necessary to revise seriously how to teach programming. The goal of education is not only to expose facts to students, but teaching how to think. Gries, in [5], mentions some principles applicable to teaching, among them presenting concepts with the correct abstraction level.

The analysis of CS1 courses may be done from different approaches, not necessarily disjoint. This analysis may include the main focus of the course, tools to be used and methodologies applied. As to the main focus, we may consider, for instance, test driven development [6], [7], [8], software engineering [9] or pair-programming [10]. As to which tools may be used in the course, Powers and colleagues [11] classify them in categories. Some of them are: a) narrative tools: they allow to program in order to tell a story; b) visual tools, which allow to construct programs by drag and drop elements; and c) tools based on modeling the flow of the program, which connect elements to represent that flow (and thus reduce complexity). Finally, related to concrete methodologies, Dugas [12] points to the use of a daily quiz as an active and cooperative learning technique. Each quiz aims to illustrate the objectives of the class and to serve as a guide to the development of further topics. Dugas's study shows that students achieved better grades in that course. Sung et al [13] promote the integration of video game and graphical programming.

Another tool related to the methodology and with emphasis on active learning are the KLAs. A KLA is a pedagogical tool which involves physical movement of the students [14]. The basic characteristic of this kind of activity is that students are actively and physically participating in the exposition and assimilation of the class

material. This participation is directly related to certain specific learning objectives. The KLAs promote interaction between the students and improve learning. Sivilotti and Pike [14] give several recommendations for the design, creation, adaptation and execution of the different KLAs. For example, they refer to the inclusion of surprise elements, involving the whole group and giving out simple instructions. Begel et al [15] point out that in general the KLAs are brief (20 minutes or less) and that they can involve some or all of the students. They can be used in small or large groups. Astrachan [16] shows examples that could be interpreted as KLAs; one example models the parameter passing among functions using flying discs. Astrachan emphasizes that these kind of exercises show a physical image that helps students build mental models. Anewalt [17] includes in the course several KLAs and reports that they boost students' interest, so that students come "awake and curious" to the class, wondering what will happen.

From the bibliographic review and analysis it can be deduced that there is an opportunity to collaborate on the search of elements that promote comprehension, abstraction, development of skills and the development of problem solving strategies for the CS1 course. Considering an active learning approach, the KLAs can be a helpful tool.

### 3 CS1 COURSE AT ORT

The CS1 course at ORT is aimed at teaching programming using OOP techniques. Its contents include basic OOP concepts and their application in a programming language, currently Java. At the end of the course, the student will be prepared to analyze simple situations, to design a possible solution and to implement it using an object-oriented approach. The course consists of 15 weeks, each week including 4 hours of theory and 2 hours of lab practice. The course details are shown in Table 1.

**Table 1.** CS1 schedule and main contents.

Week	Contents
1-3	Control structures, pseudocode and variables.
4	Classes and objects, standard classes.
5-8	Creating new classes, handling relationships and aliases.
9	Inheritance.
10-12	Collections, sorting and searching.
13	Exceptions.
14-15	Advanced use of collections.

Students must accomplish three tasks satisfactorily to approve the course: two programming projects on a computer (in groups of two students) and an individual test. The first project implies the development of a small system (3 to 4 classes). The second one is a further development on the first work. In the second project, other classes and processes must be developed, with an interesting algorithmic challenge. For each work there is an individual defense. In the defense the student must implement correctly a specific change in the task. The first project is worth a maximum of 25 points, the second one 30 and the test 45. If the student achieves 86 points or more, he or she passes the course and does not have to take a final examination. Between 70 and 85 points, the course is approved but the student must take the final examination. With less than 70 points the course is failed and must be retaken. This does not allow the student to take the following course (Computer Science 2).

During the course, 10 different KLAs were used, at a rate of up to one per week. Each KLA takes between 15 and 20 minutes. The recommendations mentioned before ([14], [15]) were considered for the design and selection of the KLAs for this course. Also, emphasis was made in considering the different learning styles [18]. Table 2 shows the denomination of the KLAs and the course weeks in which they were performed.

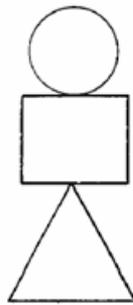
**Table 2.** Summary of KLAs

Week	KLA	Week	KLA
1	1- Drawing	7	6 - Isomorphism
2	2- Ice Cream	8	7 – Changes
3	3- Shuttle Puzzle	11	8 – Sorting
5	4- Modeling clay	14	9 – Algorithms
6	5- Roleplaying game	15	10 – Jeopardy

#### 3.1 KLA 1 - Drawing

The KLA 1 includes a set of big paper sheets with drawings. Each sheet must be orally described to another student, who must reproduce the drawing on the whiteboard without seeing the original. The sheets include, for example, a circle, a square and a triangle drawn one on top of the other (see Figure 1), or concentric circles. The student that

draws simulates being a computer, and the one describing indicates the instruction to be executed. The activity tries to show the necessity of being precise when giving out instructions, that is, when making an algorithm.



**Figure 1.** Drawing

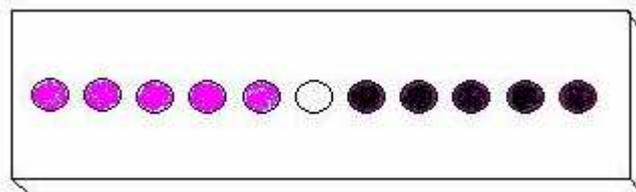
### 3.2 KLA 2 – Ice Cream

For the development of large systems, teamwork is a must. In ORT, this skill is promoted from the beginning of university studies, since all group projects are performed in groups. As Goleman [19] says, being excellent in current times depends more on emotional competences than in cognitive capacities. He refers to social skills (the capacity to induce desirable responses on other persons) and team skills (to create group synergy towards achieving the group's goals). The proposed exercise for this skill (based on [19]) is the KLA named “Ice Cream”.

This KLA asks students to “indicate which is the most tasty ice cream flavor in the world”. The exercise is presented in two stages: a) indicating the work rules, and b) formulating the question on the whiteboard. The rules are that a question will be formulated, of which no further info will be available and that each team of 4 or 5 students will have 10 minutes to give a group answer. It is recommendable to listen to each of the group's debates and take notes of the comments made, since these will be the base of the posterior discussion. When the time is up, each group announces their answer. Some groups may not have achieved consensus. At the end of the activity, each of the comments made are described and discussed, showing their strengths and weaknesses. Some examples are: “it is necessary to establish a criteria to know what is the best flavor” (which refers to establishing a working methodology), “let's vote the best flavor” (a solution by consensus), “let's split the group and give 4 answers” (which contradicts the work rules and generates conflict), “let's pick vanilla flavor because I like it” (lack of tolerance), “Chocolate? Who said it was tasty?” (an attitude that generates negative reactions), and “I could not choose my favorite flavor” (lack of participation of a student in the group). These attitudes are associated to problem solving as a team.

### 3.3 KLA 3 – Shuttle Puzzle

This KLA has as an objective to present the idea that for developing an algorithm that solves a problem, you must first clearly understand the problem. There are multiple ways to solve a problem. Among the best known methods there are Descartes's [20] and Polya's [21]. Gray [22] proposes the “Shuttle Puzzle” problem as a vehicle to present the process of analyzing a problem and synthesizing observations into an algorithm in a detailed and step-by-step manner. Using this puzzle as an example, a list of recommendations for solving problems can be given out. The puzzle consists of 4 marbles or sticks of one color, 4 of a different color and a wood block with 9 holes in line (see Figure 2). Each group of sticks is set on the extremes, leaving the center hole empty. The objective is to invert the order of the marbles, following the rules of: a) move a marble one step to an adjacent empty space, and b) jump over a different color marble to an empty position.



**Figure 2.** Shuttle Puzzle

A set of marbles or sticks is given out to teams of 3 to 4 students, the rules are explained and the students are asked to solve the problem. This process is watched and notes are taken of what the students are doing. It can be observed that few students solve the puzzle, others detect that they have not understood the rules and abandon trying to solve the puzzle, and a few others generate ideas for new problems based on the original. Some of the problem-solving strategies observed are: trial and error, problem simplification considering 2 or 3 marbles instead of 4, and attempts

to deduce general rules. As a conclusion to the activity, the general observations are discussed and the problem solving recommendations are written down on the whiteboard. These recommendations combine and summarize the general methods presented: a) understanding the objective, b) understanding the rules, c) evaluating the alternatives (simplify the problem, trial and error, analogies, etc), d) solve the problem, and e) verify the solution.

### 3.4 KLA 4 – Modeling clay

This activity is focused on improving the comprehension on the difference between object and class, and to become familiarized with the concepts of identity, objects aliasing and message passing. Modeling clay and molds is given out to the students and diverse objects are created [17]. Each mold represents a class. Each molded element is an object. The proposal in [17] is extended by representing the association of a variable (as defined in Java) written down on the whiteboard to an object by using a string from the word on the whiteboard to the object molded. Creation, assignment, object identity and message passing is shown. Also, the concept of alias is shown, when a variable “looks at” the same object (represented by another string going to the same molded object). See Figure 3.

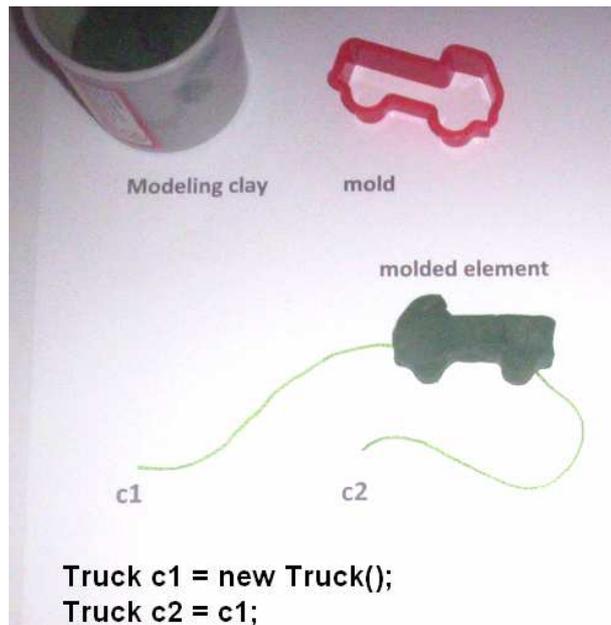


Figure 3. Object creation and aliasing

### 3.5 KLA 5 – Roleplaying game

This tool is an adaptation of [23]. The goal is to present basic objects concepts in a real application; in particular, to understand messages and methods. The classes are: Standard Calculator (which knows to add and to subtract), Creative Calculator (knows to add and subtract, but has a special subtract method), Bureaucratic Calculator (uses methods of standard calculator in its own methods), Acrobat (which knows to stand up, to sit down and to applause n times), and a Dice. In the walls of the class some big posters with the messages of each class are posted. Some students are selected and each one represents a role. Each student must represent physically the execution of the method. Message, method, polymorphism, parameters passing, class and instance differences, private variables, are some of the topics covered by this exercise. For instance, if Mary is one of the acrobats, when she receives the message “Mary applause 3”, she must applause three times. If after that she receives “Mary applause 2”, she must applause 2 more times. When the order is “Mary count”, she must say “5”. If the order is “Mary applause” without a number, she must say that she does not know the message because the signature is different. If we choose John as another acrobat and we say “John applause 2”, he applauses twice. If we say “John count”, he must say “2”. In this example Mary and John are instances of Acrobat’s class. The message is “applause”, the parameter is the quantity. Each acrobat has its own private count.

### 3.6 KLA 6 – Isomorphism

The mathematical concept of morphism establishes a correspondence between different domains where the properties are maintained. Morphisms are more detailed and specific than the usual notion of abstraction. The use of morphisms allows treating different things as the same and taking the substitute image for the real one. Use of

morphisms as a tool might help to develop mental models and to give metacognitive support. All of these ideas are presented in class.

KLA 6 is an assignment to solve a problem isomorphical to Tic-Tac-Toe. A map is given to each pair of students and they must paint it according to specific rules. The goal is to establish an isomorphism as a tool for solving problems. In particular, the exercise proposed is "JAM" [24]. JAM contains some numbered paths between cities (see Figure 4). The main goal is to paint with the same color 3 paths which arrive to the same city. Each player has a different color. In his or her turn, the player paints the full path (path with the same number). Some students said that "it is similar to Tic-Tac-Toe", "losing in this game depends on your having a distraction or not, it is analogous to Tic-Tac-Toe"; "it is a strategic game, you must anticipate your opponent's next move". The application of morphism concepts is observed in this particular problem.

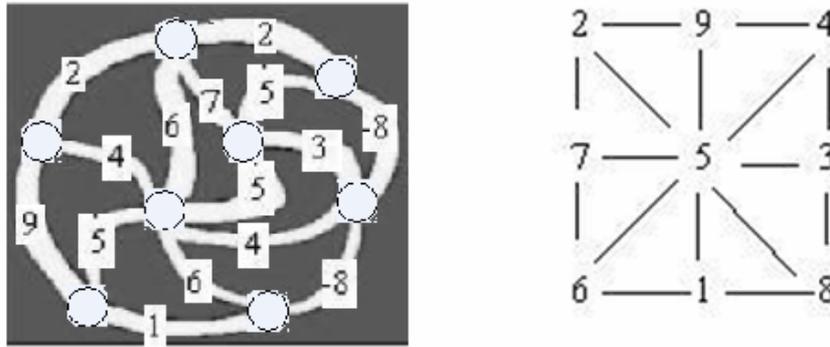


Figure 4. JAM and Tic-Tac-Toe

### 3.7 KLA 7 – Changes

This activity is oriented to manage changes during software development. Students manipulate wooden puzzles. With these puzzles, they must form a certain shape, and after that, the requirements are modified.

This exercise promotes the evaluation of different ways to solve a problem and the need to take into account a possible redesign. In the exercise proposed by De Bono [25], he asks to build some regular geometrical shapes using certain numbered pieces. A set of numbered pieces is given to each student. The goal is to form a recognizable geometrical shape by only using pieces numbered 1 and 2. After that, he or she must build another regular shape including piece number 3. Then, pieces 4 and 5 should be included to form a new regular shape. The last step is to include piece number 6. In most cases, with the previously formed shape it is not possible to include this piece. For instance, if the student begins with a square (as in left side of Figure 5), it is not possible to include the last piece. It is necessary to go back and redesign the solution. Although in each step the process was correct, it requires a revision to include the new piece (as shown in right side of Figure 5). As a conclusion of the work, we discuss about problem solving techniques.

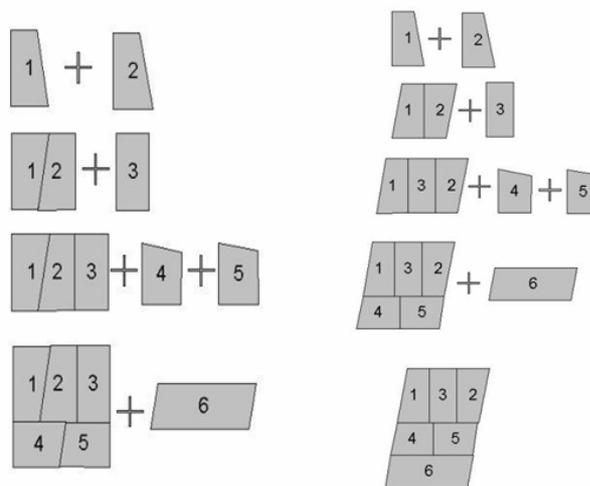


Figure 5. Constructing shapes incrementally with different piece shapes

### 3.8 KLA 8 – Sorting

To present sorting ideas, this KLA consists in distributing some magazines to the students. They must manipulate and sort them. In first place, they are asked to sort magazines according to specific criteria. We stress the idea of discovering some sorting algorithms (for instance selection sort, bubble sort, enumeration sort). In a subsequent stage, in order to present the idea of a general sorting algorithm and a "separated" sorting criteria (in Java: Collections.sort method and Comparable interface), one magazine is distributed to each student and individually, he or she is told of the criteria. Another student "simulates" the sorting algorithm, and must ask to each pair of "magazine-student" (according to the sorting algorithm sequence) if a magazine goes after or before another magazine. According to the answers, the student who is simulating the sorting algorithm reorders the "magazine-students" in the list. This example could be expanded to include the notion of Comparator, this is a student who is capable of receiving two magazines and answer if the first is less than the second, equal or greater. In this case, one student simulates the algorithm, another student represents the "Comparator" and the rest of the students hold each one a magazine. The interaction between the sort and the Comparator is presented.

### 3.9 KLA 9 – Algorithms

In this KLA, based on [26], a set of cards is arranged over the whiteboard in a rectangular grid. Some cards are stuck to the whiteboard face up, others face down. Initially, a student is asked to stick them as he or she wants. After that, with the excuse that the grid size is small, some cards are added by the teacher. These last cards are added in such a way as to complete even parity in each column and each row. The teacher walks away from the classroom and a student must turn one card. When the teacher comes back, he or she "magically" detects the changed card (see Figure 6).

This exercise allows presenting the concept of parity. Some algorithms of traversing binary arrays could be developed to check if the parity is even or not. In the first versions, most students used a counter, which was increased by one with each 1 detected, and finally they check the parity of this counter. Then they are asked to develop a new version which must not use a counter. This allows reinforcing the idea of treating parity, as a Boolean variable, and shows that changing its value with each "1" is enough for the process. Also, this exercise foments the development of different algorithms to solve the same problem.

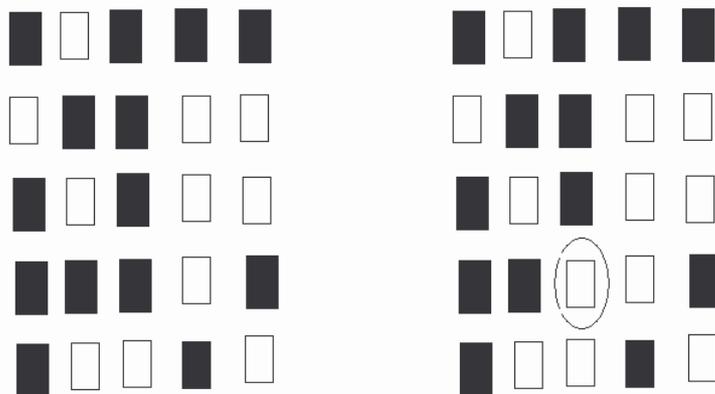


Figure 6: Card Grid (even parity grid)

### 3.10 KLA 10 - Jeopardy

"Jeopardy" is a game with several topics, each with 5 or 6 questions of ascending worth [27] (See Figure 7). In this course, some selected topics were: Java, Languages, ORT trivia and Acronyms. The implementation of the game could be done with big cards, each one with a question, or to use a specific program which manages the game. Two or three teams of students are formed. In each round, a student selects a topic and a worth value, and must answer that question. In case of a correct answer, his or her team wins a point. This exercise is oriented to review topics. Some questions are particularly interesting and could generate a discussion [15].

Course	Acronyms	Languages	Java	ORT Trivia	Contest
200	200	200	200	200	200
400	400	400	400	400	400
600	600	600	600	600	600
800	800	800	800	800	800
1000	1000	1000	1000	1000	1000

Figure 7: Jeopardy

#### 4 EXPERIMENTATION AND RESULTS

The experimentation took place during the second semester of 2008. Three random groups were selected: one for testing and two for control. Students were randomly assigned to one of the groups. Each group had 20 to 22 students. In order to evaluate the experimentation we used course outcomes (final grade giving three possibilities, (i) fail, (ii) pass with final exam and (iii) pass without final exam). It must be said that to grade each test all instructors use the same guides. Moreover, each test is reviewed by 2 teachers independently. The teachers of the 3 groups were experienced. Class materials were the same for all groups [28]. KLAs were used only in the experimental group. The results are given in Table 3.

Table 3. Computer Science 1 results

Group	Control I	Control II	Experimental group
Enrolled (EN)	22	20	22
Failed (FA)	4	7	5
Pass (PA)	18	13	17
<i>Pass with Exam</i>	12	10	3
<i>Pass without Exam (EX)</i>	6	3	14
% failed (FA*100/EN)	18%	35%	23%
% pass (PA*100/EN)	82%	65%	77%
% pass without exam (EX*100/EN)	27%	15%	64%

The percentage of passing grades was relatively similar in the three groups (between 65-82%). We observed a particularly high percentage of exam exoneration (64%) in the Experimental group, in comparison with 27% and 15% in the Control groups.

Surveys realized to the Experimental group students included questions about the usefulness and advantages of the activities. In particular, they were asked about some of the activities and their general opinion. For instance, about KLA1, they answered: "it is an interesting dynamic and helps me to understand", "very good, it appeared as simple but it was complex to put into practice", "it allowed me to make a parallelism between some thinking processes".

Related to KLA4 (modeling clay), some answers were: "good graphical support to understand instance creation", "useful", "very didactic!", "I did not understand it at the moment, but after some time I found it coherent". In relation to KLA5 (Roleplaying game) a student say: "It helped to understand concepts clearly and also to understand message passing and parameters".

Jeopardy (KLA10) was "interesting and very funny", "helps to know items you must know", "I did not know some of the answers, it was another learning instance", "promotes team work and reviewing concepts". In general terms, related to the usefulness of the tasks, they say "excellent pedagogy", "interesting and clarifying activities" and "a good way to understand things".

#### 5 CONCLUSIONS AND FUTURE WORK

According to the experimentation, a remarkable high proportion of exam exoneration results was obtained. We observed that students who participated in the experimental course obtained better results than students of standard courses. In addition, their comments in the survey showed high motivation and interest. It may be concluded that the

systematic use of KLAs allows students to obtain better results, but more experiments and deep analysis should be done to confirm –or not- those results. As future work, we propose to continue the searching of topics to generate new KLAs or to adapt some of those available.

## References

- [1] Khvilon, E. (coordinator). *Las tecnologías de la información y la comunicación en la formación docente. Guía de planificación*, UNESCO. Gráfica Futura, Uruguay , 2004.
- [2] Pendergast, M. Teaching Introductory Programming to IS Students: Java Problems and Pitfalls, *Journal of Information Technology Education*, vol 5, pp 491—515, 2006.
- [3] Robins, A., Rountree, J. and Rountree, N. Learning and teaching programming: a review and discussion, *Computer Science Education*, vol 13, No 2, pp 137--172, USA, 2003.
- [4] Gries, D. What have we not learned about teaching Programming. *Computer*, pp 81--82, October, 2006.
- [5] Gries, D. A Principled approach to teaching OO First. *Proc. of SIGCSE 2008*, pp 31--35, USA, 2008.
- [6] Janzen, D., Saiedian, H. Test-Driven Learning in Early Programming Courses. *Proc. of SIGCSE 2008*, USA, 2008.
- [7] Girard, D., Wellington, C. Work in Progress: A test-first Approach to teaching CS1. *Proc. of 36th ASEE/IEEE FIE*, M3E-19, USA, 2006.
- [8] Briggs, T., Girard, D. Tools and techniques for test-driven learning in CS1. *JCSC* 22, 3, 2007.
- [9] Rao, T., Mitra, S. An Early Software Engineering Approach to Teaching CS1, CS2 and AI. *Proc. of SIGCSE 2008*, USA, 2008.
- [10] Hanks, B. Student performance in CS1 with distributed pair programming. *Proc. of ITICSE 2006*, Portugal, 2006.
- [11] Powers, K., Gross, P., Cooper, S., Goldman, K., Carlisle, M., Mc Nallyu, MI, Prouls, V. Tools for teaching introductory programming: What works?. *Proc. of SIGCSE'06*, USA, 2006.
- [12] Dugas, C. No Computers? No Problem! Active and cooperative learning in an introductory computer science course. *Proc. of 38 th ASEE/IEEE FIE Conf.*, USA, 2008.
- [13] Sung, K., Panitz, M., Rosenberg, R., Anderson, R. Assessing Game-themed programming assignments for CS1/2 Courses. *Proc. of GDCSE'08*, USA, 2008.
- [14] Sivilotti, P, Pike, S. A collection of kinesthetic learning activities for a course on distributed computing. *ACM SIGACT news*, vol. 38, Issue 2, pp. 56--74. USA, 2007.
- [15] Begel, A., Garcia, D., Wolfman, S. Kinesthetic Learning in the Classroom. *Proc. of SIGCSE 2004*, pp 183-184 USA, 2004.
- [16] Astrachan, O. Concrete teaching: hooks and props as instructional technology. *Proc. of the 6th annual conference on the teaching of computing and the 3rd annual conference on Integrating technology into computer science education: Changing the delivery of computer science education*, Ireland, 1998.
- [17] Anewalt, K. Making CS0 Fun: an active learning approach using toys, games and Alice. *Consortium for Computing Sciences in Colleges Eastern Conference*, USA, 2007.
- [18] Felder, R, Silverman, L. Learning and Teaching Styles in Engineering Education. *Engr. Education*, vol. 78 No. 7, pp 674—681, 1988.
- [19] Goleman, D. *La práctica de la inteligencia emocional*. Ed Kairós, España, 1999.
- [20] Descartes, R. *Discurso del método*. 5th edition. Editorial Sopena Argentina, Argentina, 1964 (original edition 1637).
- [21] Polya, G. *How to solve it. A new aspect of mathematical method*. Princeton University Press, USA, 1988 (1st edition: 1945).
- [22] Gray, J.S. The Shuttle puzzle: a lesson in problem solving. *Journal of computing in higher education*, vol. 10, No. 1, pp 56--70 Fall 1998.
- [23] Bergin, J. <http://csis.pace.edu/~bergin/Java/RolePlay.html>
- [24] <http://www.juntadeandalucia.es/averroes/iesarroyo/matematicas/taller/aspectosweb/aspectosweb.htm>
- [25] de Bono, E. *El pensamiento lateral*. Ed. Paidós, Barcelona, 1998.
- [26] Computer Science Unplugged, <http://csunplugged.org/index.php/en/04-error-detection-activitiesmenu-110>
- [27] Bergin, J. Academic Jeopardy, *Proc. of the 10th annual SIGCSE conference on Innovation and technology in computer science education*, Portugal, 2005.
- [28] Kereki, I. *Programación I: Aprender programación orientada a objetos desde cero*. Empresa Gráfica Mosca, Uruguay, 2008.