Multiple Intelligence approach and Competencies applied to Computer Science 1

Alejandro Adorjan
Universidad ORT Uruguay
Montevideo, Uruguay
adorian@ort.edu.uy

Inés Friss de Kereki
Universidad ORT Uruguay
Montevideo, Uruguay
kereki_i@ort.edu.uy

***Abstract*— In order to contribute to the improvement of Computer Science 1 (CS1) course's results we designed a set of activities based on competencies and multiple intelligence approach. Hypothetically, we propose that including this kind of activities helps to obtain better results in the course. A preliminary experimentation was done in 2012. This year, the course included a low number of freshmen students. The initial results show a positive difference in student's results, particularly in freshmen students and no differences in students who are taking the course for the second or third time.**

Keywords— competencies; computer science 1; multiple intelligences.

I. Introduction

Introductory programming courses usually have high drop out rates and failure [1,2,3,4]. Our goal is to contribute to the improvement of Computer Science 1 (CS1) course's results by the incorporation of activities based on multiple intelligences and generic competencies. In this paper we include definitions of "competence" and possible classifications. We include also concepts related to multiple intelligences. After that, we detail our CS1 course, the activities proposed; the experimentation, results, conclusions and future work.

II. Competencies

Competencies represent a dynamic combination of knowledge, understanding, skills and abilities [5]. A competence is defined as the capacity of a person to use knowledge and skills in different situations either personal or professional [6]. Tovar [7] defines competence as the ability shown by the use of knowledge, technical, personal, social and methodological skills that can lead to success in professional and academic environments.

Competencies are distinguished in subject specific and generic ones. The generic competencies are also distinguished in three types: instrumental, interpersonal and systemic [5]. Transversal or generic competencies have acquired a special relevance in last years [7]. Generic competencies play a key role in the teaching and learning process [8]. Higher education must provide advance knowledge, skills and competencies that students need for their professional life [9]. The development of competencies in educational programs can significantly contribute to join reflection and work at university level [5].

Tovar and Soto found that reading, analytical comprehension and mathematical skills are part of the core competencies required to succeed in computer engineering studies, also suggests that universities must decide whether to take a comprehensive set of skills (such as the Tuning or extended version of it) [7,10].

Tuning Latin America Project [11] refers to 27 generic competencies. From this list, we select those that apply in the context of our CS1 course: capacity for abstraction, analysis, and synthesis (C1), ability to apply knowledge in practice (C2), ability to organize and plan time (C3), capacity for oral and written communication (C4), ability to use information and communication technology (C5), ability to learn and update learning (C6), ability to identify, pose, and solve problems (C7), ability to work as part of a team (C8), interpersonal skills (C9) and ethical commitment (C10). Most of these competencies are skills that software engineering graduates must possess [9].

III. Multiple intelligences

The theory of multiple intelligences (MI) was proposed by Gardner [12,13,14,15]. Gardner [16] argues that traditional ideas about intelligence must be reformed, so he proposes a new approach. Gardner defined intelligence as: "the ability to solve problems or to create products that are valued within one or more cultural settings". In this context, enumerates the intelligences: linguistic (MI1), logical mathematical (MI2), spatial (MI3), bodily-kinesthetic (MI4), musical (MI5), interpersonal (MI6), intrapersonal (MI7) and naturalist (MI8). Armstrong [17] refers two key points of MI: each person possesses all eight intelligences and most people can develop each intelligence to an adequate level of competency.

According to Gardner [16], the purpose of education should be to develop intelligence and to help people reach vocational and avocational goals that are appropriate to their particular spectrum of intelligence. In this context, Gardner [15] argues that MI theory is certainly relevant, to education, but not in itself an educational rationale or goal. Gardner 's notion of multiple intelligences shows that competencies are neither innate nor predetermined. People, with their intelligence are capable of preparing constructions based on the demands from their surroundings and they can develop specific capacities [18]. Research studies related to multiple intelligences in engineering education [19,20] suggest that students benefit from learning materials adapted to suit their intelligences.

IV. CS1 COURSE

Introductory programming courses have generally high failure and high drop out rates [1,2,3,4]. Research studies have been proposed to face the problem with different approaches: for example: games [21,22], robots [23], pair programming [24].

The course of CS1 at Universidad ORT Uruguay makes emphasis on teaching using problem-solving methodology and enables the student to analyze, design and implement simple object oriented language applications. It involves programming assignments, problem sets and a project. The duration is 15 weeks, 4 hours of lectures and 2 hours for lab session per week. The programming language used is Java for all this assignments. The main topics are: pseudo code, variables and control structures (weeks 1-3), objects and classes (week 4), association (weeks 5-7), inheritance (week 8), aggregation and collections (weeks 9-10), enumeration (week 11), sorting and searching (week 12) and advanced use of collections (weeks 13-15). Our teaching strategy is based on design activities that motivate and engage students in order to achieve lower dropout rates.

V. ACTIVITIES PROPOSED

In order to improve the results of the course, we present a set of activities designed using MI1-MI7 and C1-C10 approach:

1) Scratch (week 1): The use of Scratch in CS1 course promoted a high level of motivation, thus a positive perception of learning programming [25]. Students solve practical programming games that include sound and images.

2) Infographics (week 4): This is a graphic visual representation of information combined with words to help students explore and conceptualize Java. In this activity each student creates an infographic of Java technology (i.e. What is Java? , Who developed it?).

3) Kinesthetic (week 4): To become familiar with the concept of identifying objects, aliasing and message passing we propose a kinesthetic learning activity modeling clay [26] promoting comprehension of object oriented concepts.

4) Minute Test (week 4): Student reflects on what the main concepts of objects shown in the lecture are, answering in one minute several questions: for instance: “What do you want to find out about the theme?”, “Which concepts are important?”.

5) Wordle (week 5): Wordle [27] generates “word clouds” from a provided Java source. Students generate a cloud of words and analyze the most frequent reserved words of the language (public, private, int, …).

6) UML Modeling Game (week 6): We bring to class some child games. This activity aims to develop a simple design model of those games. Students use top down criteria identifying attributes and represent the model in UML.

7) Rubric (week 7): To improve students own learning and communication expectations, we propose and discuss a rubric and students must evaluate sample works using it.

8) Pair Programming videos (week 8): The activity is oriented to discuss the benefits of using pair programming methods analyzing some related videos.

9) Inheritance (week 8): Group of students read in class material related to a new topic (inheritance). They read chapters of several books of the course and complete a questionnaire.

10) Concept Test (week 10): Peer Instruction (PI) [28] engages students during class through activities that require each student to apply the core concepts being presented, and then to explain those concepts to their fellow students. A class taught with PI is divided into a series of short presentations, each focused on a central point and followed by a related conceptual question. We apply this activity to introduce foundational topics of arrays.

Each activity refers to some competences and intelligences (Table I and Table II).

TABLE I
ACTIVITIES - COMPETENCIES

C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
1,2,3 4,5,6 7,8,9 10	2,5,6 7,8 10	2,5	2,4,5 7,10	2,9,5	4,9 10	1,4	6,8,7 10	5,6,7 8,9	7,8

TABLE II
ACTIVITIES – MULTIPLE INTELLIGENCE APPROACH

MI1	MI2	MI3	MI4	MI5	MI6	MI7
2,5,9	1,10	1,2,5,8	3,6,8,9 10	1	1,5,6,7,8,9 10	1,2,3,4,5 10

VI. EXPERIMENTATION

One random group of 20 students was selected to participate in the activities and 16 were in the control group. None of the students had already been exposed to MI techniques nor the activities proposed. Both groups included a high number of students who are not really “freshmen”: they were taking the course for the second or third time. The repeat students had the same course contents in their previous attempts. Hypothetically, we propose that including this kind of activities helps to obtain better results in the course. An initial experimentation was done in 2012. Considering only the students who take the course for first time, in the selected group, 44% (4/9) of them had successful results and in the control group, 22% (2/9) gained the course. For students who took the course a second or third time, there were no significant differences in the results.

VII. CONCLUSION AND FUTURE WORK

The inclusion of activities to promote skills and multiple intelligences seem to provide better results in the case of students taking the course for the first time. The use of these activities in the case of students who were enrolled and did not pass the course previously showed no significant difference in the results. These differences appear in the case of new students. In 2013 we will replicate the experience, with some improvements, in order to validate or not the results.

REFERENCES

[1] Ambrosio A.P., Costa F.M., Almeida L., Franco A., Macedo J. "Identifying cognitive abilities to improve CS1 outcome", *Proc. of 41th ASEE/IEE Frontiers in Education Conference,* 2011, pp. F3G-1 F3G-7.

[2] Robins A., Rountree J. and Rountree N. "Learning and teaching programming: a review and discussion", *Computer Science Education*, Vol 13, No 2, 2003, pp. 137-172.

[3] Kinnunen P. and Malmi L. "Why Students Drop Out CS1 Course?", *ICER' 06*, United Kingdom, September 2006, pp. 97-108.

[4] Bennedsen J. and Caspersen M. "Failure Rates in Introductory Programming", *Inroads The SIGCSE Bulletin*, Vol 39, No 2, June 2007, pp 32-36.

[5] González J. and Wagenaar R. *"Universities contribution to the Bologna Process. An Introduction (2nd edition)*". Publicaciones de la Universidad de Deusto, 2008 [On-line].Available: http://www.unideusto.org/tuningeu/images/stories/Publications/ENGLISH_BROCHURE_FOR_WEBSITE.pdf , last accessed Nov 5,2012.

[6] Hoffmann M.H.W., Bargstadt H.-J., Hampe M., Heiss H.U., Muller G. and Schmitt H. "Knowledge, skills, and competencies: Descriptors for engineering education," *Education Engineering (EDUCON), 2010 IEEE*, April 2010, pp.639-645.

[7] Tovar E. and Soto O. "The use of competencies assessment to predict the performance of first year students," *Frontiers in Education Conference (FIE)*, 2010, pp.F3J-1-F3J-4.

[8] Edwards M., Caro E.T. and Sainz, O.S. "Embedding a core competence curriculum in computing engineering," *38th Annual Frontiers in Education Conference, 2008. FIE 2008.*, 2008, pp.S2E-15-S2E-20.

[9] "Software Engineering 2004" Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering, *The joint Task Force on computing Curricula, IEEE Computer Society*, Association for Computing Machinery, August 2004.

[10] Tovar E. and Soto O. "Are new coming computer engineering students well prepared to begin future studies programs based on competencies in the European Higher Education Area?," *Frontiers in Education Conference,* 2009, pp.M2D-1-M2D-6

[11] Tuning America Latina, http://tuning.unideusto.org/tuningal/index.php?option=com_docman&Itemid=191&task=view_category&catid=22&order=dmdate_published&ascdesc=DESC , last accessed Nov 5,2012.

[12] Gardner, H. *Frames of Mind: the theory of multiple intelligences*, USA: Basic Books, 2011.

[13] Gardner H. and Hatch T. , "Educational implications of the theory of multiple intelligences", *Educational researcher* , Vol. 18, No. 8, 1989, pp. 4-10.

[14] Gardner,H *Intelligence reframed: Multiple intelligences for the 21st century*, USA: Basic Books, 2000.

[15] Gardner,H. *Multiple Intelligences New Horizons in Theory and Practice*, USA: Basic Books, 2006.

[16] Gardner, H. *Multiple Intelligence the theory in practice*, USA: Basic Books, 1993.

[17] Armstrong, T. *Multiple Intelligences in the Classroom*, USA: ASCD (3rd edition), 2009.

[18] Kun Liu, Yuhua Ni and Hui He; , "Study of the theory of multiple intelligences on fundamental computer education" *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference,* 2011 , pp.267-270.

[19] Matrisciano A. and Belfiore N.P., "An investigation on Cognitive Styles and multiple intelligences model based learning preferences in a group of students in engineering" *Information Technology Based Higher Education and Training (ITHET),* 2010, pp.60-66.

[20] Ali Bushro and Zaman Halimah Badioze "Multimedia mathematics courseware based on the multiple intelligences model (MI-MathS)", 2008. *ITSim 2008 International Symposium on Information Technology*, 2008, vol.2, pp.1-5

[21] Cliburn D.C., Miller S.M. and Bowring E.; , "Student preferences between open-ended and structured game assignments in CS1", *Frontiers in Education Conference (FIE),* 2010, pp.F2H-1-F2H-5.

[22] Smallwood G.R. and Black D.V. "Observations on designing a computer science Curriculum focusing on game programming using testimonials from industry leaders" *Games Innovation Conference (IGIC), 2011 IEEE International*, 2011, pp.126-129.

[23] De Giusti A., Frati F. E., Sanchez M., De Giusti L. "LIDI Multi Robot Environment: Support software for concurrency learning in CS1," *Collaboration Technologies and Systems (CTS), 2012 International Conference* , pp.294-298, 21-25 May 2012.

[24] Radermacher A. and Walia G. "Investigating student-instructor interactions when using pair programming: An empirical study" *2011 24th IEEE-CS Conference Software Engineering Education and Training (CSEE&T),* 2011, pp.41-50.

[25] de Kereki, I.F. "Scratch: Applications in Computer Science 1," *FIE 2008. 38th Annual Frontiers in Education Conference, 2008.*, pp.T3B-7-T3B-11.

[26] de Kereki, I.F. "Incorporation of 'Kinesthetic Learning Activities' to Computer Science 1 course: Use and Results", *CLEI Electronic Journal*, Vol 13, No. 2, August 2010.

[27] Wordle, http://www.wordle.net, last accessed Nov 5,2012.

[28] Crouch C.H. and Mazur E. "Peer Instruction: Ten years of experience and results," *American Journal of Physics*, vol. 69, no. 9, p. 970, 2001.