Diversifying activities to improve student performance in programming courses

Scheila Wesley Martins, Antonio Jose Mendes and Antonio Dias Figueiredo

Abstract: *This paper presents a pedagogical strategy that intends to help students maximize learning and minimize drop-out rates in programming courses. The goal is to motivate students to develop a better programming study behaviour through the utilization of appropriate learning activities and the conscious assessment of their self-efficacy level. The paper also includes some preliminary results of the strategy application with students of Design and Multimedia.*

Key words: *Programming learning; Research Communities; Self-efficacy.*

INTRODUCTION

There is an intense effort from researchers and teachers worldwide to understand the reasons that make programming learning so difficult for many students [1]. In fact, it is common to find students that experience many difficulties to develop problem solving competences, and to use those competences to create programs that solve basic problems. It is important to make students realise that programming is, above all, a conscious exercise of mental abilities that can be developed through adequate activities and effort. Hence, any pedagogical strategy directed to programming learning should make students aware that solving programming problems is an activity that they are fully capable of accomplishing. It is important to value contexts and establish class dynamics that may motivate students to teamwork, giving evidence and convince them that individual difficulties can be solved if they get ready to “learn to think”. This should lead to a higher student commitment to their learning, including behavioural changes that may improve their performance throughout the course.

In the next sections we propose a strategy designed with the above objectives. We also describe the results obtained in a Programming course at the University of Coimbra.

TEACHING TO THINK IN PROGRAMMING

Our goal is to identify the characteristics of contexts that may make programming learning more stimulating, minimize drop-out intentions and make students learn more and better. The pedagogical strategy we propose includes a set of guidelines regarding contexts and didactic activities, computational tools and motivational measures that may assist teachers in the definition of specific learning contexts for programming courses. The proposal was developed under the perspective of learning communities, inspired by a metaphor of Mathew Lipman's communities of inquiry [2], considered to be a relevant abstraction for proposals involving the development of critical thinking and literary skills, and also as a strategy to improve the capacity to solve programming problems among university students [3].

The course context should include didactic activities planned to strengthen the student's involvement with the process of knowledge acquisition and development of competences to solve problems, through teamwork and the motivation to practice their literary skills in several ways, such as collaborative knowledge production through small projects and research activities, peer tutoring and continuous assessment. The context can include computer tools that might help learning, such as algorithm simulation or software to support competitions and testing of programs. To stimulate extra-curricular activities and to facilitate monitoring and continuous assessment tasks, it is important to use a Learning Management System, such as Moodle. Motivational measures such as comfort level, self-efficacy [4], and confidence, usefulness and satisfaction with the proposed activities should be checked regularly to support student guidance and to adequately direct the teacher's efforts in student motivation and the prevention of behaviours that may lead to students dropping out.

AN EXPERIMENT WITH DESIGN AND MULTIMEDIA STUDENTS

Our proposal was experimented with students enrolled in a Programming course, part of the Masters Degree in Design and Multimedia (MDM) at the University of Coimbra, in the academic years 2008/09 and 2009/10. However, the results included in this paper are only relative to the second year. We chose this course because it involves a much smaller number of students than other courses, and because its students usually don't show a high appetence for programming. The small number of students permitted a close student monitoring. This allowed the teacher to know well the students, and to adapt the class dynamics to a research approach during group based problem solving.

Our strategy doesn't make a clear distinction between theoretical, practical or lab classes, as occurs in many programming courses. All classes are spaces for knowledge construction and practical experimentation, making up a total of 6 weekly hours of work. Bearing in mind the artistic background of the involved students, we chose to create a context based on visual hands-on projects of growing complexity, as it would facilitate the students' involvement and interest in the activities. We used the programming language Processing [5] as it facilitates the development of artistic works, keeping the power of Java language. We also used Moodle as a basis for some activities. The course was mostly based on practical learning. The exercises and projects proposed involved a need for research, especially the review of algebra and trigonometry knowledge. We used several types of activities during the semester with specific objectives. Individual seminars on artistic projects developed in Processing were used to raise students' interest and motivation about programming. In specific moments we used individual challenges, inspired in JiTT challenges [6], as a way to stimulate individual work, especially outside the classroom. These challenges included a self-evaluation component, making the student used to critical assessment. We also proposed several small projects to be developed in groups, followed by discursive evaluation of peers' work. We included two small individual tests, which were preceded by a test simulation to allow students to have a more concrete feeling about their level, without being under real assessment. Finally students were asked to create a portfolio including their own programming projects and other related materials. All these activities were evaluated by the students in biweekly reflections about their satisfaction with their own performance, tasks, materials and class's rhythm.

During the course we used some instruments to evaluate several aspects: the Inventory of Attitudes and Study Behaviours (IABS/IACHE) [7] to get information about study behaviour, the Course Interest Survey (CIS) [8] to measure students' motivation according to ARCS model, the Student Motivation Problem Solving Questionnaire (SMPSQ) [9] to assess the level of satisfaction with the different learning activities and a self-efficacy test [10] to keep students alert regarding the quality of their learning.

This experiment took place between September 2009 and February 2010. The course had 18 registered students, although only 15 really got involved in it. Most students were recent graduates in the areas of arts and design, but two were Polish Erasmus students from Physics Department.

RESULTS

In each survey the questions were answered by students according to the intensity of their level of accordance: from 1 (means no, totally false, or totally unconfident) until 5, 6 or

7 depending on each survey (means yes, totally true or totally confident), the answer 0 (means don't know) is possible only in SMPSQ survey. The score for each aspect assessed is given by the sum of the answers to the corresponding question. The surveys' structure and their reference values (minimum, maximum and average point) are summarized in table I. There was another analysis carried out, called intensity levels, also show in table I. It was proposed to identify change on students' answers patterns, by assembling the groups of answers in three levels: low, medium and high.

TABLE I
TEST DATA SPECIFICATIONS SUMMARY

	Tests Composed		Statistic Measures			Intensity Levels of Answers		
	Questions	Answers	Min	Max	X_m	Low	Medium	High
IACHE	44	1 - 6	10 [a] 8 [b]	60 [a] 48 [b]	35 [a] 28 [b]	1 and 2 [c]	3 and 4	5 and 6 [c]
CIS	34	1 - 5	9 [d] 8 [d]	45 [d] 40 [d]	27 [d] 24 [d]	1 and 2	3	4 and 5
SESP	32	1 - 7	32	224	128	1 and 2	3, 4 and 5	6 and 7
SMPQS	15 5	0 [h], 1 - 5 0 [h], 1 - 5	15 [f] 5 [g]	75 [f] 25 [g]	45 [f] 15 [g]	1 and 2	3 and 4	5 and 6

(a) For comprehensive focus and organization dimensions; **(b)** For reproduce focus, involvement and competence personal perception dimensions; **(c)** For competence personal perception dimension, the low level is 5 and 6, and the high level is 1 and 2; **(d)** For relevance and satisfaction aspects; **(e)** For attention and confidence aspects; **(f)** For part I - motivation about academic tasks; **(g)** For part II - general motivation; **(h)** When it appears means that more research has to be done about the question (Don't Know level);

Given the qualitative nature of data, we used the Wilcoxon Signed-Rank test, a non-parametric statistic test, to analyse it. We also carried out a statistical comparison between the average results, and the analysis of the percentage of replies arranged in scales of intensity.

The IACHE survey includes cognitive, motivational and behavioural dimensions, distributed in five sub-scales: (I) Comprehensive focus, using reflection and deep content analysis, which implies an higher effort and time in learning; (II) Reproductive focus, the tendency to spend only a minimum effort on a superficial learning, based on memorization and content reproduction; (III) Competence personal perception, a measure on how students see their own competence in the course; (IV) Involvement, or motivation, related essentially with intrinsic motivation; and (V) Organization, analyzes the indications of some level of organization on study activities. Table II presents the results in each dimension of IACHE, in columns I to V, and in the self-efficacy test, in column VI. In both cases, pre-tests were made in the beginning of the course and pos-tests in the final part. The table includes average answers and the percent of students who gave answers in each of the three intensity levels.

TABLE II
IACHE AND SELF-EFFICACY MEANS AND INTENSITY LEVELS SUMMARY

	PRE						POS					
	I	II	III	IV	V	VI	I	II	III	IV	V	VI
Average	42.6	29.9	21.5	36.6	31.4	114.0	39.9	28.1	25.0	33.1	31.4	127.6
Low(%)	4	20	9	3	23	24	5	23	10	4	31	11
Mean(%)	52	47	34	42	59	62	69	52	61	64	55	75
High(%)	44	33	57	55	18	14	26	25	29	32	14	14

Comprehensive Focus-I, Reproduce Focus-II, Personal Perception-III, Involvement-IV, Organization-V, and Processing's Self-Efficacy-VI

When comparing IACHE averages in pre and post-test, we can see they have decreased, except for Personal Perception (III) that increased, and Organization (V), that had no change. However, the averages of the comprehensive focus in both tests were higher than the average of the reproductive focus. This is a good sign that may have resulted from the stability verified in the organizational dimension. We believe that the inclusion of the challenges in the pedagogical strategy demanded an organizational effort from the students, especially outside classroom, so that they could meet the different deadlines. These results can be considered positive, even though there was a small decrease in the involvement and comprehensive focus. The same can be said about the analysis by level of intensity, as we could see some migration of answers from low level to medium level, although some also migrated from high level to medium level.

The results of the Wilcoxon Signed-Rank test for IACHE test are displayed in table III. They indicate that the decreases in comprehensive and reproductive focus averages cannot be considered statistically significant. Even the result of the involvement focus (ρ=0.061) is very close to the statistical limit defined for this type of test (ρ=0.05), which doesn't give a strong support to prove a statistic relevant difference. On the contrary, the variations on personal perception and organization are statistically relevant (ρ=0.021), which suggests some modification of the students' behaviour during the course.

TABLE III
WILCOXON RANGE TEST SUMMARY

	I	II	III	IV	V	VI
Negative Ranks (Pos < Pre)	7	10	2	9	8	3
Positive Ranks (Pos > Pre)	2	2	8	2	2	9
Ties (Pos = Pre)	3	0	2	1	2	0
ρ-Value	.085	.134	**.021**	**.061**	**.021**	**.021**

Comprehensive Focus-I, Reproduce Focus-II, Personal Perception-III, Involvement-IV, Organization-V and Processing's Self-Efficacy-VI

We had expected to find an increase in the averages of involvement and comprehensive focus. However, we found a slight decrease, which means that students didn't develop as much as we expected in those dimensions. As positive points we noticed the stability in organization and the decrease in the reproductive dimension, which means that students understood that this dimensions isn't particularly relevant for programming learning. The most negative aspect was the increase in personal perception, since it means that the students' level of trust in their own skills to be successful in programming learning decreased. This is quite worrying as students with low expectations tend to invest less effort in study activities and drop out more easily.

The results obtained with the self-efficacy scale, column VI on table II, were more positive, as they revealed a positive evolution in students' confidence on Processing. The Wilcoxon test result, column IV in table III, (ρ=0.021) also proved that there was a relevant difference between pre-test and pos-test results. The analysis of intensity levels revealed that the number of low level answers has decreased, migrating to the medium level, while the concentration of high level answers remained unchanged. We also see that in the pre-test a little over 25% of the sample already presented individual scores higher than the medium point reference value. In the post-test only 25% of the sample decreased the value of its score, which means that 75% of students kept or increased their self-efficacy level for programming using Processing.

We used the CIS survey to measure the motivation levels, according to the ARCS model, aiming to discover how much relevance, motivation, confidence and satisfaction the students presented in the middle of the course. This could give important information to the teacher, detecting situations that might require his intervention, either with a particular student or with the whole course. The results can be seen in the left side of table IV. They show a higher concentration of answers in the high level, which is good. However, confidence and satisfaction dimensions show a higher value in low level answers, which shows that some students' level of trust wasn't high at that time. This was confirmed later by the rise in personal perception in IACHE. The results for attention and relevance dimensions showed that students were fully aware of the importance of the course and were consciously committed to work towards learning the necessary programming skills.

TABLE IV
CIS AND SMPSQ SURVEYS' AND INTENSITY LEVELS SUMMARY

CIS Dimension	Mean	Low (%)	Mean (%)	High (%)	Activities assess by SMPSQ	Mean Part 1	Mean Part 2	DK (%)	Low (%)	Mean (%)	High (%)
Attention	27.75	8	43	49	Seminar	50.08	15.33	1	5	32	62
Relevance	33.00	7	32	61	Code Analysis	49.80	17.50	3	10	41	46
Confidence	28.50	20	28	52	Mini-test Simulation	51.07	16.69	2	8	39	51
Satisfaction	29.92	21	31	48	Programming Challenges	40.30	13.46	1	6	43	50

The SMPSQ test was used to identify the level of satisfaction and resistance felt by the students, specifically concerning the different activities proposed. The test is divided in two parts, the first assess the motivation to perform a specific activity or task, and the second evaluates the reward expectations and the success to achieve the student's goals. The higher values obtained in the first part of the test, the less resistance or more motivated the student is. The same happens in part two, as higher values mean that the students have better personal perception levels for success. The statements answered with a 0 (Don't Know level) should be observed, as they may reveal causes for the students' resistance regarding a given activity and also possibly show their insecurity about their goals and success possibilities. The summary of SMPSQ's results is also presented in table IV. A statistical analysis doesn't show relevant differences between the various activities. However, intensity analysis shows that seminar and programming challenges activities had slightly better results. It was good to see the lower concentration in Low and DK levels. However, we were a bit puzzled to see that 3% of students' chose 0 for code analysis activities.

We also conducted a content analysis over the biweekly individual reflections that students wrote during the course. The positive and negative aspects mentioned by more students were identified. On the positive side, 93.33% of the students mentioned the motivational impact of class dynamics; 86.67% of the students referred the high level of collaboration between students, the possibility to use their creativity in programming assignments, and teacher availability to clear doubts; 80% of the students wrote about good class rhythm, individual support provided by the teacher, the learning activities, their performance in challenges, and the development of their own study behaviour; 66.67% of the students mentioned the sensation that their study effort was rewarded. On the negative side, 66.67% of the students recognized their lack of mathematical knowledge complicated learning; 60% of the students mentioned previous bad experiences in programming courses, negative expectations about their own performance, frustration about not being able to solve some problems, and insecurity about their grades; 40% of the students complained about the amount of work and the course level of demand. From this analysis we can conclude that most negative aspects were related with students' past experiences and their fears of underperforming in the course, and not with the course itself.

The approval rates in a course are usually a good measure to assess the results of a pedagogical strategy, though this measure is sometimes overrated. When we think about motivation, the drop-out levels may be more important than final grades. Anyway, the evaluation of a particular strategy should include the students' results. In our case 80% of the students managed to pass the course, although most of them with average grades. Considering students' backgrounds and the difficulty associated with programming courses, we think the results obtained were good.

The teacher makes a positive evaluation of the strategy, not only due to the results obtained, but also considering the class dynamics. However, he also acknowledges a significant increase in his work, when comparing with more conventional approaches.

CONCLUSIONS AND FUTURE WORK

The general evaluation of our experiment was considered positive by students and the course teacher. It may be considered an improvement over traditional approaches for the same course context. However, we recognize that some aspects weren't as positive as we expected. This means further reflection is necessary and possibly some improvements have to be considered in the next edition of the course.

Currently we are conducting a new experiment, this time with graduation students enrolled in Design and Multimedia Degree, more specifically in the course on Programming and Data Structures. Although in the previous year Java was used in this course, the good feedback from MDM students and teacher about Processing and the whole experience, motivated this course teacher to adopt some components of the strategy tested. This experiment takes place in the second semester of 2009/10. The authors will follow and evaluate this new experiment, in order to assess its results and get information that may lead to the improvement of the strategy. This experience is particularly interesting because the number of students involved is much higher. This will allow us to get information about the strategy feasibility with this type of courses.

REFERENCES

[1] Lahtinen, E., Ala-Mutka, K., and Järvinen, H.,"A study of the difficulties of novice programmers", SIGCSE Bull., Vol. 37, No. 3, Sep. 2005, pp. 14-18.

[2] Lipman, M., "Thinking in Education", Cambridge University Press, 1st ed., 1991.

[3] Martins, S. W., "Um modelo computacional de apoio ao desenvolvimento do pensamento crítico", MSc. thesis, Universidade Federal de Santa Catarina (UFSC), Brazil, May 2005.

[4] Bandura, A., "Self-efficacy: Toward a unifying theory of behavioral change", Psychological Review, Vol. 84, No. 2, Mar. 1977, pp. 191-215.

[5] Greenberg, I., "Processing: Creative Coding and Computational Art", Friends of Ed Press, 1st ed., 2007.

[6] Bailey, T. and Forbes, J., "Just-in-time teaching for CS0", SIGCSE Bull., Vol 37, No. 1, Feb. 2005, pp. 366-370. DOI= http://doi.acm.org/10.1145/1047124.1047469

[7] Monteiro, S., Vasconcelos, R. M., and Almeida, L. S., "Rendimento Acadêmico: Influência dos métodos de estudos", Proceedings of VII Congresso Português de PsicoPedagogia, University of Minho, 2005.

[8] Keller, J. M., "Motivational Design for Learning and Performance: The ARCS Model Approach (handbook)", NY: Springer, 2009.

[9] Margolis, H. (January,2009). "Student Motivation: A Problem Solving Focus" [Online], Available http://www.reading2008.com/MotivationProblem_Solving_Question-naire-HowardMargolis-2009Jan1-c.pdf

[10] Askar, P. and DaVenport, D.,"An Investigation of Factors Related to Self-Efficacy for JAVA Programming Among Engineering Students", The Turkish Online Journal of Educational Technology – TOJET, Vol. 8, Issue 1, Article 3, Jan. 2009.

ABOUT THE AUTHORS

PhD Student, Scheila W. Martins, Centre for Informatics and Systems (CISUC), University of Coimbra, Portugal, Phone: +351 239790078, E-mail: scheila@dei.uc.pt.

Professor, Antonio José Mendes, PhD, Centre for Informatics and Systems (CISUC), University of Coimbra, Portugal; Phone: +351 239790036, E-mail: toze@dei.uc.pt

Professor, António Dias Figueiredo, PhD, Centre for Informatics and Systems (CISUC), University of Coimbra, Portugal; Phone: +351 239790021, E-mail:adf@dei.uc.pt