Using a Real Bare Machine in a Project-Based Learning Environment for Teaching Computer Structure: An Analysis of the Implementation Following the Action Research Model

EDURNE LARRAZA-MENDILUZE, NESTOR GARAY-VITORIA, IRATXE SORALUZE, JOSÉ MARTÍN, JAVIER MUGUERZA, and TXELO RUIZ-VÁZQUEZ,
University of the Basque Country

The computer input/output (I/O) subsystem and its functioning are very abstract concepts that are difficult for undergraduate freshmen to understand. However, it is important that freshmen assimilate these low-level concepts if they are going to be taught about the operating systems (OS) working over that architecture layer, or working directly with them in embedded systems, real-time systems, or in the area of human–computer interaction (HCI). This article describes the use of a game console (Nintendo® DS, NDS) in a project-based learning (PBL or PjBL) environment in which the design of a game is the basis of the project in order to encourage the students to get more involved with the computer I/O subsystem abstraction. A 4yr experience is reported in which the action research model (planning, acting, observing, and reflecting) has been followed. The general procedure for the 4yr and the specific characteristics and achieved results for each year are reported. The aim of the study was twofold: to assess the learning effectiveness of the active PjBL educational approach and some related factors, and to analyze the motivation toward the subject fostered by the game console. The first aim is analyzed using the scores achieved by the students; the second aim is analyzed via satisfaction questionnaires.

13

CCS Concepts: • **Social and professional topics → Computing education**; • **Applied computing → Collaborative learning**

Additional Key Words and Phrases: Cooperative/collaborative learning, improving classroom teaching, interactive learning environments, pedagogical issues, teaching/learning strategies

ACM Reference Format:
Edurne Larraza-Mendiluze, Nestor Garay-Vitoria, Iratxe Soraluze, José Martín, Javier Muguerza, and Txelo Ruiz-Vázquez. 2016. Using a real bare machine in a project-based learning environment for teaching computer structure: An analysis of the implementation following the action research model. ACM Trans. Comput. Educ. 16, 3, Article 13 (May 2016), 17 pages.
DOI: http://dx.doi.org/10.1145/2891415

The authors belong to the Basque Advanced Informatics Laboratory (BAILab), grant UFI11/45, supported by the University of the Basque Country (UPV/EHU). This research work was supported by COD. 6254 “Konputagailuen Sarrera/Irteera aspisistema ikastea NDSrako proiektu bat garatuz eta sistema berri honen aplikazioaren hausnarketa” project funded by the Educational Council SAE/HELAZ at the UPV/EHU, by the Ministry of Economy and Competitiveness of the Spanish Government and by the European Regional Development Fund (projects TIN2013-41123-P and TIN2014-52665-C2-1-R), and by the Department of Education, Universities and Research of the Basque Government under grant IT395-10.
Authors’ addresses: University of the Basque Country UPV/EHU, Faculty of Informatics, Department of Computer Architecture and Technology, Manuel Lardizabal Pasealekua 1, 20018 Donostia, Gipuzkoa, Spain; emails: {edurne.larraza, nestor.garay, iratxe.soraluze, j.martin, j.muguerza, txelo.ruiz}@ehu.eus.

DOI: http://dx.doi.org/10.1145/2891415

1. INTRODUCTION

Standard computing curricula [ACM/IEEE Computer Society 2008; IEEE Computer Society/ACM 2004] emphasize the relevance of the input/output (I/O) topic, for which interfacing and I/O strategies are defined as core matters. The curriculum guidelines for undergraduate degree programs in computer engineering [IEEE Computer Society/ACM 2004] define, among others, the following learning outcomes: "*Explain how to use interrupts to implement I/O control and data transfers*" and "*Write small interrupt service routines and I/O drivers using assembly language.*" Previous research work on how the computer I/O subsystem is taught [Larraza-Mendiluze and Garay-Vitoria 2015] classifies four different approaches: (1) the purely descriptive approach, (2) the performance approach, (3) the datapath-signal approach, and (4) the programming approach. The fourth approach is the one that is better suited to improving learning outcomes and the one for which practical laboratory work and a real or emulated computing machine is needed.

Our faculty has been teaching the I/O subsystem topic following the programming approach for some time. However, we have primarily used the same teaching methods for years, which include lectures to explain theoretical concepts and laboratory assignments to put them into practice, based on personal computers. This happened when the I/O topic was taught in the first semester of the second year in a subject called Computer Architecture I (CAI).

Due to the adoption of the European Higher Education Area (EHEA), the I/O topic moved from the CAI subject to a new subject called Computer Structure (CS) in the second semester of the first year. The aim of this new subject is more or less the same as the aim of CAI, to show students the structure of a computer and how the main parts work. However, many things changed due to the rearrangement of the topics. In Section 3 of this article, main changes between CAI and CS subjects are detailed.

The I/O topic concerned the lecturers of this subject even when it was in the second year because the students had difficulty understanding and familiarizing themselves with the content. Pass rates were low (44%) and drop-out rates high for a second year (31%). Naturally, when the topic moved to the first year, the concern increased because the difficulty in understanding a topic is understandably greater in the first year, when the students are more immature. Moreover, during the first year, the students are required to pass at least two subjects to advance in the informatics degree. Around 45% of the students do not fulfill that requirement and have to leave.

Data shown in this article refers to just a topic in the CS subject. However, in order to be able to contextualize these numbers, note that, in our institution, the passing rate in the second semester of first-year subjects is around 34% and the drop-out rate around 42%, both on average, while the numbers corresponding to CS (the full subject) are around a 42% passing rate and 40% drop-out rate.

Nevertheless, it is important that freshmen understand these low-level concepts if later they are to be taught about the operating systems (OS) working over that architecture layer or working directly with it in embedded systems, real-time systems, or in the area of human–computer interaction (HCI), mostly by using interrupts and concurrent processes. Therefore, we consider interrupts and service routines to be very important and should figure in the students' knowledge from the first year.

This work is based on the hypothesis that a real bare machine, such as a Nintendo® DS (NDS) game console, together with the project-based learning (PjBL) educational approach, can form an ideal team in the educational process of the computer I/O subsystem. We use the term "bare machine" to denote a machine that can be worked on without it having an OS, in order to directly access its resources and deal with its architecture. We also study some adjustable parameters in the application of a PjBL educational approach that can be generalized for other subjects.

This article is organized as follows. Section 2 discusses related work that has influenced this study in some way. Section 3 gives the required background information about course and curricular context. Section 4 describes how the NDS features have been used. In Section 5, the data gathered and the results obtained each year are presented. In Section 6, we will describe each implementation, reflecting on the results and the adaptations made to several PjBL parameters during the four subsequent academic years in which it was applied, from 2010/2011 to 2013/2014. In Section 7, we discuss the lessons learned during this experience and present our conclusions.

2. RELATED WORK

The aim of this article is to describe our experience teaching the I/O topic in an introductory Computer Structure course while trying to introduce the PjBL educational approach. For laboratory sessions, Personal Computers (PCs) used to be an adequate option, but currently they are not very attractive, due to the complexity of the OS intervention in the management of low-level hardware, which poses some difficulties to students' hands-on learning. In order to avoid that problem, virtual machines and simulators are usually used by Black and Komala [2011], Black and Waggoner [2013], Donaldson et al. [2011], and Scott [2000]. Although the authors of those articles claim the benefits of these techniques, others [Santofimia and Moya 2009; Brorsson 2002] stress the importance of working directly with real hardware to enable students to learn about the hidden details in simulators. For this reason, some authors prefer platforms specifically designed for course development [Brorsson 2002; Ellard et al. 2002; Brylow and Ramamurthy 2009]. However, these approaches are not usually very appealing to students, because they are clearly machines not found outside the laboratory.

In order to improve student motivation, some authors prefer real machines known to students outside their learning environment, such as robots [Teller et al. 2003] or game consoles. Some attempts have been made in introductory computer-architecture courses. Tew et al. [2008] use a Game Boy machine, but they focus on machine language and assembly language rather than the I/O subsystem. Santofimia and Moya [2009] use the NDS, but all work with I/O is restricted to memory control. The last two studies mention that the students' motivation increased. Santofimia and Moya [2009] say that adopting the NDS is not a silver bullet for supporting students in their education. However, they claim that the results show an important improvement in terms of the content understood and retained by the students.

As a way of improving students' learning from a methodological point of view, confidence in PjBL as an educational approach is increasing at the expense of more traditional methodologies, in which lectures and traditional practical exercises prevail and students passively collect the information that the lecturer transmits. In contrast, in PjBL, the students themselves take responsibility for their learning process and become promoters of their own learning. They have to develop a project that is relevant to the real world. They need to obtain the relevant information and learn all of the necessary concepts while the lecturer provides guidance and counseling in the process (i.e., they learn the concepts by doing). Thus, the project stimulates learning. Moreover, PjBL is an educational approach based on collaboration. Students have to work in teams, which forces them to develop other skills [Thomas 2000]. There are many literature references on PjBL in engineering education [Díaz Lantada et al. 2013; Nepal 2013] that stress its importance and effectiveness. With regard to computer science and computer engineering, Ramachandran and Leahy [2007] provide a good list of possible projects to be carried out in the area of computer architecture. Berglund and Eckerdal [2006], Martínez-Monés et al. [2005], Stanley et al. [2007], and Urness [2007] also describe interesting implementations of the PjBL educational approach in the area of computer architecture and systems.

With all this in mind, we modified our educational approach by transitioning from classical lecture-based passive classes to PjBL active methods. We also decided to use the NDS to do practical laboratory work. This machine has several I/O devices, explained in Larraza-Mendiluze et al. [2013] and perfectly suited the objective of carrying out attractive projects. The objective of our research was to seek a way of implementing the educational approach, which would encourage students to engage with the topic and, therefore, obtain a good perspective and knowledge of it.

We knew that it would not be possible to achieve the best scenario with the first implementation. Thus, we applied the action research model, "a diagrammatical model as a spiral of cycles, each consisting of four moments or phases in action research: (1) planning; (2) acting; (3) observing; and (4) reflecting" [Altrichter et al. 2002]. What we learn during the year will be used to make the educational approach better in subsequent years.

3. THE EDUCATIONAL CONTEXT

For many years, in the Faculty of Informatics of the University of the Basque Country (UPV/EHU), the computer I/O subsystem has been taught in an introductory second-year, first-semester CAI subject. The aim of this 7.5 European Credit Transfer and Accumulation System (ECTS) subject was to introduce the functioning of the main parts of the computer. The topics were:

—*The control unit* (10% of the subject), in which microprogramming was introduced.
—*The processing unit* (20% of the subject), in which the students had the opportunity to learn how arithmetic operations are performed in a computer.
—*The main memory* (20% of the subject), in which the organization of the main memory was explained.
—*The I/O subsystem* (40% of the subject), in which the mechanisms to connect the outside world with the computer were introduced.
—*The connection subsystem or the buses* (10% of the subject), in which the main bus protocols were explained.

Due to the adoption of the EHEA, the I/O topic moved to a first-year, second-semester new subject CS. The aim of this 6 ECTS subject is more or less the same as the aim of CAI, to show students the structure of a computer and how the main parts work. However, many things changed due to the rearrangement of the topics.

The subject includes the four main topics of the "Von Neumann" structure:

—*Introduction and Memory* (5% of the subject). In this topic, an overview of the Von Neumann structure is presented, as it has already been covered in a previous subject. Then, the concept of main memory and how addressing works is discussed so that the students can work with memory addressing in the next topics.
—*Instruction set* (45% of the subject). In a previous subject on digital systems, the students have learned how the Central Processing Unit (CPU) works with its processing and control unit. Therefore, in this topic, students learn an assembler language in order to be able to make more sense of how low-level programming works.
—*I/O subsystem* (45% of the subject). In this topic, students should learn what the I/O subsystem is, how it interfaces with other subsystems such as memory, and the different required techniques, such as polling and interruption. The concept of DMA is also introduced together with interruption.
—*Interconnection systems* (5% of the subject). Students have to know that all the communication between the different parts of the computer is carried out thanks to buses. Therefore, the different communication protocols are introduced in this topic.

It has to be said that this is not an isolated case of active learning strategies. Many of the subjects that the students will find during the computing degree are student centered. Specifically, this subject is part of a computer architecture and systems group of topics (Introduction to Digital Systems Design, Introduction to Computer Technology, Computer Structure, Computer Architecture, Introduction to Operating Systems, and Operating Systems). All of these subjects are, to a different extent, following student-centered and active-learning strategies.

4. THE NDS FOR COMPUTER I/O SUBSYSTEM EDUCATION

This experience started in the school year 2010–2011. However, a few years before, we had started studying the possibility of merging the use of a game console and an educational approach based on PjBL. During the research phase, several game-consoles were analyzed and the NDS was chosen due to its price, connectivity, and support issues [Larraza-Mendiluze 2014]. Besides, the use of the NDS is itself attractive for students.

The NDS is proprietary. To our knowledge, Nintendo does not provide the development kit to anyone other than licensed developers. However, there is a homebrew community offering an open toolchain, the devkitPro [devkitPro 2014], which, among others, has a library to program the NDS, called libnds. Studying this library, we found out how to use the NDS in the educational process of the computer I/O subsystem.

The proposed platform fulfills the requirements of the subject. Moreover, the libnds library of the devkitPro toolchain [devkitPro 2014] offers an interrupt controller programmed in the assembler. Since the second topic of the subject deals with assembler programming, this gave us the opportunity to look at the code and see how it worked. The interrupt controller provides us with the possibility of programming Interrupt Service Routines (ISRs) and managing the interrupt table.

Also, some of the NDS peripherals—such as the touchscreen, keyboard, and a configurable timer—have been programmed. Every key of the keyboard can be configured to interrupt or not, providing the opportunity to analyze polling and interrupt synchronization methods.

A limitation of the NDS is that all the I/O registers (interrupt controller configuration registers and peripheral registers) are mapped in the memory; thus, the possibility of accessing registers mapped in the I/O space does not exist.

Complications related to graphical interface and programming for both processors of the NDS can be avoided. The projects developed will run only in the ARM9. Related to the graphical interface, the entire environment was given to the students with an example of how to add new backgrounds and sprites.

For more details about the characteristics of the NDS, see Larraza-Mendiluze et al. [2013, Section III.B]. Anyone interested in the teaching course resources used during this experience should contact the first author of this article.

The NDS machine together with the devkitPro toolchain, offers the possibility to work on all the necessary I/O concepts. However, the hypothesis has another component: the PjBL educational approach. Four consecutive applications of the PjBL educational approach are shown in Section 6, together with the analysis of the results obtained each year, and comments on how these results have been used to determine future changes. Meanwhile, all data used in Section 6 can be found in Section 5.

5. DATA GATHERED AND RESULTS OBTAINED

This study covers four years thus, the experience warrants some reflection to be able to achieve better implementations of the educational approach. The implementations for the educational approach have been adapted year by year in an effort to get better results or to increase student motivation following the action research model [Altrichter et al. 2002]. It has to be mentioned that, in this scenario, teachers are learners as well [Boos and Krauss 2007]. This section describes first the data gathered in order to follow the previously stated design cycle and later the results obtained each year.

Table I. Items of the Satisfaction Questionnaire

Number	Item
1.	The course syllabus (program, goals, references, tutorials) was clear from the beginning.
2.	The classes have been adjusted as provided in the syllabus of the subject.
3.	The criteria and assessment method have been clear from the beginning.
4.	The assessment has taken into account assignments, participation in class, and other activities.
5.	The aims and schedule of support hours have been clear from the beginning.
6.	The contents of this subject have been associated with other materials.
7.	There has been enough practice and sufficient problems and cases have been presented for the correct understanding of the content.
8.	The educational approach used in the subject has fostered students' reflection, synthesis, and reasoning skills.
9.	The development of the subject has properly promoted critical and active participation of the student body.
10.	The educational approach has fostered collaboration and participation of students both in classroom activities and outside.
11.	The educational approach has been good to acquire work and study habits.
12.	The educational approach has resulted in a pleasant climate during the development of classroom activities and outside.
13.	The educational approach has facilitated communication so that students express their concerns and problems.
14.	The educational approach has used the strengths and weaknesses of the students to guide them to improve.
15.	The educational approach has helped the students to deal with conflicts, changes, tensions and limitations.
16.	The educational approach has encouraged communication between the students and has offered effective mechanisms to address situations that affect them academically.
17.	The educational approach has facilitated the evaluation results to serve as a new learning option.
18.	The evaluation system used has been adapted to the educational approach.
19.	The guidance provided by the lecturer during the process has satisfied students' needs.
20.	Overall, the assessment of the subject has been adequate.
	The educational approach has been useful to help:
21.	Understand theoretical content.
22.	Link theory and practice.
23.	Relate the contents of the course and obtain an integrated view.
24.	Increase interest and motivation toward the subject.
25.	Make decisions about a real situation.
26.	Solve problems or offer solutions to real situations.
27.	Develop your communication skills (oral or written).
28.	Develop your autonomy to learn.
29.	Take a participatory attitude about your learning.
30.	Improve your skills in group work.
31.	Develop skills needed in professional practice.

Every school year, at the end of the module, different data were gathered in order to prepare the implementation for the next school year. These data were based on the scores obtained by the students who followed the PjBL educational approach and the answers to a satisfaction questionnaire on the subject as a whole, which was handed out to the students. The satisfaction questionnaire had two parts, the first one comprising 31 Likert-type items on a scale from 1 to 5 (1 = strongly disagree, 2 = disagree, 3 = neither agree nor disagree, 4 = agree, 5 = strongly agree) (see Table I). The second part has two open questions: (1) Would you change something? Do you have any suggestion in order to improve the educational approach? and (2) Comment on something that you believe is important and has not been included in the questionnaire.

The responses obtained every year have been compared with the responses obtained the previous year using the Mann-Whitney U test. Table II shows the data obtained each year: the number of responses, the average and standard deviation, and the P-value from Mann-Whitney U test. However, the first year was not comparable with the previous one, because the previous year we did not ask the students to fill out such a satisfaction questionnaire. Considering the P-values, the ones that demonstrate a significant difference appear in bold type.

Before addressing Table III, it is worth mentioning that neither actively following the subject nor the project itself was mandatory. Therefore, students have the possibility of passing the course even they do not complete the project.

Taking that in consideration, Table III shows the number of students who started with the educational approach and the number of students who ended the course following it. Similarly, the teams that were formed at the beginning and the number of teams that reached the end of the course are also represented, followed by the number of students (out of those who started and those who finished) who did not take the exam, who did not pass it (scoring lower than 5 out of 10), who got a low score (scoring between 5 and 7), who got a good score (between 7 and 9), and those who got an excellent score (scoring 9 or higher).

6. ACTION RESEARCH MODEL CYCLES

This section is based on the action research model [Altrichter et al. 2002]. It will show the overall details of the educational approach, followed by each implementation, with reflection on the data obtained (which can be found in Section 5), and the changes proposed for the next year.

In order to fulfill the requirements of the project, students must perform certain steps. Some of these steps, and even the order in which they have been carried out, have changed along the years. In this section, the steps will be defined analyzing different configuration possibilities.

PjBL is a collaborative educational approach with two important features: positive interdependence and individual accountability [Johnson et al. 1998]. Positive interdependence requires that all members of a working team participate actively and contribute to the project, avoiding members absenting themselves from the group activity without consequences. Individual accountability requires that all teammates learn with the project all the topics involved. Positive interdependence is achieved with an adequate workload, but individual accountability is not ensured with this load. For example, one of the teammates may be designing spectacular images without knowing how the I/O subsystem works. For this reason, we decided that all students had to take an examination and get a minimum mark in order for the project score to be taken into account in their final evaluation.

In adhering to the PjBL educational approach, the leadership role taken by a lecturer in the classroom is minimized and reserved only for moments in which a special explanation is necessary. The rest of the time, the students' work is crucial and performed either individually or in teams. Therefore, one of the first phases of the project is the *team building* phase. During this phase, there is the interesting parameter of team size to be considered. Smaller team sizes require the lecturer to manage more teams. However, bigger team sizes might generate more internal problems in the team. Oakley et al. [2004] advise making teams of 3 or 4 members.

Once the teams are established, it is time to start the project itself. In order to start designing and developing such an I/O project, it is important for the students to know how to graphically describe the functionality of the system that they have designed. For

Table II. Results of the Questionnaires and P-values to Decide Whether the Differences with the Previous Year Are Significant
(n.: Number; av.: Average; stdev.: Standard Deviation)

Item Number	2010–2011			2011–2012				2012–2013				2013–2014			
	n	av.	stdev	n	av.	stdev	P-value	n	av.	stdev	P-value	n	av.	stdev	P-value
1	27	3.89	1.17	42	3.40	0.96	**0.028**	48	3.52	0.92	0.516	20	3.79	0.48	0.242
2	29	3.62	0.88	42	3.19	0.97	0.060	45	3.42	0.99	0.215	20	3.82	0.67	0.091
3	28	3.46	0.99	42	3.48	0.80	0.984	48	3.48	1.05	0.928	20	3.65	0.69	0.638
4	29	4.24	0.74	42	3.67	0.95	**0.010**	46	4.07	0.85	**0.034**	20	4.15	0.66	0.960
5	28	4.18	0.59	42	4.14	0.78	0.992	48	3.88	0.82	0.150	20	4.24	0.65	0.073
6	29	3.24	0.99	41	3.80	0.87	**0.022**	47	3.13	0.88	**0.002**	20	3.68	0.73	**0.006**
7	29	**2.55**	0.92	41	**2.66**	1.11	0.726	47	2.11	0.89	**0.020**	20	3.12	0.91	**0.000**
8	28	3.54	0.79	42	3.57	1.02	0.992	48	2.67	1.15	**0.001**	20	3.09	0.83	0.114
9	29	3.79	0.85	42	3.43	1.02	0.184	48	3.21	0.97	0.267	20	3.32	0.73	0.795
10	29	3.86	0.79	42	3.79	0.90	0.749	48	3.58	1.03	0.453	20	3.50	0.71	0.562
11	28	3.11	0.75	42	3.10	1.16	0.976	48	2.88	1.10	0.418	20	3.03	0.94	0.535
12	29	3.55	0.88	42	3.26	0.99	0.219	48	2.81	1.04	0.095	20	3.41	0.86	**0.013**
13	29	3.69	0.99	42	3.38	1.29	0.347	48	3.56	0.94	0.624	20	3.47	0.86	0.555
14	29	3.14	0.99	41	**2.78**	0.91	0.168	48	2.90	1.10	0.569	19	3.30	0.85	0.112
15	29	2.83	1.17	41	3.05	1.02	0.424	48	2.56	1.07	0.064	19	2.97	1.05	0.124
16	29	3.52	1.09	41	3.39	0.80	0.575	48	3.23	0.83	0.294	20	3.41	0.99	0.379
17	29	3.21	0.95	42	3.29	0.83	0.697	48	2.83	0.91	**0.029**	19	3.15	0.76	0.134
18	29	2.97	0.94	42	3.19	1.09	0.373	48	2.81	1.07	0.084	20	3.35	1.01	**0.019**
19	29	2.72	1.04	42	**2.64**	1.03	0.865	48	2.21	1.07	**0.048**	20	3.18	0.90	**0.000**
20	29	2.83	1.17	42	3.21	0.92	0.153	48	2.46	1.07	**0.001**	19	3.21	0.86	**0.002**
21	29	**2.28**	0.92	40	**2.63**	0.87	0.124	45	2.38	1.07	0.285	17	3.10	1.04	**0.011**
22	29	3.34	1.12	40	3.43	0.90	0.772	46	2.78	1.07	**0.012**	18	3.44	0.88	**0.014**
23	29	3.00	0.67	39	3.26	0.94	0.332	46	3.02	0.98	0.294	18	3.38	0.75	0.112
24	28	3.50	1.09	40	3.23	1.07	0.332	46	2.67	1.27	**0.026**	18	3.06	0.88	0.087
25	28	3.61	0.73	39	3.44	0.85	0.535	46	3.39	1.04	0.749	18	3.19	0.90	0.478
26	28	3.50	0.82	39	3.31	0.89	0.435	46	3.02	1.02	0.159	18	3.38	0.91	0.084
27	28	3.64	0.91	40	2.95	0.96	**0.015**	46	2.67	0.99	0.230	17	3.16	0.97	**0.043**
28	28	3.57	0.88	40	3.70	1.02	0.646	46	3.72	1.00	0.834	18	3.69	0.78	0.624
29	28	3.89	1.16	40	3.68	0.83	0.184	46	3.28	1.05	0.101	18	3.53	0.95	0.222
30	28	4.00	1.12	39	3.97	0.90	0.849	46	3.59	0.93	0.056	18	3.84	0.77	0.254
31	28	3.36	0.86	39	3.51	0.97	0.569	45	3.02	1.06	**0.046**	18	3.34	0.79	0.147

Table III. Students and Groups Who Started and Finished the Project and the Scores Students Got in Their Exam

School-year	Project	# of students	# of teams	# of students who did not take the exam	% of students who did not take the exam	# of students who did not pass exam	% of students who did not pass exam	# of students who passed with a low score (5-7)	% of students who passed with a low score (5-7)	# of students who passed with a good score (7-9)	% of students who passed with a low score (7-9)	# of students who passed with an excelent score (>=9)	% of students who passed with an excelent score (>=9)
10-11	started	41	8	12	29.27	5	12.20	3	7.32	19	46.34	2	4.88
	finished	29	6	0	0.00		17.24		10.34		65.52		6.90
11-12	started	57	13	17	29.82	9	15.79	16	28.07	10	17.54	5	8.77
	finished	46	12	6	13.04		19.57		34.78		21.74		10.87
12-13	started	40	14	2	5.00	3	7.50	18	45.00	15	37.50	2	5.00
	finished	39	14	1	2.57		7.69		46.15		38.46		5.13
13-14	started	36	12	5	13.89	6	16.67	7	19.44	15	41.67	3	8.33
	finished	31	11	0	0.00		19.35		22.58		48.39		9.68

that purpose, we teach them how to build state machines, with which they show how the I/O affects the state of the system. In order for the students to become competent in this endeavor, they are required to build the state machines of some previously defined I/O systems. This phase will be called the design learning.

The *definition of the problem* is a phase that can be adapted from implementation to implementation since it refers to the purpose of the development. The students could be asked to develop a simulator, a game, or any sort of application. Depending on the type of application that they have to develop for the NDS, their interest and the results obtained might change.

Once students know the definition of the problem and know how to design a project, they are ready to go to the design phase.

The introduction of the theory. In a PjBL educational approach, the theory is supposed to be learned while developing the project. However, this is one of the difficulties of PjBL. Students are wont to ask for a theoretical basis before starting to work on the project. Also, lecturers were not used to this system and the design of the lectures without the theoretical basis is not straightforward. Therefore, this phase has also undergone changes in the different implementations in order to study how the students perform with the different strategies and methods used to learn the I/O concepts.

The introduction of the technical details. Once students know the theoretical basis of the computer I/O subsystem, they need to know the specifics of the machine in order to work with it.

The merging phase is the one in which design, theoretical knowledge, and specific technical knowledge are merged in order to achieve the final product.

Probably, the best approach would be to have a cyclic design so that the students could learn from their mistakes in each interaction. However, this is not possible due to time limitations. Therefore, the lecturer reviews the team's work in each phase, making sure that there is nothing that would make the project fail, and the lecturer guides the students so that they can find the right way with just one iteration.

6.1. The First Implementation (School-Year 2010–2011)

Team size is a very important parameter in a PjBL educational approach. In this first implementation, we were concerned about having too many groups. Therefore, during the team building phase, we formed seven teams of five and one of six teammates. We went through the design learning phase directly to the definition of the problem phase.

Using an NDS machine to develop the project may appear to imply that the project should require the students to develop a game. However, management of the memory is quite complex and the lack of handbooks made it a real challenge. On the first occasion, we decided to use only static backgrounds, which made the goal of developing a game beyond our reach. Therefore, that first year, the students were asked to implement vending-machine simulators that would run on the NDS. In their design, they would have to synchronize the touchscreen, the keyboard, and a timer.

The design phase started right away. Once the students had delivered their design, the lecturer reviewed it and the team got feedback on its suitability.

The introduction of the theory phase started with a question: “What do we need in order to make/solve the project?” A lot of needs were put forward by the students; several were related to the I/O topic itself, whereas others were not. Among the needs related to the I/O subsystem, they asked for the communication needed between computer components, how to control the timing, and so forth. At that point, they were willing to read the course notes in order to be able to answer the questions. However, following the jigsaw technique [Aronson et al. 1978], each student got only part of the course-notes, worked on them, discussed them with the students that had received the same part, then rejoined their team, at which point each student explained his or her part of the theory. At that moment, the students were ready to enter into the introduction of the technical details phase. They would have to ask the lecturer for information on the different peripherals that they would have to use so that they could merge all their knowledge into their final product.

6.1.1. Results and Reflections of the First Implementation. Out of the 41 students who started the project, 29 finished. Two teams abandoned and 2 others each lost one member, ending up with 5 teams of 5 and 1 of 4 teammates. Six different vending machines were developed. The products sold in these machines were sandwiches, drinks, candy, pictures, train tickets, and petrol.

Of the students who started the project, 58% passed the topic, while 82% of the students who finished the project passed the topic. Moreover, 72% of the students who finished the project passed with at least a remarkable mark (70%–90%). Considering the results of the students who finished the project and then went on to pass the topic, it would clearly be interesting to raise the ratio of students who finish the project.

The questionnaires also provided some data worth analyzing. Table II shows how in the first implementation items 7 and 21 got the lowest scores. These two items are directly related to understanding of the content ((7) There has been enough practice and sufficient problems and cases have been presented for the correct understanding of the content; (21) The educational approach has been useful to help understand the theoretical content). Also, the open questions showed the need that the students felt to have more theoretical explanations. Out of 21 students, 18 asked for more lectures in their open-ended questions. Despite this being contradictory to the educational approach option, it was a point to consider for the next implementation.

From our experience and the data, we concluded that, although there were not many teams to deal with (just seven), such big teams were very difficult to manage. Team internal conflict was probably a factor in 30% of students not finishing the projects. Therefore, we decided to make the teams a little smaller.

In addition, students are not used to learning while doing and they needed to have the theory available sooner.

6.2. The Second Implementation (School Year 2011–2012)

This second-year group was even bigger than the previous one, but, at the team-building phase, the decision made was to build teams with no more than five members. Some students, though, decided to form teams of three. There were 13 teams of three to five teammates.

This time, after the team-building and design-learning phases, we went on to the introduction of the theory phase. We did not want to introduce classical lectures in the educational approach, and we thought that bringing forward the introduction of the theory phase could help fill the need for lectures that the students had made evident. Again, parts of the course notes were handed to the students in order to work on following the previously explained jigsaw technique [Aronson et al. 1978]. Once each team had discussed all parts of the topic, all students received the entire set of course notes so that they could have a complete perspective. This decision was made because some of the students were unhappy having only part of the notes, because their teammates were not able to adequately complete the information gaps. This measure was also taken following the open-ended questions of the satisfaction questionnaire.

By then, we were able to move sprites on both displays. This opened the door for designing games, which we thought would be more attractive for the students. Therefore, we changed the definition of the project. The students would have to implement a simple game using the same NDS devices of the previous year. Of course, the team size was also taken into account when defining the workload of the projects.

By this time students should know the theory and should also be able to design an I/O system; therefore, it was time to start the design phase. The result of this phase would be reviewed by the lecturer and the team would get feedback on its suitability.

Meanwhile, in order to discuss and promote motivation, a new question was proposed since the theory was already known. The new question was: "How can this be done on the NDS?" This way, the design phase was developed together with the phase in which the technical details are introduced. The students posed a few questions on I/O registers, but this time they were much more attracted by the idea of developing their design than in hearing explanations from the lecturer.

6.2.1. Results and Reflections of the Second Implementation. The results of this school year show 57 students who started the project and 46 who finished it. Twelve different games were developed. However, one of the biggest teams abandoned, four lost one or two members, and two more had to be rearranged. Many factors have influenced these results, but the lecturers' observations point out that smaller groups are easier to manage. However, the pass ratio for the finished projects fell by 15 points (adding up the passes, good, and excellent ratios, the pass ratio for the first year was 82.76% and for the second year 67.39%), which may indicate that many of the teammates had been unable to follow the course but had only managed to finish through support from the team. During the project development, the lecturers had observed that the students were more interested in the game than in the topic. This observation was then reflected in the figures.

Not only did the students' marks fall, but also their satisfaction levels, as can be seen in Table II.

Although some items rose slightly, the differences for most of them were not significant, as shown by the P-values of the Mann-Whitney test. The P-values in bold show the items with statistically significant differences. Only item 6 got a significantly better score ((6) The contents of this subject have been associated with other materials.). Students have understood better the relationship among the topic and other materials. On the other hand, the students did not see how the educational approach had helped them and the class atmosphere was not seen as positive.

The observations showed that, as hypothesized, students were very attracted to the idea of building a game for a hand-held console. However, the results were not as good as hoped, perhaps because building an attractive game was more important for some students than getting to know the computer I/O subsystem. More students excelled, the ones that really cared about the subject, but many others devoted their time to drawing aesthetically pleasing images for the game and therefore failed the subject. This division could also be the cause of the decline in the classroom atmosphere. This setback was taken into account in the next definition of the problem phase, which emphasized the definition of an attractive project for the NDS but limited their graphic interface design possibilities.

The teams that performed better were those consisting of three teammates. This was probably due to their attitude, but we realized that in the bigger teams students still had management problems. Therefore, although the class was still big, we decided to form groups of three teammates.

In this implementation, all the items of the questionnaire got a score above 2.6. However, there were some below 3.0, and this time we tried to monitor and support these low scorers more closely. As well as items 7 and 21, we had to consider items 14 and 19 ((14) The educational approach has used the strengths and weaknesses of the students to guide them to improve; (19) The guidance provided by the lecturer during the process has satisfied students' needs), which showed that the students felt a lack of guidance. The results of the closed items coincided with the open-ended questions. Students asked for more lectures, more guidance, more resolved exercises, and so on.

The lecturers studied the problem and realized that this was not associated with delivering the course notes sooner or later, but more connected with the size of the group (between 45 and 65 for the most recent courses). It was very difficult for the teacher to guide all the teams in their discussions about the theory using the jigsaw technique [Aronson et al. 1978]. Thus, the lecturers decided to try a different technique.

6.3. The Third Implementation (School-Year 2012–2013)

In the third-year implementation, the proposal was for the teams to be comprised of three teammates. There should be a group of four teammates, but the students preferred to form two pairs. The phases followed the same order, but, in the introduction of the theory phase, the jigsaw technique was replaced by an inquiry-based technique [Kahn and O'Rourke 2005], in which the lecturers proposed questions that the students had to answer based on their own knowledge. This opened the way to a discussion in which the theoretical concepts of the topic were outlined. Finally, the students had to answer the questions on their own with the help of the course notes.

The project was redefined in order to get the students thinking about the topic at hand, instead of merely designing pleasing graphics. The game would be similar to the 100-doors game, in which the player has to unlock the doors using the machine controls and advance to more difficult levels. Each team would have to develop a mini-game to open one door. The design of the mini-games was carefully revised by the lecturers, so they were not too complicated to develop. Of course, the students' initiative must not be repressed and they were given the possibility for improving their projects once the I/O was controlled as the project required.

As in the previous implementations, for the introduction of the technical details, the students were encouraged to request anything they needed while developing the project.

This year, in order to achieve a better understanding of how the students were developing their knowledge of the topic, we decided to take a further step with the introduction of a new tool, concept maps. From the outset, together with the state machines, the students received a lesson on how concept maps are built. Then, while the topic was being developed the students had to deliver their concept maps, for which they received feedback. The results of the analysis of those concept maps can be found in Larraza-Mendiluze and Garay-Vitoria [2013]. All concept maps (in Basque) can be found in Appendix D of Larraza-Mendiluze [2014].

6.3.1. Results and Reflections of the Third Implementation. Out of the 40 students who started the project in 14 teams, 39 finished, developing 14 mini-games. The dropout ratio fell dramatically from almost 30% to 20% in the previous years to 5%. Also, the fail ratio fell from 12% to 15% the previous years to 7.5%. Out of the students who finished their projects, almost 90% passed. Was this due to the reduction of the team size? Due to the change from the jigsaw technique to the inquiry-based learning technique? Due to restrictions set to the project definition? Due to the concept maps? Or due to some other variables that we have not considered, such as the students themselves? Our hypothesis is that all the parameters contributed; also, the hard work students had to do to build their concept maps helped them to pursue their objective and helped them study.

Despite the high pass ratios, the satisfaction questionnaire shows an overall decline in satisfaction, statistically significant in many cases. Fortunately, that year, we had added two more questions to the satisfaction questionnaire, asking whether concept maps and the project itself had been useful during the learning process. The mean values of these questions were 2.47 and 3.88, respectively. It seemed that students did not like the use of concept maps. Moreover, they were still asking for more lectures in the open-ended questions.

This year's implementation, with small teams and a well-bounded project design, together with the new learning techniques, produced good results. Although the pass ratio was very good, there was still room for improvement, to try to raise the scores from pass to remarkable and excellent. As for satisfaction, the hypothesis was that the effort required to build the concept maps was unsatisfactory for the students, which was reflected throughout the questionnaire. We decided not to use the concept maps in the next implementation to test its effects on scores and satisfaction.

6.4. The Fourth Implementation (School Year 2013–2014)

The fourth implementation was a repeat of the third one, but without the concept-maps, solely to see whether the benefits and detriments observed in the previous year's marks and satisfaction questionnaire were directly related to the use of the concept-maps.

6.4.1. Results and Reflections of the Fourth Implementation. Out of the 36 students who started the project, 31 finished, developing 11 mini-games. As Table III shows, the fourth year's marks are closer to the second year's than to the third year's. The fourth year's marks are slightly better in the sense that the ratio of students who finished with an intermediate mark is higher, and the ratio of students who dropped out is smaller.

Also, the satisfaction questionnaire shows a general satisfaction increase, in which all the medians are three or over with no downward trend. The P values showed that many of these differences were significant. In the open-ended questions, a few students still asked for more lectures, but the number dropped considerably.

The fourth implementation's results show that the hypothesis formulated the previous year was correct. The students were not very satisfied with the effort that they had had to make to build the concept maps, although it was probably good for their marks.

Table IV summarizes the most important changes made between features related to four iterations.

7. DISCUSSION AND CONCLUSION

During the four year experience, it was found that the educational approach suited the I/O topic, as students developed expected I/O skills. On the other hand, teachers are

Table IV. Differences Among The Iterations

	2010–2011 year	2011–2012 year	2012–2013 year	2013–2014 year
# of students	41	57	40	36
# of teams	8	13	14	12
Size of teams	5–6 members	3–5 members	2–3 members	3–members
Project description	Vending machine	Open game	100-doors game	100-doors game
Type of background	Static	Dynamic	Dynamic	Dynamic
Technique used for introduction of the theory	Jigsaw	Jigsaw	Inquiry-based	Inquiry-based
Use of concept maps	No	No	Yes	No

using a PjBL educational approach in other subjects related to computer architecture and technology, operating systems, and embedded systems, in which results are similar to this topic.

Looking at the results of the questionnaires, considerable information has been extracted. For example, students' interest and motivation (item 24, increased interest and motivation toward the subject) has achieved an average value of 3.12, taking all four years into account.

With relation to the perceived mastery of students during these years, the educational approach used (item 8, educational approach used in the subject has fostered students' reflection, synthesis and reasoning skills) has 3.22 as average value; to develop skills needed in professional practice (item 31, develop skills needed in professional practice) has 3.41 as an average value. However, there are other items with lower averages, such as item 7, related to the quantity of practice, problems and cases for the correct understanding of the content, which has achieved, on average, a value of 2.61; and item 21, related to understanding theoretical content, which has an average value of 2.60. In both cases (items 7 and 21), the highest score was achieved on course 2013–2014, reflecting a considerable increase over the previous courses.

Moreover, the peer-support effect is very well appreciated throughout the four years, as can be seen by analyzing item 10, the educational approach has fostered collaboration and participation of students both in classroom activities and outside (3.68, on average); item 16, the educational approach has encouraged communication between the students and has offered effective mechanisms to address situations that affect them academically (3.39, on average); and item 30, improve your skills in group work (3.85, on average).

Maintaining positive results in the questionnaires while reinforcing the ones that are lower than 3 should be the next objective. Changing the questionnaire should also be considered, however, as we have realized that all the items have been phrased positively, possibly inserting a bias into the evaluation.

As for the parameters that have been analyzed throughout the four implementations, one is team size. Our conclusion is that, as the students are not yet used to working in teams, it is better to have smaller teams than to have a reduced number of teams. The internal obstacles within teams were more difficult to overcome in bigger teams.

Another parameter that we changed is the technique for working on the theory. At the beginning, we used the jigsaw technique [Aronson et al. 1978], but we found it difficult to guide and assess all the teams in a big group. Thus, we changed this technique to inquiry-based learning [Kahn and O'Rourke 2005], in which the whole group (all the students in class) discussed problems and issues together. The challenge for the lecturer, in this case, consisted of achieving the participation of the majority of the students.

The definition of the project is another variable for the different implementations. We started with a very well-defined project, the vending machine simulator, which may have been less attractive for the students, but worked very well. We then changed to the game, more attractive, but less defined, that became a hole in which many students got stuck because of their enthusiasm to develop an attractive game. Finally, the option that worked better was a mini-game. The design was open, but with very well-established restrictions. The topic of the project was not included in the questionnaire, nor was it studied from the students' perspective. Only the observations of the lecturers were considered. The results of the last two implementations show that the 100-doors game gave students enough freedom with the design, and was also restrictive enough for the team to be able to finish it on time.

Finally, with regard to the concept maps, the students did not appreciate the effort of having to build them, nor what was, in their consideration, being told how to learn. They articulated this feeling both personally and through the satisfaction questionnaire. Moreover, we think that this affected the overall assessment, and that this is the reason behind the overall score decrease of the third year. However, at the time of submitting the questionnaire, the students were not aware of the increase in the pass ratio in the computer I/O subsystem topic that year. Therefore, this tool will remain under consideration. More research will be needed in order to decide how or whether to use it. A possibility could be to use the concept maps in other subjects as well, so that the impact of learning the use of the tool itself might be reduced. In that case, it would be interesting to see whether the scores of the other subjects also benefit from the concept maps.

This article shows a four year action research experience using an NDS game console (a bare machine) in combination with a PjBL environment to learn I/O subsystems using the programming approach. The results achieved by students following this educational approach are appropriate; thus, we are not going to discard it. Several parameters have been adjusted during each of the four years of the experience, such as the team sizes, the educational approach for theory work, the definition of the project, and the use or not of concept maps. As the bibliography indicates, teams work better with 3 or 4 members. In our case, being a first-year subject, we recommend teams of 3 members. With more mature students, teams of 4 could also work well. Considering the technique used for working on the theory, in our case, the inquiry-based technique has worked better. We believe that the group size has had an influence on this. Concept maps seem to work well, but they are not well received by the students. This part needs more work.

Based on the acquired experience, new implementations will be applied in future years. For example, we will consider making the project mandatory for all students. We will also analyze how to apply these implementations to other subjects related to embedded systems and real-time systems for which bare machines are needed.

ACKNOWLEDGMENTS

The authors would like to acknowledge all the students who took part in the experience.

REFERENCES

ACM/IEEE Computer Society. 2008. Computer science curriculum 2008: An interim revision of CS 2001. *Report From the Interim Review Task Force, Association for Computing Machinery, IEEE Computer Society*. Retrieved April 20, 2016 from http://www.acm.org/education/curricula/ComputerScience2008.pdf.

Herbert Altrichter, Stephen Kemmis, Robin McTaggart, and Ortrun Zuber-Skerritt. 2002. The concept of action research. *The Learning Organization* 9, 3, 125–131. DOI:http://dx.doi.org/10.1108/09696470210428840

Elliot Aronson, Nancy T. Blaney, Cookie Stephan, Jev Sikes, and M. Snapp. 1978. *The Jigsaw Classroom*. Sage Publishing Company, Beverly Hills, CA.

Anders Berglund and Anna Eckerdal. 2006. What do CS students try to learn? Insights from a distributed, project-based course in computer systems. *Computer Science Education* 16, 3, 185–195. DOI:http://dx.doi.org/10.1080/08993400600912386

Michael D. Black and Priyadarshini Komala. 2011. A full system x86 simulator for teaching computer organization. In *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education (SIGCSE'11)*. ACM, New York, NY, 365–370. DOI:http://dx.doi.org/10.1145/1953163.1953252

Michael D. Black and Nathaniel Waggoner. 2013. Emumaker86: A hardware simulator for teaching CPU design. In *Proceedings of the 44th ACM Technical Symposium on Computer Science Education* (*SIGCSE'13)*. ACM, New York, NY, 323–328. DOI:http://dx.doi.org/10.1145/2445196.2445294

Suzie Boss and Jane Krauss. 2007. Reinventing project-based learning: Your field guide to real-world projects in the digital age. *International Society for Technology in Education* (ISTE).

Mats Brorsson. 2002. MipsIt: A simulation and development environment using animation for computer architecture education. In *Proceedings of the 2002 Workshop on Computer Architecture Education*: *Held in Conjunction With the 29th International Symposium on Computer Architecture (WCAE'02)*, Anchorage, AK, May 26, Article No. 12. DOI:http://dx.doi.org/10.1145/1275462.1275479

Dennis Brylow and Bina Ramamurthy. 2009. Nexos: A Next Generation Embedded Systems Laboratory. *SIGBED Review* 6, 1, 7. DOI:http://dx.doi.org/10.1145/1534480.1534487

devkitPro. 2014. devkitPro: Homebrew toolchains for wii, gamecube, 3ds, ds, gba, gp32 and psp. Retrieved April 20, 2016 from http://sourceforge.net/projects/devkitpro/.

Andrés Díaz Lantada, Pilar Lafont Morgado, Juan M. Muñoz-Guijosa, José L. Muñoz Sanz, Javier Echávarri Otero, Julio Muñoz García, Enrique Chacón Tanarro, and Eduardo de la Guerra Ochoa. 2013. Towards successful project-based teaching-learning experiences in engineering education. *International Journal of Engineering Education* 29, 2, 476–490. Retrieved April 20, 2016 from http://www.ijee.ie/latestissues/Vol29-2/19_ijee2697ns.pdf.

John L. Donaldson, Richard M. Salter, and Rebecca E. Punch. 2011. Dlsys: A toolkit for design and simulation of computer system architecture. In *Proceedings of the 2011 Workshop on Computer Architecture Education (WCAE'11)*, San Antonio, TX, February 13, 0–6. Retrieved April 20, 2016 from https://www.ncsu.edu/wcae/HPCA2011/p9-donaldson.pdf.

Daniel Ellard, David Holland, Nicholas Murphy, and Margo Seltzer. 2002. On the design of a new CPU architecture for pedagogical purposes. In *Proceedings of the 2002 Workshop on Computer Architecture Education*: *Held in Conjunction With the 29th International Symposium on Computer Architecture (WCAE'02)*, Anchorage, AK, May 26, Article No. 6. DOI:http://dx.doi.org/10.1145/1275462.1275471

IEEE Computer Society/ACM. 2004. Joint task force on computing curricula, computer engineering 2004. *Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering*, IEEE Computer Society Press and ACM Press, 2004. Retrieved April 20, 2016 from http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf.

David W. Johnson, Roger T. Johnson, and Edythe J. Holubec. 1998. *Cooperation in the Classroom*. Allyn and Bacon, Boston, MA.

Peter Kahn and Karen O'Rourke. 2005. Understanding enquiry-based learning (EBL). In *Handbook of Enquiry & Problem Based Learning*, T. Barrett, I. Mac Labhrainn, H. Fallon, (Eds.). 1–12. CELT, Galway Retrieved April 20, 2016 from http://www.aishe.org/readings/2005-2/chapter1.pdf.

Edurne Larraza-Mendiluze. 2014. *The Computer Input/Output Subsystem Education in An Undergraduate Introductory Course: A Multiperspective Study*, Servicio Editorial de la Universidad del País Vasco/Euskal Herriko Unibertsitatearen Argitalpen Zerbitzua, 2014. Retrieved April 20, 2016 from http://hdl.handle.net/10810/13238

Edurne Larraza-Mendiluze and Nestor Garay-Vitoria. 2013. Use of concept maps to analyze students' understanding of the I/O subsystem. In *Proceedings of the 13th Koli Calling 2013 International Conference on Computing Education Research (Koli Calling'13)*. ACM, New York, NY, 67–76. DOI:http://dx.doi.org/10.1145/2526968.2526976

Edurne Larraza-Mendiluze and Nestor Garay-Vitoria. 2015. Approaches and tools used to teach the computer input/output subsystem: A survey. *IEEE Transactions on Education* 58, 1, 1–6. DOI:http://dx.doi.org/10.1109/TE.2014.2310711

Edurne Larraza-Mendiluze, Nestor Garay-Vitoria, José Martín, Javier Muguerza, Txelo Ruiz-Vázquez, Iratxe Soraluze, José F. Lukas, and Karlos Santiago. 2013. Game-console-based projects for learning the computer input/output subsystem. *IEEE Transactions on Education* 56, 4, 453–458. DOI:http://dx.doi.org/10.1109/TE.2013.2255877

Alejandra Martínez-Monés, Eduardo Gómez-Sánchez, Yannis A. Dimitriadis, Iván M. Jorrín-Abellán, Bartolomé Rubia-Avi, and Guillermo Vega-Gorgojo. 2005. Multiple case studies to enhance project-based learning in a computer architecture course. *IEEE Transactions on Education* 48, 3, 482–489. DOI:http://dx.doi.org/10.1109/TE.2005.849754

Kali P. Nepal. 2013. Comparative evaluation of PBL and traditional lecture-based teaching in undergraduate engineering courses: Evidence from controlled learning environment. *International Journal of Engineering Education* 29, 1, 17–22. Retrieved April 20, 2016 from http://www.ijee.ie/latestissues/Vol29-1/03_ijee2680ns.pdf.

Barbara Oakley, Richard M. Felder, Rebecca Brent, and Imad Elhajj. 2004. Turning student groups into effective teams. *Journal of Student Centered Learning* 2, 1, 9–34. Retrieved April 20, 2016 from http://www4.ncsu.edu/unity/lockers/users/f/felder/public/Papers/Oakley-paper%28JSCL%29.pdf.

Umakishore Ramachandran and William D. Leahy Jr. 2007. An integrated approach to teaching computer systems architecture. In *Proceedings of the 2007 Workshop on Computer Architecture Education (WCAE'07)*, San Diego, CA, June 9, 38–43. DOI:http://dx.doi.org/10.1145/1275633.1275641

María J. Santofimia and Francisco Moya. 2009. Nintendo DS: A pedagogical approach to teach computer architecture. In *ESA*, H. R. Arabnia and A. M. G. Solo, (Eds.). Las Vegas, NV, 269–273. Retrieved April 20, 2016 from http://arco.esi.uclm.es/public/papers/2009-ESA-mariajose.santofimia.pdf.

Terry A. Scott. 2000. Illustrating programmed and interrupt driven I/O. *Journal of Computing Science in Colleges* 16, 1, 230–238. Retrieved April 20, 2016 from http://dl.acm.org/ft_gateway.cfm?id=357737&type=pdf.

Timothy D. Stanley, Lap K. Wong, Daniel Prigmore, Justin Benson, Nathan Fishler, Leslie D. Fife, and Don Colton. 2007. From archi torture to architecture: Undergraduate students design and implement computers using the multimedia logic emulator. *Computer Science Education* 17, 2, 141–152. DOI:http://dx.doi.org/10.1080/08993400601165735

Patricia J. Teller, Manuel Nieto, and Steve Roach. 2003. Combining learning strategies and tools in a first course in computer architecture. *Symposium on Computer Architecture*, Article 8. DOI:http://dx.doi.org/10.1145/1275521.1275533

Allison E. Tew, Brian Dorn, William D. Leahy Jr., and Mark Guzdial. 2008. Context as support for learning computer organization. *Journal on Educational Resources in Computing* 8, 3, 1–18. DOI:http://dx.doi.org/10.1145/1404935.1404937

John W. Thomas. 2000. *A Review of Research on Project-Based Learning*. Report prepared for The Autodesk Foundation. Retrieved April 20, 2016 from http://www.bie.org/images/uploads/general/9d06758fd346969cb63653d00dca55c0.pdf.

Timothy Urness. 2007. Teaching computer organization/architecture by building a computer. In *Proceedings of the 2007 Workshop on Computer Architecture Education* (*WCAE'07*), San Diego, CA, June 9, 72–76. DOI:http://dx.doi.org/10.1145/1275633.1275650

Received June 2015; revised January 2016; accepted February 2016