IJIRSET
International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

e-ISSN: 2319-8753 | p-ISSN: 2347-6710

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 10, Special Issue 3, December 2021

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 7.569

9940 572 462 | 6381 907 438 | ijirset@gmail.com | www.ijirset.com

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

Robotics Vs Game-Console-Based Platforms to Learn Computer Architecture

Muhsina T.M, Dr Rasmi A

Student at Royal College of Engineering and Technology, APJ Abdul Kalam Technological University, India

Assistant Professor at Royal College of Engineering and Technology, India

ABSTRACT: The fields of robotics and game consoles offer an interesting and broad range of lab platforms with appropriate characteristics for teaching Computer Architecture concepts. This work analyzes the impact of one approach based on game consoles and another one based on robotics from a triple dimen- sion: student motivation, acquired knowledge, and perception of the employed platform. The study has been carried out on a sample of 96 students using the Arduino-based robot and 75 students using the Nintendo-DS console. A mixed methodology is employed encompassing quantitative and qualitative approaches. Five instruments are used to measure the three aforementioned dimensions. Results show that despite both platforms performing similarly in the three considered dimensions (student moti- vation, acquired knowledge, and perception of the employed platform), the robotics platform does it slightly better than game console, based on the obtained average scores for the considered instruments. Despite this outperforming, motivation and perception decrease for the students using the robotics platform as result of some identified constraint. This suggests that changes are required in the organization of the lab sessions to promote teamwork skills and to overcome the lack of simulators to remove the obstacles hinting motivation and performance. However, a clear correlation between motivation and perception and acquired knowledge has not been identified on computer architecture. Implications of affordances and constraints of both platforms, types of activities, and their impact on results have been discussed.

KEYWORDS: Computer architecture, Arduino, Nintendo DS, game consoles.

I. INTRODUCTION

The Association for Computing Machinery (ACM) and IEEE Computer Society Joint Task Group on Computer En- gineering Curricula consider the Computer Architecture and Organization area part of the body of knowledge for the curricula of Electrical Engineering, Computer Engineering, and Computer Science [1]. Despite its importance, this area has traditionally suffered from poor student performance [2]. The fact that most Computer Engineering or Computer Science students are more interested in software aspects than in those related to hardware [3] and those with hardware interests tend to choose Electrical and Electronic Engineering degrees is one of the reasons explaining the lack of engagement and the under performance in this area in the different degrees. The imple-mentation of the European Higher Education Area brought along new teaching methodologies and evaluation methods that are not performing as well as expected [4] what is increasing the pressure for new approaches that help improving students' motivation and performance. Teaching Computer Architecture is therefore a challenging task. The role that the practical platform plays in the teaching process is essential, and many different approaches have been proposed to this end [5], [6] being a recurrent discussion whether simulators or virtual laboratories should be adopted instead of real platforms [7]. It is noticeable that students that follow courses of Computer Architecture are usually in the first course of the degrees, what implies they are frequently unexperienced with these fundamental skills. Many of them are not enough familiarized with suitable forms of notation, necessary to develop mental models [16], together with a lack of programming concepts and language syntax, what may lead to demotivation, hindering learning [13], so the pedagogical approaches used in teaching should be aimed at fill in these gaps. Robots and computer games reach this objective. Traditional approaches for teaching Computer Architecture frequently leads to a poor performance of students, being far from a constructivist approach [17] in which the knowledge is actively constructed by the learner [18], [19]. In this vein, Papert and Harel in [20] coined the term constructionism, a derivative or constructivism, based on learning is much more meaningful when students create tangible or shareable artifacts, mobilizing their knowledge and skills [21]. Robots [22] and computer games [23], [24] are good examples of engaging artifacts to discover knowledge with educational interest in Computing Education. The benefits that Nintendo DS brings as a lab platform for teaching Computer Architecture concepts have been demonstrated [25],

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

[26] and experienced by the University of Castilla-La Man- cha, pioneer in adopting this platform during the academic year 2007/2008, as a pedagogical resource. Despite being a great pedagogical asset, Nintendo DS is, at the moment, an obsolete gaming platform. Finding replacement is becoming more and more challenging. For that reason, the academic staff involved in teaching this course faced the dilemma of whether Nintendo DS should be replaced by another game console or change the approach to the currently hot-topic issue of educational robotics. The study presented here was conducted to collect evidences and make an informed decision on what approach should be implemented next: a game- console based or a robotics one. Both robotics and game- console based platforms have demonstrated positive impact on student motivation [27]–[31]. Additionally, their hardware fea- tures are very interesting, in terms of provided resources and constrained complexity, for the purpose of teaching Computer Architecture. These two approaches therefore satisfy the Com- puter Engineering and Computer Science students demand of having attractive and real platforms for experiencing with Computer Architecture concepts [32].This paper presents two specific platforms representing these two approaches: a fit-for- purpose Arduino based robot and the Nintendo-DS console. Appropriate hands-on sessions have been designed to cover the content of the course Computer Structure and learning outcomes have been gathered, analyzed and compared from the perspective of the student motivation, knowledge, and perception of the employed platform.

A. Thesis Outline

The thesis is organised in the following manner: In chap- ter 1, some overview, including the background, motivation, objectives, scope of work are presented.In chapter 2 briefly describes the existing teaching methods of computer architec- ture and its impact. In chapter 3 contains the methodology of the system. Chapter 4 contains the result and analysis of the study.Finally, Chapter 5 contains the conclusion of this seminar report and some recommendations for future work.

II. LITERATURE SURVEY

Within education, concepts such as distance learning, and open universities, are now becoming more widely used for teaching and learning. However, due to the nature of the subject domain, the teaching of Science, Technology, and Engineering are still relatively behind when using new techno- logical approaches (particularly for online distance learning). The reason for this discrepancy lies in the fact that these fields often require laboratory exercises to provide effective skill acquisition and hands-on experience. Often it is difficult to make these laboratories accessible for online access. Either the real lab needs to be enabled for remote access or it needs to be replicated as a fully software-based virtual lab[34].

A. The Professor's Dilemma

In practice computer architecture is taught by real professors to real students; and that rather complicates matters. A Session demonstrates that authors usually select a real commercially available device as a vehicle to illustrate the course; for example the Motorola 68K, the Intel Pentium 4, or MIPS. Why do professors make life difficult for themselves by using CPUs that were made by engineers wanting to maximize market penetration and company profits? Why don't they specify a simple, hypothetical teaching machine and make it easier to teach the subject? Some professors do invent their own machines; I do myself for the first two weeks of the course. Most do not. Professors I have spoken to say that students do not want to use hypothetical hardware because they feel it is unrealistic and does not give a true picture of the real world they will soon be entering. Moreover, I find that students prefer to use hardware like the PC because they feel familiar with it. When I based my courses on the 68K microprocessor, it was well received by students in the days of the Apple Mac that incorporated it. When the 68K was dropped by Apple, students became less enthusiastic[36].

B. Virtual Laboratories

Despite the importance that practical training has on the learning process to give teaching a constructivist approach and promote learning, students very often find this experience compromised by the lack of appropriate equipment [33]. In this reality, the use of software-based simulators has become a common practice for undergraduate courses, especially in science and engineering disciplines. The advent of advanced computer graphics, augmented and virtual reality are leading to sophisticated virtual laboratories that provide a close-to-real experience overcoming the inconveniences associated to real laboratories (equipment failure and maintenance, associated cost, limited number, etc.) and allowing the students to focus on the most important aspects. However, these approaches are still at a very early stage and their use is commonly reserved for supporting initial steps. Still, the use of computer-based simulators has an important

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

negative impact on the student motivation [34] who rather use traditional real equipment for more in-depth hands-on experiences [35].

C. Real Platforms

A trade-off must be found between the positive impact of using real platforms and the overwhelming details that they might entail. Students demand real platforms that provide them with a realistic picture of what is on the market, whereas on the other hand, market products are specifically designed to maximize market penetration and company profits, which make them, most of the time, inappropriate for learning pur- poses. An additional obstacle that must be faced when looking for a real platform for teaching Computer Architecture is the fact that students are polarized regarding their interest about software and hardware aspects. Finding a platform that fits these two polarized groups is not easy because those keener on hardware-related aspects might found as a too limited platform one that might be perceived as too cumbersome for those other more interested in software aspects [26].The solution to this dilemma consists in employing a platform with a fast learning curve but with enough challenging resources to meet the expectation of those more experienced students.

D. Selecting a Processor for Teaching Computer Architecture ARM

Teachers of computer architecture and organization courses have to choose a target processor to illustrate the basic principles of instruction set design. This method suggest that it is time to choose the ARM processor architecture that is markedly different to those used in most current courses. A specific computer architecture is required as a vehicle to teach about registers, addressing modes, instruction types, and so on. Resorting to a hypothetical teaching machine reduces the student's learning burden and makes their learning curve shallow, but failing to introduce them to the complexities they will encounter in the real world can destroy their motivation. Teachers are concerned not only with covering a body of knowledge; they must motivate students and create a sense of excitement. In a discipline as rapidly changing as computer science, only those students who can adapt to change are likely to thrive over the four or more decades of their career. It explains why the ARM architecture is an excellent vehicle for teaching computer architecture; in particular, its predicated execution, inclusion of shifting in all data-processing instruc- tions, and its compressed code (Thumb) mode. Moreover, the ARM has a RISC architecture with some traditional CISC architectural features[36]

E. Game-based approaches For Learning Computer Archite- cure

The use of game-based approaches has different pedagogical advantages. Previous research found that a lack of initial affinity for playing computer games was not a barrier to enjoy the scientific experience of game construction [37].Game- based construction learning as a suitable approach to de- velop higher-order thinking and Computer Science abstraction skills, becoming an enjoyable approach for teaching Computer Science Education, even if they remark as a constraint the sophisticated level of programming skills knowledge needed.

F. Educational Robotics

An alternative tool for teaching Computer Architecture, which is also very appealing to students, is the use of robotics, and more precisely the so-called educational robotics. This ap- proach has recently gained attention [40] as more evidences are being obtained demonstrating the positive results in learning [41], [42]. Due to the versatility of robot platforms [30], a wide range of applications can be pursued, turning this platform into a unique tool to target a wide audience, from primary and secondary school students up to higher education [43]. The use of robots in education has benefits as the development of thinking and social skills [44], are of high interest in all educational stages. Robots are also a powerful tool in the teaching and learning process due to their interactivity and flexibility [30], [45], and the wide variety of enjoyable hands- on experiences they provide [46].

G. Arduiono DS

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs - light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online. You can tell your board what to do by sending a set of

instructions to the microcontroller on the board. To do so you use the Arduino programming language (based onWiring), and the Arduino Software (IDE), based on Processing. Over the years Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers - students, hobbyists, artists, programmers, and professionals - has gathered around this open-source platform, their

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike. Arduino is a small device (microprocessor) connected to PC with USB. Artists, designers,architects, hobbyists, students and college students learn programming and build entire projects for automation and robotic applications. So children especially flash Leds and start programming learning. The interaction starts . Arduino platform is based on open source technologies. The software development environment based on Processing and Wiring, which are open source languages, look like C++ in some conversions, but are easier in programming understanding [47]

H. LEGO Mindstorms NXT

LEGO Mindstorms NXT robots are being increasingly used in undergraduate courses, mostly in robotics-related subjects. But other engineering topics, like the ones found in data acquisition, control and real-time subjects, also have difficult concepts that can be well understood only with good lab exercises. Such exercises require physical educational tools that should be low cost, easy to configure and use, multi- purpose and motivational for the students, being all of this hard to achieve with a single device. The "classical" solution has been to acquire specific commercial kits for each subject, or even topic, usually proprietary and expensive. Our work extends the already existing alternative of using the LEGO Mindstorms NXT robots as a training platform, but not by imitating the same approach of commercial kits (e.g., to isolate some part of the robot for teaching a particular topic); we rather aim at accomplishing all the mentioned requirements simultaneously. For that purpose, we have used only one out-of-the-box, complete robot configuration, to be shared among different subjects without hardware/software/firmware modifications. This has reduced significantly the effort of a group of professors when preparing exercises, and encouraged the reuse of their work among several topics and subjects. Also, we have collected a number of surveys on students and the professors' experiences. In this paper we describe our approach and present in detail the results, which assess the higher motivational adequacy of using a complete robot in these subjects and also the real fulfillment of the other requirements along several academic years.[49]

I. Raspberry Pi

The Raspberry Pi Single Board Computer (SBC) family has gained popularity in diverse areas, while education remains to be the fundamental driver behind the design. Low cost kits are provided, specifically for education, by the Raspberry Pi Foundation in conjunction with Google. The tutorial will

consist of examples of projects and code samples that can be quickly adapted for various learning situations. The Rasp- berry Pi family and Arduino Family can also be used for education.The Raspberry Pi's cost effectiveness and diverse utility have led to diverse educational and other uses in both developing and developed nations [51] Platforms like Arduino [47], [48], Lego Mindstorms NXT [49], [50], or Raspberry Pi [51], and programming environments such as Logo [52],[53] or Scratch [13], [54] are some of the most popular tools. These platforms and programming environments have, as their leitmotif, to provide simple, intuitive, and easy-to-use tools. The so-called maker movement or making,Giannakos2017 is a new concept that has been recently coined to describe those activities that put the focus on the process of learning by constructing. Despite being a relatively new concept it is based in the well-known philosophy of learning by mak- ing,Papert1991 and founded in a constructivist (and construc- tionist) approach of the teaching and learning process. [44]. The use of robotics in primary and secondary education is an emerging trend mainly due to the vast amount of information available online, thanks to the open-source community that is generally supporting some of these projects, as well as the attention that STEM (Science, Technology, Engineering, and Mathematics) subjects are receiving as enablers for skills such as the computational thinking [56]. Higher education is not unaware of this trend but educational robotics is being employed as a complimentary tool [30] because students, at this level, should have already acquired these skills. In this sense, the use of robotics has been mainly employed to promote motivation and student engagement [30], [31], [45], [57]. They have also pedagogical benefits in learning [58], as they are based on problem-solving and active thinking [59] especially in the way lessons have been designed in the present research. This process of problem-solving activates cognitive skills like representing (the external representation of a problem is transformed into an internal mental model), planning, executing and evaluating [60], besides the develop- ment of metacognitive thinking skills needed to solve problems [61]. Additionally, their hardware features are very interesting, in terms of provided resources and constrained complexity, for the purpose of teaching Computer Architecture. However, to the best of the authors' knowledge, there is no similar approach that employs educational robotics, as a lab plat- form, for teaching computer architecture concepts. These two approaches therefore satisfy the Computer Engineering and Computer Science students demand of having attractive and real platforms for experiencing with Computer Architecture concepts [32]. This work is also

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

motivated by the lack of works quantitatively assessing the impact that both approaches have on learning Computer Architecture. To fill this gap, this paper presents two specific platforms representing these two approaches: a fit-for-purpose Arduino-based robot and the Nintendo-DS console. Appropriate hands-on sessions have been designed to cover the content of the course Computer Structure and learning outcomes have been gathered, analyzed and compared from the perspective of the student motivation, knowledge, and perception of the employed platform.

III. METHODOLOGY

A. The Syllabus of Computer Architecture

Computer Structure is one of the courses that comprise the Computer Architecture area. This is taught as part of the Computer Engineering Degree program in the University of Castilla-La Mancha, Spain. This course is intended to lead students into basic computer architecture concepts such as the memory system, the instruction set architecture or the input/output system. To this end, the course is organized in 6 European Credit Transfer System (ECTS) 4 of which are devoted to theory and 2 for labs. These 6 ECTS will be taught for 14 weeks, having 3 sessions per week of 90 minutes each. One of these three weekly sessions will take place in the lab and the remaining two will be devoted to theoretical concepts. The syllabus is organized in five lessons as stated in Table 1.

Lesson	Lecture sessions	Hands-on-sessions
1.Introduction	2 sesions	1 session
2.Memory	5 sesions	2 sessions
3.Machine and assembly language	7 sesions	2 sessions
4.Datapath and control unit	7 sesions	0 session
5.Input/Output	6 sesions	3 sessions

TABLE I
COURSE CONTENTS ORGANIZED BY SESSIONS.

B. Objectives of Computer Architecture Course

The learning objectives of this course have been certified by the National Agency for Quality Assessment and Evaluation and Accreditation from Spain and are listed below:

- Objective 1: To understand the principles of computer architecture.
- Objective 2: To know the organization of the CPU, identify the functional units, and explain their role in the execution of the instructions.
- Objective 3: To know the organization of the Input/Output subsystem and its interface with the CPU.
- Objective 4: To relate the evolution of CPU architecture and instruction sets. To identify the differences between the CISC and RISC philosophies..
- Objective 5: To program a computer at a low level.
- Objective 6: To learn, through practice, the structure and programming of a basic computer.

To address these objectives the course is organized in five lessons, as described in figure 3.1, addressed through lectures and hands-on experience.

IV. USED METHODOLOGY

The main aim of this study was to evaluate the convenience, from a pedagogical and motivational perspective, of using either robotics or game-consoles based platforms for teaching Computer Architecture. Provided the lack of studies evaluating

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

Lesson	Content description	Activities for Nintendo DS	Activities for Arduino robot
1. Introduction	What is a computer? Functional description: von Neumann architecture. Origin and historical evolution of computers. Characteristic parameters of the computers.	Representation of different backgrounds (basic ones, like the whole screen in blue or color grading) created in the program and stored in the stack (as dynamic variables).	Implementation of different "Hello, World!" versions. The string is provided in different ways (as text, static or dynamic variablter) and depending on that, it will be located in different memory regions.
2. Memory	Memory hierarchy.	The same background is represented using different graphic modes (framebuffer, rotoscale, and tiled). Depending on the employed mode, different memory types and regions are employed (tile memory, color palette memory, framebuffer, etc.).	Implementation of different programs. The binary code, no optimized by the compiler, will be explored through the remote debugger and guided by the ABI rules.
3. Machine and assembly language	Architecture and instruction set (ISA). Addressing modes. Application Binary Interface (ABI). RISC and CISC architectures.	Specific programming exercices are provided to the students to experience both the ISA of the ARM and the ABI rules established for that architecture. The debugger and the console emulator are employed to this end.	Specific programming exercises will be proposed to students to get familiar with both the ISA and ABI of the ARM architecture. Moreover, students will programming in assembly by exploiting the asm function to embed assembly instructions into a C programm.
4. Datapath and control unit	Description of the data path. Functions of the Control Unit. Execution phases of an instruction. Micro-instructions and control signals.	This lesson is covered with the activies undertaken during the previous lesson.	
5. Input/Output	Input-output modules. Input-output modes: pooled, interruptions and direct memory access (DMA).	Implementation of a game in which different input/output strategies (pooled, interruptions, and DMA) are considered.	Implementation of a line-follower program in which different pheripheral devices are involved.

Fig. 1. Lesson contents amd proposed activities for each lab platform

and comparing these two strategies, this paper pursues the following specific research goals:

- To compare both strategies in terms of student motivation, acquired knowledge, and perceived usefulness.
- To identify the pedagogical strengths and weaknesses of both approaches.

A. SAMPLE DESCRIPTION

The sample has been selected by intentional non- probabilistic sampling. First-year students of the Degree in Computer Engineering of the University of Castilla-La Man- cha, Spain compose the sample. It is divided into two cohorts (1 and 2) both for the same academic year (2016/2017). Each of them uses a different lab platform for the development of hands-on experiences, working ingroups of 2 to 4 people. Cohort 1 (using Arduino-based robot) consists of 96 subjects (82 males and 14 females; mean age: 19.20 years, SD D 1.63 years) and the 2 cohort (using Nintendo DS console) of 75 subjects (67 males and 8 females; mean age: 20.25 years, SD D 1.69 years).

B. STUDY DESIGN

The methodology used is based on the mixed paradigm, since the research has used qualitative and quantitative instru- ments to collect data. Two of the instruments were applied both at the beginning and at the end of the course (pre-test and post-test) and the platform perception instruments (Arduino and Nintendo DS) and the inquiry instrument that were only applied at the end of the course.

C. INSTRUMENTS

The data collection process has been automated using forms. One form has been created for each of the instruments that comprise the study. The Microsoft Forms platform has been employed for being this platform integrated into the corporative network. This has simplified the process of student identification as well as data collection and processing stage. Students were informed, before accessing the question- naires, about the aim of the study. Only the professors directly involved in the study have has access to personal data (name, gender, age, etc.) which have been anonymized for their statistical analysis by assigning an ID number to each subject.

Students were also told that the anonymization process would take place prior to the statistical analysis of the data. This was intended to minimize bias in the responses of students who believed that the outcome of their responses might affect their final grade. Additionally, students were assured before the start of a questionnaire that none of the answers provided would affect their final grade. The ordinal nature of the variables in the Likert scale instruments determined the use of polychoral correlation matrix for validation purposes. Thus, the reliability of the Likert scales instruments have been tested through the ordinal Cronbach's alpha. All the Likert scale instruments were also tested in terms of content and construct validity. Motivation and acquired knowledge have been considered as it seems that the first one has an impact in the second [63], especially when computer technologies are involved [64], [65].

1. *INSTRUMENT 1: STUDENT MOTIVATION:* The work in [48] analyses the convenience of state-of-the art instru- ments for measuring the students' attitude towards intro- ductory programming. The same conclusions apply to our study since learning low-level programming is one of the main competences of the Computer Structure course. We have therefore put the focus on low-level programming. Among the considered instruments the TAM model [66] is considered to most appropriate one because, apart from being successfully used in [48], it was originally envisioned to estimate the acceptance of innovative information systems. Similarly, to the work in [48] a translation was conducted precising low-level programming when a reference to programming was made.

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

The main constructs of the TAM model are:

- The perception of usefulness,
- The perception of ease of use and the behavioural inten- tion to use.

This model was therefore employed to inspired this instrument that measures the student motivation towards the low-level programming skill about to be acquired during the course.

The questionnaire, listed in Table 3.2 is comprised of 10 questions which use a Likert scale with 5 possible answers, as known: Totally agree (5),agree (4), neither agree nor disagree (3), disagree (2), totally disagree (1).Despite being based on a validated questionnaire [48], the validity and reliability param- eters have been calculated for our study. The content validity was tested by a panel often experts on Computer Architecture, and they agreed that the items were suitable for the research, well written and easy to understand. The construct validity was tested by an Exploratory Factor Analysis (EFA). The Kaiser- Meyer-Olkin sample suitability test (0.758) and the Bartlett's sphericity test (p¡0.01) confirmed the pertinence of an EFA. The EFA revealed a structure in which items are grouped into three factors, according to the theoretical construct considered for its development: Factor 1: the perception of usefulness (items1, 2, 3, 4), Factor 2: the perception of ease (items 5, 6, 7), and Factor 3: the behavioral intention to use (items 8, 9, 10). The lreliability of the scale was tested by the ordinal Chronbach's alpha. The results indicated a very good reliability of the scale [67] as a whole (0.78) and by factors (0.77 for factor 1, 0.72 for factor 2 and 0.76 for factor 3).

	Option a	Option b	Option c	Option d
1. What kind of electronic circuits make up the basic structure of a digital computer?	Analogue circuits	Combined logic circuits	Sequential logic circuits	**Combined and sequential logic circuits**
2. On which part of a computer are the calculations performed?	**In the Central Processing Unit (CPU)**	In the memory	In the registers	In the CPU and memory
3. Which are the basic elements of a computer?	The processor and the memory	**CPU, memory and input/output**	The ALU, the registers and the communication buses	The peripheral devices and the processor
4. What characterizes a micro-controller?	The micro-controller integrates in a single chip a CPU, volatile and non-volatile memory and specific input and output peripherals	Micro-controllers are usually employed in specific applications where a single program is run	Micro-controllers usually have a reduced cost and consumption	**All the answers are correct**
5. The size of a memory is determined by:	**The word size and word number**	The number of available addresses	The number of available memory chips	The number of memory modules that comprise it
6. A memory of 1KiB:	It has 1024 memory positions of 1 byte each position	It has $2^{*}10$ memory positions of 1 byte each	It can be addressed using 10 address lines	**All the answers are correct**
7. Comparing a RAM memory and a Flash memory:	Both memories are volatile	RAM memory is non-volatile and Flash memory is volatile	Both memories are non-volatile	**RAM memory is volatile and Flash memory is non-volatile**
8. Regarding the time required to access the memory:	They are always very fast and do not affect the execution of a program	**It depends on the position of each memory type in the memory hierarchy.**	They do not have any influence on the execution time of an instruction	They are not related to the cost and size of the memory
9. Regarding the computer buses:	These are the external wiring of the computer	**These are the lines of communication between the different functional units that make up the computer**	They are only used to communicate with peripherals	None of the above answers is correct
10. The instructions that a computer executes:	They are stored in the main memory	They always have the same duration in number of clock cycles	There can be different types: arithmetic, logic, jump, data transfer,...	**The first and third options are correct**

Preceived Usefulness of programming
Item 1.Knowing low-level programming will help me find a job Item 2.It will be easier to finish my studies if i know low level programming Item 3.During my studies it will be useful for me to know low level programming Item4.Knowing low-level programming will be useful for my work.
Preceived Ease of programming
Item 5.It is easy for me to learn low level programming. Item 6.It is easy to make the low level program do what i want to. Item 7.Low level programming is easy
Intention to program
Item 8.I intended to use low level programming during my studies. Item 9.In my job i will code low level programs that will be helpful for me Item 10.Low level programming will form part of my profession

TABLE II

QUESTIONNAIRE EMPLOYED TO STUDY THE STUDENT MOTIVATION

2. *INSTRUMENT 2: ACQUIRED KNOWLEDGE:* This in- strument consists in 10 multiple-choice questions (with four answers) designed ad-hoc for the present study as shown in figure 3.2. This instrument was addressed to measure the achievement of the learning objectives pursued by the course, listed in Section III: objective 1 (questions 1, 2, 3, 4), objective 2 (questions 2 and 3), objective 3 (questions 3 and 9), objective 4 (questions 2 and 10), objective 5 (questions 5, 6, 7), objective 6 (8, 9, 10). The validity of this instrument was carried out by a panel of 10 experts (8 professors of the Computer Architecture and Technology area, 1 undergraduate student, and 1 postgraduate student). Following the first revision, the proposed amendments were undertook and a second version of the questionnaire was resubmitted to this panel of expert who finally consider it valid. This version was applied through a pilot test of 10 students of the Degree in

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

Computer Engineer- ing, proving that there were no doubts in the understanding of any of the items. Due to the dychotomical character of these variables, the reliability of the instrument was calculated using the ordinal Chronbach's alpha. The result (0.67) indicated a good internal consistency of the instrument and makes it suitable for the research. Additionally, this questionnaire also collects information about students' previous experience. To this end four questionswill formulated: 1) Do you have previous experience in programming microcontrollers?; 2) If so, which ones?; 3) Do you have previous experience in computer programming?; 4) If so, what languages have you used?

3. *INSTRUMENT 3: STUDENT PERCEPTION OF THE ARDUINO-BASED ROBOT PLATFORM:* Students' percep- tion of the computing platforms used in teaching provides important information for assessing the teaching practice [68]. This instrument is intended to evaluate the student percep- tion of the Arduino-based robot platform. The questionnaire proposed in [48] inspires the current instrument to evaluate the use of the Arduino board in the context of low-level programming. The instrument is comprised of 10 items which use a Likert scale with 5 possible answers, as known: Totally agree (5),agree (4), neither agree nor disagree (3), disagree (2), totally disagree (1). This questionnaire, listed in Table 3.3, has been inspired in [48].

Fig. 2. Questionnaire employed to measure the learning outcomes with correct answers in bold.

Preceived Usefulness of the Arduino based robot
Item 1.When low level programming for Arduino based robot I learn more Item 2.With the Arduino based robot I learn better low level programming Item 3.The arduino based robots helps me understand low level programming Item 4.I learn more qucikly when working with the Ardion based robot
Preceived Ease of use of the Arduino besd robot
Item 5.It is easy to work with Arduino bsed robot. Item 6.The Arduino base robot is easy to use. Item 7.It is easy to low level program the Arduino based platform
Enjoyment when using the Arduino based robot
Item 8.Lab sessions with the Arduino based robot are more entertaining Item 9.I have fun working with the Arduino based robot Item 10.Working with the Arduino based is more enjoyable

TABLE III

QUESTIONNAIRE EMPLOYED TO STUDY THE STUDENT PERCEPTION OF THE ARDUINO-BASED ROBOT PLATFORM.

This instrument is based in a previous one, already vali- dated, but it was necessary to validate it in the context of the present research. Firstly, the content validity was tested by a panel of experts on Computers Architecture, who agreed in the suitability of the items for the research purposes, written expression and ease of understanding. Secondly, the construct validity was tested by an EFA. The pertinence of carrying out an EFA was confirmed by the Kaiser-Meyer-Olkin sample suitability test (0.874) and the Bartlett sphericity test (p¡0.01) results. The EFA revealed a construct structure of three factors, according to the theoretical construct considered for its development: Factor 1: Perceived usefulness of the Arduino board (items 1, 2, 3, 4), Factor 2: Perceived Ease of
Use of the Arduino board (items 5, 6, 7), and Factor 3 (items 8, 9, 10). The reliability was tested by the ordinal Chronbach's alpha, due to the ordinal nature of the variables. The results indicated an excellent reliability of the scale [67] as a whole (0.9) and by factors (0.88 for factor 1, 0.85 for factor 2 and 0.90 for factor 3).

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

4. *INSTRUMENT 4: STUDENT PERCEPTION OF THE NINTENDO-DS PLATFORM:* This instrument is intended to

evaluate the student perception of the Nintendo-DS platform. The questionnaire proposed in [48] has been adapted to refer to Nintendo DS instead of the Arduino board, as in its original version. This questionnaire is listed in Table 3.4. The instrument is comprised of 10 items which use a Likertscale with 5 possible answers, as known: Totally agree (5),agree (4), neither agree nor disagree (3), disagree (2), totally disagree (1).

Preceived Usefulness of programming
Item 1.When low level programming for Nintendo DS I learn more Item 2.With Nintendo DS I learn better low level programming Item 3.Nintendo DS helps me understand low level programming Item 4.I learn more quickly when working with Nintendo
Preceived Ease of programming
Item 5.It is easy for me to learn low level programming. Item 6.It is easy to make the low level program do what i want to. Item 7.Low level programming is easy
Intention to program
Item 8.Lab sessions with Nintendo DS are more entertaining. Item 9.I have fun working with Nintendo DS Item 10.Working with Nintendo DS is more enjoyable

TABLE IV
QUESTIONNAIRE EMPLOYED TO STUDY THE STUDENT PERCEPTION OF THE NINTENDO DS PLATFORM

The instrument was validated in terms of validity and reliability for this research case study. The content validity was tested, as in the previous cases, by a panel of ten experts on computers education. They agreed that all the items were pertinent and well written, easy to understand and to reply. The construct validity was tested with an EFA. The Kaiser-Meyer- Olkin sample suitability test (0.841) and the Bartlett sphericity test (p¡0.01) confirmed the pertinence of carrying out an EFA. The results confirmed that the items were grouped in three factors, according to the theoretical construct considered in the design: Factor 1: Perceived usefulness of the Nintendo DS platform (items 1, 2,3, 4), Factor 2: Perceived Ease of Use of the Nintendo DS platform (items 5, 6, 7),and Factor 3: Perceived Enjoyment when using the Nintendo DS platform (items 8, 9, 10). The reliability of the scale was confrmed by the ordinal Chronbach's alpha, due to the ordinal nature of the variables. The results indicated a very good reliability of the scale [67] as a whole (0.81) and by

5. *INSTRUMENT 5: QUALITATIVE EVALUATION OF THE EMPLOYED PLATFORM:* This instrument consists of the following open question: "Briefly tell us your impressions, assessment, or suggestions for the future about the course and the hands-on experience with the Arduino-based robot/Nintendo DS". This instrument was applied to a panel of 10 experts composed of undergraduate and postgraduate students and professors from the area of Computer Architec- ture and Technology. The instrument was applied in a pilot test with 10 students.

6. *DATA ANALYSIS PROCEDURE:* The SPSS v. 23, R and Factor software were used for data analysis. The Kolmogorov-Smirnov test (as the sample is larger than 50 subjects) was applied to the variables corresponding to the different instruments based on the Likert scale. Since these variables do not meet the normality criterion (p¡0.05), non-parametric statistics have been used [69]: Mann Withney's U test for independent samples and Wilcoxon test for related samples. The analysis was complemented with the effect size with r, due to the nonnormality of the variables, but also the mean and standard deviation as indicative descriptors.

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

V. RESULTS AND DISCUSSIONS

A. RESULTS:

1. *STUDENT MOTIVATION:* When the inter-group analy- sis is done using the Wilcoxon test for related samples (see figure 4.1), in the Arduino group it turns out that in 4 of the 10 items there are statistically significant differences between the pre and post test (items 3, 5, 9, 10). In these four items they score more in the pre test than in the post-test (they are demotivated in these four aspects). However, in the Nintendo- DS group, statistically significant differences appear in 3 of the 10 items (1, 2, 6), and in all three, this group scores more in the post test than in the pre test.

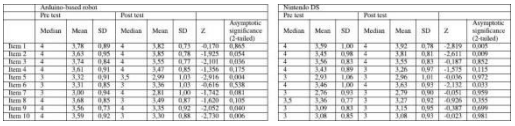

Fig. 3. Descriptive statistics and Wilcoxon Test results for motivation test.

	Pre test		Post-test	
	Mann-Whitney's U	Asymptotic significance (2-tailed)	*Mann-Whitney's U*	Asymptotic significance (2-tailed)
Item 1. Knowing low-level programming will help me find a job	3181.500	.191	3304.000	.296
Item 2. It will be easier to finish my studies if I know low-level programming	3144.500	.149	3412.000	.812
Item 3. During my studies it will be useful for me to know low-level programming	3145.000	.141	3586.000	.963
Item 4. Knowing low-level programming will be useful for my work	3156.000	.215	3164.500	.196
Item 5. It is easy for me to learn low-level programming	2855.000	.020	3539.500	.844
Item 6. It is easy to make the low-level program do what I want to	3128.500	.192	3132.500	.120
Item 7. Low-level programming is easy	3034.500	.103	3522.000	.799
Item 8. I intend to use low-level programming during my studies	2791.500	.018	3016.500	.109
Item 9. In my job I will code low-level programs that will be helpful for me	2448.000	.000	3213.000	.201
Item 10. Low-level programming will form part of my profession	2465.500	.000	3193.000	.177

Fig. 4. Mann Withney's U Test results for motivation test.

The results listed in figure 4.2 of Mann Withney's U test (comparing independent samples) reveal that in the pre test both cohorts had statistically non-significant differences in scores on 6 of the 10 items (items 1, 2, 3, 4, 6, 7). In the first four and 7, the students in the Arduino cohort showed greater motivation, and in 6, less than those in Nintendo-DS cohort. In the post test, no statistically significant differences were found in any of the items of this instrument.

The analysis of the results measuring the student motivation towards the use of low-level programming shows a decrease in the perception of the usefulness for those using the Arduino- based robot. This decrease affects both to the use of low-level programming during their studies and professional career. Par- ticularly interesting is the decrease experienced in item 5, that regards with the easiness of learning low- level programming. This same item does not suffer significant differences for the Nintendo DS group, although the post test score was lower,

as an average score, than that obtained for the Arduino group (2.99 for the Arduino group versus 2.96 for the Nintendo DS group). For items 9 and 10, that regards with the perceived utility of low-level programming for the future professional career, a similar situation occurs as described for item 5: from a pre test with higher values in the Arduino group there is a statistically significant decrease despite the fact that the final scores are still higher(on average) than those obtained by the Nintendo DS group, in which there are no statistically significant changes between the pre test and post test. Note that r gets low values (see figure 4.3) for both Nintendo DS and Arduino and therefore the effect of the intervention is not relevant in terms of motivation.

3) STUDENT PERCEPTION OF ARDUINO PLATFORM: This instrument was applied to the Arduino group at the end of the intervention. A descriptive statistical analysis has therefore been carried out. The results are shown in figure 4.5. The item that obtained the lowest score regards the perception of the utility of knowing low-level programming during their studies (item 3). On the contrary, the one that obtained the highest score regards the perceived utility of knowing low- level programming for their future jobs (item 9).

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
Median	4	3	4	3	3	4	3	4	4	3
Mean	3,4896	3,2500	3,4362	3,2813	3,4000	3,4896	3,3723	3,5104	3,6000	3,3229
SD	1,0760	1,0761	0,9898	1,0926	1,1051	1,0563	0,9944	1,2896	1,1522	1,1002

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

	Arduino-based robot	Nintendo DS
	Effect size r	Effect size r
Item 1. Knowing low-level programming will help me find a job	-0.02	-0,33
Item 2. It will be easier to finish my studies if I know low-level programming	-0,20	-0,30
Item 3. During my studies it will be useful for me to know low-level programming	-0,21	-0,02
Item 4. Knowing low-level programming will be useful for my work	-0,14	-0,18
Item 5. It is easy for me to learn low-level programming	-0,30	0,00
Item 6. It is easy to make the low-level program do what I want to	-0,06	-0,25
Item 7. Low-level programming is easy	-0,18	-0,01
Item 8. I intend to use low-level programming during my studies	-0,17	-0,11
Item 9. In my job I will code low-level programs that will be helpful for me	-0,21	-0,04
Item 10. Low-level programming will form part of my profession	-0,28	0,00

Fig. 5. Results of the effect size using the r for motivation

2) ACQUIRED KNOWLEDGE: The results of Mann With- ney's U test show that there are no satistically differences between the two cohorts in either the pre test(p=0.153) or the post test(p=0.319).

	Pre test			Post test		
	Median	Mean	SD	Median	Mean	SD
Arduino-based robot	7,0000	6,84375	1,8025	9,0000	8,0625	1,6144
Nintendo DS	7,0000	7,16	1,830	8,0000	7,8533	1,5997

Fig. 6. Learning outcomes(Scores in the range of 0 to 10).

Figure 4.4 summaraizes the learing outcomes comparing the score in the score in the pre and post test,withscores in the range of 0 to 10.The differences found between grouos are not satistically significant,in the pre test,the Nintendo DS group scores slightly better than Arduino group.In the post test,it is the Arduino group scores better than the Nintendo DS group.A comparison has been also carried out considering independent questions .Results of the Chi-Square test show that initially both cohorts showed non-significant differences in 6 of the 10 questions that composed the questionnaire, and after the interventions, non-significant differences were found in all the items. The questions with lower scores after the interventions were those related to micro controllers, memory, and instructions. It is worth noting that students' previous experience in both cohorts is very similar. 28% of the students in the Arduino-based robot group have previous experience in programming microcontrollers (all of them reported hav- ing used Arduino or Arduino-compatible board) and 19% have previous experience with the C programming language, whereas for the Nintendo DS cohort, 22% have previous experience with microcontrollers like Arduino or similar and 17% have previously programmed in C.

Fig. 7. Statistical results for the student perception of the Arduino platform

4. *STUDENT PERCEPTION OF NINTENDO-DS LABS:* This instrument was applied to the Nintendo DS group at the end of the intervention. A descriptive statistical analysis has therefore been carried out. The results are shown in Table14. The item that obtained the lowest score regards the perceived simplicity of learning low-level programming (item 5). The one that obtained the highest score regards the perceived utility of knowing low-level programming for their future jobs (item 9).

	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	Item 10
Median	4	4	4	3	3	3	3	4	4	3
Mean	3,4133	3,3200	3,3200	3,2133	2,8800	3,0133	3,0000	3,4933	3,5467	3,2667
SD	0,9739	0,9028	0,8724	0,8589	0,9147	0,9078	0,9153	0,8601	0,8268	0,8110

Fig. 8. Results of the student perception of the Nintendo-DS platform.

	Arduino		NDS	
	N (Total number of answers)	Percentage (%) with respect to the total number (N=96) of subjects in that cohort	N (Total number of answers)	Percentage (%) with respect to the total number (N=75) of subjects in that cohort
Issue 1. Issues regarding the lab platform	11	11,46	1	1,33
Issue 2. Issues regarding the platform programming	1	1,04	7	9,33
Issue 3. Problems understanding the course content	3	3,13	6	8
Issue 4. Regretting the replacement of the NDS platform	0	0	26	34,67
Issue 5. Issues leading to student frustration	20	20,83	6	8
Issue 6. Gap or lack of connection between the theory and lab	2	2,08	4	5,33
Issue 7. More time required for the final project	30	31,25	0	0

Fig. 9. Results of categorized answers

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

5. *QUALITATIVE EVALUATION OF THE EMPLOYED PLATFORM:* This tool was applied to both groups. Out of the 171 answers, 54 were discarded because they were left blank or did not know the answer. 67 answers of the cohort 1 (Arduino group) and 50 of the cohort 2 (Nintendo-DS group) were analyzed. Table15 shows the results of the categorized answers, organized by cohorts.

B. DISCUSSIONS

This study compares two approaches to teach Computer Ar- chitecture: Arduino, a robot-based approach widely used from primary to higher education, and Nintendo DS, a game-based one. The results show that, despite the a priori advantages of Arduino as a more engageable tool [30] and pedagogical affordances related to easiness of low-level programming and machine and assembly language, the differences among both approaches at the end of the interventions are not conclusive in terms of motivation and learning outcomes. These results are discussed in this section regarding the different types of learn- ing activities, pedagogical affordances and constraints, and attitudes developed by the students in both didactic sequences. Students' motivation towards programming was measured as previous literature revealed it has an impact on motivation to learn and acquired knowledge [63], especially when robots are used for teaching [30]. It is stunning how motivation in the Arduino group decreases after the intervention, on the contrary to what happens with the Nintendo DS group, even when the Arduino group was expected to be more motivated according to previous literature [30], [63] given the characteristics of the tool that facilitates programming and provides more interaction with hardware (calibration of sensors, servo, robot pieces, etc.). The Arduino group was indeed initially more motivated than the Nintendo DS one, probably because the use of robots is usually more motivating to learn programming [63]. But at the end, differences in motivation between both groups were not significant. Particularly, the Nintendo DS cohort increased their motivation towards low-level programming, and they perceive that after the intervention it was easier than initially expected. This can be explained by the fact that the game-development process has a positive impact on motivation and harder work when solving problems [13], [70]. Further explanations might be found in different aspects. The work in [71] points out that the use of robots does not necessarily increases the relevance, confidence or satisfaction when programming. The results of our study support these ones, as other aspects play a role in the teaching and learning process that should not be overlooked. Thus, regarding the differences in the low-level programming teaching approaches, the Nintendo DS group resorts to the use of different graphic representation modes (frame-buffer, rotoscale, and tiles), while Arduino group employs simple programs to be analyzed with a remote debugger. The use of graphic modes has an inherent complexity, especially for those facing for the first time computer graphics. On the other hand, first sessions of the Arduino group spin around the use of simple programs which will softly guide them through the different lesson contents. This implies that the robot-approach entails less complexity in terms of low-level programming competences than that for console-approach, taking into account that the Nintendo DS activities need a basic understanding of how graphics and memory work. This different level of complexity in the first sessions is a pedagogical advantage of Arduino over Nintendo DS. However, the fact that first activities of the Arduino group are carried out on the Arduino board (without involving the robot until lesson 5), may lead students to perceive these programming activities as decoupled from the rest of the sessions (implementation of the robot functionality). This is a constraint that directly impact on the learning process for the Memory- lesson concepts. The Nintendo DS cohort performs better that the Arduino one on questions regarding the memory system design. All these issues have determined that the Nintendo DS group increases it motivation towards the usefulness of low-level programming, counterbalancing initial motivation differences among groups. The different characteristics of Nintendo DS and Arduino that determine the learning activities can also explain the differences on perception of both platforms, after the lessons, regarding the easiness to program. Arduino cohort has a better perception of the platform than the Nintendo DS one. Arduino, through its Arduino IDE (programming environment), provides a simple and easy-to-use environment to program the board, what has turned it into a widely used platform from primary to higher education. This is not the case for the Nintendo DS having to resort to command- line tools which might overwhelmed students unfamiliar with the GNU/Linux like systems. This result is of interest as it points out to Arduino, over Nintendo DS, for those students unfamiliar with GNU/Linux systems. This requires further research as this question was not initially part of the research and there were no statistically significant differences between the two cohorts in terms of acquired knowledge either for the pre-test or the post-test.

The differences found between both approaches can also be explained by the way teaching activities have been organized. Arduino activities needed from robots, and were carried out during the lab sessions, in groups. Nintendo DS activities based on simulators and real consoles, enable students to work at home, in an individual manner, preventing some teamwork issues from occurring, even when both approaches were based on teamwork. The higher load of teamwork is a strong point of the Arduino learning activities, as it helps students to develop social skills and work

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

in an environment that is closer to their future work environment and very valuable skills in Computer Science [72]. However, teamwork has inherent difficulties to deal with, as those derived from the conflicts that may emerge among peers, as the answers of the students to the open question show. This result is especially valuable because, even when previous literature has focused on the benefits of developing teamwork skills of game-based approaches [73], for robotics further research was claimed [74], [75], and this work contributes to give light to this issue. Nevertheless, aspects specifically derived from teamwork have not been specifically assessed, further research is needed to evaluate conflicts and their reasons, and the way they assume responsibilities, and how these issues affect the results of the interventions. Nonetheless, the work in [76] recommends working in groups of two people rather than in larger groups because in this way the benefits of pair programming are enhanced and the exchange of leadership roles is facilitated. The design of both teaching approaches was also different in terms of the introduction of the console and the robot. The Nintendo DS group combine both the console and its emulator counterpart from the first session, what enables a seamless transit to the real platform for less experienced students [35]. On the other hand, in the Arduino group, the robot was not introduced until the Input/output lesson (last lesson of the

syllabus). Initially, it was unclear whether this later use of the platform could have an impact on acquired knowledge. The Nintendo DS group started from lower scores than the Arduino one in the knowledge questionnaire (pre test), but in the post test there were no statistically significant differences among both cohorts, being scores around 8 out of 10 in both cases. These high scores support the evidences found by other authors [25], [30], [32], [37], [44], [48], [74], [77]about the convenience of both strategies, robotics and game- based approaches, for teaching Computer Architecture. That implies that both approaches are equally suitable in terms of learning outcomes, and valid to teach Computer Architecture with similar learning results. Besides, these learning out- comes seem not to be correlated to motivation, in contrast to previous research that suggest that motivation has an impact on acquired knowledge [63], especially when computer technologies are involved [64], [65]. In this case, both approaches seem to homogenize the final perception, independently of the em- ployed platform. It is also outstanding how more than a third of the Nintendo DS cohort regrets the possibility of new students not having the opportunity of programming a game for the console, what aim at the acceptance that the platform has despite its obsolescence. Despite the lack of statistically significant differences both in the acquired knowledge and at the level of the single questions comprising the questionnaire, it can be noticed that there are three questions (questions number 4, 5 and 10) whose results are under the 75.

VI. CONCLUSION

The results of this study show the suitability of the robot- based (Arduino) and the game-based (Nintendo DS) approaches to teach Computer Architecture in terms of learning outcomes. Nevertheless, the different affordances and constraints had also an impact on results due to their differences from a pedagogical perspective. Arduino is revealed as a powerful approach to teach low- level programming to poorly experienced students due to the facilities that the tool presents. It is also useful to teach input/output concepts since it gives students the opportunities to program different behavior in the robot (line follower, obstacle avoidance, light follower, etc.) and deal with external conditions that affect the robot sensors or other peripheral devices. On the contrary, Nintendo DS is especially interesting because of the different graphic modes it support and the different peripheral devices (keys, touch screen, timers, etc.). The design and implementation of a game, including the representation of graphics and its logic involve dealing with different type of memories and input/output techniques. Students can experience the three types of input/output techniques: pooled to manage the console keys, interrupt to use timers, and DMA to display different backgrounds (copying backgrounds to the video memory). However, besides these pedagogical approaches, teachers must consider other constraints and affordances in their teaching, as the results of this research reveal. The impact that external conditions have on the robot sensors means that it is necessary to calibrate them whenever conditions change. This process is time-consuming, especially in the beginning when students are not familiar with the process. It is therefore desirable to make robots available in addition to the lab hours to minimize frustration that might demotivating them. This is not necessary for game-based approaches, as a simulator is available. This enables students to work individually at home. In this sense, teamwork should not be overlooked, as even when it resemble a real working context, it may trigger conflicts causing delays that could eventually lead to increase the time needed to complete the proposed tasks. It seems also advisable to provide students with the opportunity to work with robots platform from the beginning of the course, avoiding this way the perception of lab sessions being decoupled from input/output lesson. This might be responsible for a low perception of the usefulness of low-level programming. Regarding game-based approaches, teachers should pay attention at initial knowledge requirements on low-level programming to prevent students from stucking from the beginning.

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

A. Future Scope

Despite the obsolescence of Nintendo DS, the results of this work are interesting for teachers aimed to design teaching plans based on other game platforms, and of course, to design their robot-based lessons, even using other platforms than Arduino. Further research in other contexts as in vocational

REFERENCES

[1]J. Impagliazzo, S. Conry, J. Hughes, J. Junlin, A. McGettrick, E. Durant,H. Lam, R. Reese, and L. Herger, Curriculum Guidelines for Undergrad- uate Degree Programs in Computer Engineering. New York, NY, USA: Association for Computing Machinery, 2016.

[2]L. Porter, S. Garcia, H.-W. Tseng, and D. Zingaro, "Evaluating student understanding of core concepts in computer architecture," in Proc. 18th ACM Conf. Innov. Technol. Comput. Sci. Edu. ITiCSE, New York, NY, USA, 2013, pp. 279–284.

[3]J. M. Lopez-Guede, I. Soto, A. M. F. D. Leceta, and J. M. Larra n˜aga, "Educational innovation in the computer architecture area," Procedia Social Behav. Sci., vol. 186, pp. 388–394, May 2015.

[4]J. C. Cuevas-Martinez, A. J. Yuste-Delgado, J. M. Perez-Lorenzo, and A. Trivino-Cabrera, "Jump to the next level: A four-year gamification experiment in information technology engineering," IEEE Access, vol. 7, pp. 118125–118134, 2019.

[5]E. Larraza-Mendiluze and N. Garay-Vitoria, "Approaches and tools used to teach the computer input/output subsystem: A survey," IEEE Trans. Educ., vol. 58, no. 1, pp. 1–6, Feb. 2015.

[6]B. Nikolic, Z. Radivojevic, J. Djordjevic, and V. Milutinovic, "A sur- vey and evaluation of simulators suitable for teaching courses in com- puter architecture and organization," IEEE Trans. Educ., vol. 52, no. 4, pp. 449–458, Nov. 2009.

[7]J. R. Brinson, "Learning outcome achievement in non-traditional (virtual and remote) versus traditional (hands-on) laboratories: A review of the empirical research," Comput. Edu., vol. 87, pp. 218–237, Sep. 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0360131515300087

[8]V. J. Shute, C. Sun, and J. Asbell-Clarke, "Demystifying computa- tional thinking," Educ. Res. Rev., vol. 22, pp. 142–158, Nov. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S1747938X17300350

[9]H.-T. Hung, J. C. Yang, G.-J. Hwang, H.-C. Chu, and C.-C. Wang, "A scoping review of research on digital game-based language learning," Comput. Edu., vol. 126, pp. 89–104, Nov. 2018, doi: 10.1016/j.compedu. 2018.07.001.

[10]J. M. Wing, "Computational thinking," Commun. ACM, vol. 49, no. 3, pp. 33–35, 2006, doi: 10.1145/1118178.1118215.

[11]S. Grover and R. Pea, "Computational thinking in K–12: A review of the state of the field," Educ. Researcher, vol. 42, no. 1, pp. 38–43, Jan. 2013.

[12]K. Howland and J. Good, "Learning to communicate computationally with flip: A bi-modal programming language for game creation," Com- put. Edu., vol. 80, pp. 224–240, Jan. 2015.

[13]D. Topalli and N. E. Cagiltay, "Improving programming skills in engi- neering education through problem-based game projects with scratch," Comput. Edu., vol. 120, pp. 64–74, May 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131518300113

[14]J. Kramer, "Is abstraction the key to computing?" Commun. ACM, vol.50, no. 4, pp. 36–42, Apr. 2007.

[15]J. M. Wing, "Computational thinking and thinking about computing," Philos. Trans. Roy. Soc. London A, Math. Phys. Sci., vol. 366, no. 1881, pp. 3717–3725, 2008.

[16]J. J. Lu and G. H. Fletcher, "Thinking about computational thinking," in Proc. ACM SIGCSE Bull., vol. 41, no. 1, 2009, pp. 260–264.

[17]L. Moreno, C. Gonzalez, I. Castilla, E. Gonzalez, and J. Sigut, "Applying a constructivist and collaborative methodological approach in engineer- ing education," Comput. Edu., vol. 49, no. 3, pp. 891–915, Nov. 2007.

[18]L. S. Vygotsky, Mind in Society: The Development of Higher Psychologi- cal Processes. Cambridge, MA, USA: Harvard Univ. Press, 1978.

[19]J. G. Greeno, A. M. Collins, and L. B. Resnick, "Cognition and learning," Handbook Educ. Psychol., vol. 77, pp. 15–46, Jan. 1996.

[20]S. Papert and I. Harel, "Situating constructionism," Constructionism, vol.36, no. 2, pp. 1–11, 1991.

[21] U¨ . C¸ akirog˘lu, M. Yildiz, E. Mazlum, E. T. Gu¨ntepe, and S¸ . Aydin, "Explor- ing collaboration in learning by design via weblogs," J. Comput. Higher Edu., vol. 29, no. 2, pp. 309–330, Aug. 2017.

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

[22]J. Leonard, M. Mitchell, J. Barnes-Johnson, A. Unertl, J. Outka-Hill,R. Robinson, and C. Hester-Croff, "Preparing teachers to engage rural students in computational thinking through robotics, game design, and cul- turally responsive teaching," J. Teacher Edu., vol. 69, no. 4, pp. 386–407, Sep. 2018.
[23]M. Carbonaro, M. Cutumisu, H. Duff, S. Gillis, C. Onuczko, J. Siegel,J. Schaeffer, A. Schumacher, D. Szafron, and K. Waugh, "Interactive story authoring: A viable form of creative expression for the classroom," Comput. Edu., vol. 51, no. 2, pp. 687–707, Sep. 2008.
[24]G.-J. Hwang, C.-M. Hung, and N.-S. Chen, "Improving learning achieve- ments, motivations and problem-solving skills through a peer assessment- based game development approach," Educ. Technol. Res. Develop., vol. 62, no. 2, pp. 129–145, Apr. 2014, doi: 10.1007/s11423-013-9320-7.
[25]E. Larraza-Mendiluze, N. Garay-Vitoria, J. I. Martin, J. Muguerza,T. Ruiz-Vazquez, I. Soraluze, J. F. Lukas, and K. Santiago, "Game- Console-Based projects for learning the computer input/output subsys- tem," IEEE Trans. Educ., vol. 56, no. 4, pp. 453–458, Nov. 2013.
[26]M. J. Santofimia and F. Moya, "Nintendo DS: A pedagogical approach to teach computer architecture," in Proc. Int. Conf. Embedded Syst. Appl., (ESA), H. R. Arabnia and A. M. G. Solo, Eds. Las Vegas NV, USA: CSREA Press, Jul. 2009, pp. 269–273.
[27]D. Kumar, "Curriculum descant: Pedagogical dimensions of game play- ing," Intelligence, vol. 10, no. 1, pp. 9–10, Mar. 1999.
[28]S. Bergin and R. Reilly, "The influence of motivation and comfort-level on learning to program," in Proc. 17th Workshop Psychol. Program. Interest Group. Brighton, U.K.: University of Sussex, 2005, pp. 293–304.
[29]B. D. Coller and M. J. Scott, "Effectiveness of using a video game to teach a course in mechanical engineering," Comput. Edu., vol. 53, no. 3, pp. 900–912, Nov. 2009.
[30]N. Spola oˆr and F. B. V. Benitti, "Robotics applications grounded in learn- ing theories on tertiary education: A systematic review," Comput. Edu., vol. 112, pp. 97–107, Sep. 2017.
[31]D. Bazylev, A. Margun, K. Zimenko, A. Kremlev, and E. Ruku- jzha, "Participation in robotics competition as motivation for Learn- ing1," Procedia Social Behav. Sci., vol. 152, pp. 835–840, Oct. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S187704281405397X
[32]E. Brunvand,"Games as motivation in computer design courses: I/O is the key," in Proc. 42nd ACM Tech. Symp. Comput. Sci. Edu. SIGCSE, 2011, pp. 33–38.
[33]P. Tiernan, "Enhancing the learning experience of undergraduate tech- nology students with LabVIEW software," Comput. Edu., vol. 55, no. 4, pp. 1579–1588, 2010.
[34]C. A. Canizares and Z. T. Faur, "Advantages and disadvantages of using various computer tools in electrical engineering courses," IEEE Trans. Educ., vol. 40, no. 3, pp. 166–171, Aug. 1997.
[35]V. Potkonjak, M. Gardner, V. Callaghan, P. Mattila, C. Guetl, V.M. Petrovic´, and K. Jovanovic´, "Virtual laboratories for education in science, technology, and engineering: A review," Comput. Edu., vol. 95, pp. 309–327, Apr. 2016. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S0360131516300227
[36]A. Clements, "ARMs for the poor: Selecting a processor for teaching com- puter architecture," in Proc. IEEE Frontiers Edu. Conf. (FIE), Oct. 2010, pp. T3E–1–T3E–6.
[37]M. Carbonaro, D. Szafron, M. Cutumisu, and J. Schaeffer, "Computer- game construction: A gender-neutral attractor to computing sci- ence," Comput. Edu., vol. 55, no. 3, pp. 1098–1111, Nov. 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/ S0360131510001399
[38]A. Tlili, F. Essalmi, and M. Jemni, "A mobile educational game for teaching computer architecture," in Proc. IEEE 15th Int. Conf. Adv. Learn. Technol., Jul. 2015, pp. 161–163.
[39]S. Tabik and L. F. Romero, "16 cards to get into computer organiza- tion," Ensen˜anza y Aprendizaje de Ingenier´ıa de Computadores, vol. 2014, no. 4, pp. 5–14, 2014.
[40]S. Papavlasopoulou, M. N. Giannakos, and L. Jaccheri, "Empirical studies on the maker movement, a promising approach to learning: A litera- ture review," Entertainment Comput., vol. 18, pp. 57–78, Jan. 2017, doi: 10.1016/j.entcom.2016.09.002.
[41]G. Chen, J. Shen, L. Barth-Cohen, S. Jiang, X. Huang, and M. Eltoukhy, "Assessing elementary students' computational thinking in everyday rea- soning and robotics programming," Comput. Edu., vol. 109, pp. 162–175, Jun. 2017, doi: 10.1016/j.compedu.2017.03.001.
[42]M. U. Bers, L. Flannery, E. R. Kazakoff, and A. Sullivan, "Com- putational thinking and tinkering: Exploration of an early childhood robotics curriculum," Comput. Edu., vol. 72, pp. 145–157, Mar. 2014, doi: 10.1016/j.compedu.2013.10.020.

|| e-ISSN: 2319-8753, p-ISSN: 2320-6710| www.ijirset.com | Impact Factor: 7.569| || Volume 10, Special Issue 3, December 2021 ||

National Conference on Emerging Technologies [NCET' 2021]

Organized by

Royal College of Engineering and Technology, Thrissur, Kerala, India

[43]M. Ruzzenente, M. Koo, K. Nielsen, L. Grespan, and P. Fiorini, "A review of robotics kits for tertiary education," in Proc. Int. Workshop Teach. Robot. Teach. Robot., Integrating Robot. School Curriculum, Riva del Garda, Italy, 2012, pp. 153–162.
[44]Y.-W. Cheng, P.-C. Sun, and N.-S. Chen, "The essential applica- tions of educational robot: Requirement analysis from the perspec- tives of experts, researchers and instructors," Comput. Edu., vol. 126, pp. 399–416, Nov. 2018. [Online]. Available: http://www.sciencedirect. com/science/article/pii/S0360131518302033
[45]C.-W. Chang, J.-H. Lee, P.-Y. Chao, C.-Y. Wang, and G.-D. Chen, "Explor- ing the possibility of using humanoid robots as instructional tools for teaching a second language in primary school," J. Educ. Technol. Soc., vol. 13, no. 2, pp. 13–24, 2010.
[46]D. Alimisis, "Educational robotics: Open questions and new challenges," Themes Sci., Technol. Educ., vol. 6, no. 1, pp. 63–71, 2013.
[47]T. S. Sklirou, "Programming in secondary education: Applications, new trends and challenges," in Proc. IEEE Global Eng. Edu. Conf. (EDUCON), Apr. 2017, pp. 580–585.
[48]M. A. Rubio, R. Romero-Zaliz, C. Ma n˜oso, and A. P. de Madrid, "Closing the gender gap in an introductory programming course," Comput. Edu., vol. 82, pp. 409–420, Mar. 2015.
[49]A. Cruz-Mart ´ın, J. A. Ferna´ndez-Madrigal, C. Galindo, J. Gonza´lez- Jime´nez, C. Stockmans-Daou, and J. L. Blanco-Claraco, "A LEGO mindstorms NXT approach for teaching at data acquisition, control sys- tems engineering and real-time systems undergraduate courses," Com- put. Edu., vol. 59, no. 3, pp. 974–988, Nov. 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0360131512000838
[50]I. Calvo, I. Cabanes, J. Quesada, and O. Barambones, "A multidisci- plinary PBL approach for teaching industrial informatics and robotics in engineer- ing," IEEE Trans. Educ., vol. 61, no. 1, pp. 21–28, Feb. 2018.
[51]N. S. Yamanoor and S. Yamanoor, "High quality, low cost education with the raspberry pi," in Proc. IEEE Global Humanitarian Technol. Conf. (GHTC), Oct. 2017, pp. 1–5.
[52]P. Molins-Ruano, C. Gonzalez-Sacristan, and C. Garcia-Saura, "Phogo: A low cost, free and 'maker' revisit to logo," Comput. Hum. Be- hav., vol. 80, pp. 428–440, Mar. 2018. [Online]. Available: http:// www.sciencedirect.com/science/article/pii/S0747563217305551
[53]I. Paliokas, C. Arapidis, and M. Mpimpitsos, "PlayLOGO 3D: A 3D interactive video game for early programming education: Let LOGO be a game," in Proc. 3rd Int. Conf. Games Virtual Worlds Serious Appl., May 2011, pp. 24–31.
[54]O. Erol and A. A. Kurt, "The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement," Comput. Hum. Behav., vol. 77, pp. 11–18, Dec. 2017.
[55]M. N. Giannakos, M. Divitini, and O. S. Iversen, "Entertainment, engagement, and education: Foundations and developments in digi- tal and physical spaces to support learning through making," Enter- tainment Comput., vol. 21, pp. 77–81, Jun. 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1875952117300307
[56]D. Weintrop, , E. Beheshti, M. Horn, K. Orton, K. Jona, L. Trouille, and U. Wilensky, "Defining computational thinking for mathematics and science classrooms," J. Sci. Edu. Technol., vol. 25, no. 1, pp. 127–147, 2016.
[57]R. Mitnik, M. Nussbaum, and M. Recabarren, "Developing cognition with collaborative robotic activities," J. Educ. Technol. Soc., vol. 12, no. 4, pp. 317–330, 2009.
[58]A. Felicia and S. Sharif, "A review on educational robotics as assistive tools for learning mathematics and science," Int. J. Comput. Sci. Trends Technol, vol. 2, no. 2, pp. 62–84, 2014.
[59]M. Barak and M. Assal, "Robotics and STEM learning: Students' achieve- ments in assignments according to the p3 task taxon- omy—Practice, prob- lem solving, and projects," Int. J. Technol. Design Edu., vol. 28, no. 1, pp. 121–144, 2018.
[60]D. H. Jonassen, Learning to Solve Problems: An Instructional Design Guide, vol. 6. Hoboken, NJ, USA: Wiley, 2004.
[61]M. Akcaoglu and M. J. Koehler, "Cognitive outcomes from the game- design and learning (GDL) after-school program," Comput. Edu., vol. 75, pp. 72–81, Jun. 2014.

INNO SPACE
SJIF Scientific Journal Impact Factor

Impact Factor:
7.569

doi cross ref

ISSN INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

9940 572 462 6381 907 438 ijirset@gmail.com

Scan to save the contact details