

Gamifying the Project in Hardware-based Courses

<http://dx.doi.org/10.3991/ijep.v5i4.4711>

Sasko Ristov, Nevena Ackovska and Vesna Kirandziska
Ss. Cyril and Methodius University, Skopje, Macedonia

Abstract—Using elements of games in non-gaming systems is known as gamification. It has recently been used in many areas to improve some processes, for example, it facilitates the teaching and learning. Students understand some learning objectives more easily when they play games in which they are challenged with rivalry factors and can collaborate more with one another. This paper presents a new approach of gamifying the practical project in the hardware-based course “Microprocessors and Micro-controllers” for computer science students. This change has significantly improved the course - not only that it has improved the average grade of the passed students, but it provoked the students to enroll in other hardware courses of the upper semesters. Even more so, some diploma theses involving microcontrollers were developed for the first time.

Index Terms—education, hardware, gamifying.

I. INTRODUCTION

Gamification is the process of using elements of video games in non-gaming systems [1]. By invoking the same experience as games often do, it improves the user experience and engagement [2]. It provides positive effects which are greatly dependent on the users that use it and on the context in which it is implemented [3]. Gamification is an important process in business. Gartner [4] predicts that at least 50% of companies that manage innovation processes will gamify some aspects of their business this year. Small children also use gamification. Simoes et al. [5] explored how to incorporate the distinctive elements from social games in order to apply them to social learning environments.

Gamification provides different benefits for various socio-demographic target groups. Koivisto and Hamari [6] reported that women achieve greater social benefits than men. They also reported that the benefit of gamification is reduced for the elderly.

Education can also benefit from gamification at different levels [7,8,9]. Students are more motivated, attend the lectures more and learn the material more easily compared to other non-gamified courses [10]. Introducing rivalry factors [11] or challenging problems [12] also affect students' performance. Kim and Lee [13] created a dynamic model and an equation by which one can construct the theoretically ideal game with the highest educational effectiveness.

Many teachers introduced playing games that cover one particular or several general learning objectives of the course. Sutton and Doyle [14] have designed the Spectrum Wars game that covers the basic principles of wireless telecommunications. Students play in teams by creating

and maintaining wireless connections. Playing games increases the collaboration among students, develops students' creativity, and increases the students' interest in computer engineering in general [15]. This also increases the students' mood for learning new and unknown areas, and gaining new knowledge.

This paper introduces a new approach of using gamification, that is, gamifying the practical projects in the unattractive hardware-based course in the (CS) curriculum. Students should develop games on the microcontrollers and by developing and further playing and contesting with them improved the project grades, but also the overall grades of the course. As a consequence, some of them enrolled in other hardware-based courses of the upper study year.

The rest of the paper is organized in several sections. Section II focuses on the specifics of hardware based courses, especially for CS students. It also addresses the need to improve the students' mood, to awake their willingness to learn how the low-level hardware works and to challenge their software skills to be used for controlling hardware. Section III describes the Microprocessors and Microcontrollers (MM) course' specifics, i.e. its difficulties as a classical hardware-based course for CS students. The new concept of gamifying the practical project in the course is presented in Section IV. Section V elaborates on the most interesting game projects created in this MM course. It also elaborates on the problems and difficulties that CS students have solved using their software skills. The course's improvement by gamifying the practical project is presented and discussed in Section VI. Section VII shows the lessons learned from gamifying the practical project, especially for its impact on each gender group. Finally, Section VIII concludes the work.

II. SPECIFICS OF HARDWARE BASED COURSES

It is impossible to imagine today's world without powerful computer hardware (based on microprocessors) or smart embedded systems (based on microcontrollers). This “smart” environment would be impossible without innovative, creative designers and engineers. Knowing how hardware works is very important for CS and computer engineering students, as they will become future engineers. Thus, universities should teach them well and prepare them for the labour market.

Hardware-based courses are represented mainly in two knowledge areas (Architecture and Organization and Platform-Based Development) in the undergraduate CS curricula, mainly acknowledged as core tier hours [16]. Students should be aware of hardware as a platform for executing their software applications, information or control

systems. Some hardware-based areas are elective, such as robotics in the Intelligent Systems knowledge area.

Unfortunately, the number of CS or computer engineering students who are interested in understanding how this modern equipment works in order to improve it, is reduced. Hardware-based courses are less popular among CS students [17]. Students, in general, neither enroll in the elective hardware-based courses, nor learn the core ones to meet the learning outcomes.

Several possible reasons exist that explain this situation. For example, the CS students usually have poor background knowledge in computer engineering and thus lack the knowledge about how hardware works in general. Although the open educational resources (OER) [18], virtual and remote laboratories can facilitate the learning and enhance students' learning activities and outcomes [19], the predefined experiments and exercises are usually limited [20]. The increased usage of e-learning tools sometimes makes this even worse. Some hardware-based courses are mainly theoretical with less practice, so students do not get in touch nor work with specific hardware devices. Another reason might be that historically these courses are considered more difficult than software-oriented ones. However, the course teachers and the educational process, might influence the students' interest in a particular course, regardless of its nature. This is particularly challenging for the teachers of hardware-based courses - how to transform a less wanted subject into an interesting one [21].

The innovative methodology described in this paper is not based on making the hardware-based courses easier to pass, but to make them more interesting, especially for the most important practical parts of the course. The main question is how this can be achieved? Since the CS students have already proved their software-based skills in the previous courses, why not give them more challenging problems, but in their field, that is, more software-oriented problems. Give the students practical projects where they should develop some well known game and change their terminology from the words of low-level hardware language (signals, controls, pins and lack of resources) with the words of high-level programming language (variables, arrays, pointers, graphics, optimization etc).

Some of the changes proposed for making the MM more interesting to the students are presented in the next section of this paper.

III. ABOUT THE COURSE

This section presents the course organization and difficulties that students are facing. MM, or even broader - embedded systems, is usually learned in computer engineering study programs as a core course, but they are elective and thus marginalized in CS study programs.

A. The Course Organization

The course is elective for CS students in their third year of studies. The learning objectives of the course are covered with 2 classes of theoretical lectures, 1 class of theoretical exercises and 2 classes of practical exercises per week. The practical exercises are conducted in computer laboratories and are the most important part for successfully studying hardware based courses [22,23,24].

The main objective of the course is to teach the students about how microprocessors and microcontrollers function,

as well as to examine their main similarities and differences. Many hardware-based learning objectives are already covered in the course, such as low-level hardware interfacing, interrupt handling, communication between processor, memory, bus and peripheral devices, sensors, as well as embedded systems. However, this was traditionally considered a very unpopular subject.

In order to improve its bad reputation, several changes to the MM course took place in the past six years. The course underwent radical changes in the teaching methodology, course organization, hands-on type exercises [25] and so on, and significant improvements were achieved [26]. Still, the results were impressive mostly for the good students, that is, the quality of the course pass rate was improved, rather than the quantitative aspects of the pass rate. The idea was to make the course even more interesting for the students that are easily challenged by interesting projects.

B. Challenges While Working on Hardware

Student projects are usually complex programming assignments that students mostly do at home. For the MM course, after they prepare their algorithm at home, they usually come and work in the faculty laboratory on real hardware (PIC16F887 microcontroller) to overcome the hardware limits. That is, many problems arise when working with real hardware, rather than simulators. These problems are very different from the usual programming assignments that the students are acquainted to. Problems with communication, electrical energy, hardware settings, hardware malfunction and other are typical problems while working on real hardware.

Even more so, in order to pass the course, the students should work with microcontrollers and real embedded systems. Working on microcontrollers is even more demanding than dealing with the microprocessors. Although the microcontrollers can be usually programmed more easily than the microprocessors, they lack vital resources (RAM and ROM). That is, the CS students are used not to care about the hardware because they are expressed in gigabytes for RAM memory and terabytes in hard drives for computers (PC, workstations, notebooks or servers), compared to the bytes of RAM and kilobytes of ROM for microcontrollers. Facing these limitations is a nightmare for unmotivated students.

Since the CS students do not have any background knowledge in electrical engineering, they have difficulties to completely understand how a microcontroller works in reality. All of these challenges put a severe pressure on the teaching staff of the MM course. The instructors should bring the course closer to the students so that they can understand it easily and enjoy in learning it. One effort for making the course more attractive for the students is described in the following section.

IV. GAMIFYING THE PRACTICAL PROJECT

This section presents the new concept of gamifying the practical project in the MM course.

While the lectures and exercises are intended to cover all learning objectives of the course, the practical projects on real hardware and the practical exam can show the achieved level of the relevant learning outcomes. In order to make these projects more interesting for students, many of the projects were gamified starting from 2012 (winter

semester). This means that many expected results were in the form of a well known type of game that students are interested in. For example, such games are: Millionaire, Tic Tac Toe, Minesweeper, Battleship, Snake, Formula, Frogger, Pacman, controlling basketball or tennis game, and so on. Programming games made this assignment more attractive to students. Also, programming games in pairs and playing against each other is convenient for building up the teamwork spirit of the students.

Although these gamified projects are more interesting, they are far from being easy to create. However, the students are challenged to solve them effectively and successfully by using their software skills and knowledge. For example, they were faced with a lack of additional memory and needed additional optimization to define and print Cyrillic letters on Graphic LCD, using the small EPROM (bytes) as an additional memory, etc. More details about some of these game projects are given in the next section.

The most common problem in embedded systems is lack of memory, both RAM and ROM. If a student develops a program with many loops (FOR, WHILE, ...) then they will probably lack the RAM memory during the runtime. One possible solution is to unroll the loops as separate command lines, but in this case the student will probably be faced with a lack of ROM memory. Therefore, students must be very careful and use intensive software skills in developing their gamified projects.

V. GAME PROJECTS

This section elaborates on several interesting and challenging game projects that were successfully created in the MM course. Each game project is described with its specifics and the additional problems, difficulties and challenges that students have solved by using their software skills are explained.

A. Tic Tac Toe

Tic Tac Toe is a simple game. Two players play on a 3x3 board, as presented in Fig. 1. One player marks one field on the board with a circle, while the other player marks a field with an X sign. The students play interchangeably. A player is the winner if they are able to make 3 of their marks in the same row, column or diagonal.

Two types of the Tic Tac Toe game were programmed in one student project. The project was created using the PIC16F887 microcontroller. The first version of the project expected two students to play the game one against the other. In the second version, a student plays against the microcontroller. In this type, the game has three different difficulty levels. The first difficulty level is programmed to make an arbitrary guess. In the second level the controller tries not to make a mistake only with the next move, i.e. to prevent the opponent to make three marks. The third level is the most important and challenging, since it integrates the knowledge of artificial intelligence (AI) that the students have learned previously. This AI knowledge was integrated with the knowledge of microcontrollers. More than one move ahead is predicted and the best possible next move is played by the microcontroller.

Programming this game does not require much memory, neither RAM nor ROM. However, it is challenging since it integrates the artificial intelligence for pro-

gramming more game levels. That is, the students use their software skills to program the game on the microcontroller.

B. Battleship

Another interesting and challenging game project is Battleship, which is presented in Fig. 2. This game is played with two players. First, each player places their ships on a fixed predefined 15x15 board. Later, each player makes a shot to a position where they believe that the opponent's ships are placed. The winner is the player who will discover the position of all ships of the other player and destroy them all.

When implementing this game an interaction between two PIC16F887 microcontrollers was enabled. Serial communication was used for the interaction. This game was programmed by a group of two students. The first student developed the administration part, where both players can place their ships on the board. The other student developed the game interactions, given boards, as well as the serial-port communication.

The battleship project, in the worst case, requires huge memory resources. Students that worked on this game faced the memory problem that comes from the hardware implementation. In order to store two 15x15 boards, if each field is saved in one byte, 450 bytes are needed. This is a big memory resource for the PIC16F887 microcontrollers. The problem was solved by storing the information where the battleships are, instead of the whole board state. Because only 20 fields are part of the battleships, their positions are stored in 2 bytes. As a result, instead of using 450 bytes, only 80 bytes were used. Of course, the lack of memory resources increased the calculating time to check if the shot of the opponent player was a hit or a miss.

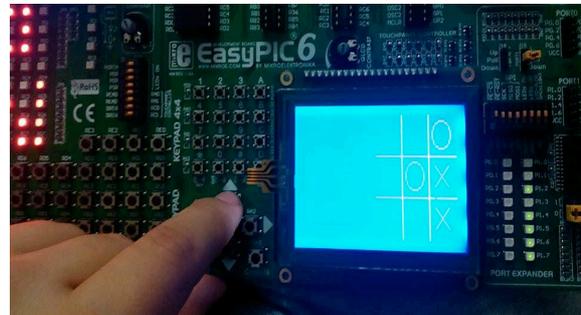


Figure 1. The Tic Tac Toe game project

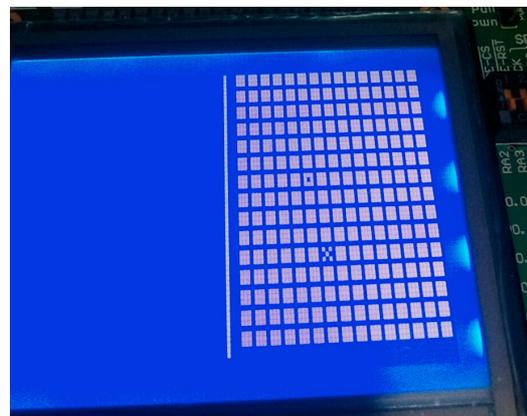


Figure 2. The Battleship game project

C. Memory

The memory is a game where a player opens two hidden fields and they remain open if both are the same. The game finishes when all fields are opened. It is played by one person. An even number of hidden fields are shown on the LCD of the PIC16F887 microcontroller. Different pairs of signs are hidden under these fields. The player should try to remember the signs shown when the fields are revealed. The player's task is to match all signs as soon as possible. Different modules are used to program this game.

Lack of memory was also resolved successfully by using the student's software skills of storing the information in a reduced manner.

D. Frogger

This is the well-known game, where a frog should cross the street containing five lanes while the vehicles are passing. The frog should not get hit by the moving vehicles in the process. When the player reaches the next level, the vehicles drive faster. Fig. 3 presents the game on GLCD (Graphic LCD).

For the project implementation, three vehicle types are included. Vehicles on lanes 1, 3 and 4 are moving from left to right, while the other vehicles (lane 2 and 5) are moving in the opposite direction, from right to left. The frog can move in four directions: up, down, left, right using the keypad. The game finishes when the player loses all lives. If a player achieves greater number than the current record, they will write their names in EEPROM.

Several challenges exist in this game project. As Fig. 3 shows, there are Cyrillic letters on the GLCD, although it does not support the Cyrillic alphabet. That is, a student must use their mathematical and software skills to draw literals. Additional accommodation is also needed for the vehicles. However, these drawings require additional memory usage. To overcome the shortage of memory, compression is implemented.

E. Minesweeper

This is the well known game of sweeping the mines. The player should open all of the fields that do not have a mine. If the opened field does not have a mine, then it shows the number of mines around it. If this number is zero, then all neighbor fields are being opened recursively. If the field is a mine, then the player loses the game. Fig. 4 presents the development of this game project.

The Minesweeper project is very demanding and the student was faced with the following challenges: Lack of RAM (368 bytes), too small call stack of 8 bytes, hard and slow drawing of the Cyrillic characters on the map. All these challenges were solved using the previously gained student's software skills. For example, one byte (8 bits) of RAM is used to store information for a field for several issues: one bit whether the field is opened or not, another bit if a field is a mine or not, and four bits for storing the number of the mines in the neighboring fields. The recursion is redeveloped iteratively, although it resulted with greater time requirements. Also, some code was moved from the call methods to the part where the methods are called (from a specific function to the main function). This change reduced the RAM requirements, but increased the ROM requirements. The third challenge was twofold: drawing smiley and Cyrillic letters that is not supported. It

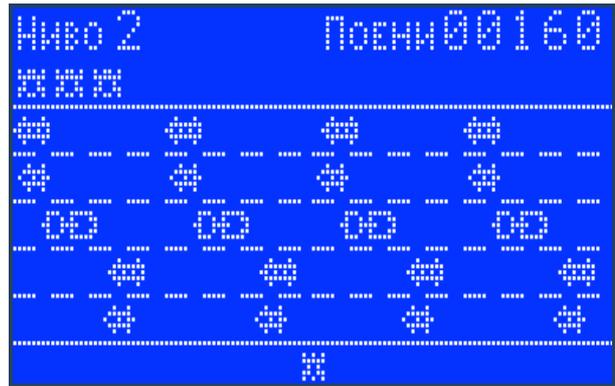


Figure 3. The Frogger game project

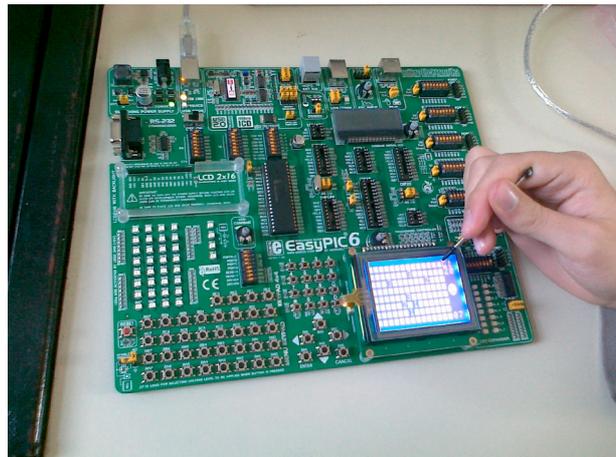


Figure 4. The Minesweeper game project

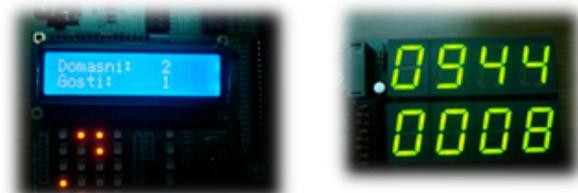


Figure 5. Controlling a basketball game

was resolved as a bitmap. The lack of memory was alleviated by writing in the EEPROM.

F. Controlling a basketball game

In this project a player does not play the game, but controls it, as shown in Fig. 5. The controller can start the attack (countdown from 24 to 0), stop (pause) the time, and restart to 24 seconds again. Also, he/she can add 1, 2 or 3 points for both teams (home / guests), but also can reduce the score for 1, 2 or 3, if the judges change their opinion after some suspicious situation. The 7-segment display was used both for the total time and the time of the current attack.

The student faced classical hardware problems in this project:

- Problems with measuring time
- Catching the interrupts
- Problem with the power supply
- Sensitivity of keypad and pins
- Lack of RAM memory

The additional software problem appeared for the regular expression

```
if(A && B)
    {Expression}
```

where the microcontroller did not execute neither the *if* clause, nor the expressions within. This challenge was also solved by using the software testing procedures. That is, the code was changed to

```
if(A)
    if(B)
        {Expression}
```

which worked properly as expected.

G. How to be a Millionaire

How to be a millionaire is a well known show and developing the project for simulating it was a challenge. A player answers one by one question until the first answer is not answered correctly or up to 15 questions.

Fig. 6 presents the implementation of this game project. GLCD is also used for presenting the questions and choosing the correct answer.

Since the questions require RAM, they are stored in the PC, and then transferred to the microcontroller via a serial port. Lack of RAM disallowed writing Cyrillic letters.

H. Tetris

This is also a well-known and popular game. A random figure is falling down, which can be rotated or moved left or right, as depicted in Fig. 7. When a whole row is fulfilled, it disappears. The player loses if 10 objects are placed one above each other.

The map is 40x40 and is stored as a matrix in EEPROM. This increased the starting time to a few seconds. Similar challenges appeared as described in the previous projects.

I. Formula

This game project consists of moving the formula left or right to avoid the objects that are approaching. Also, the player should not hit the border. The game has seven levels so that the formula speed increases with each level. The next level is reached after all objects of the current level have been avoided successfully. Otherwise, the game ends. If the player successfully finishes all the seven levels then he/she wins the game.

J. Snake

The snake project was created following the rules of the popular Snake game. The snake can move in the four directions (left, right, up, down) and is controlled by the keypad. It was implemented on the microcontroller in three different levels. In each level, the snake should collect 17 pieces of food that appear on the screen, but with more forbidden areas to make the game more difficult.

K. Pacman

Pacman is also a well-know computer game. The goal of the game Pacman is to collect all the points in a matrix maze environment. However there are ghosts that chase Pacman. If the ghost catches Pacman, the game is lost. The player controls Pacman using the keypad and every

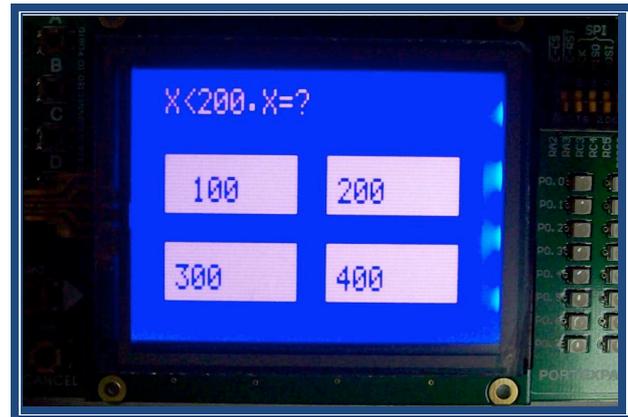


Figure 6. Millionaire game project

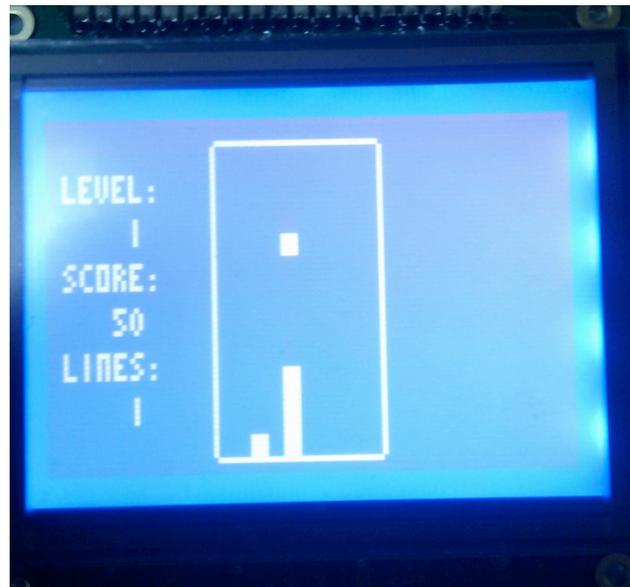


Figure 7. Tetris game project

movement is displayed on the GLCD. A player wins if he/she collects all points.

In the implementation of this game students faced several difficulties. Due to memory limitations the game environment in which Pacman moves was reduced to 5x5. Due to the constraint on the microcontrollers the ghost moves back and forth only in the lowest row. No other intelligence was introduced.

VI. THE RESULTS OF GAMIFYING THE PRACTICAL PROJECT

This section presents the results of the average project grade and the average total grade during the period of five study years. Gamifying the practical project was introduced in the study year 2012/2013. It also analyzes the correlation between the points the students gain for the project part and the final MM grade. Additionally, the socio-demographic issues are analyzed, that is, how the gamifying the practical project impacted to a particular gender group.

A. Number of students and gender distribution

Fig. 8 represents the number of students during these years. We observe that the number of students has decreased in the last two years. The trend was expected, i.e.,

starting from 2013/2014 this course was declared fully elective. However, what was very interesting is that even though it is considered as a tough, hardware-oriented course, and students were not obliged to take it, the number of the enrolled students was always satisfactory.

Another important issue is the gender distribution. Almost each year, the females are 20% to 27%, except 40% in 2012/2013. This shows that, although the course became elective, the percentage of females remained.

B. Results for the practical project

Fig. 9 shows the average results of the practical projects through the years, in the period from 2009 to 2014, separately for each gender group and total. The first noticeable improvement of the MM course happened in 2010/2011, when the average grade was increased by 1.28. The average project grade was increased for both males and females. This is the year of the course revolution, when it underwent dramatical changes in the syllabus [17,21]. But, the next huge improvement happened in 2012/2013 (jump of 0.77), exactly when the projects were gamified. A greater jump is noticed for the females. In 2012/2013 females had an average project grade of 2.45 higher than the one that females achieved in 2011/2012. On the other hand, the average project grade for the males showed an improvement of only 0.14. Before gamifying the practical project, males had better average projects results, while after gamifying the practical project, the females were better.

The standard deviation of the average points that students achieved per year is presented in Fig. 10. It showed the values greater than 2 before gamifying, while it reduced its value to 1.07 in the last year.

The standard deviation is greater for males than females, (except in 2010/2011, when many honorable students were enrolled, as explained later). Although this value is greater, it is close to the standard deviation of the whole group of students together.

C. Results for the course's final grade

Gamifying of the project had an impact on the final grade as well. If the student performs with 50%-60% of the required knowledge of the MM course, the grade acquired is 6 (the lowest passing grade), 61-70% is 7, etc, and if the performance is 91%-100% the student gets the maximum grade 10. The student can achieve up to 10% of the final grade from the project.

Overall, the average grade of the course has increased by more than 1, that is, from 6.66 in the study year of 2009/2010 to 7.88 in 2013/2014. Fig. 11 explains the students' accomplishments through the years in general. However, one can observe that there is an improvement in the students' average final grade through the years after the implementation of gamification in the year 2012/2013, which is shown in Fig. 11. Although the project can impact up to one grade in the final grade, it is obvious that more interesting projects make students more interested in the subject.

Compared to the years before gamifying, both males and females students obtain a higher average final grade after the introduction of gamifying the practical project for the MM course. The average final grade of the males has grown with gamifying the practical project to 7.17 in 2012/2013 and 8.08 in 2013/2014. The females' im-

provement in general is also observed, but is not as significant as the one for males in both study years with gamifying the practical project. That is, the average final grade of the females has grown with gamifying the practical project to 7.10 in 2012/2013 and 7.40 in 2013/2014. This means that still, in general, females got worse average final grade than the males.

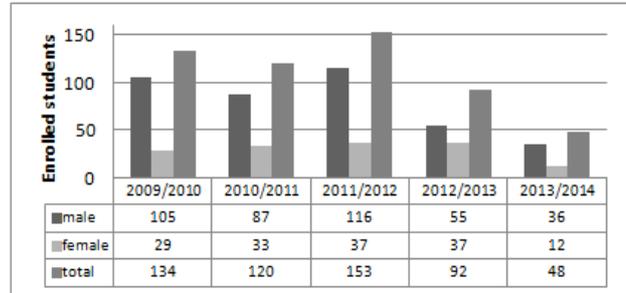


Figure 8. Number of enrolled students during past years

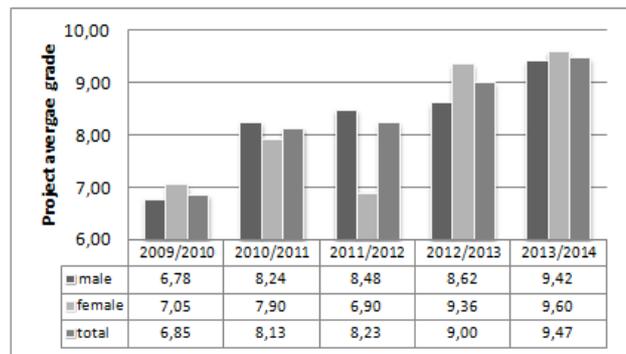


Figure 9. Average project points per year (maximum 10 points)

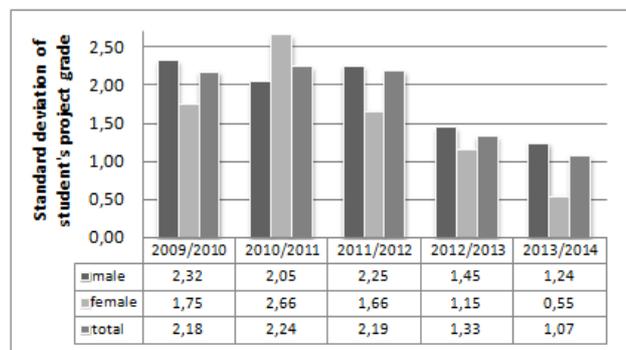


Figure 10. Standard deviation of average project points per year

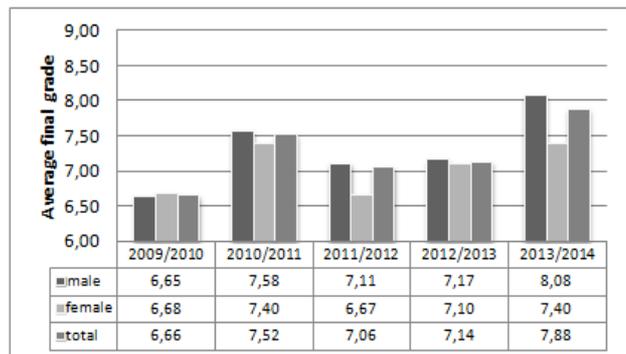


Figure 11. Average course grade per study year

Analyzing Fig. 11, a small jump in the average grade is observed in 2010/2011. The highest average grade in 2010/2011 is 7.52, which is higher than in 2011/2012 and in 2012/2013. The explanation is that the year 2010/2011 was the first study year with the new revolutionary MM curriculum, and also 10 of the 120 students that were enrolled in 2010 were honorable students, who later were chosen as demonstrators at our faculty.

The standard deviation of the average grades per year is presented in Fig. 12. We observe that the standard deviation rises from 1.18 to 1.45 during years and follows the average final grade (apart of 2010/2011 study year). The standard deviation of females in the 2011/2012 study year is smaller than males in the same year and also smaller than females in other years because the gap for females appears also for the females' grades in the same year.

VII. LESSONS LEARNED

The power of the teacher is not only in their ability to teach the lectures and exercises professionally, but to motivate the students as well, especially for the courses which the students are not interested in. This paper presents the positive experience of gamifying the practical project in the hardware-based MM course. Several projects that were developed during the MM course are discussed, as well as the challenges that the students faced while working on those projects. Lessons learned from this approach are that using the CS students software skills for hardware control succeeded completely. Additionally, the students started to enroll in other hardware-based courses in the upper semesters (robotics, for example), even though most of them are elective. Some students developed the same games in the robotics, such as Tic Tac Toe which is presented in Fig. 13. Even more so, two diploma theses have been written, for the first time at the faculty, whose main topics are Microcontrollers games.

Gamifying the practical project shows different results of various target groups. It has had a huge impact on the females' average project grade. The female average grade project had increased by impressive 30.10%. Gamifying the practical project improved the males' average project grade, as well, for 15.11%. One can notice that improvement of the females was twice greater than for the males. This benefit of gamifying the practical project is similar of the one that Koivisto and Hamari achieved in their research [6].

Although the project only contributes 10% to the overall points gained in the course, gamifying the practical project had an observable impact on the learning of the other parts of the course. Although the small improvement of only 0.14 points of the project average grade for the males does not have any direct mathematical impact on the average final grade (each grade has an interval of 10 points), still it impacted indirectly on the other parts of the course. That is, the males improved the average final grade in the last two study years (with gamifying the practical project) by 7.21% compared to the former three study years. However, this improvement is totally opposite for the females. That is, they improved their final grade only by 4.81%, although their improvement in average project grade was significantly better than the males taking the MM course.

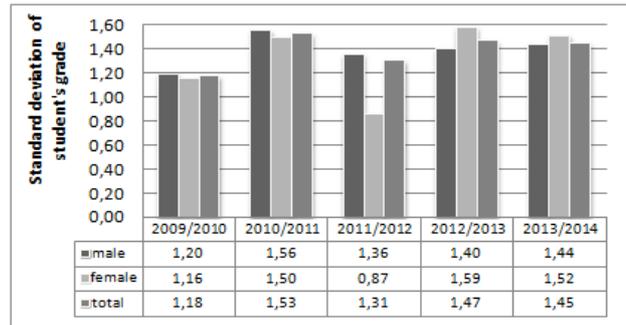


Figure 12. Standard deviation of grades per year

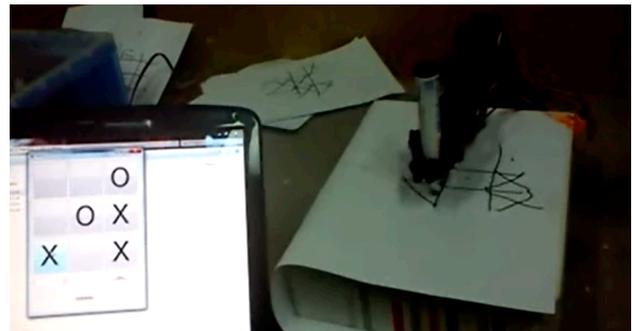


Figure 13. Example of gamifying the projects in robotics (Tic Tac Toe)

VIII. CONCLUSIONS

Working on hardware courses is a real challenge in the CS curricula. Many students find the hardware courses difficult and not interesting. This was the motivation to make improvements in the MM course, and to make it more popular among the students. This course underwent many transformations. The last one is gamifying the course's practical project that was explained in this paper. The results show that it is very important to raise the interest of the students by making the subject more fun, yet similar to what they already know from other subjects.

The game concept seems to be important when limitations are real. Namely, working with limited hardware resources is not what the CS students are used to. They usually face problems where the memory is not a resource. However, in working with microcontrollers in particular, these resources are scarce. To enable the students to face these limitations more easily gamifying the practical projects was enforced.

Gamifying the practical project has improved the course significantly. Starting from a very unpopular subject, where students were barely passing, they now achieve an average grade of 9, with more than 60% of students achieving the maximum grade 10 for practical projects. This is a great achievement for a difficult and demanding course.

The results show that gender is an important socio-demographic parameter with gamifying the practical project. Both genders have improved the average project grade, as well as the average final grade. The females improved their average project grade two times more than the males for this hardware-based course. Even more, the enrollment of the female students remained although this traditionally male course became elective. However, this improvement has affected the males' final grades more than the females.

REFERENCES

- [1] S. Deterding, M. Sicart, L. Nacke, K. O'Hara, D. Dixon, "Gamification. Using game-design elements in non-gaming contexts," in *CHI '11 Extended Abstracts on Human Factors in Computing Systems (CHI EA '11)*. ACM, pp. 2425-2428.
- [2] K. Huotari, J. Hamari. "Defining gamification: a service marketing perspective," in *Proceeding of the 16th International Academic MindTrek Conference*. ACM, 2012, pp. 12-17.
- [3] J. Hamari, J. Koivisto, H. Sarsa, "Does gamification work?--a literature review of empirical studies on gamification." *System Sciences (HICSS), 2014 47th Hawaii Int. Conf. on*. IEEE, 2014.
- [4] Gartner, "Gartner says by 2015, more than 50 percent of organizations that manage innovation processes will gamify those processes", <http://www.gartner.com/newsroom/id/1629214>, April 12, 2011.
- [5] J. Simões, R. D. Redondo, A. F. Vilas, "A social gamification framework for a K-6 learning platform," *Computers in Human Behavior*, Vol. 29, No. 2, March 2013, pp. 345-353.
- [6] J. Koivisto, J. Hamari, "Demographic differences in perceived benefits from gamification," *Computers in Human Behavior*, Volume 35, June 2014, pp. 179-188.
- [7] K. Erenli, "The Impact of Gamification - Recommending Education Scenarios," *Int. Journal of Emerging Technologies in Learning*, vol. 8, Special Issue ICL2012, pp. 15-21.
- [8] M. T. Restivo, A. Cardoso, "Online Experimentation in Education and Training," *International Journal of Engineering Pedagogy*, vol. 4, no. 2, pp. 52-56.
- [9] Y. Chen, T. Burton, M. Vorvoreanu, D. M. Whittinghill, "Cogent: A Case Study of Meaningful Gamification in Education with Virtual Currency," *International Journal of Emerging Technologies in Learning*, vol. 10, no. 1, pp. 55-59.
- [10] G. Barata, S. Gama, J. Jorge, D. Goncalves, "Engaging Engineering Students with Gamification," *Games and Virtual Worlds for Serious Applications (VS-GAMES), 2013 5th International Conference on*, pp.1-8, Sept. 2013.
- [11] M. Laskowski, "Implementing gamification techniques into university study path - A case study," *Global Engineering Education Conference (EDUCON)*, 2015, Estonia, pp.582-586.
- [12] M. Mihova, M. Jovanov, E. Stankov, "On the role of challenging math problems in the discrete mathematics courses," *Global Engineering Education Conference (EDUCON), 2015 IEEE*, 18-20 March 2015, Tallinn, Estonia, pp.730-736, best paper award.
- [13] J. T. Kim, W.-H. Lee, "Dynamical model for gamification of learning (DMGL)," *Multimedia Tools and Applications*. 2013, pp. 1-11. <http://dx.doi.org/10.1007/s11042-013-1612-8>
- [14] P.D.Sutton, L. E. Doyle, "The gamification of Dynamic Spectrum Access & cognitive radio," *Global Engineering Education Conference (EDUCON), IEEE*, 2014, Turkey, pp.107-113.
- [15] O. M. Zamyatina, P. I. Mozgaleva, Y. O. Goncharuk, O. V. Marukhina, "Game technologies in teaching "Mathematical modeling"," *Global Engineering Education Conf. (EDUCON), 2015 IEEE*, 18-20 March 2015, Tallinn, Estonia, pp.847-851.
- [16] ACM/IEEE-CS Joint Task Force on Computing Curricula, "Computer science curricula 2013," ACM Press and IEEE CS Press, Technical Report, December 2013. [Online]. Available: <http://dx.doi.org/10.1145/2534860>
- [17] S. Ristov, M. Stolikj, N. Ackovska, "Awakening curiosity — hardware education for computer science students," *MIPRO, 2011 Proc. of the 34th International Convention*, pp.1275-1280, 2011.
- [18] N. Ackovska, S. Ristov, "OER Approach for Specific Student Groups in Hardware-Based Courses," *Education, IEEE Transaction on*, vol. 57, no. 4, pp. 242-247, Nov. 2014
- [19] E. Sancristobal, S. Martin, R. Gil, P. Orduna, M. Tawfik, A. Pesquera, G. Diaz, A. Colmenar, J. Garcia-Zubia, M. Castro, "State of art, initiatives and new challenges for virtual and remote labs," *Advanced Learning Technologies (ICALT), 2012 IEEE 12th International Conference on*, pp. 714-715, 4-6 July 2012
- [20] D. Pop, D. G. Zutin, M. E. Auer, K. Henke, H.D. Wuttke, "An online lab to support a master program in remote engineering," in *Proceedings Frontiers in Educations Conference, 2011*, pp. GOLC2-1-GOLC2-6.
- [21] M. Stolikj, S. Ristov, N. Ackovska, "Challenging students software skills to learn hardware based courses," *Information Technology Interfaces (ITI), Proceedings of the ITI 2011 International Conference on*, pp.339-344, 2011
- [22] I. Kastelan, D. Majstorovic, M. Nikolic, J. Eremic, and M. Katona, "Laboratory exercises for embedded engineering learning platform," in *MIPRO, 2012 Proceedings of the 35th International Convention, 2012*, pp. 1113-1117.
- [23] J. Qian, R. Wang, S. Shi, Y. Zhu, and Z. Xie, "Simplifying and integrating experiments of hardware curriculums," in *ICCSIT, 2010 3rd IEEE Int. Conference on*, vol. 9, 2010, pp. 610-614.
- [24] D. Kehagias and M. Grivas, "Software-oriented approaches for teaching computer architecture to computer science students," *J. of Comm. and Computer*, vol. 6, no. 12, pp. 1-9, Dec. 2009.
- [25] N. Ackovska and S. Ristov, "Hands-on improvements for efficient teaching computer science students about hardware," in *EDUCON, 2013 IEEE*. Germany: March 2013, pp. 295-302.
- [26] Ristov, S.; Ackovska, N.; Kirandziska, V.; Martinovikj, D., "The significant progress of the Microprocessors and Microcontrollers course for computer science students," *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 37th International Convention on*, pp.818-823, 2014

AUTHORS

Sasko Ristov, Nevena Ackovska and Vesna Kirandziska are with the "Ss. Cyril and Methodius" University, Faculty of CS and Engineering – Skopje, Macedonia ({sashko.ristov, nevena.ackovska, vesna.kirandziska}@finki.ukim.mk).

This work was supported in part by the Faculty of Computer Science and Engineering, "Ss. Cyril and Methodius" University, Skopje, Macedonia. This article is an extended version of a paper presented at the IEEE EDUCON2015 Conference, held from March 18th-20th, 2015, in Tallinn, Estonia. Submitted 09 May 2015. Published as resubmitted by the authors 10 October 2015.