# CONGRUENCE EXAMINATION OF NXT ROBOTS IN THE EDUCATION OF PROGRAMMING AT KF GAMF COLLEGE

**© Róbert PAP-SZIGETI**
**© Attila PÁSZTOR**
**(Kecskemét College)**

**pap-szigeti.robert@gamf.kefo.hu**
**pasztor.attila@gamf.kefo.hu**

*At Kecskemét College a new experiment was started in the beginning of 2008. We examined the pedagogical effects of using this technology while studying programming in a control group designed experiment: only the test group used robots. Our hypothesis was that this technology makes the way of learning more enjoyable, and we can improve both the motivation for learning and the programming knowledge of our students to improve and be more applicable. We measured the motivation for learning, programming self-concepts and basic programming skills in the beginning and at the end of the semester assess results as a comparison. In this article we tried to show how new toy-like devices can be used in the education of programming both at beginner and intermediate levels.*

**Keywords**: teaching of programming, motivation to learn, model robots

In today's technical higher education in Hungary a great emphasis has been put on teaching IT as well as teaching programming and robotics. For every student developing their programming competence is extremely important. Furthermore getting their algorithmic thinking and problem solving abilities to a higher level can be even more important. Without these their future job is impossible even if the task is to solve other than certain programming problem. Beyond learning different programming languages and programming techniques our task is to develop abilities like getting students be able to understand and solve both theoretical and practical problems in different fields.

In this article we tried to show how new toy-like devices can be used in the education of programming both at beginner and intermediate levels. We hope that if the syllabus is interesting, if the way of learning is enjoyable, if the lesson is spectacular, the result is much more significant. ”The task, you are thinking of, may seem to be simple, but if it arouses your interest it mobilizes your inventiveness and finally you can solve it by yourself, you can experience the excitement and triumph of exploration”. By using model robots we can make teaching programming more interesting and we can also give the students the impression of just playing.

We use two robots of LEGO, the Mindstorms RCX and the NXT. In college education our goal is to make students like the subject with the help of this technology at the beginning of their studies, and towards the end of the course, to be ableto put their programming knowledge into practice (including Artificial Intelligence, communication, image recognition, parallel processing, swarm intelligence). To demonstrate results, we also show some simple model robots (that follows routes or looks for a spotlight) as well as more difficult ones (e.g. that play 'stone - paper - scissors' game, or behave like a scanner or the simple communication between the robots). Applying this technology in teaching can help motivation of students to become stronger and both 'beginners' and 'experts' can acquire a deeper knowledge in programming through solving problems with various difficulties. To present the usage of one or two fields or to make some creative algorithms can also be a proper theme for dissertation and students' scientific research (TDK).

At the Kecskemét College, Faculty of Mechanical Engineering and Automation at the department of Information Technology the IT students are thaught programming in C/C++ language four lessons per week, during three semesters. Students can get enrolled in subjects dealing with programming from the second semester and they study programming during three semesters. In the first semester they study the most basic programming, like different loops, branches, writing simple functions, and some more complicated structures. In the next semester text and binary file operation and more difficult data structures are examined, while in the third semester they get acquainted with object-oriented programming.

Unfortunately, together with my colleagues we have seen that students' results are getting worse ad worse over the years and more and more students have to study programming again, because of their unsuccessful exams. Furthermore they enjoy this subject less and less. To confirm empirically our experiences we made a survey.

The survey was made with 100 participants who have finished their first semester (maybe after more attempts) (Kiss & Pásztor, 2006). So these students have already had time to shape either a positive or a negative attitude to the traditional teaching of programming. In the survey we examined students' previous knowledge of programming at secondary school, to see the entry level of students, how easy they take things in programming at the college for the first time, how much they like the subject and how important do they think programming for their future career.

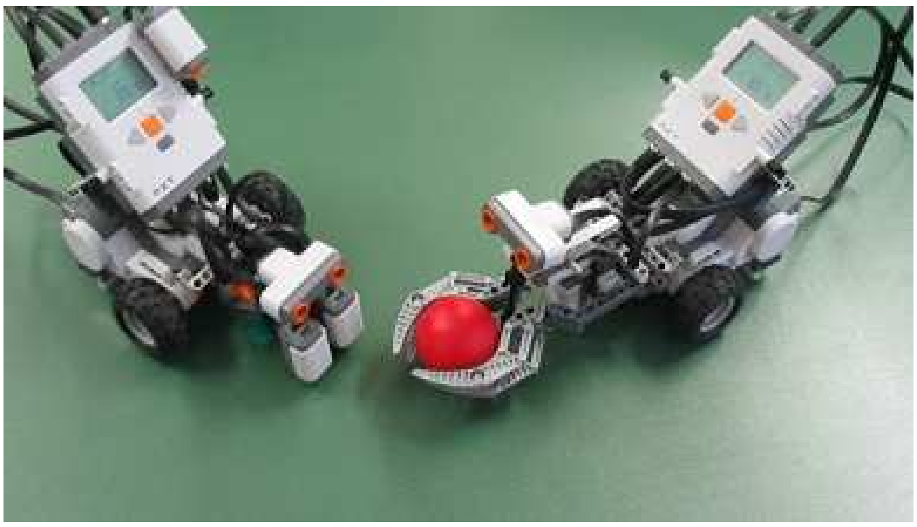
The survey showed that students entering into higher education have different levels of knowledge in programming. Forty-nine per cent of them have never studied programming at all and the other half is varied. Some studied it for only one hour per week while others had six or eight lessons per week for years. Of course their grades show that students who already had background knowledge, reached better results, the average grades of students, who have not studied informatics before was 2.86, while the average grades was 3.31 for those who have studied informatics before. This difference is significant on a 95% probability level. It turns out from the survey that 60% of those who have not studied programming before failed after the first semester while 80% of those who have studied it before passed successfully. The difference is significant ($\chi^2 = 12.95$; $p < 0.05$) (Pásztor, 2008).

Besides these bad results, it was also embarrassing that sixty per cent of students were indifferent or did not like programming and fifty per cent did not think that programming could be important for their future carrier. The mean value of the attitudes towards IT is 3.19 converted into the traditional five-level scale, which is just above the neutral level. This is a bit better than the level of attitude towards the science subjects at the end of public education, but not at all joyful, because students chose our school on account of IT (Csapó, 2000). The high rate of failures, the bad results, the weak attitudes made us decide to find a way for improvement and increased motivation. Because of these bad results and the students indifference we started to search for such new methods and technology that could help teaching become more effective, spectacular and easier, as well as make students enjoy the subject. It is likely that there is a considerable divergence between students' algorithmic ability and abstraction level. Our work with these technologies may help to develop students' thinking skills (Inhelder & Piaget, 1967/1984).

## The new subject – Programming model robots

One of the devices we found was introduced in the 6th ICAI conference (Istenes, 2004). It was a programmable robot set by LEGO, called Mindstorms RCX. The robot can be seen as a toy for the first sight but with the help of it teaching can be more interesting, effective and spectacular. A second device we used the LEGO NXT robot set. From the robot set we can be build humanoids, animals or cars. It has more sensors and bigger memory than the RCX, and the communication between the robots and the computer is done via blue tooth. The NXT won the innovative first prize at the Nürnberg Toy Exhibition in 2006. Thanks to its new sensors (sound, ultrasonic sensor), servomotors and Blue Tooth techniques it became very popular (figure 1).

*Figure 1. LEGO NXT Robots*



A proper operating system runs in the robots which are loaded through the infra red port or via blue tooth. The programs are written in C-like programming languages called NQC or NXC. In this languages there are various loops, if-then and switch-case branches, function with value and reference passes, tasks and includes files like in most C-type programming languages but its greatest advantage is that after loading the ready-made programs into the robot the results can be seen visually in the form of behaviour.

After purchasing the robots, we initiated a new subject called Programming Model Robots. Students can study programming with the help of robots playfully. They work in groups and can use the previously studied theory in practice. During the practical lessons when they plan and solve their tasks, students can work in teams. The role of teachers changes its inflexible practice to become rather coordinators, who raise problems.

Participants plan the working process, they share the job, communicate with each other, finally they summarize and present the results together. While doing their task they can get enough theoretical and practical experience which can be very useful for their later career.

Beginners develop various applications during practice. For example they build cars and the robot cars follow a route with the help of two light sensors. One of the sensors is next to the route and the other is on it. The other example is that the robots carry some balls to the end of the table and drop it down. Both of these very simple programs were made by students after six-seven hours of practice. After eight or nine hours of practice students build scanners, which read in a fifty by twenty resolution a character drawn on the paper. Then the robot records the read data in the text file.

Some practical applications are good examples that show how we can use in practice spectacularly the previously studied theory with the help of the robots. For example we use the 'Maximum choice' theory in one of these exercises. The robot has to choose the lightest point of the room before it leaves from the corner. First it moves round and round looks for the right direction then it moves towards the chosen point. During this process, if we hide the source of the light, the robot calibrates again and finally it finds the lightest point. A similar problem arises when the robot, with its temperature sensor, seeks for the hottest part of the room.

Some students with better qualities develop more difficult tasks. They built humanoids or animals and the robots can communicate with each other through their IR ports or via blue tooth. Taking the advantage of opportunities they can make different kinds of applications in which two or more robots follow each other or play 'stone-paper-scissors', or carry something together.

## Our empirical results

### Our hypothesis

Our first hypothesis was that model robots can make learning enjoyable. When students use these devices, they can experience the growth of knowledge and the pleasure of discovery, so students can get involved deeply in work. It may increase the efficiency of learning because it operates as a very strong motivation (Csíkszentmihályi, 2001). Furthermore, the challenging power of the learning exercise can be optimal with these devices, so it may activate their mastery motivation, which plays a fundamental role in the process of developing skills (Józsa, 2005).

The second hypothesis was that the motivation to learn (especially programming self-concept) can be improved by model robots. The success in problem solving within programming and in using robots can affect students' self-concept.

As third hypothesis we formulated that we can increase our students' programming knowledge to a higher level and make them to be able to apply what they learnt. For those who never learned programming before during their secondary studies, the level of abstraction within college education can seem to be a bit too high at the beginning. We hoped that the model robots can develop students' skills at level of experiences (Inhelder & Piaget, 1984; Nagy, 2000), which can lay the foundations for the successful acquisition of higher order thinking skills.

## The sample and the measuring devices

We wished to support our first hypothesis only in an indirect manner, with the examination of the frequency of students' participation. It is expected, that if students study enjoyably with the mobile robots, then the quantity of absences decreases.

We organized a post-test to confirm our second and third hypothesis. It was organized with an experimental group (41 students who used model robots during the experimental semester) and a control group (46 students, who were taught by the traditional methods). All sample members have learnt the subject Programming I earlier. The two sub-samples were chosen in such a way that the differences between them were not significant in the main variables (see later in this section).

A programming test with 15 items and a questionnaire with 17 items were filled in by the experimental and control group. The test measured basic programming knowledge and skills; its reliability was Cronbach's alpha = 0.72. We hope we can increase the reliability of our measure by some new items in the future, but we have to apply a measuring device which can used within a lesson.

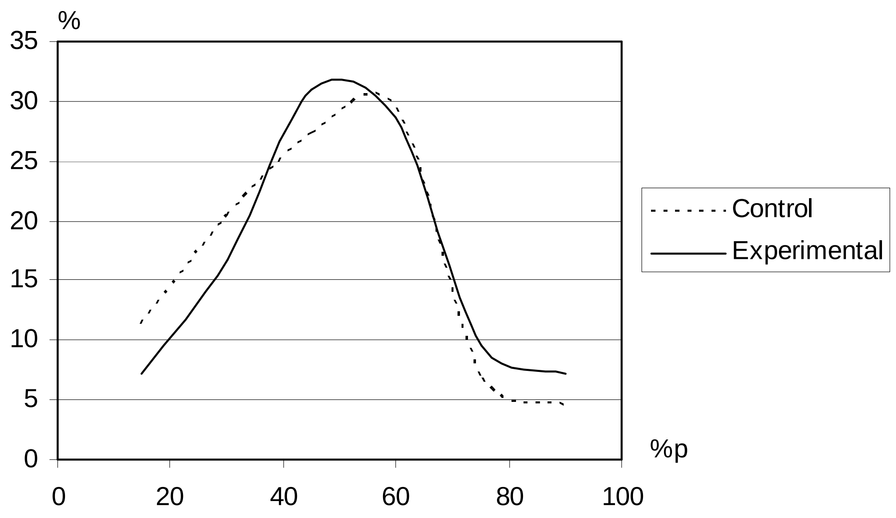
The questionnaire contained a Likert scale answer format with five degree. Background data, the attitude to programming, and the programming self-concept of the students were measured. The pre-test was filled out at the beginning of the semester; the post-test was at the end of the semester, so the period of development lasted three months. The same test and questionnaire was used for pre- and post-tests.

## Results of the pre-test

The majority of the students (78%) do not have any family members who could help with programming at home. There is no significant difference between two sub-samples ($\chi^2 = 0.141$; $p = 0.71$). The experimental group and control group are similar in average number of high school semesters that they learnt programming ($\chi^2 = 6.58$; $p = 0.16$); 43% of our students have not learnt programming in their high school.

We assessed our students' programming self-concept by six Likert scale questions. The answers were arranged into one factor (KMO = 0.86; Bartlett-test: $\chi^2 = 354.95$; $p < 0.001$). The created factor with percentage point scale converged to normal distribution quite well. The programming self-concept of the experimental group members was as high as that of the test group members' results concerning self-concept ($x_e = 46.7$ %p; $x_c = 45.8$ %p; $F = 0.01$; $p = 0.95$; $t = 1.54$; $p = 0.13$). The frequency of the experimental group's and the control group's self-concept is shown in figure 2.

*Figure 2. Frequencies of programming self-concept in the pre-test*



We could not find significant difference between the experimental group's and the control groups' programming skills and knowledge ($x_1$ = 45.8 %p; $x_2$ = 40.7 %p; F = 0.91; p = 0.34; t = 1.61; p = 0.11). The two sub-samples have very similar starting parameter, that is why we suppose that the differences measurable at post-test are caused by the educational affects of the new devices and methods.

## Development of our sub-samples

The experimental period was only three and a half month. From the teacher's point of view it is an enjoyable experience when students take part in course work with a delight. Students who got enrolled in the subject Programming model robots were absent significantly less than those in the control group ($\chi^2$ = 3.22; p = 0.03).

Neither experimental group's ($x_{pre}$ = 45.8 %p; $x_{post}$ = 49.1 %p; t = -1.23; p = 0.23) nor control group's ($x_{pre}$ = 40.7 %p; $x_{post}$ = 43.0 %p; t = -1.01; p = 0.34) programming knowledge developed significantly during this semester. The correlations between the pre-test and post-test are similar ($r_e$ = 0.631; $r_c$ = 0.618). It is possible, that the period of development was too short. But we can suppose that the restructuring and the qualitative improvement of our students has begun. It may cause a temporary declension in the experimental group members' achievement.

The average programming self-concept of the control group was stagnating ($x_{pre}$ = 45.8 %p; $x_{post}$ = 43.6 %p; t = -0.45; p = 0.66). However, we found a significant development in the experimental group's programming self-concept ($x_{pre}$ = 46.7 %p; $x_{post}$ = 51.7 %p; t = -2.60; p = 0.01). The difference between the two sub-samples is observable in frequencies of self-concept (figure 3).

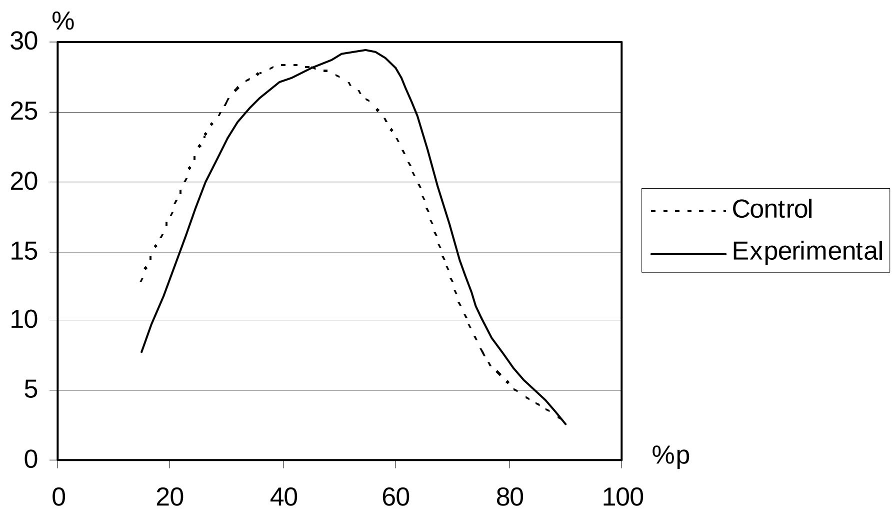


Figure 3. Frequencies of programming self-concept in the post-test

This results show that new, enjoyable technology can cause measurable increase even during a short period. It is very important, because the well-developed self-concept can strongly affect students' academic achievement in the future. The attitudes for teachers are constant ($x_{pre-e}$ = 75.6 %p; $x_{post-e}$ = 76.5 %p; $x_{pre-c}$ = 75.8 %p; $x_{post-c}$ = 77.9 %p; the differences are insignificant), it indicates that the effects are mainly due to the new devices and methods.

## Conclusions

The new devices, such as LEGO robots give a chance for us to build thinking in experimental level to the process of teaching and learning. The knowledge and skill learning being founded on the direct experiences can help in development of deeper level of abstraction.

Introducing a new device usually causes a well-motivated situation. It is necessary to the lasting effect beyond the direct motivation that in order to develop the personality and its motivational base with the help of the achievements and the joyful learning experiences. Our examination indicates that the mobile robots help it already on a short distance (practically beside identical measurable knowledge) to the development the programming self-concept of students and the favorable effect of this may express in the course of the additional learning of the programming.

We're planning to repeat our experiment with a larger sample in the future, and we hope that we can present our new results on a next article.

## References

CSAPÓ, B. (2000): A tantárgyakkal kapcsolatos attitűdök. *Magyar Pedagógia*, 100/3. 343-366.

CSÍKSZENTMIHÁLYI, M. (2001): Flow. Az áramlat. A tökéletes élmény pszichológiája. Akadémiai Kiadó, Budapest.

INHELDER, B. & PIAGET, J. (1984): A gyermek logikájától az ifjú logikájáig. Akadémiai Kiadó, Budapest.

ISTENES, Z. (2004): Learning serious knowledge while playing with robots. In: 6th International Conference on Applied Informatics, Eger.

JÓZSA, K. (2005): A képességek és motívumok kölcsönös fejlesztésének lehetősége. In: Kelemen, E. and Falus, I. (Eds.): Tanulmányok a neveléstudomány köréből. Műszaki Könyvkiadó, Budapest, pp. 283-302.

KISS, R. & PÁSZTOR, A. (2006): Programozható robotok felhasználása a programozás oktatásban. Szakmai Nap, Kecskeméti Főiskola GAMF Kar, Kecskemét.

NAGY, J. (2000): XXI. század és nevelés. Osiris Kiadó, Budapest.

PÁSZTOR, A. (2008): How can IT be Toght Playfully with the Help of Programmable Robots? INTED, Valencia.