SPIKE UP PRIME INTEREST IN MATHEMATICS

Pavel Petrovič

Comenius University Bratislava (SLOVAKIA)

Abstract

One of the main drivers of robotics in education are programmable robotics construction sets. Among them, the long-term popularity in the schools remains with the LEGO robotics sets. Their versatility, universality, and long-lasting quality is now proven by decades. They provide a full-range of robotics experience - from mechanics, through sensor and motor control, to programming and data representation. In fact, we are still regularly using also the sets produced about 15 years ago with both students at our faculty and pupils in elementary school. The sets are often successfully used for introducing robotics, and computational thinking. They are also excellent tools for constructivist and constructionist learning and project-based learning. What is still overlooked and remains for an intensive research and didactic work is making the sets more ready for the interdisciplinary scenarios, which are attracting an increasing attention while the school systems in various countries are still being transformed to better serve the pupils in preparing them for the ever-changing world. What we mean here is using the sets in other subjects, such as Physics, Mathematics, English, Biology, Art, and others. In the previous work, we have studied the ideas of using the new robotics sets in the settings of physics class, and we have shown a set of activities that can be directly taken to a classroom. The focus of this article is to build up on that work and describe a set of exercises for using the sets to teach Mathematics - summarizing our previous experiences, collecting experiences of others, and providing a unifying view.

Keywords: Spike Prime, LEGO robotics sets, mathematics.

1 INTRODUCTION

Robotics sets are a multi-purpose educational tool. We have been using them in various educational scenarios continuously for more than 25 years, starting as a teacher in elementary school in the pre-Mindstorms era, teaching programming 11 years old children with LEGO Dacta ControlLab sets in the 90s. If the potential and role of robotic sets could be summarized in a single sentence, it would be: “This is the kind of school, I like to attend”, uttered by a member of our after-school robot club aged 12, about 12 years ago, who later organized a team of his peers to win national rounds of RoboCup Junior in several countries including European Championship, after they have been individually working towards that goal for several years, see compotes.net, if interested. The potential of the robotics sets thus have been intensely harnessed in events such as FIRST LEGO League, RoboCup Junior, World Robot Olympiad, and many other local contests. In fact, we organize one of the kinds too, named Robot League, see liga.robotika.sk [1], which is a unique creative on-line contest. In addition to starting and being centrally involved in FLL in Slovakia for its first 10 years and starting and still being centrally involved in RoboCup Junior in Slovakia for about 20 years, running various hands-on workshops and being a certified LEGO Education Academy Teacher Trainer, we have always been looking for ways of supporting the teachers to use this technology for the benefit of their students. However, there is an area, where we feel the educational potential of robotics sets have not yet been sufficiently taken advantage of, and those are the cross-disciplinary scenarios of teaching various subjects in classes with the help of this technology. Our long-term goal is to investigate this potential and contribute to its wider exploitation by designing activities that would be ready for direct use by the science teachers. We were happy about a positive feedback on the set of activities for Physics that we designed two years ago, and we are now launching a similar set of activities in the same format meant to be used to enhance the activity in the mathematics classrooms. In the remaining sections of the paper, we provide our point of view on the history of LEGO robotics sets, touch on the philosophy of learning of LEGO Education, review relevant literature of using robotics sets in teaching Mathematics and introduce our own work through five example projects.

ISBN: 978-84-09-42484-9

2 BRIEF HISTORY OF LEGO ROBOTICS SETS

Robotics sets are characterized by the following properties: 1) they allow the child to alter the behaviour of the model by entering or changing its program, 2) they allow building models that perform physical actions in their environment from their own initiative (autonomously), 3) they allow the models to perceive information from its environment using sensors, 4) they are versatile and thus allow building potentially unlimited set of models, a set bounded only by the creativity of the child or a teacher, meaning they are not limited to a single pre-designed purpose or usage scenario. On the contrary, a model of almost anything from real-life can be built using the elements of the construction sets and rung to life by programming it to become interactive and exhibit some model behaviour. Such interactive models are irreplaceable educational tools and environments for exploration and discovery and a very useful platform for boosting both creativity and confidence in STEM subjects. The above properties allow designing educational situations, in which the user must undergo a creative open-ended problem-solving process with unlimited number of correct solutions. They are so unique that no other educational tool has the capacity to match their potential.

The first LEGO products that can be classified into the category of robotics sets began to appear in the 1980s after Seymour Papert, founder of the didactical approach Constructionism and inventor of the LOGO programming language started to cooperate with LEGO educational department. Extending on the established line of LEGO Technic products that was already successfully on the market since 1977, the first systems Technic Control I, and II, and later Control Center, Control Center II, and finally LEGO Dacta Control Lab in 1993 – they all allowed programming physical models on a personal computer or directly, while the models were connected to a PC over an interface module and a wire. [2]. Using the Logo programming language, it was possible to create interactive projects, where the state and information from the real-world could have been visualised on PC screen and which allowed the user to enter commands, input data, or in some other way interact with or steer the behaviour of the model during its active runtime. Already the first TC Logo has supported tail recursion, running dynamically created code, saving and loading pages to disk, printing texts, drawing graphics, keyboard interactive input, and more [3]. Control lab went even further, it supported catching errors (*carefully*, and *errormessage* statements), plotting graphs of data and functions with the option of analysing data in user own code (*datalist* statement), running parallel code (*launch* statement), working with files (*loadtext*, *loaddata*, *loadtools*, *loadshapes*, *loadpict*, and corresponding *savedata*, *saveproject*, *savetext* statements), interaction with mouse, sliders, buttons, picture boxes, background images, and some reflection features (such as *thing* statement) [4]. Logo as being a dialect of LISP programming language allowed easy data manipulation with user own data representations up to the level of artificial intelligence. Control lab was and by today even after 30 years of further development still is the most universal and didactically most sound LEGO robotics solution providing the largest educational potential. Users were writing code in a general-purpose (yet child-friendly and child-centered) programming language running on PC with input/output capabilities and with the possibility to use (program) graphical display of the PC. None of the later products provide that, unless significantly tweaked by an enthusiastic programmer. All products since that era are closed proprietary solutions with limited possibilities and very limited programming languages with weak expressive power. On the benefit side, it makes it possible to use the products for everyone. Yet, the original LEGO Dacta solutions had a low entry-level difficulty and good support for the teacher, student worksheets, example projects. It was not difficult to use it by children in school and had positive effects on their cognitive skills, see [5] for an example of a study, with further references. While being busy with achieving the noble aims of wide acceptance of this technology, producers not only derailed advanced users who were pioneering its progress, but unfortunately misunderstood the main educational philosophy behind these products: exploring creative constructionist projects in an interactive manner with the help of a computer. This philosophy has been lost with the advance of the LEGO MINDSTORMS products and has unfortunately still not been rediscovered.

Since 1998, i.e. the last 24 years, the programmable LEGO robotics sets follow the program-download-run deployment cycle. This includes the RCX (1998), NXT (2006), EV3 (2013) and Spike Prime (2020) – the solutions at the most advanced level, but also WEDO (2009), WEDO2 (2016) and Spike Essentials (2021) – solutions targeted at younger audiences. That means, the user can only write code that first needs to be transferred to a programmable robotics brick, and then it runs on its embedded built-in computer. While this allows designing truly autonomous systems, it makes it much more difficult and impractical to experiment, debug, and play with the features. Some of the systems require or allow the use of an active BlueTooth connection while the code runs/or its control flow is steered from the PC, however, a very limited output-only functionality is available to the programmer at the side of the computer. Namely: playing sounds on the PC (which is required when the hardware itself cannot produce sounds), plotting graphs in predefined graph boxes, and showing text or images in some text or image boxes. No interactive features. In our robot-club we were able to achieve interactivity with the recent Spike Prime sets in so-called "streaming mode", when changes in the code are immediately reflected in the robot's behavior. By editing the code – repeatedly modifying a numeric constant that was checked in a loop, children could steer a pacman robot from the computer keyboard, see [6]. As we can see, the current solutions force the users to apply anti-patterns when more interesting functionality and use cases are attempted. Interestingly, MINDSTORMS Inventor product in "non-educational" product line has an integrated remote control serving this use-case.

Let us explore the family of programming languages that were used for programming these robotics sets after Logo language has been practically abandoned in more details. The first and strong alternative player that supported different hardware platforms and that started already before LEGO MINDSTORMS entered the scene was the RoboLab programming language developed by Tufts CEEO [7]. It is one of very few truly iconographic programming languages (i.e. statements are depicted only by various pictograms), and it is based on LabView. It was also a main inspiration for the official LEGO software for NXT and EV3, before it was (again) abandoned for the same reasons of greater accessibility to general public. The strength of philosophy of LabView, which is an advanced programming language for laboratory instrumentation developed by National Instruments, lies in being based on dataflow as contrasted to control-flow. As such, it is one of the very few ways of software decomposition with high levels of parallelization without blowing a programmer's mind, another one is the paradigm of functional programming – a flashback to Papert's Logo language. A code can be naturally parallelized when there is a guaranteed isolation of data and processes, which is not the case of main-stream programming languages and model. That is going to be a major issue soon as the 80 years old von Neumann architecture will eventually stop influencing the hardware evolution. It is already happening, see [8] for an example. Unfortunately, authors of these robotics languages kept the icons, but almost dropped the data-flow aspect and misused it for traditional control-flow paradigm, resulting in quite impractical language, where it was difficult to remember the semantics of various icons for many users, it was difficult to edit the code, especially when more structured code has been entered, and it was also very difficult to implement the tools themselves, which lead to many bugs and instability problems (sudden crashes of the software leading to loss of all the work were common) in the production version of the official NXT-G software from LEGO. That made the lives of many child developers difficult. However, before the NXT-G (and its successor for LEGO MINDSTORMS EV3) arrived, the first LEGO MINDSTORMS platform: RCX introduced a different paradigm: event-oriented CASE programming tool style language. By events we mean here the changes in the environment that can be detected with the robot sensors. Use of CASE tools (Computer-aided software engineering) has been popular in software engineering in 80s and 90s with the goal of improving software quality [9]. CASE programming tools (lower-CASE) were used to simplify and automate writing of code. Transferring it to the educational programming language scenario, use of such technique made it more difficult to write the code because editing is cumbersome. However, it helped the novice user (which appears to be the mainstream and decision-triggering audience) because 1) they do not remember the language statements, 2) they have difficulties with typing syntactically correct code without making errors, and 3) it may help with the resulting code readability. As we know from the software-engineering practice today, all three problems are successfully tackled by modern IDEs that provide intellisense, but those are probably yet too advanced for designers of educational software. The event-oriented paradigm brought a couple of issues: 1) since the event-triggered pieces of code are launched in parallel, it is difficult to understand the details of the execution model behind it even for an expert programmer (not even for a child), 2) it is in conflict with the event-processing code snippets, which are written in traditional control-flow style and 3) it limits the program to be very simple due to the fact that events are global and not dynamically activated/deactivated. Finally, partially to the influence of other solutions in educational robotics community, and partially due to influence of other solutions in informatics education community, LEGO also converged in their programming languages to a Scratch-like programming language, which is akin to the original Robotics Invention Studio of RCX from 98. The case boxes are now smaller, and the language brings a new concept of global messages, which are another kind of event that can be triggered by the program itself. However, the problem of globality of the events remains one of the major drawbacks of this solution, adding on top of the impractical editing of the code, difficult to understand execution model, poor code modularity, a very limited language power, and of course the inability to write code that runs both on the PC and the robot. We believe the upper part of the room of the so often spelled '*low floor, high ceiling, wide walls*' philosophy [10] has been cut off and the room has been made much narrower recently. On the positive side, the widespread use of Scratch-like programming languages makes it easier for the child to use many different educational tools without having to learn a new language for each of them, and this is probably the decisive winning argument of the struggle.

Another wide-spread trend in the programming languages world, and the robotics sets have not been immune to it neither, is the Python programming language. On one hand, it offers a promise of interactivity – being itself an interactive interpreted language, but unfortunately this (probably the only) positive aspect of Python is completely ignored and the “console” in Spike Prime software serves only the purpose of printing error messages instead of issuing interactive commands. Something that was possible in 1987, when Papert was involved, and something that is not yet possible in 2022. We are still learning how to fly to the Moon again.

A missing paradigm. When discussing robot programming and formalisms, it would be a mistake to omit a well-known formalism, which is also used in software modelling, namely finite-state machines. Their suitability in robot programming scenarios originates from the similarity between robot tasks and the structure of the state machines. When a robot is solving a task, it typically passes through different stages, which are best represented as states. In each state, it responds to various events, i.e. events are local to state, not global! The transitions between the states can lead to loops, and cyclic behavior. A collection of state machines can provide further modularity where individual modules can all have different state or coordinate through a shared state. State machines have been proposed by well-known robotics researchers, such as [11], decades ago, unfortunately, this paradigm is seldom seen in educational robotics platforms, with the exception of some case-studies [12], where other programming formalisms are utilized and forced to behave as state machines. A direct approach could be more useful.

3 LEGO EDUCATION PHILOSOPHY OF LEARNING THROUGH PLAY

The educational branch of LEGO Group has spent significant efforts researching the relationship of *play* and *learning* [13]. Their findings confirm that playing has a central role in the learning process. A playful situation has the following characteristics: it is a joyful, meaningful, actively engaging, iterative, and socially interactive activity. That is in a radical contrast with generic encyclopedic definitions of the word *play*, which unfortunately often pertain in the common sense, for instance: “*to exercise or employ oneself in diversion, amusement, or recreation*; *to do something in sport that is not to be taken seriously.*” (dictionary.com). Instead, LEGO believes in play as the core activity that enables learning through a process that Papert called *hard fun*. [14], a process when a child is put in an engaging structured environment, formulating and solving creative challenges, making discoveries on its own while enjoying the process as being entertaining. We can understand the success of this approach when we understand the motivations that drive the child activity, which are the desire to satisfy her or his goals and ambitions. And that is exactly what learning should look like – voluntarily obtaining skills and knowledge one needs for life, while being aware why are they useful and interesting for the learning individual.

LEGO Education provides its solutions with a good support for teachers – preparing series of activities and materials, smaller and larger projects that can be used by teachers directly in the education process. Older products used materials organized according to their 4C method (Connect-Construct-Contemplate-Continue) [15], while the newer products adopted a more known 5E model of instruction (Engage-Explore-Explain-Elaborate-Evaluate) [16]. We have also followed this model and arranged the on-line learning materials that are described in this paper in a similar manner.

4 MAIN GOALS

We have prepared a set of constructivist project activities that we describe in this article with the aim of providing teachers with material that they can use in different scenarios when teaching Mathematics. The perspective is the one of interdisciplinary, cross-subject use. We would like the teachers to bring the robotic sets to other subjects, not only informatics, robotics, or technology as is the most common practice. We would like to see the teachers of Physics [17] and now also of Mathematics to take the sets to their classes, run the hands-on workshops with their students, assign individual group work and perform the experiments themselves to demonstrate the science concepts to their students. The main goal is to let the students collect actual practical experience, which is very important for the learning process to establish analogies and link the theoretical material to personal practical experiences of students.

1 Workshop. The first scenario is a hands-on workshop. Students work in pairs or small groups, build the models, run the experiments with the prepared programs, perform measurements, and elaborate further by modifying, exploring, and trying to answer and solve follow-up questions and challenges. Finally, they present, share and compare their findings and experiences with

other groups. For this scenario to work, the teacher needs to have access to sufficient number of robotic sets. Then he or she needs to select an activity that suits his or her pedagogical goal in the course he or she teaches. One activity can be explored in a one-hour or ideally two-hour workshop.

2 Individual group work. Under these settings, teacher might have only one robotics set available, perhaps one that is borrowed from another place, or one that he or she acquired through some campaign or competition. He or she identifies suitable and motivated small group of students, makes the robotic set available to this group in the afternoon, after the regular lessons, or even for home activity. The task for the students is to build the model and perform one of the activities on their own. It may take a single afternoon, it may take a week, and they may need to search for additional information to be able to complete the tasks. It becomes a little research project for that group. Finally, they present to share their findings in a larger group, or a class, show the model and its functionality and the concepts it can demonstrate.

3 Presentation. In this last way of using the math activities, we propose that the teacher brings the model that he or she has built (this task can be delegated to a group of interested students – now it only requires building model according to building instructions). With the help of students in the class, the teacher can run the experiments on that single model, perform the measurements together with the class, while allowing the individual selected students to "touch the model" and perform hands-on activities in front of the whole class. This provides a wonderful platform for a common discussion, explanation, and brainstorming to facilitate constructivist learning. During the process, the class can be divided into smaller discussion groups to discuss the presented phenomena by each student with its peers and compare against the results demonstrated with a real model after or before that.

5 RELATED WORK

Mathematics as a science is a very specialized area of focus of a few brilliant minds, whereas mathematics as a tool is present in everyday life in all areas of systematic human activity. That applies to robots as well: mathematics can be applied and is required in robot design and construction, programming, action, sensing, planning, navigating, reasoning, etc. However, to teach mathematics with robots we have to carefully select those applications that will truly improve the learner's experience. Some studies focus on a particular aspect of learning or mathematics and investigate whether using robots in the curriculum has any influence on the learning outcome, and on a particular kind of student performance, for example [18]. Such studies may require long schedules, large resources, lengthy administrative work with evaluation of the findings, and at the end of the day, their results could be questioned: is it always impossible to separate that particular issue that is subject of a study from other factors, such as the personality of the teacher, temporal motivation of the students that can be influenced by a trend of a particular season, or a temporal novelty of certain technology? Differences may also be an outcome of the actual efforts that are put into teaching the children (more efforts are likely to bring better outcome). On the other hand, negative outcome can be the result of using topics and activities that are not carefully selected, become confusing to students, may be boring or difficult to grasp for the students. An example of such evaluation with negative outcome is in [19]. Authors analysed a robotics curriculum that spread over 9 weeks and had the aim to teach mathematical concepts in the 8th grade of public middle school with 99% minority and 94% economically disadvantaged students. They report that "so many different topics were raised that it was most likely difficult for students to learn any one of them well". A solution could be to prepare activities that give an intense focus on single or very few concepts each. In another study of [20], researchers used participant observation study with a group of almost 50 7th grade students taking a course where LEGO artefacts were used as a tool for learning of mathematics and science concepts through technology practice. Their findings suggest that 1) Construction activities give the potential for students to achieve many learning outcomes in science, mathematics and technology, 2) Scaffolding in the form of explicit details was needed for some students to make links between activities and outcomes, 3) some students achieved higher levels than expected, 4) Student manifestation of science and mathematics outcomes were directly related to the technology activities. Our point of view is that it is not interesting anymore to ask "whether". Possible positive effects of using technology in learning process has been confirmed multiple times, see for example [21]. An important aspect is the social one: working with robots encourages cooperation and groupwork and gives possibilities for mentoring. Authors in [22] focus on collaborative learning in middle school and write "the key benefit of the project being the improvement in the students' ability to work collaboratively". However, it is interesting and important to discuss "how". Serious work of this kind is taken in the group of [23]. We suggest the discussion to concentrate around reasoning and argumentation over actual substrate: complete, well documented, and independent modules that can be taken straight to the teaching practice, actual qualitative evaluation of experiences and iterative improvement of the existing modules based on them. Instead of providing a complete curriculum to teach mathematics using robotics, we believe in individual engaging project activities potential of generating the state of flow, eureka effect, interesting discussions. They role is to enrich a more standard pedagogical process.

6 FRAMEWORK OF CREATIVE CONSTRUCTIONIST PROJECTS

All our activities share a common form of presentation. Each activity contains a model to be built of varying complexity. Some models can be built in 5 minutes, others may require at least 30 minutes. The parts of a single LEGO EDUCATION Spike Prime set should be sufficient to build any of the models. All builds were modelled also in the stud.io 3D modelling software, and therefore in addition to photographs of the built model, each activity shows also a computer rendered model, downloadable stud.io model for detailed analysis or further improvements, carefully prepared step-by-step building instructions in PNG and PDF formats, program that demonstrates some concept from mathematics with the help of that model, one or more videos that show its functionality and usage, possible other materials, such as charts, data or image files that may be useful in running the experiments, and the 5E guidelines for running the activity – with ideas for elaboration and evaluation phases.

7 OVERVIEW OF MATHEMATICS PROJECTS

During the years of working with the robotics sets in various settings – in many public workshops, regular after-school robot club, intensive summer schools, occasional visits in the class or competition preparation situations in elementary or secondary schools, and various robotics courses at the college level, we have collected various project ideas for experiments with robots and mathematics. They include activities of a very wide range – from very first years of elementary school up to advanced concepts in secondary school. We believe that by including them in a common resource, we maximize the potential for creative inspiration, encourage cooperation of students of different ages and levels and their teachers as well. The activities are meant to by dynamically evolving, and the resource is welcoming external contributions.

8 DETAILED DESCRIPTION OF EXAMPLE PROJECTS

In this section, we present five selected projects in greater detail for the purpose of demonstrating the way they are prepared. In addition to the activities selected here, the set of activities we designed includes several other activities.

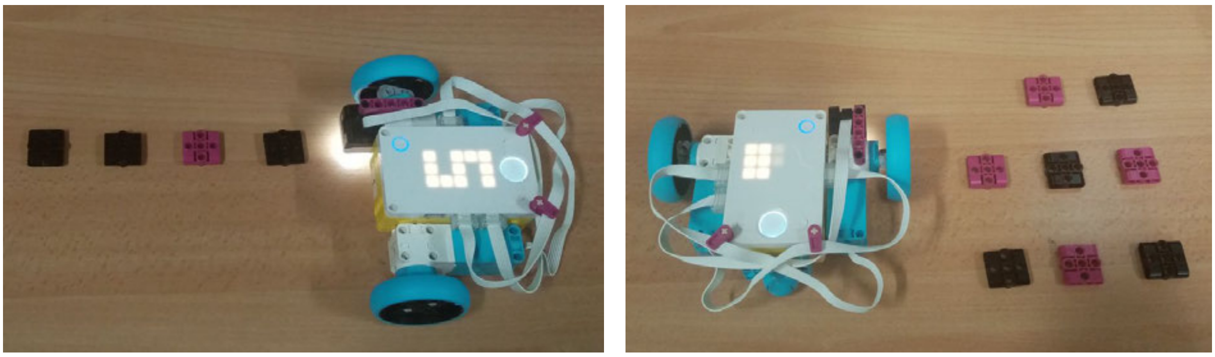

Figure 1. A calculator robot performing addition (left) and division (right).

8.1 Calculator

The first activity is to help the students to become confident about basic arithmetical operations – addition, subtraction, multiplication, division, and reminder and to have some fun performing these operations and learning about the numbers. The robot is counting bricks – having an "accumulator" that can be either added to or subtracted from, visualizing the number using its matrix display – either as a decimal number or as the number of illuminated pixels. The activities either consider both kinds of biscuits – black and purple as a positive token, and let the user decide upon the operation using the arrow buttons on the robot, or, alternately, the colour can give a semantics of addition and subtraction. The experiments with multiplication and division assume the items are arranged in a rectangular shape, and they are added layer-by-layer to demonstrate that multiplication is formed of a series of additions. Remainder is visualized with pixels illuminated in lower intensity. The resulting programs are quite complex, containing a couple of hundreds of blocks, but older students (grade 5-9) could be able to modify them to recognize more colours, or perform other operations.

8.2 Ellipse

In this exercise a robot is drawing ellipses using the formal definition of ellipse as a set of points that have a constant sum of distances from the two focal points. This exercise serves several purposes: 1) to understand the ellipse definition and the role of focal points and their distance, 2) to understand a difference and relation between circles and ellipses, and 3) to explore the options of expressing some notion of ellipse eccentricity. The provided program uses a certain method to calculate a numerical value somewhat related to the ellipse eccentricity. The task for the students is to come up with different ideas how this concept can be quantified and possibly measured by the robot.

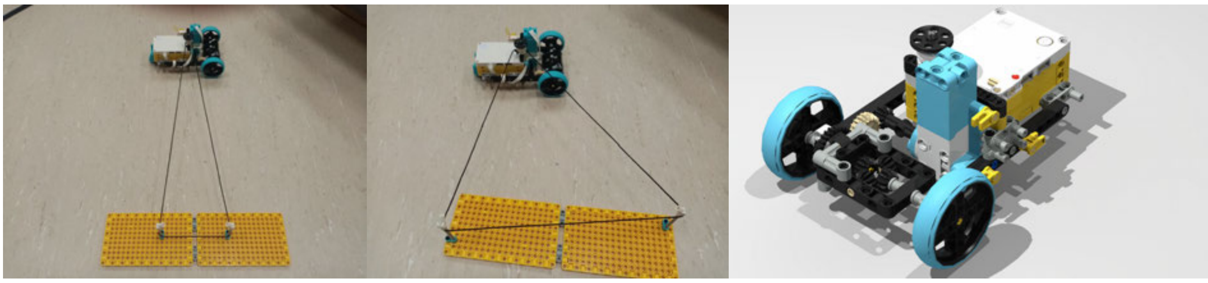

Figure 2. Ellipse drawing robot with different focal distances (left, centre), and rendered model (right).

8.3 Intersection

A robot is placed in front of two line segments marked by a black tape on the floor. The line segments are not parallel, but do not intersect. The task for the robot is to navigate to the virtual intersection point of the lines extending the segments and unload a cargo – for example a LEGO figure at this location. In this activity, we demonstrate the functionality of the program to the students and let them try to analyse what has happened. How could the robot figure out where to drive and stop and unload the cargo. They can propose their own solutions to this problem.

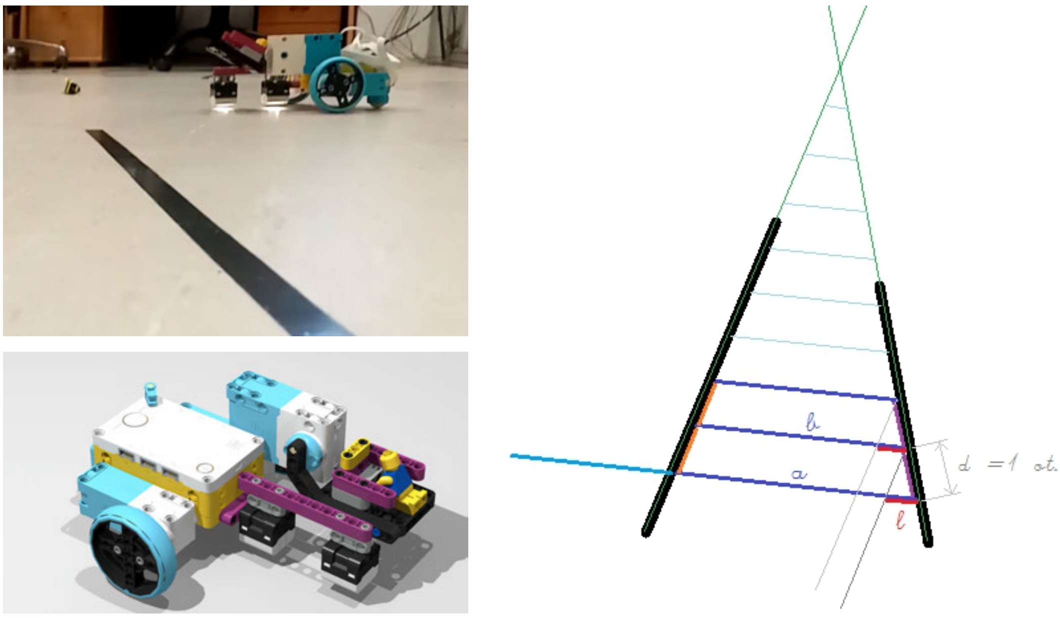

Figure 3. Situation after unloading a figure (top left), rendered model (bottom left), and analysis of the solution to this problem (right). We invite the reader to enjoy this analysis with his or her own thoughts.

8.4 Learning Robot

One of the central problems of AI robotics is how robots can learn. For this objective, it is important to understand the role of knowledge representation and the role of a learning rule. In this activity, students work with a simple learning robot. It can learn how to act in various situations, which leads to learning different behaviours – without the necessity of modifying a program. The actual mathematical problem in this task lies in efficient way of translating the sensory reading into an index in the behavioural lookup table that the robot is learning. One (but not the only one) way is to use the binary number system, encode the discretized sensory readings as binary digits that together form the index of the row in behaviour describing table that is learned in the learning stage and used for action selection in the run stage. Students in this activity are presented with the problem of calculating this row index. They can either be asked to invent their own formula, requested to understand and analyse how the formula works in the presented solution, or elaborate on the presented solution and improve it for more detailed sensitivity of the sensors.

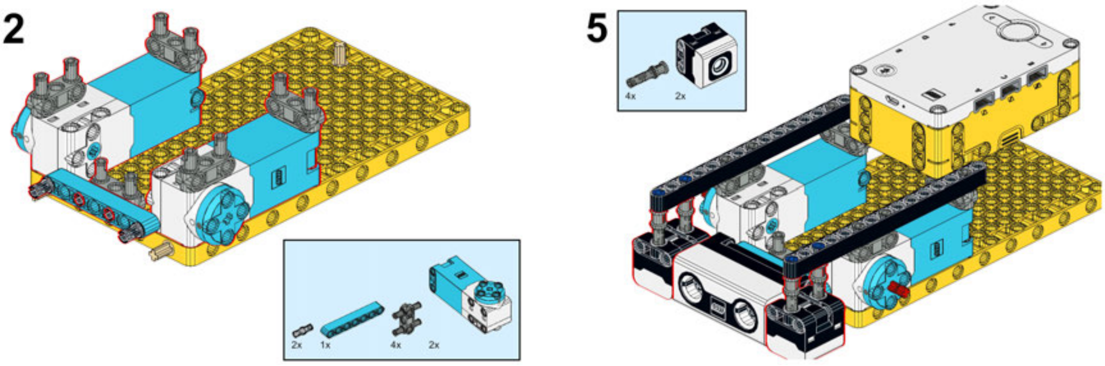

Figure 4. Two example steps from the building instructions generated with the stud.io software.

8.5 Entrance Lock

In a similar, yet completely different activity at the same time students are facing the problem of representing numerical information in space, or more precisely the problem of utilizing space to represent numbers. The scenario is a usual hotel-room entrance lock with a credit card slot and a bar code printed on a paper that is taped over a standard-size credit card. The machine can read the code and interpret the bar code as numbers. Students are encouraged to discuss various ways of encoding numeric information in bar codes. They can even go as far as to study the versions of the international standard of bar coding, which are publicly available, but that would require a camera hardware to be recognized. Since reading a delicate code using the sensor of reflected light is something that can depend on the toner and paper reflectance and possibly some external ambient illumination, this activity may require a calibration process. That involves scanning an example card first to find a proper level at which the width of the narrow and wider black stripes can be clearly separated. In the next step of calibration, the card can be scanned using this tuned intensity level as monochromatic pattern with gaps of different length. A distinguishing length of a gap can be found and fed into the final program that then performs more reliable card pattern scanning.

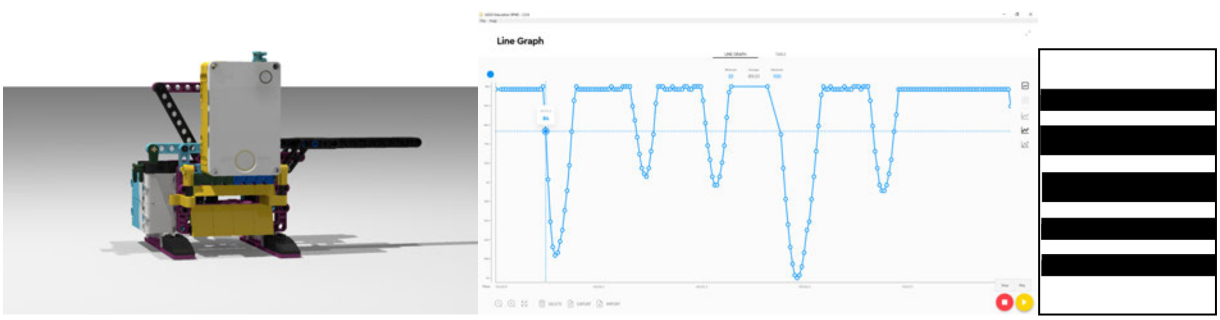

Figure 5. Rendered model (left), graphical output from the calibration process (centre) and example pattern to fasten on top of a plastic credit card (right).

9 INSTEAD OF EVALUATION

Most of the activities we have included in this platform have been tested in a Robot League competition [1] throughout its 10 years history. Some of our solutions are inspired by the solutions of the teams where we worked in the role of a team leader. While the scenario in Robot League is somewhat different – teams are given a task and their goal is to find solution, whereas teachers using our activities in their lessons already have the solutions available. Nevertheless, the fact that many of the solutions were found by the teams suggest that 1) they were proven to work in other settings than our own, 2) they can be conceptually understood by children. The age of participants in the Robot League is 9 – 16 years. Finally, we list here the tasks of the Robot League that are related to mathematics. Typically, several solutions or solution attempts can be found underneath each of the tasks. They are all available through archive at webpage liga.robotika.sk.

Table 1. Math-related tasks in Robot League competition.

Task name	Year/ round	Area	Task name	Year/ round	Area
Calculator	2014/4	Arithmetics	Geometry advanced	2019/8	Geometry
Secret treasure	2014/7	algebra	Antikythera	2020/7	Arithmetics
Second hand	2016/5	fractions	The slowest movement	2020/8	Fractions
Builder	2016/7	Geometry	ΕΥΡΗΚΑ! num=Δ+Δ+Δ	2021/1	Algebra
Information transfer	2017/3	Algebra	Roulette	2021/2	Algebra
Spirals	2017/7	Geometry	Persistent message	2021/3	Advanced algebra
Tax office	2018/4	Algebra	Ellipse	2022/5	Geometry

10 CONCLUSIONS

Building on top of our previous work where we designed a set of activities for using robotics sets in or out of classroom settings for teaching Physics, we have developed in this work a series of activities that demonstrate concepts from Mathematics in a similar format. Each such activity is arranged in a 5E instructional model, providing motivation, a task, building instructions, ready to use programs, photo and video documentation of example solutions, editable 3D model, ideas for further experimentation, and optionally also supplementary information or files. As such, it is ready for a teacher to adopt it and simply use it in the lesson. Most of the tasks were adopted from tasks given in Robot League contest we have been organizing for 10 years and that in total contains more than 100 educational tasks with creative solutions of different teams. The tasks were redesigned for the new LEGO Education Spike Prime platform so that they can be built using one robotics set and adjusted to fit in our framework. Children successfully found solutions to the tasks, which gives us positive clues that they could be used for demonstration of the concepts in mathematics. We plan to use them in this year summer school in elementary school near Bratislava with which we have a long-term cooperation. The activities are available at the webpage robotika.sk/spike.

REFERENCES

[1] R. Balogh, P. Petrovič, “Robot League – A Unique On-Line Robotics Competition,” in: *Robotics in Education. RiE 2019. Advances in Intelligent Systems and Computing*, vol. 1023, Springer, 2020.

[2] M. Hocker, The Brothers Brick LLC, *A History of LEGO Education*, 2020. Retrieved from https://www.brothers-brick.com/2020/01/14/a-history-of-lego-education-part-1-strong-foundations-feature/

[3] *LEGO TC logo Reference Guide, LEGO Dacta, 2nd revision*, Logo Computer Systems, 1989.

[4] *LEGO DACTA ControlLAB Reference Guide*, LEGO Group, 1993.

[5] J. Lindh and T. Holgersson: “Does lego training stimulate pupils’ ability to solve logical problems?” *Computers and Education,* vol. 49, pp. 1097-1111, Elsevier, 2007.

[6] Šachisti – tigre: Robot League, *PAC-MAN*, solution to 4th round of Robot League 2022. Retrieved from: https://liga.robotika.sk/?page=solution&id-assignment=90&id-team=226

[7] B. Erwin, M. Cyr and C. Rogers, "LEGO Engineer and RoboLab: Teaching Engineering with LabView from Kindergarten to Graduate School", *International Journal of Engineering*, vol. 16, No. 3, pp. 181-192, 2000.

[8] S. Cass, "Taking AI to the edge: Google's TPU now comes in a maker-friendly package," in *IEEE Spectrum*, vol. 56, no. 5, pp. 16-17, May 2019

[9] G. Premkumar, Michael Potter, "Adoption of computer aided software engineering (CASE) technology: an innovation adoption perspective", *Advances in Information Systems*, vol. 26, no. 2-3, 1995.

[10] M. Resnick, Medium.com, *Designing for Wide Walls*, 2020. (Originally published in Design.blog in August 2016). Retrieved from: https://mres.medium.com/designing-for-wide-walls-323bdb4e7277

[11] R. A. Brooks, "A robot that walks; emergent behaviors from a carefully evolved network," in *Proceedings, 1989 International Conference on Robotics and Automation*, pp. 692-4+2 vol.2, 1989.

[12] R. Balogh, D. Obdržálek, "Using Finite State Machines in Introductory Robotics", in: *Robotics in Education. RiE 2018. Advances in Intelligent Systems and Computing*, vol 829. Springer, 2019.

[13] R. Parker and B. S. Thomsen, *Learning through play at school*, The LEGO Foundation, 2019.

[14] S. Papert*, Hard fun*, Article for the Bangor Daily News, 2002. Retrieved from: www.papert.org/articles/HardFun.html

[15] LEGO Education Academy, *Planning Your Lesson*, Retrieved from: https://education.lego.com/en-us/academy-training/planning-your-lesson, Retrieved on: May 1st 2022.

[16] R. W. Bybee, *The BSCS 5E Instructional Model: Creating Teachable Moments*, Corwin Press, 2016.

[17] P. Petrovič, "Spike up Prime Interest in Physics", in *International Conference on Robotics in Education (RiE) 2020*, pp. 146-160. Springer, 2020.

[18] Casler-Failing, L. Shelli, "The Effects of Integrating LEGO Robotics into a Mathematics Curriculum to Promote the Development of Proportional Reasoning", in *Proceedings of the Interdisciplinary STEM Teaching and Learning Conference*, vol. 2, article 5, 2017.

[19] E. M. Silk, and C. D. Schunn, "Using robotics to teach mathematics: Analysis of a curriculum designed and implemented", in *2008 Annual Conference & Exposition*, pp. 13.1353.1-15, 2008.

[20] S. J. Norton, "Using Lego to integrate mathematics and science in an outcomes based syllabus", in *AARE Annual Conference*, 2004.

[21] C. K. Bakke, *Perceptions of professional and educational skills learning opportunities made available through K-12 robotics programming*, doctoral dissertation, Capella University, 2013.

[22] G. Ardito, P. Mosley, and L. Scollins, "We, robot: Using robotics to promote collaborative and mathematics learning in a middle school classroom", Middle Grades Research Journal, vol. 9, no. 3, 2014.

[23] E. Cejka, C. Rogers, and M. Portsmore, "Kindergarten robotics: Using robotics to motivate math, science, and engineering literacy in elementary school", *International Journal of Engineering Education*, vol. 22, no. 4, pp. 711, 2006.