Let's have a look behind the code

The Big Mathematics Day 2016 (Netherlands) about coding without computer.

versie 8-5-2016

Mieke Abels, Vincent Jonker, Ronald Keijzer & Monica Wijers

Utrecht University, Freudenthal Institute

Princetonplein 5

3584 CC Utrecht

the Netherlands

m.j.abels@uu.nl, v.jonker@uu.nl / +31620416766, r.keijzer@uu.nl, m.wijers@uu.nl




Abstract

Students in Dutch primary schools spend quite some time working on mathematics. The average lessons however are limited to relatively short interactive introductions of new content; the rest of the time pupils spend on paper and pencil work. There is little time for mathematical reasoning and problem solving that inspires both students and teachers. In 2004 the Freudenthal Instutute of Utrecht University started an annual event (the 'Grote Rekendag', the Big Mathematics Day) to promote inquiry learning in the mathematics lessons of primary school. In 2016 we organized this event with the theme "Let's have a look behind the code", a theme inspired by the activities from 'CS unplugged' and by other educational ideas and the upcoming interest in coding and programming for young children. Using interviews (pupils age 9-12 and teachers) and a questionnaire we investigated what pupils and teachers liked about the theme and the activities and what they think they learned from these. Results show that teachers and pupils liked the activities. Teachers indicated that their pupils learned about coding and procedures, and less about how a computer works.

Keywords

computer science; inquiry learning; primary education; mathematics education

Theoretical background

The Big Mathematics Day (www.fi.uu.nl/en/wiki/index.php/Big_Mathematics_Day) is a whole day event for primary schools based on the view of 'Mathematics as a human activity' and the approach of 'Realistic Mathematics Education' as the central pedagogical and didactical concepts.

Realistic Mathematics Education, or RME, is the Dutch answer to the world-wide felt need to reform the teaching and learning of mathematics (Van den Heuvel-Panhuizen, 2000; Gravemeijer, 1994). The roots of the Dutch reform movement go back to the early seventies when the first ideas for RME were conceptualized. It was a reaction to both the American "New Math" movement that was likely to flood our country in those days, and to the then prevailing Dutch approach to mathematics education, which often is labeled as "mechanistic mathematics education."

In RME mathematics is seen as a human activity: pupils guided by the teacher re-invent and construct mathematical concepts, tools and ideas (Freudenthal, 1991). Problem solving, mathematical thinking, reasoning and communicating are core activities. Another aspect of RME is the intertwining of learning strands, not only within mathematics but also between mathematics and science and technology. See also the related concepts of mathematical literacy (Jablonka, 2003), techno-mathematical literacy (Hoyles et al, 2003) and scientific literacy (De Jong et al, 2001)

In the so called 21st century skills documents (Trilling & Fadel, 2009; Edens et al, 2010) emphasis is placed on providing pupils with a new set of competencies – besides foundational skills - that will enable them to adapt to an ever-changing environment (Gresnigt et al, 2014). These include analytical and problem-solving skills, communications skills, interpersonal and collaborative skills, global awareness, and financial, technological (Cunningham, 2009) and civic literacy.

In recent years more and more value is placed on computer related skills for everyone. Knowing how a computer works, basic understanding of (computer) coding, computational thinking and learning the basics of programming, according to this view should be part of the curricula, starting in primary school at an early age (Mishra & Koehler, 2006; Kuhlemeier & Hemker, 2007). These skills together are part of what is nowadays labelled as *digital literacy*. According to Wikipedia (en.wikipedia.org/wiki/Digital_literacy) a digitally literate person will possess a range of digital skills, knowledge of the basic principles of computing devices, skills in using computer networks, an ability to engage in online communities and social networks while adhering to behavioural protocols, be able to find, capture and evaluate information, an understanding of the societal issues raised by digital technologies (such as big data), and possess critical thinking skills. Gui & Argentin (2011), considered digital skills not only in terms of actual know-how but also as a measure of the awareness of the technical and logical structures beneath digital environments.

On a worldwide scale more and more classroom activities and materials are being designed to implement digital literacy in (primary) education (Libow Martinez, 2014). The materials of Computer Science unplugged (http://csunplugged.org) show that a lot of the aforementioned skills can be also learned without a computer. The activities introduce students to Computational Thinking through concepts such as binary numbers, algorithms and data compression, without having to use computers or programming.

Inspired by these ideas we combined mathematical thinking as part of mathematical literacy and computational thinking as part of digital literacy, to design a whole day of activities for primary school students, with both emphasis on having good classroom activities and good support for the teacher (a manual that was sent to all schools about 5 weeks prior to the Big Mathematics Day 2016).

The activities

We designed activities for the Big Mathematics Day 2016 in which students (grades 4 to 6, age group 9 to 11) 'invent' and inquire how they can instruct machines (computers and robots) to carry out specific tasks. The emphasis is on the concepts behind coding, rather than staritng with coding on the computer. In most tasks students do activities like ordering, planning, sorting and (de)coding. In all cases they try (and learn) to think in steps that a computer would take, and use and find ways to describe these steps using symbols, schemes, patterns and structures. In this respect mathematical and digital literacy are almost similar. Most activities can be used in different grades, albeit on different levels.

We discuss four typical activities that all deal with coding and we illustrate these with exemplary work from pupils.

1. Colour by Numbers (*no computer*)
2. Live Turtle (*no computer*)
3. Coding your Pin (*no computer*)
4. Building with blocks (*computer*)

Activity 1 - Colour By Numbers

In this paper and pencil activity pupils explore how images are displayed and coded, based on the pixel as the building block (see Figure 1). In particular, the great quantity of data in an image means that we need to use compression to be able to store and transmit it efficiently.

The representations in the grid use numbers to indicate which pixels are turned on and which are turned off (black or white). There are two different versions: one where each pixel is coded individually using 0 or 1 and one using codes for ranges of pixels, like 1-3-1 alternating white and black (see Figure 1).

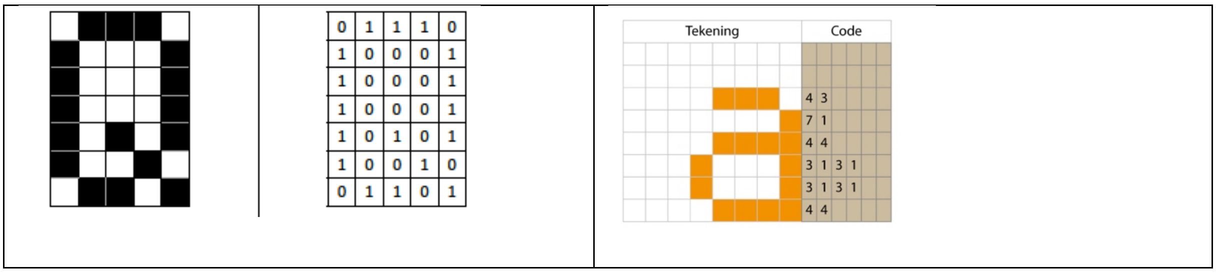

Figure 1. Two ways of coding

Students explore different ways of coding, and invent new ways to shorten codes or to include colouring. They design drawings, code them and check in pairs if their codes work out well,

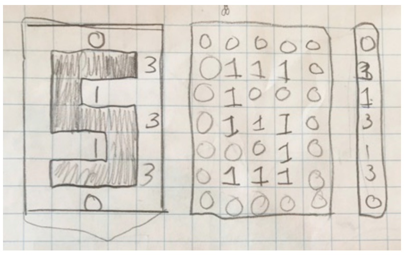

Figure 2. Student work of the activity Colour by Numbers

The shortenend way of coding invented by this student (Figure 2) is not really ‘working’. The student may not yet have fully understood what is essential, When you condsider the code in the array of numbers on the right (0,3,1,3,1,3,0) it is impossible to get to the 'S' as the decoded result. This array only holds the number of pixels you need to color in each row; what is missing for each row is the starting point of the range of colored pixels. By exploring their own coding systems, students discover what are essential characteristics to make the coding work. The example presented here comes from: csunplugged.org -> image-representation. Similar activities are published as logical puzzles in journals or on webiste.

Activity 2 - Live Turtle

This activity, also done without the use of a computer, is the 'embodied' version of turtle logo 'programming language' (Papert, 2003). This activity is presented in two versions: one for the lower grades and one for the upper grades. In the lower grades pupils guide a robot through a labyrinth on paper using arrow-codes. In the upper grades pupils work in pairs on their own designs. One pupil draws a simple shape and writes a program coding it using commands like 'turn 90 degrees, walk 10 steps', etc.). The other child ‘runs’ the program by performing each 'step', one step at the time, like a robot on the schoolyard (Figure 3). If this is done correctly the drawing of the first pupil appears again.

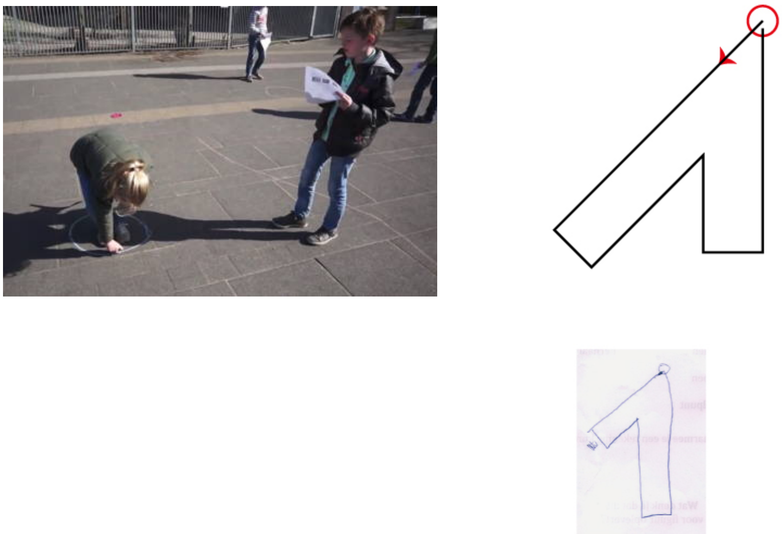

Figure 3. Students working together in the Live Turtle activity and their drawing

Even for students in the upper grades this turned out to be a difficult activity. They often drew seemingly simple shapes (Figure 3), that turned out to be complicated to code. Especially the commands for making turns were hard for them. This can be understood if we realize that to draw an angle of 45 degrees, as in the upperpart of the shape in figure 3, the command is not ‘turn 45 degrees, but ‘turn 135 degrees’. The outer angle is the ‘turn-angle’.

Acrivity 3 – Coding your PIN

This activity for the higher grades, draws on the use of binary numbers. A square divided in 8 segments is presented. Each segment in the lower left half of the square represents the number 1, 2, 4 or 8 (see Figure 4). By colouring the appropriate segments each digit 0-9 can be represented as a pattern. The lower left square in figure 4 is coded to represent the number 5. Colouring segments in the upper right half of the square is used only to generate a nice pattern and cause confusion in order to make decoding harder.

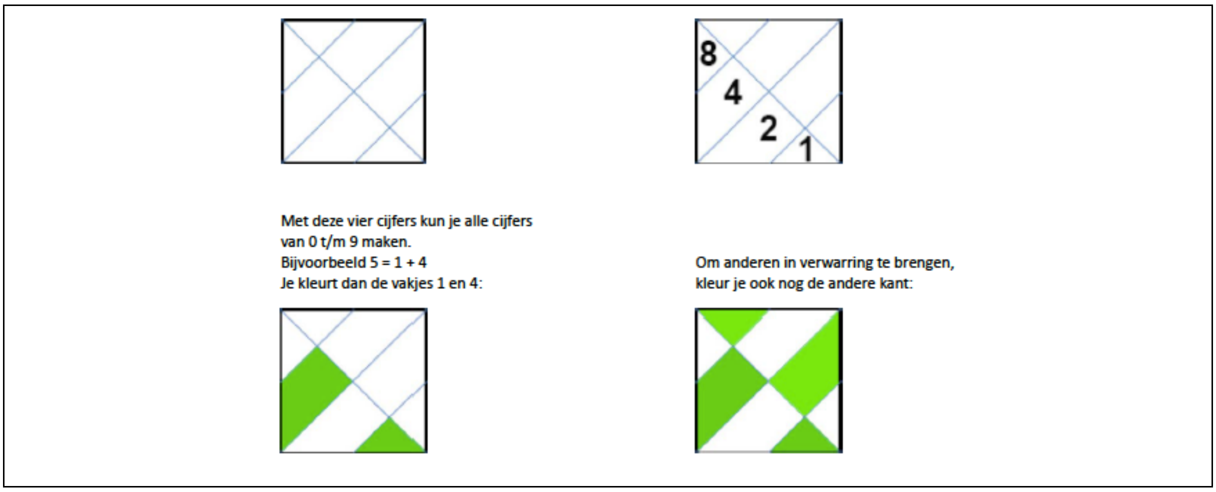

Figure 4. Instructions and example for coding a PIN

Students use four squares to code a PIN. This activity was one of the favourites, partly because students could use creativity as well as thinking and reasoning. Of course students also had to explore this new way of coding. This also means they had to find out how and why they needed only four segments - numbered 1, 2, 4 and 8 - to code all digits. Pupils also reasoned about other numbers that could be coded with 1, 2, 4 and 8 and they found out how to extend this way of coding by adding one or more extra segments, in order to extend the range of numbers to code.

Activity 4 - Building with Blocks

This activity (with a computer) is based on the popular small application 'building with blocks' used in many classrooms for primary education in the Netherlands (www.fi.uu.nl/toepassingen/28432/). In this version pupils can 'automate' the building by programming it. They can create their own programs using commands with 'coordinates' to build exciting shapes on the computer (Figure 5).

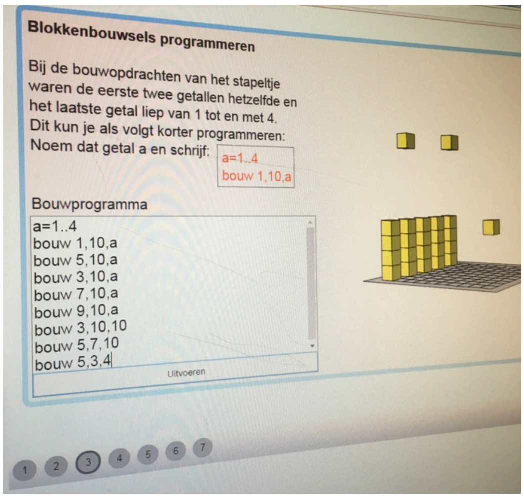

Figure 5. Student creating a 'building' by coding

In the example (Figure 5) the student is discovering the role of a variable/parameter, something new to most children age 11-12. The interface is easy and there is immediate feedback on the screen.

Methodology

We observed in three different schools participating in the Big Mathematics Day (a researcher was present during the activities), and interviewed both students and teachers in order to get an idea what they understood and liked about the tasks.

We also designed a questionnaire for teachers with the following questions (among others).

Table 1 - Questions used from the online questionnaire of the Big Mathematics Day 2016

	Question	*Type*
1	How many students in school?	open
2	Rate the activitiy in grades 5-6 (age 11-12), with a number 1 to 10	rate 1-10
3	Students have learned about procedures (rate 1 to 10)	rate 1-10
4	Students have learned about coding (rate 1 to 10)	rate 1-10
5	Students have learned about how computer work (rate 1 to 10)	rate 1-10
6	Overall impression (What is your opinion about the day, and the activities)	rate 1-10

For the questions 2 to 6 there was also the possibility to react in an open field.

Results

Students enjoy to work with the activities from the Big Mathematics Day, and get a better understanding of how computer programs are responsible for subsequent activities in a task. Teachers have difficulties in supporting their students because the content of the tasks is also new to them, and they try to find a different, more open and supporting approach to guide the children in their discoveries. Let's have a closer look at the data that came from the online questionnaire (N = 293 schools, see Table 2) and the observations/interviews in the three schools that were visited by researchers.

Table 2 - Data from the online questionnaire of the Big Mathematics Day 2016

	Results (N=293 schools)	*N*	*Average*	*SD*
1	How many students in school?	278	242	135
2	Rate the activity in grades 5-6 (age 11-12), with a number 1 to 10	270	7,8	0,9
3	Students have learned about procedures (rate 1 to 10)	284	7,4	1,1
4	Students have learned about coding (rate 1 to 10)	286	7,8	1,0
5	Students have learned about how computer work (rate 1 to 10)	270	7,0	1,7
6	Overall impression (What is your opinion about the day, and the activities)	289	7,9	1,0

Dutch schools for primary education differ in size (Question 1, with an average of 242 students per school, standard deviation of 135). For this research it is simply a fact that during the Big Mathematics Day 2016 about 65 thousand students (age 6 to 12) participated in the activities (about 17 thousand students age 11-12).

The overall impression of the Big Mathematics Day is that teachers (and students) enjoyed the activitivies of the Big Mathematics Day (Question 2, score 7,8). This is in line with the previous Days that were organized (from 2004 onwards).

Students have learned about procedures (Question 3). Some reactions (from the open field of Question 3):

- "It was good to see that children work together and then learn how to solve the posed problems following the procedures"
- "The 'smart' children (children that like new problems) are better prepared for this kind of activity. Especially when the activities are completely new to them."

Students have learned about coding (Question 4). Some reactions (from the open field of Question 4):

- "They have learned what coding is and especially designing your own code is a strong approach."
- "Sometimes the students were quick in discovering and explaining to each other how coding works"
- "Some tasks were really difficult for the children"

Students have learned about how computers work (Question 5). This is a little lower than Question 3 and 4, but still a good score. Some reactions (from the open field of Question 5):

- Children nowadays do have more devices to work with (laptop, tablet, phone) so they already have important experiences
- The activities point at 'how the computer work' and that is enough for this setting

In the analysis of Questions 4 and 5 we saw different responses from teachers where they point out that the activities were 'too difficult'. In the interviews we found that this observation is a mix of what the students gave back as a response and the behavior of the teacher. Some of the teachers are less involved in 'more scientific subjects' and they also have difficulties with the 'more open structure' of the didactics (inquiry learning). More 'exploration space' and discussion and interaction for the students means that the teacher sometimes only has to follow the findings of the students and to support them and give additional feedback, and this approach can differ in subject and quality.

This last observation (sometimes it is too difficult for student and teacher) is part of the whole approach of the Big Mathematics Day. It is also meant as a source for new didactical approaches and for new content for challenging and engaging lessons.

Question 6 gives the general feedback that teachers were really involved in the activities of this day, by scoring quite high for the 'overall impression'.

Conclusions

The approach described (coding activities for students age 9-12, during a whole morning, mainly without computer) gives a good introduction for learning about coding and understanding procedures and rules. For most teachers it meant a first step in their lessons in the area of computers and coding, and of course this activity must get a follow-up in a wide range of other activities (some materials of the Big Mathematics Day are published by the Utrecht University in the online repository of classroom materials for STEM, www.freudenthal.nl -> english, and we see a little rise in the amount of users of this kind of materials).

An important issue to be discussed with teachers but also with teacher trainers is the question if extra attention to (computer) coding must be given in the mathematics lesson. With this example of the Big Mathematics Day we hope to have given an example of how you can make a connection between mathematics education and coding. This approach is only going to work if this is also part of the textbooks used in primary education.

This approach of inquiry learning in the area of coding must get a follow-up in teacher training (for both new and experienced teachers). In the Netherlands this will get more attention in the next few years.

References

Cunningham, C. M. (2009). Engineering is elementary. *The Bridge. Linking Engineering and Society, 39*(3), 11-18.

De Jong, O., Savelsbergh, E. R., & Alblas (Eds.). (2001). *Teaching for scientific literacy: context, competency, and curriculum*. Utrecht: Utrecht University, Faculty of Sciences, Freudenthal Institute.

Drijvers, P., Doorman, M., Boon, P., Reed, H., & Gravemeijer, K. (2010). The teacher and the tool: instrumental orchestrations in the technology-rich mathematics classroom. *Educational Studies in Mathematics, 75*(2), 213-234.

Edens, K. M. (2010). Preparing Problem Solvers for the 21st Century through Problem-Based Learning. *College Teaching, 48*(2), 55-60. doi: 10.1080/87567550009595813

Freudenthal, H. (1991). *Revisiting Mathematics Education. China Lectures*. Dordrecht: Kluwer Academic Publishers.

Gravemeijer, K. P. E. (1994). *Developing realistic mathematics education.* CDbeta press, Utrecht. Retrieved from http://www.fisme.science.uu.nl/toepassingen/20014/ (FIsme Scientific Library 14)

Gresnigt, R., Taconis, R., Van Keulen, H., Gravemeijer, K., & Baartman, L. (2014). Promoting science and technology in primary education: a review of integrated curricula. *Studies in Science Education, 50*(1), 47–84. doi: 10.1080/03057267.2013.877694

Gui, M., & Argentin, G. (2011). Digital skills of internet natives: Different forms of digital literacy in a random sample of northern Italian high school students. *New Media & Society, 13*(6).

Hoyles, C., Noss, R., & Kent, P. (2003). Techno-mathematical literacies. Retrieved July 17, 2013, from www.ioe.ac.uk/tlrp/technomaths

Jablonka, E. (2003). Mathematical literacy. In A. J. Bishop, K. Clements, C. Keitel, J. Kilpatrick & F. K. S. Leung (Eds.), *Second International Handbook of Mathematics Education* (pp. 75-102). Dordrecht: Kluwer Academic Publishers.

Libow Martinez, S., & Stager, G. (2014). *Invent To Learn. Making, Tinkering, and Engineering in the Classroom*.

Mishra, P., & Koehler, M. (2006). Technological Pedagogical Content Knowledge: A Framework for Teacher Knowledge. *Teachers College Record, 108*(6), 1017–1054.

Papert, S. (2003). MicroWorlds EX. Retrieved from http://www.microworlds.com/

Trilling, B., & Fadel, C. (2009). *21st Century Skills: Learning for Life in Our Times*. Hoboken, NJ: John Wiley & Sons.

Van den Heuvel-Panhuizen, M. (2000). Mathematics education in the Netherlands: A guided tour. Freudenthal Institute Cd-rom for ICME9. Utrecht: Utrecht University.

Van Galen, F., & Jonker, V. (2013). *Experimenteren in de rekenles. Combineren van rekenlessen met lessen wetenschap & techniek*. Utrecht: Kenniscentrum Talentontwikkeling, Wetenschap & Techniek. Universiteit Utrecht.