Sprego: case study on the effectiveness of teaching spreadsheet management with schema construction

Gábor Csapó 1 • Mária Csernoch 1 • Kálmán Abari 2

Received: 28 June 2019 / Accepted: 20 September 2019 / Published online: 11 November 2019
© The Author(s) 2019

Abstract
In the modern, information driven society managing and handling data is unavoidable. The most common form of data handling is to organize data into tables and complete operations on them in spreadsheets. Sprego (Spreadsheet Lego) is a programming-oriented methodology focusing on schemata construction and authentic problem-solving working with only a limited number of general-purpose functions. In our current study the goal is to present Sprego as an alternative method for spreadsheeting, and to measure its effectiveness in education compared to the traditional surface approach methods. We also aim to highlight the advantages of teaching datamanagement, spreadsheeting, and introduction to programming by applying an algorithmic and schemata centric method in a user-friendly interface. The teaching and testing were carried out in three classes of a local middle and high school with two experimental and one control groups. Based on our results, it is found that the Sprego methodology is significantly more effective than the traditional surface approach methods. Furthermore, it is also proved, in accordance with similar studies in sciences, that students, who had studied traditional spreadsheet management in advance to this experience, have difficulties switching to Sprego. Although these students alternate between the two approaches, our measurements clearly prove that the traditional approach is pushed into the background, as students prefer to solve problems using Sprego. Our findings also imply that traditional methods do not develop long-lasting knowledge which students could rely on, and have a negative effect on their development, while Sprego seems much more reliable.

Keywords Spreadsheet management • ICT education • Algorithmic skills • Computational thinking

Gábor Csapó
csapo.gabor@inf.unideb.hu

Extended author information available on the last page of the article
1 Introduction

1.1 Spreadsheet management in education

Spreadsheet management is integrated into curricula worldwide – either into ICT or other school subjects. Teaching spreadsheeting usually focuses on interfaces, operating system commands and several problem-specific functions built into the spreadsheet management environments (Csernoch 2017; ECDL Foundation 2016; ICAEW 2016; JobTestPrep 2016; Katz 2010; Microsoft 2016, 2019a; Test ECDL 2019a, b, c; Walkenbach 2010). Students, who follow the instructions of teachers and/or coursebooks, tutorials and wizards, are meant to use these functions with their specific arguments and use-cases without any background knowledge and information and without the ability to evaluate their results in problem-solving. However, research indicates that students starting their tertiary studies do not have the required level of knowledge in various fields of ICT, including spreadsheet management (Biró et al. 2015; Csernoch et al. 2015). This also means that students complete secondary education without a stable and reliable knowledge in data-management. This negative effect can be explained by the approaches currently running in ICT education which focus primarily on the second level – usage – of the three levels of understanding defined in the ACM and IEEE report (2013):

- Familiarity: understanding the problem,
- Usage: applying the concept based on a technique,
- Assessment: consideration, discussion, debugging.

This is an interface-centered approach (Angeli 2013; Ben-Ari 1999), which ignores the first level and consequently makes the assessment level inaccessible. Therefore, students focus on the interface during their surface-navigation activities, instead of analyzing the problem, creating a well-structured algorithm, discussing and debugging the results (Bell and Newton 2013; Biró and Csernoch 2013; Champagne et al. 1983; Panko 2013; Pólya 1954). This leads to erroneous documents, mishandling of data, and therefore potential financial losses for the individual or for the company (EuSpRIG 2019; Garrett 2015; Panko 2013, 2016; Panko and Port 2013). To avoid such threats and to make education able to prepare students for the requirements of our computer and information-driven society, a paradigm shift and a novel approach is required worldwide in ICT education that teaches students data-management instead of focusing on the features of interfaces.

1.2 Sprego

Sprego (Spreadsheet Lego) is a methodology to teach spreadsheet management while developing the students’ computational thinking and algorithmic skills (Wing 2006) without forcing the use of the ever-increasing number of problem-specific functions (Csernoch 2014; Csernoch et al. 2014; Csernoch and Biró 2015a, b, c). The methodology focuses on twelve general-purpose functions (Table 1) and encourages students to build up algorithms of well-known programming principles (Wakeling 2007). In the usage phase (ACM and IEEE report 2013), students code their algorithms by using
composite array formulas in the functional language of spreadsheet programs, which is announced as the latest built-in feature of recently published spreadsheet versions (Microsoft 2019c; Williams 2019). Using this algorithm construction and coding process, Sprego avoids the hindrance of the problem-specific functions used in traditional methods, while it puts a great emphasis on discussing and debugging the input and output values of each step of the algorithms. Consequently, it relieves students of the unnecessary burden of learning several hundreds of function names, use cases, and arguments (ECDL Foundation 2016; ICAEW 2016; Katz 2010; Microsoft 2016, 2019a; Walkenbach 2010).

The Sprego methodology facilitates schema construction and intelligent learning (van Merriënboer and Sweller 2005; Skemp 1971) and creates long-lasting knowledge (Carr 2011; Kahneman 2011; Csernoch and Biró 2015b; National Research Council 2000) through invoking analytic slow thinking and routine fast thinking (Csernoch 2017; Kahneman 2011). Since students work with and reuse general-purpose functions in Sprego to build up their algorithms throughout the topic, they experience less change in their learning and work environment and can progress more efficiently (Kátai et al. 2016; Osztián et al. 2017). Furthermore, this high-mathability (Baranyi and Gilanyi 2013; Biró and Csernoch 2015a, b; Chmielewska et al. 2016; Csernoch and Biró 2015a) approach facilitates knowledge-transfer between different ICT topics, other sciences and school subjects (Csernoch 2019a, b). Students learn datamanagement through working with authentic data gathered and converted from the Internet (Biró and Csernoch 2017b; Csernoch and Dani 2017) from contexts that they are familiar with and interested in their everyday life. Using such datatables, students learn spreadsheeting in depth, as required by curricula widely accepted, through the following sub-topics:

- string operations
- conditional calculation and formatting
- linear and binary search

Sprego is suitable for several age groups – with various methodologies and contents – and therefore it can be applied at different levels of ICT education. The well-known spreadsheet management environments used in both education and in the industry (for example Microsoft Excel (Microsoft 2019b), LibreOffice Calc (The Document Foundation 2019), Google Sheets (Google 2019) and Apple Numbers (Apple 2019)) all support Sprego programming, which makes the methodology platform-independent.

Based on the fundamental programming principles, Sprego is suitable for teaching

---

**Table 1** The twelve general-purpose spreadsheet functions grouped into three categories

| Sprego text | Sprego number | Sprego pro |
|-------------|--------------|------------|
| LEFT()      | MIN()        | IF()       |
| RIGHT()     | MAX()        | INDEX()    |
| LEN()       | SUM()        | MATCH()    |
| SEARCH()    | AVERAGE()    | ISERROR()  |

---
spreadsheet management, and serves as an introduction to database management and programming.

1.3 Unplugged and semi-unplugged Sprego tools

To help students understand the algorithms behind the problems, evoking the proper use of fast- and slow thinking (Kahneman 2011) and making the classwork more streamlined, the methodology uses various unplugged (Bell and Newton 2013; Biró and Csernoch 2017a) and semi-unplugged tools (Biró and Csernoch 2017b; Csapó 2017a, b; Csapó and Sebestyén 2017; Gulácsi and Dienes 2018; Sebestyén et al. 2018).

One of the unplugged approaches which we apply in Sprego classes is the play-along method, where students play the algorithms in class, using their own handmade tools serving as props. Another tool, which we apply, serves the better understanding of multilevel functions. The students work with 3D printed Matrjoska dolls and painter’s tape, or with paper boats folded at the scene (Fig. 1). In either case, both teachers and students write the steps of the algorithm with their input and output values on the tools and using the physical properties of the items, they simulate the behavior of composite functions – how a smaller-level output value serves as a higher-level input. The students work with as many pieces of these handy tools as the steps of the algorithm. By putting the dolls/boats into each other they create the formulas starting at the innermost and working towards outward. After building the algorithm with unplugged tools and coding it in a spreadsheet document, the painter’s tape or boats, which substitute the students’ notetaking, are attached to the students’ notebook. In this way the method ensures valid and uniform progress for the whole class, and these tools can be used for further reference, while handling novel problems that require the same algorithm.

Regarding semi-unplugged tools, Sprego uses authentic tables converted from the Internet, as mentioned above. These tables are an essential part of Sprego classes as students analyze and process their data. In contrast to decontextualized classroom materials, which overflooded both the online and printed teaching materials (Angeli 2013; Csernoch 2017), the authentic sources provide students with meaningful data and leaves the unnecessary practice of typing tables. Even though these tables are reused in several Sprego tasks, it is important to note that the skills students acquire and develop while learning with this methodology, are not associated with specific authentic or pre-constructed data sets.

Along with the webtables, Sprego offers two educational applications (2D and 3D) to visually present the algorithms of two of the most common programming problems. The 2D application (Csapó and Sebestyén 2017; Sebestyén et al. 2018) available on

![Fig. 1](image-url)  
Samples of the unplugged tools used in Sprego problem-solving: 3D printed doll set (left) and origami boats (right)
Google Play (Csapó and Sebestyén 2019) presents the conditional counting problem, both with equality (Fig. 2) and inequality in the condition, and the linear search algorithm. The software also includes a formula evaluator panel, based on the same tool of the spreadsheet management software, to provide an easy-to-read and visually engaging interface for students to analyze the steps of the problem at hand.

The 3D application available for desktop operating systems (Gulácsi and Dienes 2018) builds on the experience and feedback received from students using the 2D version. At the time of writing the present paper, the current version includes the conditional counting (Fig. 2) and linear search presentations using the same contexts as their 2D variants. Besides the different graphical implementation, this version of the application also encompasses interactive forms of each presentation, where the students can play along. The program poses the yes/no questions of the conditions, which the students must answer correctly to continue the evaluation. Note that while the two applications present similar Sprego problems, they serve different purposes, based on the design of their interface and the supported platforms.

Another semi-unplugged tool of the method is the Sprego Virtual Collaboration Space (Csapó 2017a, b) which has the aim to put all the Sprego contents together in a 3D space to provide an all-inclusive learning environment for students. While the space is suitable for classroom use, its main purpose is to facilitate self-learning and practice with the Sprego method. The space is based on the MaxWhere collaboration space software package (Berki 2018; Horváth 2018a, b; Galambos et al. 2010, 2011; Lampert et al. 2018; MISTEMS 2019). The MaxWhere environment is constantly gaining popularity and is used in varying fields of sciences from education to remodeling – presenting destroyed monuments, ruins, inaccessible places and documents, etc. – in a virtual space (Gilányi and Virágos 2013; Gilányi et al. 2015a, b).

These tools are all designed to support the Sprego methodology in education, which is strengthened by experience and feedback provided by teachers. Both the unplugged and semi-unplugged tools seem to be efficient, although the scientific measurements and statistical analyses of their effectiveness are currently in progress and not part of the present study.

2 Testing the effectiveness of Sprego

2.1 Sample and testing methods

Our goal was to test and measure the effectiveness of the Sprego methodology compared to traditional approaches for teaching spreadsheet management. The research
was carried out according to the following process. Before the experiment took place, all the students filled in a pretest, to measure their prior knowledge (Section 2.2). The experiment covered teaching spreadsheeting in three classes of a local middle and high school, detailed in following paragraphs. After completing the educational process, the post-test was administered, where the tasks of the pre-test were repeated.

The experiment consisted of three classes in a local middle and high-school: two grade-7 classes and one grade-10 class. All the groups were assigned to the same teacher throughout the whole school year, which means that from our point of view the groups were randomly selected. The grade-10 group and one of the grade-7 groups studied the subject using the Sprego methodology (experimental groups). The other grade-7 class, on the other hand, was taught using the official textbook available for high-schools (grades-9 and 10) (Dancsó and Korom 2013), with a traditional interface and problem-specific function-focused approach (control group). The textbook was analyzed beforehand, and no cognitive limits were found which would cause difficulties for grade-7 students. However, the textbook ignores the background knowledge of the intended school groups and starts spreadsheeting (Siersted 2018) at the absolute beginner level, using basic spreadsheet functions as well as typing tables. It is important to note that the two grade-7 groups had no prior knowledge in the subject – which was proved in the pretest (Section 2.5) – while the grade-10 class had previously studied spreadsheet management for two years according to the frame curriculum (OFI 2012, 2013). They encountered the topic in their grade-7 studies (1 lesson/week for 8 lessons) and, most recently, last year in grade-9 (2 lessons/week for 12 lessons). Furthermore, several grade-10 students acquired ECDL (European Computer Driving License) certifications from the spreadsheet management module, which indicates that not only should they be familiar with the topic, but should also have a sound knowledge of it, which they can use confidently and upon which we can build during the classes.

Regarding the number of classes, the experimental and the control grade-7 groups studied the topic for 13 and 12 lessons, respectively, with 1 class a week. The experimental grade-10 group studied the subject for 13 lessons with 2 classes a week. Considering the numerous disadvantages of typing data in classes (Csernoch 2017), we deviated at an early stage from the official textbook in such a way that instead of getting students to type data as the book instructed, we provided them with the tables and moved on to the topic of spreadsheet functions.

In the experimental groups, the first and second phase of the problem-solving process – understanding the problem and devising a plan (Pólya 1954) – were carried out with the coaching method (Burton and Brown 1979; Chmielewska and Matuszak 2017; Creasy and Paterson 2006; Flaherty 2006; Fletcher and Mullen 2012; Knight 2009; Rogers 2012; Van Nieuwerburgh 2012). During this stage, a continuous interaction runs between teacher and students. In the third phase – carrying out the plan – all students worked individually at separate computers following the teacher’s guidance. In the fourth phase the discussion and debugging – looking back – took place, again in the form of coaching.

Considering the tasks, carried out in the classes, the control group progressed following the steps, the instructions, and the spreadsheet functions presented in the official textbook. The experimental groups solved problems according to the Sprego methodology, detailed in Section 1.2. In Sprego classes, as mentioned above, the problem-solving method of Pólya (1954) was followed, combined with the coaching
interactive method and our invention of unplugged and semi-unplugged tools. The unplugged (3D printed doll sets) and semi-unplugged tools (authentic sources and the applications) were presented in the Sprego classes. The doll sets were used to guide the students through the algorithm construction progresses (as detailed in Section 1.3).

Considering the authentic sources, the following tables were used in classes during the experimental period:

- Top 250 movies based on the ratings of IMDb (Internet Movie Database) (IMDb 2019),
- The 100 tallest buildings in the world (Council on Tall Buildings and Urban Habitat 2019),
- The countries of the world (OFI 2004).

The Sprego applications were applied to reiterating knowledge items primarily introduced in previous classes, to help students practicing algorithms. In all groups the teacher constantly followed the students’ work, checking and correcting their progression.

Although all the participating students completed the requirements of the curriculum, there are slight differences between the number of students participating in the pre- and post-tests, as presented in Table 2. These differences can be explained by absences and various unforeseen school activities, making students miss their classes at the time of the tests. The number of paired tests shows the number of students who were present on both test days, with the grade-7 experiment group being the best attended ($N=12$), followed by the grade-10 experiment and grade-7 control groups ($N=11$ and $N=11$, respectively).

### 2.2 The composition of the tests

The tests were designed (1) to cover contents with the datamanagement and the programming aspects of spreadsheeting in mind and (2), to make students work with unfamiliar data using their analytical, computational-thinking and algorithmic skills and schemata they had built up during practice. To provide increasing levels of difficulty and a logical structure in which each task builds on the knowledge required to solve the previous one, we separate the test into two main sections:

- Basic formula-management and execution-order tasks (Task 1–Task 4),

### Table 2 The number of students in each group and their distribution considering the tests

|                     | Experimental groups | Control group |
|---------------------|---------------------|--------------|
|                     | grade-7             | grade-10     | grade-7 |
| Students in class   | 15                  | 18           | 13      |
| Pre-test            | 14                  | 16           | 11      |
| Post-test           | 13                  | 13           | 13      |
| Paired tests        | 12                  | 11           | 11      |
• Formula creation and function usage tasks (Tasks 5a–g).

In Task 1 the students are required to finish an incomplete spreadsheet formula for calculating averages. With this task our aim is to collect data on fundamental syntactic knowledge of how formulas and functions are constructed in spreadsheets. This is followed by tasks handling arguments (Task 2) and references of vectors and matrixes (Task 3). The last question in this section aims to measure the students’ ability to recognize and analyze the execution order of a composite array formula. Students are asked to put the commands in a correct execution order (Task 4). We must note here that the tests contain more rows, as a distractor, than the number of commands in the formula. If students listed the arguments of the functions as commands, we considered it a less serious mistake than writing the commands in the wrong order.

The second section of the test focuses (Tasks 5a-5g) on solving problems – creating spreadsheet formulas – based on the table and tasks presented (Fig. 3). For this section we use an authentic table – converted from a webtable – containing the names, code, and prices of cakes from a local pastry shop (Mandula Familia Kft 2017), which the students are familiar with (Fig. 3). In the first two tasks the names (Task 5a) and codes (Task 5b) have to be separated for each cake, because these two pieces of data are originally stored in one data field as seen in Fig. 3. The next task is to convert the prices – originally stored as a string (Fig. 3, column B) – to numbers (Task 5c). Task 5d focuses on forming logical conditions (asking yes/no questions) and then an IF() function with three arguments: the yes/no questions and two string constants for the output values.

For the next task, solving a conditional counting problem, the students have to compose either a multilevel array formula or a built-in problem-specific function (Task

### Table 1

|   | A                                      | B   | C   |
|---|----------------------------------------|-----|-----|
| 1 | Név - kód                              |     |     |
| 2 | Áfonya Hercegnő cukormentes tortája - KT057 | 3950 Ft | XXXX |
| 3 | Őrség zöld aranya torta ország tortája 2016 - KT058 | 3950 Ft |
| 4 | Az ország tortája 2015 - KT050           | 3950 Ft |
| 5 | Oreo keksz torta - KT052                | 3950 Ft |
| 56| Málnácska torta - KT056                 | 3800 Ft |
| 57| Lekváros kardinális torta - KT044       | 3500 Ft |
| 58| Karamellás csoki mousse torta - KT098   | 3800 Ft |

### Table 2

|   | A                                      | B   | C   |
|---|----------------------------------------|-----|-----|
| 1 | Name - code                            |     |     |
| 2 | Blueberry Princess’s sugar free cake - KT057 | 3950 Ft | XXXX |
| 3 | Őrség green gold cake of the country 2016 - KT058 | 3950 Ft |
| 4 | The cake of the country 2015 - KT050   | 3950 Ft |
| 5 | Oreo biscuit cake - KT052              | 3950 Ft |
| 56| Raspberry cake - KT056                | 3800 Ft |
| 57| Cardinal cake with jam - KT044        | 3500 Ft |
| 58| Chocolate mousse cake with toffee - KT098 | 3800 Ft |

Fig. 3 An extract from the table used in the second part of the tests (upper image) listing the name, code (column A) and price in Hungarian forints (column B) of cakes, respectively. The lower image is created of the same table translated to English.
5e), along with cell C2 as a variable (Fig. 3): count the number of cakes which are more expensive than the value stored in C2. The next two tasks focus on conditional calculation problems, to provide the sum and the average of prices based on a condition, where the prices are lower than the value stored in variable C2 (Tasks 5f and 5g, respectively). Similar to Task 5e, either multilevel array formulas or problem-specific built-in functions can be used to solve the problems.

2.3 Evaluation notes

Similarly to any programming tasks, there were multiple solutions for several problems to which we set up the items separately. Solutions based on the values of Column B (Fig. 3) – Tasks 5d–5g – were accepted, regardless of whether students handled the data type conversion or not.

During the lessons, students solve problems using computers, applications provided by the school, and occasionally unplugged tools. The data collection of the tests was carried out using unplugged tools in all groups to see clearly the algorithm-building processes and abilities of students. With this method the trial-and-error solutions were avoided.

We must further note that the grade-10 experimental group learned and solved problems with the algorithm of linear search as well, but since it was not present in the lower grades’ curriculum, it was excluded from the current study.

2.4 Hypotheses

Considering the nature of this study and our goal to measure the efficiency of the Sprego methodology, we posed the following hypotheses:

1. The Sprego method is more effective in teaching the spreadsheet-management topic than the traditional approaches.
2. The students’ prior knowledge in spreadsheet management taught with traditional methods affects negatively the effectiveness of Sprego.
3. Students who learned both with Sprego and traditional methods prefer to follow the “Sprego way” than using problem-specific functions to solve tasks that can be completed with either approach.
4. The traditional approach to teaching spreadsheet management does not develop long-lasting knowledge.

2.5 Results of the pre-tests

For our fourth hypothesis we compared the results of the pre-tests in the grade-10 and grade-7 groups, the latter involving both the experimental and control groups (Table 3). In order to reveal the differences between the novice and the experienced groups, we used Mann–Whitney tests. The results show that the grade-7 students completed the test with almost zero prior knowledge of the topic, because they had not encountered it in ICT education beforehand. It follows that in almost every task grade-10 students earned significantly higher scores, except in four tasks. Both groups completed Task 2 at a 25.00% average score with no significant difference ($p = 0.5829$). This result demonstrates that despite their prior
knowledge, the grade-10 students could not recognize correct argument formats better than those students who had never worked with spreadsheet functions. A similar phenomenon can be observed working with string functions and data conversion (Tasks 5a–5c). While in these tasks grade-10 students earned higher scores than grade-7 students, the difference was not significant in any of these cases \((p = 0.0794, p = 0.2301\), respectively). Note, that for Task 5c both groups scored 0.00 points and therefore, their difference could not be calculated with the Mann–Whitney test. This can be traced back to the practice that string functions are not part of the traditional spreadsheet approach, regardless of their importance in programming.

Considering the overall pre-test, grade-7 groups earned a 5.52% average score, while the grade-10 students completed it with an 18.75% result, making the difference between the groups significant \((p = 0.0000)\) (Fig. 4). However, it is important to keep in mind that the grade-10 students earned this significant difference with very low scores in the second part (Tasks 5a–5g: using spreadsheet functions) (below 8%) compared to the 0% of the grade-7 students. Furthermore, grade-10 students have more developed abstraction and cognitive skills and, additionally, several of them have ECDL certifications that they seemingly could not take advantage of in the test. All these things considered, the assumed abilities of grade-10 students would imply much higher results when doing tasks using spreadsheet functions. Therefore, we cannot say safely that the traditional approaches develop long-lasting knowledge regardless of the statistically proven significant differences. In this light, we conclude that this last hypothesis is statistically rejected but consider it proved nonetheless, based on the details discussed above.

### 2.6 Results of the grade-7 groups

Considering our first hypothesis, which compares the results of the post-tests of grade-7 experimental and control groups, we analyzed each task of the tests separately to get an

| Tasks | grade-7 | grade-10 | \(p\) |
|-------|---------|-----------|------|
| 1     | 20.00   | 64.58     | 0.0000 |
| 2     | 25.00   | 25.00     | 0.5829 |
| 3     | 2.67    | 44.79     | 0.0000 |
| 4     | 13.00   | 49.48     | 0.0002 |
| 5a    | 0.00    | 2.50      | 0.0794 |
| 5b    | 0.00    | 0.63      | 0.2301 |
| 5c    | 0.00    | 0.00      |       |
| 5d    | 0.00    | 7.81      | 0.0036 |
| 5e    | 0.00    | 3.91      | 0.0285 |
| 5f    | 0.00    | 3.75      | 0.0285 |
| 5g    | 0.00    | 3.75      | 0.0102 |
| Total | 5.52    | 18.75     | 0.0000 |
overview of the effectiveness of the Sprego method in the spreadsheet management topic, as well as in various subfields of it (Table 4). We calculated the average results of the students for each task and for the whole test and used Mann–Whitney tests to analyze the significance between the scores of the two grade-7 groups.

The data in Table 4 show that students who learned with the Sprego methodology completed the tasks more successfully in all but one task – Task 5d –, which focused on composing an IF() function with constant outputs. A similar tendency was revealed during our previous experiments, and we must draw attention to this phenomenon. Students learning with the traditional approach tend to achieve better results for the tasks focusing only on constant IF() functions. However, these students usually stop there, do not reach higher levels of abstraction and have a hard time completing tasks that require variable values and more complex algorithms. This previously observed

Table 4 The average results (%) for each task and the p-values, based on the differences between them for both grade-7 experimental and control groups in the post-test

| Tasks | grade-7 experiment | grade-7 control | p   |
|-------|--------------------|-----------------|-----|
| 1     | 94.87              | 74.36           | 0.0175 |
| 2     | 61.54              | 42.31           | 0.0082 |
| 3     | 96.15              | 87.18           | 0.5505 |
| 4     | 79.49              | 67.31           | 0.5009 |
| 5a    | 60.00              | 1.54            | 0.0001 |
| 5b    | 40.00              | 1.54            | 0.0002 |
| 5c    | 42.31              | 0.00            | 0.0000 |
| 5d    | 39.10              | 53.21           | 0.3941 |
| 5e    | 76.15              | 41.35           | 0.0046 |
| 5f    | 68.46              | 29.23           | 0.0161 |
| 5g    | 61.54              | 20.00           | 0.0041 |
| Total | 65.42              | 38.00           | 0.0013 |
pattern (Csernoch et al. 2015) was confirmed by our statistical analyses in Tasks 5d–5g (Table 4). The experimental group achieved significantly better results than the control group in the following tasks:

- syntax of formulas (Task 1, \( p = 0.0175 \)),
- correct argument formats (Task 2, \( p = 0.0082 \)),
- returning text with a constant number of characters (Task 5a, \( p = 0.0001 \)),
- returning text with a varying number of characters (Task 5b, \( p = 0.0002 \)),
- converting numbers stored as text to numbers (Task 5c, \( p = 0.0000 \)),
- a conditional counting problem (Task 5e, \( p = 0.0046 \)),
- a conditional summing problem (Task 5f, \( p = 0.0161 \)),
- a conditional averaging problem (Task 5g, \( p = 0.0041 \)).

Although the experimental group completed Tasks 3 and 4 more successfully, the results showed no significant differences in either case (\( p = 0.5505 \) and \( p = 0.5009 \), respectively). Furthermore, while the control group achieved better scores in Task 5d the difference was not significant between the two groups (\( p = 0.3941 \)).

Considering the results of the whole post-test, the experimental group completed the tasks more successfully (65.42%) than the control group (38.00%) with a significant difference (\( p = 0.0013 \)) (Fig. 4). We can conclude, based on the data of the post-tests of the grade-7 experimental and control groups, that our first hypothesis proved that grade-7 students who learned with Sprego complete the post-tests more successfully than the control group.

2.7 Results of grade-7 and 10 experimental groups

The second hypothesis focuses on one of the differences between the experiment groups (grade-7 and grade-10 experimental groups): how previous studies influence progress in Sprego. An extract of the relevant data is presented in Table 5. In accordance with Champagne et al. (1983), we found that previous native and/or shallow knowledge hinders the acquisition of high-mathability problem-solving skills. Grade-10 students, despite their various advantages – age, experience, previous classes, etc. –, could not achieve higher scores than the grade-7 experimental group.

In 8 out of 11 tasks the grade-7 group performed better; however, the difference between the two groups is not significant. The grade-10 students only achieved better results in three tasks: returning text with a constant number of characters (Task 5a), using the IF() function with constant outputs (Task 5d), and the conditional averaging task (Task 5g). Similarly to the tasks where grade-7 students performed better, the differences between the two experimental groups were not significant in any of these three tasks. Also note that we accepted several possible solutions for tasks that can be solved either with Sprego or traditional approaches, and for the grade-10 group, the current results encompass not only Sprego answers, but also those achieved with other methods.

While the numbers in Table 5 clearly show no significant difference between the results of the two Sprego groups, it is worth taking into consideration the fact that the grade-10 students had studied the topic beforehand twice, in grades 7 and 9, and, as
mentioned before, by their age they have more developed abstraction and cognitive abilities. Filling in the same tests resulted in minimal differences between the two groups. At this point, going beyond how previous surface approaches affected the problem-solving abilities of grade-10 students, we must draw attention to the results, which prove that surface approach methods do not support the development of sequencing, spatial, and mnemonic abilities (Kruck et al. 2003). If grade-10 students have higher abstraction and problem-solving abilities, this is more likely to be acquired in other classes, rather than classes focusing on spreadsheets.

Considering that the average score of the grade-10 experimental group (64.34%) is not higher than that of the grade-7 experimental group (65.42%) — although the difference is not significant \((p = 0.7388)\) — the second hypothesis is proved (Fig. 4). In other words, the grade-10 experimental group does not achieve significantly better results in the post-tests than the grade-7 experimental group.

### 2.8 Selection of methods

For our third hypothesis we analyzed the selection of methods for solving the problems. In the grade-10 post-test we also wanted to see how students, who studied with both traditional and Sprego methodologies, build and choose items from their function base. We assumed that students would be influenced by the “Sprego way”, but we were afraid, given previous results, that their prior studies would have a negative effect on their development (Champagne et al. 1983; Kruck et al. 2003; Moyo et al. 2016). For this part of our analysis, only Tasks 5e–5g (Table 6) were relevant because these are the ones that show stronger differences in their solutions between the two approaches. The numbers of students dealing with the problems are detailed in Table 7, separated by their groups and selected methods. As students could not possess knowledge of the

### Table 5

The average results of grade-7 and 10 experimental groups for each task with the \(p\)-values based on the differences in the post-test

| Tasks | grade-7 experimental | grade-10 experimental | \(p\)  |
|-------|----------------------|------------------------|------|
| 1     | 94.87                | 92.31                  | 0.6524 |
| 2     | 61.54                | 59.62                  | 0.8691 |
| 3     | 96.15                | 93.59                  | 0.4229 |
| 4     | 79.49                | 72.44                  | 0.5654 |
| 5a    | 60.00                | 69.23                  | 0.7028 |
| 5b    | 40.00                | 39.23                  | 0.8762 |
| 5c    | 42.31                | 34.62                  | 0.5831 |
| 5d    | 39.10                | 57.69                  | 0.1708 |
| 5e    | 76.15                | 59.04                  | 0.1666 |
| 5f    | 68.46                | 66.92                  | 0.6380 |
| 5g    | 61.54                | 63.08                  | 0.9586 |
| Total | 65.42                | 64.34                  | 0.7388 |
Sprego methodology prior to their Sprego classes, we counted empty submissions as part of the traditional approach.

Table 7 shows that based on the methodology applied, with minimal deviation, both grade-7 groups tend to follow one specific path, i.e. the method, which was introduced in their class. In contrast, we can observe stronger differences regarding the grade-10 post-test.

While almost all students in the grade-10 experimental group used problem-specific functions to solve the tasks of the pre-test, a high number of them switched to Sprego solutions in the post-tests (Tables 6 and 7). The conditional counting task (Task 5e) shows the smallest difference (only 3 more students followed the Sprego method than the traditional) which can be accounted for by the widespread use of the COUNTIF() function that students learn at an early stage of their ICT studies and revisit almost every time the topic is encountered. However, the conditional sum and average tasks (Tasks 5f and 5g) were solved by applying the algorithm they share and coded with the general-purpose Sprego functions, with only a few exceptions (2 and 3 students, respectively).

The tendency towards switching to Sprego is obvious based on the comparison of the data in Table 7. To analyze significant differences between the number of students who used different approaches in the grade-10 post-tests, we used binomial tests. In the tasks involving conditional counting (Task 5e) and averaging (Task 5g) the students showed no significant differences in their selection of methods ($p = 0.5811$ and $p = 0.0923$, respectively). However, in Task 5f, which requires the same algorithm as Tasks 5e and 5g, the difference was significant ($p = 0.0225$).

The comparison of the selected problem-solving methods of grade-7 and 10 experimental groups clearly shows that the students’ prior knowledge influences their

| Tests  | Tasks | Grade-7 exp. | Grade-7 cont. | Grade-10 exp. |
|--------|-------|--------------|---------------|---------------|
|        |       | Trad.        | Sprego        | Trad.         | Sprego        | Trad. | Sprego |
| Pre-test | 5e    | 14           | 0             | 11            | 0             | 14    | 2      |
|         | 5f    | 14           | 0             | 11            | 0             | 13    | 3      |
|         | 5g    | 14           | 0             | 11            | 0             | 12    | 4      |
| Post-test | 5e   | 1            | 12            | 13            | 0             | 5     | 8      |
|         | 5f    | 1            | 12            | 13            | 0             | 2     | 11     |
|         | 5g    | 1            | 12            | 12            | 1             | 3     | 10     |

Table 6 The Sprego solutions of Tasks 5e–5g. The formulas clearly show that the tasks are based on the same algorithm: (1) yes/no question, (2) the output values in the case of the yes (true) answers, and (3) counting (sum, sum, and average, respectively).

{=SUM(IF(B2:B58 > C2,1))} Task 5e
{=SUM(IF(B2:B58 < C2,B2:B58))} Task 5f
{=AVERAGE(IF(B2:B58 < C2,B2:B58))} Task 5g
decisions and approaches in the selection of functions. This is also confirmed by our experiences while correcting the tests, as we encountered several cases where grade-10 students tried to use either both methods (unsuccessfully) or approached the tasks using problem-specific functions which they did not remember how to use correctly. We also encountered the IF() function used with the logic and arguments of COUNTIF(), or SUM() and AVERAGE() functions with the arguments of SUMIF() and AVERAGEIF().

In Table 8 we present the number of students with their selection of methods without the empty submissions being counted as the traditional approach. We can observe considerable drops in the number of students in the traditional categories compared to the data in Table 7, except for the grade-7 control group’s post-test.

The grade-10 experimental group’s pre-test data further strengthens our fourth hypothesis discussed in Section 2.5, because it shows that most of the students (81%, 81% and 75% of the groups, respectively) who studied beforehand with traditional approaches could not even begin to solve these tasks. Considering the grade-10 experimental group’s post-test data, the students follow the pattern observed before, given that they tend to choose to work with Sprego. Using binomial tests, the significance differs slightly from our previous analyses: $p = 0.2266$, $p = 0.0064$ and $p = 0.0117$ for Tasks 5e, 5f and 5g, respectively. This leads us to the conclusion that if we disregard the empty submissions, the students choose Sprego with a significant difference in 2 out of 3 tasks.

The choice of using the Sprego approach is important in cross-topic learning and knowledge-transfer, as well. The higher number of students who used the Sprego approach follows on the logical, sequencing, spatial, and function-structuring abilities, because the algorithms the students learn and work with are the same in programming practices and build on the logic of programming (Table 6). As a higher number of grade-10 students chose to work with Sprego in the post-tests, but fewer than those in the grade-7 experimental group, our third hypothesis is proved, in accordance with previously published results. However, the tendencies are clear; these results prove that traditional surface approach methods and native knowledge have a negative effect on the students’ problem-solving skills.

| Tests  | Tasks | Grade-7 exp. | Grade-7 cont. | Grade-10 exp. |
|--------|-------|--------------|---------------|---------------|
|        |       | Trad. | Sprego | Trad. | Sprego | Trad. | Sprego |
| Pre-test | 5e | 0 | 0 | 0 | 0 | 1 | 2 |
|         | 5f | 0 | 0 | 0 | 0 | 0 | 3 |
|         | 5g | 0 | 0 | 0 | 0 | 0 | 4 |
| Post-test | 5e | 0 | 12 | 10 | 0 | 3 | 8 |
|          | 5f | 0 | 12 | 10 | 0 | 1 | 11 |
|          | 5g | 0 | 12 | 8 | 1 | 1 | 10 |
3 Summary

3.1 Sprego or traditional methods?

In this study we aimed to measure the effectiveness of the Sprego methodology in teaching spreadsheeting as against the traditional surface approaches. Following the results of our analyses and our hypotheses, it is clear that Sprego has advantages over the traditional methods. The methodology is significantly more effective in the teaching context, resulting in a considerably higher score on the post-tests than the group who were taught with a traditional approach, using problem-specific functions without datamanagement and conscious algorithm construction. This concludes that Sprego students can work with their function base and can build algorithms more effectively, covering the same functionality and problems as the students who rely on problem-specific functions, or in other words, pre-prepared functions, without being familiar with the algorithms behind them.

When students study through multiple approaches and methodologies over several years, some discrepancy is expected in their knowledge structure. Based on these previous findings, we expected grade-10 Sprego students, who had previously studied with traditional methodologies, to have no significant difference in the post-tests compared to those who were introduced to spreadsheet management through Sprego. The analysis detailed in this study verified our expectations that students who had prior knowledge of problem-specific functions cannot achieve better results than the younger students who were studying spreadsheet management for the first time with Sprego. This result can be explained by the finding that grade-10 students alternate between the Sprego and traditional approaches when building their algorithms and solving the tasks. In other words, students who have prior knowledge of the problem-specific approach and later of Sprego try to work based on the schemata they developed in Sprego classes and construct their solutions accordingly with a strong influence deriving from their prior studies, in spite of that they cannot apply those functions without errors.

In the pre-tests, students who studied with traditional methods beforehand scored very low (below 8%) on tasks where they had to use spreadsheet functions. Considering the prior knowledge of the students and their age group, we found these results extremely low, despite the fact that statistically they are significantly better than the scores of the group (0%) who had no prior knowledge of the topic. These results show that learning with traditional approaches does not develop long-lasting knowledge regarding problem-solving with spreadsheet functions. The low scores of the students indicate that they possessed a knowledge level that they could not build on before they started learning with Sprego.

3.2 Conclusions

Teaching spreadsheet management has an important role in ICT education as students learn various aspects of computer science and develop connecting skills throughout the topic, including spreadsheet management, handling data structures, database management, programming principles, logical and computational thinking and algorithmic skills. Teaching this topic efficiently helps not only the workflow of the participants in the learning progress, but also prepares students for their future studies and to fulfil
the requirements modern jobs and careers demand of employees. Sprego is a methodology to teach datamanagement in spreadsheet interfaces, focusing on a limited number of general-purpose functions and building up algorithms. It promotes schemata construction through authentic problem-solving and algorithm construction. Following the data and results discussed in the present study, Sprego is a more effective and obvious choice to teach the topic, compared to traditional approaches. We encourage ICT educators to learn the Sprego method and bring it into classrooms where they and the students both can benefit from its advantages. Furthermore, we would like to point out that Sprego is not only suitable for the field of ICT education, as it is also able to cover the spreadsheet requirements employers expect and, based on its approach to problem-solving, it has the potential to reduce the error rate and therefore the potential financial losses a company can suffer due to spreadsheet errors.

In our current work we analyzed the effectiveness of the Sprego methodology on a small sample available to us. While the results are positive considering our hypotheses, we would like to point out that our current measurement is not representative based on the size of the sample. However, the results of this study may serve as basis for future measurements and studies using the Sprego methodology. Future work would include continuous development of this method and its unplugged and semi-unplugged tools. Furthermore, we plan to test the effectiveness of this method on a larger sample of students enrolled in secondary education. The preparations of this measurement are already under discussion.

Acknowledgements Open access funding provided by University of Debrecen (DE).

Funding information This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

ACM, & IEEE. (2013). Computer science curricula: Curriculum guidelines for undergraduate degree programs in computer science. New York: ACM Press, and IEEE Computer Society Press. https://doi.org/10.1145/2534860.

Angeli, C. (2013). Teaching Spreadsheets: A TPCK perspective. D. M. Kadijevich, C. Angeli, & C. Schulte, Improving Computer Science Education. Routledge.

Apple (2019). Numbers – Apple. https://www.apple.com/numbers/. Accessed 28 Jan 2019.

Baranyi, P., & Gilanyi, A. (2013). Mathability: emulating and enhancing human mathematical capabilities. 4th IEEE International Conference on Cognitive Infocommunications, CogInfoCom, 555–558.

Bell, T., & Newton, H. (2013). Unplugging computer science. In D. M. Kadijevich, C. Angeli, & C. Schulte (Eds.), Improving computer science education. Abingdon: Routledge.

Ben-Ari, M. (1999). Bricolage Forever!. PPiG 1999, 11th Annual Workshop, 5–7 January, 1999. University of Leeds: Computer-Based Learning Unit, UK. http://www.ppiog.org/papers/11th-benari.pdf. Accessed 1 Dec 2016.

Berki, B. (2018). Desktop VR and the Use of Supplementary Visual Information. 9th IEEE International Conference on Cognitive Infocommunications, CogInfoCom, 333–336.
Csernoch, M., Biró, P., Abari, K., & Máth, J. (2014). Programming-oriented spreadsheet functions. In Hungarian: Programozásorientált táblázatkezelői függvények. XIV. Országos Neveléstudományi Konferencia. Debrecen, Hungary.

Csernoch, M., Biró, P., Máth, J., & Abari, K. (2015). Testing algorithmic skills in traditional and non-traditional programming environments. Informatics in Education, 14(2), 175–197. https://doi.org/10.15388/infedu.2015.11.

Dancsó, T., & Korom, P. (2013). Informatics 9–10. for high-schools. In Hungarian: Informatika 9–10. a gimnáziumok számára. Budapest: Nemzedékek Tudása.

ECDL Foundation (2016). ECDL Module | Spreadsheets: Syllabus Version 6.0. http://ecdl.org/media/ecdlspreadsheets_6.01.pdf. Accessed 14 March 2019.

EuSpRIG, European Spreadsheet Risk Interest Group (2019). EuSpRIG Horror Stories. http://www.eusprig.org/horror-stories.htm. Accessed 28 June 2019.

Flaherty, J. (2006). Coaching: Evoking excellence in others. Development and Learning in Organizations: An International Journal, 20(6).

Fletcher, S., & Mullen, C. A. (2012). Sage handbook of mentoring and coaching in education. Thousand Oaks: Sage.

Galambos P., Barna R., & Baranyi P. Z. (2010). Introduction of Virtual Collaboration Arena (VirCA). The 7th International Conference on Ubiquitous Robots and Ambient Intelligence, 575–576.

Gilányi, A., & Virágos, M. (2013). Library treasures in a virtual world. 4th IEEE International Conference on Cognitive Infocommunications, CognInfoCom, 563–566.

Gilányi, A., Bálint, M., Museum, H., Hajdu, R., Tarsoly, S., & Erdős, I. (2015a). A visualization of the medieval Church of Zelemér. 6th IEEE International Conference on Cognitive Infocommunications, CognInfoCom, 449–453.

Gilányi, A., Bálint, M., Hajdu, R., Tarsoly, S., & Erdős, I. (2015b). Presentation of the Church of Zelemér in the Virtual Collaboration Arena (VirCA). 6th IEEE International Conference on Cognitive Infocommunications, CognInfoCom.

Google (2019). Google Sheets: Free Online Spreadsheets for Personal Use. https://www.google.com/sheets/about/. Accessed 15 Jan 2019.

Gulácsi, Á., & Dienes, N. (2018). 3D Software Environment for Educational Sprego Programming. The Turkish Online Journal of Educational Technology, INTE 2018 november (2), 837–844. http://www.tojet.net/special/2018_12_3.pdf. Accessed 21 Jan 2019.

Horváth, I. (2018a). Evolution of teaching roles and tasks in VR / AR-based education, 9th IEEE International Conference on Cognitive Infocommunications, CognInfoCom, 355–360.

Horváth, Zs. T. (2018b). Another e-learning method in upper primary school: 3D spaces. 9th IEEE International Conference on Cognitive Infocommunications, CognInfoCom, 405–408.

ICAEW (2016). Spreadsheet competency framework: A structure for classifying spreadsheet ability in finance professionals. London: ICAEW. http://www.icaew.com/-/media/corporate/files/technical/information-technology/it-faculty/spreadsheet-competency-framework.aspx. Accessed 21 Nov 2016.

IMDb (2019). Top Rated Movies. https://www.imdb.com/chart/top. Accessed 7 Sept 2019.

JobbyTestPrep (2016). SAM (Software Asset Management) Intermediate/Advanced Excel Practice Tests for Interview – 2016. https://www.jobtestprep.com/excel-practice-test-advanced-2016. Accessed 14 March 2019.

Kahneman, D. (2011). Thinking, fast and slow. New York: Farrar, Straus and Giroux.

Kátaí, Z., Osztáin, E., & Vekov, G. K. (2016). Promoting computational thinking by artistically enhanced algorithm visualization. INFODIDACT 2015: Informatika Szakmódzsertani Konferencia. https://people.inf.elte.hu/ezsva/InfoDidact16/Manuscripts/KZOEVGK.pdf. Accessed 12 Jan 2019.

Katz, A. (2010). Beginning Microsoft Excel 2010. Apress. https://doi.org/10.1007/978-1-4302-2956-8.

Knight, J. (2009). Instructional coaching (pp. 29–55). Coaching: Approaches and perspectives.

Kruck, S. E., Maher, J. J., & Barkhi, R. (2003). Framework for cognitive skill acquisition and Spreadsheet training. Journal of End User Computing, 15(1), 20–37.

Lampert, B., Pongracz, A., Sipos, J., Vehrer, A., & Horvath, I. (2018). MaxWhere VR-Learning Improves Effectiveness over Classical Tools of e-learning. Acta Polytechnica Hungarica, 15(3), 125–147. Retrieved from: http://www.uni-obuda.hu/journal/Lampert_Pongracz_Sipos_Vehrer_Horvath_82.pdf. Accessed 12 Sept 2018.
Mandula Familia Kft. (2017). Kismandula Cukrászda. http://www.kismandula.hu/klasszikus-tortak/tortak-2. Accessed 10 Aug 2017.

Microsoft (2016) Microsoft Official Academic Course Microsoft Excel 2016: Includes coverage of the following Microsoft Office Specialist (MOS) exam: MOS EXAM 77–727: EXCEL 2016. John Wiley and Sons. https://wwwdit.ie/media/ittraining/msoffice/MOAC_Excel_2016_Core.pdf Accessed 14 March 2019.

Microsoft (2019a). Excel functions (alphabetical). https://support.office.com/en-us/article/excel-functions-alphabetical-b3944572-255d-4efb-bb96-c6d90033e188. Accessed 11 Sept 2019.

Microsoft (2019b). Spreadsheet Software - Excel Free Trial - Microsoft Excel. https://products.office.com/en-us/excel. Accessed 21 January 2019.

Microsoft (2019c). Dynamic array formulas in non-dynamic aware Excel. https://support.office.com/en-us/article/dynamic-array-formulas-in-non-dynamic-aware-excel-696e164e-306b-4282-ae9d-a88f5502fa2. Accessed 11 Sept 2019.

MISTEMS Ltd. (2019). MaxWhere Store - VR workspaces. http://www.maxwhere.com/. Accessed 4 Jan 2019.

Moyo, M., Tiba, A. C., & Madzima K. (2016). Impact of Pre-Service Teachers’ Prior Knowledge of Information Technologies on Perceptions and Beliefs on Computers in Education Modules. http://uir.unisa.ac.za/bitstream/handle/10500/22888/Moses%20Moyo%2C%20Anyen%20Chantylee%20Tiba%2C%20Kudakwashe%20Madzima.pdf?sequence=1&isAllowed=y. Accessed 3 Feb 2019.

National Research Council. (2000). How people learn: Brain, mind, experience, & school (Expanded ed.). Washington: National Academies Press.

OFI (2004). [Countries], Informatics, Mock Examination 2004. https://www.oktatas.hu/bin/content/dload/erettseg/probaerettseg_2004/info_em_flap.zip. Accessed 23 June 2017.

OFI (2012). National Base Curriculum. In Hungarian: Nemzeti alaptanterv. Budapest: Oktatáskutató és Fejlesztő Intézet. http://ofi.hu/sites/default/files/attachments/mk_nat_20121.pdf. Accessed 22 Oct 2016.

OFI (2013). Central curriculum framework for year 9–12 students. In Hungarian: Kerettanterv a gimnáziumok 9–12. évfolyama számára. Budapest: Oktatáskutató és Fejlesztő Intézet. http://kerettanterv.ofi.hu/03_melleklet_9-12/index_4_gimn.html. Accessed 09 Nov 2016.

Osztian, E., Kátaí, Z., & Vekov, G. K. (2017). Multi-Dimensional Expansion of Algo-Rythmics. The Turkish Online Journal of Educational Technology, INTE 2017 November, 573–578. http://www.tojet.net/special/2017_11_2.pdf. Accessed 10 Jan 2019.

Panko, R. (2013). The Cognitive Science of Spreadsheet Errors: Why Thinking is Bad. Proceedings of the 46th Hawaii International Conference on System Sciences. Maui.

Panko, R. (2016). What We Don’t Know About Spreadsheet Errors Today: The Facts, Why We Don’t Believe Them, and What We Need to Do. arXiv preprint. http://arxiv.org/abs/1602.02601. Accessed 21 Jul 2016.

Panko, R., & Port, D. (2013). End user computing: The dark matter (and dark energy) of corporate it. Journal of Organizational and End User Computing, 25(3), 1–19.

Pólya, G. (1954). How To Solve It. A New Aspect of Mathematical Method. 2nd edition, 1957. New Jersey: Princeton, Princeton University Press.

Rogers, J. (2012). Coaching skills: A handbook. McGraw-Hill Education: United Kingdom.

Sebestyén K., Csapó G., & Csemoch M. (2018). Visualising Sprego Inequality Problems With 2D Representations. The Turkish Online Journal of Educational Technology, INTE 2018 november (2), 888–898. http://www.tojet.net/special/2018_12_3.pdf. Accessed 21 Jan 2019.

Siersted, M. (2018). Does the teaching of spreadsheeting skills have the potential to be a significant global education policy intervention? Dissertation for the Degree of MSc International Public Policy Analysis. Department of Social and Policy Sciences, Faculty of Humanities and Social Sciences, University of Bath. (Accepted).

Skemp, R. (1971). The psychology of learning mathematics. New Jersey: Lawrence Erlbaum Associates.

Test ECDL. (2019a). Exam simulation ECDL Module 4 | Spreadsheets. https://www.testecdl.com/module-4-spreadsheets.html. Accessed 22 Feb 2019.

Test ECDL. (2019b). Test 1 - Module 4 | Spreadsheets. https://www.testecdl.com/test-1-module-4-spreadsheets.html?view=quiz. Accessed 22 Feb 2019.

Test ECDL. (2019c). Test 5 - Module 4 | Spreadsheets. https://www.testecdl.com/test-5-module-4-spreadsheets.html?view=quiz. Accessed 22 Feb 2019.

The Document Foundation (2019). Home | LibreOffice - Free Office Suite - Fun Project - Fantastic People. https://www.libreoffice.org/. Accessed 14 Jan 2019.

van Merriënboer, J. J. G., & Sweller, J. (2005). Cognitive load theory and complex learning: Recent developments and future directions. Educational Psychology Review, 17(2), 147–177. https://doi.org/10.1007/s10648-005-3951-0.
Van Nieuwerburgh, C. (2012). *Coaching in education: Getting better results for students, educators, and parents*. London: Karnac Books.

Wakeling, D. (2007). Spreadsheet functional programming. *Journal of Functional Programming, 17*(1), 131–143.

Walkenbach, J. (2010). *Excel 2010 Bible*. Indianapolis: Wiley Publishing.

Williams, C. (2019). Dynamic Arrays in Excel. European Spreadsheet Risk Interest Group Conference, EuSpRIG2019.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM, 49*(3), 33–35. https://doi.org/10.1145/1118178.1118215.

**Publisher’s note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Affiliations**

Gábor Csapó¹ · Mária Csernoch¹ · Kálmán Abari²

Mária Csernoch
csernoch.maria@inf.unideb.hu

Kálmán Abari
abari.kalman@arts.unideb.hu

¹ Faculty of Informatics, University of Debrecen, 26 Kassai út, Debrecen 4028, Hungary

² Institute of Psychology, University of Debrecen, 1 Egyetem tér, Debrecen 4032, Hungary