

# Mathability in the Fields of 3D Printing and Modeling

**Marianna Zichar**

Faculty of Informatics, University of Debrecen, Egyetem tér 1, 4032 Debrecen, Hungary, zichar.marianna@inf.unideb.hu

---

*Abstract: Each term in the title is relatively new, but all of them emerge more and more frequently in various contexts. News regularly introduces some interesting events from the world of 3D printing (that is strongly related to 3D modeling) and Mathability also has an impact on many scientific research projects. This paper makes an attempt to explore how Mathability applies in the field of 3D modeling, by examining the features of Tinkercad. The investigation is completed by analyzing assignments submitted by students. My objective was to get some impression of the extent, of the students use, of their algorithmic skills, in 3D modeling and to explore whether a non-typical programming interface encourages or discourages the application of their programming skills. The results show that the students who majored in the Computer Science Bachelor Program, had a weak affinity, to demonstrate how they could apply tools of higher Mathability, in Tinkercad and thus, the unusual framework and the task did not sufficiently motivate them.*

*Keywords: 3D printing; 3D modeling; Tinkercad; Mathability*

---

## 1 Introduction

The word *mathematics* may sound frightening for most people although human life cannot be imagined without applying mathematical knowledge. Children of a very young age start to get to know the world of numbers and this learning curve does not terminate at least till the end of their education. The skills related to mathematics are applied and developed through many different courses and activities, while they are essential to be educated in other fields such as physics, and chemistry.

Human ability and the knowledge of something, differ from each other and have always been subjects of research. Some typical questions from this field: in what extent can be a skill developed, or are there differences in the measure of possible development concerning distinct disciplines [1] [2]? Besides, numerous studies examined for example relations between gender and mathematical ability, which

have attracted intensive attention because of the well-known stereotypes about the possible differences [3] [4]. Some typical examples for stereotypes mentioned frequently: women cannot be talented at mathematics, men are better engineers, to be a pilot is not for women [5]. Although society is struggling to demolish these beliefs, they have impacts on many decisions even if they cause some damage.

But disciplines have been evolving (which is a must since they have to reflect on new achievements) and new terms, aspects, points of view appear making it possible to consider new characteristics and starting novel examinations. A similar process happened in 2012 when a new field of science called *cognitive infocommunication* (CogInfoCom) was born. According to the authors it “explores the link between the research areas of infocommunications and cognitive sciences, as well as the various engineering applications which have emerged as a synergic combination of these sciences” [6]. The scientific community justified the *raison d'être* and more and more scholars have extended their works with examining cognitive aspects of their results as well [7-9]. The interdisciplinary characteristics of this special field are demonstrated with a great number of research works dealing with rather interesting topics, such as cognitive data visualization [10-14], cognitive aspects of human-machine interface systems [15-19], virtual and augmented reality [20-22], or even advanced manufacturing technology [23] [24]. This increased interest was a good reason for giving birth to a comprehensive overview of cognitive infocommunications [25].

The concept of traditional Mathability was redefined and extended in the context of cognitive infocommunication with the objective of investigating any combination of artificial and natural cognitive capabilities relevant to mathematics [26]. The vision of the authors is that infocommunication devices will be characterized by levels of mathematical capabilities. Thus, not only classical physical properties of these devices will be overviewed before selecting the appropriate one for a specific purpose, but this newly defined measure will be taken into consideration too. We have to notice that some properties (mainly related to the hardware) could provide some hints about the mathematical capabilities of devices earlier, but this new, more general measurement can be used to get much more accurate information about it. A device without a driver or software products running on it is nothing else than a pure piece of hardware. Thus, Mathability depends also on the software that has the device run. There are specific programs designed to solve directly mathematical problems, but some of them do not occur isolated but are embedded into a more complex system. Examples for the first case are described in [26], while the other interpretations of Mathability are discussed in numerous papers presented in the next section. The levels of Mathability are different of course, but in each case investigation on Mathability provides an added value to get a complex characterization of the mathematical capability of the device.

These days, the world around us changes dynamically and this change influences the job market. The news tells us that robots and artificial intelligence applications

are going to have an increasingly important role, new job types are being given birth while the way of employment is also transforming: more people work from home, or take part-time jobs or become freelancers. In this fast-changing world, the role of knowledge is appreciated and research works dealing with facilitating knowledge transfer are also set in focus.

This paper first, provides an overview of what we already know about Mathability, then, extends the investigation, to the new field of 3D modeling. 3D modeling has not yet been analyzed from the point of view of Mathability. Finally, results of an experiment are presented, where university students enrolled in a course of 3D printing and modeling were involved to provide information on how much they are ready to host greater Mathability levels.

## **2 Diversity in Research Works Related to Mathability**

As mentioned earlier the scientific community has shown a great interest in hosting the redefined concept of Mathability, which is proved by the significant number of papers touching or focusing on Mathability. This section provides a classified overview about a relevant part of published works to understand more clearly the state-of-the-art in this field. Classification is based on the main objectives of investigations that are all related to mathematics to some extent. First, the strictly mathematics-oriented works are presented, then followed by the educational aspects of Mathability.

### **2.1 Mathematics Measured in Term of Mathability**

Solving mathematical problems is often supported with computer-based applications, in which Mathability can be investigated. The complexity of mathematical problems may vary on a large scale thus; the capability of applications can also be diverse.

Solving functional equations and inequalities does not belong to the simplest problems in mathematics that is why several computer algebra systems provide help in dealing with them. A small team of researchers who used Maple for solving problems in the above-mentioned field showed examples of how programs developed in Maple are capable of computing solutions for some classes of functional equations. They demonstrated that the programs increased the level of Mathability of the computer algebra system Maple [27].

Another research work demonstrates an animation developed in the Wolfram Mathematica System to visualize the m-convex hull of sets consisting of finitely

many points on the Cartesian plane. The computer-assisted method uses the features of the host system, but finally, it is the intellectual product of the authors considering that they developed a specific program for the animation that is an added value comparing to the original functionality of the system [28].

## **2.2 Education**

Many scholars who are working in educational institutions share their time between research and education. Thus, educational aspects of new theories, concepts, and procedures are often explored and analyzed too. It is trivial that education needs special attention from the scientific community, otherwise we are going to lack the new generation of scholars. Several papers have analyzed the general relationship between education and Mathability [29-31]. This section gives an overview of some distinct fields of education where Mathability has already been investigated.

### **2.2.1 General Aspects**

Before discussing the different fields in education, where Mathability can be interpreted, a more general approach has to be mentioned. Some researchers found that mathabililty could be considered general in terms of computer problem-solving. First, SOLO (Structure of the Observed Learning Outcome) categories of understanding were matched to their typology consisting of approaches used to solve a problem with a computer. Finally, levels of Mathability were assigned to the items of their typology. Although both typologies have 5 categories, matching resulted in only four pairs [32].

### **2.2.2 Spreadsheet**

A study put the spreadsheet tools in the focus of Mathability examinations since the authors found that spreadsheet programs can be used on any level of Mathability [33]. Besides determining the Mathability level of the help surface, click-on usage, or course books, the researchers introduced Sprego that is a concept- and CAAD-based approach to problem-solving in spreadsheet management. Several examples demonstrate in the paper how programming skills can be used when following Sprego methodology, and to what extent algorithmic skills can be developed. To sum up, the appropriate application of Sprego can increase the Mathability level of spreadsheet tools, which finally improves the effectiveness of the application.

### 2.2.3 Programming

Programming and mathematics are in a close relationship. Thus, it is not surprising that some scientists made research on solving word problems by computer programming. They pointed out that there were certain analogies between classical mathematical problem-solving methods and the steps of thinking when implementing an algorithm for a specific problem. They not only introduce their teaching method but also analyze it from the point of view of Mathability [34].

### 2.2.4 Mathematics

The relationship between Mathability and mathematics is of high interest not only at the level of mathematicians but also at the level of teaching mathematics. As it was mentioned in the introduction, acquiring knowledge of mathematics is essential for everybody. The depth of the necessary amount of knowledge is different of course, but looking for effective methodologies is always in the focus. Several papers discuss questions related to the quantification of artificial mathematical capabilities and education [27] [35]. Authors suggest that an exact quantification of Mathability of devices, such as calculators, smartphones would be a great help when we have to decide which device is allowed to be used in special situations like an exam. In the study [35], some examples demonstrate how the application of devices with high-level Mathability can support the work of students with difficulties in understanding higher mathematics.

Another arena of mathematics studies is participating in competitions. There is a competition called MaTech where the main goal is to measure the creativity, problem-solving skills, efficiency of teamwork, and application of digital knowledge while solving real mathematical problems. Authors of paper [36] as organizers of this competition present an analysis that identifies the characteristics of the mathematical knowledge and the relationship with the skill of creative presentation and performance.

## 3 Mathability in 3D Design

Activities in general, including studying, working, traveling, cooking, etc. require an effective collaboration of several skills and knowledge we have or should have. Our success depends on how we combine our different skills and knowledge to find the solution for a problem and to do the necessary steps. Our capabilities are not distinct from each other, they interact and help each other via knowledge transfer.

The SOLO categories of understanding were defined for programming, however this typology is more general and can be extended to any computer related-activities [32]. Three-dimensional modeling is a challenging field of producing models that can serve various purposes. Animations, virtualizations, and engineering use 3D models intensively, and new emerging technologies such as 3D printing also need those [37]. The unique properties of 3D printed objects attract several scientists to research how these exceptional characteristics could be beneficial for their disciplines. But first, an appropriate model has to be designed with 3D modeling software, which process can be and should be introduced into education at several levels [38].

The variegation of a 3D modeling software called Tinkercad motivated me to investigate its Mathability. The following sections introduce and analyze its several interfaces that are followed by discussing an experiment about the willingness of students to adopt higher-level Mathability tools.

### **3.1 A Scalable 3D Modeling Software -Tinkercad**

Tinkercad is a popular, easy-to-use 3D CAD design tool from Autodesk. The primary goal of designing in Tinkercad is to prepare a 3D model for 3D printing. Fortunately, the availability of the simplest, inexpensive desktop FDM (Fused Deposition Modeling) 3D printers, has increased significantly, thus the challenge of making a design has become more attractive for many people. Numerous educational institutions have introduced 3D printing and modeling into their teaching programs, although their exact forms can be diverse. At the same time, we can find pioneers among educators from almost all disciplines who try to benefit from the advantages 3D printing can provide for them [39-42].

There are several reasons why the majority of users who want to learn 3D modeling, select Tinkercad as the first tool to study, these reasons are:

- **Platform**  
As one of the most accessible 3D modeling software around, this online CAD tool can be accessed through your browser giving maximum flexibility in the field of platforms. Your private cloud stores your models that also can be shared easily.
- **Free**  
Tinkercad was designed for mainly educational purposes, thus it was essential to make it free. Educational institutions and people usually try to avoid using proprietary software.
- **Scalability**  
Although the first impression after entering your Tinkercad account is that mainly the young generation will enjoy it, that is not true. Applying the principle of CSM (Constructive Solid Modeling) is just one approach

designing is based on. The possibility of coding at different levels is also available providing the opportunity of matching the tool to the proficiency of the user.

- **Users**  
Scalability implies that a wide range of people can use Tinkercad. While children will operate with dragging and dropping primitives onto the workplane, shape generators already offer the option of sophisticated parametrization, and coding provides unlimited possibilities for users with enough mathematical knowledge.
- **Community**  
A large, vivid, and supportive online community provides help, advice, and plenty of educational resources for users. A design can be shared easily with other users without the need of downloading the model, since models are stored in the cloud.

At different rankings, Tinkercad has always had a good position. According to the last survey of All3DP (leading 3D printing magazine), Tinkercad is the first on the Top 10 Free 3D Modeling Software for Beginners published in January 2020 [43]. Although this list ranked design software for only beginners, Tinkercad can provide an appropriate interface for users from novice till professional and can be integrated into the education easily [44-47].

## 3.2 Mathability Levels in Tinkercad

Mathability of a device or software communicates information about their capability related to mathematics. The usage of many software can have two different forms. The simplest one relies only on built-in tools, while the other form is about to create something new, such as scripts, codes, functions, which finally enhance the original capabilities of the system, and demands deeper knowledge, and more creativity. Three-dimensional modeling requires many computations, thus investigating Mathability of such software is reasonable.

### 3.2.1 Built-in Tools

As Figure 1 shows, the first impression about Tinkercad, is that it supports the CSM by providing several predefined primitives that form groups, such as, Basic Shapes, Text and numbers, Characters and Connectors. Unfortunately, these primitives can be combined by applying only two operations: union and difference. To determine the intersection of two primitives (that is a frequently used operator) requires following several steps. The simplest design process, based on building with primitives and operators, is of a low-Mathability.

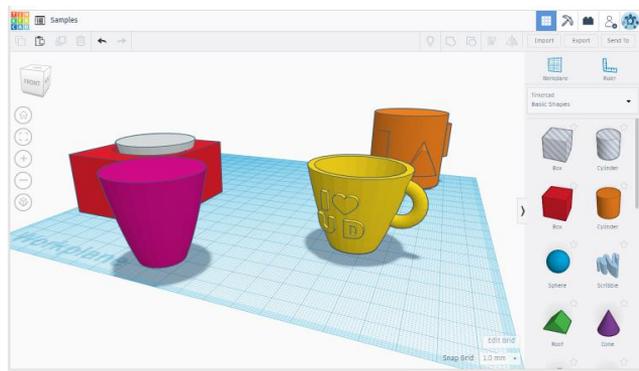


Figure 1

General view of Tinkercad's user interface

Although even basic shapes can be controlled by some parameters, more complex, still built-in tools are available in Tinkercad. They are listed under the group of Shape Generator, and typically provide more sophisticated parameters, and an interactive interface to change their default appearance. For example, Figure 2 represents some settings of a circular array, where the shape of the items arranged around a circle is determined by a sketch. In this case, four points as knots and endpoints of their tangents can be used to influence the shape of items. These tools are still embedded, but their conscious usage requires already more mathematical knowledge, thus its level of Mathability is greater.

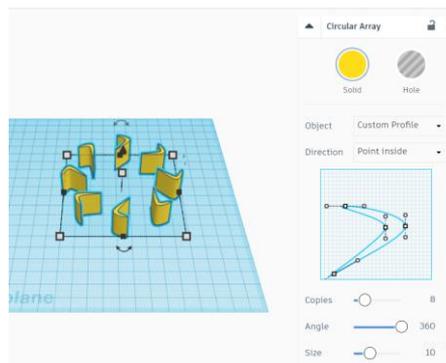


Figure 2

A shape generator in use

### 3.2.2 Codeblocks

Tinkercad Codeblocks, released in 2018, allows for procedurally generated shapes, using visual programming. To design things in Codeblocks, you simply snap blocks of code together to form a series of sequential commands or actions.

These actions control the shape of your model and result in a 3D design that appears in the 3D viewer. Using Codeblocks can save time, especially when repeating actions form a model, or when manual shape creation, could take an excessive amount of time.

There are six categories of blocks:

- **Shapes**  
Currently, 17 different 3D primitives (such as box, cylinder, sphere, cone, text, etc.) are available that have some parameters to control their exact shape. Every shape is loaded at the origin point at the center of the workspace.
- **Modify**  
This set of blocks contains basic transformations (such as move, scale, rotate), and also blocks to define an object (an individual item of the design identified by a name), to work with an object (such as copy, delete, select all), and to make a grouping.
- **Control**  
Two loop types are available (repeat and count controlled), and one more block for making a pause.
- **Math**  
These blocks can be used to define a variable, to set its value, to generate random numbers, to specify a coordinate triplet, to use basic mathematical functions (such as trigonometrical, logarithmic, rounding, etc.)
- **Data**  
User-defined local variables are listed in this block in order to make it possible to refer them in the blocks.
- **Markup**  
It contains only two blocks: making comments and text outputs.

After having the first impression of the blocks, it is clear that this computational design workspace provides a special interface to automate building from primitives and can save human effort when building a model consisting of many items. Visual programming is attractive for young generations, but people with programming skills can have fun with it and create complex models easily (Figure 3). Designing with Codeblocks can motivate students toward STEAM (Science, Technology, Engineering, Art, Math) studies and can definitely improve their algorithmic skills, which can be rather beneficial for them, by transferring their knowledge to other fields.



Figure 3

A simple design in Codeblocks

Concerning Mathability of Codeblocks, it has to be noted, that it has already a higher level of Mathability, than the classical workspace introduced in the previous section. Nevertheless, the opportunity for applying variables also contributes to increasing the level of Mathability, since designing with computations requires a more complex way of thinking. Usage of parameters in the control drawer (that can be expanded or collapsed by clicking the arrow) supports minimizing the number of available shapes and introduces a generalization in determining the real form. This can be embarrassing if we consider that the same shape block can result in significantly different outlines depending on the parameter values [48]. Figure 4 represents two pieces of torus differing from each other in only one parameter: the number of sides.



Figure 4

Two different parametrizations of a torus

### 3.2.3 JavaScript

Although Tinkercad provides many generic geometric shapes that can be combined in many ways to create more complex models, you may need other shapes. The set of shapes can be extended by defining new ones that are created on the creative platform of Tinkercad using JavaScript. If there is a series of shapes you combine frequently, and you describe them with coding in JavaScript then you have created a new shape generator that allows you to automate the

creation of the combined model. Already existing shape generators can be customized too, with editing, or also a brand new one can be created from scratch if you have enough knowledge of mathematics and coding.



```

1 // Convenience Declarations For Dependencies.
2 // 'Core' is Configured In Libraries Section.
3 // Some of these may not be used by this example.
4 var Conversions = Core.Conversions;
5 var Debug = Core.Debug;
6 var Path2D = Core.Path2D;
7 var Point2D = Core.Point2D;
8 var Point3D = Core.Point3D;
9 var Matrix2D = Core.Matrix2D;
10 var Matrix3D = Core.Matrix3D;
11 var Mesh3D = Core.Mesh3D;
12 var Plugin = Core.Plugin;
13 var Tess = Core.Tess;
14 var Sketch2D = Core.Sketch2D;
15 var Solid = Core.Solid;
16 var Vector2D = Core.Vector2D;
17 var Vector3D = Core.Vector3D;
18
19 // Template Code:
20 /*
21 Empty shape example.
22
23 Tinkercad developer documentation is at:
24 https://tinkercad.com/developer/
25
26 To create parameters in the user interface,
27 create a 'params' array on the top level of this script.
28 For example:
29 params = [
30   {
31     "id": "radius", "displayName": "Radius", "type": "length", "rangeMin": 1, "rangeMax": 50,
32   }
33 */
34 function process(params) {
35   var mesh = new Mesh3D();
36
37   Plugin.warning("This shape script is empty.");
38
39   return Solid.make(mesh);
40 }

```

Figure 4

Empty Shape Generator with the pre-populated code lines

In Tinkercad, this platform has the highest level of Mathability and requires the deepest professional skills from the user. At least a basic understanding of JavaScript is necessary, while the depth of geometry knowledge influences the complexity of a newly developed shape generator. According to the classification in paper [26], Mathability provided by a system can be interpreted in two ways depending on applying only existed tools of the system or enhancing the original set of tools. In the case of writing codes, we face the second approach, while using a shape generator created by anybody else, and shared with us is already belongs to the first class only.

### 3.3 Experiment with Students

Every semester I have a course called *3D printing and modeling* at the University of Debrecen, where Tinkercad is the first design software taught in classes. The topic of the course is rather attractive for the students and I try to do my best to keep the curriculum interesting and up to date [49]. The spring semester of 2019/20 was the first when I introduced also Codeblocks to the students. The power of control structures, the possibility for parametrization with the help of defining variables were emphasized since these tools provide the main advantages of using Codeblocks. After completing the classes about Codeblocks, I asked them to design something for the sake of demonstrating the capability of Codeblocks, to practice, and to have some fun. All the students were enrolled in Computer Science bachelor degree program, so they were familiar with coding, although they were not enrolled in the same year because this course is optional.

I was expecting stunning models with high complexity, large diversity in applied block types, clear proof of algorithmic skills, and no significant difference between the performance of Hungarian and International Students.

After collecting the URL of models, I have analyzed the codeblocks, and created some statistics to answer some questions about the complexity, number of control statements, degree of diversity. This section demonstrates the most compelling results.

### 3.3.1 Complexity

The instruction for the assignment was simple: demonstrate the capability of Codeblocks, consider the visual coding as a game, and use the power of loops. Although I thought that the instructions were clear, finally half of the assignments had to be discarded. Unfortunately, some of them ignored the task, while others either copied a sample code or just made a minor modification in a shared code. So, I could examine 15 acceptable assignments.

The first aspect I was interested in examining is the complexity of their codes. Since students are experienced in coding, the tool is not complicated for their programming skills, and the course, is optional. I did expect attractive models. Just for the sake of curiosity, I counted the number of code lines, loops, and shapes generated in loops (Figure 5). It is clear that loops were not preferred, and most students did not want to use programming tools of higher Mathability levels. Only two of them introduced objects that are separated set of codes that could be referenced by their name.

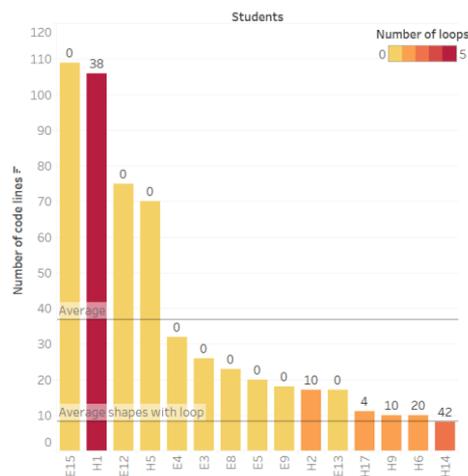


Figure 5

Number of lines, used loops, and shapes generated in loops (number on the bars) by students

### 3.3.2 Application of their Programming Skills

Three-dimensional designing makes people think of working in a visual environment without writing even one line of code. When introducing the possibility of coding for students majored in computer science you may think, that this way of making 3D models is more comfortable for them. Although the range of classical controlling statements in Codeblocks is limited (two types of loops), other tools (such as variables, work with objects) provide the possibility to use algorithms when building a model. This was my motivation to compare the number of shapes generated in and outside the loops (Figure 6). Concerning the number of shapes coming from a loop, there were two outliers describing two different code structures. The student with the highest number of generated shapes had no shapes outside the loop, while the student in the second place used the second largest number of shapes without loops. Unfortunately, students were not interested in applying their general programming skills, which was surprising.

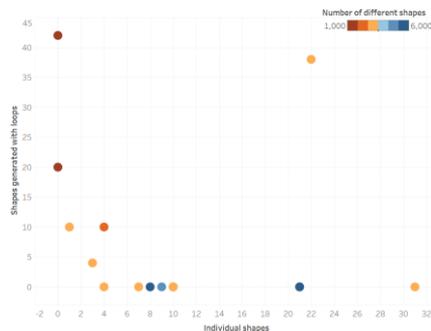


Figure 6

The number of shapes generated in and outside the loops

### 3.3.3 Variety of Design

Currently, the number of the available shapes in Codeblocks is 17 that is quite impressive. Nevertheless, the number of visually really different shapes is even higher, because the flexible parametrization can result in a significant change in the geometry of the shape as you could see in Figure 4. The students used 13 shapes out of the 17 in their works, but the average number of the different shapes per student was only 3.26 (Figure 7). This shows that the variety of shapes did not motivate them to use more shapes.

We can also consider the Figure 5 and 7 together to look for additional findings. The students who preferred using different shapes and functions did not insert any loops into their codes. The longest codes contained only the average number of different functions and shapes. The model with the highest number of shapes coming from loops consisted of the smallest number of code lines and used only

one type of shape and three different function types. The work of student H1 was remarkable, since it was the second longest code, with a high number of shapes generated in several loops, however, the number of different functions and shapes was average.

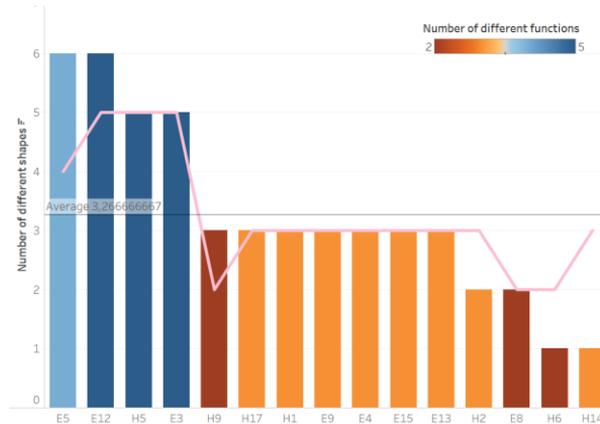


Figure 7

The number of different shapes colored with number of different functions

Figure 8 shows the total number of students per using different shape types. Three shapes (cylinder, box, and text) gave the half portion of the used shapes, which demonstrates again that the students were not engaged in diversity. They preferred using the already known shapes, with different parametrization.

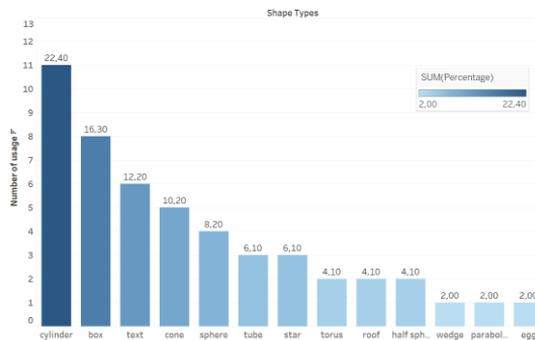


Figure 8

The number students selected the different shapes, and their contributions to the whole

Due to the small number of evaluable codes, I have not analyzed the difference between the performance of Hungarian and International students. Nevertheless, I have to remark here, at least one phenomenon: International students did not use loops or variables at all.

## Conclusions

After a thorough Literature Review of the current research related to the concept of Mathability, this paper explored the relationship between Mathability and the emerging technology of, 3D printing and modeling. The concept and the levels of Mathability can also be interpreted in the context of 3D modeling, which was proven by examining the scalability of the popular modeling software product, Tinkercad. It is clear that the concept interpretation is wide, and several levels can be identified. Related to the levels of Mathability, an experiment was carried out, where assignments of university students, majoring in computer science, were analyzed to check their affinity for applying tools of a higher Mathability in Tinkercad. The results showed that my instructions and suggestions were not motivating enough as the students preferred keeping the models very simple and avoided using their programming skills. It has to be mentioned, that the students used the graphical designing interface of Tinkercad, for four weeks, which could have a negative impact on their willingness to use coding, even if Codeblock generation has a user-friendly interface. At the same time, some of them felt that use of Codeblocks was not designed for users with programming skills, which is not true. Complex models cannot be designed in Tinkercad by only using the traditional interface.

I plan to repeat the experiment in the upcoming years, with more clear and probably, more effective instructions. In the case of a higher number of evaluable codes, more aspects should be analyzed. Also, a follow-up questionnaire could provide more information concerning the reasons for a students' activity or inactivity in coding. Data visualization could be enhanced by online interactive dashboards, that would facilitate the identification of more complex relationships among the data.

## Acknowledgement

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.

## References

- [1] B. Göransson, I. Josefson (Eds.): Knowledge, skill and artificial intelligence, Springer Science & Business Media, 2012
- [2] O. Uzumcu, E. Bay: The effect of computational thinking skill program design developed according to interest driven creator theory on prospective teachers. *Education and Information Technologies*, Vol. 26, 2021, pp. 565-583
- [3] S. A. Sorby, C. Leopold, R. A. Górská: Cross-Cultural Comparisons of Gender Differences in the Spatial Skills of Engineering Students, *Journal of Women and Minorities in Science and Engineering*, Vol. 5, No. 3, 1999, pp. 279-281

- 
- [4] B. Nemeth, M. Hoffmann: Gender differences in spatial visualization among engineering students, *Annales Mathematicae et Informaticae* Vol. 33, 2006, pp. 169-174
- [5] A. Powell, A. Dainty, B. Bagilhole: Gender stereotypes among women engineering and technology students in the UK: lessons from career choice narratives. *European Journal of Engineering Education*, 37(6), 2012, pp. 541-556
- [6] P. Baranyi, Á. Csapó: Definition and Synergies of Cognitive Infocommunications, *Acta Polytechnica Hungarica*, 9(1), 2012, pp. 67-83
- [7] I. Papp, R. Tornai, M. Zichar: What 3D technologies can bring into the education: The impacts of acquiring a 3D printer, *Proceedings of 7<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Wroclaw, 2016, pp. 257-261
- [8] P. T. Kovács, T. B. Holografika: 3D Display technologies and effects on the human vision system, *Proceedings of 2<sup>nd</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Budapest, 2011, pp. 1-33
- [9] Sz. Szeghalmy, M. Zichar, A. Fazekas: Gesture-based computer mouse using Kinect sensor, *Proceedings of 5<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Vietri sul Mare, 2014, pp. 419-424
- [10] Gy. Papp, I. Papp, R. Kunkli: Three-dimensional connection visualization based on tabular data, *Proceedings of 8<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Debrecen, 2017, pp. 289-290
- [11] R. Idzikowski, K. Kluwak, T. Nowobilski, T. Zamojski: Analysis of possibility of visualization of danger factors in the building environment, *Proceedings of 8<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Debrecen, 2017, pp. 363-368
- [12] Á. Török, Zs. Gy. Török, B. Tölgyesi: Cluttered centres: interaction between eccentricity and clutter in attracting visual attention of readers of a 16<sup>th</sup> century map, *Proceedings of 8<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Debrecen, 2017, pp. 433-438
- [13] D. Balla, T. Mester, Á. Botos, T. J. Novák, M. Zichar, J. Rásó, A. Karika: Possibilities of spatial data visualization with web technologies for cognitive interpretation, *Proceedings of 8<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Debrecen, 2017, pp. 17-20
- [14] D. Sik, K. Csorba, P. Ekler: Implementation of a Geographic Information System with Big Data Environment on Common Data Model, *Proceedings of 8<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Debrecen, 2017, pp. 181-184

- 
- [15] A. Kővári, J. Katona, C. Costescu: Evaluation of Eye-Movement Metrics in a Software Debugging Task using GP3 Eye Tracker, *Acta Polytechnica Hungarica*, 17(2), 2020, pp. 57-76
- [16] A. Kővári, J. Katona, C. Costescu (2020): Quantitative Analysis of Relationship Between Visual Attention and Eye-Hand Coordination, *Acta Polytechnica Hungarica*, Vol. 17, No. 2, 2020, pp. 77-95
- [17] Á. Török: From human-computer interaction to cognitive infocommunications: a cognitive science perspective, *Proceeding of 7<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Wroclaw, 2016, pp. 433-438
- [18] A. Fazekas, K. Bertók: Face recognition on mobile platforms, *Proceedings of 7<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Wroclaw, 2016, pp. 37-42
- [19] Sz. Szeghalmy, M. Zichar, A. Fazekas: Gesture-based computer mouse using Kinect sensor, *Proceedings of 5<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Vietri sul Mare, 2014, pp. 419-424
- [20] I. Horváth: Innovative engineering education in the cooperative VR environment, *Proceedings of 7<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Wrocław, 2016, pp. 359-364
- [21] Gy. Bujdosó, O. C. Novac, T. Szimkovic: Developing cognitive processes for improving inventive thinking in system development using a collaborative virtual reality system, *Proceedings of 8<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom)*, Debrecen, 2017, 79-84
- [22] M. Csernoch, Gy. Bujdosó, M. Borbély, E. Dani, M. Némethi-Takács, K. Koltay, L. Balázs: LibSearchNet: Analyses of library log files to identify search flows, *Proceedings of 4<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, Budapest, 2013, pp. 543-548
- [23] F. Erdős, R. Németh: AMT-based Real-Time, Inter-Cognitive Communication Model, *Acta Polytechnica Hungarica*, 16(6), 2019, pp. 115-127
- [24] I. Papp, M. Zichar: 3D Modeling and Printing Interpreted in Terms of Cognitive Infocommunication In: Klemm, Ryszard; Nikodem, Jan; Zoltán Baranyi, Péter (ed.) *Cognitive Infocommunications, Theory and Applications* Cham (Switzerland), Switzerland: Springer International Publishing, 2019, pp. 365-389
- [25] P. Baranyi, Á. Csapó, Gy. Sallai: *Cognitive Infocommunications (CogInfoCom)*, Springer, 2015
- [26] P. Baranyi, A. Gilányi: Mathability: emulating and enhancing human mathematical capabilities, in *Proceedings of 4<sup>th</sup> IEEE Conference on*

- Cognitive Infocommunications (CogInfoCom), Budapest, 2013, pp. 555-558
- [27] G. Gy. Borus, A. Gilányi: Solving systems of linear functional equations with computer, Proceedings of 4<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Budapest, 2013, pp. 559-562
- [28] A. Gilányi, N. Merentes, R. Quintero: Mathability and an animation related to a convex-like property, Proceedings of 7<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Wrocław, 2016, pp. 227-232
- [29] K. Chmielewska, A. Gilányi: Educational context of Mathability, Acta Polytechnica Hungarica, 15, 2019, pp. 223-237
- [30] K. Chmielewska, A. Gilányi, A. Łukasiewicz: Mathability and Mathematical Cognition, Proceedings of 7<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Wrocław, 2016, pp. 245-250
- [31] K. Chmielewska, D. Matuszak: Mathability and coaching, Proceedings of 8<sup>th</sup> IEEE International Conference on Cognitive Infocommunications (CogInfoCom), Debrecen, 2017, pp. 427-432
- [32] P. Biró, M. Csernoch: The Mathability of computer problem solving approaches, Proceedings of 6<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Győr, 2015, pp. 111-114
- [33] P. Biró, M. Csernoch: The Mathability of spreadsheet tools, Proceedings of 6<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Győr, 2015, pp. 105-110
- [34] K. Bubnó, V. L. Takács: Cognitive Aspects of Mathematics-aided Computer Science Teaching, Acta Polytechnica Hungarica, Vol. 16, No. 6, 2019, pp. 73-93
- [35] K. Chmielewska, A. Gilányi: Mathability and computer aided mathematical education, Proceedings of 6<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Győr, 2015, pp. 473-477
- [36] A. Kovari, M. Rajcsanyi-Molnar: Mathability and Creative Problem Solving in the MaTech Math Competition, Acta Polytechnica Hungarica, Vol. 17, No. 2, 2020, pp. 147-161
- [37] F. Erdős, R. Nemeth: Inter-cognitive Communication Model Using Additive Manufacturing Technology, Proceedings of 9<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom), Budapest, 2018, pp. 217-222
- [38] I. Papp, M. Zichar: Modeling and printing in 3D at higher education, Proceedings of 10<sup>th</sup> International Conference on Applied Informatics, Eger, 2017, pp. 235-243

- [39] S. Ford, T. Minshall, Invited review article: Where and how 3D printing is used in teaching and education, *Additive Manufacturing* 25, 2019, pp. 131-150
- [40] M. Zichar: Added values of additive manufacturing to geoinformatics Proceedings of VIII. Theory meets practice in GIS, Debrecen, 2017, pp. 437-443
- [41] I. M. Alhamad, K. A. Waleed, H. Z. Ali, H. AlJassmi: 3D Printing Applications in Mechanical Engineering Education, In: Khine, Myint Swe; Ali, Nagla (Eds.) *Integrating 3D Printing into Teaching and Learning*, Brill Sense, 2020, pp. 90-131
- [42] C. Schelly, G. Anzalone, B. Wijnen, J. M. Pearce: Open-source 3-D printing technologies for education: Bringing additive manufacturing to the classroom, *Journal of Visual Languages & Computing*, 28, 2015, pp. 226-230
- [43] <https://all3dp.com/1/best-free-3d-modeling-software-for-beginners/>
- [44] L. M. Díaz, C. M. Hernández, A. V. Ortiz, L. S. Gaytán-Lugo: Tinkercad and Codeblocks in a Summer Course: an Attempt to Explain Observed Engagement and Enthusiasm, 2019 IEEE Blocks and Beyond Workshop (B&B), Memphis, TN, USA, 2019, pp. 43-47
- [45] P. R. Sajil Raj, A. Anshadh, S. B. T. Raj, A. N. Ahsana: Design of an innovative coconut grating machine using Tinkercad, *International Journal of Research in Mechanical Engineering*, Vol. 4, Issue 3, 2016, pp. 178-182
- [46] S. Yi, U. Jung, Y. Lee: A Study on the direction of 3D Modeling Education Considering Computational Thinking Factors at Elementary School in South Korea, *Proceedings of E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, Vancouver, 2017, pp. 505-508
- [47] O. Ng: Exploring the use of 3D Computer-Aided Design and 3D Printing for STEAM Learning in Mathematics, *Digital Experiences in Mathematics Education* 3, 2017, 257-263
- [48] M. Zichar, I. Papp: Interaction between 3D printing and geometry studies Proceedings of 18<sup>th</sup> International Conference on Geometry and Graphics: 40<sup>th</sup> Anniversary, Milan, 2018, pp. 1177-1190
- [49] M. Zichar: Exploring the role and possibilities of 3D printing in IT studies: Is my curriculum good enough?, *Proceedings of 10<sup>th</sup> IEEE Conference on Cognitive Infocommunications (CogInfoCom)*, 2019, pp. 221-226