Using “Turtle Geometry” in the XXIst Century for teaching mathematics? Classic ideas and new software and hardware

Eugenio Roanes-Lozano
eroanes@ucm.es
Instituto de Matemática Interdisciplinar (IMI) &
Departamento de Didáctica de las Ciencias Experimentales, Sociales y Matemáticas
Facultad de Educación
Universidad Complutense de Madrid
Spain

Abstract: *In “Turtle geometry” there is a graphic cursor (the “turtle”) that obeys elementary orders related to the position and orientation of the turtle itself: move forward, move backward, turn right and turn left. That is, graphics are not based on a classic Cartesian reference system. Turtle Geometry is best known for its incorporation into Logo language. As monitors had no graphical capabilities in the 1960s, the movements ordered to the turtle from the first Logo versions were carried out by electromechanical devices connected by cable to the computers. After a great success in the '80s and' 90s Logo has fallen into disuse, but there are also very modern visual computer languages based on the use of “blocks” for programming such as Scratch and Snap! Moreover, nowadays there are affordable programmable robots that use Turtle Geometry. Summarizing, wonderful powerful software and hardware that use Turtle Geometry, appropriate for teaching mathematics are available in 2021.*

1. Introduction

The author has taught mathematics with ICT in teacher training programs since the late 1980s using computer languages including implementations of “Turtle Geometry”. He has also developed implementations of Turtle Geometry for different computer languages (e.g. *Maple*[1]).

In Turtle Geometry (also known as “Turtle Graphics”) there is a graphic cursor (the “turtle”) that obeys elementary orders related to the position and orientation of the turtle itself. They are:

- move forward,
- move backward,
- turn right, and
- turn left.

That is, graphics are not based on a classic Cartesian reference system, and, consequently, the list of commands required to draw a certain pattern neither depends on where the pattern is to be allocated nor on its leaning. Another advantage with respect to working with Cartesian coordinates is that the trigonometric calculations regarding positioning are performed internally.

The Turtle Geometry applies constructionist ideas [1] and its range of possibilities is absolutely impressive, as can be seen in the seminal work [2].

Turtle Geometry is best known for its incorporation into *Logo* language [3]. For example, the image of Figure 1.1 is generated in *Logo* by typing `FORWARD 200 RIGHT 90 FORWARD 100`. Many regular and repetitive geometric designs (for instance some fractals) are very easy to program in *Logo*. A simple example can be found in Figure 1.2.

[1] All product names, trademarks and registered trademarks are property of their respective owners

Seymour Papert co-founded the MIT Artificial Intelligence Laboratory with Marvin Minsky in the early 1960s and *Logo* was introduced in 1967. It was a very powerful language that incorporated the latest trends: it was procedural and recursive, it handled lists, etc.

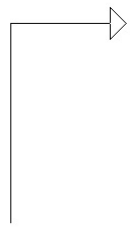

Figure 1.1 A *Logo* very simple drawing.

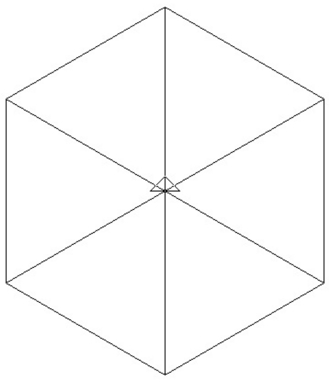

Figure 1.2 Drawing a regular hexagon and its three diagonals (that is, its six radii) in *Logo* is very simple nesting two `REPEAT` iterative loops (this way six equilateral triangles are drawn sharing a common vertex): `REPEAT 6 [REPEAT 6 [FORWARD 200 RIGHT 120] RIGHT 60]`

Meanwhile, "turtle robots" or "tortoises" were autonomous robotic creatures introduced by Grey Walter in the late 1940s [4,5]. As monitors had no graphical capabilities in the 1960s, the movements ordered to the turtle from the first *Logo* versions were carried out by electromechanical devices connected by cable to the computers [6,7].
After a great success in the '80s and' 90s *Logo* has fallen into disuse, although there are exceptions like [8].
Nevertheless, there are nowadays very friendly and powerful free versions available, such as *FMSLogo* [9] (used for drawing Figures 1.1 and 1.2) and *UCB Logo* (*Berkeley Logo*) [10]. A

detailed comprehensive list of *Logo* language implementations, mentioning more than 250 dialects, can be found in [11].

2. Present situation (software)

Apart from *Logo*, there are very modern visual languages based on the use of "blocks" for programming that base their computer graphics upon Turtle Geometry, like:

- *Scratch* [12], a project of the Lifelong Kindergarten Group at the MIT Media Lab (Figure 2.1), and
- *Snap!* [13,14], presented by the University of California at Berkeley (Figure 2.2). Although less well known than *Scracth*, it has some advantages like the existence of a `REPORT` block that allows to easily implement recursive procedures.

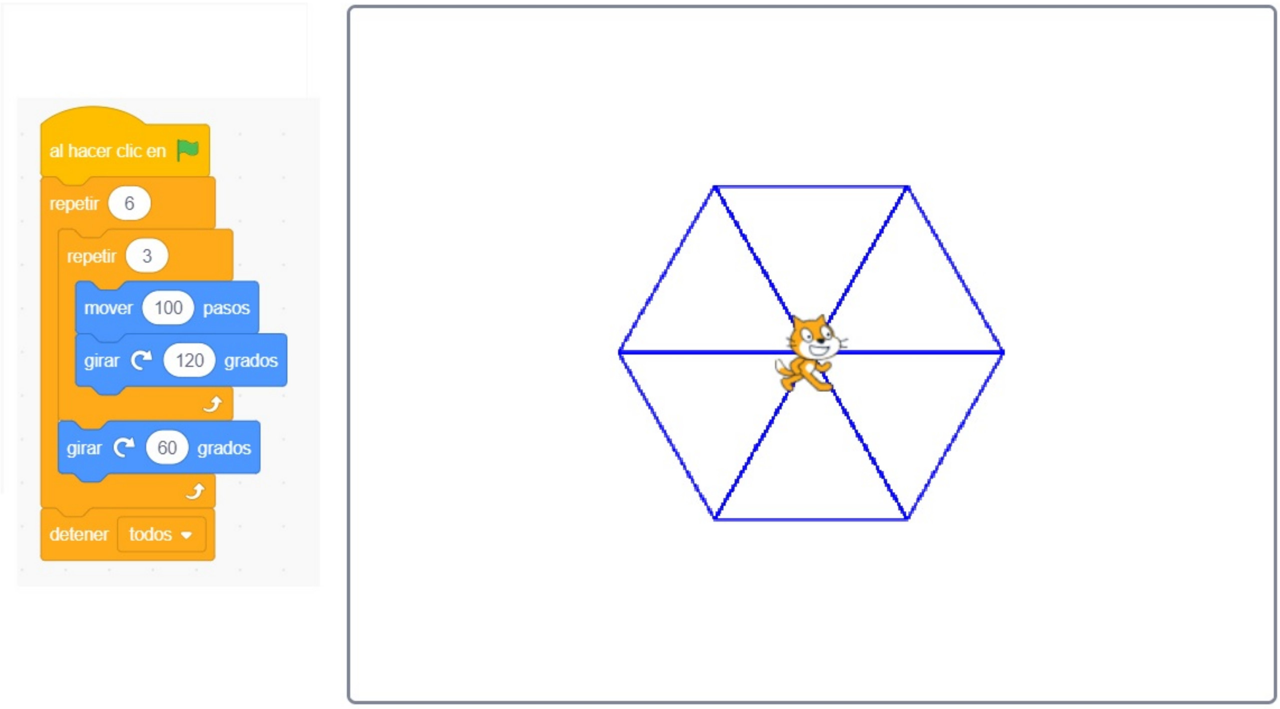

Figure 2.1 Drawing the regular hexagon and its three diagonals of Figure 1.2 with *Scracth 3* (in Spanish).

The use of blocks (that are grouped in "categories", with different colors in *Scratch* and *Snap!*) has advantages for beginners with respect to traditional programming. For instance:

- the end user doesn't have to remember the names of the commands as he/she only has to look for them in the corresponding category (that is intuitively chosen) and drag and drop them in the right position in the "programming window",
- the colors help to identify the different types of commands,
- loops are clearly visible, without having to "indent" the code (Figure 2.1),
- …

An expert programmer types faster than dragging and dropping blocks, but that is not the goal of these computer languages.

Figure 2.2 Drawing a row of square tiles with *Snap!*

A sophisticated example of the use of Turtle Geometry to draw fractal trees related to the spread of virus by this author (intended to visually explain it to children and to raise their awareness) can be found in [15] (*Scratch 3* version, see Figure 2.3) and [16] (*Maple* version, implemented on the Turtle Graphics implementation [17]).

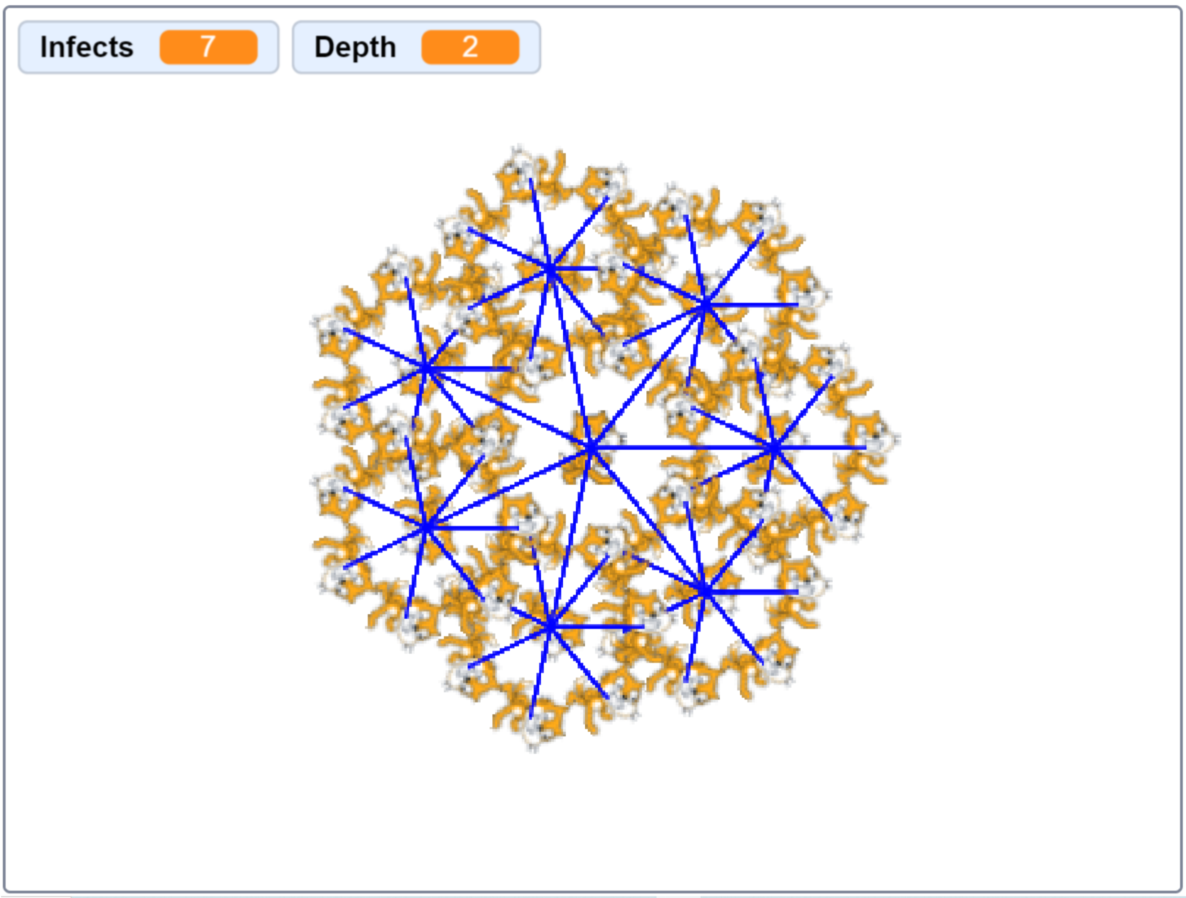

Figure 2.3 A figure illustrating virus propagation drawn with *Scratch 3*.

3. Present situation (hardware)

Nowadays, with the downsizing and cheapening of electronic chips, affordable programmable robots that use Turtle Geometry are available.

Already back in 1979 *Big Trak* was introduced. It was a six wheels programmable tank that moved using a reduced version of Turtle Geometry [18].

Much more recently a programmable car named *Pro-Bot* (Figure 3.1), incorporating a very close to *Logo*'s implementation of Turtle Geometry was introduced. It has a display showing the program, it has connectivity with computers, it incorporates sensors, etc. [19]. It is very well suited for Primary and Secondary Education.

There are other similar programmable robots with a much simplified version of Turtle Graphics incorporating keyboards without letters or numbers. Four arrows (↑, ←, →, ↓) indicate Forward, Left, Right and Backwards, respectively, but they have no numerical input. Movements are restricted to 1 step forward or backwards and turns are restricted to 90 degrees clockwise or counterclockwise. For instance, a path like that of Figure 1.1 can be programmed typing:

$$\uparrow\uparrow\uparrow\uparrow\rightarrow\uparrow\uparrow .$$

Therefore they are well suited for Early Childhood Education and can be used by children that still can't read and write but can count. Examples are *Bee-Bot/Blue-Bot* [20] and *Code & Go Mouse* (Figure 3.2).

These hardware somehow take us back to the early times of the mechanical turtles.

Figure 3.1 The keyboard and screen of *Pro-Bot*.

Figure 3.2 The keyboard of *Code & Go Mouse*.

4. The formal attitude with respect to the use of technology in the mathematics class

The CEMAT (Comité Español de Matemáticas –Spanish Mathematics Committee) has recently published a comprehensive report about the Primary and Secondary School mathematics curricula [21]. The need to connect teaching mathematics with programming as a positive synergic experience in problem solving is underlined ([21], p. 5). Although neither specific computer languages nor computer systems are explicitly mentioned, the use of different technologies such as graphic calculators, spreadsheets, dynamic geometry systems (DGS) and computer algebra systems (CAS) is recommended ([21], p. 15). In the more informal interview to some of its authors [22], the use of *Scratch* or *Snap!* in the mathematics class at Primary School level is specifically recommended

According to my experience, the authorities and the experts in Spain recommend the use of ICT but its use is not generalized and, when used, they are many times underused (for instance some Spanish high school textbooks propose to use the DGS *GeoGebra* to plot functions and perform some computations, but neither propose to use its dynamic geometric capabilities nor its computer algebra ones). My experience, after speaking with colleagues from different countries, is that the situation is not very different elsewhere.

5. The goal of this work. Our experience

I have thought for a long time that it is a pity to forget about using Turtle Geometry for mathematics teaching just because it is supposed to be outdated old stuff. This work gives a brief introduction to Turtle Geometry and a panoramic view of its present possibilities. The goal of this work is to try to convince the reader that modern Turtle Geometry implementations deserve a place in the set of useful pieces of software for mathematics teaching.

We have studied different aspects of Turtle Geometry and mathematics teaching:

- Regarding the effect on skills acquisition, we carried out an experience along three consecutive academic years about the effect of working with the Turtle Geometry (using Scratch 2) in the learning of geometric concepts of future Primary School teachers [23]. It took place with students of the Universidad Complutense de Madrid and the experimental study carried out showed an improvement on both the academic performance as well as the student satisfaction. This was an extensive study, involving several teachers and 366 students.

- Another study about the attitude of 27 students of a Master's Degree in Secondary Teacher Training regarding their preferences: using computers versus using programmable robots (*Scratch 3* versus *Pro-Bot*) in their future work as teachers was detailed in [24]. The future teachers were really enthusiastic when working with the robots, in contrast with what we could qualify as normal interest in *Scratch*. But, curiously, the showed in the polls a clear preference for teaching using traditional hardware (possibly due to a lack of self confidence in the use of the new hardware). [25]. There are many articles about the use of robots in STEM, like the well-known [26,27], but we haven't found a similar one, about device comparisons.
- A possible use of Turtle Geometry to visualize abstract processes has been already mentioned in Section 3 (virus propagation). Spanish versions of the tale and video about a cat that propagates a virus (illustrated with images generated by *Scratch 3*) are available from the Instituto de Matemática Interdisciplinar (IMI) web page [28]. The English version of the tale (illustrated with images generated by *Maple*) can be found in *Mapleprimes* web page [29]. The good reception of these tales gave rise to the already mentioned articles [15,16].

Summarizing our experience with Turtle Geometry, it offers a very appropriate and comfortable environment for certain specific tasks (teaching some geometry concepts, illustrating fractals-related issues, etc.) on different kinds of devices.

6. Conclusions

There are fashions in the world of mathematics teaching too. For instance, Euclid's *The Elements* was the standard textbook at the Victorian period in England, a didactic approach opposite to that of "modern mathematics". But there are no unanimous opinions on these topics: consider, for example, the recent attacks against the reform-based curriculum in the Netherlands described in [30].

Moreover, some ideas that are considered "modern" are older than expected. For instance, "recreational mathematics" are probably best known thanks to Martin Gardner's works, like those in the "Mathematical Games" column in Scientific American [31], but there are also very interesting earlier works such as [32] (an extensive collection of problems, many of them reprinted from magazines and newspapers of the time). Recreational mathematics could have a much active role in the curriculum, as they provide a wide variety of strategies for solving mathematical problems. Many of them can be solved and/or checked using mathematical and computational techniques in fruitful synergy.

Turtle Geometry is suffering an unjust purgatory of oblivion, despite the arrival of these mechanical devices that somehow close the circle sending us back to the tortoises of the origins of *Logo*. Wonderful powerful software and hardware that use Turtle Geometry, and are appropriate for teaching certain mathematical issues, are available in 2021.

In my opinion, Turtle Geometry adapts very well to theories for mathematics education such as Realistic Mathematics Education [33], which six principles have been recently reformulated as: activity principle, level principle, intertwinement principle, interactivity principle and guidance principle.

Summarizing, as final conclusion of this work: despite the success of the (many times underused) DGS *GeoGebra*, the scarce use of CAS and the oblivion of Turtle Geometry, I believe there is place for the three of them in mathematics teaching. There are nowadays very interesting implementations of Turtle Geometry on different hardware that worth been used for mathematics teaching. I hope I have convinced the reader that Turtle Geometry shouldn't be obliterated.
I would like to finish with a paragraph of the 1868 book *Cuentos del día* (*Tales of the day*) by the Spanish writer Ventura Ruiz Aguilera:

> *From the old, we will keep all the sacred, everything beautiful and everything useful and applicable to the construction I propose... Everything else, out! Old, just for being old, does not deserve even a tear.*

Acknowledgements Partially funded by the research project PGC2018-096509-B-100 (Government of Spain).
I would also like to thank the anonymous reviewers for their most valuable comments and suggestions.

References

[1] Papert, S. (1980). *MINDSTORMS: Children, Computers, and Powerful Ideas*. New York: Basic Books, Inc.
[2] Abbelson, H., and diSessa, A. (1981). *Turtle Geometry. The Computer as a Medium for Exploring Mathematics*. Cambridge, MA: The MIT Press.
[3] Logo Foundation (2015). *Logo History*. https://el.media.mit.edu/logo-foundation/what_is_logo/history.html
[4] Walter, W. G. (1950). An Imitation Of Life. *Scientific American* 182/5, 42-45. Available from: https://www.jstor.org/stable/10.2307/24967456
[5] Holland, O. (2003). Exploration and high adventure: the legacy of Grey Walter. *Philosophical Transactions of The Royal Society A. Mathematical Physical and Engineering Sciences*, 361, 2085-2121.
[6] *Roamer. The History of Turtle Robots*. https://roamerrobot.tumblr.com/post/23079345849/the-history-of-turtle-robots
[7] Cyberneticzoo.com (n.a.). *1969 – The Logo Turtle – Seymour Papert et al (Sth African/American)*. http://cyberneticzoo.com/cyberneticanimals/1969-the-logo-turtle-seymour-papert-marvin-minsky-et-al-american/
[8] Goldman, R., Schaefer, S., and Ju, T. (2004). Turtle geometry in computer graphics and computer-aided design. *Computer-Aided Design* 36, 1471–1482.
[9] *FMSLogo: An Educational Programming Environment*. http://fmslogo.sourceforge.net/
[10] Harvey, B.: *Berkeley Logo (UCBLogo)*. http://www.cs.berkeley.edu/~bh/logo.html
[11] *Logo Tree*. https://pavel.it.fmi.uni-sofia.bg/logotree/
[12] https://scratch.mit.edu/
[13] https://snap.berkeley.edu/
[14] Mönig, J., and Harvey, B. (2020). *Snap! Reference Manual 6.8*. https://snap.berkeley.edu/snap/help/SnapManual.pdf
[15] Roanes-Lozano, E., and Solano-Macías, C. (n.a.). Using Fractals and Turtle Geometry to Visually Explain the Spread of a Virus to Kids: A STEM Multitarget Activity.

Mathematics in Computer Science. (Available online). https://link.springer.com/article/10.1007/s11786-021-00500-9

[16] Roanes-Lozano, E., and Roanes-Macías, E. (2021). *A Simplified Introduction to Virus Propagation Using Maple's Turtle Graphics Package Suitable for Children*. In: R. M. Corless, J. Gerhard, and I. Kotsireas (eds.). Maple in Mathematics Education and Research. Cham /Switzerland: Springer Nature Switzerland AG. Communications in Computer and Information Science, CCIS 141, pp. 334-349.

[17] Roanes-Lozano, E., and Roanes-Macías, E. (1994). An Implementation of "Turtle Graphics" in Maple V. *MapleTech* Special Issue, 82-85.

[18] Wikipedia (2020). *Big Trak*. https://en.wikipedia.org/wiki/Big_Trak

[19] *Terrapin Downloads Center. Robot Downloads. Get the Pro-Bot Robotics Manual (PDF)* (n.a.). https://www.terrapinlogo.com/downloads.html

[20] *Terrapin Downloads Center. Robot Downloads. Getting Started with Blue-Bot App (PDF)* (n.a.). https://www.terrapinlogo.com/downloads.html

[21] Calvo Pesce, C., Carrillo de Albornoz Torres, A., de la Fuente Pérez, A., de León Rodríguez, M., González López, M. J., Gordaliza Ramos, A. Guevara Casanova, I., Lázaro del Pozo, C., Monzó del Olmo, O., Moreno Verdejo, A. J., Rodríguez Muñiz, L. J., Rodríguez Taboada, J., and Serradó Bayés, A. (2021). *Bases para la elaboración de un currículo de Matemáticas en Educación no Universitaria*. Unknown: CEMat (Comité Español de Matemáticas). Available from: https://www.icmat.es/wordpress/wp-content/uploads/2021/06/Bases-para-la-elaboracion-de-un-curriculo-de-Matematicas-en-Educacion-no-Universitaria.pdf

[22] Torres Menárguez, A (2021). Más razonamiento y menos cáculo a mano: como enseñar Matemáticas en el colegio, según los matemáticos. *El País* (newspaper). June 6th 2021. Available from: https://elpais.com/educacion/2021-06-06/mas-razonamiento-y-menos-calculo-a-mano-como-ensenar-matematicas-en-el-colegio-segun-los-matematicos.html

[23] Martínez Zarzuelo, A., Rodríguez Mantilla, J. M., Roanes Lozano, E., and Fernández Díaz, M. J. (2020). Efecto de Scratch en el aprendizaje de conceptos geométricos de futuros docentes de Primaria. *Revista Latinoamericana de Investigación en Matemática Educativa (RELIME)*, 23(3), 357-386.

[24] Roanes Lozano, E., and Fernández-Salinero, C. (n.a.). *Una experiencia de clase con hardware (robots programables) y software (programación por bloques) llevada a cabo con futuros profesores de Educación Secundaria de matemáticas*. In: Actas del congreso AprendETIC. Madrid: Editorial Complutense. (To appear).

[25] Roanes Lozano, E., and Fernández-Salinero, C. (n.a.). *La actitud de futuros profesores de Secundaria ante el uso de robots programables en la clase de matemáticas*. In: Investigación en Educación Matemática XXIV. Actas del SEIEM XXIV. Valencia: SEIEM. (To appear).

[26] Highfield, K. (2010). Robotic Toys as a Catalyst for Mathematical Problem Solving. *Australian Primary Mathematics Classroom* 15/2, 22-27.

[27] Leoste, J., and Heidmets, M. (2019). The Impact of Educational Robots as Learning Tools on Mathematics Learning Outcomes in Basic Education. In: T. Väljataga and M. Laanpere (eds.). *Digital Turn in Schools. Research, Policy, Practice*. Singapore: Springer Singapore, pp. 203-217.

[28] Universidad Complutense de Madrid. Instituto de Matemática Interdisciplinar (IMI). Other activities 2020: https://www.ucm.es/imi/other-activities-2020

[29] Mapleprimes. Posts: https://www.mapleprimes.com/posts/212674-Why-Staying-At-Home-Is-Good-To-Avoid
[30] van den Heuvel-Panhuizen, M. (2010). *Reform under attack – Forty Years of Working on Better Mathematics Education thrown on the Scrapheap? No Way!* In: L. Sparrow, B. Kissane, & C. Hurst (eds.). Shaping the future of mathematics education: Proceedings of the 33rd annual conference of the Mathematics Education Research Group of Australasia. Fremantle, Australia: MERGA, pp. 1-25.
[31] Mulcahy, C. (2014). *The Top 10 Martin Gardner Scientific American Articles*. https://blogs.scientificamerican.com/guest-blog/the-top-10-martin-gardner-scientific-american-articles/
[32] Dudeney, H. E. (1917): *Amusements in Mathematics*. London-Edinburgh-New York: Thomas Nelson and Sons, Ltd.
[33] van den Heuvel-Panhuizen , M., and Drijvers, P. (2014). *Realistic Mathematics Education*. In: S. Lerman (ed.). Encyclopedia of Mathematics Education. Dordrecht: Springer Science+Business, pp. 521-525.