



HAL
open science

Cubeling: Building connections between spatial reasoning and computational thinking

Heiko Etzold, Kevin Larkin

► **To cite this version:**

Heiko Etzold, Kevin Larkin. Cubeling: Building connections between spatial reasoning and computational thinking. Thirteenth Congress of the European Society for Research in Mathematics Education (CERME13), Alfréd Rényi Institute of Mathematics; Eötvös Loránd University of Budapest, Jul 2023, Budapest, Hungary. hal-04407937

HAL Id: hal-04407937

<https://hal.science/hal-04407937>

Submitted on 21 Jan 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Cubeling: Building connections between spatial reasoning and computational thinking

Heiko Etzold¹ and Kevin Larkin²

¹University of Potsdam, Germany; heiko.etzold@uni-potsdam.de

²Griffith University, Australia

This theoretical/pedagogical based paper examines a technology supported teaching episode, which focussed on the relationship between building a block structure in a virtual environment, and coding the instructions to build the structure, when using an iPad application called Cubeling. We suggest that the use of Cubeling can support children, via the use of two key representations, in developing both their computational thinking and their spatial reasoning. We discuss our initial observations and reflections from a series of lessons where children in Grade 5 (age 10-11) used the Cubeling app to build virtual three-dimensional (3D) structures. We conclude by suggesting further avenues of research to investigate the connection between computational thinking and spatial reasoning and the role of a digital tool in supporting learning in both domains.

Keywords: Spatial reasoning, Computational thinking, Apps, Primary education, Mathematics.

Introduction

In this paper we discuss our observations and reflections based on a series of teaching episodes, involving a class of Grade 5 students using an iPad application called *Cubeling* (Etzold, 2021). The aim of this paper is to establish a possible research agenda for better understanding the connection between the two areas of *computational thinking* and *spatial reasoning*. The app *Cubeling* was designed to support children in understanding both areas and uses a range of iPad affordances to demonstrate the relationship between two conceptually different ways of building the cube structures (a physical building mode and a coding mode). In this initial phase of our research, we are interested in investigating further the conceptual connections between coding instructions for building structures and the physical manipulation of blocks to create the structures. According to Wing (2008), computational thinking is an analytical approach to understanding and solving problems that shares some features with mathematical problem solving, without the requirement of geometric or algebraic “proofs”. According to Lowrie et al. (2020, p. 176) spatial reasoning refers to a “broad suite of cognitive skills involved in the mental manipulation of two-dimensional and three-dimensional relations between and within objects”.

Research regarding computational thinking and spatial reasoning

There are many intervention studies showing a correlation between competences in (aspects of) computational thinking and spatial reasoning (Köllich, 2022, p. 26). In most of these studies, children program robots (e.g., BeeBot, Ozobot) to follow a path or find a way on a plane (Brainin et al., 2021; Diago et al., 2021; González-Calero et al., 2019; Sisman et al., 2021). One of the most important advantages of this programming is that the children use it to directly practice spatial visualisation tasks. They first must imagine a route and then program this route. As the robot follows their instructions, they get immediate feedback as to whether they have correctly programmed the robot to

complete the intended task (Brainin et al., 2021, p. 10). More difficult spatial visualisation skills are required when the robot has a different orientation than the observing children (Diago et al., 2021, p. 3) for example, the robot is facing the child and thus “lefts” become “rights” and “forwards” become “backwards”.

In activities with programmable robots, although occurring in a three-dimensional space, the movements of the robot are in fact only two-dimensional. This means that the children may not be learning all the skills that are normally assessed in many standardised spatial reasoning tasks, which include mental rotation or spatial visualisation tasks in three-dimensional space (e.g., folding nets or rotating objects). This limitation is noted in research by Francis et al. (2016) where they compared robot paths in *Scratch* and *LOGO*. These authors note that “using a two-dimensional screen for programming a robot to move in three-dimensional space additionally requires different spatial thinking (i.e., *locating* and *orienting* in two dimensions, and then *locating*, *orienting*, and *pathfinding* in three dimensions). This shifting between dimensions adds considerably more complexity to the spatial reasoning required for programming a robot compared with moving an on-screen turtle or cat (Francis et al., 2016). Only a limited number of studies focussed on *real* three-dimensional movements of a robot arm (Verner, 2004) or a drone (Milicic, 2020), or the creation of three-dimensional objects in a CAD software for realising it with a 3D printer (Dilling & Vogler, 2021); however, such technology is beyond the financial capabilities of most primary schools and is not included in primary school curricula.

Overall, the studies describe how computational thinking and spatial reasoning develop respectively and discuss aspects of learning that are visible in children’s actions as they use the learning environments; however, in the literature there is limited research inquiring about the deeper *connections* between these two areas and how they can be coherently described and explained. An exception is Köllich (2022) who addresses the methodological problem of the two-dimensional robots’ movement in the three-dimensional room by creating an adapted computational thinking test in which only two-dimensional relations are required.

As mathematics educators, we are particularly interested in how the relationship between computational thinking and spatial reasoning can be developed, and how a digital tool can maximise children’s engagement and learning in these two domains. As a starting point in understanding this relationship, and to propose a classroom based approach to investigate the relationship in a rigorous way, in this paper we discuss a series of teaching episodes, which were delivered by the lead author, to support the development of computational thinking. Coding is recognised as an important component of computational thinking (Cretin et al., 2022) and in the app *Cubeling* (Etzold, 2021), children can use a coding environment to create 3D cube structures (on a 2D screen) in several ways (outlined below).

Cubeling was initially created to support spatial reasoning via the digital manipulation of cubes in a range of ways. It was only in later versions of the app, that the option to build structures via a coding environment was added. In *Cubeling*, the coding environment was created to a) document the process of the children building the structure (as the children move the blocks in the three-dimensional view the coding is automatically created and as the children create the code, the blocks are automatically

created in the three-dimensional view) and b) allow the children to experiment with a different type of representation that might make activities such as building repeated structures easier via loops of code. In our view, this differentiates *Cubeling* from other programs such as *Scratch* where the emphasis in the latter is the coding (with the *Scratch* cat used to verify the coding) and where the direction of the relationship is unidirectional (i.e., coding to 3D). In *Cubeling*, the coding is available to the children to both document their spatial reasoning and as an alternative way to conceptualise the building of the structures.

Brief description of Cubeling

The app *Cubeling* supports several representations of block structures, which are directly connected to each other (See two examples in Figure 1). If one representation is changed by the user, all other representations are changed simultaneously. There are always two of the five available representations presented side by side, and in this research, we focussed on the following two representations:

- *3D View*: In this view the block structure is shown as a spatial representation. The user can rotate or “zoom” the representation and even look at the structure from beneath. To add a cube, the user must tap a grid square or another cube. Cubes can also be easily removed.
- *Coding View*: In this view it is possible to “code” the block structure by using commands such as “buildCube(at: (?,?))” as well as loops and variables. The coding environment can be interpreted as *another* representation of the block structure, with the unique affordance of also describing the building *process*. This affordance was the focus during the teaching episodes reported here. A detailed description of *Cubeling* can be found in (Etzold, 2022).

Typical scenarios, supported by the app, are comparing and translating, between a range of 3D block structure representations and the respective coding representation. This functionality is important in contexts where there is some ambiguity in the representation (Bönig & Thöne, 2018, p. 23) (e.g., being able to rotate the structures, or look at the coding, helps children to determine the number of blocks in a structure that would not be possible to determine in a static representation, where only the front or top view are available).

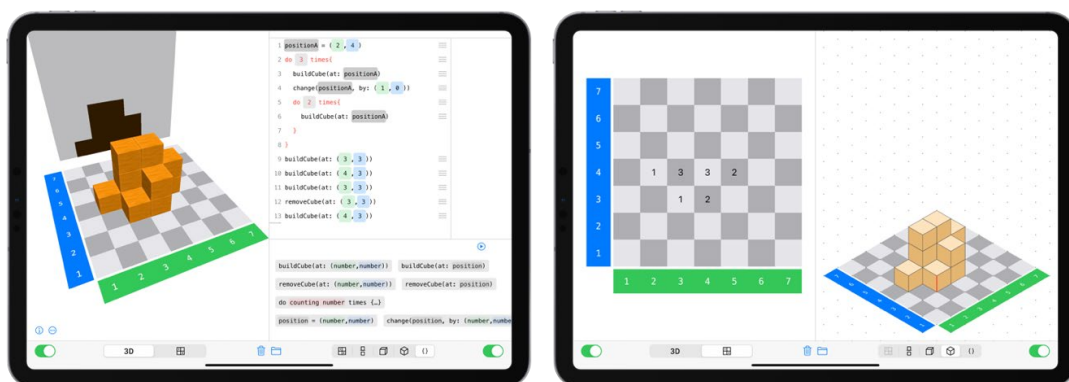


Figure 1: Cubeling App with 3D and Coding View (left), and Building Plan & Isometric View (right)

Background to, and description of, the lesson sequence.

In this section we briefly provide information about the participants in the study and explain the series of lessons that the children undertook in learning to use the *Cubeling* app.

Participants

The participants were 25 Grade 5 children from a school in Brandenburg, Germany. Their teacher was a student teacher from Potsdam University. The researcher had previously created the *Cubeling* app (with focus on spatial reasoning). The student teacher and the researcher worked together on developing both the coding environment of the app and the learning environment that included the use of the app. They worked with the children, who had never used *Cubeling* previously, over a period of four, 90 minutes lessons held over three days.

Teaching episodes

The goal of the teaching intervention was not to teach coding using digital cubes, as there are other coding languages that are more appropriate for teaching programming (e.g., *Scratch* or *Minecraft*). Rather, the aim was to develop an algorithmic perspective, via the process of code construction, that would support the children's haptic experience (and off app physical experience) of building structures with cubes. This process is documented within the app, made translatable for a computer, and basic coding structures like loops and variables are made visible to highlight the connection between the 3D View and the Coding View. Below is an outline of the learning plan.

- *Episode 1: Introduction to processes of describing and building physical cube structures*
- *Episode 2: Building structures using the 3D View and the Coding View in Cubeling*
- *Episode 3: Building structures using loops and loop variables*
- *Episode 4: Advanced exercises with loops, loop variables and double loops*

We now focus on several instances within the first three teaching episodes, as they best demonstrate a connection between the coding and spatial reasoning components when using the app. Our understanding of connection is based on a series of observations of the lessons by the lead author and the experiences of the teacher student who hold these lessons.

Episode 1: Introduction to processes of describing and building physical cube structures

The children were required to build, unseen by their partners, a block structure containing ten cubes. They then took turns to describe to their partner how to build a replica of their structure. With an eye to the future coding that the children would be completing, we encouraged the children to not only describe their finished structure, but also to describe the process (i.e., the command structure) of how it was built. In discussing this experience, the children used a variety of strategies (e.g., build the base and then specify the number of levels; describe the position of the blocks using a coordinate system; or subdivide the overall structure into smaller units). Many of these ideas and approaches are helpful to call on later when the children are "coding". For example, the breaking down into "smaller units" is an essential approach in informatics and is reflected in the coding in *Cubeling* (e.g., building a row using loops). The idea of numbering is particularly prominent in the app through the checkerboard

pattern and the desire for concrete locations of the cubes. Positions, and changes of position, can be used to specify exactly in what direction the cubes are to be built.

Episode 2: Building structures using the 3D View and the Coding View in Cubeling

In this episode, the children initially repeated similar tasks to the first episode; however, in this episode the app was available. In the app, as a block is placed in the 3D View, the children see the code that represents that action in the Coding View (see left side of Figure 1) for example, a block placed in the 3D View at the coordinates (2,3), generates the following text “buildCube(at: (2,3))” in the Coding View. In the next step, the children write the code to build a desired structure. To complete this task, the children need a mental representation of the structure they want to build, and an understanding of how such a structure can be represented in a coordinate system. By way of example, in building a one level structure of three blocks, children need to picture the structure, then recognise that it can be described as one block placed respectively on (2,3), (2,4), and (2,5), and then type the following three lines of code “buildCube(at: (2,3)); buildCube(at: (2,4)); buildCube(at: (2,5))”.

The important learning in this episode is that the app encourages children to think in a bi-directional manner – they can build a structure by adding and removing blocks in the 3D View, or they can build the same structure using coded instructions. With both views simultaneously in use, children can see the connection between the two representations. At this juncture, the main advantages of the coding view are that it demonstrates the structure of the 3D view and documents the building process.

As an example of children understanding the relationship between the Coding View and the 3D View, some children asked if it is possible to change the cube colour or texture in the coding view, to create a house with windows in the 3D View, by using glass instead of wooden cubes. Such options (not supported by the app currently) could help to develop additional connections between spatial reasoning and coding as the children would have to identify both the desired position of the window and the type of material. In future iterations of *Cubeling*, the use of glass cubes, for example, might be coded as “buildCube(at: (2,3), texture: .glass)”.

Episode 3: Building structures using loops and loop variables

In this episode the children used loop commands and loop variables, for the first time, to create more complex cube structures. The use of loops and loop variables is a more complex coding skill and saves children from having to type repetitive commands. The commands “do ... times {}”, “position = (?,?)” and “change(?, by: (?,?))” are available for the children to use to build such structures. It was surprising for us that they experienced little difficulty in using the “change” command, although the coordinates given in it represent a dynamic shift rather than a static position. The “change” command supports this with its notation “change(..., **by**: ...),” whereas positions are written as “buildCube(**at**: ...)”. The different use of coordinates for positions and shifts (coding) is comparable to the position and direction vectors (spatial reasoning) that children will later experience in secondary school.

For a deeper understanding, the app has a code tracing option, which can be used to follow the code step by step. When activating this option, a diagram next to the code is shown, where the action of each code line can be followed by pressing a down-button. The 3D View then shows the cubes built

up to the current line. This tracing is especially useful to understand the change of positions when using variables inside a loop (see Figure 2). While the Code View represents the building process for simple structures (see Episode 2), the tracing diagram now can support the children’s understanding of the building process for more complex buildings.

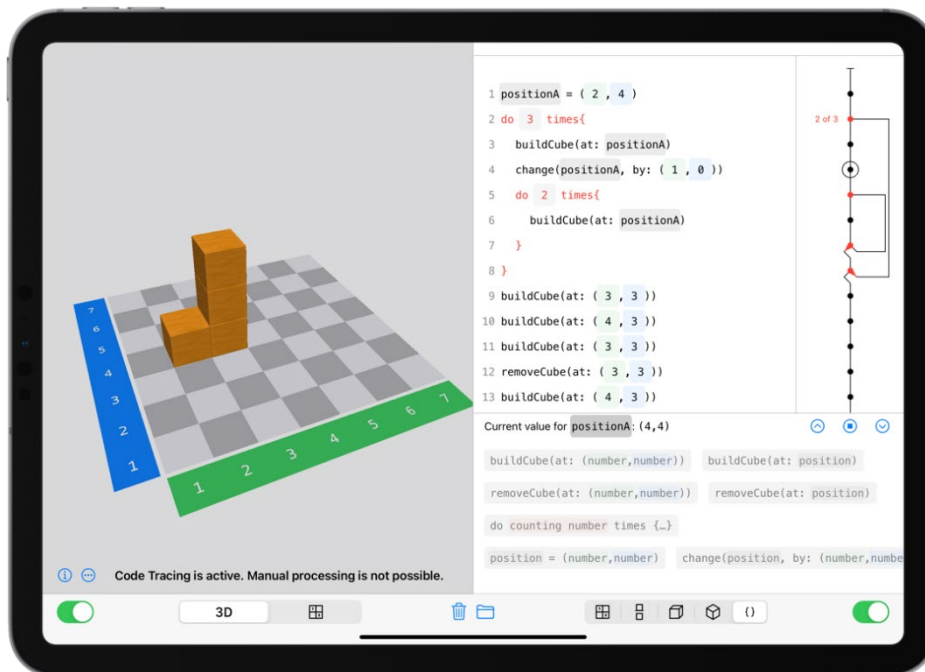


Figure 2: Code tracing in Cubeling App

Discussion and conclusion

Based on the research literature, feedback from children, and our interpretation of the lesson sequence, we now present a summary of some key observations.

From a didactic point of view, we want to emphasise once again that the app is not designed to teach children coding and is therefore limited in terms of the complexity of the Coding View. For example, currently coloured cubes and the stackable are not currently supported in the Coding View. We do not consider this problematic, as the purpose of the Coding View is to document children’s spatial thinking and to promote spatial reasoning via the use of a coding environment. If teachers specifically wish to teach coding, there are dedicated programs to do so. In terms of student engagement, at the conclusion of the lesson sequence, the children were asked to evaluate the project phase and express any views that they had regarding the *Cubeling* app. A common response was that, although they had initially shied away from working with it, they had learned how practical and powerful the Coding View is, and that many of them now actually preferred using it to build the structures.

Child At the start I did not find coding very exciting, I preferred building in the 3D View. But now that we have the loops, I prefer to do it with the code.

There was significant enthusiasm for learning “how to be an architect” and for further learning about how to code, with many children asking questions even after the bell had rung. We were also frequently asked if there was a way to use the app for playing at home. The interest of the children, and the enthusiasm about this unusual way of solving tasks, was very apparent.

A working hypothesis for us is that the simultaneous use of a code-related and a space-related representation, where changes in one are automatically reflected in the other, consolidates the underlying mental structures. Thus, instead of a choice of one learning object (coding) or another (spatial), there are two learning objects that are linked to each other via the *Cubeling* artifact (here we use the term *artifact* to describe the app as a tool). In essence, instead of a *Duo of Artifacts* (Maschietto & Soury-Lavergne, 2013) we have a *Duo of Objects* within one artifact. That means, the app (as an artifact) supports the connection of two mathematical objects (computational thinking and spatial reasoning), in difference to the common use of several artifacts (e.g., virtual and analogue manipulatives) to support one mathematical object. In this understanding, *Cubeling* can be used as a (manipulable) representation for creating connections between computational thinking and spatial reasoning. We see advantages for both children's computational thinking and their spatial reasoning in the use of dual representations; however, further research is required, with experimental measures, to determine the exact nature of the relationship between the two forms of representation.

As this is our initial investigation of *Cubeling*, a future research agenda can include investigating some of the following questions:

- What is the impact on children's learning in one representational form (e.g., spatial) on their learning in a different representational form (e.g., coding)? To what extent does developing one area support the development of the other area? Can both areas be supported simultaneously, or will this result in a diminished understanding of both?
- What methodological approaches are required to better understand the relationship between computational thinking and spatial reasoning?
- How can the teaching environment be structured to make the best advantage of the respective affordances in the spatial and coding environments?

Acknowledgment

We would like to acknowledge the support of Joanna Radack who assisted with reviewing relevant literature for this paper. The *Cubeling* app is freely available for use and the authors have no commercial interest in the app.

References

- Bönig, D., & Thöne, B. (2018). Die Klötzchen-App im Mathematikunterricht der Grundschule – Potentiale und Einsatzmöglichkeiten [The block app in primary school maths lessons – potential and possible uses]. In S. Ladel, U. Kortenkamp, & H. Etzold (Eds.), *Mathematik mit digitalen Medien – konkret* [Mathematics with digital media – in concrete terms] (1st Edition, S. 7–28). WTM-Verlag. <https://doi.org/10.37626/GA9783959870788.0.02>
- Brainin, E., Shamir, A., & Eden, S. (2021). Robot programming intervention for promoting spatial relations, mental rotation and visual memory of kindergarten children. *Journal of Research on Technology in Education*, 1–14. <https://doi.org/10.1080/15391523.2020.1858464>
- Critten, V., Hagon, H., & Messer, D. (2022). Can pre-school children learn programming and coding through guided play activities? A case study in computational thinking. *Early Childhood Education Journal* (2022) 50, 969–981.

- Diago, P. D., González-Calero, J. A., & Yáñez, D. F. (2021). Exploring the development of mental rotation and computational skills in elementary students through educational robotics. *International Journal of Child-Computer Interaction*, 32, 100388. <https://doi.org/10.1016/j.ijcci.2021.100388>
- Dilling, F., & Vogler, A. (2021). fostering spatial ability through computer-aided design: A case study. *Digital Experiences in Mathematics Education*, 7(2), 323–336. <https://doi.org/10.1007/s40751-021-00084-w>
- Etzold, H. (2021). *Cubeling* (7.1) [iPad App]. <https://apps.apple.com/en/app/cubeling/id1027746349>
- Etzold, H. (2022). *Cubeling – Teachers’ Guide*. <https://heiko-etzold.github.io/cubeling-material/en/2.1/>
- Francis, K., Khan, S., & Davis, B. (2016). Enactivism, spatial reasoning and coding. *Digital Experiences in Mathematics Education*, 2, 1–20. <https://doi.org/10.1007/s40751-015-0010-4>
- González-Calero, J. A., Cózar, R., Villena, R., & Merino, J. M. (2019). The development of mental rotation abilities through robotics-based instruction: An experience mediated by gender. *British Journal of Educational Technology*, 50(6), 3198–3213. <https://doi.org/10.1111/bjet.12726>
- Köllich, L. (2022). *Questioning the relationship between mental rotation abilities and computational thinking* [master thesis]. Universität Graz. <https://unipub.uni-graz.at/obvugrhs/content/titleinfo/7835597>
- Lowrie, T., Resnick, I., Harris, D., & Logan, T. (2020). In search of the mechanisms that enable transfer from spatial reasoning to mathematics understanding. *Mathematics Education Research Journal*, 32(2), 175–188. <https://doi.org/10.1007/s13394-020-00336-9>
- Maschietto, M., & Soury-Lavergne, S. (2013). Designing a duo of material and digital artifacts: The pascaline and Cabri Elem e-books in primary school mathematics. *ZDM*, 45(7), 959–971. <https://doi.org/10.1007/s11858-013-0533-3>
- Milicic, G. (2020). Algorithmen und Neue Medien [Algorithms and new media]. *Mathematik im Unterricht*, 11, 34–42. <https://doi.org/10.25598/miu/2020-11-4>
- Sisman, B., Kucuk, S., & Yaman, Y. (2021). The effects of robotics training on children’s spatial ability and attitude toward STEM. *International Journal of Social Robotics*, 13(2), 379–389. <https://doi.org/10.1007/s12369-020-00646-9>
- Verner, I. M. (2004). Robot Manipulations: A synergy of visualization, computation and action for spatial instruction. *International Journal of Computers for Mathematical Learning*, 9(2), 213–234. <https://doi.org/10.1023/B:IJCO.0000040892.46198.aa>
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 366(1881), 3717–3725. <https://doi.org/10.1098/rsta.2008.0118>