

15. Learning Dynamic Geometry: Implementing Rotations

Angel Gutiérrez

Dpto. de Didáctica de la Matemática; Universidad de Valencia
Apartado 22045, 46071 Valencia, Spain

*

Abstract. This paper presents research in which we observed students of various ages and ability levels solving problems by using several pieces of software for solid geometry; with different user interfaces. We evaluated the influence of the software on students' learning, ways of reasoning, and the kind of mental images generated.

15.1. Introduction

One of the reasons for this book is that the emerging culture of mathematics teaching based on or assisted by computers is offering new tools and new ways of working, both to teachers and students. This is uncovering many new research questions for mathematics educators.

Some of these questions are of a cognitive nature, dealing with the way students develop concepts in their interaction with a computer. Other questions are of a socio-affective sort, since they refer, for instance, to the changes in student's interactions with other students, or to changes in their beliefs about mathematics. Other questions specifically involve the interaction between the students and the software and/or hardware they are using.

* CITE AS: Gutiérrez, A. (1995). Learning dynamic geometry: Implementing rotations. In diSessa, A. A., Hoyles, C., Noss, R. (eds.), *Computers and exploratory learning* (pp. 275-288). Berlin: Springer.

Although the last kind of question may be viewed as technical, it is as important as the other ones. The way the computer presents the information, or the students get it, is one of the factors that determine how concepts are developed and formed in students' minds. A well-known example is the difference between the concept of polygon as generated from Logo (in which the software emphasizes exterior angles) and from other pieces of software that describe polygons using interior angles.

This paper has to do with this kind of question. I present some considerations and results from an experiment in which we analyzed how several pieces of software with different user interfaces influence the success of the students' learning, and how every interface can be more or less appropriate for students of various ages or ability levels.

In section 2, I provide a general description of the research we have carried out, a part of which is the focus of this paper. Section 3 consists on a detailed review of the main characteristics of the different software programs used. Section 4 is devoted to a discussion of the results. Finally, section 5 presents a summary of the paper and some conclusions.

15.2. Hypothesis and Organization of the Study

As part of a research project, carried out some years ago and aimed to elaborate a teaching unit for solid geometry; in primary school, we began a pilot study to learn about the possibilities of software capable of showing and moving 3-dimensional polyhedra on the screen. After looking examples of such software, it was apparent that different pieces of software required different input from students and produced different visual output. We conjectured that each type of software could be appropriate for a range of students, according to their ability. Indeed, that first pilot study, carried out with a group of three sixth graders, confirmed our hypothesis. Because of its limited size, the study motivated a more specific and detailed research, the one I am reporting here.

This new research project[†] had the following objectives:

1) To analyze the nature of student interaction with this kind of software in the context of teaching solid geometry;.

When primary or secondary school students are asked to move a solid by hand to a given orientation, their control over their hands is so perfect that they perform complex movements in just a fraction of a second, without paying

[†] The research project reported in this paper was funded by the Vice-rectorado de Investigación of the Universidad de Valencia in the period 1992-93.

attention to the rotations they apply to the solid. However, when students are limited to a few movements (usually the rotations around the three coordinate axes), they must consider each available movement and, hopefully, visualization; its result in order to select one of the rotations, or a sequence of them.

Although the nature of the tasks may suggest that students would use dynamic mental images of the solids to ascertain the movement(s) they would ask the computer to perform, well-known results from Krutetskii (1976) and Presmeg (1985) cautioned us against such superficial conclusion and suggested that some students might use analytic reasoning. Thus, a related objective was to determine and analyze the strategies used by the students when solving this kind of problem (movement of 3-dimensional objects on the screen to a given orientation).

2) To check the hypothesis that there is a relationship between the students' ability for spatial visualization; and the usefulness of a given piece of software for promoting learning. In other words, we believe that students can learn more when they are using a piece of software that is easy enough to master, but which still is challenging for them.

When control over a piece of software by the students is so difficult that they cannot master it in a short time, that software is likely not useful for teaching geometric concepts; in this case, usually students shift focus from the mathematical activity to the tool, a phenomenon known as "metadidactical gliding" ("glissement metadidactique"; Brousseau, 1986).

On the other hand, a simple piece of software usually allows only a limited variety of easy activities. Able students do not gain from such activities because they are too easy.

3) To experiment with activities designed to develop students' abilities of visualization; and spatial imagination. Some of these activities dealt with the movement of polyhedra in space, and the others dealt with several types of plane representations of 3-dimensional objects.

In this paper I only present results related to the second objective. To organize the experiment, we had three main variables in mind: i) age and ability of the students, ii) kind of representation of the polyhedra on the screen, and iii) style of the software. Our considerations for each variable were:

i) The students. We wanted to have a sample representing the eight grades of the Spanish primary school. We selected groups of students from 2nd, 4th, 6th, and 8th grades (7 to 14 year olds); each group consisted of three students, except

the group from 6th grade, which had only two students. Students in each grade were selected so to have a range of ability levels in each group. Selection was made by their mathematics teachers on the basis of the students' level in mathematics, and all of them were judged of average ability. However, we discovered, with respect to the content of the experiment, that they represented a wide range of ability when working in the experimental teaching units.

Each of the 2nd, 4th, 6th, and 8th grade groups of students worked independently of the others. Each group worked an average of 25 sessions of 60 minutes; two computers were available, so the 3 children in each group worked in a 2 + 1 configuration. Each child had a booklet with the text and pictures of the activities, a set of physical polyhedra, and the supplementary materials they might need (pencil, blank paper, ...). All the sessions were video recorded for subsequent analysis.

ii) The polyhedra. Several familiar polyhedra were chosen: Cube, tetrahedron, octahedron, square pyramid, and cuboid. All the students in grades 4 or above knew the basic properties of these polyhedra, that is, their shapes, the shapes of the faces, the names of the elements of a polyhedron, etc. The students in second grade had not studied solid geometry;.

We presented each one of these polyhedra to the students in two ways: As polyhedron:transparent; polyhedra (as made of straws), in which only the edges were apparent, and as polyhedron:opaque; polyhedra (as made of cardboard), in which each face of the polyhedron had a different shade.

We also used a figural cube; whose faces were decorated with six different figural pictures (Figure 1). This figural cube; was intended to help less able students, who might have difficulties distinguishing the different shades of the polyhedron:opaque; polyhedra and identifying the parts of the polyhedron:transparent; ones. To avoid the possibility that children memorize the relative orientations of the figures, the figural cube; was made of pieces of Polydron, so it could be re-assembled easily and the relative orientations of the figures modified.

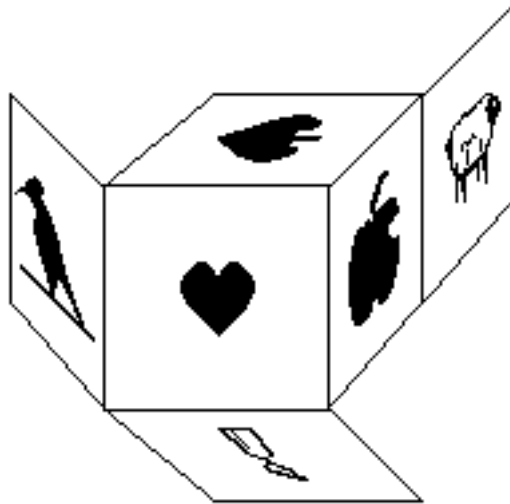


Figure 1. A figural cube;.

iii) The software. From a previous analysis of different pieces of software capable of representing and rotating 3-dimensional solids, we found that all of them fit into one of the following types: Software allowing only rotations of a certain number of degrees vs. allowing rotations of any amount of degrees, and software showing a rotation:continuous; rotation of the solid on the screen vs. making an rotation:instantaneous; step from the current orientation to the new one after having received the input of an rotation:angle of;. We also found that all the software allowed only rotations around the three orthogonal axes X (horizontal), Y (vertical), and Z (perpendicular to the screen); I will refer to these six rotations (positive and negative directions around each axis) as the "standard" ones. Three pieces of software were chosen, each one fitting into a different above-mentioned class; in section 3, I provide detailed descriptions of this software.

iv) The activities. The values of the different variables mentioned in the previous paragraphs are summarized in Table 1, which also contains the names of the pieces of software used in the experiment. On the basis of this previous analysis, we organized a learning environment based on the use of manipulatives, paper representations, and software representations of the polyhedra. Students had to solve activities that combined two out of the three representations. Although some of the activities involved manipulatives and paper representations, most of them involved computer representations. Figure 2 presents the different kinds of activities designed; in this paper I concentrate on those requesting a rotation of a polyhedron on the computer. In these activities, we provided the students with a polyhedron in one of the three contexts (a real

polyhedron, a perspective drawing on a sheet of paper, or one on the computer's screen) and a target model of the same polyhedron in a different context and a different orientation. Then, we asked the students to rotate their polyhedron into the same orientation as the target. The most frequent activity was to provide the students with a sheet of paper having several pictures of a polyhedron and to ask them to move the polyhedron on the computer to each one of the orientations shown on the sheet. More detailed descriptions of the whole set of activities can be found in Gutiérrez (1992) and (1994), and Gutiérrez and Jaime (1993).

Software Polyhedra	90° rotation	rotation:Continuous ; free rotation	rotation:Instantaneous; free rotation
Opaque figural cube;	HyperCard;	- - -	- - -
polyhedron:Opaque; shaded	- - -	- - -	Phoenix 3D;
polyhedron:Transparent;	- - -	3D Images;	Phoenix 3D;

Table 1. Variables considered in the organization of the experiment.

We presented the same activities to each of the students. However, the activities really engaged by each student depended on the student's ability, since we tended to avoid activities that proved to be too easy or too difficult[‡].

[‡] Another part of the research project had to do with different plane representations of solids (layers, isometric or orthogonal projections), although this part is not relevant to the results reported here.

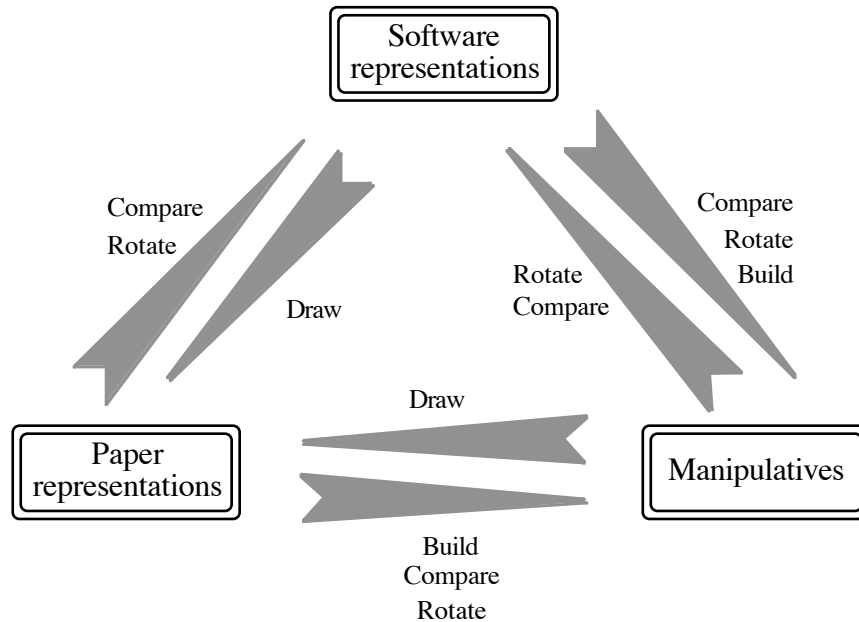


Figure 2. The different kinds of activities designed for the experiment.

15.3. Main Characteristics of the Software

We used three different pieces of software, all of them running on Macintosh SE computers. In this section I describe the relevant characteristics of each program. All of them represent polyhedra in perspective, and allow the user to give directions so the computer can perform rotations of the polyhedron on the screen.

One of the computer programs was a HyperCard; stack we designed. It presents a figural cube; on the screen (Figure 3) with its front face parallel to the plane of the screen, and six button-arrows corresponding to the different standard rotation:direction of; s of rotation around the axes. The arrows are labelled "Up", "Down", "Left", "Right", "Turn +", and "Turn -", all the arrows referring to rotations of the front face of the figural cube; (shaded in the small cube amid the arrows).

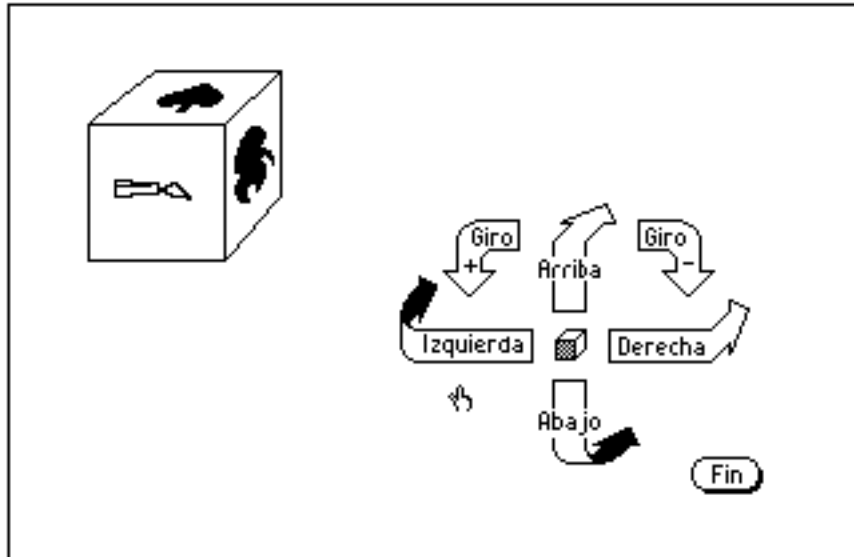


Figure 3. The HyperCard; stack: When an arrow is pushed, the figural cube; rotates 90° in that rotation:direction of;.

Because of the limitations of HyperCard; that prevented the possibility of a real rotation:continuous; rotation of the polyhedra, we decided to write the stack so that only 90° rotations would be available: When an arrow is pushed, the figural cube; vanishes, to be substituted by a cloud representing the appropriate movement. Then the cloud vanishes and the figural cube; appears in its new orientation. So, the computer show a pseudo-rotation of the cube since its intermediate orientations between the first and last orientations are not really shown, but represented by the cloud.

This program is intended to be a starter, for the students to learn the meaning of the icons representing the rotations that they will use in all the programs. It is also useful for young children since it makes the simplest rotations of the figural cube;.

Another program used in the experiment was 3D Images; (Hoffer, 1993). This program is only capable of representing polyhedron:transparent; polyhedra, but it allows the polyhedron on the screen to rotate automatically and freely around each axis (1° is the smallest rotation): While one of the six buttons displayed on the screen (corresponding to the six standard rotation:direction of;s of rotation) is being pushed, the polyhedron rotates rotation:continuous;ly in the selected rotation:direction of; (Figure 4).

This program has the capability, unique among the programs, of showing students the different orientations of the polyhedra during the rotation. 3D Images; can also accept as an input the number of degrees of a rotation and, in this case, it just shows the final orientation, without any movement. However, this utility was not used by students in the experiment.

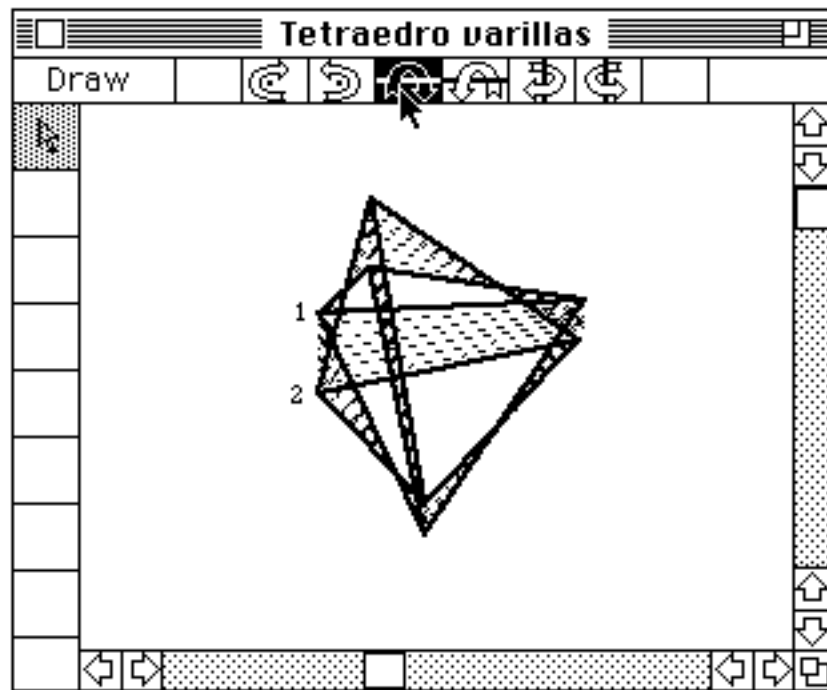


Figure 4. The program 3D Images;: rotation:Continuous; rotation from orientation 1 to orientation 2. Intermediate orientations of the tetrahedron are showed by dotted lines.

The third program was Phoenix 3D; (Dreams of the Phoenix, 1986), which can display both polyhedron:opaque; and polyhedron:transparent; polyhedra. This program does not produce a rotation:continuous; movement, but just an rotation:instantaneous; step from the current orientation to a new one: When the user has selected a rotation:direction of; rotation, the program waits for the rotation:angle of; rotation. Phoenix 3D; uses the metaphor of a hand catching a little pyramid (Figure 5) and rotating it. Then, while the cursor (hand) is moving along the screen in the proper direction, the pyramid rotates rotation:continuous;ly in real time, but the polyhedron is displayed on the background and it remains static. The movement of the pyramid gives the user

visual information about the size of the rotation. While the hand and the pyramid are moving, the program also displays the value of the rotation:angle of; rotated, so it is also possible to select the rotation:angle of; the rotation to be made and to move the cursor until this value appears on the screen.

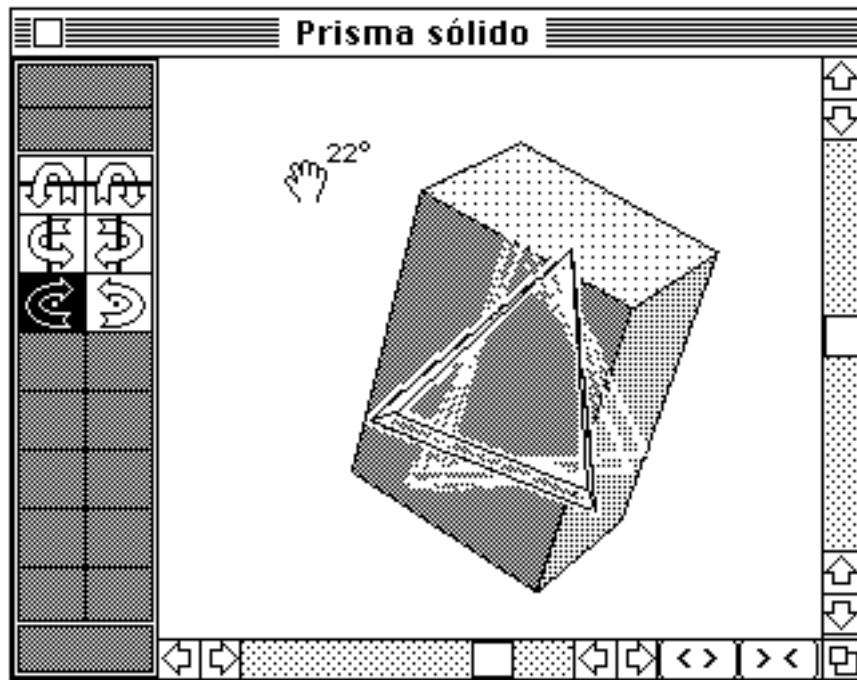


Figure 5. The program Phoenix 3D:: When the hand goes to the right side of the screen, the pyramid rotates, the length of the hand's movement determining the size of the rotation.

As can be seen, the three programs are quite different from the point of view of the user interface. Table 2 summarizes these properties.

	rotation: Angle; of rotation	Movement on the screen	Type of solid rotated
HyperCard;	90°	Pseudo continuous	Figural cube;
3D Images;	Any	rotation: Continuous;	polyhedron: Transparent;
Phoenix 3D;	Any	rotation: Instantaneous;	polyhedron: Opaque; & polyhedron: Transparent;

Table 2. Characteristics of each program used in the experiment.

15.4. Analysis of Some Results of the Experiment

A comparison of the three programs described in section 3 is necessary to determine which one might be more useful to particular students. This comparison must be based on aspects of the students' behavior when they interact with the computer, such as the strategies they used for rotating the polyhedra, the ways they identified an intended movement with one of the six available rotations, how they used the pairs of rotation:opposite; rotations, or their ability to plan one or more rotations. Results referring to the strategies used by the students and to their ability for planning movements have been presented in detail elsewhere (Gutiérrez, 1994; Gutiérrez and Jaime, 1993).

In Gutiérrez (1994) we noted that all the students produced the same strategy for rotating the figural cube; from its initial orientation to a new one, although second graders and some less able fourth graders could not apply the strategy successfully. We also showed in that paper that the strategies used for rotating polyhedron:opaque; or polyhedron:transparent; polyhedra differed among the students. On the other hand, in Gutiérrez and Jaime (1993) we showed that even students in the same school grade mastered the use of each program in quite different ways; while some were only able to make rotations rotation:by guessing;, others could plan some rotations in advance.

Usually, the solution process (from the moment a student starts imagining the rotation that has to be made until a rotation is completed by the computer) has several steps: Decide which way the solid should rotate; associate the desired rotation to one of the six standard rotation:direction of;s allowed by the computer, that is, to one of the six buttons; and to determine the amount of the rotation. Let us analyze each one of these steps, to see how the characteristics of the software influence this process.

First, when students have to determine the next rotation (or sequence of two or

more), they have to imagine the solid's path to the new orientation. In a piece of software that allows any orientation of the solid, this path does not usually fit any one of the six standard rotation:direction of;s. Students are forced to consider a decomposition of the rotation they imagined into a sequence of some available ones, or to reject their first attempt and to look only for one of the six standard rotations. The situation is quite different when the software allows only rotations of 90° . In this case the orientations of the solid (a figural cube; in the experiment reported) are much more limited, and the rotations imagined by students were, in most cases, one of the standard movements. Sometimes children in grades 2 and 4 pushed an arrow once, expecting to have the figural cube; rotated 180° , but the computer surprised them rotating the figural cube; only 90° . As angles and degrees were still meaningless for those students, this situation troubled them. They needed practice to distinguish angles of different sizes, and help from the researchers to understand why the rotation was not the one they expected.

In our experiment, this first step in the process of rotating polyhedra was straightforward. Students used physical movements consistently (mainly of their hands) to represent the rotation they wanted the solid to do, or they believed the solid should do. Presmeg (1986) described this process as using kinesthetic mental images.

The election of a rotation to be executed by the computer was often made rotation:by guessing; work by less able students, who were not able to analyze and compare the solid on the screen and the model on the sheet. The kind of software strongly influences the frequency of rotations rotation:by guessing;. Programs like 3D Images; promote guessing due to the possibility of observing the polyhedron during its rotation:continuous; rotation, looking for a "familiar" orientation (that is, an orientation that resembles the target one). An example of this behaviour was a sixth grader who pressed continuously one of the buttons on the screen, watching the polyhedron and looking for a useful orientation; if she did not recognize such orientation, then she switched to another button (some times the opposite; to the previous one), and repeated the procedure. Finally, when she recognized a suitable orientation of the polyhedron, she stopped the movement, and she started a series of short rotations aimed at moving the solid to an accurate orientation, that is as close as possible to the orientation shown in the statement of the activity. This behaviour was also quite frequent in 2nd and 4th grade students.

On the other hand, Phoenix 3D; produced an opposite effect on the students' behavior when they had no idea what rotation should be made. As this program does not produce automatic movement, students had to decide every rotation in advance (i.e., to determine the rotation:direction of; and the rotation:angle of;). If, after a few rotations, they were not successful, they felt unable to solve the task and they abandoned it. This was the behaviour of the student mentioned in the preceding paragraph.

The HyperCard; stack performed between the extremes, although nearer to 3D Images;. This program does not provide rotation:continuous; movement of the figural cube; but, because the fixed rotation:angle of; rotation of 90; allows only a few rotations, students can relatively easily move the figural cube; to its final orientation (or to one near it that they are able to relate to the final one). We observed that the less able second graders were the only students who systematically made rotations rotation:by guessing;; older students did rotations rotation:by guessing; only when the figural cube; on the screen and the one on their activity sheet had no face in common.

It is necessary to note that in some cases, when there is no useful information linking the polyhedron on the screen to the figure on the activity sheet, the only possibility students have is to rotation:by guessing;; hoping that the new solid's orientation will provide some relevant feedback. So we considered a rotation:by guessing; as a wrong procedure only in cases where students could have obtained from the screen or the text of the task some useful information to decide the rotation, but they failed to do that. We found that only students in the lower grades (especially in 2nd grade) systematically used rotation:by guessing;; Only the less able second graders did this when using HyperCard;; and all the second graders acted in this way on Phoenix 3D; and 3D Images;. The less able fourth graders did the same on 3D Images;. The other fourth graders and the less able sixth graders rotated rotation:by guessing; on 3D Images; during the first hours of the experiment, but as they gained mastery, they were more able to make reasoned choices of rotations. Finally, the more able 6th graders and the 8th graders made rotations rotation:by guessing; only when it was necessary.

Once students had decided which rotation should be made, they had to convert the movement of their hands or the one they imagined into an input to the computer. This was a two step task. First, they had to convert their intended rotation into one of the six standard ones, and then they have to choose the corresponding button on the screen.

The first step (the selection of a standard rotation) was usually correctly completed by students in 4th to 8th grades using HyperCard; and Phoenix 3D;. However, all the students had difficulties in 3D Images; because of the visual characteristics of polyhedron:transparent; polyhedra (Figure 6): When students are not aware of the rules of perspective drawing, the cubes on the left side of Figures 6-A and 6-B look similar, as do the cubes on the right side. However, to rotate from each cube on the left side to its corresponding cube on the right, reverse rotations are required.

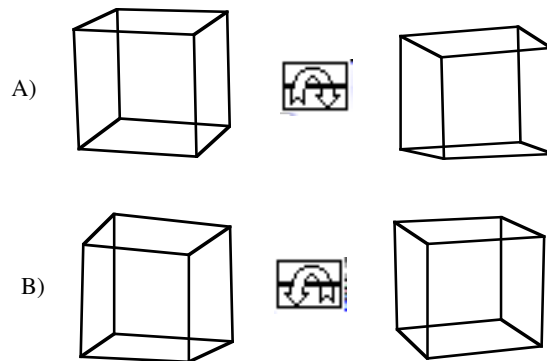


Figure 6.

Concerning the second graders, as was mentioned before, rotation rotation:by guessing; was the usual way of working, except when the more able solved tasks with figural cube;s using HyperCard;. They were unable either to imagine a required rotation, or to choose the correct button when asked to move the figural cube; in a rotation:direction of; demonstrated by a researcher with a hand or with a real figural cube;.

When a standard rotation was identified by students, they had to push the appropriate button to move the polyhedron. This may be seen an obvious task, but it was not. We have found an unexpected difference among the designs of the HyperCard; stack, on the one hand, and that of the other two programs, on the other.

As shown in Figure 3, the arrows in the HyperCard; stack were labelled to help younger students to associate the abstract rotation:direction of;s of rotation to concrete referents, describing the rotations of the front face of the figural cube;. When students paid attention to the front face, and they imagined the rotation that should make that face, they usually selected the correct arrow (excepting, again, the second graders). However, when students paid attention to other face on the screen, the labels were no longer useful as referents. On the contrary, they frequently induced mistakes.

For instance, if a student wanted to move the apple (right face in Figure 3) to the top face, the usual students' procedure was to think that the rotation to be performed should move the apple up ("arriba"). So the arrow to be pressed should be "Arriba" (Up). This difficulty was important for students in grades 2 and 4, and for the less able 6th graders. Older students only made this mistake inadvertently, and they recognized their mistake immediately.

In contrast, 3D Images; and Phoenix 3D; have unlabelled arrows, so students were discouraged from interpreting rotations in terms of absolute directions (up, left, ...). In this case, the usual method was to represent the rotation with their hands, or with some object, and then to look for the arrow representing the same rotation:direction of; movement. The problem of associating the wrong button with the imagined rotation was much less frequent.

Of course, it was quite usual for most of the students to make mistakes (i.e., to perform wrong rotations), so they dealt with the issue of correcting the mistakes. Some instructional profit can be obtained from this fact, since it offers an opportunity to work on the relation of rotation:opposite; rotations and on positive and negative angles. Only Phoenix 3D; had "undo rotation" in the menu bar, which students participating in our experiment used whenever necessary. In the other two programs, all the students, including the 2nd graders, frequently used the inverse rotation when they had to undo a wrong rotation. However, younger students exhibited a problematic behavior in this regard. When they were trying rotations rotation:by guessing;, they often shifted from a rotation:direction of; rotation to the rotation:opposite; one without realizing that both moved the polyhedron to the same orientations.

15.5. Conclusions

I have presented results from a research project aimed at analyzing the behavior of primary school students of different ages (7 to 14 year olds) when they are solving activities of solid geometry; involving rotations or using plane representations of geometric solids. We were interested in the psychological aspects of students' activity, and also in the differences in the use of the several representations of polyhedra and different pieces of software. We described here results having to do with the interfaces of the different pieces of software and the means of interaction between students and computer. These results mark some directions for effective use of dynamic software in geometry classes at different school levels. The most relevant conclusions are:

- * The sizes of rotation angle accepted by a program determines the variety of possible orientations of solids and, therefore, the difficulty of activities involving rotations of polyhedra: The smaller the angle, the greater the number of possible orientations of the solids, and the more difficult the activities will be.
- * The style of the software interface influences students' capability for choosing correct rotations and the frequency of rotation:by guessing;. Our HyperCard; stack was the easiest program for all the students, since it only allowed 90; rotations to a figural cube;. Phoenix 3D; was the most difficult because of the way of transmitting the information about rotations to the computer (by moving the cursor in the appropriate direction an adequate

distance); in particular, it was rejected by most children in the earlier grades. The program 3D Images; induced more rotations rotation:by guessing;, since it produces automatic rotation:continuous; rotation of polyhedra.

- * One might want to prevent students from associating rotation:direction of;s of rotation with absolute directions like "turn up", "turn right", etc. In plane geometry, rotations are usually named "clockwise" and "counterclockwise", but there is no generally accepted rule for rotations in 3-dimensional geometry. In the HyperCard; stack, the labels on the arrows were useful when students paid attention to the front face of the figural cube;, but they were a distraction when students focused on another face. This problem seldom appeared in the other programs.
- * As a global evaluation of the different programs used in the experiment, we can conclude: Software like our HyperCard; stack, with figural cube; polyhedra and rotations that maintain the polyhedra invariable in space, is useful for children in the lower primary grades (up to 4th grade). Software like 3D Images;, making automatic rotation:continuous; rotations of polyhedra that can be observed by students, is useful for students in the middle primary grades (up to 6th grade). And software like Phoenix 3D;, which requires complete identification of the rotation to be made as an input prior to the movement, is useful for older primary and secondary students (from grade 6 up).

More generally, from this research we conclude that it is necessary to consider the computer's interface as another factor to be taken into account when designing a unit to teach mathematics using computers. Interface design can be as important as other cognitive, mathematical, and didactic considerations.

References

- Brousseau, G. (1986) Fondements et méthodes de la didactique des mathématiques. *Recherches en didactique des mathématiques*, 7/2, 33-116
- Dreams of the Phoenix (1986) Phoenix 3D. Jacksonville, USA: Dreams of the Phoenix Inc.
- Gutiérrez, A. (1992) Exploring the links between Van Hiele levels and 3-dimensional geometry. *Structural topology*, 18, 31-48
- Gutiérrez, A. (1994) The aspect of polyhedra as a factor influencing the students' ability for rotating them, in A. Batturo (ed) *New directions in research on geometry and visual thinking*. Brisbane, Australia: Queensland University of Technology

- Gutiérrez, A. and Jaime, A. (1993) An analysis of the students' use of mental images when making or imagining movements of polyhedra, in I. Hirabayashi *et al* (eds) *Proceedings of the 17th international conference for the psychology of mathematics education*, 2, 153-160. Tsukuba, Japan: University of Tsukuba
- Hoffer, A. (1993) 3D Images. Acton, IL: W.K. Bradford
- Krutetskii, V.A. (1976) The psychology of mathematical abilities in schoolchildren. Chicago, USA: University of Chicago Press
- Presmeg, N.C. (1985) The role of visually mediated processes in high school mathematics: A classroom investigation, unpublished PhD dissertation
- Presmeg, N.C. (1986) Visualization in high school mathematics. *For the learning of mathematics*, 6/3, 42-46