DOCUMENT RESUME

ED 260 792 PS 015 253

AUTHOR Kuschner, David
TITLE A Study of the Possibilities for Reversible Actions in Software for Young Children.
PUB DATE Jun 85
NOTE 24p.; This study was partially funded by a faculty research grant from the Office of Research and Program Development, University of North Dakota.
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Graphics; *Computer Software; Developmental Stages; Early Childhood Education; *Educational Media; *Instructional Materials; *Kindergarten Children; *Logic; Young Children
IDENTIFIERS *Reversibility (Psychology); Software Design; *Software Evaluation

ABSTRACT

A study was made of the extent and ways different types of educational software allow young children to manipulate their own actions and, more specifically, to engage in reversible activity. Part 1 of the study consisted of an analysis of the types of reversible actions two kindergarten children engaged in and the actions they wanted to engage in but could not because of the nature of the software program they used. Part 2 of the study consisted of an analysis of a number of software programs designed to be of educational value for children between the ages of 5 and 8 years in which the basic activity is the construction or creation of pictures on the monitor screen. Programs analyzed were Grandmas' House, Kids at Work, Kidwriter, Pic.Builder, Rainbow Painter, Stickers, Story Maker, and StoryMaker. Analysis focused on the transformational activity related to the development of mental reversibility: negation; modifying of objects and their positions, movements, and arrangements; and the placing of objects into relationships through combining, separating and rearranging parts and wholes. After a brief review of the Piagetian concept of reversibility, results are discussed. Concluding remarks focus on characteristics of optimal "constructive" educational software. (RH)

Paper presented at the Fifteenth Annual Symposium of the Jean Piaget Society
June 6-8, 1985, Philadelphia, Pennsylvania

A Study of the Possibilities for Reversible Actions in Software for Young Children[1]

David Kuschner
Center for Teaching and Learning
University of North Dakota

At a recent national conference focusing on the uses of computers with young children, I heard a number of intriguing metaphors used to describe different pieces of educational software. One presenter compared software to an electronic sandbox, while another presenter characterized a software program as being similar to a set of blocks. I also heard two representatives from software companies describing their wares as being just like "Legos."

I took those metaphors home with me as I returned to an observational study I had begun in the fall. One morning per week I was watching two kindergarten children, who had never worked at a computer before, play with a storymaking program called, Kidwriter.[2] This program gives children the opportunity to construct a picture on the top half of the monitor screen and then to write a story, using a rudimentary word processing program to accompany their picture on the bottom half of the screen. The picture is constructed by making selections from a list of 99 predrawn graphics. Stories can consist of multiple pages, and then can be saved and later loaded back into the computer for reading.

1. This study was partially funded by a faculty research grant from the Office of Research and Program Development, University of North Dakota.

2. A full list of the software programs discussed in this study appears at the end of the paper.

In light of what I heard at the conference, I began to notice something in their work with the computer which made me wonder about those comparisons between computers and sandboxes, blocks, and Legos. On a number of occasions, the children would become frustrated because of what the software and/or computer would not let them do. There were times that the children wanted to change something about their picture by manipulating a graphic already placed, but the option for doing so was not built into the program. As I watched them struggle with these obstacles to their activity, I couldn't help but think about Piagetian theory and its emphasis on transformational activity, reversibility, and the coordination of actions. This led me to the basic question of the study presented in this paper: to what degree, and in what ways, do different types of educational software allow young children to manipulate their own actions, and more specifically, to engage in reversible activity.

Part 1 of the study consisted of an analysis of the types of reversible actions the two kindergarten children engaged in and the actions they wanted to engage in but couldn't due to the nature of the software program.

Part 2 of the study consisted of an analysis of a number of educational software programs designed for young children. These programs were constructive in nature, i.e., children are given the options of building/drawing graphics and/or groups of graphics on the screen. Each of the programs was played with over a period of time to discover what kinds of reversible actions were possible or not possible.

Before preceding to the discussion of the results of the study, however, it is important to briefly review what is meant by the concept of reversibility within Piaget's theory of development.

The concept of mental reversibility is one of the more well known and much written about aspects of Piaget's theory. The development of mental reversibility is directly related to the transition from preoperational to concrete operational thinking, and has therefore been of great interest to those researchers and teachers involved with the preschool and primary age child. It is not the purpose of this paper to present a detailed discussion of the nature of reversibility, but it is necessary to elaborate on a few points in order to make the later discussion of the potential for reversible action in computer software more meaningful.

Since reversibility has as its roots the coordination of actions, according to Piaget two kinds of activity are necessary for its development. Piaget maintained that children must actively transform objects by modifying their positions, movements, and arrangements, and by enriching the objects by placing them into new relationships (Piaget, 1971). Virtually all of the tasks Piaget used to distinguish between preoperational and concrete operational thinking involved the effects of such transformations. To be a concrete operational thinker is to understand, mentally and instantaneously, that there are potential actions inherent in any single isolated action.

To explain the relationship between activity and operations, Piaget posited the existence of a type of mental activity he called reflexive abstraction. Reflexive abstraction is the interiorization of the form of coordinated actions, and is not simply the imagination of a particular act. Piaget used the term virtual action to refer to that type of mental activity (Inhelder and Piaget, 1969). The development of mental reversibility, therefore, does not simply mean a child understands that an act can be undone; young children know that a block placed down on the floor can be picked up again. What the young child may not realize until mental reversibility develops is that there is a group of potential actions related to the single act of placing the block down on the floor. The block cannot only be picked up and removed, but its orientation and position can be changed, and it can be put into conceptual and spatial relationships with other blocks.

This relatively brief exposition of the characteristics and importance of reversibility raises a number of important questions concerning the design of educational software for young children. What are the range of actions possible? Does the software encourage reflexive abstraction by allowing children to see how their actions are in fact related to each other? Does a particular software program emphasize virtual actions and what Piaget called empirical return (the basic understanding that an action can be undone), or does the program provide the options for carrying out a variety of related manipulations? These questions guided the analysis of both the children's activity and of the software described in the following section of this paper.

What the children did (and wanted to do)

Before discussing the types of reversible actions observed in the computer play of the two 5-year-old children, it is necessary to describe the software program they were using.

Kidwriter by Spinnaker is a storymaking program. Children first select from a list of 99 predrawn graphics the ones they want to use in the picture portion of their story. They choose graphics one at a time and then have a number of options for manipulating them, each option controlled by a single keystroke. The controllable manipulations include: position (up, down, right, left), color (four choices), and size (four choices). Only the most recently selected graphic can be manipulated; once a second graphic is selected, the graphics placed earlier can no longer be manipulated. In addition, the child can change the background scene, but only before selecting the very first graphic.

After completion of the picture, the child can write a story on the bottom half of the screen, using rudimentary word processing functions built into the program. Once the story is completed, picture and story can be saved to a diskette for later viewing, but cannot be entered again for graphic or text manipulation.

Of all the possible graphic manipulations, the children used change of position most often. When a graphic is initially selected, it appears in the middle of the screen. Only once did a child want to leave the graphic in that position. There was good reason not to leave it there because any other graphic selected would appear right on top of it. Usually the first thing the child would do would be to use the arrow keys to move the graphic. During the first few sessions the children played quite frequently with this function; later they used it more deliberately, even planning where to put the graphic by pointing at various positions on the monitor screen before depressing the selection key.

The children at times used the arrow keys to center one graphic in relationship to others, and they were often quite exact about where they wanted an object to be placed, as K.'s words, "Move it over a titch," illustrate. On another occasion, K. directed R. to, "Put the umbrella in her hand," referring to where she wanted R. to move that particular graphic.

All of these examples involve either moving graphics away from the initial center position, or moving a graphic around the screen, either in play or for some deliberate purpose. There were a number of instances when the children did seem to be aware of the relationship between the movements they were creating on the screen. R. chose the letter 'I' to place on the screen, and as noted before, a graphic initially appears in the middle of the screen. He said that he wanted it in the middle and didn't have to move it. K., on the other hand, seemed to be caught between habitual action (moving the graphics away from the middle) and R.'s statement. She responded by saying: "You could move it out and put it back." It seems reasonable to conclude that her words reflected an understanding that the graphic began at one position and can be moved to a second position, and by implication could be moved back.

A second example is again related to the fact that graphics start out in the middle of the screen. R. was in the process of putting up a series of '7s' and as he was using an arrow key to move one of them he said to himself: "Back that sucker out." The choice of wording, "back...out," suggests a car in a garage: it can move in and it can move out, implying a relationship between moving backwards and moving forwards.

The children also engaged in the other forms of reversibility and transformations inherent in the software. In order to select a graphic, they had to move through the graphic list one at a time and they became quite adept at moving backwards and forwards to find the graphic they wanted. On a number of occasions they would accidentally go past the graphic they wanted to select and would have to reverse their direction. (The list is viewed by pressing 'F' for forward and 'B' for back.) Almost every time a graphic was chosen, its size was changed either before or after its position on the screen was determined. Interestingly enough, the children never seemed to make a simple change of size, but would move through all four possible sizes before settling on one; it was as if they had to see each size in relation to all the others before they could decide on a particular one.

There was some amount of erasing of graphics but not very much. Erasing is accomplished by selecting a color for the graphic which matches the background color, which is most cases is white. The children experimented with this but never elected to use this function as a way of changing their minds about the graphic chosen. It was as if they felt that they must stay with the first graphic they picked. (This may have been affected by the fact that they were taking turns selecting graphics; they may have felt that their turn would be lost unless they stayed with the graphic they selected.)

Finally, the children engaged in a type of reversibility that was not a function of the software program, but a function of their imagination. On a number of occasions they would be creating an integrated, conceptual picture when one of them would put up an incongruous graphic. During one session, for example, they were making a picture of a living room scene when R. selected the graphic of the number '7.' When K. exclaimed her displeasure at what she took to be a mistake, R. quickly responded by saying that it was a picture on the wall, and then centered the '7' to confirm his statement. (See Figure 1) An object, therefore, could be transformed simply by one of the children deciding that it was going to be something other than what it appeared to be.

As much as the children utilized the possible options for reversible actions provided by the software, the instances when they wanted to change or act on their own manipulations but couldn't, were the ones I found most intriguing and in fact served as the impetus for this study. There are several limitations inherent in the Kidwriter program. These include: not being able to manipulate any graphic other than the one most recently selected, not being able to work on a story once it had been saved to diskette, and not being able to manipulate the orientation of a graphic. It was this last limitation that the children found to be the most frustrating. The following two examples illustrate this frustration.

The first involves three of 99 predrawn graphics: a table and two chairs. One of the chairs faces to the right and one faces to the left.

Once the children discovered these graphics, they were drawn to them often as possible objects for their pictures, perhaps because they formed a conceptual set. The first time they worked with the table and chairs they selected the table and moved it slightly to the left and then selected the chair which faces to the right. (See Figure 2a) At this point the children paused. They wanted to have the chair placed at the table in the correct orientation, but for a moment did not know how to accomplish that goal. R. then realized that you could use the left arrow key to slide the chair over to the other side of the table and then go back to the graphic list to choose the other chair, the one which faces in the opposite direction.

During the very next session, the children once again selected the table, and once again selected the chair and then slid it into the correct position, but then they found themselves in some trouble. Instead of choosing the other chair, the one facing in the opposite direction of the first one chosen, they chose the same chair again. (See Figure 2b) Once they realized their problem, they could not find a solution. They did not know how to undo their choice and since there isn't any way of reversing the orientation of a graphic, they were clearly frustrated. (Since the purpose of this observational study was to see what the children would do and discover, I did not provide any assistance either in the way of direct instruction or hints.)

A few sessions later I observed another example of the children again being somewhat frustrated at not being able to manipulate the orientation of a graphic. R. had selected the number '7' from the list and placed it approximately in the middle of the screen. He then selected another '7' and as he slid it towards the left, he stopped and indicated that what he wanted to do was to turn it around so that it could be placed against and facing the first '7': he seemed to be attracted to the visual symmetry of that particular configuration. When he could not discover a way of turning the '7' around, he altered his plans and decided to fill up the entire screen with a row of '7s.'

Watching these children grapple with the problem of reversing the results of their own actions, coupled with the memory of those statements comparing software to children's constructive play, increased my curiosity about educational software in general and led to the second part of this study, the analysis of a selected number of programs. The programs reviewed were selected according to the following criteria: they are designed to be used with children ages 5 - 8, the basic activity inherent in their use is the construction or creation of pictures on the monitor screen, and the companies which market them claim that they have educational value. The purpose of this part of the study was not to recommend any particular software program nor to be critical of specific programs. The software programs reviewed were used as a sample and as a springboard for considering the general question of the potential for reversible actions in software for young children.

What the software allows (and doesn't allow)

The analysis of the software focused on the transformational activity related to the development of mental reversibility: negation, modifying of objects and their positions, movements, and arrangements, and the placing of objects into relationships (combining, separating, and rearranging parts and wholes). To restate the importance of such activity: through the process of reflexive abstraction, the child dissociates the form of these actions from their contents, and the forms become coordinated as reversible operations, thus leading to the understanding, by virtue of logical and deductive necessity, that single, isolated actions imply a related group of potential actions.

Two broad categories of software were reviewed. On the one hand there were programs with which the child draws the graphics he wants, using a variety of input modes, including keys, joystick, touch tablet, or mouse. The remaining programs operate on a placement basis: discrete, predrawn graphics (shapes or pictures) are placed by the child where the child wants them to be on the monitor screen, again using a variety of input modes. The types of transformational actions possible were to a large part dependent on whether the constructive activity was based on drawing or placement.

Negation. In terms of computer activity, an action by a child results in something occurring on the monitor screen. The result might take the form of a graphic appearing, a mark being made, or something on the screen being put into motion. To negate this result would be to move a graphic back to an original position, or cause the graphic or mark to disappear. There were four basic ways of accomplishing negation in the nine programs reviewed: erasing, replacing, removing, and undoing.

The only program with a true "erasing" function was StoryMaker by Scholastic. By placing the cursor on an eraser icon the child can then select either a thick or thin eraser and actually rub out any part of the picture by moving the cursor. Rainbow Painter provides a slightly less direct way of erasing in that it is possible to rub out part of a picture by selecting a color for the cursor which is identical to the background color of the screen; in actuality what happens is that a piece of the picture is being painted over even though the action "feels" like the action of erasing or rubbing out. In both programs it is possible to erase a part of the picture at any time, including after saving to diskette and reloading.

Similar to the act of repainting but less like the action of erasing is the replacing mode of negating, which is found in Stickers, Kidwriter, and Pic.Builder. In these three programs a predrawn graphic is placed over another to in effect replace it. In the case of Pic.Builder and Kidwriter the child is able to completely negate an action, but in Stickers that possibility is limited by the colors and shapes available. Kidwriter allows replacing only before saving to diskette while the other two permit the child to engage in the activity at any time.

Two of the programs, Kids at Work and Grandma's House give the child the ability to remove any piece of the picture by simulating the action of grabbing a graphic and carrying it to another position or entirely off screen. In both cases the figure of a person is moved next to the target graphic, a key is pressed to "grab" the graphic, and then person and graphic can be moved, at which point another keypress releases the graphic from the character's hand. The act of removal can be done at any time, including after saving the completed picture to diskette.

The final way in which an action can be negated is called undoing, and is the action least related to a physical action in real life, and most related to the power of the computer. Three of the programs (Story Maker by Sierra, StoryMaker by Scholastic, and Rainbow Painter) had single key functions which would make a graphic or section of a drawing disappear. The action of undoing is virtually an instantaneous process. For two of the programs (Rainbow Painter and StoryMaker by Scholastic) the undoing function can only be used prior to the carrying out of the next action. Story Maker by Sierra lets the child undo at any time, although the undoing will take place in sequence, i.e., one action at a time beginning with the most recent.

Modifying of actions.

Focusing again on the results of actions at the computer--the creation of graphics or marks--there were four ways in which these results could be modified. A graphic or mark could be changed in terms of its color, size, shape, or position.

Color. Each of the drawing programs (Story Maker by Sierra and Rainbow Painter) gives the child the option of choosing a color for both the pieces of the drawing and for filling in parts of the drawing. With some limitations any part of the picture can be painted over with a different color. Two of the placement programs (Kidwriter and Stickers) let the child use a single keystroke to change the color of predrawn graphics, but only before the next graphic is selected. The remaining programs (Kids at Work, Grandma's House, and Pic.Builder) do not have options for changing the colors of predrawn graphics.

Size. All of the drawing programs allow the child to draw a graphic to any size desired (within the limits of the computer and screen). Once drawn, however, there is no way to change its size without redrawing or adding on to or eliminating part of the drawing. Two of the placement programs (Kidwriter and StoryMaker by Scholastic) do contain single key functions which change the size of a graphic: in the case of the latter program there are two sizes to choose from, and in the case of the former there are four. The remaining placement programs are similar to the drawing programs in that a graphic's size can be modified by adding on to it or by eliminating parts of it.

Two of the programs provide a special way of modifying the shape and size of a graphic, specifically geometric figures. Rainbow Painter and Story Maker by Sierra facilitate the drawing of rectangles, circles, and lines by means of what is sometimes called a "rubber band" action (D'Ignazio, 1984). After selecting this option, the child puts the cursor on a particular point and "stretches out" the geometric figure until it is the desired shape and size. This operation can only be carried out at the time of the original drawing; previously drawn shapes and pictures cannot be "stretched" in this manner.

Shape. The question of modifying the shape or form of a graphic is similar to that of modifying size. Two of the programs provide the "rubber band" function, and all of the programs allow some type of adding on or taking away of pieces, accomplished through the process described in the section on negation. Two of the programs, Stickers and Kids at Work, capitalize on the fact that this activity is taking place with a computer. These programs allow children to work at the pixel level and make detailed changes in any of the predrawn graphics. In both cases the modifications must take place before a graphic is positioned on the screen; Kids at Work, however, lets the child remove a graphic at any time for reworking.

Position. Whether or not a program permits the repositioning of a piece of the whole depends in part on whether or not it is a drawing program or a placement program. In terms of drawing programs, a picture can be erased and redrawn in another position, and Story Maker by Sierra uses this process to produce rudimentary animation. Once a picture is drawn, however, neither of the drawing programs provides a way to move an entire picture from one position on the screen to another.

Two of the placement programs, Kids at Work and Grandma's House, provide the option of moving a part of the whole picture at any time. Pic.Builder produces the effect of a piece moving by allowing the covering of a piece in one position and the uncovering of the same piece in another position. The child moves graphics around the screen in the remaining three placement programs (Kidwriter, StoryMaker by Scholastic. Stickers) but only prior to selecting an initial position for it; once that position is selected the graphic cannot be moved again.

In light of the observations of the two kindergarten children described earlier, the question of whether or not the programs provided the option of modifying the orientation of graphics was of particular interest. The drawing programs let a child position a graphic in any desired orientation, but neither of the programs have a mechanism for changing the orientation of a graphic once it is drawn. Two of the placement programs (StoryMaker by Scholastic, and Stickers) do have functions which change the orientation of predrawn graphics. In StoryMaker any graphic can be flipped from left to right or vice versa anytime before another graphic is selected. The predrawn shapes in Stickers can be rotated through 90 degree turns before choosing one to place in the picture. However, neither program allows the changing of a graphic's orientation after its initial selection.

Creating relationships.

The only programs in which the acts of combining, separating, and rearranging could easily and directly occur were the placement programs Kids at Work, Grandma's House, and Pic.Builder. In each of these programs discrete graphics can be placed and at any time moved and placed somewhere else, thus facilitating the building of groups or the establishing of spatial relationships, and the subsequent rearranging of these relationships. The other three placement programs (Kidwriter, StoryMaker by Scholastic, and Stickers) let the child place graphics in relationship to each other but because of the constraint of being able to only move the most recently selected graphic, it is not possible to rearrange the parts of the whole that is created. As was discussed earlier, the drawing programs do not have any option for "lifting out" a piece of a whole and manipulating it in some way.

Closing discussion.

One important point needs to be made at the outset of this discussion: in terms of maximizing the potential for reversible actions there are no perfect software programs. As the previous discussion suggests, a program may provide one kind of option but not another. It may even be possible that computer technology limits the options that may be part of any one program. The discussion which follows focuses on what ought to be part of "constructive" educational software, not necessarily what can be or what is.

It is important to remember that a computer is a tool; children's hands are removed from the direct manipulation of what happens on the monitor screen. As Piaget notes, when children use tools they do not necessarily notice the relationship between their own actions and the resultant actions of objects (Piaget, 1977). Software which emphasizes building or construction should mimic the real actions which children engage in, e.g., the grabbing and moving actions of Kids at Work and Grandma's House. The undoing function, although utilizing the power of the computer, may be so unrelated to anything a child does in real life that it takes on the quality of magic, which is precisely one of the dangers Cuffaro (1984) sees in the use of computers with young children. Children may be able to imagine an object disappearing, but as Inhelder, Sinclair, and Bovet (1974) have written, "The ability to imagine certain actions which will reconstitute the initial elements does not immediately and automatically lead to reversible actions" (p. 94). Perhaps two modifications might help: slowing the undoing act down allows the child to reflect on the process as being that of removing or emptying, and pairing the disappearance of a graphic or color fill with the appearance of the graphic or color someplace else might also suggest the relationship of actions.

A second "fact" to remember about computers is that at this point in time children are manipulating two dimensional worlds when they are using the computer. They need to have the power to manipulate actions and graphics in as many ways as possible. Programs should provide the option of changing the orientation of a graphic by rotating through a continuous transformation. As Forman (1985) has pointed out, continuous transformations foster the understanding of the relationship between changes that take place when the shape or position of an object is altered. The stretching effect of "rubber band" lines that characterized one of the options in the Rainbow Painter program is a good example of a continuous transformation. When constructing a rectangle, the child is experiencing the relationship between the two dimensions of the figure. It would be a powerful program indeed that provided the option of stretching out any graphic that was on the screen.

A number of sophisticated graphics programs, for example Mousepaint (Apple) and Dazzle Draw (Broderbund) have functions which would be important additions to the type of software for children being discussed in this paper. In both of these programs the user is able to define a section of the screen which is to be deleted, copied, or moved to another part of the screen. Having this power would allow children to be able to identify components of their constructions which could be used elsewhere, and would also facilitate the combining, separating, and rearranging of conceptual and spatial relationships. Kids at Work and Grandma's House provide these options, but children are working with predrawn graphics in these programs. It would be interesting if they had this power for use with their own drawings or constructions.

An "ideal" program, therefore, would combine drawing and predrawn graphics, let children manipulate the results of their actions at any time, and provide them as much manipulative power as possible. They should be able to change, modify, move, and rearrange any and all parts of what they are creating at any time. Such an "ideal" program would maximize the potential for children to experience a wide range of actions which then have the possibility of being related one with the other.

Software Programs

Grandma's House (Spinnaker Software Corporation, 215 First Street, Cambridge, Massachusetts)

Kids at Work (Scholastic Inc., 2931 East McCarty Street, Jefferson City, Missouri)

Kidwriter (Spinnaker)

Pic.Builder (Weekly Reader Family Software, Middletown, Connecticut)

Rainbow Painter (Springboard Software Inc., 7807 Creekridge Circle, Minneapolis, Minnesota)

Stickers (Springboard)

Story Maker (Sierra On-Line Inc., Coarsegold, California)

StoryMaker (Scholastic)

References

Cuffaro, H.K. (1984). Microcomputers in education: Why is earlier better? Teachers College Record, 85(1), Summer, 559-568.

D'Ignazio, F. (1984). Babes in microland. In S. Ditlea (Ed.), Digital Deli. NY: Workman.

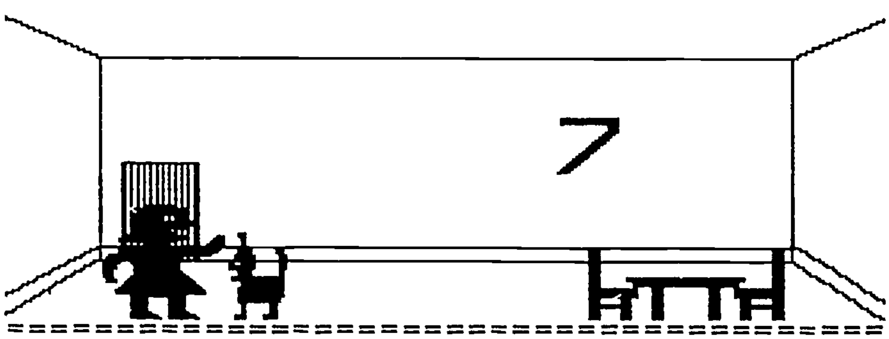
Forman, G.E. (1985). The value of kinetic print in computer graphics for young children. En E. Klein (Ed.), Children and computers. San Francisco: Jossey-Bass.

Inhelder, B., & Piaget, J. (1969). The early growth of logic in the child. NY: Norton.

Inhelder, B., Sinclair, H., & Bovet, M. (1974). Learning and the development of cognition. Cambridge, MA: Harvard University Press.

Piaget, J. (1971). Psychology and epistemology. NY: Viking Press.

Piaget, J. (1977). The development of thought: Equilibration of cognitive structures. NY: Viking Press.



THE NUMBER 7 BECOMES
A PICTURE ON THE WALL.

FIGURE 1

A CHAIR IS SLID INTO POSITION

FIGURE 2a

A PROBLEM OF CHANGING THE ORIENTATION OF THE CHAIR

FIGURE 2b