DOCUMENT RESUME

ED 388 682 TM 023 716

AUTHOR Kafai, Yasmin
TITLE Making Game Artifacts To Facilitate Rich and Meaningful Learning.
PUB DATE Apr 95
NOTE 20p.; Paper presented at the Annual Meeting of the American Educational Research Association (San Francisco, CA, April 1995).
PUB TYPE Reports - Research/Technical (143) -- Speeches/Conference Papers (150)

EDRS PRICE MF01/PC01 Plus Postage.
DESCRIPTORS Computer Assisted Instruction; *Computer Games; Computer Software Development; *Elementary School Students; *Fractions; Grade 4; *Instructional Design; Intermediate Grades; *Learning
IDENTIFIERS *LOGO Programming Language

ABSTRACT

The artifacts (instructional games) created by a class of fourth-grade students engaged in designing educational games are analyzed. To facilitate the analysis, these artifacts were compared with products created by students in a similar design context who were creating instructional software. In both situations, inner-city fourth graders were asked to create products in Logo to teach third graders about fractions. Of the 32 products created, 16 were instructional games. Evaluation of the development process indicated that both instructional and game designers improved their understandings of Logo and their own knowledge of fractions. Analysis of the project demonstrated that the given design task made a difference in terms of the product and process and in terms of the learning experience. Instructional software designers used Logo code to create fraction representations, but game designers usually preferred to use modified geometric shapes. Game designers tended to place the game in the foreground; software designers made fractions central. Game designers also preferred active manipulation and animated scenes. For these students, the openness of the design task varied according to the task specifications. (Contains 16 figures representing computer screens and 17 references.) (SLD)

# Making Game Artifacts to Facilitate Rich and Meaningful Learning

Yasmin Kafai
*University of California, Los Angeles*

Session 31.37: Artifacts of Learning: A Perspective on Students' Learning Processes and Strategies through their Learning Products

American Educational Research Association
April 1995, San Francisco

**Address for Correspondence:**

UCLA Graduate School of Education & Information Studies
405 Hilgard Avenue Mailbox 951521
Los Angeles, CA 90095-1521
phone: 310 206 8150
fax: 310 206 6293
e-mail: kafai@gse.ucla.edu
yasmin@media.mit.edu

> "To see the child as craftsman means to see him or her as a person who wants to be good at something. It suggests that the child continually wishes to take pride in accomplishments and build a sense of integrity about his own work, regardless of the actual level of the work produced. ... The inclination toward craftsmanship no doubt is influenced by effectance motivation which leads to a sense of competence, but the craftsman image is intended to go beyond this to include a more direct link to specific fields of endeavor and to suggest why some activities are so much more compelling to a given child than others. ... Perhaps the most important implication of this image is to suggest that the main purpose of education may well be to provide conditions under which each child can pursue and achieve more advanced levels of mastery within a chosen field or fields of work."
>
> D. Feldman (1980), *Beyond Universals in Cognitive Development*, pp.165-166

David Feldman's image of the child as a craftsman resonates well with the idea of artifacts as representations of learning. This image brings together motivational and cognitive aspects whose importance has been recognized by proponents of project-based learning (Blumenfeld et al., 1991; Lehrer, 1991; Carver, 1991); these educational researchers and practitioners stress the importance of self-directed, complex and personally meaningful activities for students' learning. Constructionist theory (Papert, 1980; 1993) also emphasizes that learning happens particular well if the learner is engaged in creating an external and shareable product such as a robot, computer game or book—in short, an artifact.

One of the pedagogical challenges, among many others, is it to find tasks that stimulate and sustain students' cognitive engagements with the subject matter at hand and, at the same time, are personally meaningful to students. Video games are a central part of children's culture of the late 20th century (Greenfield, 1984; Provenzo, 1991). The number of hours spent in front of screens and the quality of engagement that children show for these games speak for the energizing nature of video games. Yet, video games are rarely in discussion as rich contexts for academic learning. The approach chosen by most researchers to harness the motivating power of video games is to create educational games for students to play and learn with.

A different approach places students in the roles of producers (rather than consumers) of educational video games to capitulize on their motivating power (Kafai, 1993, 1995). In this context, students are asked to design software game artifacts that teach mathematical concepts to younger students. It uses as a foundation the learning through design approach developed by Harel (1988, 1991) that sees the resulting video games as "an artifact that is a shareable, critiquable externalization of knowledge" (Blumenfeld et al., 1991). But it extends this view by considering the artifact not just a product of knowledge telling but also as an interesting and unique piece on its own reflecting students' personal interests and choices. Furthermore, it emphasizes the notion of a fully finished product is part of the learning experience.

As a case in point for the learning experience, I present and analyze the 16 artifacts created by a class of fourth grade students that were engaged in designing educational games to teach fractions to a class of third graders (Kafai, 1995). To facilitate the analyses of the students' end products I chose to compare them with products produced by children in a similar design context yet with one important difference: students were creating instructional software and not instructional games. In other words, although both classes of students were creating a piece of software intended for the use of others, the nature of the artifact itself was different. This comparison will allow me to evaluate the aspect of the nature of the artifact in more detail and the processes which led to its completion.

In the following sections, I will briefly introduce theoretical background, research setting and methods, and students that participated in the two research studies. I will then summarize the main results of the comparative evaluation before moving over to an analysis and discussion of the artifacts created by children in both design situations.

## LEARNING THROUGH MAKING SOFTWARE ARTIFACTS

Learning through making games is based on Harel's (1988, 1991) learning through design approach that sees learning as an integrative process paying careful attention to the interplay of cultural and individual aspects in the construction of interactive objects and how they affect children's thinking, learning, feeling, and socializing. In learning through design students use their knowledge in an active fashion when designing a product—their educational games or software—for the use of others. The design of educational software or games puts the students in charge of their own learning by letting them decide what theme to choose for their software or games, what features of their software or games to implement at what time point, and what questions to ask about fractions. It allows students to approach this task in their own personal way. One central aspect is the length and intensity of involvement in this project. Students were engaged over an extended period of time in designing and implementing all features of their software. This aspect gave students time to mess around and to build relationships in special ways, not only with their computer programs but also with the subject matters involved.

Learning through designing either instructional software or games, integrated the learning of programming and mathematical concepts in a new fashion (see Kafai, 1995, for expanded version of literature review). From a point of view of mathematics education, it emphasizes the role of constructing external representations of fractions as a way to build internal representations (e.g., Harel, 1988; Janvier, 1987; Papert, 1991; Streefland, 1991) instead of using given manipulatives such as Cuisenaire rods. From a point of view of programming instruction, it emphasizes the reflexive nature of programming knowledge: programming is not only good for its own sake but can also be supportive of other learning such as fractions. Learning through designing instructional software or games also makes the learning instrumental to a larger intellectual and social goal: Products such as instructional software or educational games are explicitly designed for use by others. Designing, then, puts students in charge and engages them in a continuous dialogue with their own ideas and with the ideas of intended users and co-designers. Students assume control in their learning through asking questions, gathering information, and putting all this to work by creating an educational games or software for younger students to learn about fractions.

## RESEARCH CONTEXT

Two studies, the instructional software design project (Harel, 1991) and the game design project (Kafai, 1993) provided the research context. In one situation, students were asked to design instructional software in Logo to teach fractions to third graders; in the other situation, students were asked to design instructional games in Logo to teach fractions to third graders. For a period of four months, students in both design situ tions worked every day for one hour on programming their software artifacts. Each day, the students spent the first 5 minutes writing their plans and ideas in their notebooks before they went to work on their own computers for 45 minutes. As students were working on their artifacts, they were allowed to walk around, to see and discuss each others projects. Students then returned to their classroom and wrote again about their experiences, problems and their plans for the next day. These daily sessions were complemented by several focus sessions in which students discussed issues related to their projects, their ideas or difficulties about fractions, and how to represent them. In total, the students spent approximately 60 days on creating their software designs.

Both projects took place at an inner-city public elementary school. The participants had mixed ethnic background and were half boys and girls. Most of them had joined the school only for the fourth and fifth year, hence their beginning programming experience was not very extensive.

## METHODS OF DATA COLLECTION AND ANALYSES

Extensive pre- and post-tests assessed students' knowledge of fractions and Logo programming before and after the project for both design situations. The software development of the student designers was observed every day over the course of four months and resulted in a portfolio for each student: (1) a final software product (a piece of either instructional software or instructional game); (2) daily log files of their Logo programs; (3) daily notebook entries by each student about design plans, progress, and problems; and (4) video interviews conducted before, during and after the project.

The analyses of these data were conducted in three parallel strands: case studies of individual students' development, comparative analyses of students' performance to those instructed by other pedagogical means, and documentation of the class' evolution as a whole.[1] For the purpose of this paper, I will concentrate on the final software products implemented by each student in the respective design, game or instructional, situation. This resulted in a total of 32 software artifacts, 16 of which were instructional games.

## RESULTS

During the 60 days of thinking, designing, programming, modifying, and playing their computer programs, the students were involved in many activities and touched upon many issues. The results from the evaluation indicate that both the instructional and game designers improved significantly their understanding of Logo programming and fractions knowledge when compared to students instructed by other pedagogical means. However, when comparing the differences between pre- and posttest results, the growth of the game designers' knowledge was not of the same order as the instructional designers' performance.

One of the initial expectations was that students in both design situations would perform equally well in understanding fractions and in their programming performance. The results of the evaluation did not provide a clear-cut answer and suggest that one must take a closer look at the nature of the design projects themselves—the software artifacts as they were designed by the students and the processes that the students followed. From the outside, both design activities seem to offer similar opportunities for students to engage in Logo programming and fractions. However, it seems clear that the experience of participating in either the Instructional Software Design Project or the Game Design Project had a substantially different impact on the students' learning and thinking. The following discussion focuses on these differences as they expressed themselves in the students' processes and products: the processes of constructing fraction representations, the centrality of fractions to the software artifact, and the instructional format of the software artifact.

### Processes of Constructing Fractions Representations

One of the central conclusions from the Instructional Software Design Project (ISDP) (Harel, 1991; Harel & Papert, 1990) was that programming fraction representations in Logo contributed in an essential manner to the students' understanding of fractions. More precisely, instructional design provided students with an opportunity to engage with fractions on a new and different representational level: Through Logo programming, they were able to create procedural representations of fractions, thus extending their repertoire of symbolic, written, and pictorial representations. Creating fraction representations in Logo gave students a flexible and different medium to reformulate their knowledge on fractions. Students dealt with represen-

[1] For a more detailed description and rationale of the research methods and analyses, I refer the reader to Kafai (1995).

tations on different levels: they created representations in Logo, they accompanied them with symbols or words, they dealt with their own representations in a different medium (i.e., Logo), and they designed representations for the use of others. Students provided rich and various examples of representations coming from their own experiences, and in this process, their relationship between the everyday and practical side of mathematical knowledge evolved.

Several screens by instructional designers serve as an illustration. One example is Michaela's kitchen, which applied the ratio of water and juice to a fraction task (see Fig. 1).

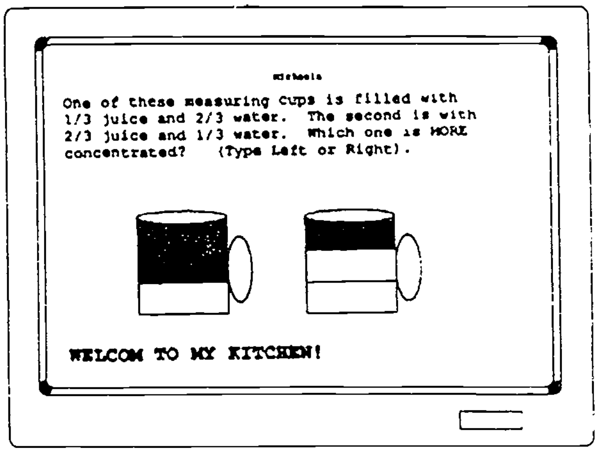



FIG. 1. Instructional Design Project: Michaela's kitchen.

A further example comes from Nicole's dollar screen (see Fig. 2). She used different representations (symbolic, pictorial, and written) for money (one dollar bill, quarters) and combined them with symbolic representations of fractions.

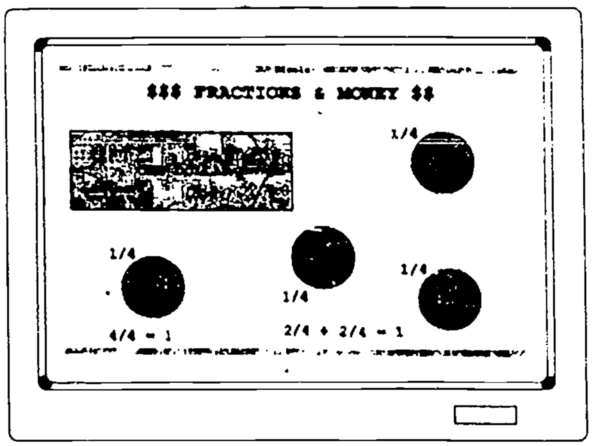



FIG. 2. Instructional Design Project: Nicole's money and fractions.

Another example is Sharifa who made the connection between half an hour and the fraction one half while designing her fractions clock (see Fig. 3).

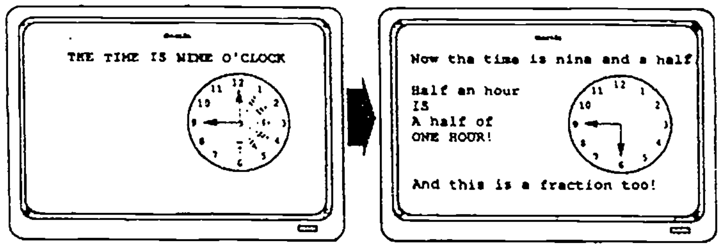



FIG. 3. Instructional Design Project: Sharifa's fraction clock.

The fraction representations that were created by the students of the Game Design Project were different in several aspects from the ones created by the students of the Instructional Software Design Project. For instance, all of Gloria's fraction shapes (Figs. 4 and 5) were pictorial representations of fractions on shapes.

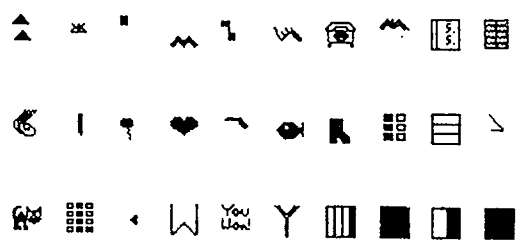


FIG. 4. Game Design Project: Gloria's fraction shapes. Notice the shapes in the second and third rows, which represent 3/6, thirds, 5/12, 1/4, 1 whole, 1/2. The fraction shapes that Gloria designed are not all used in her game The Teacher.

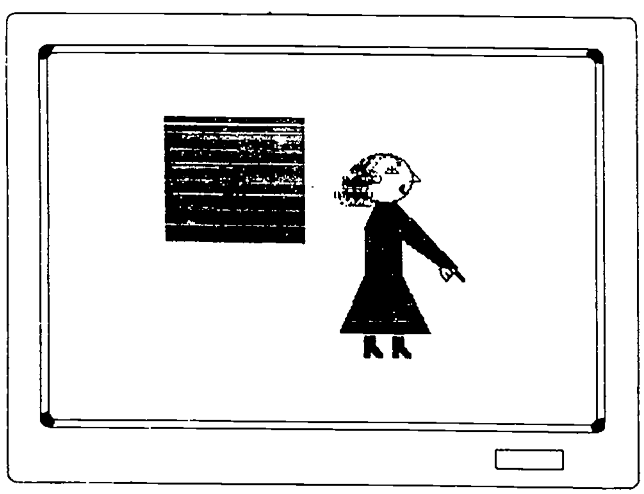


FIG 5. Game Design Project Gloria's instructional screen with the teacher, who was supposed to represent her classroom teacher. In the command center at the bottom of the screen the following question appears: "Can you tell me what fraction this is? Please type."

Another example is Gaby's shapes page, which included her fractions shapes (Fig.

6). As some other students did, she used shapes to create written fraction representations rather than using Logo commands such as PRINT or LABEL.

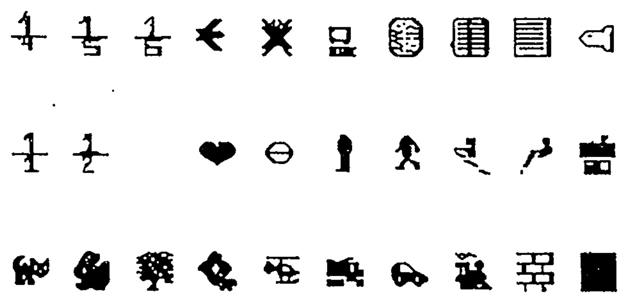


FIG. 6. Game Design Project: Gaby's Fraction Shapes. In contrast to most other students, Gaby used only a few shapes of her whole page for her game. Four shapes for fractions, 1/1, 1/2, 1/4, 1/5, 1/6, the fly, and the spider in the first row.

Barney's instructional screen used fractions and connected them to the use of money (see Fig. 7).

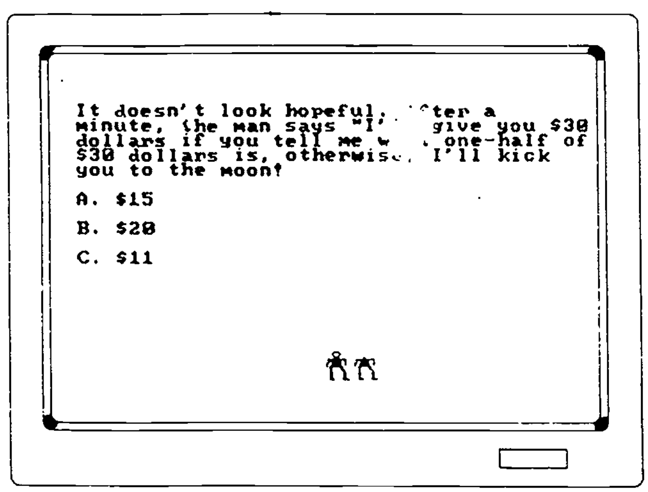


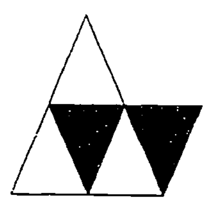

FIG. 7. Game Design Project: Barney's instructional screen, which appeared in the middle of his game story Jose in the Fraction World.

Here the contrast to Nicole's money screen, one of the instructional designers, emerged most clearly. Barney did not provide any pictorial representation of the money involved. Instead, he resorted to a more textbook like representation in presenting a question and three choices. In contrast, Nicole created a screen with different representations of money and how these relate to fractions. The difference between these two design projects resided in the nature and content of creating the fractional representations in Logo.

The fraction representations in the Instructional Software Design Project were designed by programming the Logo turtle. The students of the Game Design Project did not use the turtle to draw these representations; instead, they used the shapes to represent fractions. (A shape in Logowriter is a form other than the turtle form that can be given to the turtle). In order to do so, students had to go to a special page where 30 slots for different shapes were given. In order to create new shapes, students selected a slot and then used the cursor and the space key to fill in an area consisting of 16 x 20 small squares. The process used by the game designers to create their fraction shapes resembled more the process that Streefland's (1991) students used to design representations on paper with pencils than the process of creating procedural representations with Logo. The question is, then, to what extent these external representations (e.g., Gaby's or Gloria's fraction shapes) are a reflection of the quality of internal representations built by the students. The test results do not provide us with clear-cut answers here in favor of the one or the other design situation.

The major reason why students designed fractions in the shapes mode and not in the turtle mode was that the nature of the games required animation. It was much easier to achieve impressions of real-time animation with shapes than with objects drawn by the turtle. The implications of this difference can be found in the interpretation of Question #50 from the Rational Number test (see Fig. 8).

50) What is the denominator of the fraction that tells us what part of the picture below is shaded?



a. five-thirds b. five c. three d. two e. not given

FIG. 8. Rational Number Test Question #50.

One of the skills involved in answering this question was the decomposition of the given picture into its geometrical components, a common process in Logo programming and a skill students acquired in their ongoing programming experience. Therefore, the perceptual distraction (the little triangle outside of the big triangle area) was probably not a distraction for students who decomposed it with their Logo eyes into five geometrical objects. The instructional designers' programming of procedural representations might have helped them in answering this question. The game designers, however, did not go through this process in designing their shapes and, hence, did not perform as well in answering this question. One conclusion is that students of the Game Design Project had a different learning experience with fractions, because they did not use the turtle to create procedural representations of fractions. This points out the importance of programming external representations of fractions for students' deep understanding of fractions.

### Centrality of Fractions to the Software Artifact

A further difference was found when comparing the centrality of fractions to the artifact in both design situations. Although the game design task incorporated the instructional component (these were supposed to be educational games!), students situated their interactions with fractions in a different way. Both projects allowed students to personalize their products, but the Instructional Software Design Project clearly emphasized or foreground the idea of fractions as Figs. 9 and 10 show.

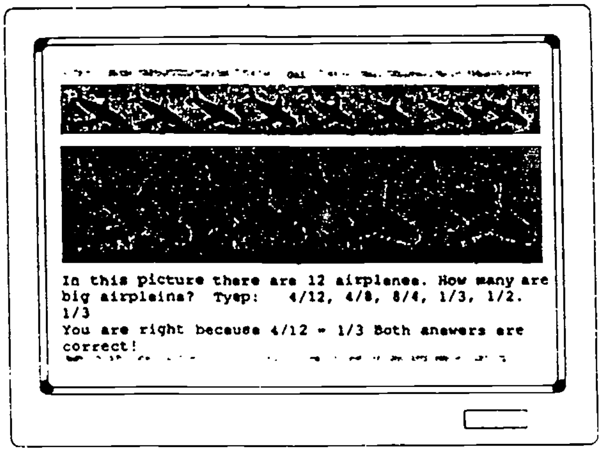



FIG. 9. Instructional Design Project: Oai's screen. Notice how Oai used pictorial representations of airplanes. When the learner gives an answer to the question, the program provided instructional feedback.

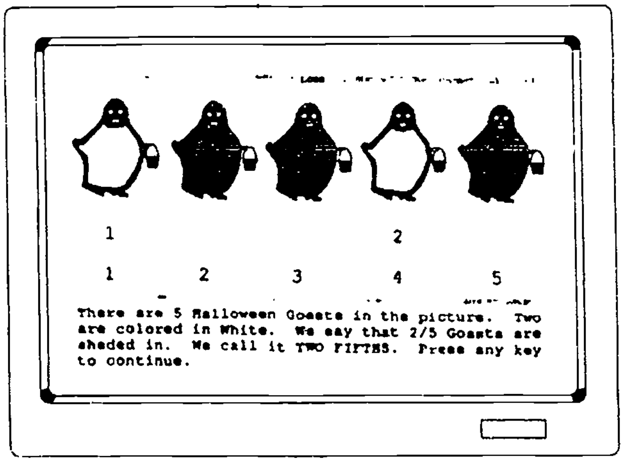



FIG. 10. Instructional Design Project. This is an example for the "Show and Tell" format. The student explains the fraction to the player without any interactions.

In the Game Design Project, creating an interesting or playful game context was the most dominant feature. It drove the students' software products. Different contexts were provided in the game themes for the instructions of fractions, such as skiing down a mountain, running in a maze, or battling with a spaceship. Nevertheless, there were some students who used fractions as their integral/central game idea. One example of a game in which fractions were tangential was Shaun's Fraction Killer: His game was about a person who has to find a hidden fraction wand—a tool that teachers use to teach fractions. In the game itself, the player had to answer fraction questions and could then shoot the attacking spaceship into pieces (see Figs. 11 and 12).

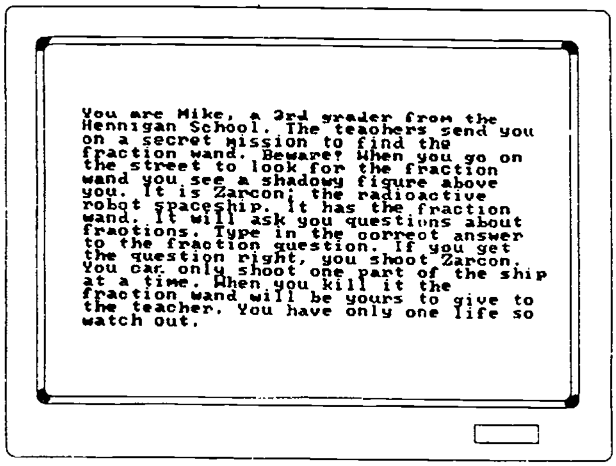



FIG. 11. Game Design Project: Shaun's introductory screen for his game Fraction Killer.

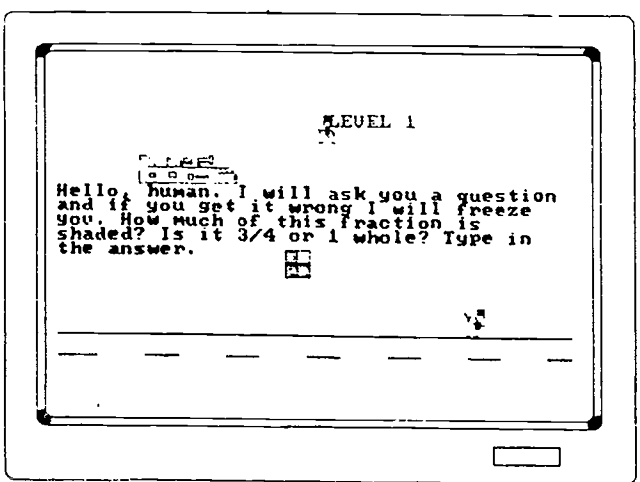



FIG. 12. Game Design Project: Shaun's first instructional screen. If the player answered the question correctly, then a shape moved toward the ship and the middle part of the ship exploded. If the question was not answered correctly, then a beam from the ship moved toward the player and transformed him or her into an ice block.

An example for a game in which the notion of fractions was central was Amy's Greek Myths. Her game was developed around the theme of a map ripped into four pieces (see Figs. 13 and 14). The player reassembled fourths of the map as he or she progressed through the game by answering fraction questions.

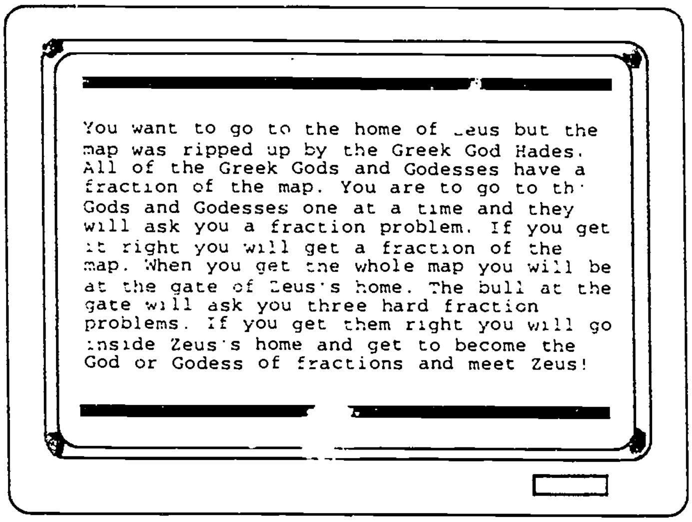



FIG. 13. Game Design Project: Amy's introduction to her game Greek Myths.

FIG. 14. Game Design Project: Amy's shapes. Once the player answered a question correctly, a screen appeared with the corresponding number of pieces (e.g., three questions correctly answered show 3/4 of the map) and a statement: "You have solved this question and now have 3/4 of the map. Please continue."

Most of the games did not center around the idea of fractions; the idea of fractions was in many cases external to the game, a process that could also be observed in the design development of the games. Across projects, there was a similarity in format: The screen displayed one or more fraction representations, accompanied by either an explanatory text or a question to be answered. In the Instructional Software Design Project, all of the screens and the software products focused on the representation of fractions: fraction clocks (see Fig. 3), measurement in fractions, house scenes, and halves. In contrast, in the Game Design Project, the students designed fraction representations that were geometrical areas cut into parts (see Fig. 14), some of them shaded in. The game context dominated the software product, interlacing it with the instructional screens. With a few exceptions, fractions were not central to the product. For most children, aspects of the game predominated (such as Shaun, who had hidden the fraction wand in a spaceship); for a few others (such as Amy, who created a map divided into fractions), the idea of fractions predominated. Students did not establish the same connections between the everyday and practical side of mathematical knowledge. Personal preferences might have guided instructional designers in choosing particular situations or objects for representing fractions (e.g., Debbie's house scene, Sharifa's fraction clock, Michaela's kitchen). This is not to say that students in the Game Design Project did not show these preferences in the design of their games, but instead of displaying these preferences in their choice of fraction representations, they chose to focus on the game context.

### Processes of Programming the Instructional Software Format

A further distinction pointed toward the different formats that designers chose for their software products to present fractions. The instructional designers, for the most part, produced tutorial quizzes. In contrast, the game designers programmed instructional screens embedded in an interactive play mode. One interpretation might be that the tutorial format offered a complex yet not too complicated format to program. Students had to adjust their design ideas to their programming skills, because they learned programming as they went along in designing and implementing their projects. The design ideas of the tutorial could be a mix of what children knew from their previous Logo programming (i.e., to show something) and of what is easy to program. Incidentally, the progression of Debbie in Harel's study reflected this transition from the focus of showing things (e.g., house scene for 1/2, about 2/3 as a difficult fraction used most by teachers) to the scenes where she included the user and offered choices. Most of the instructional designers' products reflect the major emphasis on "Show, Tell, and Quiz," in contrast to the game designers' "Play and Quiz." This was a clear indication that different formats cultivated the development of different programming skills.

The game designers also used the instructional format to program their interactions with the player, but there was very little "Show and Tell." For the most part, the game designers preferred a dialogue format in which they asked questions and responded to the user's replies. For example, Shaun's and Barney's instructional screens showed this very clearly (see Figs. 11 and 7). In addition to this, the game designers embedded these interactions in the playing of the game. For example, in Shaun's game Fraction Killer, the player's correct response initiates an interaction between the figure on the screen (representing the player) and the battleship. Fig. 15 shows Shaun's program; Fig. 11 shows the corresponding screen.

| | |
|---|---|
| To Question 1 | |
| wait.for.user ct pr [] pr [] pr | The instruction PR [] places an empty line on the screen. |
| [] pr [] pr [] pr [Hello human. I will ask you a question and if you get it wrong I will freeze you. How much of this fraction is shaded? Is it 3\/4 or 1 whole?] pu setpos [-10 100] setsh 4 name readlistcc "a | |
| if :a = [3/4] | If the answer is correct . . . |
| [cc tell 3 seth 270 rt 33 setc | Sets the direction and heading of turtle toward the battleship. |
| 10 setsh 3 rt 1 setpos [85 -35] | |
| ct st repeat 8 [pu fd 20 wait 5] | Moves the beam up to the battleship. |
| setpos [-55 50] setsh 2 setc 0 | Sets the shape for the exploded middle part of the battleship. |
| pd stamp pu setc 8 setsh 5 pd | |
| stamp pu wait 20 gp "games2 question2] | Gets the next page GAMES2 and the next question QUESTION2. |
| if :a = [1 whole] | If the answer is wrong |
| [seth 270 rt 30 ct setpos | Sets the shape to a ball that goes from the battleship to the player. |
| [-40 40] setsh 9 st setc 15 repeat 8 [pu bk 20 20 wait 5] wait 10 pr [] pr [] pr [] pr | |
| [A freeze ray hits you!] | Prints out statement on screen after hit. |
| setpos [100 -50] wait 5 | |
| setsh 2 setc 0 pd stamp pu | Sets the shape to an ice cube. |
| setsh 6 setc 11 pd stamp pu ct wait 5 pr [] pr [] pr [] pr [Now you are an ice cube fore life!] pr [] pr [] pr [Better get used to the north pole! Pretty funny huh?] tone 200 10 tone 100 10 tone 200 10 cc rg pr [Game over (You're dead)] tone 100 5 tone 75 5 tone 25 5 setsh 30 setc 5 st pd stamp pu] end | |

FIG. 15. Game Design Project: Shaun's Logo code for the first question.

Shaun's Logo code shows how he programmed the intricate interplay between the player and the interactive feedback on the screen. In this case, the right or wrong answer is not accompanied by direct instructional feedback (e.g., "Wrong answer") but with the corresponding action on the screen. In his programming, Shaun had to coordinate a number of aspects: the different positions of the shapes on the screen (the battleship and the player), the change of shapes (from player to player being destroyed to player turned into ice cube), and the timing of animation. Shaun's programming of the fraction representations, the instructional interaction, and the animation is exemplary of most other game designers.

One exceptional example to the instructionist format in the Game Design Project was Rosy, who devised a fraction design tool (see Fig. 16). Rosy's game included a hunt for a treasure, but in the middle of the game the player comes to a page that invites him or her to move around the turtle with the arrow keys on the screen. Afterward, the player is told "Now draw a fraction. When you are done press Q" without giving specific instructions about which fraction to do.

| | |
|---|---|
| to startup | |
| cc ct rg | Clears the screen and command center. |
| type se [Use the arrow keys to move the turtle. When you are done press S.] char 13 | Appears in the command center. |
| draw | Calls procedure DRAW. |
| end | |
| to draw | |
| drive ascii readchar | Starts the recursive procedure DRIVE. which reads in key strokes in ascii format. |
| end | |
| to drive :key | |
| pd if :key = 72 [seth 0 fd 5] | Arrow key up and five steps forward. |
| pd if :key = 80 [seth 180 fd 5] | Arrow key down and five steps forward. |
| pd if :key = 77 [seth 90 fd 5] | Arrow key right and five steps forward. |
| pd if :key = 75 [seth 270 fd 5] | Arrow key left and five steps forward. |
| pd if :key = 115 [rg cc DIR draw] | Key "S" and calls procedures DIR and DRAW again. |
| pd if :key = 113 [gp "GAMES6 stop] | Key "Q" gets a new page and stops the recursive DRIVE procedure. |
| drive ascii READCHAR | Procedure DRIVE calls itself again. |
| end | |
| to dir | |
| type se [Now draw a fraction. When you are done press q] char 13 | |
| end | |

FIG. 16. Game Design Project: Rosy's code for the fraction design tool. My explanations of her programming are included in the right column. The numbers associated with the keys are the ascii code for the corresponding arrow keys up, down, left, and right, and the letters S and Q.

The following scene serves as a good example of how Rosy herself used the fraction design tool to explore fractions.

[Rosy is moving the arrow keys and is in the process of drawing a rectangle divided into four smaller ones.]

*Yasmin*: What fraction will this be?

*Rosy*: Wait, I am not done yet [she divides up one of the rectangles into eight smaller ones by moving the arrow ].

*Yasmin*: So what fraction is it now?

*Rosy*: One fourths or ..... thirty twos .... [her eyes are on the screen and her lips move silently before she continues] It is eight thirty seconds, eight thirtytwos.

Rosy was the only one in the game design project who had programmed a fraction design tool; as she learned how to program keys on the keyboard, she turned them into a tool for making fractions. Besides her powerful conceptual teaching idea, this program segment also showed the complexity of her programming to give the player/learner at the same time freedom to explore fractions as well as to keep him or her in the context of the game (once the player decided he or she has enough of "drawing fractions" and pressed the Q-key, the program took care of bringing the player back into the game by calling LogoWriter page GAMES6).

## DISCUSSION

The analyses of the software artifacts pointed out that the given design task—instructional games versus instructional software—makes a difference, not only in terms of the design product and process but also to a certain extent for the learning experience:

- students in each design situation used different programming techniques to create fraction representations: instructional designers used Logo code to create fraction representations whereas most game designers preferred to use modified geometrical shapes.
- students assigned fractions a different role in the software product: for instructional designers fractions were central to their software whereas for most game designers the games were in the foreground.
- students chose different instructional formats for the artifact: instructional software designers preferred a "Show & Tell" mode whereas the game designers preferred active manipulation and animated scenes to engage the younger students in fractions.

These results indicate that the instructional design context integrated not only the students' programming and design of fraction representations in a different way but also integrated the idea of fractions into the software product. A result that needs to be further discussed concerns the quality of subject integration into the software product. Most of the software products could be divided into two categories: extrinsic and intrinsic integration of subject matter in the software. The instructional software projects demonstrated more clearly the intrinsic integration approach as all the screens and questions designed by the students focused on fractions. When the game designers were asked to construct an educational game to teach fractions, most of them used the extrinsic integration method. As in commercially made games, the greater number chose the easiest extrinsic integration by stopping the action at key places to ask the player a question. It is a strength and a weakness of the extrinsic integration that domains of knowledge become almost interchangeable. It is a strength because the integration is relatively easy: When answering a question correctly is what allows the next move in a game, the question can be on any topic. But this is also a weakness, because it causes the designer to loose the incentive to think deeply about the particular piece of knowledge.

A conclusion to be drawn from this difference is that the instructional design did not permit large degrees of latitude, which is at least one reason why the various instructional software products speak more directly to the idea of fractions. The game projects, in contrast, gave children more latitude in determining how central the notion of fractions would be to their games. This was particularly true for the integration of fractions into their game design. For most children, aspects of the game predominated (e.g., Shaun, who hid the fraction wand in a spaceship); for others, the idea of fractions predominated (e.g., Amy, who created a map divided into fractions).

The given artifact initiated not only student' thinking about factions in different ways but also impacted their perception of the final product. Students tended to think of their "project" in different ways: in terms of the entire software or in terms of individual screens. There was a stronger incentive for the games to be thought of as a whole. One indicator of this was the story format chosen for the game design by most students. For a typical narrative, the designer introduced the story in the beginning of the game and led the player through a number of adventures before bringing the game to a conclusion, hence resolving the tensions. In contrast, Debbie (an instructional designer) considered each instructional screen a project in itself. For example, Debbie's house scene took only a few days to implement (March 26–30) and could be evaluated separately from other screens that she had designed before or after. There was no interdependency among the different screens in regard to the instructional purpose or the player's ability to use the software. A game, however, even though it consists of different scenes, cannot be played until it is finished. The instructional software was in the form of an interactive lesson with the purpose of teaching something, whereas the educational game was a piece of software that performed its instructional purpose in the context of a game. It is my interpretation that students' thinking about the game as final product instead of individual scenes made the design task an even more challenging enterprise. The degrees of freedom accorded to the game designers placed a heavier emphasis on the game aspect to the detriment of the subject matter.

The combination of fractions with games raised the issue of competing design issues: Learning fractions is a subject that many students feel is rather boring (the statements in the pre-interviews were a testament to this; even students who were "good" in mathematics stated that they found schoolwork with fractions boring), whereas the playing of games is something "not like school"—it is "fun." The decision of many game designers was to focus more on the "fun" part of the game design task (meaning the design of the game) than on the fractions task. This preference may indicate that students were driven in their design by a simple motivation: They decided to invest their energies in the design of the game with the understanding that what makes a game exciting is when it is fun, and that learning fractions could only be fun in a fun context. As Shaun expressed in a reflection on the educational aspects of arcade games, they were concerned how to design a game that would be fun but at the same time also educational. In this context, the students also broke away from standard representations, but in a different way. They continued to use designs of fraction representations, but found nonstandard contexts (e.g., haunted houses, fraction killers, games with gods) to think about fractions. The games designed by the students point out how difficult it is to find a game idea that on one hand is central to fractions but on the other hand is also fun.

A comment on the instructional pedagogy employed by most students might be at the right place. In both design projects students took the tutorial or "Show and Tell" format as the most used format for introducing their learners to fractions. An interesting observation here is that the student designers produced an instructionist type of software in a constructionist learning environment (see also Lehrer, 1992). One possible explanation might be that the designers reflected in the adoption of the tutorial format their personal and current ideas on teaching. Their thinking about how one ought to be taught might intersect with their own experiences of the ways they are taught in school, at home, and through other cultural media. Many of these models emphasized the instructionist mode of teaching—passing knowledge on. Thus, the tutorials and games could be considered a reflection of the students' own experience and knowledge of teaching.

## CONCLUSION

The lessons learned from these analyses spell out some suggestions for the "design" of future design tasks. Students might be faced with competing design issues when the task is too complex. The conclusion to be drawn from this finding is that the openness of the design task can vary according to its specifications. The question remains how to engage students in more dialogue about making intrinsically integrated educational games. In a first take, I explored whether game design combined with a different subject matter would result in better integration. In a follow-up study of the game project, I asked a class of fourth graders to design games to teach third graders about the solar system. One of my reasons for this choice was that there are many well-known games, plays, and stories around the theme of space. I assumed that this particular choice would facilitate intrinsic integration of knowledge about the solar system into the game theme. A preliminary analysis of my data gathered during the five-month-long project led me to the following observation: Yes, children's game themes were all centered around the space theme, but a majority of students were still creating games with extrinsic integration.

One of the pedagogic goals of future work is to encourage wider use of the more difficult but epistemologically more rewarding process of intrinsic integration. An essential step in the direction of encouraging intrinsic integration is creating a culture in which this approach is recognized and valued by teachers and well represented in attractive example games made available to the students.

## REFERENCES

Blumenfeld, P. C., Soloway, E., Marx, R. W., Krajcik, J. S., Guzdial, M., & Palincsar, A. (1991). Motivating project-based learning. Sustaining the doing, supporting the learning. *Educational Psychologist, 26*(3 & 4), 369-398.

Carver, S. (1991, April). *Interdisciplinary problem Solving*. Paper presented at the American Educational Research Association, Chicago, Il.

Greenfield, P. M. (1984). *Mind and Media. The Effects of Television, Video Games, and Computers.* Cambridge, MA: Harvard University Press.

Harel, I. (1988). *Software Design For Learning: Children's Construction of Meaning for Fractions and Logo Programming*. Unpublished Doctoral Dissertation. Media Laboratory, Massachusetts Institute of Technology, Cambridge, MA.

Harel, I. (1991). *Children Designers*. Norwood: Ablex.

Harel, I. & Papert, S. (1990). Software Design as a Learning Environment. Interactive Learning Environment, 1, 1-32.

Janvier, C. (Ed.). (1987). *Problems of presentation in the teaching and learning of mathematics.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Kafai, Y. B. (1995). *Minds in Play: Computer Game Design as a Context for Learning*. Unpublished Doctoral Dissertation. Harvard Univerrsity, Cambridge, MA.

Kafai, Y. B. (1995). *Minds in Play: Computer Game Design as a Context for Learning*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Lehrer, R. (1992). Book review. *Journal Educational Computing Research, 8*(2), 255–258.

Lehrer, R. (1991). *Knowledge as design*. Paper presented at the American Educational Research Association, Chicago, Il.

Palumbo, D. B. (1990). Programming language/problem-solving research: A review of relevant issues. *Review of Educational Research, 60*(1), 65–89.

Papert, S. (1993). *The Children's Machine*. New York: Basic Books.

Papert, S. (1991). Situating Constructionism. In I. Harel & S. Papert (Eds.). Constructionism. Norwood: Ablex.

Papert, S. (1980). *Mindstorms*. New York: Basic Books.

Provenzo, E. F. (1991). *Video kids: Making sense of Nintendo*. Cambridge, MA: Harvard University Press.

Streefland, L. (1991). *Fractions in realistic mathematics education*. Dordrect: Kluwer.