7. Stepping Back

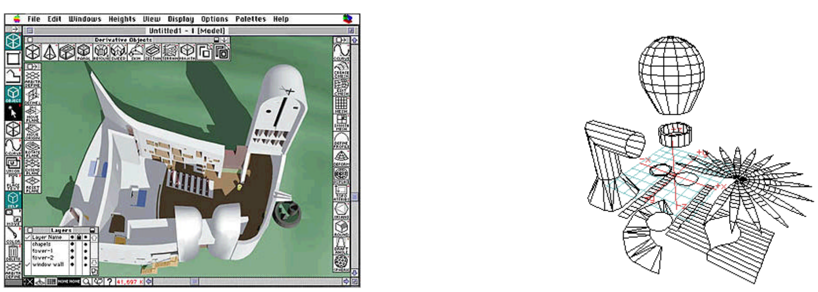
7.1 Related Work

7.1.1 Systems that Generate Folding Nets

The problem of unfolding three-dimensional models is not a new one (c.f. Samek, et al. 1986), nor is it one limited to the domains of paper sculpture or spatial cognition. Architects and engineers use folding nets to construct scale models of their projects; package designers produce many paper prototypes of their ideas; and people who work with sheet metal often use paper as a first step in testing their designs. The first two systems discussed below were designed with these types of activities in mind. The third system was created for children, but was built with design principles substantially different from those in HyperGami and JavaGami.

- *form·Z*

The software package form·Z (auto·des·sys) is a high-end design tool in which object unfolding comprises a small part of an extensive package of rendering and modeling tools. Although it supports sophisticated graphics and modeling capabilities, the product interface is extremely complex -- it is geared toward an audience of professional architects, special effects designers, and engineers.

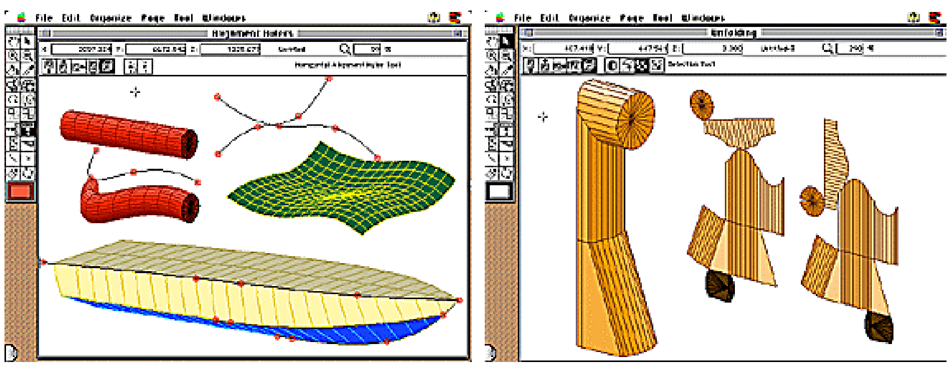


(a) (b)

Figure 7-1. (a) One of the shape rendering interfaces in form·Z; (b) Solid objects and their folding nets in form·Z. (From the form·Z website, (W1)).

- *Touch-3D*

Touch-3D (Script Software) is a software environment oriented expressly toward creating tangible real-world models. It supports an extensive collection of tools to model objects in 3D, and the software will unfold those objects into folding nets. Touch-3D's rendering capabilities are not as extensive as those in form·Z, but this software package has made the generation of folding nets for three-dimensional objects its primary activity and has a somewhat less daunting interface than that of form·Z. However, the environment is still quite complex and is clearly designed with the needs of engineers and architects in mind rather than the concerns of children or hobbyists.

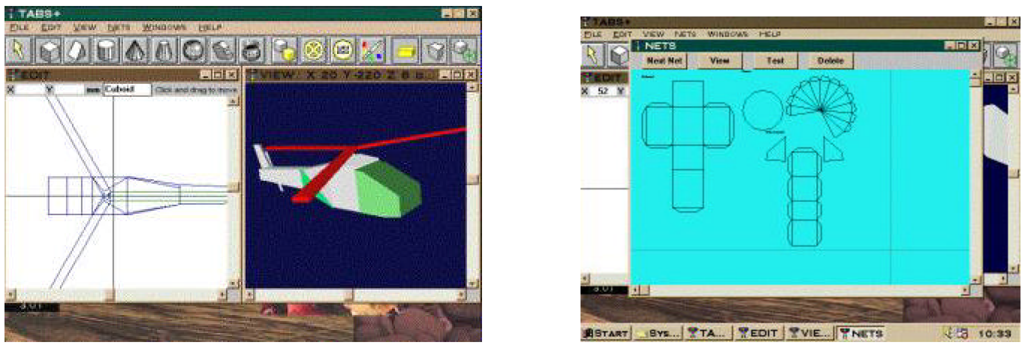


(a) (b)

Figure 7-2. (a) The shape-modeling interface in Touch-3D, and (b) an unfolded model. (From the Touch-3D website, (W2)).

- *tabs+*

Unlike form·Z and Touch-3D, tabs+ (ASPEX software) is a design tool built for children in grades 4-10. This software package supports a set of built-in shapes which children combine on-screen to create models. The interface for shape combination is relatively straightforward, with windows for wireframe and solid representations and a toolbar across the top as shown in Figure 7-3 (a) and (b). However, the primary limitation of this software package is that children do not have functions such as truncation, slicing, and linear maps to create their own customized shapes. Moreover, the nets of the models can be printed to create a physical model of the design, but there are no built-in paint tools for students to do online decoration of their nets.



(a) (b)

Figure 7-3. (a). The modeling interface for tabs+, and (b) the folding net interface. (From the tabs+ website, (W3)).

7.1.2 Educational Software for Spatial Thinking

The three software systems discussed below are environments designed to promote different types of spatial thinking. The first system is oriented towards elementary-school children; the second is appropriate for upper-elementary and middle-school children; the third is primarily intended for older children and adults.

- *Shape Up!*

Shape Up! (Sunburst Communications) is a software environment for children which provides five different activity areas: Plato's World, Pattern Block World, Tangram World, 2D World, and 3D World. Each activity area features an interface similar to the one shown for Plato's World in Figure 7-4 below. Children drag the objects on the screen to design pictures -- or in the case of the Tangram and Pattern Blocks world, they may select an online challenge -- a picture or pattern to recreate on the screen.

The interface of Shape Up! includes tools to scale, rotate, and produce the mirror image of a shape, along with simple decoration tools to paint shapes in solid colors or to add text or other pictures from a clipboard to the screen. The software also includes a function for printing blank folding nets of the Platonic Solids, but otherwise does not approach the concept of unfolding solids.

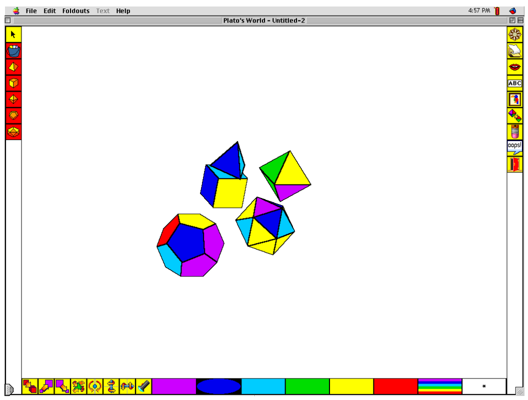


Figure 7-4. Plato's World in Shape Up! (Sunburst Communications)

Shape Up! provides children with practice in plane manipulations with tangram-like objects, and gives them some practice with rotations of 3D-shape representations. This is an instance of software that might also be combined with mathematical real-world manipulatives such as actual tangram blocks and models of the Platonic Solids (perhaps built from the folding nets provided in the software) so that children may work with both concrete and on-screen objects.

- *The Factory*

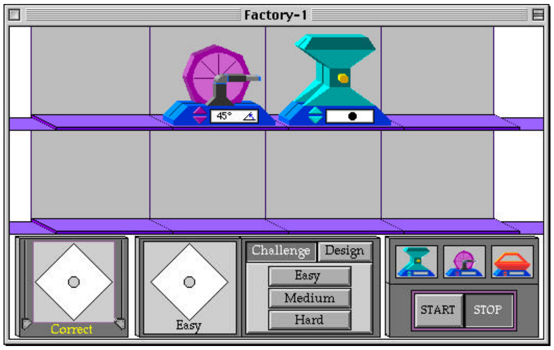


Figure 7-5. Screen interface of The Factory (Wings for Learning).

The Factory (Wings for Learning) is a game in which children place machines on an assembly line to operate on a card that passes through them. The machines perform three types of operations: they draw lines, punch holes, or turn the card. The child's objective is to position the appropriate machines along the assembly line so that a blank card moving through them ends up with lines and holes in the same orientation as a provided template.

McClurg and Chaillé (1987) studied children's performance on the ETS Mental Rotations task (described in more detail in section 5.7) and found that interaction with this software environment produced "significant effects" (p. 108) on their spatial thinking as measured by this test.

- *VizAbility*

Scott Kim, the puzzlemaster described at the beginning of Chapter 2, was part of the team that produced VizAbility (PWS Publishing), a cd-rom package designed to enhance visual thinking abilities. The focus of this package is primarily on drawing and learning to "communicate and invent ideas through quick sketches" (Kim, 1999, (W4)). The software includes visual puzzles, drawing lessons, and interviews with visual thinkers.

Overall, this variety of packages and applications suggests a natural direction for future work. It would be informative to study changes in children's spatial thinking as a result of interacting with these software environments, and to compare those results to the JavaGami and HyperGami results described in this paper to determine whether there are any fine-grained differences in the specific types of spatial thinking promoted by the different systems. It would likewise be of interest to see whether students place as much emphasis on personalizing their experiences with the software in these alternative systems.

7.1.3 Related Work in Spatial Cognition

Spatial assessment in JavaGami and HyperGami in part expands upon work in which McClurg et al. (1997) performed a small exploratory study with 12 children in grades 6-9 using HyperGami to assess some aspects of their spatial learning before and after six hours of work with the software. The results of the study in general indicated "that HyperGami is a rich environment for developing spatial visual thinking skills" (p. 257).

The study included a net-solid matching task, a timed net-assembly task, and a face/vertex-counting task in which students were asked to predict numbers and types of faces and vertices after a solid was truncated. The study's conclusions were that (1) "each subject made gains in their ability to recognize the nets of solids" (p. 262); (2) students in the timed-net assembly task group "reduced time for folding (a) complex net" (p. 262); and (3) students' face and vertex predictions before and after truncation did not show improvement after work with HyperGami, but the investigators felt that "(t)he students in this experiment did not seem to have the knowledge and experiential background necessary for abstract reasoning problems dealing with truncation." (p. 262).

McClurg et al. caution against over-generalizing the results of the study -- there were just four children in each experimental group -- but they found early results promising and suggested "... (f)urther study aimed at identifying the cognitive processes involved." (p. 262).

7.2 Contributions

This dissertation offers a perspective into cognitive and social aspects of combining manipulatives and software in the context of activities to enhance spatial thinking. The cognitive results are promising: keeping in mind the caveats discussed at the end of Chapter 5, children have shown increases in sophistication in their descriptions of shapes, in their renderings of folding nets of shapes, and on their performance on standardized tests of spatial thinking. But looking at the larger picture, what HyperGami and JavaGami have also given children is a way to personalize their experiences with software while engaging in activities which are both spatially and mathematically rich.

At a time when educational and recreational activities of children have become increasingly tied to virtual environments -- high-powered video games, web surfing, online chat rooms, and even electronic pets -- educators and software designers are well-advised to keep sight of the idea that children *need* hands-on experiences with materials. No matter how rich or elaborate the virtual world, in the opinion of the author, there is no substitute for the cognitive experiences that take place when children build and handle objects like Froebl's materials of kindergarten. This is not to say that computers should be abandoned; rather, we can use them to create to create a new class of personalizable, concrete manipulatives. By employing real-time modeling capabilities of computers, children experiment with dynamic manipulatives on-screen, add their own personal touches by changing the forms or designs on the manipulatives, and then create concrete versions of these objects. These customized objects have affective significance for children and tie them closer to mathematical activity: Cuisenaire rods are wonderful mathematical objects, but what child would want to present one to a parent as a gift?

These design ideas are not only limited to the domains of solid geometry and spatial cognition. Designers can take these lessons to a wide variety of educational domains: one might envision combining manipulatives and software when teaching children about kaleidoscopes or tessellations or optical illusions.

To elaborate with one specific example: suppose that a designer wanted to apply some of the principles discussed in this thesis to the domain of automata building. As in polyhedra-building, the construction of moving toys -- like the flying carpet machine shown in Figure 7-6 below -- is tied to activities rich in educational content (cf. Onn and Alexander, 1998; Williams and Jinks, 1985; Eichelberger and Larson, 1993). And also as in the activity of polyhedra-building, we can envision a seamless moving back-and-forth between virtual and real-world activities. The designer might create an environment where a student explores and combines a multitude of mechanical devices -- such as cams, linkages, levers, gears -- on-screen. The virtual environment allows him to experiment freely with different combinations of devices to develop the specific type of motion he desires in the automaton. By developing different combinations of devices, the student can in effect build his own algebra of mechanisms.

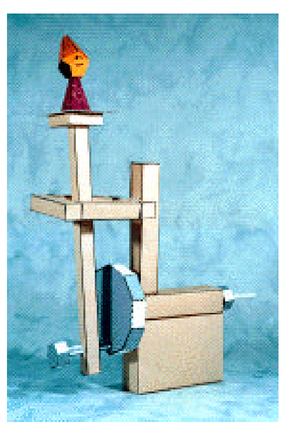


Figure 7-6. *Scheheherezade.* A crank-shaft automaton designed by the author.

Once he arrives at a design he wants to build, the student may then employ the system to produce a template which he can use to construct his real-world device out of paper, balsa wood, or sheet metal. As illustrated in the work with HyperGami and JavaGami, an important goal of this kind of software environment is to bring activities back to the real world. It is not only essential for the student to have tactile experience to hone what educators in design technology call "making skills" (Ritchie, 1995, p. 133) -- work with wood, paper, glue, and scissors -- but at the end of the process, the student will have a real toy that he may take home and share with others. As we have seen, possessing a form of social currency which he can show to others, present as a gift, or proudly place on display, helps the child to make an educational activity expressly his own.

Another design principle illustrated in this thesis is that the computing environment should also enable children to customize their activities. Automata are more than just a collection of gears. They can be supremely funny, disgusting, or anxiety-provoking, and people familiar with the world of automata can often identify the artist of an automaton by the touches of personal style injected into the object, just as in other kinds of visual art. The designer should take this into account and build tools which allow children to not only to customize mechanisms, but also to add personal touches to the general appearance of the contraption, enabling them to tell a story through both the movement and appearance of their machine. At the very least, the designer might integrate online tools so that children can integrate different animals, vehicles, or background scenes into the automata they build.

(a) (b)

Figure 7-7. (a) *The Barecats* by Paul Spooner and Matt Smith. (b) *The Mill Girl and Toff* by Paul Spooner. (From the Cabaret Mechanical Theatre website, (W5)).

The ideas and techniques discussed in the assessment portion of the thesis may be adapted to other domains as well. To continue our automata example, an educator might assess children's learning about mechanisms by analyzing the vocabulary children use to describe machines before and after work with the system; he might develop pre- and post- analysis techniques for children's sophistication in rendering concepts of mechanisms, gears, or linkages; and he might administer tests of mechanical ability to determine whether any transfer of children's learning takes place. Just as importantly, he might conduct detailed case studies similar to the profile of Jesse in Chapter 6 to develop a finer-grained picture of the way a child interacts with the online and construction-based activities of the environment.

7.3 Future Plans

7.3.1 Assessment

As discussed at the end of Chapter 5, assessment of students' work in HyperGami and JavaGami is still at an early stage. While the results of pilot studies are informative, future studies should include a larger number of subjects and comparison with control groups. The results from the children's' folding net renderings are intriguing, and more study in this area may lead to a finer-grained characterization of what it means for one folding net to be more sophisticated than another, and if a hierarchy of experience-based stages of folding net drawings exists. Further exploration of students' transfer of learning should also take place in the context of standardized tests and other measures.

7.3.2 Interface work in JavaGami

JavaGami is just a first step in the creation of a rich tool for paper sculpture and spatial thinking. The most obvious features currently missing from the software are undo operations for both net decoration and solid modeling and ways to easily save, load, and print solid objects and folding nets. A future version of JavaGami would be enriched with multiple representations of folding nets, functions which allow children to rearrange folding nets by dragging the mouse over polygons on the screen, and tools that allow children to see how the decorations on their folding nets will map to finished solids. Advanced functions -- such as giving the student the option of truncating a solid at a vertex while varying the distance of truncation along particular edges -- would provide even more flexibility in creating custom solids.

As an environment for artwork, later versions of JavaGami should support a much richer collection of tools for children to decorate folding nets including (as suggested by students): tools for typing on folding nets, an online collection of rubber-stamps for children to place pre-drawn pictures onto nets, an online help system, an extended assortment of colors (as well as tools for children to mix their own online colors), and the inclusion of pattern and texture renderers. A function to place tabs onto the edges of nets would also help the crafting process.

7.4 What is REAL?

> "What is REAL?" asked the Rabbit one day, when they were lying side by side near the nursery fender ... "Does it mean having things that buzz inside you and a stick-out handle?"
>
> "Real isn't how you are made," said the Skin Horse. "It's a thing that happens to you ... once you are real you can't become unreal again. It lasts for always."
>
> -- from *The Velveteen Rabbit*, (Williams, 1922)

HyperGami and JavaGami are intended to be instances of software in which children create things that are *real* in the sense evoked by Margery Williams' classic story. Ideally, HyperGami and JavaGami objects are not real merely in the sense that they are tactile; they are real because they hold emotional meaning as well. It is perhaps no exaggeration to say that much of the current culture of educational computing focuses on the building of systems that "buzz with a stick-out handle"; but the argument of this work is that educational computing would be better served by focusing on activities that are real to children.