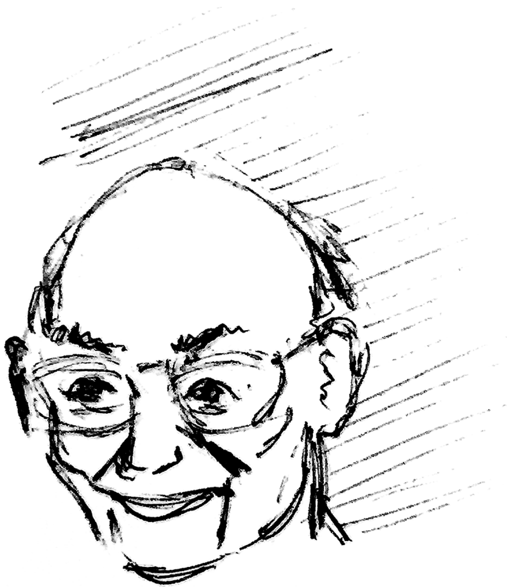
Marvin Minsky and the Ultimate Tinkertoy

Alan Kay

Let me confess in advance: I love Marvin! I say this in the present tense because only his physical body passed away—Marvin lives on vividly in the minds of all who were fortunate enough to know him, either personally or through his many talks and writings like the ones in this book. I think most people who read these will also come to feel they know him, as "the Marvin of Ideas" starts to live on vividly in their minds. As Mike Travers points out in his excellent introduction to this book, one of Marvin's main rules of thumb is that we should try to internalize great thinkers inside our own minds in such a way that *they can keep thinking to help us think*. I'm going to try to do this—to "channel Marvin"—in this afterword.

I have especially loved this particular essay of Marvin's because he wrote it for child readers to help support a large vision he believed in: that the new medium of computing, if really understood and used well, could make an enormous positive difference in helping children grow up with a much more powerful view of the world and a set of mental tools for dealing with that world. He appeals directly to children because they are the touchers of computers most likely to see what computers are really about, whereas most adults—including many computer professionals—have wound up adopting very weak notions about computing.



A key difficulty in learning “powerful ideas” is that we always start “where we are,” and try to fit new ideas and things into what we already think we know. If the new ideas/things are very different, then we often either bypass them entirely or force them into distorted meanings that fit our current “private universe”: the one between our ears! Marvin would like children to always have the thought: “I think this idea/thing is this way, but it could also be another way, and it could be made from things I’m not thinking about at all.” (Marvin was great at this rule of thumb!)

Marvin uses analogies to Tinkertoys because there is an amazing range of things that can be made from a few simple parts—very like bricks and other simple fundamental building blocks, and especially like “computer stuff.” Marvin was a fabulous Tinkertoy builder as a child and knew, as every child discovers, there is immense satisfaction in building something that turns out *just right*. The learning comes freely from the joy of happily focusing on making something neat. As Maria Montessori pointed out many years ago, play is the work of the child.

The first thing we usually do with Tinkertoys is make structures that go beyond the parts. If we start fooling around with the parts we'll come up with *something*. I wound up with a skyscraper tower (figure 1.2).

We can see right away that what's most important about Tinkertoys is not the details of the components. Other construction materials will have their own basic components and ways to be put together. Most of the powerful ideas are in the designs of the combinations.

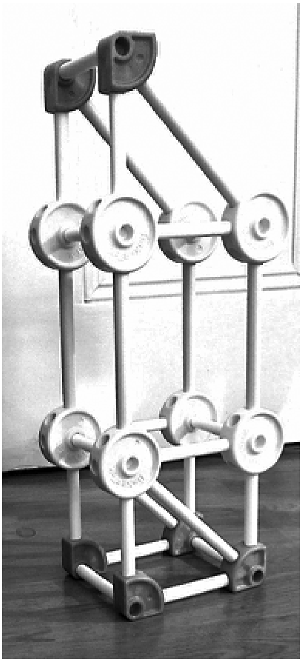


Figure 1.2

A good way to think of this powerful idea is: *As more complex things are made, architecture dominates materials*! In other words, what lies behind what we touch and think about are often wonderful things that are mostly *organizational* in nature. As Marvin says in essay 1, "What matters is more how the parts affect each other—and less about what they are, themselves."

Some of the Tinkertoy parts allow movement, so we can make a toy vehicle that we can push or pull around. Maybe a tractor with a cab?

What about vehicles that are automobiles, that can go by themselves? Tinkertoy does not provide direct ways to make such vehicles, but we can improvise them. For example, we can put the tower we made on our vehicle, make a weight from something heavy (like coins in a baggie), and use our string to wrap around the rear axle to be pulled and turned by the weight.

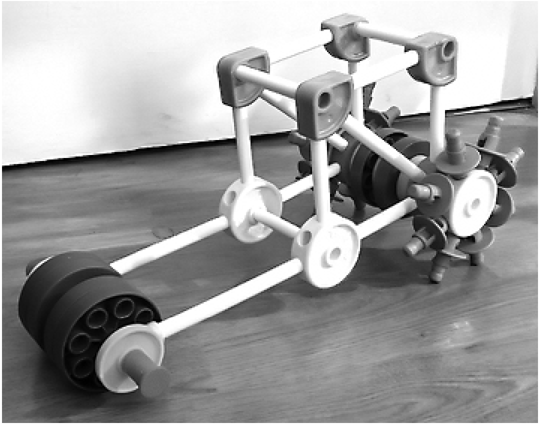


Figure 1.3

If we look at our self-powered vehicle, we can see that it is not smart enough to prevent itself from crashing into a wall or falling off a table. Could we make it smart enough to avoid these? And what do we mean by “smart”?

Figure 1.4

Marvin worked with Seymour Papert for many years in a joint quest for good working models of intelligence. They started looking beyond traditional programming and computing for better parts, organizations, and designs to make “thinking stuff.”

One source of inspiration came from the field of cybernetics, which Marvin had delved into while in college, and another from the autonomous and somewhat intelligent robot tortoises of Grey

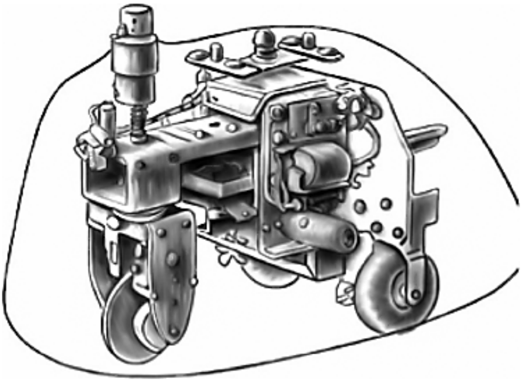


Figure 1.5

Walter, which could not only negotiate arbitrarily complex environments but also learn a conditioned reflex (like Pavlov's dog). Marvin knew Grey Walter, and Grey Walter's ideas helped inspire the later Logo turtle geometry system for children.

Marvin and Seymour could see that most interesting systems were *crossconnected* in ways that allowed parts to be *interdependent* on each other—not hierarchical—and that the parts of the systems needed to be *processes* rather than just "things."

At the same time that he wrote this essay, Marvin was writing a book about how minds might work—called *The Society of Mind*—and said in the beginning: "To explain the mind, we have to show how minds are built of mindless stuff, from parts that are much smaller and simpler than anything we'd consider smart. … But what could these simpler particles be, the 'agents' that compose our minds?"[1]

We've just made several Tinkertoys that are *more able* than their parts. Can we use Tinkertoys to make *agents* that are *smarter* than the agents that are their parts?

In order to make our self-powered vehicle smarter, we have to make it able to sense what is going on far enough ahead of time to make something happen that will cause the vehicle to stop before any disaster occurs. One way to do this is to give it a "sensor probe" that will make contact with barriers in front of the vehicle and connect the probe to a brake that will activate when the probe touches something. If we extend the probe out the back, then we'll also have a "prod sensor" that will release the brake when our vehicle is bumped from behind and prompt it to start up and move out of the way.

Children are very happy when they have a big idea, build it, and have just enough Tinkertoys parts to make it! But then more parts are still needed because as ideas get built they also cause new and better ideas to appear.

One of the many wonderful things about computers—especially today—is there are always enough parts for any idea, and those ideas definitely cause more wonderful ideas to appear. There is essentially no friction, and no limits these days except for our imaginations.

What the children are doing with toys like Tinkertoys are *intellectually honest versions of what adults do*—in this case, the designing and building parts of engineering and art. The design process has not been removed from them by giving them prefabbed special parts and pre-done designs by others (as many contemporary "construction toys" do today). Too much pre-purposing puts success, no matter how empty, ahead of the joyous struggles of learning and doing. Real learning of powerful ideas and the power that comes from them are the results of the big changes that happen *in our minds* as we struggle with, eventually grasp, and become fluent with the difficult-to-learn ideas. Many "learning difficulties" are critically important! Seymour Papert called this "Hard fun."[2]

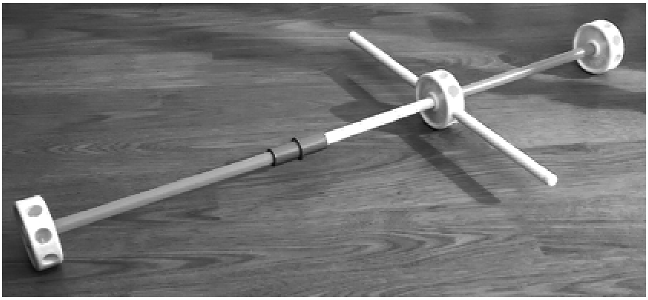


Figure 1.6a

Figure 1.6b

Inspired by Grey Walter's robot tortoises, a brilliant idea for the Logo world was to make a physical turtle for children to program. Its "mind" would be supplied by programs they would write! This brings what Grey Walter so cleverly did with a few wires and vacuum tubes into their world in a real and deep way.

The combination of a physical turtle with a children's programming language brought out the best of both worlds, and many interesting projects were invented, including making the turtle more able by telling it how to draw and do things—and having the children learn real mathematics in the bargain.[3] The project to make the turtle "smarter" by having children build little brains for it to think and learn with let the children gain important insights into biology,

Figure 1.7

psychology, themselves, and *thinking itself* (as Marvin and Patrick Winston discuss insightfully in essay 6 and its introductory remarks).

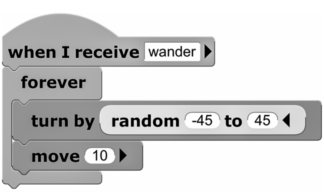
For example, the following figure shows a simple Logo program—an "agent made from mindless actions"—for getting the turtle to carry out the Grey Walter "explore" behavior. Over and over, the turtle will go forward "a little" and then randomly turn somewhere between 45 degrees to the left and 45 degrees to the right.

```
to wander
 forever
   forward 10
   right (random 90) – 45
```

"**to**" means define a procedure, with the following name
"**forever**" means do what is indented below over and over
Move turtle **forward** 10 little steps
Turn turtle to the **right** the number of degrees given by doing the arithmetic

Figure 1.8

I'll now turn to a descendent of Logo—a drag-and-drop blocks programming language called GP for *general purpose*; it is easily accessed online through any web browser.[4] GP has several useful and more modern features in addition to what Logo provided. Figure 1.9 shows the same Grey Walter "wander" program done in GP.[5]



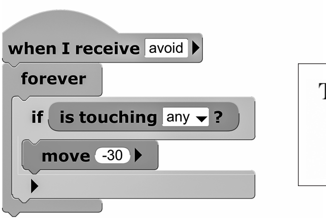
Run the process below when "wander" is broadcast

"**forever**" means to do what is indented below over and over
Turn turtle randomly between left 45 and right 45 degrees

Move turtle **forward** 10 little steps

Figure 1.9

This program is more "able" than "smart" because it can be trapped by walls and other obstacles. Let's add in an "avoid"



Test to see if we are touching something
If yes, jump turtle backwards 30 little steps
If no, just do the forever again

Figure 1.10

behavior to run simultaneously. It will use the touch sensor to see if we've bumped into something, and if so, we'll want to jump back.

Note that there is always a random turn whether going forward or backward, and this helps the turtle eventually get away from an obstacle (figure 1.11). Does the track of the turtle remind you of the much later "Roomba"? Two simple agents can make a lot happen!

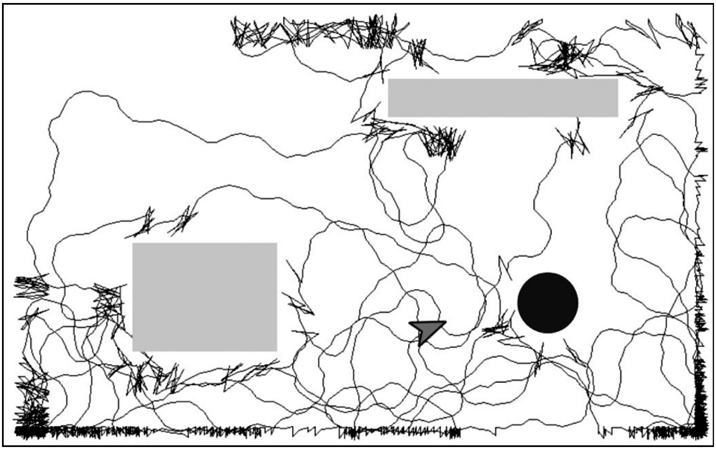


Figure 1.11

Tinkertoy Is the Ultimate Thinkertoy

The “TicTacToe”-playing computers done by Danny Hillis and Brian Silverman, then two of Marvin’s students and now well-known computer scientists, almost show the limits of what can be made with Tinkertoys. These computers included many ideas that the inventors of Tinkertoys never thought about!

Tinkertoys’ limits here are not logical—it is logically possible to build a whole computer out of them—but practical. Tinkertoys have various kinds of errors, frictions, and other difficulties that add up to a frozen immobility after a point. The second, simpler Tinkertoys computer was created to minimize as many practical difficulties as possible.

We can see why Marvin used Tinkertoys as an analogy in essay 1 for the nature and powers of children to “make things come to be, where nothing ever was before” via designing and programming. We make not just to have, but to *learn*—first with our hands and eyes that can feel and see, and then, by stepping back to think more deeply, we move from “tinkering” to “pondering.” Adding the computer allows us to turn our handmade things and ponderings into dynamic behaviors to give us qualitatively more to think about. Not just Tinkertoys—computers are “Thinkertoys”!

In designing and programming on computers, we are following in a long tradition that Marvin had a part in, one that started with Sketchpad, the invention of modern interactive graphical computing by Ivan Sutherland in 1962. Sutherland’s advisors were Marvin and Claude Shannon.

We can see today’s personal computing here! Just as the rough sketch of a bridge was easily transformed by Sketchpad fifty-five years ago into a dynamic simulation that calculated the stresses and strains as well as showed the appearance of the bridge, today we can have children draw diagrams of animal neural structures, make them work, and try them out in simulated animals. Sketchpad was done on a supercomputer the size of a football field. Today's children have vastly more computer power at their disposal on their personal laptops that cost only a few hundred dollars and are the size of just the Sketchpad display (the football field–sized computer is now on the back of the display, and much faster!).

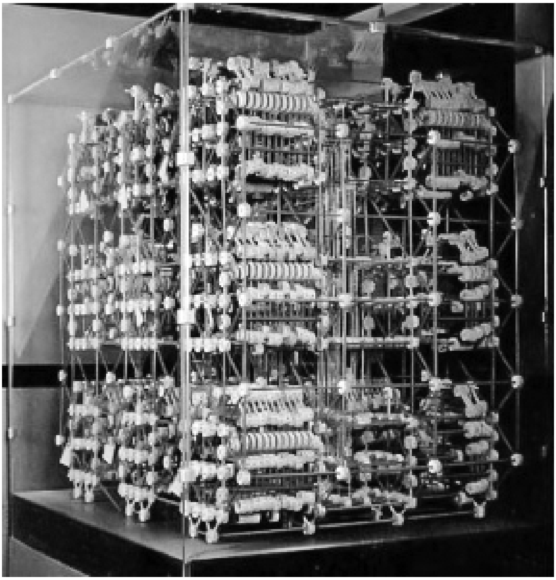


Figure 1.12
Brian Silverman and Danny Hillis, *TinkerToy TicTacToe Computer*, 1975. Image courtesy of Mid America Science Museum.

Figure 1.13a
Ivan Sutherland, *Ivan Sutherland Operating the Sketchpad System*, 1963 [digital image]. http://www.dspace.cam.ac.uk/handle/1810/243359. Courtesy of Creative Commons 2.0 License.

Marvin had a part in how laptops and children's computers came about. When I was in grad school at the University of Utah in 1968, Marvin gave an exciting talk at a nearby educational computer conference, and much of it was about Seymour and his thoughts on helping children learn powerful ideas.[6] Later that year I visited Seymour's and Cynthia Solomon's Logo classroom outside of Boston, and what they were doing blew my mind—and still does to this day! On the plane back to Utah, I made this sketch of "a personal computer for children of all ages": a "dynamic medium for creative

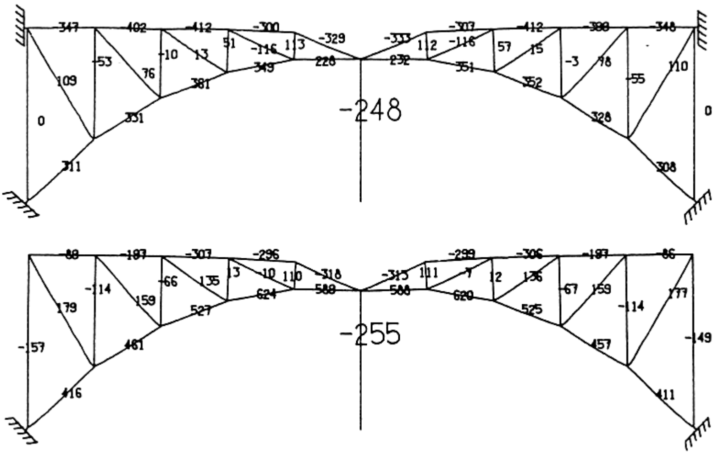


Figure 1.13b
Ivan Sutherland, *Diagram of Bridge Loading Forces Created Using Sutherland's Sketchpad*, 1963 [digital image]. http://www.dspace.cam.ac.uk/handle/1810/243359. Courtesy of Creative Commons 2.0 License.

thought" inspired by Sketchpad, Logo, and other ideas from our research community. This was part of what grew into the personal computers, laptops, and tablets we now use.

The children of 2018, learning about how conditioned reflexes work by programming working models that bring their sketches to life, are using ideas and technologies directly descended from Ivan, Marvin, Seymour, Cynthia, and the other pioneers of that time. The examples in this afterword show that what's really important about computers, especially for children, is how they can go beyond

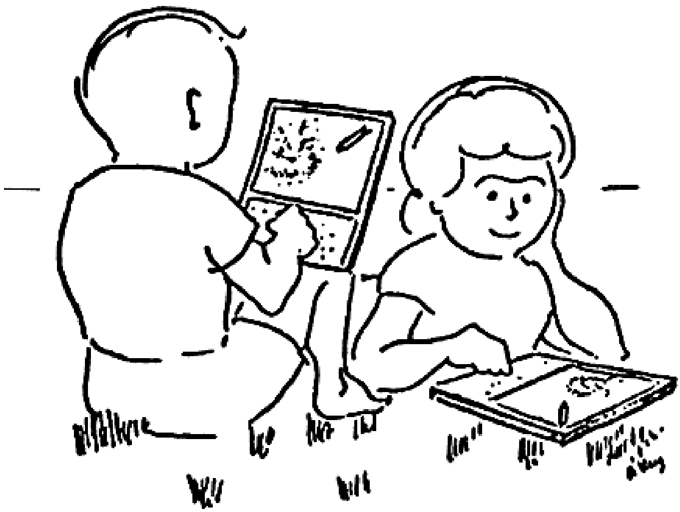


Figure 1.14

imitating conventional media to give rise to new and dynamic ways to represent ideas, understand and learn complex systems, model scientific theories, and create new art and artifacts that cannot exist without them.

What do *you* think children should be doing with their computers?

Acknowledgments

I am deeply indebted to John Maloney for extensive help in shaping the computer examples in GP and for very helpful critiques of the writing. I'd like to thank Yoshiki Ohshima for valuable insights. Thanks also go to Mike Travers, who pioneered many of the ideas for modeling animal minds for children. And I'd also like to thank K-5 instructional technology teacher Jen Lavalle for a careful critique that increased clarity and understandability. Most of all I want to thank the many dedicated teachers we've worked with over the years—we've learned a lot from you!

Finally, thanks to Cynthia Solomon and Xiao Xiao for inviting me to write this afterword to one of my favorites of Marvin's writings. And it's been great to join my friends and colleagues of many years (many decades!) to celebrate one of our most delightful and influential mentors. It's been a lot of fun putting this together!