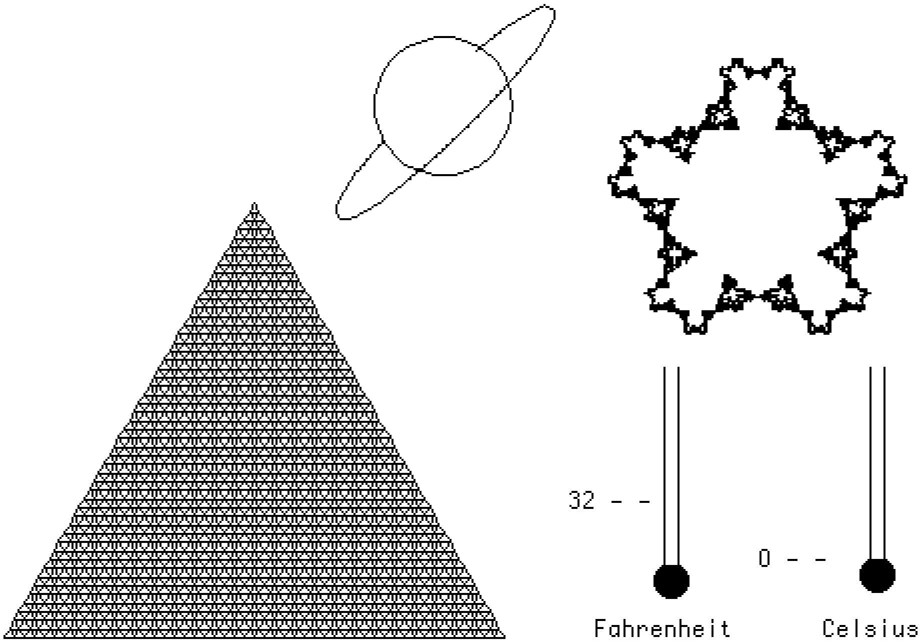
www.logofoundation.org

From Polygons to Functions to Orbits to Fractals: A Year of Logo Work With a Mathematically Gifted Student

by

Eadie Adamson



Acknowledgments
Thanks to Michael Tempel and Becque Olson for their comments, advice, and final editing. Special thanks to my fellow collaborator, S. K. ("Seb"). This paper would never have been written had it not been for the inspiration and excitement about learning that we shared.

Abstract

If it is true, as Sylvia Weir said, that learning-disabled children show us "a magnified, slowed-down view" of learning (Weir, Sylvia *Cultivating Minds*, Harper & Row, NY 1987), then the experience of working with an exceptionally gifted child shows us learning that is decidedly in the "fast forward" mode. A mathematically gifted fourth grade student was permitted to spend the year working with LogoWriter as a substitute for fourth grade mathematics class. For the teacher, this work was a challenging and exciting collaboration in which the changing role of teacher came strongly to the fore. For the student, it was a year in which intellectual challenge and pursuit of knowledge turned in new directions. The activities pursued in this one-on-one year of study give us a glimpse into the challenges and benefits of one very individualized collaborative approach to learning.

Background

"Seb" was just nine years old when I first met him, a year younger than the rest of his fourth grade class. Seb is one of those children who is capable of conversing with an adult on an adult level. Yet when I observed him playing in the classroom, the sophistication disappeared and he was a typical nine-year-old, comfortable with his peers, full of mischief, excitement, and delight in the world around him.

Seb had already been identified as an extraordinarily gifted child. He possessed strong talents in every academic area, but the particular strength which set him apart from his classmates was in mathematics. His parents, aware of his gifts and cognizant of the fact that the home atmosphere can be an important influence on a child's intellectual development, provided a great deal of intellectual stimulus in their home and maintained a keen interest in Seb's progress.

I was asked to provide Seb with an intellectually stimulating year of work with Logo as a substitution for math class. There were no guidelines as far as topics to be covered. In fact, I was told that it did not matter what we did (!) as long as he was kept busy, interested, and challenged. I thought that the content should be directed towards mathematics and that the obvious first topic should be Logo geometry.

We met daily during his class's math time (about an hour every morning). Seb also had a Logo class that met weekly with an additional class period on alternate weeks. As Seb began working on projects with me, he often used the class time to continue pursuing his ideas, generously sharing what he learned with interested classmates.

The Content

Geometry

When we began, I estimated Seb would spend a few sessions working out regular polygons. To my astonishment, during that first session he progressed from deriving the angles for polygons *and* writing procedures with variables to writing a generalized polygon procedure. Seb's progress was like that of a very adept adult learner. It was clear this was going to be an exciting experience for us both!

The default setting for numbers for Macintosh LogoWriter is four decimal places, but this can be expanded to 16. While working with the polygons, Seb rather playfully repeated a septagon multiple times. To our surprise, the figure began to change. We realized that the calculation of the angle had to be off by a very small number. This led us to thinking about the importance of accuracy and about situations in which such a small difference could be significant. Suddenly we were discussing chaos theory. I suggested that he look at the first chapter of James Gleick's book, *Chaos* (Gleick, James *Chaos*, Viking Penguin, NY 1987) (Seb's reading level was quite high), which discussed a similar problem with recording weather data, and that we talk about the ideas after he read it. Meanwhile we wanted to see what would happen if the septagon continued. How much off course would it go? and what would it look like? We set up an experiment on the computer and let it run all day. Seb took on the task of determining how many times it would be necessary to repeat the procedure so that the experiment could run over an entire weekend.

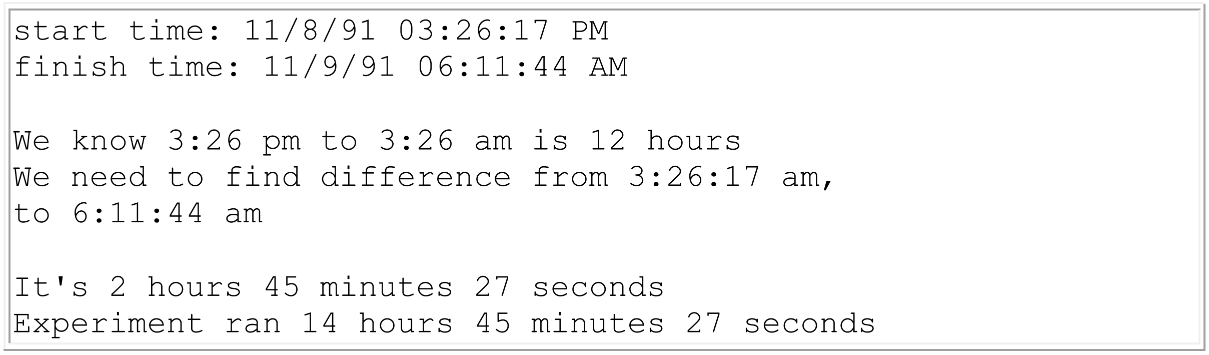
He made one essential error (which later occasioned a discussion of scientific method): he hid the turtle which had been showing during the first test and also changed the size of the septagon, making it smaller. Changing these two variables shortcut the time radically. When the turtle is hidden, Logo no longer needs to redraw it at every move. While the time this takes is not obvious in normal use, in a "Logo Overnight" style of experiment, the fractions of seconds cut down the time considerably. Changing the size of the septagon had a similar effect on the length of time the experiment would run.

```
to test
ht
print se
   [start time:] clock
print heading
print pos
repeat 11 [test1]
print se
   [finish time:] clock
print heading
print pos
end
```

```
to test1
repeat 99999 [sept 35]
end


to sept :size
repeat 7
   [fd:size rt 51.4286]
end
```

Because Seb had changed two variables, the program that he expected would run all weekend stopped by the next morning. When he returned and found the start and finish times displayed, he was a little dismayed. But I found another math problem for him. How long did it take? Here, in his own words, are his calculations for the length of time the experiment ran (this only took a few minutes for him to do):



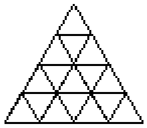
```
start time: 11/8/91 03:26:17 PM
finish time: 11/9/91 06:11:44 AM

We know 3:26 pm to 3:26 am is 12 hours
We need to find difference from 3:26:17 am,
to 6:11:44 am

It's 2 hours 45 minutes 27 seconds
Experiment ran 14 hours 45 minutes 27 seconds
```

We returned to a discussion of small errors. We knew we could change the number of decimal places in LogoWriter. When Seb changed the decimal places to 16 (**setnf 16**) and ran the instructions **show 360/7** he got a more accurate number: 51.428571428571429. He adjusted his procedure to reflect the new information. Then he started a repeated septagon again. This time, even with 5000 repetitions, there was no apparent error. Although we should have followed with a test running over a weekend to be sure that there would be no aberration in the septagon with the more accurate angle, Seb was anxious to move on to other explorations.

Seb was a strong visual thinker, able to break something into pieces and to visualize the parts and how they combined. I asked Seb to devise a way to build a triangular stack of regular triangles. Seb's response was rapid and quite interesting. Rather than looking at the shape as a series of small triangles building up from a base, he saw the problem from another perspective, working from the outside in. Seb made a preliminary drawing. He drew a large triangle first. Then he saw that there was a smaller triangle within this, but upside down. He drew that next. After making a sketch of what it would look like, he quickly constructed a single procedure to build his stack. Notice in the copy of the procedures that he also used multiplication and division of the size, keeping the number constant. Although he had appropriated variables he had not yet grasped that the **tri.stack** procedure could also be written with variables. He had, however, written the procedure in such a way that it was an easy next step.



```
to tri :size
repeat 3 [fd :size rt 120]
end

to tri.stack
rt 30
tri 34 * 2
fd 34
rt 60
tri 34
fd 34 / 2
rt 60
tri 34 / 2
tri 34
rt 180
tri 34 / 2
fd -34
tri 34 / 2
pu lt 60
fd 34
repeat 2 [fd 34 / 2 rt 120]
pd
lt 60
tri 34 / 2
rt 120
fd 34 / 2
rt 150
end
```

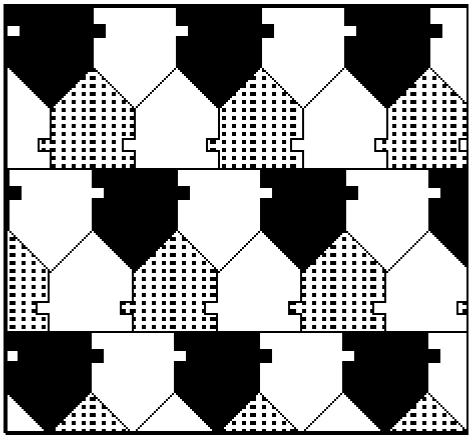
"Given a base number of triangles, if you build a triangular stack of triangles of any size, can you predict how many triangles will be formed at the base?" I asked. Looking at the drawing he had made, Seb responded quickly, "Of course! It's 2n - 1." He was already using algebraic terms. This may have been because he followed the math work his older sister was doing. Seb had an excellent memory for almost anything he heard **or** saw.

Tessellations

When we progressed to work with tessellations, Seb showed his understanding of the use of variables in an interesting way. First he drew a sketch of a figure which he planned to use to fill the plane. Then, using a single variable, he developed a procedure to draw the shape. I watched astonished as he programmed the procedure without moving back and forth from the editor to the screen. He moved and turned the turtle in his head, keeping track of its position through many twists and turns, seeming to have no need to try it out first! Perhaps it was because of his highly developed visual sense that working with the Logo turtle was particularly compelling for him.

Below is the procedure he developed for the "motif," as we called it. Alongside it is a fragment of the completed design, which is created from a single variable for size (notice how he used division and multiplication of the variable within the procedure):

```
to motif :size
lt 45
fd :size
rt 45
fd :size / 2
rt 90
fd :size / 5
lt 90
fd :size / 5
lt 90
fd :size / 5
rt 90
fd (:size / 2) - (:size / 5)
rt 90
fd :size * 1.4
rt 90
fd (:size / 2) - (:size / 5)
lt 90
fd :size / 5
rt 90
fd :size / 5
rt 90
fd :size / 5
lt 90
fd :size / 2
rt 45
fd :size
rt 135
end
```



When Seb built his tessellation, he constructed a "spaghetti code" procedure; a long unformatted string of commands filling several pages. Before we met again, I took the basic motif and constructed the beginnings of a working example. I broke up the process into subprocedures, such as **moveover**, **row**, **moveup**, **movedown**. I added a superprocedure which used these to make the beginnings of the design. When Seb arrived, we looked at his procedures. Then I showed him the example I had created for him, showing him how he could take repeated sets of commands and put them into smaller procedures of their own. Seb absorbed that, agreed with the premise that it made reading the superprocedure as well as the debugging process easier, then calmly observed, "But you made the design upside down and backwards."

Nonetheless, that day he made the adjustments to his own work.
Later, when the basic design was completed, Seb decided to fill the shapes with solid color or pattern. Again, I needed to press the idea of subprocedures. I left him a note:

As a general rule in programming, whenever you find yourself doing the same thing more that twice, it is wise to create a small procedure, using variables if necessary, to replace a sequence of instructions.

Looking at your procedure, notice that you repeatedly

picked up the pen
moved to a position
put the pen down
used either the command **fill** or the command **shade**

Here are two procedures to do this. Each has a position for input. This replaces all the commands for that step, save for entering the position. The sample below is a corrected version of tessellation. I simply deleted up to the list of points (the position) and substituted either **fillit** or **shadeit**. Notice how much shorter the procedure is. Actually it is also easier to understand what is going on.

Make these changes to your procedure. Be careful! Delete up to the position, but look below it first to see whether to insert **shadeit** or **fillit** (if the command below the setpos is **fill**, use **fillit**, etc...)

```
   to fillit :pos      to shadeit :pos
   pu                  pu
   setpos :pos         setpos :pos
   pd                  pd
   fill                shade
   end                 end
```

Planning

I printed out a copy of my revisions and left them for Seb, since it was a day when I had another class during half of our hour together. About twenty-five minutes later Seb had made his corrections.

I found work with Seb continually surprising. Problems that could have taken a long time to solve were explored and solved with dispatch. This affected my planning. If he came up with a quick solution I needed to be prepared to discuss new ideas. I began to keep a little list of possible new directions that we could choose from when the latest exploration had reached its conclusion.

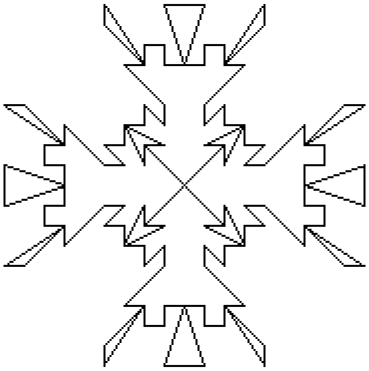
What was interesting about this list-keeping was that, rather than having a set program, I was looking at the work with Seb as a collaboration. I would think of things that might interest him, problems that interested me, or sometimes areas that neither of us knew much about. The feeling as we worked and planned together was much more of an intellectual journey than it was of "mathematics class" or "lesson time."

Cartesian Coordinates

Sometimes, however, Seb's intellectual curiosity led him to pursue a project in greater depth and over a longer time than I expected. This happened first when we worked with Cartesian coordinates.

There was a day when, due to construction, there were no computers free during our session. Rather than give up the time, it seemed a good opportunity for off-computer work. Seb had learned already about positions on the screen. He had never explored drawing with coordinates. Armed with pencils and graph paper, we found a quiet place to work. After explaining how the coordinate system worked, I suggested to Seb that he make a symmetrical design on the paper and then figure out the points to which the turtle should move.

Seb created a beautiful symmetrical design (see below) and then on paper began writing out the positions for each move of the turtle. When he got back to the computer, Seb put his data into a procedure to draw his design.



I gave Seb a tool procedure (below) to help him with this task, fully expecting that he would tire of the process before the design was completed. To my surprise, he worked steadily on it for two sessions. His intense interest in solving this problem was clear.

```
to plot :list
if empty? :list [STOP]
setpos first :list
plot butfirst :list
end
```

Here is Seb's procedure for his design:

```
to design
plot [[-20 20][-10 20][-30 30][-20 30][-20 40][-10 30]
 [-10 40][-30 60][-20 60][-40 80][-40 90][-20 60][-20 70]
 [-10 70][-10 60][0 60][-10 90][10 90][0 60][10 60][10 70]
 [20 70][20 60][40 90][40 80][20 60][30 60][10 40][10 30]
 [20 40][20 30][30 30][10 20][20 20][0 0][20 20][20 10]
 [30 30][30 20][40 20][30 10][40 10][60 30][60 20][80 40]
 [90 40][60 20][70 20][70 10][60 10][60 0][90 10][90 -10]
 [60 0][60 -10][70 -10][70 -20][60 -20][90 -40][80 -40]
 [60 -20][60 -30][40 -10][30 -10][40 -20][30 -20][30 -30]
 [20 -10][20 -20][0 0][20 -20][10 -20][30 -30][20 -30]
 [20 -40][10 -30][10 -40][30 -60][20 -60][40 -80][40 -90]
 [20 -60][20 -70][10 -70][10 -60][0 -60][10 -90][-10 -90]
 [0 -60][-10 -60][-10 -70][-20 -70][-20 -60][-40 -90]
 [-40 -80][-20 -60][-30 -60][-10 -40][-10 -30][-20 -40]
 [-20 -30][-30 -30][-10 -20][-20 -20][0 0][-20 -20][-20 -10]
 [-30 -30][-30 -20][-40 -20][-30 -10][-40 -10][-60 -30]
 [-60 -20][-80 -40][-90 -40][-60 -20][-70 -20][-70 -10]
 [-60 -10][-60 0][-90 -10][-90 10][-60 0][-60 10][-70 10]
 [-70 20][-60 20][-90 40][-80 40][-60 20][-60 30][-40 10]
 [-30 10][-40 20][-30 20][-30 30][-20 10][-20 20][0 0]]
end
```

Obviously, in this association, there was to be almost no limit to the lengths to which we could go in a short period of time. Seb was like a sponge, eager to soak up every new idea.

Functions

Since our work with Logo was to take the place of math class, it seemed a good idea to incorporate still more mathematical inquiries. During the times each week when I was busy with another class during part of our working time, I usually left Seb messages on the desk or on his program about what to do next.

At one point, I had just spent a few days in Montréal and had to deal with weather reports with temperatures reported in Centigrade rather than Fahrenheit. On the day I left, the airport temperature was -18deg.. That sounded really cold to me! I had forgotten the precise method to calculate the difference. My puzzlement about temperatures made a nice connection with the work Seb and I were doing using the appendix of Phil Lewis's book, *Approaching Precalculus Mathematics Discretely*.* The appendix has a wonderful section on functions in Logo set up so that there was a nice sequence of problems to be worked out. There were plenty of interesting tasks that Seb could work on by himself.

At the time I returned from my Montréal trip, Seb was already writing conversions such as feet to inches, inches to centimeters, and so on. I could now personalize at least one problem for Seb to solve. I told him about my trip and left him this assignment:

Seb:

Today try writing functions to change:

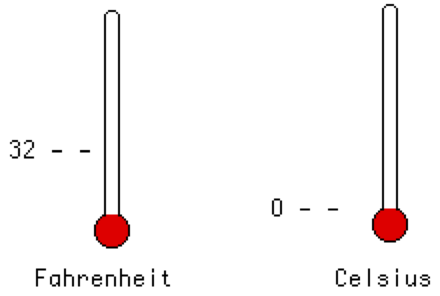
Fahrenheit to Celsius
Celsius to Fahrenheit

Test them out with a variety of cases.

Then think about a thermometer project which will display the information.

Seb wrote conversion procedures making use of his previous work with functions from the Lewis appendix. (Lewis, Phil, *Approaching Precalculus Mathematics Discretely,* MIT Press, Cambridge, MA 1991)

Seb was delighted to work with graphics too. Connecting the conversions to a visual display really interested him. He set to work eagerly. Here's what his thermometer project looked like:

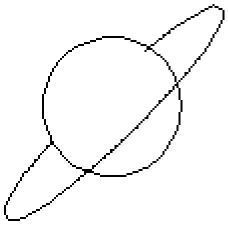


Seb always had a different picture in mind from what I anticipated. Notice that he didn't make 32deg. and 0deg. show at the same level. Interestingly, his program didn't take into account temperatures that would "burst" the thermometers. This was a source of much delight for Seb: he loved to show his project and then give a temperature that would make the red line push past the top. He wasn't at all concerned with the imperfection this exposed. He saw it as a source of fun instead.

Orbits

Towards the end of the year Seb's science class was studying the planets. Seb used the LogoWriter *HyperMedia Tools* to create his report on the solar system. He decided to illustrate his introductory page with an animation of a meteor orbiting a satellite.

Again I observed this phenomenon of persistence beyond the point at which another student would give up. Seb worked on solving the problem of creating a perfect orbit, an ellipse, for well over a week. Several times he nearly gave up, then found another partial solution and continued confronting the problem. As I had already observed, Seb was a very independent learner. He was not interested in being given solutions. Occasional "nudges" over rough spots could sometimes be slipped in, but in general, each problem became his property to solve. The orbit problem led us to discuss kinds of curvatures, headings, positions, and methods to determine that the meteor was following the same track with each "rotation." Here is his solution to the problem:

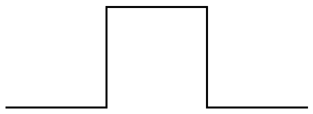


```
to orbit
pu
setpos [178 68]
pd
seth 50
st
repeat 15 [fd 1 rt .125]
fd 2
repeat 10 [fd 2 rt 1.5]
fd 4 rt 28.25
repeat 3 [fd 2 rt 27]
fd 4 rt 28.25
repeat 10 [fd 2 rt 1.5 ]
repeat 35 [fd 1 rt .125]
repeat 65 [fd 1 rt .125]
fd 2
repeat 10 [fd 2 rt 1.5]
fd 4 rt 28.25
repeat 3 [fd 2 rt 27]
fd 4 rt 28.25
repeat 10 [fd 2 rt 1.5 ]
repeat 20 [fd 1 rt .125]
pu
ht
repeat 65 [fd 1 rt .125]
st pd
orbit
end
```

Fractals

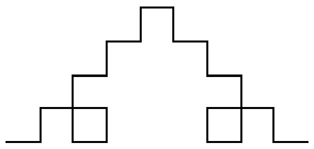
Our final topic was fractals. I began by drawing a single example of a "square" fractal, in which the center third of a line is replaced by a square "bump." I asked Seb how he would program it. He quickly wrote a procedure with a variable to draw the form:

```
to root :size
fd :size
lt 90
fd :size
rt 90
fd :size
rt 90
fd :size
lt 90
fd :size
end
```



Then I explained how the next "level" of a fractal would work, drawing for him a half-size version on each side of the original. "I get it," he exclaimed, and programmed a procedure which used the first, cut by three to build a second level.

```
to root2 :size
root :size / 3
lt 90
root :size / 3
rt 90
root :size / 3
rt 90
root :size / 3
lt 90
root :size / 3
end
```



Seb didn't stop there, but went on to program 20 levels in this fashion. **Root3** used **root2** as its measure; **root4** used **root3** as its measure. He continued!!! Once I tried to interrupt, thinking it was time to show him how to write a recursive procedure to do this all. He would have none of it. His excitement built as he began to generate more and more levels. Eventually I asked, "*Now* would you like to see another way to do this?" My point here is that with any learner there is an appropriate time to interject and an inappropriate one. Sometimes it is difficult to find that moment. For the average learner my help would probably have been accepted the first time I tried to intervene. However, Seb was deeply involved in an investigation of his own which he wanted to push to its limits. A very independent learner, he was truly excited about the process he was building.

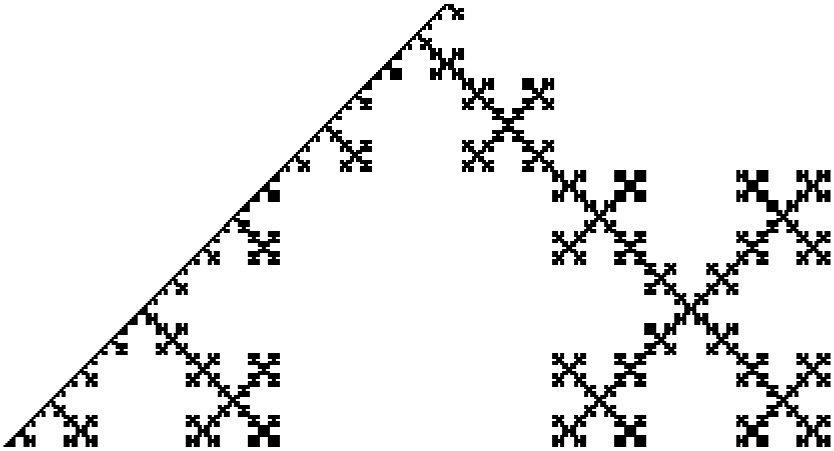
Together it seemed we had found an interesting way to help beginning programmers start working with fractals, without trying to explain levels of recursion or simply taking a fractal procedure "as is" and using it. I tried this approach with several other interested students, all with very positive results.

We did another "Logo Overnight"-style experiment. Seb set Logo into "window" mode, which allows the turtle to keep moving (in our imagination, since it goes out of sight) without wrapping. He changed his **root15** procedure to record the start and finish times (**print clock** prints the date and time on the screen at the start and finish of the procedure; **print pos** prints the position of the turtle at the finish):

```
to root15 :size
print clock
root14 :size / 3
lt 90
root14 :size / 3
rt 90
root14 :size / 3
rt 90
root14 :size / 3
lt 90
root14 :size / 3
print clock
print pos
end
```

We neglected to write down what input he used for the size, but from the result, and knowing Seb's fondness for large numbers, I would guess it was at least 1000. Here's what we found the next day:

```
4/27/92 09:58:39 AM
4/28/92 09:45:21 AM
Position: [1109.0097 1129.2179]
```



What we should have pursued - but didn't - was: where exactly is the turtle in relation to the computer? Would it be just feet away or rooms away? An interesting problem.

Once we had arrived at the point of using the classic fractal procedure, we turned to analysis of form. How could this procedure be altered so that the "bump" was a triangle? a pentagon? a hexagon? We looked at the geometry involved in the turns and came up with a general rule for the angle to turn: 180deg. - the angle of the shape to be inserted always gave us the amount of the starting and ending turn to insert a particular polygon into the center of the fractal. Seb worked out all these variations as well. He also explored putting several together to form different patterns. Seb discovered that turning left between forms rather than turning right would produce dramatically different shapes.

He had begun working with designs based on these fractals just as the year ended.

Who Deserves a Chance Like This?

When this project was just beginning, I was telling Seymour Papert about it, and about my misgivings over working with a single student when there were at least four other students who would have benefited from and enjoyed the opportunity to explore mathematical ideas with Logo. I had brought up the subject with the administrators at the school, who replied, "But there's no one else who could possibly miss math class!" Seymour's response to this is worth thinking about: "*What* would they miss?"

The Best Learning *is* Play

The progress of a very bright young student working with Logo resembles an adept adult in the speed of learning. What distinguishes the child learner is the playful inquiry into ideas. Sadly, adults have learned to leave this spirit of play behind. Teachers need to allow students time to be playful as they explore new ideas. The playful aspects of Logo are doorways opening into new knowledge. Seb's journey during the year would not have been so astonishing or so interesting without this aspect of play which often led our investigations in new directions. It also required tolerance from the teacher and a willingness to see (and sometimes participate in) this play as a necessary part of the learning process.

Do any of us play enough anymore?