Profiles (continued)

different speeds. It had to introduce the full range of required course concepts gradually and provide thoroughly for long-term repeated review and practice. I found excellent in these respects the so-called Incremental Approach texts discussed by Klingele and Reed, Phi Delta Kappan, June '84.

In practice, I now distribute homework assignments in writing a week in advance. This not only saves time but also takes care of class cancellations due to Chorus rehearsals, snow days, and the like. We set aside a period a week to do corrections in class. I assign seating so that potential student instructors are well distributed around our one large conference table.

Problems done with incorrect outcomes (wrong answers) are treated as procedures that have bugs, and *debugging* is what we do during the homework correction periods. Just to slip more into the metaphor, "every mistake you make, you get smarter in the debugging, particularly if you discuss it with your classmates. Don't worry about mistakes, welcome them. Right answers the first time mean you already know your stuff, so you've wasted your time. Mistakes are the valuable part. Please don't ask me to help you with you're ratio problem when Alice and Willie here are experts. You don't need me to teach you, you can learn by yourselves.

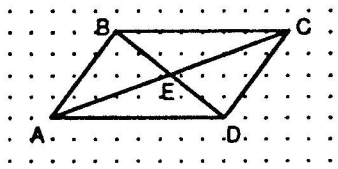
These sessions can get noisy, something is given up in order and control. Of course we miss the goals by a wide margin. Of course I help out a lot. Of course kids don't get fun out of mistakes, but their attitudes towards them *have* changed, and they are getting used to talking math with each other. "Do you like this course?" I ask them. Some grumbling..."It's different", is an answer I like. What it is, of course, is Logo, and they havn't touched a computer. But now they're learning the course material at an adequate pace, and we have two or three days a week to supplement and enhance. Computer Lab, here we come.

◊

Problem Solving

Help wanted!

Some of us are working on a "traditional" geometric microworld. A typical problem: "Given the positions of the vertices of parallelogram ABCD, where do the diagonals intersect?" We have a recursive procedure to find and output the position of the intersection of any two nonparallel lines, *but its execution is disappointingly slow.*



The inputs to our intersection procedure are "AC and "BD, two points on each line, and an initial :STEP. Turtle starts at the endpoint of one input, say A, and takes STEPs towards C, comparing at each step the sum of its distances to B and D *before* the step with the same sum *after* the step. If distance *after* is less than distance *before*, recurse with same step, else step back, and recurse with a reduced step (STEP/10). The decreasing STEP size provides the stop condition. IF :STEP < .00001, the positon outputs and execution stops.

If we hurry things up and allow stopping with a step of, say, .01, subsequent returns from procedures that output AREA don't look convincingly "equal" to students. Two areas that are supposed to be equal might return numbers like 2345.67 and 2329.87. With this degree of imprecision we lose the "aha!" impact. The intersection procedure needs very small steps to produce subsequent area measurements comparable to 4 significant figures, but our method just takes too long.

Write with your suggestions to the Editor.

R.W.◊

Getting a Feel for π

ORBIT

by Ihor Charischak

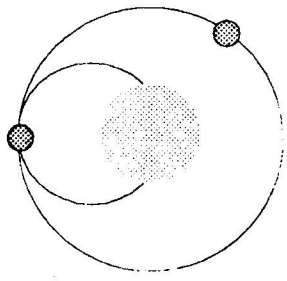
This Activity was done in Logowriter, and takes advantage of the four turtles which can simulate concurrent motion. The procedures can be modified for any version of Logo.

Goals: Among the goals of this activity is student knowledge that a circle's circumference is a little more than 6 times the radius of the circle.

Tool procedures: EARTH and ORBIT

```
TO EARTH :X :Y
MAKE "EARTHPOS SE :X :Y
PU SETPOS SE :X :Y
SETSHAPE 12 SETCOLOR 4 PD STAMP
SETSHAPE 0
END

TO ORBIT :TURTLE :DISTANCE :CIRCUM
TELL :TURTLE
SETSHAPE 0 SETCOLOR 2 ST
PU SETPOS :EARTHPOS
FD :DISTANCE
RT 90
CIRCLE :CIRCUM
END
```



EARTH positions a stamped earth shape at location :X:Y . ORBIT requires three inputs:

1) :TURTLE - - one of four turtles (choices 0 to 3)

2) :DISTANCE - - how far out in orbit (in turtlesteps) you wish to send your turtle.

3) :CIRCUM - - the length of the orbit in turtlesteps.

Continued next page...