Multiplidungeon

RICK NORWOOD

One of my jobs, as co-ordinator of math basic skills at William Paterson College, is to explore ways in which computers can be used to teach basic skills. At the same time, I feel that I should also explore ways in which computers should not be used to teach basic skills. Let me tell you about an experience of my own

A number of years ago, my daughter, Erin, was having trouble learning her multiplication table. The real trouble was that she did not want to learn her multiplication table. So, I thought, I'll make a game out of it I'll make a dungeon on the computer, but to open the doors or get the treasure, she will have to solve a multiplication problem. I'll call it *Multiplidungeon*

So, I disappeared down the stairs to the basement room where our Apple lived and my wife and kids didn't see me for a long time.

After about three days, I emerged into the light, bearded and bathless, and I presented Erin with her new computer game She booted the disk, and this is what she saw:

> You are in a large cavern. There are exits to the north, south, east, and west There is a sword in a stone here To take the sword, multiply 7×3.

Now, there are several things about this program that I was rather happy with. One was the little logic problems which I had built into the dungeon to make it more interesting. Another was the fact that the part of the program which generated the multiplication problems (using a random number generator) was completely detached from the main program, so instead of multiplication any short answer type of question could be put in, without altering any other part of the program. Then there was the way the rooms were built Each block of one hundred numbers refers to a room, so that, for example, room 17 is in the lines of program numbered 1700 to 1799. This means that you can go in and redesign one of the rooms without changing anything else in the program.

Multiplidungeon was by far the longest program I had written at that time. It was one of the first programs I wrote in Basic. And I learned a lot by writing it. Erin, on the other hand, did not learn her multiplication table from it. In no time at all, she figured out that since the multiplication problems were generated randomly, she only need to try a few times to get a problem like 1×5, which she knew the answer to. She could safely ignore the hard problems.

I thought about going back to build in some sort of penalty for a wrong answer, but before I got a chance to do so, Erin's mother bought a set of flash cards and taught Erin multiplication in less time than it had taken me to write the program.

This is one problem with using a computer to teach basic skills. The computer all too often does something which a cheaper tool could do as well or better. There is no point in using a computer as a clumsy book or an expensive mimeograph

There is, however, a more serious problem with using the computer to teach basic skills. Most of the programs I have seen which teach math skills emphasize rote learning. But the students I see do not have a problem with rote learning. Their problem is a lack of understanding, and an inability to explain what they are doing and why they are doing it, even when they do it correctly.

Often a student will say "three times seven" when adding three and seven to get ten. They can change an improper fraction to a mixed number, but they cannot tell you which is the improper fraction and which is the mixed number if you show them both forms and they certainly cannot tell you which situations call for one and which situations call for the other. They cannot explain why it is OK to multiply $.5x = 3$ by two to get $x = 6$ but not OK to multiply $.5x + 3$ by two to get $x + 6$. They call subscripts exponents and they call exponents coefficients. In short, they show no understanding, only blind manipulation Little wonder that many students cannot solve even the simplest word problem.

Here is what I would like to see a high school graduate able to do.

I would like a student to be able to look at problem such as $x^2 + 3x - 5$ and say, "That expression is an equation, because it has an equal sign, so I assume I should solve it rather than simplify it. In any case, it is already in its simplest form. It is a quadratic equation, so I know I can solve it, and that I should expect two solutions. I know two methods for solving quadratic equations, factoring and the quadratic formula I know a third method, completing the square, but the quadratic formula is easier, and always works. This particular equation doesn't look as if it will factor, so let's use the formula The equation is already in the proper form, since all of the non-zero terms are on the same side of the equal sign, so $a = 1$ (the understood coefficient of x), $b = 3$, and $c = -5$. Plugging those numbers into the formula gives me

$$x = (-3 + \sqrt{29})/2$$

This means that I have two answers, $(-3 + \sqrt{29})/2$ and $(-3 - \sqrt{29})/2$. These are both irrational real numbers, so I could get a decimal approximation for them. The square root of 29 is a little more than 5, so the answers are roughly one point something and negative four point something. If I needed a more accurate answer than that, I would look

the square root of 29 up in a table, or use a pocket calculator. I could get one or two decimal places by hand, but it would be a lot of trouble."

When there is a computer program which can teach a student to give an answer like that, then I will be all for using computers to teach basic skills. Until then, I think instead of working their way through basic skills games and programs, the students will learn a lot more by writing programs of their own.

Here are some examples of simple programming problems which can help students to learn basic skills.

1. Have the students write a program in Basic which will input a linear equation and solve it. For the average student, allow them to input the equation in some standard form, such as $ax + b = c$. The better students could tackle the problem with the equation input in any form.

2. Give the same problem, but with quadratic equations, and have them program the computer to check the answer given by the quadratic formula by plugging the answer into the original equation. Then ask them why the answer does not usually check, to see if they understand what round off error is all about.

3. Have the students write a program to change a fraction to a decimal. This is a very easy problem. Then challenge the better students to write a program which will change a decimal to a fraction (much harder).

4. Ask the students to write a program which will accept as input two mixed numbers and add them. This will help students to understand what a mixed number really is, and why the notation "3½" is so hard to get a computer to accept. This should also teach them something about the hierarchy of operations.

5. Program the computer to graph an equation. This is fun, because it gets into graphics. (It is too bad that the Apple graphics coordinates begin in the upper left hand corner of the screen instead of the lower left hand corner, but dealing with this problem will at least destroy some of the students' preconceptions about there only being one right way to do a task.) After the students can graph in the first quadrant ask them to try to graph in all four quadrants. After they can graph lines, ask them to graph circles.

These are just a few examples of programming problems which will force the student to think about what numbers are and what numbers mean, which is at least as important as the rote memorization of number facts, though the latter also has its place.

When I sat down to write a program to teach my daughter basic skills, I did not teach her anything at all. But in writing the program, I taught myself a great deal. In most cases, I think that the proper activity for students on computers is programming.

Contributors

M. BRUCKHEIMER
Department of Science Teaching
Weizmann Institute of Science
Rehovot, Israel

P. COBB
Education Building
Purdue University
West Lafayette, IN 47907, U.S.A.

P. GERDES
Faculty of Education
Eduardo Mondlane University
C.P. 257, Maputo, Mozambique

G. HANNA
MECA-OISE
252 Bloor Street West
Toronto, Ontario, Canada M5S 1V6

U. LERON
Department of Science Education
Technion: Israel Institute of Technology
Haifa 32000, Israel

J.H. MASON
Faculty of Mathematics
The Open University
Milton Keynes, MK7 6AA, U.K.

R. NORWOOD
Department of Mathematics
William Patterson College
Wayne, NJ 07470, U.S.A.

A. SENIENI
235 Villeneuve O.
Montréal, Québec
Canada H2T 2R8