



Mathematics and a Magical Trick of David Copperfield

M.A. Murray-Lasso

Unidad de Enseñanza Auxiliada por Computadora

Departamento de Ingeniería de Sistemas. División de Estudios de Posgrado

Facultad de Ingeniería, UNAM

(recibido: junio de 2001; aceptado: octubre de 2001)

Abstract

In order to make the learning of mathematics more attractive many of the topics should be linked to activities that the students enjoy such as games, magic, computers, animation, multimedia, etc. In this article the author links some mathematical concepts such as the conception of a function, its domain and range, the composition of functions, and some concepts of graph theory and logic to a magic trick performed on TV by the renowned magician David Copperfield. After a mathematical analysis of the trick and why it works, the trick is automated in a computer, given that it does not depend on sleight of hand. The language used for the automation is LogoWriter, a version of Logo. Listings of Logo procedures with explanations are provided in the article. The whole activity is an example of a computerized educational project in which a group of students under the guidance of a teacher solve a problem on a computer in several class sessions. The students, first of all, have a lot of fun. Second, they learn problem-solving. Third, they learn the use of computers in problem-solving. Fourth, they learn about the subject matter of the problem solved. The process provides a learning experience of the highest cognitive level in which students discuss among themselves possible approaches, they plan their actions, they execute them with intermediate tests, and at the end they can ascertain in a very natural way the degree of success of their efforts, with an opportunity to progressively refine the end result. The introduction of magic into the mathematics class in the way described sparks a phenomenon that the mathematics teacher seldom has the opportunity to witness: the students demand that the teacher provide a mathematical proof that the magical trick always works and they make honest efforts to understand it. This is something new in the teaching of mathematics.

Keywords: mathematical magic, function, domain, range, Logo.

Resumen

Para lograr que las matemáticas sean más atractivas para los estudiantes, muchos de sus temas deberían ligarse con actividades que les gustan, como juegos, magia, computadoras, animación, multimedios, etc. En este artículo el autor liga algunos conceptos matemáticos tales como el concepto de función, su dominio, su rango, la composición de funciones y algunos conceptos de teoría de gráficas y lógica a un truco mágico presentado en la televisión por el afamado mago David Copperfield. Después de un análisis matemático del truco y por qué funciona, éste se automatiza en una computadora, dado que no depende de la destreza manual del mago. El lenguaje que se utiliza es el LogoWriter, una de las versiones de Logo. Listados completos con explicaciones de procedimientos Logo se proporcionan en el artículo. La actividad completa es un ejemplo de un proyecto educativo computarizado en el cual un grupo de estudiantes bajo la

supervisión de un maestro resuelven un problema en una computadora en varias sesiones de clase. Los estudiantes, ante todo se divierten mucho. Segundo, aprenden a resolver problemas. Tercero, aprenden a utilizar las computadoras en la solución de problemas. Cuarto, aprenden sobre la materia del problema resuelto. El proceso les proporciona a los estudiantes una experiencia de aprendizaje al más alto nivel cognitivo, en el cual discuten entre ellos los posibles enfoques, planean la acción, la ejecutan con pruebas durante el proceso, y al final pueden revisar de una manera muy natural el grado de éxito de sus esfuerzos, lo que les permite refinar el resultado paulatinamente. La introducción de la magia a la clase de matemáticas produce un fenómeno que pocas veces el profesor de matemáticas tiene oportunidad de presenciar: los estudiantes le exigen al profesor que dé una demostración matemática de que el truco siempre funciona y se esfuerzan por entenderlo cabalmente. Esto resulta novedoso en la enseñanza de las matemáticas.

Descriptores: magia matemática, función, dominio, codominio, Logo.

Introduction

There is an urgent need to provide motivation for students studying mathematics at all levels. It is one of the subjects with lowest rates of successful completion, probably because it is presented in a formal, arid style by most teachers. On the other hand, students seem to enjoy magic, TV, music, electronic games such as Nintendo, computers, and puzzles. To couple the teaching of mathematics with these things students enjoy is a step in the right direction. That is the purpose of this paper. We intend to couple magic, computers and some mathematical topics.

Recently, in one of his TV shows, David Copperfield, the famous magician, performed a neat magical trick which I thought would be of interest to teachers interested in the use of computers to improve mathematical education. I have become interested in mathematical magic, or mathemagic, as some authors refer to it (Heath, 1953) because I have found that students become very motivated by it (just as computers motivate students) as they find it amusing and are even interested in mathematical proofs when they provide assurances that the trick will not fail under stated conditions. It is for this reason that the author is presenting the mathematical explanation and automation of this trick.

Description of the Trick

Because I did not write down the precise instructions given by Copperfield in his show, the following

description is only approximate, for example, his final answer differs from mine, but the essence of the trick is preserved. We start with the face of a handless clock with its twelve numbers arranged in the conventional manner as shown in Fig. 1

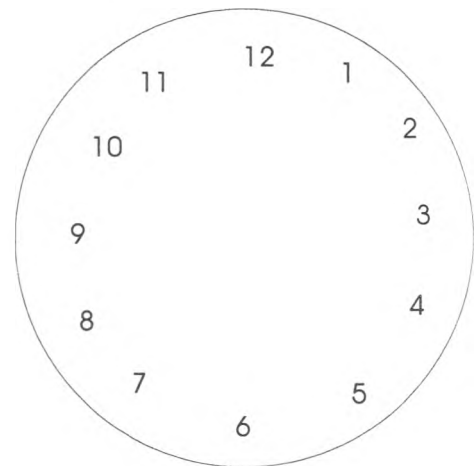


Figure 1

The instructions are as follows:

1. Think of any number from 1 to 12 and, starting at 12 on the clock, count as many hours as the number you thought and put your finger on the corresponding number. [For example if you think of the number 6 and you count six hours starting from 12 you end with your finger in 6].
2. By spelling the number where you have your finger advance as many hours as there are letters in the name of the number where you have your finger.

We will call this process SPELL YOUR NUMBER. [For the example, if you have your finger in 6 then by spelling SIX which has 3 letters — “S” corresponds to 7, “I” corresponds to 8, and “X” corresponds to 9 — your finger winds up in 9].

3. Once more SPELL YOUR NUMBER. (Using, of course, the number where your finger landed in the previous step).

4. I know that your finger is not in number 12, nor in 2, nor in 3, nor in 5, nor in 7, nor in 10, nor in 11, hence to make things easier they will be removed and the clock face now looks like

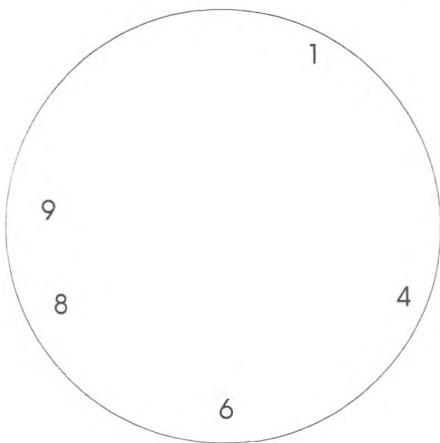


Figure 2

5. Once more SPELL YOUR NUMBER advancing for each letter, not one hour, but to the next number that is still in the clock. [For example if your finger was in 1 (ONE) then you advance to 8 because “O” takes you to 4, “N” takes you to 6, and “E” takes you to 8].

6. Once again SPELL YOUR NUMBER. Now I can remove all the numbers except 8, because my magic powers tell me your finger is definitely in number 8!

When Copperfield presented the trick I followed the instructions and, sure enough, I had my finger where he said. I reasoned that if millions of people were watching the show then all of them must have had ended with their fingers in the same number, which meant that, because with such a large number of people thinking initial numbers, there must have been many thousands thinking of every single number between 1 and 12, and therefore the trick should work for any initial number. I thus

proceeded to analyze the trick for every possible initial number as discussed in the next section.

Mathematical Analysis of the Trick

To analyze the trick for every possible initial number I built the collection of trios of numbers shown as table I

Table I

1	–	4	–	8
2	–	5	–	9
3	–	8	–	1
4	–	8	–	1
5	–	9	–	1
6	–	9	–	1
7	–	12	–	6
8	–	1	–	4
9	–	1	–	4
10	–	1	–	4
11	–	5	–	9
12	–	6	–	9

In table I, the first number is the initial number thought of, the second number is the number one lands in after spelling the initial number, and the third number is the number one lands in after spelling the second number. If we build the table by columns, we do not have to do any spelling to write the third numbers, we can simply consult the first column to see what number follows a given number. For example, in writing 8 as the third number in the first row we consult which number follows 4 (the second number of the first row) by consulting the second number of the fourth row.

We notice the following facts from table I: In the set of second numbers the numbers 2, 3, 7, 10, 11 are missing. We can deduce immediately that these numbers can not appear subsequently because they did not appear from a situation where the starting numbers entail all 12 numbers. We also notice that in the set of third numbers of the rows of table I, in addition to the numbers that did not appear in the set of second numbers, the numbers 5, 12 did not appear. If we call PREDECESSORS of A

those numbers which when spelled land in the number A, and SUCCESSOR of A the number we land in after SPELLING A, we notice that a number may have several predecessors but only one successor. The numbers that did not appear in the set of second numbers in Table I have no predecessors. If all the predecessors of a number are in the set of numbers missing in the set of second numbers (that is, of the numbers having no predecessors) then they will not appear in the set of 3rd numbers or any subsequent sets of numbers generated if the process were to be continued. Once a number starts missing at a certain stage it never appears again. This implies that the quantity of numbers not appearing in the different stages is a non decreasing function of the stage.

By the way, we have mentioned the word "function," a central concept in all mathematics which is very important the students get right from the very beginning. The preceding discussion is a good vehicle to introduce the concept of function. We would be talking of the function SPELL YOUR NUMBER, whose *domain* is the set of 12 numbers in the face of a standard clock, which coincides with the set of numbers we can think of at the beginning of the process. The *range* of the function is the set of numbers that have a predecessor which coincides with the set of numbers that appear in the second column of Table I. The teacher can also use the example to introduce the composition of functions and define the square of the function SPELL YOUR NUMBER. The range of the square of SPELL YOUR NUMBER is the set of numbers appearing in the third column of Table I which coincides with the set of numbers appearing in the face of the clock in Fig. 2. Although we will not go into greater details here the interested reader can consult teachers' books such as (Vannatta et al., 1962).

Graphs to Aid the Analysis

Some of the statements made can be seen more clearly if one draws a graph of transitions between numbers. A state transition graph is shown in Fig. 3.

All nodes can appear in the first stage because we can think of any number initially. We notice that

since nodes 2, 3, 7, 10 and 11 have no arrows entering them then they have no predecessors and will therefore be missing in the second stage and all stages following that since there is no way of getting to them except in the first stage when they can be "thought of". Suppose we eliminate these nodes with no predecessors and the arrows that lead from them to their successors. If in the updated graph there are some nodes without any arrows pointing to them (in other words, nodes with no predecessors) these would be eliminated in the next stage. Such is the case of nodes 5 and 12. The process could be continued and we could erase nodes 5 and 12 and the arrows pointing to their successors. Any nodes in the updated graph without predecessors would be eliminated in the next stage. In our case it would be node 6. The next step would eliminate node 9 and no further nodes can be eliminated without changing the successor function.

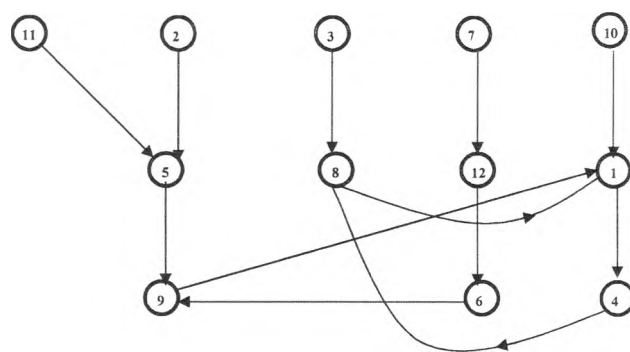


Figure 3

The process of eliminating nodes always ends with a graph where every node has at least one arrow entering and one leaving and hence through every node there passes at least one cycle. This can be seen because no nodes can have only arrows leaving, otherwise they could be eliminated; thus each one of the remaining nodes must have at least one arriving arrow, and by the nature of the nodes, each one must have one arrow leaving. Therefore starting at any node we can go to another node and exit from that node going either to a new one or returning to a previously visited node. If we return

to a visited node we have discovered the cycle; if we go to a new node, the new node can be entered and exited as before and after sufficient steps we must run out of new nodes because the graph has a finite number of them, so eventually we must return to a previously visited node and thus complete a cycle.

One more thing we can deduce is that, unless each number of the set that includes the numbers from 1 to 12 map one-to-one through the spelling process into the numbers 1 to 12, some numbers will disappear from stage 2 on. We can deduce this because if two numbers map into the same number, then that number has at least two predecessors which implies that at least one number must have none, given the fact that each number has only one successor.

The application of the previous facts to the trick is that at some stage we can be sure that the finger of no one is in certain numbers that have disappeared by that stage and we can safely drop them from consideration without objection from the audience. After the elimination of some numbers we have a new initial situation and new successor function in which hopefully additional numbers can be eliminated. If this is the case then we can continue to drop numbers until only one number remains.

For the English language the process converges to 8 when following the instructions given. The trick, however, accepts variations. Among the choices the magician has are the decision when to erase numbers from the clock and which numbers to eliminate among the possible ones. These two choices affect the successor function and can lead either to no final convergence or to convergence to different numbers. For example, if we spelled our number two more times before passing to the stage of erasing numbers (only the numbers 1, 4, 8 would remain) and left a clock with the three remaining numbers, we would have no convergence because in the clock with numbers 1, 4, 8, no number would have two predecessors given that 1 goes into 1, 4 into 8 and 8 into 4. Therefore it turned out to be convenient to erase numbers at the stage we did. On the other hand, we could have gone two additional stages until only 1, 4, 8 appear and erased some but not all of the numbers that had disappeared by the 4th stage,

for example leaving 1, 4, 6, 8, 10 without erasing. Even though the numbers 6 and 10 will not appear (they can not be starting numbers of a second round since they had already disappeared) their presence alters the successor function and the process will converge to number 8 in two more spelling stages. Graphically the situation of a clock with only the numbers 1, 4, 8 is shown in Fig. 4 while the situation with numbers 1, 4, 6, 8, 10 left in the clock is shown in Fig. 5. From the figures it is evident that the first case does not converge while the second converges to 8. The dotted arrows in Fig. 5 are never traversed but are shown because they could play a role in a possible variation which will be discussed in the following section.

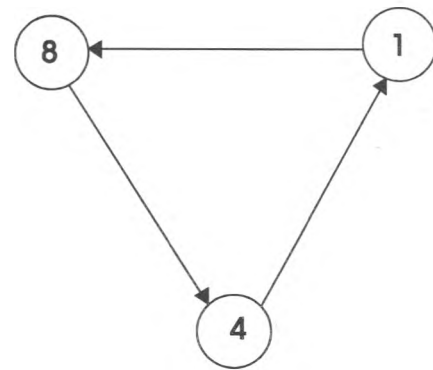


Figure 4

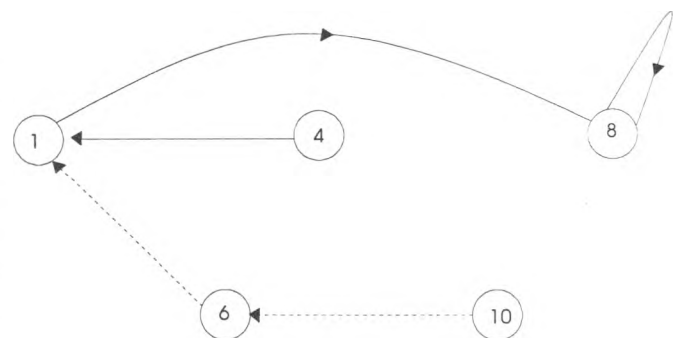


Figure 5

A Puzzling Variation

There are some points in the process where the magician can introduce a variation which will help in puzzling the audience. If when erasing numbers all of them eventually converge on a final number as

was the case of Fig. 5, then it does not matter in which of them a subject has his finger and one can introduce random moves without altering the final result. For example, the magician can ask the subjects to spell the first names of their mothers, in which case the finger could move arbitrarily to any of the nodes of a graph. Several such moves can be introduced for extra effect and as long as they are followed by three SPELL YOUR NUMBER stages (for the case of Fig. 5) the subject's finger will always end in 8. Another place where random moves can be introduced is at the very beginning of the trick since all 12 choices eventually converge. Here the magician could, for instance, ask the subjects to start at any number they choose, then spell their first name, then spell their last name traversing the clock in the opposite direction and then start the two stages of SPELL YOUR NUMBER. The process does not care about how the subjects arrived at the number before the two stages of SPELL YOUR NUMBER; in the next SPELL YOUR NUMBER stage it will eliminate the numbers 2, 3, 7, 10, 11 anyway.

Automating the Trick

Because the trick described does not depend on the magician's sleight of hand it can be performed by a computer. The program is very simple, it consists of exhibiting instructions in the form of text and drawing a clock, erasing some of the numbers from the clock, and finally erasing all the numbers except 8. The listing of a LogoWriter program that does this follows:

```

TO CLOCK
CT RG CC HT
INSTRUCTIONS1 MAKE "X READCHAR
CT DRAWCLOCK 80 PLACENUMBERS
INSTRUCTIONS1B MAKE "X READCHAR
INSTRUCTIONS1C MAKE "X READCHAR
INSTRUCTIONS1D MAKE "X READCHAR
INSTRUCTIONS2 MAKE "X READCHAR
ERASENUMBERS [2 3 5 7 10 11 12]
INSTRUCTIONS3 MAKE "X READCHAR
INSTRUCTIONS4 MAKE "X READCHAR
INSTRUCTIONS4B MAKE "X READCHAR
INSTRUCTIONS5 MAKE "X READCHAR
ERASENUMBERS [1 4 6 9]

```

```

END

TO DRAWCLOCK :R
CG
MAKE "RA :R
PU HOME FD :R RT 90
BK 3.1416 * :R / 36 PD
REPEAT 36[FD ((2 * 3.1416 * :R) /
36) RT 10]
END

TO PLACENUMBERS
MAKE "PED 3.1416 * 65 / 12
PU HOME FD 65 RT 90
FD :PED RT 360 / 12
MAKE "N 1
REPEAT 12 [PU FD :PED PD SETSH :N
STAMP PU FD :PED RT
360 / 12 MAKE "N :N + 1]
HT
END

TO ERASENUMBERS :LIST
MAKE "PED 3.1416 * 65 / 12
SETC 0 SETSH 38
PU HOME FD 65 RT 90 FD :PED
RT 360 / 12
TRAVERSE :LIST 12 1
END

TO TRAVERSE :LIST :V :N
IF (OR EMPTY? :LIST :V = 0) [STOP]
IFELSE :N = (FIRST :LIST) [PU FD
:PED PD STAMP PU FD :PED RT
360 /12 TRAVERSE BF :LIST
:V - 1 :N + 1][PU FD 2 *
:PED RT 360 / 12
TRAVERSE :LIST :V - 1 :N +
1]
END

TO INSTRUCTIONS1
CT
PRINT "|You will think of a number
from 1 to 12 and place your
finger on it in the clock that
will be drawn shortly. Then

```



```

you will repeat two times the
following operation called
SPELLING YOUR NUMBER:|
PRINT "
PRINT "|Advance your finger as
many hours as letters has
the number your finger is
pointing at. For example, if
your finger is on 6 (SIX)
you advance to 9 since "S"
takes you to 7, "I" takes
you to 8, and "X" takes
you to 9. Every time you
SPELL YOUR NUMBER you do it
with the letters of the
number where your finger
landed in the previous
step.|
PRINT " PRINT "
PRINT "|Press F1 to continue|
END

TO INSTRUCTIONS1B
CC
SHOW "|Place your finger on the
number you thought.|
SHOW "|Press F1 to continue|
END

TO INSTRUCTIONS1C
CC
SHOW "|SPELL YOUR NUMBER|
SHOW "|Press F1 to continue|
END

TO INSTRUCTIONS1D
CC
SHOW SPELL YOUR NUMBER once
again.|
SHOW "|Press F1 to continue|
END

TO INSTRUCTIONS2
CC
SHOW "|Your finger is not at 2,
nor at 3, nor at 5, nor at
7, nor at 10, 11 or 12. I
will remove them. Press F1
to continue|
END

TO INSTRUCTIONS3
CC
SHOW "|Advancing to the next
existing number per letter,
spell your first name. Press
F1 to continue|
END

TO INSTRUCTIONS4
CC
SHOW "|SPELL YOUR NUMBER
advancing to the next
existing number per
letter. Press F1 to continue|
END

TO INSTRUCTIONS4B
CC
SHOW "
SHOW "|SPELL YOUR NUMBER once
again.|
SHOW "|Press F1 to continue|
END

TO INSTRUCTIONS5
CC
SHOW "|I will now remove all the
numbers except where you
have your finger! Press F1|
END

```

Programming the magical trick can be the subject of a computerized educational project (Murray-Lasso, 1999). We now provide a short explanation of the program exhibited.

Explanations of the Listings of the Logowriter Procedures

The main procedure is CLOCK and it calls several auxiliary procedures that do specific tasks. The names of the procedures are self explanatory. The first four short two-letter instructions in the line

immediately following the title line of the procedure clear the text, resets the graphics, clears the commands, and hides the turtle. In the next line INSTRUCTIONS1 is called. This is the first in a package of instructions that will appear in the screen so the user knows what to do when. After each set of instructions the program waits till a key is pressed to continue executing. This is accomplished by the instruction MAKE "X READCHAR, which simulates capturing a character typed by the user and assigning its value to variable X. In actual fact it does not matter what character is stored in X, it is never used, the purpose is to wait till the user presses some key. Once the keyboard is pressed (the instructions ask for the key F1 to be pressed) the program clears the text of the instructions and with the procedure DRAWCLOCK and PLACENUMBERS draws the face of the handless clock and puts its 12 numbers in place. In each one of the next four lines sets of instructions are displayed by calling procedures with different names. These procedures appear at the end of the listing above. At the end of INSTRUCTIONS2 the program announces it is going to eliminate numbers 2, 3, 5, 7, 10, 11, 12. It does so by calling procedure ERASENUMBERS and give it a list which contains the numbers to be eliminated. Once this is done, several sets of instructions are displayed whose procedures also appear at the end of the listing. Finally at the end of INSTRUCTIONS5 the program announces it will erase all numbers except the number the user has her finger on (which is 8), so the program proceeds to erase all numbers except 8, then it erases the numbers 1, 4, 6, and 9.

We now explain the workings of DRAWCLOCK, whose listing appears immediately after the main procedure CLOCK. The procedure DRAWCLOCK accepts one parameter which is the radius of the circle representing the clock. The value used for the radius is 80 turtle steps. The procedure starts by erasing the area for drawings of LogoWriter (instruction CG). It stores the value of the radius in the auxiliary variable RA. With the pen up so that the turtle (which is hidden) does not draw while moving, locates the origin of coordinates in the center of the screen, points vertically upward, and advances R (radius) turtle steps and turns right 90 degrees. It then backs up one $\frac{1}{36}$ th of the length of the circle

and puts the pen down and repeats 36 times the following steps: advances one $\frac{1}{36}$ th of the length of the circle, turns right 10 degrees. After 36 times 10 degrees, the turtle will have turned 360 degrees corresponding to the full circle.

The procedure PLACENUMBERS starts by calculating the value of PED which is set to $\frac{1}{24}$ th of the length of a circle with radius 65 turtlesteps (recall the radius of the circle corresponding to outer rim of the clock is set at 80 turtlesteps). Next it takes the turtle, with its pen up, to the center of the screen and pointing up, and advances the turtle 65 turtlesteps and turns it right 90 degrees, advances it PED turtlesteps and turns it 30 degrees ($\frac{1}{12}$ th of a complete turn) and then starting with a counter N at 1, it repeats 12 times the following steps: raises the pen, advances PED turtlesteps, put the pen down, sets the Nth shape, stamps it, raises the pen, advances PED turtlesteps, turns right 30 degrees and advances the counter by one. For the procedure to work properly it is necessary that the shape file of LogoWriter have stored in shapes 1 to 12 turtle shapes resembling the numbers from 1 to 12. These shapes will be stamped in the proper places along a circle of radius equal to 65 turtlesteps.

The procedure ERASENUMBERS takes a list of numbers to be erased as an argument and with the aid of procedure TRAVERSE "walks" around the circle of numbers of the clock and when it finds a number that is in the list of numbers to be erased it stamps with the color of the background (color 0) a square shape which covers all the shape area, and thus has the effect of erasing what was stamped there before. For the procedure to work properly a square occupying the full area of the turtle shape must be stored in shape 38 of the turtle shape file.

The procedure TRAVERSE is a recursive procedure that takes three arguments: the list of numbers to be erased, the number of hours still to go on the clock (which is initiated at the value 12), and the value of a counter (which is started at 1). The procedure stops when either the list of numbers to be erased is empty (which means it has erased all the numbers required) or when it has "walked" through all 12 numbers of the clock and there are no more hours to go. The numbers in the list must be in

ascending order. Whenever the counter N coincides with the first number in the list, a square shape of the color of the background is stamped in the position occupied by the turtle, otherwise the turtle "walks" right through the number without doing anything (but through a recursive call without doing anything to the list, advancing the counter by one and diminishing the number of hours to go V by one). After each erasing action the procedure calls itself recursively with list of numbers to erase with its first element amputated, the counter advanced by one and the hours to go diminished by one.

In the INSTRUCTIONSx procedures there are only CLEARTEXT (CT), PRINT, CLEAR COMMANDS (CC), AND SHOW instructions. The PRINT instruction writes in the work area and the CT instruction clears that area. The SHOW instruction writes in the command area and the CC clears that area. The listings are self explanatory.

To operate the magic trick all the user does is to type CLOCK and press return. The program will provide instructions which the user must follow. In order to have the program place the numbers like a clock the user must have previously filled the turtle shapes in the FIGURES file from 1 to 12 with the numbers drawn by darkening the appropriate pixels. The procedure ERASENUMBERS uses shape 38 which is completely filled to erase the numbers stamped with PLACENUMBERS using the background color (0). Thus shape 38 must be a totally filled shape.

Conclusion

The author has taught school at various levels for the last 42 years and has found that mathematics is a subject that although ordinarily hated by many students can be made attractive if one makes a special effort. Students may find mathematics attractive when the computer is introduced in a game-like fashion with drawings, animation, color, sound and interaction. A second ingredient that students like is magic. They love to try the magic on their friends and relatives. But they also want assurances that the trick will not fail and are therefore willing to examine mathematical proofs that a trick will work. For some gimmick to be called a

trick, it should not be obvious that it will work, therefore, in general, a proof is required to convince a doubting Thomas. In this way, I have been able to interest students in algebra through well-known problems such as: "Think of a number. Multiply it by two. Add 10. Take half. Subtract the number you thought of. Add 2. Your answer is 7" and variations of the same theme, some of which can work with coins, cards, matches or other objects (Murray-Lasso (1), 1994). When the teacher explains why the trick always works, the students start to take a liking towards algebra. I have introduced tests for divisibility and arithmetic module N through tricks that depend on the fact that any multiple of 9 has 9 as its digital root and zero as its remainder after dividing by 9 (Gardner, 1961). Almost all geometrical theorems can be presented as magic tricks or puzzles if one works at it (Murray-Lasso 2, 1994). Binary arithmetic can be introduced through games like NIM (Gardner (2) 1956); the sum of arithmetic sequences can be introduced through magic squares (Andrews, 1960), and so on. In general Recreational Mathematics can be a good source of motivation on the part of students for learning mathematics (Gardner (1) 1956; (1) 1977); (Gardner (2) 1956, 1961); (Andrews, 1960); (Gardner (2) 1977).

In this article we have given a mathematical analysis of a magical trick performed on TV by a famous magician. Logic and graphs were used to analyze why the trick works and how to design variations of it. One can introduce graphs (Ore, 1962 and Mayeda, 1972) and finite automata as by-products (Hopcroft and Ullman, 1979). The concepts of function and relation can be explained when discussing predecessors and successors (Vannatta et al., 1962). The students can practice their interactive programming skills by automating the trick. The automation of the trick provides opportunities for the students to learn programming through a meaningful project that the students enjoy and through which they can learn problem solving and other useful things. For example, in the project of the article, drawing a circle of a desired radius is a problem to solve through the computer. Placing the numbers on the face of the clock is a second problem that requires the students learning how

recursive procedures work and how to stamp shapes and erase them. We gave a program listing in LogoWriter. The students can test their understanding of the theory behind the trick by using the Spanish language (the names of the numbers are: UNO, DOS, TRES, CUATRO, CINCO, SEIS, SIETE, OCHO, NUEVE, DIEZ, ONCE, DOCE), or French (the names are UN, DEUX, TROIS, QUATRE, CINQ, SIX, SEPT, HUIT, NEUF, DIX, ONZE, DOUZE). Other languages such as German, Italian, Polish, Portuguese can be used.

References

- Andrews W.S. (1960). *Magic Squares and Cubes*, Dover Publications, Inc., New York.
- Gardner M. (1) (1977). *Mathematical Carnival*, Vintage Books, New York.
- Gardner M. (2) (1977). *Mathematical Magic Show*, Alfred A. Knopf, New York.
- Gardner M. (1) (1956). *Mathematics, Magic and Mystery*, Dover Publications, Inc., New York.
- Gardner M. (2) (1956). *The Scientific American Book of Mathematical Puzzles & Diversions*, Simon and Schuster, New York.
- Gardner M. (1961). *The 2nd. Scientific American Book of Mathematical Puzzles & Diversions*, Simon and Schuster, New York.
- Heath R.V. (1953). *MATHEMAGIC: Magic, Puzzles, and Games with Numbers*, Dover Publications, Inc., New York.
- Hopcroft J.E. and Ullman J.D. (1979). *Introduction to Automata Theory, Languages and Computation*, Addison-Wesley, Reading, MA.
- Mayeda W. (1972). *Graph Theory*, John Wiley & Sons, Inc., New York.
- Murray-Lasso M.A. (1) (1994). La computemagia y la enseñanza de la aritmética en segunda vuelta. *Proceedings INFOCOM '94*, Ediciones Funda Austral, Buenos Aires, October, pp. 256-272.
- Murray-Lasso M.A. (2) (1994). Invariantes y acertijos matemáticos educativos, *Proceedings INFOCOM '94*, Ediciones Funda Austral, Buenos Aires, October, pp. 276-291.
- Murray-Lasso M.A. (1999). Proyectos educativos con logo y el aprendizaje de la geometría plana, *Proceedings of the 15th International Symposium on Computing in Education SOMECE '99*, Mexican Society for Computers in Education, Guadalajara, MEXICO, pp. 87 – 97.
- Ore O. (1962). *Theory of Graphs*, American Mathematical Society, Providence.
- Vannatta G.D., Goodwin A.W. and Fawcett H.P. (1962). *Algebra One: A Modern Course, Teacher's Annotated Edition*, Charles E. Merrill Books, Inc., Columbus, OH.

Semblanza del Autor

Marco Antonio Murray-Lasso. Realizó la licenciatura en ingeniería mecánica-eléctrica en la Facultad de Ingeniería de la UNAM. El Instituto de Tecnología de Massachusetts (MIT) le otorgó los grados de maestro en ciencias en ingeniería eléctrica y doctor en ciencias cibernéticas. En México, ha laborado como investigador en el Instituto de Ingeniería y como profesor en la Facultad de Ingeniería (UNAM) durante 39 años; en el extranjero, ha sido asesor de la NASA en diseño de circuitos por computadora para aplicaciones espaciales, investigador en los Laboratorios Bell, así como profesor de la Universidad Case Western Reserve y Newark College of Engineering, en los Estados Unidos. Fue el presidente fundador de la Academia Nacional de Ingeniería de México; vicepresidente y presidente del Consejo de Academias de Ingeniería y Ciencias Tecnológicas (organización mundial con sede en Washington que agrupa las Academias Nacionales de Ingeniería) y secretario de la Academia Mexicana de Ciencias. Actualmente, es jefe de la Unidad de Enseñanza Auxiliada por Computadora de la División de Estudios de Posgrado de la Facultad de Ingeniería de la UNAM, investigador nacional en ingeniería, consejero educativo del MIT y consultor de la UNESCO.