Int J Comput Math Learning (2007) 12:81–84
DOI 10.1007/s10758-007-9112-4

Computational diversions

Puzzles, problems, activities, and other things to destroy what little social life any of us might have managed to cobble together

Michael Eisenberg

Published online: 28 March 2007

This column will be devoted to ''recreational computing''. I'm almost tempted to end the entire column with that one sentence, in the anticipation that since everyone will have a different idea of what that means, people might be tempted to submit all sorts of strange or kooky or gorgeous ideas. My guess is that this would be a good thing. I'd like the column, if possible, to take an expansive view of both ''recreation'' and ''computing''. That is to say, a classic ''puzzle'' type of submission (e.g. ''write a simple program to generate the following colour pattern/turtle drawing/list of numbers'') could be just fine—but it needn't be the only type of entry for a column like this. Maybe your idea of recreation has more to do with building robots; or using ambient computing around your home to create a treasure hunt; or somehow (I have no idea how) weaving computers or computational ideas into party activities like pinning a tail on a donkey or playing charades; or generating amusing sound effects; or doing amateur magic tricks; or somehow using computational thinking in unexpected, eccentric, or otherwise mundane situations (like, say, determining a novel algorithm for avoiding the loss of socks in the dryer, or sharing a plate of French fries, or choosing flowers to bring on a date).

The basic criteria, then, that I will try to use for determining what might be included in this column are:

(a) The puzzle/problem/activity should be fun;
(b) It should prominently involve some sort of computational or algorithmic element;
(c) It should have some connection to, or spark some sort of thinking upon, an interesting idea in mathematics or science.

This leaves a lot of leeway for experimentation. Maybe it would be possible to incorporate computational ideas into otherwise classic puzzle ''chestnuts'' (like liars-and-truthtellers, or farmer-brings-bunch-of-stuff-across-the-river). Maybe it would be possible to add puzzle elements to otherwise classic recreational-computing standards such as generating Julia sets or snowflake curves, or running prisoner's-dilemma tournaments. Maybe ''static'' problems in geometry could be rethought as problems involving dynamic

M. Eisenberg (✉)
University of Colorado, Boulder, CO, USA
e-mail: ijcml-diversions@ccl.northwestern.edu

geometry or computer animation. Perhaps one might generate problems or activities around computational themes such as fuzzy logic, or distributed computing systems (as in the ‘‘dining philosophers’’ problem), or random graphs. Even the most nuts-and-bolts topics in the computer science curriculum (dealing with round off error, say) might be a source of new types of problems.

Readers of this column are thus requested to send in examples of original puzzles or activities (and accompanying solution or solutions, where applicable) for publication in this column. Submissions, comments, solutions, and any other correspondence should be sent to the email address above: please put ‘‘IJCML Puzzle Column’’ in the subject line, followed by a word or two indicating what type of correspondence this is (e.g., ‘‘IJCML Puzzle Column Submission’’, ‘‘IJCML Puzzle Column Solution’’, and so forth).

There will be a mild and probably not entirely predictable preference for ‘‘activities’’—things that get the reader involved in creating his or her own programs or examples—over less participatory examples (like, say, riddles). There will be a mild preference for aesthetics (e.g. examples involving computer graphics or beautiful geometric effects) over plain old text (e.g. ‘‘In-eight-years-I-will-be-twice-as-old-as-my-son’’ type puzzles). There will be a mild preference for creative or offbeat types of puzzles as opposed to usual suspects. Puzzles or activities that can be done in the tub or on the commuter train are fine; but if the puzzle or activity for some reason has to be done in a strong wind, or in the presence of your dog, or in a group of at least six people, that’s great too. Nerdy activities that sabotage the atmosphere of a fancy-dress party, hot tub gathering, prom, college reunion, or jury room deliberation are particularly welcome.

Stuff that’s discouraged would probably include puzzles that aren’t really puzzles at all—like word search grids, for example. (I really have no idea why anyone does word search; like listening to seventies music all day, it is simply a test of dogged patience.) Video game themes or situations might just be okay; but ‘‘tips-’n’-tricks’’ are discouraged. Not to be too snobby about matters, but anything even remotely involving identifying celebrities, lines from movies, pop song titles, TV characters, and so forth better have something really strong going for it in order to be considered. Bridge, chess, backgammon, or Go puzzles are slightly suspect (even assuming they incorporate some novel computational or algorithmic element); Twister- or Mousetrap-themed puzzles might prove more promising. Sports-themed examples should be comprehensible (no croquet), dignified (no wrestling, even if it’s genuine), and should in fact have to do with sports (no synchronized swimming). Sudoku variations would need to be very strange, disturbing, or anxiety-provoking in order to make the cut.

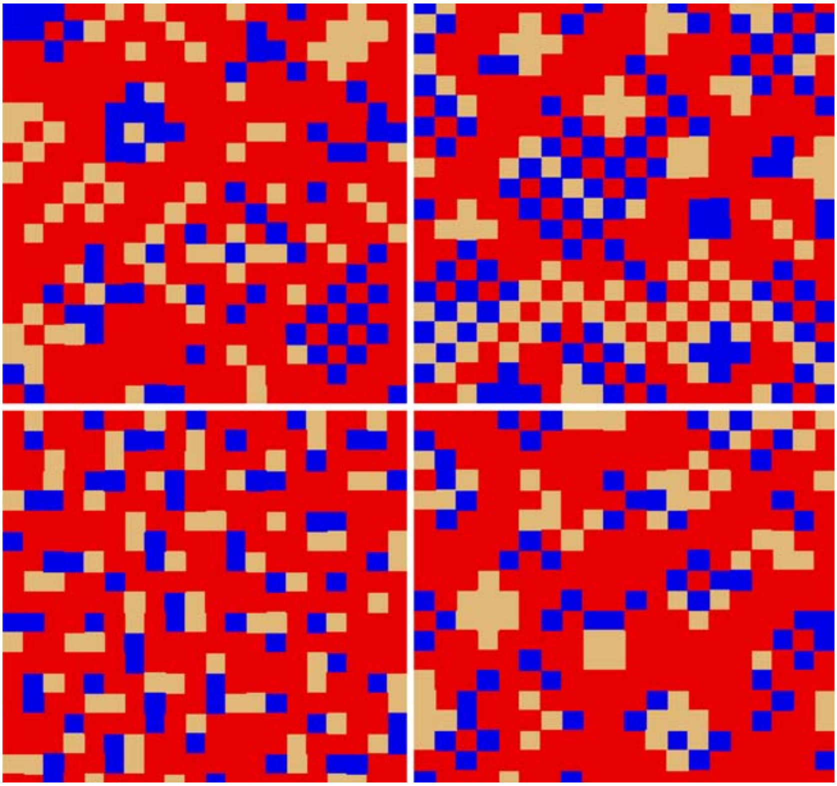
I expect to publish this column in every issue of IJCML. We’ll now begin with the first of what we hope is a long series of ‘‘computational diversions’’.

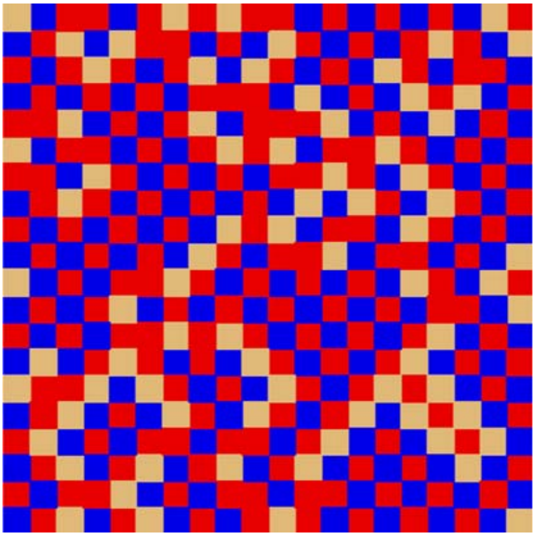
1 Diversion-12.1

This issue’s diversion concerns a visual example—a set of colour patterns produced by a simple one-parameter procedure. Imagine that we have a 20-by-20 grid, divided up into 400 unit squares, on a torus (that is, our 20-by-20 grid has wraparound both horizontally and vertically). Initially all the squares are empty (or white, if you like). We now colour each square with red, blue, or yellow according to the following rule:

```
Choose a critical Manhattan distance D.
Repeat until all squares are filled:
    Choose a random unfilled square.
    If the square is within D of both a red and blue square, and not D away from another
    yellow square, fill it with yellow.
    If the square is within D of a red square and not D away from another blue square, fill it
    with blue.
    Otherwise fill it with red.
```

Not too complicated. Yellows ‘‘want to be near’’ reds and blues, but not precisely D away from each other; blues ‘‘want to be near’’ reds, but not precisely D away from blue. Reds go anyplace else. Again, note that distances are measured as ‘‘Manhattan distances’’; that is, if you have to ‘‘hop’’ at least N squares in any of the four cardinal directions to get from one square to another, they are separated by N units in Manhattan distance. Now, the following pictures were generated with D values of 1, 2, 3, 4, and 5; but of course they are not necessarily shown in that order. Your task is to determine which of the drawings below corresponds to each value of D.





You may also notice a region or two of interest in these pictures (for example, the occasional ‘‘checkerboard’’-type region in several of the pictures). Naturally, readers are encouraged to play with this basic idea–try adding in more colours, varying critical distances for different colours (maybe blue squares need to be near red, but yellow squares have to be at least one unit away from red), increasing the size of the grid, and so forth.

This is not a particularly tricky example, but it’s at least somewhat representative: it involves computation, sparks a few questions, and invites further investigation. As a reader of this column, you might simply wish to send in a solution to the puzzle (that is, an assignment from pictures to values of D); in this case, the subject line of your email to the address above should read: IJCML-Diversion12.1-Solution. (Names of early solvers will be included in the next column.) If you have an interesting variation or extension on the basic program structure of this puzzle, the subject line of your email should read: IJCML-Diversion12.1-Variation. Depending on the interest and number of such variations, they may be included (I make no promises, though) in future columns or, once we get it going, on a column website. Finally, if you have a new puzzle or activity (or random interesting thought), send email to the address above with the subject line: IJCML-Diversion-New-Submission. The best such suggestions will be included in future columns.