Tech Know Learn (2011) 16:267–272
DOI 10.1007/s10758-011-9188-8

Computational Diversions: WordWiring, a Network-Themed Party Game

Michael Eisenberg

Published online: 7 December 2011

Fans of "mathematical games" (I'm one) often end up with a small library of books on the subject, collectively containing descriptions of a vast trove of games. There are board games, or informal pencil-and-paper games, with applications to (e.g.) graph theory, number theory, combinatorics, logic, and many other branches of mathematics. In fact, the range and variety of mathematical games is so wide—and so rich in content—that one could plausibly base a pretty good introductory mathematics curriculum on these pastimes. Perhaps a few of the more advanced subjects would be underrepresented in this exercise (real analysis? differential geometry?); but a typical library of mathematical games provides, overall, a broad portrait of mathematics in general.

Now, coverage of *mathematics* is one thing; but does our library provide reasonable coverage of what we mean by *games*? Interestingly, there are certain important types of games that tend to be near-invisible in recreational mathematics. For one thing, almost all the examples in our library are either solitaire games (which by some definitions of the term would hardly qualify as "games" at all), or two-person games. Think of the standards: tic-tac-toe, Nim, Sim, Sprouts, chess, Go, and so forth. Occasionally there are three- or four-person variations of these games, but for the most part we're talking about n players, where n is a small positive integer. (Perhaps, if we program a computer to play a mathematical game against itself, we could say that $n = 0$ in that case?)

Not all games are like this. After all, baseball is a game, isn't it? You need to round up 18 people at the very least to play a weekend-afternoon informal game of baseball. Even a pick-up basketball game will usually have at least six players; touch football would require a dozen or so. And how about children's favorites, such as musical chairs? Starting a game of musical chairs with only two or three kids on hand seems hardly worth the effort. Other children's classics—Tag, Duck-Duck-Goose, Capture the Flag—are best played with numbers ranging from a half-dozen (at the low end) to twenty or more. For older children and adults, there are games such as charades, or twenty questions, and these are typically played in moderate-sized groups (three would be a bit small for these games). In other

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

words: in many non-mathematical situations, two-person games are the exception rather than the rule. Lots of people play games that require lots of people.

This being the case, how come mathematicians only come up with one- or two-person games? It could be that mathematicians don't have many friends, but that's too depressing a line of thought to pursue here. In any event, as a computer scientist, it seems to me that there are ample possibilities for computationally-oriented playground and party games. Computer scientists are used to thinking about (e.g.) parallel processors, or multiple computations going on at once, or issues in voting and auctions, or emergent macroscopic phenomena arising from large numbers of simpler concurrent actions. The potential for translating these thematic ideas into group pastimes seems very strong, if we put our minds to it.

Not long ago, I brainstormed with my colleagues at the University of Colorado, Antranig Basman and Clayton Lewis, about these issues. Would it be possible to come up with a group game—say, a party game, or one that could be played in a classroom—that would highlight interesting computational ideas? Collectively, we came up with a few broad directions; in this column, I'm going to describe a game (or really a template for a family of games) that we imagined, and that illustrates interesting computational ideas. To be frank, I'm not sure that Antranig, Clayton, and I have hit upon exactly the perfect game along these lines—I have the sense that we're *close* to a good game, but not quite there. You, as reader, can judge for yourself: and if you can improve on the variants presented here, by all means let me know! (Or market your own perfected version and feel free to pursue your fortune.)

The game is called *WordWiring* and it's basically a variant on the summer-camp classic "Telephone". If you're not familiar with Telephone, the rules are as follows (at least as far as I can remember them): you begin with a line of children, perhaps as many as twenty or twenty-five. You write down a sentence—preferably, one that's a little complex or difficult to recall easily. Let's say the sentence is, "I have never been a particular fan of statically typed languages." (The fact that many summer-campers wouldn't generally say a sentence like this is actually a plus.) Then you whisper the sentence to the first child in the row; that child is supposed to turn to the next child and whisper the exact same sentence; that child then turns to the next child and whispers the sentence; and so on, all down the row. At the very end of the row, the final child writes down the sentence as they remember hearing it whispered from the next-to-last child in the row. By the time the sentence has passed through twenty or more "relays" of this sort, it will generally look quite different than the original, even if (a pretty big "if") all the children in the row are trying their hardest to repeat exactly what they've heard. Not infrequently, the final sentence hardly even shares any nouns or verbs with the original. The original and final sentences are now read out to the assembly of children, ideally to the general mirth of the company.

WordWiring shares with Telephone the essential idea of a communicative network, but unlike Telephone it is not really about the propagation of errors, but rather about the nature of verbal concepts. Again, there are several potential variations of the game; let's start with the simplest, which we'll call WordWiring 1 (maybe not the easiest thing to pronounce aloud). Here's the idea: imagine a room of fifteen players, lined up in five rows of three each, as in Fig. 1. Now, we'll imagine that there are still two other players, beyond the fifteen in our array: the first is a "word provider" and the second is a "word guesser". The "word provider" chooses a simple starting word (say, "dog"), writes it down on five slips of paper, and hands the slips to the first player in each row. Now, each of those five players writes down a word associated with (but not identical to) the word they were given, and passes it on to the second player in the row. The second player in each row now writes down a new associative word (associated, that is, with the particular word they received) and sends that new version to the third player. Finally, each of the third (and last) members of each row writes down a new association.

In short, then, we have begun with five instances of the word "dog" (at the beginning of each row), and conclude with five third-order associations. For example, as illustrated in Fig. 1, we might see the following pathways of associations in the five rows:

Dog → Cat → Mouse → **Cheese**
Dog → Bone → Tendon → **Muscle**
Dog → Cat → Dog → **Cat**
Dog → Puppy → Frisky → **Peppy**
Dog → Wolf → Coyote → **Fox**

Now that we have five fairly distant associations to "dog" (namely, *cheese*, *muscle*, *cat*, *peppy*, and *fox*), the job of the word guesser is to guess the original word with which all the rows started.

To be honest, I haven't yet tested out this game structure on a roomful of willing students, so I don't know for sure how playable it is. I *think* it could work, but there are open questions. For example, how likely is it to guess "dog" from the five word associations given above? It seems possible, if the guesser is fairly intuitive; but maybe the rate of success would be painfully low in general. Of course, we could try the game with a larger number of rows (and players); what if there were ten, as opposed to just five, third-order associations with "dog"? Would that make the game substantially easier?

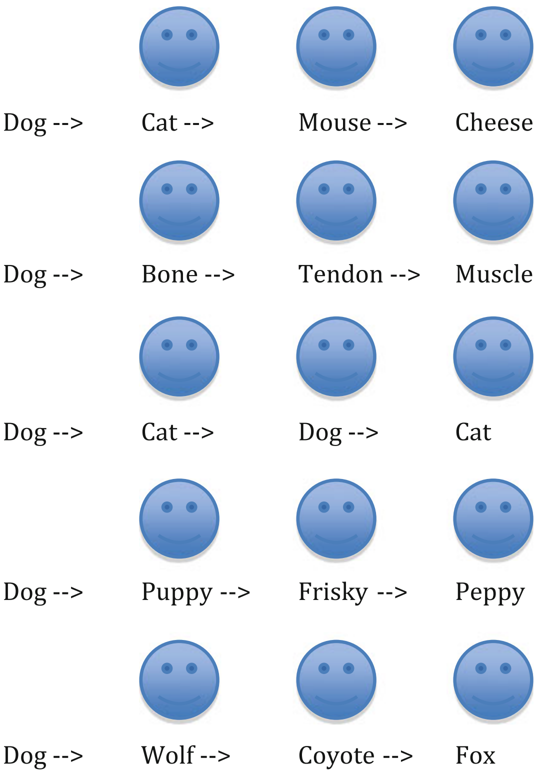

Fig. 1 A diagram representing the basic structure of WordWiring. Here, there are five rows of three players each. The first player in each row is given the same starting word (here, *dog*); that player then chooses an association word, and passes it over to the second player in the row. The second player in each row then, in turn, passes an association to the third player in the row. Finally, the third player forms still another association. The five words in the rightmost column of players now represent five "clues" to the original word *dog*. Another player, the "word guesser" (not shown) now has the job of guessing the original word

A slight extension of WordWiring 1, which we can call WordWiring 1.1 (even harder to pronounce than the simpler game!), might allow the word guesser to "travel back" along the rows. Here's the idea: suppose the guesser can't arrive at "dog" from the set *cheese*, *muscle*, *cat*, *peppy*, *fox*. In that case, the second member of each row can provide their associations to the guesser: *mouse*, *tendon*, *dog*, *frisky*, *coyote*. (Note that one of the "associated words" is in fact the original word *dog*; that's allowed in this version of the game, and we'll return to that point shortly.) Now could the guesser succeed? And if not, we could provide the guesser with the first-order associations: *cat*, *bone*, *cat*, *puppy*, and *wolf*. It seems likely that many folks could in fact correctly guess *dog* from that first-order set.

We could try yet another extension to our version 1.1 game; let's call it WordWiring 2, since it involves a slightly larger change in the rules. In this case, each player in the row passes along, not a single word, but the chain of words up to this point. Thus, in our example above, the first player in the first row wouldn't just pass along the word *cat* to their neighbor, but instead would pass on "*dog* → *cat*". The rules would now stipulate that the next player may not choose an association word that appears in the path up to this point.

Here's a hypothetical pattern of associations for WordWiring 2, using the same starting point (*dog*) for each row, and assuming that the first player in each row makes the same initial association as before:

Dog → Cat → Mouse → **Rabbit**	(originally, **Dog** → Cat → Mouse → **Cheese**)
Dog → Bone → Bark → **Growl**	(originally, **Dog** → Bone → Tendon → **Muscle**)
Dog → Cat → Kitten → **Yarn**	(originally, **Dog** → Cat → Dog → **Cat**)
Dog → Puppy → Frisky → **Peppy**	
Dog → Wolf → Coyote → **Fox**	

This pattern of associations is just an imagined sample, off the top of my head; but in any case, it's clear that our rule change would make a difference in at least some sequences of associations. In the first row, I imagine that the final player in the row, seeing not just *mouse* but *dog* → *cat* → *mouse* as input, would be more likely to name yet another small mammal. In the second row, I imagine that the second player in the row, receiving not just *bone* but *dog* → *bone*, would tend to have a different interpretation of the word *bone* (as dog food, not as a unit of a skeleton). In the third row, the second player simply can't reuse the word *dog* according to the new rules, and thus might choose something like *kitten*.

Now, would WordWiring 2 be any easier, or more playable, than WordWiring 1.1? I'm not sure; the third-order associations in this particular example (*rabbit*, *growl*, *yarn*, *peppy*, *fox*) still look like a pretty difficult set from which to guess *dog*. On the other hand, if we allow "traveling back up the rows" as in WordWiring 1.1, then the guesser might now get to see the second order associations: *mouse*, *bark*, *kitten*, *frisky*, *coyote*. It certainly seems at least possible to guess *dog* from that set.

There are a few computational themes and ideas that are touched upon in the structure of WordWiring, but perhaps the most prominent is the idea of a *semantic network*, a staple data structure of classical artificial intelligence courses. [See, for example, Russell and Norvig (2010, pp. 453–456) and Schalkoff (2011, pp. 20–25).] Without going into too much detail, a typical semantic network is a graph in which the vertices represent concepts of some sort (say, *dog*, *cat*, *bone*, *wolf*, and so forth) and edges between vertices represent a plausible association between concepts. Again, there is much more to say about the uses and limitations of semantic networks; but for our purposes, it is enough to note that many programs employing semantic networks make use of the idea of *spreading activation*. This means that if one renders a particular vertex of a semantic network "active" (say, the concept *dog*), that vertex will in turn communicate lower levels of activation to its neighboring concepts in the graph (likely including such conceptual nodes as "*cat*", "*bone*", "*wolf*", and so forth); and then these mildly-activated nodes will in turn spread activation to still other nodes. Rendering a node active is thus intended to be analogous to "thinking of a concept", or "having a concept suggested to one"; and spreading activation is analogous to the way in which one concept unconsciously suggests still others. With this broad analogy in mind, semantic networks have been used (among other purposes) to provide at least loose representations of human declarative memory.

The connection between our WordWiring games and semantic networks is readily apparent: in effect, our rows of students are acting as pathways of spreading activation from an initial concept, and the goal of the guesser is to think backward along the associative pathways to the original concept. As already noted, I'm not sure that WordWiring would quite work as a large-group game, but it doesn't seem too far from an activity that could be a lot of fun, and that sparks thinking about the nature of concepts and their associations.

Before leaving our development of WordWiring, we can suggest at least one more large-scale change in the rules: let's call this version WordWiring 3. In this case, we imagine that the second player in each row doesn't just receive one association (from their immediate neighbor in the row) but rather receives all the associations from the first player in every row. In our previous example (from WordWiring 2), this would mean that the second players in rows 1 through 5 would each receive, in full, the following information:

Set 0: Dog
Set 1: Cat, Bone, Cat, Puppy, Wolf

Now, the second player in each row independently comes up with yet another association based on Sets 0 and 1, and contributes these to Set 2, again subject to the proviso that they cannot repeat any words from their own input. The resulting cascade of sets might now look like this:

Set 0: Dog
Set 1: Cat, Bone, Cat, Puppy, Wolf
Set 2: Mouse, Kitten, Bark, Mouse, Chase

This entire collection of sets would now be passed to the third player in each row, and they produce the final set of associations:

Set 3: Meow, Collar, Squeak, Leash, Litter Box

This final set of associations is then given to the word guesser; and as in WordWiring 1.1, an unsuccessful guesser may ask, by steps, for the previous sets of associations. Thus, if the guesser couldn't manage *dog* from *meow*, *collar*, *squeak*, *leash*, and *litter box*, they could try guessing from the set *mouse*, *kitten*, *bark*, *mouse*, and *chase*.

Here, we've extended the original structure of WordWiring so that each column of players receives all the inputs from previous columns. In the earlier versions of the game, each row worked completely independently of the others, and in parallel; now we have something closer in spirit to the connection pattern of a feedforward neural network. I suspect that WordWiring 3 would provide tighter association patterns than the earlier versions, and would result in a somewhat easier task for the word guesser; on the other hand, it might be harder to implement logistically in a room (think of all those slips of paper passing back and forth!).

It's an empirical question whether any of these games would provide fun for a large number of players—say, at a party. It might take a relatively nerdy gathering of folks to enjoy playing WordWiring; but then again, those are the sorts of gatherings that I personally am more likely to attend. In any event—to return to the theme that opened this column—it's about time that we think of recreational computing (or recreational mathematics) beyond the context of one or two players. If WordWiring doesn't quite fill the bill, then it's up to you readers to improve on it, or (better yet) to come up with your own brand-new group and party games. Readers with ideas along these lines are encouraged to send them to this column at: ijcml-diversions@ccl.northwestern.edu.

References

Russell, S., & Norvig, P. (2010). *Artificial intelligence: A modern approach*. Upper Saddle River, NJ: Pearson.

Schalkoff, R. (2011). *Intelligent systems: Principles, paradigms, and pragmatics*. Sudbury, MA: Jones and Bartlett.