SOFTWARE—PRACTICE AND EXPERIENCE, VOL. 2, 93-96 (1972)

Computer Recreations

by

$\aleph_0$

‘Darwin’

Ten years ago, Vyssotsky† invented something completely different from the usual computer games. Darwin is a game between computer programs as programs. The object is survival; programs may ‘kill’ one another, and may create copies of themselves in unoccupied store.

The game is played in a region of store called the arena and is mediated by an umpire. The programs (organisms) of one player constitute a species. The organism currently in control may communicate with the umpire by means of the three umpire calls PROBE, KILL and CLAIM. PROBE is used to ask what is in a particular part of the arena, KILL to kill another organism and CLAIM to claim some empty space for the organism to reproduce itself.

The detailed rules are as follows:

(1) A species has a species number N, a size $S(N)$, a start location $G(N)$ and a set of protected locations $P(N,1)$, $P(N,2)$, ... , $P(N,R)$.
  (a) The number of protected locations, R, is an implementation-dependent function of the species size $S(N)$.
  (b) Each of $G(N)$, $P(N,1)$, ... , $P(N,R)$ must be less than $S(N)$.
  (c) $S(N)$ must be less than an implementation-defined constant S .

(2) The game is played in a region of core called the arena, of a size A which is implementation-dependent.

(3) The game is managed by a program called the umpire, which is outside the arena.

(4) An organism of species N is a program within the arena, which obeys the following conditions:
  (a) It consists of $S(N)$ locations, L, $L+1$, ... , $L+S(N)-1$.
  (b) Its protected locations are $L+P(N,1)$, ... , $L+P(N,R)$.
  (c) Its contents may be arbitrary, save that when executed starting at $L+G(N)$:
    (i) It may read any location within the arena, but may not read any location outside it.
    (ii) It may write to or execute or jump to any location within itself or within an area claimed by CLAIM, or within an organism of the same species found by PROBE.

† V. A. Vyssotsky, Bell Telephone Laboratories, New Jersey.

© 1972 by John Wiley & Sons, Ltd.

(iii) It may in appropriate circumstances obey any of the three umpire calls, PROBE, KILL and CLAIM.

(iv) It may not make use of input/output or of traps of any kind that might return control to it after it had lost control. Traps within the program may, however, be used.

(d) An organism may use any computational procedures whatsoever subject to the above restrictions. Supervisor calls (Extracodes, Executive calls, or whatever) may be used, provided that they obey the above restrictions also (where while executing a supervisor call the interior of the Supervisor is regarded as part of the organism).

(e) Distinct organisms of the same species need not contain the same code.

(5) PROBE is a call to the umpire with two arguments, the species number, N, of the calling organism, and a location LOC within the arena.

(a) If LOC is a protected location of an organism, control is transferred to that organism (at its start location).

(i) The new organism receives no indication of how it obtained control.

(ii) The old organism receives no indication of how it lost control other than the fact that when (if ever) it regains control, it is entered at its start location rather than at the return from PROBE.

(iii) On a transfer of control, all record of which locations have been probed is lost by the umpire.

(b) If LOC is not a protected location of some organism, then PROBE returns three numbers: HISNO, BOT and TOP.

(i) HISNO is zero if LOC is empty, and otherwise the species number of the organism occupying LOC.

(ii) BOT and TOP are the limits (first and last + 1) of the largest contiguous space surrounding LOC, which is empty except for the occupant of LOC, if any.

(6) KILL is a call to the umpire with two arguments as for PROBE.

(a) LOC must be a location probed by an organism of the calling species, and there must be an organism at LOC, distinct from the calling organism, but not necessarily of a different species (suicide is forbidden but cannibalism is allowed).

(b) The effect of KILL is to destroy the organism at LOC. The carcase is left.

(7) CLAIM is an umpire call with two arguments as for PROBE.

(a) LOC must have been probed by an organism of the calling species, and the $S(N)$ locations starting with LOC must be empty, possibly as a result of an intervening KILL.

(b) The effect of CLAIM is to cause the umpire to record the presence of an organism of species N at LOC.

(c) The calling organism may then write to or jump into the area CLAIM'ed (presumably it will copy itself there).

(8) It is at the option of the implementation which infractions of the rules are detected.

(a) A detected infraction of the rules results in extermination of the offending species.

(b) In implementations where not all illegal actions are detected by the umpire in the course of the game, players are required to show one another their programs after a game and, if necessary, explain any procedures used in order to verify that they keep to the rules.

(9) To play a game, each player submits $K(N)$ initial organisms, which need not be identical, written in an agreed language, together with a specification of the size, start location, and protected locations of his species.

(a) $K(N) =$ integral part of $\dfrac{\text{size of arena}}{2 * (\text{number of players}) * S(N)}$.

(b) The organisms are loaded at pseudo-randomly chosen locations in the arena.

(c) Control is initially given to some organism, chosen in an implementation-dependent way.

(10) The game ends when only one species remains, or at the end of a fixed time, whichever occurs first.

(b) The winner is the player whose species has most organisms left.

Vyssotsky and his co-players recently sent me a description of Darwin, which they had implemented on an IBM 7090. The first thing Archimedes and I did was to make our own implementation. The rules marked implementation-dependent above mostly mark the places where we differed from Vyssotsky.

The original function R, defining the number of protected locations, referred to in rule 1(a) was 20 (regardless of size). However McIlroy† invented a virus—an unkillable organism. This was of length only 15 (7090 instructions), so that all its locations were protected. It could only PROBE and KILL, but this was enough to be able to win a few games. Another disadvantage of a fixed number of protected locations is that it discourages sophisticated strategies since these lead to large organisms with large vulnerable areas. We have experimented with square-root and linear laws for R, but we do not yet have enough results to know which is best.

Neither implementation detects breaking of rules other than 6(a) and 7(a). To do so would require either unusual memory protection hardware, or interpreting the programs. Both were discarded in favour of the 'honesty rule' 8(b). We differ, however, about the precise meaning of rules 6(a) and 7(a): the Bell Laboratories umpire remembers only the last probed location; ours keeps a map and marks all locations between BOT and TOP as probed.

The pseudo-random loading is difficult to deal with in practice; most loaders are not well suited to scatter loading. At Bell Laboratories they used a program which punched out a suitable set of loader origin cards and a table for the umpire. Ten years of system development later, we also use a pre-pass to set up loading data, but we are then able to enter the loader directly. Our system produces copies of each organism from a species template, so that the possible initial variety envisaged by rule (9) is not permitted. We feel we have not lost much; at Bell Laboratories they experimented with a bisexual species, with two types of organism, one to PROBE and KILL, and one to reproduce either sort. However the two types spent much of the time trying to find one another, and of course the whole species could be killed by exterminating all of either sort.

The species which eventually wiped out all opposition at Bell Laboratories was devised by Morris‡. This occupied 44 memory cells (of the 7090) and used an ingenious adaptive search strategy. Once an organism of his species had located the top of a gap, and thus the bottom of some other unknown organism, it would pick a small integer and PROBE that far up into the unknown organism. If it successfully found an enemy, after

† M. D. McIlroy, Bell Telephone Laboratories, New Jersey.

‡ R. Morris, Bell Telephone Laboritories, New Jersey.

killing it and reproducing itself if possible in the space freed, it would use the same probing increment on the next organism in the arena, and so on. If, on the other hand, it lost control, if and when it regained control it randomly chose another probing increment. A newly reproduced organism was initialized with the known successful increment of its parent. In this simple manner Morris produced a species each subspecies (more exactly, clone) of which was adapted to be lethal against the pattern of protection of some other species. We think that other more sophisticated strategies will beat Morris's if the protection function is not weighted so heavily in favour of small organisms. However, we have not been playing with our own Darwin system for long enough for steady trends to emerge.