THE CORE WAR *NEWSLETTER*

The Official Newsletter of the International Core War Society
Celebrating Ten Years of Core War: 1984 — 1994

Features

Letter From The Director **2**
by Jon Newman

Imp Rings, Part I **3**
by Steven Morrell

Articles

RoboWar — A Macintosh Combat Game **7**
by Jeff Lewis

How Can You Train Warriors? **9**
by Adam Ryba

Columns

The MADgic Corner **10**

Volume 7, Number 4 Fall 1994 An ICWS Publication

Letter From The Director

by Jon Newman

ICWS members:

It has been my intention for some time to seek candidates to replace me as Director of the ICWS. While the ICWS has been active and stable under my tenure, there are others who are more active in the Core War community, hopefully at least one of whom is willing to take on the task of Director.

Due to the conditions under which I assumed the Directorship of the ICWS, the ICWS election for Director will be advisory only, with the understanding that I will not necessarily step down if I cannot ensure that the winner of the election will actually become Director. If I do remain Director, I will discuss with the winner how we might share the responsibilities of the Directorship, including newsletter publication, maintenance of the membership list, etc. through another official position.

This issue of The Core War Newsletter marks the beginning of the election procedure. I propose that we follow this process:

Candidates for the Directorship must announce themselves to the current Director (me) before the next Core War Newsletter. I reserve the right to require candidates to demonstrate their earnestness by submitting a sample Newsletter as they might publish it. The next Newsletter will contain a ballot listing all candidates. To encourage voting, all members who vote will have their membership extended for one additional Newsletter. Directors Emeritus are also entitled to vote, but Branch Section members who are not also paying ICWS members are not entitled to vote. Candidates are encouraged to submit a statement to be published as part of the ballot.

The Newsletter after the ballot will contain the final voting tally. The recipient of the most votes will win this election. There will be no runoff. The current Director will break ties.

Most importantly, I hope that those members who want to improve the ICWS and speed

Continued on Page 11

Imp Rings, Part I

by Steven Morrell

[Editor's Note: The following is exceprted from Steven Morrell's book-in-progress and its content has been edited for this newsletter. The book is an introductory collection of Core War warriors with commentary.]

On October 14, 1992, Anders Ivner posted a warrior that revolutionized the game of Core War. "The IMPire Strikes Back" scored about 170 on the original King-Of-The-Hill and only suffered 10% losses, putting it firmly in first place. A. Ivner had invented a way to kill other programs with imps — the world's first imp-ring.

D. Nabutovsky improved the launch code a bit by making an imp-spiral and adding a stone in his "Impressive", which lost only 2% and scored 195 when it started on the hill. Since that time, most warriors on the hill have either been imps or something hostile to imps.

```
; Program #1
; Name: Wait
; Speed: None
; Size: 1
; Durability: Strong
; Effectiveness: None
; Score:

wait   JMP    wait
       END    wait
```

Wait is the simplest warrior. Its small size makes it difficult to locate. However, it has no attack, so it only wins if the enemy program self-destructs. We shall be using this program for fodder.

```
; Program #2
; Name: Imp
; Author: A.K.Dewdney
; Speed: 100% of c
; Size: 1
; Durability: Strong
; Effectiveness: Poor
; Score:

imp    MOV    imp,   imp+1
       END    imp
```

Imp presents the enemy with a small, moving target that will not die without a direct hit. It often ties, and is vulnerable to the imp-gate (see Program #3).

When Imp is loaded and before it executes, it looks like this:

```
MOV 0, 1 (1)
```

(The (1) shows where the first process will execute. When we talk about multi-process warriors, we shall use (2), (3) and so on for the other processes.) When process (1) executes, it first copies its instruction to the next address and then moves to the next instruction:

```
MOV 0, 1      ; The original.
MOV 0, 1 (1)  ; The copy.
```

Process (1) now executes again. Since all addressing is relative, the process copies its instruction to the next address and advances.

```
MOV 0, 1
MOV 0, 1
MOV 0, 1 (1)  ; The 2nd copy.
```

And so it goes, overwriting anything in its path with `MOV 0, 1` instructions. So when it encounters enemy code, it replaces the enemy code with `MOV 0, 1` instructions, turning the enemy processes into imps. Note that although the enemy code is gone, the enemy processes live on, so imps do not win unless the enemy code self-destructs.

```
; Program #3
; Name: Imp Gate
; Speed: None
; Size: 1
; Durability: Strong
; Effectiveness: Excellent
; against imps, extremely
; poor against others.
; Score:

gate   EQU    wait-10
wait   JMP    wait, <gate
       END    wait
```

Imp Gate waits and destroys imps that happen to pass ten instructions before it. It is seldom overrun by imps and its small size makes it difficult to locate. The imp gate is defensive by nature, and will not win against a stationary enemy unless this enemy self-destructs.

The process running at `wait` jumps to the A-value of this command, *i.e.* back to `wait`. However, it also decrements the B-field of `gate`. Thus, the B-field of `gate` is decremented every turn. When an enemy imp comes by this is what happens:

```
MOV 0,  1 (1); imp
DAT 0, -5     ; gate
```

The imp copies itself and advances onto the gate:

```
MOV 0, 1
MOV 0, 1 (1)  ; gate
```

The gate decrements:

```
MOV 0, 1
MOV 0, 0 (1)  ; gate
```

The imp copies this instruction to itself (effectively doing nothing) and advances, falling off the end:

```
MOV 0, 1
MOV 0, 0      ; gate
DAT 0, 0 (1)
```

The gate decrements again (but the damage is done):

```
MOV 0,  1
MOV 0, -1     ; gate
DAT 0,  0 (1)
```

The enemy process executes an illegal instruction and dies.

```
; Program #4
; Name: Worm
; Speed: 25% of c
; Size: 1.75
; Durability: Very Strong
; Effectiveness: Poor
; Score:

launch SPL b
       SPL ab
aa     JMP imp
ab     JMP imp+1
b      SPL bb
ba     JMP imp+2
bb     JMP imp+3
imp    MOV imp,  imp+1
       END launch
```

Worm is a symbiotic collection of imps. The only vulnerable parts of the worm are the tail instruction and the instruction about to execute, hence the effective size of 1.75 (25% of the time, the tail instruction is the instruction about to execute). It is very difficult to kill, because each imp must be disposed of individually. However, it is still vulnerable to imp gates. As with Imp, Worm overwrites enemy code but preserves enemy processes.

First, we launch the worm using a binary launch:

```
SPL 4, 0 (1)
SPL 2, 0
JMP 5, 0
JMP 5, 0
SPL 2, 0
JMP 4, 0
JMP 4, 0
MOV 0, 1
```

After the first split:

```
SPL 4, 0
SPL 2, 0 (1)
JMP 5, 0
JMP 5, 0
SPL 2, 0 (2)
JMP 4, 0
JMP 4, 0
MOV 0, 1
```

Process (2) is skipped and process (1) splits again:

```
SPL 4, 0
SPL 2, 0
JMP 5, 0 (1)
JMP 5, 0 (2)
SPL 2, 0 (3)  ;used to be (2)
JMP 4, 0
JMP 4, 0
MOV 0, 1
```

Process (2) is skipped and process (3) splits:

```
SPL 4, 0
SPL 2, 0
JMP 5, 0 (1)
JMP 5, 0 (2)
SPL 2, 0
JMP 4, 0 (3)
JMP 4, 0 (4)
MOV 0, 1
```

Process (4) is skipped and process (1) jumps:

```
SPL 4, 0
SPL 2, 0
JMP 5, 0
JMP 5, 0 (2)
SPL 2, 0
JMP 4, 0 (3)
JMP 4, 0 (4)
MOV 0, 1 (1)
```

Processes (2), (3) and (4) jump:

```
SPL 4, 0
SPL 2, 0
JMP 5, 0
JMP 5, 0
SPL 2, 0
JMP 4, 0
JMP 4, 0
MOV 0, 1 (1)
DAT 0, 0 (2)
DAT 0, 0 (3)
DAT 0, 0 (4)
```

The worm will now start crawling though memory. Note that if processes (2), (3) or (4) executed right now, they would execute an illegal instruction and die. But process (1) executes, copying the MOV instruction to where process (2) is going to execute:

```
SPL 4, 0
SPL 2, 0
JMP 5, 0
JMP 5, 0
SPL 2, 0
JMP 4, 0
JMP 4, 0
MOV 0, 1
MOV 0, 1 (1) (2)
DAT 0, 0 (3)
DAT 0, 0 (4)
```

Now process (2) executes, copying the MOV instruction to process (3).

```
SPL 4, 0
SPL 2, 0
JMP 5, 0
JMP 5, 0
SPL 2, 0
JMP 4, 0
JMP 4, 0
MOV 0, 1
MOV 0, 1 (1)
MOV 0, 1 (2) (3)
DAT 0, 0 (4)
```

And after (3) and (4) have executed, the worm has crawled forward an instruction, leaving a slimy MOV 0,1 trail behind.

```
SPL 4, 0
SPL 2, 0
JMP 5, 0
JMP 5, 0
SPL 2, 0
JMP 4, 0
JMP 4, 0
MOV 0, 1
MOV 0, 1 (1)
MOV 0, 1 (2)
MOV 0, 1 (3)
MOV 0, 1 (4)
```

```
; Program #5
; Name: Ring
; Speed: 100% of c
; Size: 1
; Durability: Average
; Effectiveness: Fair

c      JMP imp-2666
launch SPL c
       SPL imp+2667
imp    MOV 0,       2667
       END launch
```

[To be continued. Next Issue: Rings]

RoboWar
A Macintosh Combat Game

by Jeff Lewis

RoboWar is a Macintosh shareware game written by David Harris. Features of RoboWar include animated combat, sound effects, color graphics, and a complete robot development system. Robots are programmed using a unique language (RoboTalk) based on Reverse Polish Notation. An assembler, interpreter, and debugger are included along with an icon editor and recording studio.

RoboWar is up to version 3.1 and RoboWar tournaments abound. The most prestigious tournaments are those run biannually by David Harris and friends. These include entry fees and prize money for the best robots, best robot team, and best icons.

The first of these RoboWar tournaments was held June 1, 1990 and was won by Matt Sakai's robot "Silo Plus". Silo Plus would skirt around the edge of the arena (a "wall-hugger") launching fast attacks against any robots it would bump into. In mid-battle, Silo Plus changes to a more central movement pattern, throwing out potent but imprecise missiles.

The second RoboWar tournament was won by Jon Newman's "Rammer AMT" (yes, this is the Jon Newman who is current ICWS director). Rammer AMT bounces around the screen in a careful search pattern, attacking any wall-huggers spotted but mainly trying to take out enemies in collisions.

New "hiding" robots like Tom Morrison's "Chicken & Corn II" appeared in the second tournament with the strategy of hiding in corners and evading any approaching fire. Evading robots have done very well in tournaments; Doug Harris's "Pacifist Penguin III" won the fourth RoboWar tournament.

Possibly the ultimate in dodging robots is Jeff Rommereide's "Excelsior" which won the fifth RoboWar tournament. Excelsior is an awesome robot to watch in action; it moves to the center of the arena and smoothly sidesteps any approaching shots. Excelsior's weakness is in group combat (where six robots battle in the small arena

simultaneously) where it is often impossible to avoid the dense amount of fire. Excelsior's defensive agility comes at a price of little defensive shielding.

Robots by Robert Hogg, Eric Foley, and Tim Seufert also deserve recognition for their accomplishments in tournaments. Hogg absolutely dominated the third RoboWar tournament and came close to winning the fourth and fifth tournaments as well.

January of 1994 the seventh RoboWar tournament was held with 35 robots entered. Paul Schmidt had the winning entry in a tournament featuring many impressive robots. Stephen Linhart's robots, "Darling" in the sixth tournament and "Sweetums" in the seventh tournament, have unleashed a new strategy that I call the "hunting" robot. They will actually chase after other robots and deliver a death blow, often before the victim even spots its attacker.

The hunting robots seem to be most effective in team contests. Andrew Lindsey's robot "His All Seeing Eye III" is a very nimble hiding robot during group battles that becomes an extremely effective hunting robot when only one opponent is present. It tied for second in the seventh RoboWar tournament - Paul Schmidt's robots "Vortex v2.2" and "Cloak Folk's Revenge" took first and second.

My choice for the best robot in the seventh tournament (and maybe ever) is Mike Fleming's "Mjolnir". It moves into a corner and sprays explosive bullets with good accuracy, and it is surprisingly nimble in dodging opponent's shots.

RoboWar is available from the major FTP archive sites, and all of the information to get started can be found in the application. RoboWar applications come slightly crippled, which can be remedied by sending in a $10 shareware registration fee.

~~~~~~~~~~~~~~~~~~~~~~~~

For further information, contact:

David Harris
RoboWar HindQuarters
1112 Evelyn Ct.
Ridgecrest, CA 93555

~~~~~~~~~~~~~~~~~~~~~~~~

How Can You Train Warriors?

by Adam Ryba

[Editor's Note: Translated from the original Polish.]

When you want to create a good warrior, you must have an equally good idea (warrior's strategy). But even the best idea is not enough because you can not just write a warrior, you must also train it.

There are many different methods of training and I want to describe a method that I use. This kind of method is trial and error.

First I write a program which simply realizes my idea, and I do not pay great importance to the quickness and the length of the code. After that, I divide the warrior into parts that I can improve independently. In this way, I do not need to improve the whole program, only to bring to perfection the separate parts.

This is much easier. One part could be responsible for throwing bombs, one part could be responsible for moving the warrior, etc.

Next I work out the many variations of each part of the program. Then I put each variation into the warrior's code and verify the efficiency of the warrior, in order to determine the best variation. I try to check as many variations as possible; even those which are seemingly non-sensical.

When I have improved each part, the warrior is finished. If it is not as good as I want, I go back to improving the parts of the code or work out a new strategy.

When I want to determine the efficiency of a program, I fight tens of battles (*e.g.* twenty) with each warrior from my fighting group. The program's score is directly proportional to its efficiency.

The best warriors that I have enter into the composition of my fighting group. My fighting group is composed of ten to fifteen warriors, and this composition is fairly constant.

If you are patient and you have many good ideas, you can train great warriors using this method.

The MADgic Corner

First, this news from the ICWS:

Adam Ryba has announced the formation of the Polish Branch Section of the ICWS. The Polish Branch Section of the ICWS will hold a yearly tournament and send ten winners to the ICWS tournament. Send inquiries to:

Adam Ryba
Os. XXX-lecia 6/2
47-303 KRAPKOWICE
POLAND

Next, my usual apologies for the length of time between newsletters. Much has gone on over the last several months, both for me personally and for the ICWS. Jon and I went back and forth all summer saying "Hold the presses!" as things continued to develop. Please note that although this issue ot *TCWN* is Volume 7, Number 4; there were no issues Number 2 and Number 3 in Volume 7.

I wanted very much for this year of Core War to be extra special because of it being the year of the tenth anniversary of Core War. I started the ball rolling two years ago with the push for a new standard. I had hoped that the standard would come to a vote and be approved this year. That seems unlikely now, due to our need to emphasize choosing a new Director.

I had also hoped for a return to the Boston Computer Museum for our annual tournament. Perhaps we will be able to return in 1996 for the tenth anniversary of our first tournament, which was held there in 1986.

I hope you will forgive me while I reminisce over the last decade. The number of things which have changed these last ten years, and the ways in which they have changed, seem too incredible to believe.

I remember when I thought 64 Kilobytes of RAM was more than I would ever need. (It was Core War, ten years ago, which convinced me otherwise). Now, 64 Megabytes is not unheard of. I used to get by at a speed of 1 MHz. Today, 100 MHz is tops for CISC and slow for RISC.

In my opinion, nothing short of global disaster can top the experience of the end of the Cold War, the dissolution of

the Soviet Union, the toppling of the Berlin Wall and the re-unification of Germany. It was so very unexpected (at least by me).

There were personal losses along the way. Our first Director, Mark Clarkson, lost his house to a fire. (He seems to be doing well now. You can often read his work in *Byte* magazine.) Our next Director and Founding Editor of *TCWN*, William Buckley, lost a dear friend and mentor. I most recently lost my father to cancer.

Ten years ago, all sorts of computers and operating systems were battling to be the next big thing. Today, some of the players have changed, some have gone (R.I.P. Commodore), and the partners have shifted, but we are about to undergo the same battles, even though PC-compatible appears to be the hardware equivalent of FORTRAN.

In the Core War community, I have had the honor of communicating with literally hundreds of enthusiasts. People come and people go, but everyone has been a joy to get to know, even if just a little bit. My only regret is that I have yet to meet another ICWS member face-to-face. I almost made it to the first ICWS tournament in Boston to meet Mark Clarkson and A. K. Dewdney, (I even had my plane ticket), but the date was changed and I had an irreconcilable conflict. To this day, I wish I had made it there. (Maybe my warriors would have assembled had I been there).

We have seen one book chapter, two standards, three *Scientific American* columns, three Directors, eight tournaments, ten years, hundreds of authors, and thousands of warriors. Here is to another ten years!

~~~~~~~~~~~~~~~~~~~~~~~~

Letter From The Director
Continued from Page 2

progress on projects such as the 94 Standard and the Big Disk of Core War will step forward as candidates for Director. Even members who want to work on specific projects, but not to take on the full burden of the Directorship, should use this opportunity to introduce themselves to the new leadership and to the Core War community.
~~~~~~~~~~~~~~~~~~~~~~~~

The Core War Newsletter

Mark A. Durham
Editor

AMRAN
Founding Publisher

William R. Buckley
Founding Editor

The Core War Newsletter, the official newsletter of the International Core War Society since 1987, is published quarterly by the International Core War Society for Core War enthusiasts everywhere.

The entire contents of this issue of *The Core War Newsletter* are Copyright ©1994 International Core War Society. Do not reproduce or distribute without the express permission of the ICWS. Subscription is solely through ICWS membership. Current annual fees are $15.00 drawn on a US bank.

International Core War Society

Jon Newman
Director

Send all correspondence, including renewals, changes of address, and back issue requests to:

ICWS
c/o Jon Newman
13824 NE 87th Street
Redmond WA 98052-1959 USA
email: jonn@microsoft.com

Submissions of articles, letters, comments, suggestions, etc. to:

TCWN
c/o Mark A. Durham
18 Honeysuckle Terrace
Spartanburg SC 29307-3760 USA
phone: (803)579-0431
email: durhamm@delphi.com