THE CORE WAR *NEWSLETTER*

The Official Newsletter of the International Core War Society

Features

From Proteus to Smooth Noodle Map 6, Part II **2**
by Matthew Hastings

State of the Society Address **7**
by Jon Newman

Articles

ICWS Tournament '92 Results **10**
by Jon Newman

Columns

The MADgic Corner **13**

Quarterly Challenge **13**

Volume 6, Number 1 Winter 1993 An ICWS Publication

From Proteus to Smooth Noodle Map 6, Part II

by Matthew Hastings

[Editor's Note: This is Part Two of Two. Part One appeared last issue. At the end of Part One, Smooth Noodle Map version 4.1 was at the top of the hill on KotH, the ongoing tournament server. Mr. Hastings may be reached via email at hastings-matthew@yale.edu*]*

At this time, a group of programs called variously Ike, Backfire, and others began bombing both locations and the locations pointed to by those locations. This gave them an advantage against programs which copied code, since usually many of the instructions' B-fields were left pointing at the destination.

The launch for Smooth Noodle Map (and a lot of other people's programs) was modified by using indirection on the launch so that only one location would point at useful code.

Smooth Noodle Map version 4.1 stayed on the hill until it was removed, after surviving over 400 successful challenges. Over 400 other programs had been knocked off the hill in the meantime. Add to that Smooth Noodle Map4's 100+ challenges and the program looks very old. When it had drifted down to number 12 without any sign of returning to its high position, I wrote Smooth Noodle Map 6 (after many test versions going under the name Smooth Noodle Map 5).

The essential difference between Smooth Noodle Map 6 and earlier versions is simply that Smooth Noodle Map 6 is `SPL 0` protected so that it tends to have very good results against "Stone"s and other programs that would do better against Smooth Noodle Map4.

The `SPL 0` protection creates a major change in the exit from the main loop since only one of the processes can leave. This, however, is not too difficult to understand, especially given the Redcode at the end. For this reason I will mostly discuss some minor features.

Smooth Noodle Map 6 contains many features developed over several versions. These include the decoys, the positioning of the copied code, the bombing density, the step value, the forward core-clear, the number and nature of the replicators, and the timing. I'll discuss

these, then give its results against other programs on the hill.

The decoys at the end of the main code are all given B-field = 1 so that any DJN mutating programs will keep dropping out of loop. To illustrate, a typical main loop for a slaver/bomber has become:

```
      SPL 0
      MOV bomb,    @bomb
      ADD inc,     bomb
      DJN -2,      <-2000
; Helps mutate core to gain
; extra offense and waste
; scanners' time.
```

The unity B-fields make this slow down the program.

The positioning of the code is also important. Here enters something I call probability correlation. The chance of a piece of code being hit changes if it is known whether or not another piece of code has been hit.

Two code segments located close by will tend to both be hit at once. Since one purpose of the multiple code segments of this program is to increase redundancy, they must be located far apart.

Similarly, it makes sense to place the decoys a few hundred instructions beyond the main loop. This way, if a DJN mutation stream is about to hit you, it will likely have gone through all of the decoys first, and so things will not be so bad.

Of course, one may decide that the advantage of slowing the opponent down is useless if the DJN stream is about to hit you, since you will likely lose anyway. Therefore, put the decoys far away.

Smooth Noodle Map 6 will probably *not* lose however. Slowing the opponent down gives it time to finish its bombing and start the replicator.

Bombing density is critical. It used to be, in previous versions, closely related to timing. The time to stay in the main loop was inversely proportional to the bombing density. This version can leave the main loop before it is done bombing because of the multiple processes in the main loop.

This, of course, would not work with a 64 process count

maximum. The mod 8 bombing with evenly interspersed mutation is more even than anything else after 3000 cycles. Further, the changes to core at mod 4 suffice to disable almost anything.

The step value has remained relatively constant in later versions. It is a computer optimized mod 8 step, given the criterion that after 3000 cycles one should maximize the sum, over each bomb dropped, of the distance between that bomb and the nearest bomb dropped at the time. I believe Nandor Sieben first suggested this criterion.

This does not take into account the mutational bombing, since that is less effective. Neither does it take into account the few non-bombed locations due to the `ADD` being executed twice in a row.

It gives a sufficiently even step. It is also large enough that the value can be re-used in the `SPL` instruction. It goes to the replicator by `SPL -2584,2584`.

Some other optimized steps are mod 2:1226 and mod 4:948. Optimized steps when a small step value is desired are mod 2:34 (originally used in Sucker), mod 4:44, and mod 5:155.

A small step value would be necessary if the step were to be re-used in a core-clear of the form `MOV @34, <-34`. This idea is also from Sucker.

Paul Kilroy first used a forward core-clear. My code is nothing like his, but it does have the same feature of clearing forward. The purpose of this is: when fighting a program which makes a large number of 2-3 line bombers, each clearing downwards through core, the forward clear will often help kill before you are killed.

One way around this for these programs is for each bomber to start clearing a considerable distance down-core from itself. That way the bomber cleared in the forward clear will not be the one about to kill the forward-clearer. However, this alternate technique is, I believe, more difficult to get working well, and also is more vulnerable to normal backwards core-clears, which are much more prevalent.

Smooth Noodle Map 6 actually gets speed of light (one

instruction/cycle) mutation, although only every 3rd location is hit. Plus there is a 1/3 speed of light clear with DATs. The mutation wraps around, hits the main loop and starts a reverse addition core-clear. This clears by simply adding constants to other instructions. The first instance of this on KotH was Andy Pierce's Twill, but I can claim to have separately invented this idea.

Actually, it would be better to just continue the DAT clearing, but I could not figure out a good way to do this. This current core-clear is very short at least and involves no modifications to the main loop.

The number of replicators is back to two. Since there are many processes still running in the main loop, if one replicator is vampiric captured, this will probably not matter. Further, the vampiric pit may be wiped out by the forward core-clear, which will go much faster than the captured processes.

Also, the 2nd replicator is useful since, with so many processes not replicating and only one replicator, the program would not replicate much and the core-clear starts to get lots of the replicators. Of further interest is that the length of the replicators is 7, a prime number.

This way, unless an opposing program overwritten has exactly 6 processes, all one same location, the replicator will die, taking the enemy with it. With the old value of 6, in one or two test battles, the enemy would have one process and execute side by side with the main process. Wins became ties.

The extra DAT 0, 0 helps deadliness and also helps in mysterious ways when multiple replicators are produced near each other. The extra zero B-field helps when the JMZ are sometimes damaged.

The timing is also critical. The original constant to determine time in loop was given by DJN -2, #900. This meant that the replicators got very little time and often were too late. The program had poor results.

To speed up the replicators, I tried adjusting them to overwrite the main loop and kill off those processes, but in that event a portion of core

would often not be cleared. By reducing the constant sharply, it was possible to give a decent time slice to the replicators while still executing the core-clear.

If there is one thing I have learned from this program, it is that one must successfully kill the enemy. The whole core must be cleared. The replicators must guarantee death to an overwritten opponent.

Check out the `MOV @1, -3` instruction. In fact, the `MOV @` is an idea of Andy Pierce. The theory is that if no one has bombed the next instruction, it will point to itself and be a `DAT`. If bombed with a non-deadly instruction such as `SPL 0`, it may be changed to point to another location.

Actually better would be `MOV @30`. `MOV @1` has the disadvantage that if the next instruction is mutated it will point to a known non-deadly instruction. This has very recently been changed.

Results are very good. Against replicators the program goes about 60-40-0 (wins-ties-losses). It usually wins 60% or more against scanners. It does reasonably well against Return of the Living Dead types, with an advantage against most such. It beats most Stone types. It is now about 4-7 on KotH.

One last variant should be mentioned, Trinity, which started with a Stone (in fact, 2 Stones in a latter version) and `SPL` to a core-clear and a slaver. It could be replicators. At first it had a slight edge against almost all programs. Later it had a slight disadvantage against most programs as technology developed.

Code for Smooth Noodle Map 6 follows. Code for Proteus, Smooth Noodle Map, and Smooth Noodle Map 4.1 appeared in the previous article.

```
; Smooth Noodle Map 6:
      MOV s5,     <ptr1
start MOV s4,     <ptr1
arb   MOV s3,     <ptr1
      MOV s2,     <ptr1
      MOV s1,     <ptr1
      MOV s0,     <ptr1
      MOV s7,     start-307
      MOV r4,     <ptr2
      MOV r3,     <ptr2
      MOV r2,     <ptr2
      MOV r1,     <ptr2
      MOV ro,     <ptr2
      MOV rm,     <ptr2
```

Continued on page 9

State of the Society Address

by Jon Newman, Director

Financial Status:
The International Core War Society has a current account balance of US$ –5.39. ICWS assets such as postage and other materials leave the organization slightly ahead. This does not include the liability to send future issues of *TCWN* to current members. *[Editor's Note: Fulfilling paid current subscription liabilities will cost ~$50 at current postage and supply rates].* There are seven additional unpaid *TCWN* subscriptions, four to Branch Sections and three to Emeritus members. Free or reduced rate memberships are available to persons in disadvantaged nations, but so far no one has asked.

Membership Status:
The ICWS has 38 paying members (including Eugene Lilitko, whose contest prize was applied toward a 10-year paid membership). There are three active Branch Sections: the Argentine Branch Section, the Electronic Branch Section, and the Spanish Branch Section. Alexander Poroshin is attempting to rebuild the CIS Branch Section (formerly the Soviet Branch Section) after the recent developments there.

This membership count is historically low and a disappointing total. With this TCWN however, the last of the memberships from the William Buckley period expire; many of these members lost interest in Core War while the ICWS was inactive. It should also be noted that the Electronic Branch Section has made Core War competition and many ICWS materials available without ICWS membership, so overall interest in Core War may not be declining as much as these figures indicate.

Activity Status:
I will now review the promises and proposals in my article in the Spring 1992 *TCWN*, "A Letter From the New Director."

Promises:
• All ICWS memberships from the Buckley period have been honored; the last of these memberships (except for Eugene Lilitko's) expires with this TCWN.

• *The Core War Newsletter* has appeared once every three months. We have not had to resort to the "four-page

TCWN" I threatened in that article, but our archives are looking thin and we are definitely open to any new submissions reasonably related to Core War.

• The ICWS has held two ICWS tournaments during my tenure. The 1992 Tournament was a stopgap affair including only entrants from a single Branch Section; the 1992 Tournament was more successful, with entrants from two Branch Sections and several individual members.

• Only three Branch Sections appear to have survived the interregnum; of these, the Electronic Branch Section is the most active.

Proposals:

• Core War Simulator Certification: We have made little progress toward this goal, which is largely my own fault.

• Electronic Data Transfer: The Electronic Branch Section makes many of these materials available across the Internet. As part of the Society's new back issue policy, past Tournament rosters are now available by mail on IBM and Macintosh format floppy disks.

• New Core War Standards and Extensions: We have made little progress toward this goal, although the Electronic Branch Section is experimenting with several extensions.

• Other Programming Games: We have made little progress toward this goal, although the Summer 1992 issue of *TCWN* did contain information about the artificial life system Tierra and also the game C-Robots.

Evaluation:

The International Core War Society is just now emerging from a difficult leadership transition. This transition has drained away all but the most dedicated Core War gamers.

At the same time, a new center of Core War activity has developed on the largest global computer network, the Internet. Unfortunately, relatively few people have access to the Internet, especially outside the United States and a few other industrialized nations. Much of the interest in Core War has traditionally come from outside the high-tech community, and it would be a great loss if Core War became an activity solely for the technological elite.

The overall goal and purpose of the ICWS is to foster Core War competition and development. Changes in communication technology offer great potential to remove some of the headaches and bottlenecks from Core War gaming, while also threatening to divide Core War gamers into the well-connected "haves" on the computer networks, and the disadvantaged "have-nots" communicating by telephone and postal mail. Therefore, the International Core War Society must make it a primary task to coordinate these two communities, and to bring the advantages of electronic communication to all Core War gamers.

Back Issue Policy

Starting now, recent *TCWN* back issues are available from the ICWS. In addition, computer disks containing the complete roster of Tournament entries for every contest since 1986 can be ordered for the same price. Each item costs US$3 (US$4 for non-ICWS members) and should be paid in the same form as ICWS memberships.

Tournament rosters are available on 5.25" (double-density, 720K) IBM format and 3.5" (double-density, 800K) Macintosh format disks.

Send your orders with appropriate payment to Jon Newman, 13824 NE 87th Street, Redmond WA 98052-1959 USA. Please make checks out to "Jon Newman", and please remember that for non-US orders, US currency is more reliable than non-US checks, postal money orders, etc.

Continued from Page 6

```
      MOV r4a,     <ptr3
      MOV r3,      <ptr3
      MOV r2a,     <ptr3
      MOV r1,      <ptr3
      MOV ro,      <ptr3
      MOV rm,      <ptr3
      JMP start-314-1, <ptr1
s0    SPL 0,       2
s1    ADD 3,       1
s2    MOV <-2586,  2586
s3    DJN -2,      #477
s4    SPL -2584,   2584
s5    JMP -4807+440+4, 0
s7    DAT #-4,     #-28
rm    MOV #7,      7
ro    MOV <-1,     <2
r1    JMN -1,      -2
r2    SPL @0,      1273
r3    JMZ rm,      rm
r4    MOV @1,      r1+1
r2a   SPL @0,      1833
r4a   MOV @1,      -3
ptr1  DAT #0, #start-307-2
ptr2  DAT #0,      #arb-2891
ptr3  DAT #0, #start-4668
; Lots of decoys here ....
```

ICWS Tournament '92 Results

by Jon Newman

The big news this quarter is the ICWS 1992 Tournament! Twenty-one programs were entered into the 1992 contest, including ten from the Electronic Branch Section, five from the Argentine Branch Section, and six from individual members. As reported earlier, the rules were: ICWS'88 Core War Standard, core size of 8192, maximum of 8000 processes per side, ties declared after 100 000 cycles, programs score 3 for wins, 1 for ties, and 0 for losses. These programs were divided into two qualifying rounds, each of which sent four programs to the finals. The preliminary rounds and the finals were held as round-robin tournaments, with each program facing each other program twice, each time at a random relative position.

The final results were very close, indicating the high overall quality of this year's entrants. In fact, two programs tied for first place, and two others tied for third. Special congratulations go to Anders Ivner of the EBS with two of the top four programs.

I will make use of the glossary terms defined in the Fall 1992 TCWN to describe these programs, and add one or two of my own invention. Here are the programs and brief descriptions.

Tied For First Place:

RotLD 2.2 ("Return of the Living Dead"): EBS entry by Nandor Sieben (anxs@asuvm.inre.asu.edu). Nandor describes his program as follows:

"RotLD is really a brute force bomber. It makes eight copies of a small three-line self-splitting bomber (these are the living deads) and starts them synchronized. A living dead has a great amount of life-force, since it can survive any B-field increments and many kinds of bombs, that is why I call it a living dead. It is good against bombers, and fairly good against `SPL 0` bombers because of the great amount of self-splitting. It is not so good against vampires, Imp rings, and paper-type programs."

Leprechaun 1b: EBS entry by Anders Ivner (d91andiv@und.ida.liu.se). Anders describes his program as "a combined bomber/scanner, with a main loop that achieves a total speed

of 67% of c (one SPL 0 bomb and one scanned location every three instructions). When a possible enemy is found, the location is bombed with the standard SPL 0 / JMP -1 bomb to slow it down before killing it off with a backward core-clear that ends in an Imp-stomping gate. The program starts by launching itself to a different location in memory, leaving behind a decoy of DAT #1s to slow down other scanners and DJN-using warriors."

Tied For Third Place:
Griffin 2: EBS entry, also from Anders Ivner. He describes this one as "a highly optimized CMP-scanner, with approximately half the instructions decrement protected (i.e. the program will function correctly even if the B-field is decremented), thus making it more resistant against bombers that use pre-decrement of DJN to bomb the core."

Twimp: EBS entry from Monika Keindl (agmok@acvax.inre.asu.edu). Twimp is an ingeniously constructed pattern bomber which attains a 66% speed (half bombs and half predecrements), seems to strike just about every location in core simultaneously, and only has an active footprint of three instructions.

Here are the programs, their scores, and their included comments, in order of finish.

All Finalists:

Leprechaun 1b (score 27)

RotLD 2.2 (score 27)

Griffin 2 (score 25)

Twimp (score 25)

Imprimis 8192 (score 19): EBS entry by P. Kline. This is the only one of the new-generation "Imp rings" to reach the finals, with a respectable fifth-place showing. It contains two three-point Imp spirals with a stone capable of beating other Imps (8 - 18% of the time).

Plasma (score 16): EBS entry by Wayne Sheppard (gt7804b@prism.geotech.edu). Plasma scans until it finds something non-zero. This is either the enemy program or a bomb. All locations in that area are bombed. Then Plasma goes into a replicator mode. This strategy allows some success against other scanners, beats all

bombers, and ties other replicators or Imps.

Antibody (score 14): EBS entry by Nandor Sieben. Off-axis CMP-scanner (scan width 3457) contains special code against Imp rings.

NS6 (score 2): ABS entry by Lorenzo Jorquera. NS6 follows a 57% speed round of SPL/JMP pattern bombing with a speed of 80% SPL/JMP core-clear, then finally a DAT core-clear. This was the only finalist not submitted by the EBS.

Qualifying Round A:
Griffin 2 (score 48)
RotLD 2.2 (score 38)
Plasma (score 36)
NS6 (score 31)

Impressive (score 26): EBS entry by Dan Nabutovsky (fedimit@wisipc.weizmann.ac.il). Two Imp-spirals plus a self-splitting Dwarf.

Scout (score 25): Independent entry by William Hamaker. CMP-scanner with scan-width 8, bombs with SPL 0, and follows up with a core-clear.

Buffy (score 24): Independent entry by Chris Lindensmith.

Broom2 (score 23): Independent entry by Mark A. Durham.

Roaches (score 23): EBS entry by Steven Morell (morell@math.utah.edu).

Troll (score 14): ABS entry by Pablo Destefanis.

IMPire (score 9): EBS entry by Anders Ivner.

Qualifying Round B:
Imprimis 8192 (score 38)
Twimp (score 37)
Antibody (score 33)
Leprechaun 1b (score 31)

Sauron (score 22): EBS entry by Steven Morrell. Nine-point Imp ring with SPL 0 bomber plus core-clear auxiliary.

EriTQ (score 20): ABS entry by Hernán G. Freedman.

Murder (score 15): ABS entry by Hernán G. Freedman.

Presidio (score 15): ABS entry by Fabio Friedlaender.

Exorcist (score 13): Independent entry by Chris Lindensmith.

Brooms (score 11): Independent entry by Mark A. Durham.

The MADgic Corner

Winter is a good time to stop and take a minute to reflect back on the past year's goals and accomplishments. It was with that idea that I suggested to Jon Newman to write a "State of the Society" address for this issue.

His article strikes a particular resonance with my life as I move from a "have" to a "have-not" in terms of electronic communication. The magnitude of this transition was made clear to me when I read the reference to another former netter as "posthumous", even though he is "dead" only to the net and not the rest of the world.

As I am unable to maintain a well-tended email address, I regret that I find it in the best interest of *The Core War Newsletter* to leave the post of Editor. I hope I will be able to return in the not-too-distant future, and I guarantee that my absence from the net does not mean that I am absent from Core War. On the contrary, I will try to increase my contributions elsewhere.

This is my sixth issue of *TCWN*. I hope you have found them all to be informative, entertaining, and timely.

Quarterly Challenge #8

The response to the Quarterly Challenges has been underwhelming, to say the least. If there are people out there who enjoy the column (as I do), I need to hear from them in order to continue.

Your challenges for this issue are: 1) Move an A-field to a B-field, and 2) Move a B-field to an A-field. These are two problems which often occur when writing Redcode programs.

The solution this issue is the answer to Problem #1 from QC#5.

```
; QC#5, Problem #1 Solution
; by Mark A. Durham
; for Core Size 8192
code  EQU top-to+1
size  EQU top-from
from  DAT          #top
start MOV #size,    from
      MOV <table,   to
wait  JMZ wait,    table
loop  MOV @from,   <to
      DJN  loop,    from
      SPL @to
      JMP  start
to    DAT          #4096+code
table DAT          #top+1
      DAT          #13+code
      DAT          #32+code
      DAT          #64+code
      DAT          #128+code
      DAT          #256+code
      DAT          #512+code
      DAT          #1024+code
      DAT          #2048+code
top   DAT          #4096+code
      END start
```

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 ©1993 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 and changes of address, 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. also go to the above address.

Mark A. Durham
18 Honeysuckle Terrace
Spartanburg SC 29307-3760 USA
phone: (803)579-0431
email: durham@ricevm1.rice.edu

Back issues of *TCWN* prior to 1992 are available from AMRAN, 5712 Kern Drive, Huntington Beach, CA 92649-4535, USA - email: xwbuckley@fullerton.edu.