

spacewar 4.0 2/2/63 ddp : pt. 1

```
3/      jmp sbf          / ignore seq. break
      jmp a40
      jmp a1             / use test word for control, not iot 11 co
```

/ interesting and often changed constants

/symb loc usual value (all instructions are executed,
/ and may be replaced by jda or jsp)

```
tno, 6,      law i 41      / number of torps + 1
tv1, 7,      sar 4s        / torpedo velocity
rlt, 10,     law i 20      / torpedo reload time
tlf, 11,     law i 140     / torpedo life
foo, 12,     -20000        / fuel supply
maa, 13,     10            / spaceship angular acceleration
sac, 14,     sar 4s        / spaceship acceleration
str, 15,     100           / star capture radius
me1, 16,     6000          / collision "radius"
me2, 17,     3000          / above/2
ddd, 20,     -0            / 0 to save space for ddt
the, 21,     sar 9s        / amount of torpedo space warpage
mhs, 22,     law i 10      / number of hyperspace shots
hd1, 23,     law i 40      / time in hyperspace before breakout
hd2, 24,     law i 100     / time in hyperspace breakout
hd3, 25,     law i 200     / time to recharge hyperfield generators
hr1, 26,     scl 9s        / scale on hyperspatial displacement
hr2, 27,     scl 4s        / scale on hyperspatially induced velocity
hur, 30,     40000         / hyperspatial uncertainty
ran, 31,     0             / random number
grv, 32,     sar 6s        / gravitational constant
```

/ place to build a private control word routine.
/ it should leave the control word in the io as follows.
/ high order 4 bits, rotate ccw, rotate cw, (both mean hyperspace)
/ fire rocket, and fire torpedo. Low order 4 bits, same for
/ other ship. Routine is entered by jsp cwg.

40/

```
cwr,      jmp mg1          / normally iot 11 control
. 20/     / space
```

/ routine to flush sequence breakes, if they occur.

```
sbf,      tyi
          lio 2
          lac 0
          lsm
          jmp i 1
```

```
          define
xincr X,Y,INS
          lac Y
          INS ssn
          dac Y
          lac X
          INS scn
          dac X
          term
```

```
          define
yincr X,Y,INS
          lac Y
          INS scn
          dac Y
          lac X
          -INS+add+sub ssn
          dac X
          terminate
```

```
          define
dispatch
          add (. 3
          dap . 1
          jmp .
          term
```

```
          define
dispt A,Y,B
          repeat 6      B=B+B
          lio Y
          dpy-A+B
          term
```

```
          define
scale A,B,C
          lac A
          sar B
          dac C
          term
```

```
define
diff V,S,SF
  add i V
  dac i V
  xct SF
  add i S
  dac i S
  term
```

```
define
random
  lac ran
  rar 1s
  xor (355670
  add (355670
  dac ran
  term
```

```
define
ranct S,SS,C
  random
  S
  SS
  sma
  cma
  dac C
  terminate
```

```

define
    varsft
    dzm  $\overline{xys}$ 
    dac  $\overline{t1}$ 
    idx  $\overline{xys}$ 
v2,   idx  $\overline{xys}$ 
    lac  $\overline{t1}$ 
    scr 2s
    dac  $\overline{t1}$ 
    sza
    jmp v2+R
    scr 2s
    swap
    terminate

define
    undosft
    dac t1
    dio  $\overline{t2}$ 
    lac  $\overline{xys}$ 
    add sft
    dap .+1
    lac .
    dac .+6
    dac .+6
    xor (10000 / change scr to scl or scl to scr.
    dac  $\overline{xyt}$ 
    lac  $\overline{t1}$ 
    dio  $\overline{t2}$ 
    scr .
    scr .
    terminate

define
    integrate A,B
    cli
    lac i A
    scr 9s
    scr 1s
    div  $\overline{t1}$ 
    hlt
    cma+cli-opr
    xct  $\overline{xyt}$ 
    xct grv
    dac B
    terminate

sft,   lac .-1
    scr 7s
    scr 6s
    scr 5s
    scr 4s
    scr 3s
    scr 2s
    scr 1s
    scr
    scl 1s

```

/sine-cosine subroutine·Adams associates
/calling sequence= number in AC, jda sin or jdacos.
/argument is between +2 pi, with binary point to right of bit 3.
/answer has binary point to right of bit 0. Time = 2.35-? ms.
/changed for auto-multiply , ddp 1/19/63

```
cos,      0
          dap csx
          lac (62210
          add cos
          dac sin
          jmp .+4

sin,      0
          dap csx
          lac sin
          spa
si1,      add (311040
          sub (62210
          sma
          jmp si2
          add (62210

si3,      ral 2s
          mul (242763
          dac sin
          mul sin
          dac cos
          mul (756103
          add (121312
          mul cos
          add (532511
          mul cos
          add (144417
          mul sin
          scl 3s
          dac cos
          xor sin
          sma
          jmp csx-1
          lac (377777
          lio sin
          spi
          cma
          jmp csx

          lac cos
csx,      jmp .

si2,      cma
          add (62210
          sma
          jmp si3
          add (62210
          spa
          jmp .+3
          sub (62210
          jmp si3

          sub (62210
          jmp si1
```

```

/integer square root
/input in ac, binary point to right of bit 17, jda sqt
/answer in ac with binary point between bits 8 and 9
/largest input number = 177777

```

```

sqt,      0
          dap sqx
          law i 23
          dac sq1
          dzm sq2
          lio sqt
          dzm sqt

sq3,      isp sq1
          jmp .+3
          lac sq2
sqx,      jmp .

          lac sq2
          sal 1s
          dac sq2
          lac sqt
          rcl 2s
          sza i
          jmp sq3
          dac sqt
          lac sq2
          sal 1s
          add (1
          sub sqt
          sma+sza-skp
          jmp sq3
          spa
          cma
          dac sqt
          idx sq2
          jmp sq3

sq1,      0
sq2,      0

```

```

/outline compiler
/ac=where to compile to, call jda oc
/ot=address of outline table

```

```

define
    plinst A
    lac A
    dac i oc
    idx oc
    terminate

```

```

define
    comtab A, B
    plinst A
    jsp ocs
    lac B
    jmp oce
    terminate

```

```

ocs,    dap ocx                /puts in swap
        dio i oc
        idx oc
        dio i oc
        idx oc
ocz,    jmp .

```

```

oc,      0                    /outline compiler proper
        dap ocx
        lac i ocx
        dap ocg
        plinst (stf 5
        dap ocm
        idx ocx

```

```

ock,    plinst (lac  $\overline{s}x1$ 
        plinst (lio  $\overline{s}y1$ 
        clf 6

```

```

ocj,    setup  $\overline{occ},6$ 
ocg,    lio .                /outline table
och,    cla

```

```

        rcl  $\overline{3}s$ 
        dio  $\overline{oci}$ 
        lio (rcl  $\overline{9}s$ 
        dispatch
        opr

```

```

        jmp oc1
oco,    jmp oc2
ocq,    jmp oc3
ocp,    jmp oc4
ocr,    jmp oc5
        jmp oc6

```

```

        plinst (szf 5          /7 code
        add (4
        dap ocn
        plinst ocn
        plinst (dac  $\overline{s}x1$ 
        plinst (dio  $\overline{s}y1$ 
        plinst (jmp sq6
        plinst (clf 5
        plinst (lac  $\overline{s}cm$ 
        plinst (cma
        plinst (dac  $\overline{s}cm$ 
        plinst (lac  $\overline{s}sm$ 
        plinst (cma
        plinst (dac  $\overline{s}sm$ 
        plinst (lac  $\overline{c}sm$ 
        plinst (lio  $\overline{s}sd$ 
        plinst (dac  $\overline{s}sd$ 
        plinst (dio  $\overline{c}sm$ 

        plinst (lac  $\overline{s}sc$ 
        plinst (lio  $\overline{c}sn$ 
        plinst (dac  $\overline{c}sn$ 
        plinst (dio  $\overline{s}sc$ 
        plinst ocm
ocx,    jmp .

ocm,    jmp .
ocn,    jmp .

oc1,    plinst (add  $\overline{s}sn$ 
        jsp ocs
        lac (sub  $\overline{s}cn$ 
oce,    dac i oc
        idx oc
        jsp ocs
        plinst (ioh
        lac (dpy-4000
ocd,    dac i oc
        idx oc
        lio  $\overline{oc}i$ 
        count  $\overline{oc}c$ , och
        idx  $\overline{oc}g$ 
        mp  $\overline{oc}j$ 

oc2,    comtab {add  $\overline{s}cm$ , {add  $\overline{s}sm$ 
oc3,    comtab {add  $\overline{s}sc$ , {sub  $\overline{c}sm$ 
oc4,    comtab {sub  $\overline{s}cm$ , {sub  $\overline{s}sm$ 
oc5,    comtab {add  $\overline{c}sn$ , {sub  $\overline{s}sd$ 
oc6,    szf 6
        jmp oc9
        stf 6
        plinst (dac  $\overline{s}sa$ 
        lac (dio  $\overline{s}si$ 
        jmp ocd
oc9,    clf 6
        plinst (lac  $\overline{s}sa$ 
        lac (lio  $\overline{s}si$ 
        jmp ocd

```

/display a star

define

starp
add bx
swap
add by
swap
ioh
dpy-4000
terminate

blp, dap blx /star

szs 60
jmp blx
random
rar 9s
and (add 340
spa
xor (377777
dac bx
lac ran
ral 4s
and (add 340
spa
xor (377777
dac by
jsp bpt
ioh

blx, jmp .

bpt, dap bpx
random
sar 9s
sar 5s
spa
cma
sal 3s
add (bds
dap bjm
cla cli clf 6-opr-opr
dpy-4000

bjm, jmp .

bds, repeat 20, starp
szf 6

bpx, jmp .
stf 6
cma
swap
cma
swap
jmp bjm

/background display • 3/13/62, prs.

```
define
dislis J, Q, B
  repeat 6, B=B+B
  clf 5
  lac flo+R
  dap fpo+R
fs,   dap fin+R
      dap fyn+R
      idx fyn+R

fin,   lac           /lac X
      sub fpr       /right margin
      sma
      jmp fgr+R
      add (2000
frr,   spq
fou,   jmp fuu+R
fie,   sub (1000
      sal 8s
fyn,   lio           /lio Y
      dpy-i+B
      stf 5
fid,   idx fyn+R
      sad (lio Q+2
      jmp flp+R
      sad fpo+R
      jmp fx+R
      dap fin+R
      idx fyn+R
      jmp fin+R

fgr,   add (-20000+2000
      jmp frr+R

fuu,   szf 5
fx,    jmp flo+R+1 /return
      idx flo+R
      idx flo+R
      sas (Q+2
      jmp fid+R
      law J
      dac flo+R
      jmp fid+R

flp,   lac (lio J
      sad fpo+R
      jmp fx+R
      dap fin+R
      law J+1
      dap fyn+R
      jmp fin+R

fpo,   lio
flo,   J
      terminate
```

```

        define
background      jsp bck
                 termin

bck,          dap bcx
              szs 40
              jmp bcx
              isp bcc
bcx,          jmp .
              law i 2
              dac bcc
              dislis 1j,1q,3
              dislis 2j,2q,2
              dislis 3j,3q,1
              dislis 4j,4q,0
              isp bkc
              jmp bcx
              law i 20
              dac bkc
              law i 1
              add fpr
              spa
              add (20000
              dac fpr
              jmp bcx

bcc,          0
bkc,          0
fpr,          10000

```

```

mul=mus
div=dis

```

```

start

```

spacewar 4.0 ddp 2/2/63 pt.2

/main control routine for spaceships

```
nob=30                                /total number of colliding objects

m10,  load mtc, -4000                /delay for loop
      init m11, mtb                  /loc of calc routines
      add (nob
      dap mx1                          / x
nx1=mtb nob
      add (nob
      dap my1                          / y
ny1=nx1 nob
      add (nob
      dap ma1                          / count for length of explosion or torp
na1=ny1 nob
      add (nob
      dap mb1                          / count of instructions taken by calc routine
nb1=na1 nob
      add (nob
      dac mdx                          / dx
ndx=nb1 nob
      add (nob
      dac mdy                          / dy
ndy=ndx nob
      add (nob
      dap mom                          /angular velocity
nom=ndy nob
      add (2
      dap mth                          / angle
nth=nom 2
      add (2
      dac mfu                          /fuel
nfu=nth 2
      add (2
      dac mtr                          / no torps remaining
ntr=nfu 2
      add (2
      dap mot                          / outline of spaceship
not=ntr 2
      add (2
      dap mco                          / old control word
nco=not 2
      add (2
      dac mh1
nh1=nco 2
      add (2
      dac mh2
nh2=nh1 2
      add (2
      dac mh3
nh3=nh2 2
      add (2
      dac mh4
nh4=nh3 2
nnn=nh4 2
```

```

law ss1
xor mtb
sza
jmp mdn
law ss2
xor mtb 1
sza
jmp mdn
law 1          / test if both ships out of torps
add ntr
spa
jmp md1
law 1
add ntr 1
spa i
jmp mdn
md1, xct tlf      / restart delay is 2X torpedo life
sal 1s
dac  $\overline{n}$ td
jmp m11

mdn, count  $\overline{n}$ td,m11
stf 1
stf 2
law ss1
xor mtb
sza
clf 1
sza i
idx  $\overline{1}$ sc
law ss2
xor mtb 1
sza
clf 2
sza i
idx  $\overline{2}$ sc
clf 2
jmp a

```

```

a1,      law mg2          / test word control
        dac cwg
        jmp a

a40,     law cwr          / here from start at 4
        dac cwg
        jmp a6

a,        lac gct
        sma
        jmp a5
        count gct, a5
        lac 1sc
        sas 2sc
        jmp a4
        law i 1
        dac gct

a5,      lat
        and (40
        sza i
        jmp a2

a4,      lac 1sc
        lio 2sc
        hlt
        lat
        and (40
        sza
        jmp a2
        dzm 1sc
        dzm 2sc

a6,      lat
        rar 6s
        and (37
        sza
        cma
        dac gct

a2,      clear mtb, nnn-1 / clear out all tables
        law ss1
        dac mtb
        law ss2
        dac mtb 1
        lac (200000
        dac nx1
        dac ny1
        cma
        dac nx1 1
        dac ny1 1
        lac (144420
        dac nth

```

```

law nnn          / start of outline program
dac not
lio ddd
spi i
jmp a3
jda oc          / compile outline
ot1
a3, dac not 1
jda oc
ot2
xct tno
dac ntr
dac ntr 1
lac foo
dac nfu
dac nfu+1
law 2000
dac nb1
dac nb1 1
xct mhs
dac nh2
dac nh2 1
jmp ml0

```

/ control word get routines

```
mg1,    dap mg3
        cli
        iot 11
mg3,    jmp .

mg2,    dap mg4
        lat
        swap
mg4,    jmp .

ml1,    lac .                / 1st control word
        sza i                / zero if not active
        jmp mq1              / not active
        swap
        idx  $\overline{m}$ oc
        spi
        jmp mq4
        law 1
        add ml1
        dap ml2
        law 1
        add mx1
        dap mx2
        law 1
        add my1
        dap my2
        law 1
        add ma1
        dap ma2
        law 1
        add mb1
        dap mb2
mot,    lac .
        dap sp5
ml2,    lac .                / 2nd control word
        spq                  / can it collide?
        jmp mq2              / no
mx1,    lac .                / calc if collision
mx2,    sub .                / delta x
        spa                  / take abs val
        cma
        dac  $\overline{m}$ t1
        sub me1              / < EPSILON ?
        sma
        jmp mq2              / no
my1,    lac .
my2,    sub .
        spa
        cma
        sub me1              / < epsilon ?
        sma
        jmp mq2              / no
        add  $\overline{m}$ t1
        sub me2
        sma
```

```

        jmp mq2
        lac (mex 400000 / yes, EXPLODE
        dac i m11      / replace calc routine with explosion
        dac i m12
        lac i mb1      / duration of explosion
mb2,    add .
        cma
        sar 8s
        add (1
ma1,    dac .
ma2,    dac .
mq2,    idx mx2          / end of comparison loop
        idx my2
        idx ma2
        idx mb2
        index m12, (lac mtb nob, m12

```

```

mq4,    lac i ml1      / routine for calculating spaceship
        dap . 1        / or other object and displaying it
        jsp .
mb1,    lac .          / alter count of number of instructions
        add mtc
        dac mtc
mq1,    idx mx1        / end of comparison and display loop
        idx my1
        idx ma1
        idx mb1
        idx mdx
        idx mdy
        idx mom
        idx mth
        idx mas
        idx mfu
        idx mtr
        idx mot
        idx mco
        idx mh1
        idx mh2
        idx mh3
        idx mh4
        index ml1, (lac mtb nob-1, ml1
        lac i ml1      / display and compute last point
        sza i          / if active
        jmp mq3
        dap . 1
        jsp .
        lac i mb1
        add mtc
        dac mtc
mq3,    background    / display stars of the heavens
        jsp blp        / display massive star
        count mtc, .  / use up rest of time of main loop
        jmp ml0        / repeat whole works

```

/ misc calculation routines

/ explosion

```
mex,    dap mxr
        lac i mdx
        sar 3s
        add i mx1
        dac i mx1
        lac i mdy
        sar 3s
        add i my1
        dac i my1
        law mst
        dap msh
        lac i mb1    / time involved
        cma cli-opr
        sar 3s
        dac mxc
ms1,    sub (140
        sma
        idx msh
mz1,    random
        and (777
        ior (scl
        dac ml1
        random
        scr 9s
        sir 9s
msh,    xct .
ml1,    hlt
        add i my1
        swap
        add i mx1
        dpy-i 300
        count mxc, mz1
        count i ma1, mxr
        dzm i ml1
mxr,    jmp .
mst,    scr 1s
        scr 3s
```

/ torpedo calc routine

```
tcr,    dap trc
        count i ma1, tc1
        lac (mex 400000
        dac i ml1
        law i 2
        dac i ma1
        jmp trc
tc1,    lac i mx1
        sar 9s
        xct the
        add i mdy
        dac i mdy
```

```
    sar 3s
    add i my1
    dac i my1
    sar 9s
    xct the
    add i mdx
    dac i mdx
    sar 3s
    add i mx1
    dac i mx1
    dispt i, i my1, 1
trc, jmp .
```

/ hyperspace routines

/ this routine handles a non-colliding ship invisibly
/ in hyperspace

```
hp1,    dap hp2
        count i ma1, hp2
        law hp3          / next step
        dac i m11
        law 7
        dac i mb1
        random
        scr 9s
        sir 9s
        xct hr1
        add i mx1
        dac i mx1
        swap
        add i my1
        dac i my1
        random
        scr 9s
        sir 9s
        xct hr2
        dac i mdy
        dio i mdx
        setup hpt,3
        lac ran
        dac i mth
hp4,    lac i mth
        sma
        sub (311040
        spa
        add (311040
        dac i mth
        count hpt, hp4
        xct hd2
        dac i ma1
hp2,    jmp .
```

/ this routine handles a ship breaking out of
/ hyperspace.

```
hp3,    dap hp5
        count i ma1, hp6
        lac i mh1
        dac i m11
        law 2000
        dac i mb1
        count i mh2, hp7
        dzm i mh2
```

```

hp7,    xct hd3
        dac i mh3
        lac i mh4
        add hur
        dac i mh4
        random
        ior (400000
        add i mh4
        spa
        jmp hp5
        lac (mex 400000
        dac i ml1
        law i 10
        dac i ma1
        law 2000
        dac i mb1
hp6,    lac i mx1
        dispt i, i my1, 2
hp5,    jmp .

```

/ spaceship calc

ss1, dap srt / first spaceship
 jzp i cwg
 dio scw
 jmp sr0

ss2, dap srt / second spaceship
 jzp i cwg
 rir 4s
 dio scw

sr0,

sc1, lio scw /control word
 clf 6 cla-opr /update angle

 spi
 add maa
 ril 1s
 spi
 sub maa

mom, add .
 dac i mom
 szs 10
 jmp sr8
 dzm i mom
 ral 7s

sr8, ril 1s
 spi
 stf 6
 lio i mfu
 spi i
 clf 6

mtb, add .
 sma
 sub (311040
 spa
 add (311040
 dac i mtb
 jda sin
 dac sn
 dzm bx
 dzm by
 szs 60
 jmp bsg
 lac i mx1
 dac t1
 mul t1
 scr 1s
 dac acx
 cla
 scr 2s
 dio fox
 lac i my1
 dac t1
 mul t1
 scr 1s
 dac acy

```

cla
scr 2s
swap
add  $\bar{t}_{ox}$ 
swap
scl 2s
add  $\bar{a}_{cx}$ 
add  $\bar{a}_{cy}$ 
sub str
sma i sza-skp
jmp pof
add str
varsft
dac  $\bar{t}_1$ 
jda sqt
mul  $\bar{t}_1$ 
undosft
scr 9s
scr 6s
szs i 20          / switch 2 for light star
scr 2s
sza
jmp bsg
scr 1s
dio  $\bar{t}_1$ 
integrate mx1,  $\bar{b}_x$ 
integrate my1,  $\bar{b}_y$ 
bsg,
cla
sad i mfu
clf 6
lac i mth
jda cos
dac  $\bar{c}_s$ 
sar 9s
xct sac
szf i 6
cla
add  $\bar{b}_y$ 
diff  $\bar{m}_{dy}$ , my1, (sar 3s
lac  $\bar{s}_n$ 
sar 9s
xct sac
cma
szf i 6
cla
add  $\bar{b}_x$ 
diff  $\bar{m}_{dx}$ , mx1, (sar 3s
sp1,
scale  $\bar{s}_n$ , 5s,  $\bar{s}_{sn}$ 
sp2,
scale  $\bar{c}_s$ , 5s,  $\bar{s}_{cn}$ 
lac i mx1

```

```

sub ssn
dac sx1
sub ssn
dac stx
lac i my1
add scn
dac sy1
add scn
dac sty
scale sn, 9s, ssn
scale cs, 9s, scn
lac ssn
dac ssm
add scn
dac ssc
dac ssd
lac ssn
sub scn
dac csn
cma
dac csm
lac scn
dac scm
cla cli-opr
dpy-4000
sp5, jmp .
sq6, ioh
      ranct sar 9s, sar 4s, src
      lio scw
      ril 2s
      spi i / not blasting
      jmp sq9 / no tail
sq7, scale sn, 8s, ssn
      scale cs, 8s, scn
      count i mfu, st2
      dzm i mfu
      jmp sq9
st2, yincr sx1, sy1, sub
      dispt i, sy1
      count src, sq7
sq9, count i ma1, sr5 / check if torp tube reloaded
      dzm i ma1 / prevent count around
mco, lac . / previous contro word
      cma
      szs i 30
      clc
      and scw / present control word
      ral 3s / torpedo bit to bit 0
      sma
      jmp sr5 / no launch
      count i mtr, st1 / check if torpedos exhausted
      dzm i mtr / prevent count around
      jmp sr5
st1, init sr1, mtb / search for unused object
sr1, lac .
      sza i / 0 if unused
      jmp sr2
      index sr1, (lac mtb nob, sr1
      hlt / no space for new objects
      jmp .-1

```

```

sr2,    lac (tcr                / set up torpedo calc
        dac i sr1
        law nob
        add sr1
        dap ss3
        lio stx
ss3,    dio .
        add (nob
        dap ss4
        lio sty
ss4,    dio .
        add (nob
        dap sr6
        add (nob
        dap sr7
        add (nob
        dap sr3
        add (nob
        dap sr4
        lac sn
        xct tvl
        cma
        add i mdx
sr3,    dac .
        lac cs
        xct tvl
        add i mdy
sr4,    dac .
        xct rlt
        dac i ma1      / permit torp tubes to cool
trf,    xct tlf        / life of torpedo
sr6,    dac .
        law 20
sr7,    dap .          / length of torp calc.
sr5,    count i mh3, st3 / hyperbutton active?
        dzm i mh3
        lac i mh2
        sza i
        jmp st3
        lac scw
        cma
        ior i mco
        and (600000
        sza
        jmp st3
        lac i ml1
        dac i mh1
        lac (hp1 400000
        dac i ml1
        xct hd1
        dac i ma1
        law 3
        dac i mb1
st3,    jmp .
st,

```

LAC T MCO
 LAC SCW
 DAC T MCO
 → JMP .

/ here to handle spaceships dragged into star

/ spaceship in star

```
pof,      dzm i mdx
           dzm i mdy
           szs 50
           jmp po1
           lac (377777
           dac i mx1
           dac i my1
           lac i mb1
           dac ssn
           count ssn, .
           jmp srt
```

```
po1,      lac (mex 400000    / now go bang
           dac i ml1
           law i 10
           dac i ma1
           jmp srt
```

/ outlines of spaceships

ot1, 111131
 111111
 111111
 111163
 311111
 146111
 111114
 700000

. 5/

ot2, 013113
 113111
 116313
 131111
 161151
 111633
 365114
 700000

. 5/

 constants
 variables
p, . 200/

/ space for patches

mtb,

/ table of objects and their properties

spacewar game saver patch

mg1=1510

ran=31

/+ tw → punch

/- tw → read

40/

cwr, dap cwx
 lat

cw2, cks
 ril 1s
 spi+spa i-skp
 jmp cw1
 rrb

 rpa-i
cw3, rir 4s
cwx, jmp .

cw1, ril 3s
 spi+sma i-skp
 jmp cw2
 jsp mg1
 ril 4s
 ppa-i
 jmp cw3

6000/

/new starting address

go, lat
 sma
 jmp pu
 rpb
 dio ran
 rpa-i
 jmp 4

pu, law i 200
 dac $\overline{pc}$
 cli
 ppa
 isp $\overline{pc}$
 jmp .-2
 lio ran
 ppb ril 6s
 ppb ril 6s
 ppb-i
 jmp 4

variables

start go

1j	6077
1q	6117
1sc	3151
2j	6121
2	6141
2sc	3152
3j	6143
3q	6403
4j	6405
4q	7747
a	1402
a1	1374
a2	1437
a3	1470
a4	1420
a40	1377
a5	1414
a6	1431
acx	3162
acy	3164
bcc	1857
bck	746
bcx	752
bds	534
bjm	533
bkc	1260
blp	466
blx	513
bpt	514
bpx	735
bsg	2317
bx	3135
by	3136
cos	100
cs	3165
csm	3127
csn	3132
csx	146
cwg	3153
cwr	40
ddd	20
fgr	32
fjd	22
fie	15
fin	6
flo	55
flp	45
foo	12
fou	14
fpo	54
fpr	1261
frr	13
fs	3
fuu	34

fx	35
fyn	17
gct	3154
grv	32
hd1	23
hd2	24
hd3	25
hp1	2001
hp2	2061
hp3	2062
hp4	2051
hp5	2123
hp6	2120
hp7	2074
hr1	26
hr2	27
hur	30
iox	3163
ma1	1610
ma2	1611
maa	13
mb1	1624
mb2	1604
mco	2466
md1	1346
mdn	1352
mdx	3140
mdy	3141
me1	16
me2	17
mex	1667
mfu	3142
mg1	1510
mg2	1514
mg3	1513
mg4	1520
mh1	3144
mh2	3145
mh3	3146
mh4	3147
mhs	22
mi1	1731
ml1	1521
ml2	1552
ml0	1262
moc	3155
mom	2143
mot	1550
mq1	1627
mq2	1612
mq3	1662
mq4	1621
ms1	1706
msh	1730

mst	1745
mt1	3156
mtb	3371
mtc	3137
mth	2157
mtr	3143
mx1	1555
mx2	1556
mxo	3157
mxr	1744
my1	1565
my2	1566
mz1	1711
na1	3501
nb1	3531
nco	3653
ndx	3561
ndy	3611
nfu	3645
nh1	3655
nh2	3657
nh3	3661
nh4	3663
nnn	3665
nob	30
nom	3641
not	3651
ntd	3150
nth	3643
ntr	3647
nx1	3421
ny1	3451
oc	230
oc1	375
oc2	420
oc3	426
oc4	434
oc5	442
oc6	450
oc9	460
occ	3123
ocd	411
oce	402
ocg	252
och	253
oci	3124
ocj	250
ock	241
ocm	373
ocn	374
oco	264
ocp	266
ocq	265
ocr	267

ocs	222
ocx	372
ocz	227
ot1	2626
ot2	2643
p	3171
po1	2621
pof	2605
ran	31
rlt	10
sac	14
sbf	61
sc1	2134
scm	3125
scn	3114
scw	3160
sft	66
si1	112
si2	147
si3	117
sin	106
sn	3161
sp1	2353
sp2	2356
sp5	2417
sq1	220
sq2	221
sq3	171
sq6	2420
sq7	2437
sq9	2463
sqt	162
sqx	174
sr1	2504
sr2	2514
sr3	2543
sr4	2547
sr5	2556
sr6	2553
sr7	2555
sr8	2151
src	3170
sr0	2134
srt	2604
ss1	2124
ss2	2130
ss3	2522
ss4	2526
ssa	3133
ssc	3131
ssd	3130
ssi	3134
ssm	3126
ssn	3113

st1	2502
st2	2451
st3	2602
str	15
stx	3166
sty	3167
sx1	3121
sy1	3122
t1	3116
t2	3117
tc1	1757
tcr	1747
the	21
tlf	11
tno	6
trc	2000
trf	2552
tv1	7
v2	3
xys	3115
xyt	3120