```
IDENTIFICATION DIVISION.
PROGRAM-ID.     Droids.
*
* DATE-WRITTEN. March 15, 1985.
* AUTHOR. Ken Richardson.
*
* To the glory of God.
*

ENVIRONMENT DIVISION.

CONFIGURATION SECTION.

SPECIAL-NAMES.
        symbolic        control-g       is 08
                        line-feed       is 11
                        carriage-return is 14
                        shift-out       is 15
                        shift-in        is 16
                        control-r       is 19
                        control-z       is 27
                        escape          is 28
        .

SOURCE-COMPUTER. VAX-11.
OBJECT-COMPUTER. VAX-11.

INPUT-OUTPUT SECTION.

FILE-CONTROL.
        select  score-file
                assign to       "ci$games:droids.dat"
                organization is sequential
        .

DATA DIVISION.

FILE SECTION.

FD      score-file
        .

01  score-record.
    02  score-score                             pic zzz,zz9.
    02  filler                                  pic x(02).
    02  score-date                              pic x(11).
    02  filler                                  pic x(02).
    02  score-username                          pic x(12).

WORKING-STORAGE SECTION.

01  constants.
    02  line-feed-char                          pic x value line-feed.
    02  carriage-return-char                    pic x value carriage-return.
    02  control-r-char                          pic x value control-r.
    02  control-z-char                          pic x value control-z.
    02  difficulty-increment                    pic s9(9) comp value 5.
    02  null-char                               pic x value low-values.
    02  room-pieces.
        03  dead-human                          pic x value "+".
        03  droid                               pic x value "A".
        03  dust                                pic x value "*".
        03  human                               pic x value "X".
    02  score-file-title                        pic x(34) value "======== Droids Champions ========".
    02  shift-out-character-set.
        03  filler                              pic x value escape.
        03  special-graphics-set                pic x(2) value ")0".
    02  terminal-event-flag                     pic s9(9) comp value 32.
    02  ws-bell                                 pic x value control-g.
    02  ws-false                                pic x value "F".
    02  ws-true                                 pic x value "T".


01  display-item-sizes.
    02  clear-screen-size                       pic s9(9) comp value 4.
    02  crlf-size                               pic s9(9) comp value 2.
    02  display-square-size                     pic s9(9) comp value 9.
    02  double-wide-size                        pic s9(9) comp value 3.
    02  display-player-size                     pic s9(9) comp value 8.
    02  home-cursor-size                        pic s9(9) comp value 3.
    02  room-column-size                        pic s9(9) comp.
    02  room-line-size                          pic s9(9) comp.
    02  terminal-buffer-size                    pic s9(9) comp value 1.


01  room-limits.
    02  north-limit                             pic s9(9) comp value 1.
    02  south-limit                             pic s9(9) comp value 24.
    02  east-limit                              pic s9(9) comp value 40.
    02  west-limit                              pic s9(9) comp value 1.


01  room-and-player.
    02  formatted-room-line occurs 24.
        03  room-line.
            04  square occurs 40                pic x.
    02  display-player.
        03  filler                              pic x value escape.
        03  filler                              pic x value "[".
        03  display-player-y                    pic 99.
```

```
                03  filler                             pic x value ";".
                03  display-player-x                   pic 99.
                03  filler                             pic x value "H".


        01  output-buffer.
            02  filler                                 pic x(4000).


        01  display-square.
            02  filler                                 pic x value escape.
            02  filler                                 pic x value "[".
            02  display-square-y                       pic 99.
            02  filler                                 pic x value ";".
            02  display-square-x                       pic 99.
            02  filler                                 pic x value "H".
            02  display-square-char                    pic x.


        01  score-array.
            02  score-array-size                       pic s9(9) comp.
            02  score-array-sub                        pic s9(9) comp.
            02  score-array-limit                      pic s9(9) comp  value  20.
            02  score-array-item                                       occurs 20.
                03  score-array-score                  pic zzz,zz9.
                03  filler                             pic x(02).
                03  score-array-date                   pic x(11).
                03  filler                             pic x(02).
                03  score-array-username               pic x(12).


        01  character-sequences.
            02  clear-screen.
                03  filler                             pic x(1) value escape.
                03  filler                             pic x(3) value "[2J".
            02  crlf.
                03  filler                             pic x value carriage-return.
                03  filler                             pic x value line-feed.
            02  double-wide.
                03  filler                             pic x(1) value escape.
                03  filler                             pic x(2) value "#6".
            02  home-cursor.

                03  filler                             pic x(1) value escape.
                03  filler                             pic x(2) value "[H".


        01  instructions-1.
            02  filler                                 pic x(55) value "You have landed on a space-station populated by droids.".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(42) value "Each room in the space-station is 40 x 24.".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(45) value "Each room contains progressively more droids.".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(63) value "These droids have just one function:  To destroy all intr
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(35) value "(Unfortunately, this includes you).".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(56) value "Droids are programmed to walk directly toward intruders."
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(51) value "They explode upon impact, leaving a pile of debris.".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.


        01  instructions-2.
            02  filler                                 pic x(33) value "You have three choices of action:".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(65) value "    1.  You can move in any direction (including standing
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(37) value "    2.  You can teleport at any time.".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(50) value "        (careful; you may end up next to a droid).".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(63) value "    3.  You can use the ominous Sonic Zapper once in each
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(56) value "        (This disintegrates all droids adjacent to you)."
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(54) value "Your goal is to destroy all droids, by causing them to".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
            02  filler                                 pic x(51) value "walk into each other or into piles of droid-debris.".
            02  filler                                 pic x value carriage-return.
            02  filler                                 pic x value line-feed.
```

```
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(14) value "Your controls:".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(43) value "    1-9     Move one step in any direction.".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(21) value "    T       Teleport.".
       02  filler                                pic x value carriage-return.

       02  filler                                pic x value line-feed.
       02  filler                                pic x(25) value "    S       Sonic Zapper.".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(39) value "    L       Last stand (for this room).".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(27) value "    Ctrl-R  Refresh screen.".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(8) value "Scoring:".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(41) value "    10 points for each droid you explode.".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x(46) value "     1 point for each droid you Sonically Zap.".
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.
       02  filler                                pic x value carriage-return.
       02  filler                                pic x value line-feed.


   01  totals.
       02  totals-base-position.
           03  filler                            pic x value escape.
           03  filler                            pic x value "[".
           03  totals-y                          pic 99.
           03  filler                            pic x value ";".
           03  totals-x                          pic 99 value zero.
           03  filler                            pic x value "H".
       02  totals-top-line.
           03  filler                            pic x value shift-out.
           03  filler                            pic x(27) value "lqqqqqqqqqqqqqqqqqqqqqqqqqk".
           03  filler                            pic x value shift-in.
           03  filler                            pic x value carriage-return.
           03  filler                            pic x value line-feed.
       02  totals-room-line.
           03  left-border.
               04  filler                        pic x value shift-out.
               04  filler                        pic x value "x".
               04  filler                        pic x value shift-in.
               04  filler                        pic x value space.
           03  filler                            pic x(17) value "Room #:".
           03  display-room-number               pic zz,zz9.
           03  right-border.
               04  filler                        pic x value space.
               04  filler                        pic x value shift-out.
               04  filler                        pic x value "x".
               04  filler                        pic x value shift-in.
           03  filler                            pic x value carriage-return.
           03  filler                            pic x value line-feed.
       02  totals-droids-line.
           03  left-border.
               04  filler                        pic x value shift-out.
               04  filler                        pic x value "x".
               04  filler                        pic x value shift-in.
               04  filler                        pic x value space.
           03  filler                            pic x(17) value "Droids in room:".
           03  display-droids-in-room            pic zz,zz9.
           03  right-border.
               04  filler                        pic x value space.
               04  filler                        pic x value shift-out.
               04  filler                        pic x value "x".
               04  filler                        pic x value shift-in.
           03  filler                            pic x value carriage-return.
           03  filler                            pic x value line-feed.

       02  totals-score-line.
           03  left-border.
               04  filler                        pic x value shift-out.
               04  filler                        pic x value "x".
               04  filler                        pic x value shift-in.
               04  filler                        pic x value space.
           03  filler                            pic x(17) value "Score this round:".
           03  display-score                     pic zz,zz9.
           03  right-border.
               04  filler                        pic x value space.
               04  filler                        pic x value shift-out.
               04  filler                        pic x value "x".
               04  filler                        pic x value shift-in.
           03  filler                            pic x value carriage-return.
           03  filler                            pic x value line-feed.
       02  totals-running-score-line.
           03  left-border.
```

```
            04  filler                          pic x value shift-out.
            04  filler                          pic x value "x".
            04  filler                          pic x value shift-in.
            04  filler                          pic x value space.
        03  filler                              pic x(14) value "Total score:".
        03  display-total-score                 pic z,zzz,zz9.
        03  right-border.
            04  filler                          pic x value space.
            04  filler                          pic x value shift-out.
            04  filler                          pic x value "x".
            04  filler                          pic x value shift-in.
        03  filler                              pic x value carriage-return.
        03  filler                              pic x value line-feed.
    02  totals-bottom-line.
        03  filler                              pic x value shift-out.
        03  filler                              pic x(27) value "mqqqqqqqqqqqqqqqqqqqqqqqqqj".
        03  filler                              pic x value shift-in.
        03  filler                              pic x value carriage-return.
        03  filler                              pic x value line-feed.


01  instruction-prompt                          pic x(30) value "Do you want instructions? [n] ".
01  next-page-prompt                            pic x(32) value "Press return for the next page: ".
01  press-return-prompt                         pic x(25) value "Press return when ready: ".
01  game-over-message                           pic x(12) value "Game over.  ".


01  variables.
    02  difficulty-level                        pic s9(9) comp value 5.
    02  display-x                               pic +999.
    02  display-y                               pic +999.
    02  distance                                pic s9(9) comp.
    02  droid-count                             pic s9(9) comp.
    02  droid-x                                 pic s9(9) comp.
    02  droid-y                                 pic s9(9) comp.
    02  droids-moved-counter                    pic s9(9) comp.
    02  ending-x                                pic s9(9) comp.
    02  ending-y                                pic s9(9) comp.
    02  formatted-total-score                   pic zzz,zz9.
    02  hold-droid-count                        pic s9(9) comp.
    02  line-sub                                pic s9(9) comp.
    02  new-droid-x                             pic s9(9) comp.
    02  new-droid-y                             pic s9(9) comp.
    02  output-buffer-pointer                   pic s9(9) comp.
    02  player-x                                pic s9(9) comp.
    02  player-y                                pic s9(9) comp.
    02  random-number-variables.
        03  ws-time                             pic s9(18) comp.
        03  filler redefines ws-time.
            04  random-number-seed              usage comp-1.
            04  filler                          usage comp-1.
        03  random-number                       usage comp-1.
        03  random-x                            pic s9(9) comp.
        03  random-y                            pic s9(9) comp.

    02  room-line-sub                           pic s9(9) comp.
    02  room-number                             pic s9(9) comp value zero.
    02  score-file-tries                        pic s9(9) comp.
    02  score-this-round                        pic s9(9) comp.
    02  starting-x                              pic s9(9) comp.
    02  starting-y                              pic s9(9) comp.
    02  target-player-x                         pic s9(9) comp.
    02  target-player-y                         pic s9(9) comp.
    02  teleport-x-current                      pic 99v999.
    02  teleport-x-difference                   pic 99.
    02  teleport-x-increment                    pic s99v999.
    02  teleport-y-current                      pic 99v999.
    02  teleport-y-difference                   pic 99.
    02  teleport-y-increment                    pic s99v999.
    02  total-score                             pic s9(9) comp value zero.
    02  username                                pic x(12).


01  switches.
    02  ctrlc-entered-flag                      pic x value "F".
    02  end-of-score-file-sw                    pic x.
        88  end-of-score-file                   value "T".
    02  last-stand-entered-sw                   pic x.
        88  last-stand-entered                  value "T".
    02  normal-command-entered-sw               pic x.
        88  normal-command-entered              value "T".
    02  player-is-dead-sw                       pic x value "F".
        88  player-is-dead                      value "T".
    02  sonic-screwdriver-used-up-sw            pic x.
        88  sonic-screwdriver-used-up           value "T".


01  terminal-related-data.
    02  terminal-buffer                         pic x(1).
    02  terminal-channel                        pic s9(9) comp.
    02  special-read-command                    pic s9(9) comp.
    02  special-write-command                   pic s9(9) comp.
    02  read-command                            pic s9(9) comp value external io$_readvblk.
    02  cvtlow-read-modifier                    pic s9(9) comp value external io$m_cvtlow.
    02  noecho-read-modifier                    pic s9(9) comp value external io$m_noecho.
    02  nofiltr-read-modifier                   pic s9(9) comp value external io$m_nofiltr.
    02  timed-read-modifier                     pic s9(9) comp value external io$m_timed.
    02  trmnoecho-read-modifier                 pic s9(9) comp value external io$m_trmnoecho.
    02  write-command                           pic s9(9) comp value external io$_writevblk.
    02  noformat-modifier                       pic s9(9) comp value external io$m_noformat.

    02  iosb.
```

```
        03  iobytes                             pic s9(4) comp.
        03  ioterminator                        pic s9(4) comp.
        03  ioterminator-size                   pic s9(4) comp.


PROCEDURE DIVISION.

play-droids.
        perform 1-initialize-program
        perform 2-provide-instructions
        move    ws-false to player-is-dead-sw
        perform 3-play-one-round
                until   player-is-dead
                        or
                        ctrlc-entered-flag = ws-true
        perform 4-read-top-scores
        perform 5-update-top-scores
        display clear-screen, home-cursor
        call    "lib$do_command"
                using   by descriptor   "type ci$games:droids.dat"
        .


1-initialize-program.

        compute special-read-command =  read-command +
                                        cvtlow-read-modifier +
                                        noecho-read-modifier +
                                        nofiltr-read-modifier +
                                        trmnoecho-read-modifier
        compute special-write-command = write-command +
                                        noformat-modifier
        call    "sys$assign"    using
                by descriptor   "sys$input:"
                by reference    terminal-channel,
                by value        0,
                by value        0

*       Enable a CTRL-C AST.
        call    "ci$enable_ctrlc_ast"
                using   by reference    ctrlc-entered-flag

*       perform varying line-sub from north-limit by 1
*               until   line-sub = south-limit
*
*               move    carriage-return-char to cr ( line-sub )
*               move    line-feed-char to lf ( line-sub )
*
*       end-perform
*       move    null-char to cr ( line-sub )
*       move    null-char to lf ( line-sub )

        compute room-line-size = east-limit - west-limit + 1
        compute room-column-size = south-limit - north-limit + 1

        move    1 to output-buffer-pointer

*       Initialize the G1 (shift-out) character set.
        display shift-out-character-set

*       Initialize the random number seed.
        call    "sys$gettim"
                using   by reference    ws-time
        .


2-provide-instructions.
        display instruction-prompt with no advancing
        accept  terminal-buffer
                at end
                move    ws-true to ctrlc-entered-flag
        end-accept
        if      ctrlc-entered-flag not = ws-true
                and
                (       terminal-buffer = "y"
                        or
                        terminal-buffer = "Y"
                )
        then
                display clear-screen, home-cursor, instructions-1, next-page-prompt with no advancing
                accept  terminal-buffer
                        at end
                        move    ws-true to ctrlc-entered-flag
                end-accept
                if      ctrlc-entered-flag not = ws-true
                then
                        display clear-screen, home-cursor, instructions-2, press-return-prompt with no advancing
                        accept  terminal-buffer
                                at end
                                move    ws-true to ctrlc-entered-flag
                        end-accept
                end-if
        end-if
        .


3-play-one-round.
        perform 3-2-generate-room
        perform c-1-display-entire-room
        perform 3-3-play-one-move
                until   player-is-dead
```

```
                        or
                        droid-count = zero
                        or
                        ctrlc-entered-flag = ws-true
        add     score-this-round to total-score
        perform 3-4-display-running-score
        perform 3-5-increment-difficulty-level
        .


3-2-generate-room.
        add     1 to room-number
        perform varying line-sub from north-limit by 1
                until   line-sub > south-limit

                move    spaces to room-line ( line-sub )

        end-perform
        perform 3-2-1-generate-droid
                difficulty-level times
        move    difficulty-level to droid-count
        perform 3-2-2-generate-human
        move    ws-false to sonic-screwdriver-used-up-sw
        move    ws-false to last-stand-entered-sw
        move    zero to score-this-round
        .


3-2-1-generate-droid.
        perform c-4-generate-random-coordinates
                with    test after
                until   square ( random-y, random-x ) = space
        move    droid to square ( random-y, random-x )
        .


3-2-2-generate-human.
        perform c-4-generate-random-coordinates
                with    test after
                until   square ( random-y, random-x ) = space
        move    human to square ( random-y, random-x )
        move    random-x to player-x
        move    random-y to player-y
        .


3-3-play-one-move.
        if      not last-stand-entered
        then
                perform 3-3-1-get-command
        end-if
        if      ctrlc-entered-flag = ws-false
        then
                if      not last-stand-entered
                then
                        perform 3-3-2-do-command
                end-if
                if      normal-command-entered
                then
                        move    droid-count to hold-droid-count
                        move    zero to droids-moved-counter
                        perform 3-3-3-move-droids
                                varying distance from 1 by 1
                                until   droids-moved-counter = hold-droid-count
                                        or
                                        (       player-x - distance < west-limit

                                                and
                                                player-x + distance > east-limit
                                                and
                                                player-y - distance < north-limit
                                                and
                                                player-y + distance > south-limit
                                        )
                        perform c-5-display-changed-squares
                end-if
        end-if
        .


3-3-1-get-command.
        move    null-char to terminal-buffer
        call    "sys$qiow" using
                by value        terminal-event-flag
                by value        terminal-channel
                by value        special-read-command
                by reference    iosb
                by value        zero
                by value        zero
                by reference    terminal-buffer
                by value        terminal-buffer-size
                by value        zero
                by value        zero
                by value        zero
                by value        zero
        .


3-3-2-do-command.
        move    ws-true to normal-command-entered-sw
        evaluate terminal-buffer
                when    "S"
```

+2
+5
+17
20
BH = 86
BC 24

```
                if      sonic-screwdriver-used-up
                then
                        display "" with no advancing
                        move    ws-false to normal-command-entered-sw
                else
                        perform 3-3-2-1-use-sonic-screwdriver
                end-if
        when    "T"
                perform 3-3-2-2-teleport-player
        when    "1"
                compute target-player-x = player-x - 1
                compute target-player-y = player-y + 1
                perform c-3-move-player
        when    "2"
                compute target-player-x = player-x
                compute target-player-y = player-y + 1
                perform c-3-move-player
        when    "3"
                compute target-player-x = player-x + 1
                compute target-player-y = player-y + 1
                perform c-3-move-player
        when    "4"
                compute target-player-x = player-x - 1
                compute target-player-y = player-y
                perform c-3-move-player
        when    "5"
                continue
        when    "6"
                compute target-player-x = player-x + 1
                compute target-player-y = player-y
                perform c-3-move-player
        when    "7"
                compute target-player-x = player-x - 1
                compute target-player-y = player-y - 1
                perform c-3-move-player
        when    "8"

                compute target-player-x = player-x
                compute target-player-y = player-y - 1
                perform c-3-move-player
        when    "9"
                compute target-player-x = player-x + 1
                compute target-player-y = player-y - 1
                perform c-3-move-player
        when    control-r-char
                perform c-1-display-entire-room
                move    ws-false to normal-command-entered-sw
        when    "L"
                move    ws-true to last-stand-entered-sw
        when    other
                display "" with no advancing
                move    ws-false to normal-command-entered-sw
end-evaluate
.

3-3-2-1-use-sonic-screwdriver.
        compute starting-x = player-x - 1
        perform varying droid-x from starting-x by 1
                until   droid-x > player-x + 1

                compute starting-y = player-y - 1
                perform varying droid-y from starting-y by 1
                        until   droid-y > player-y + 1

                        if      (       droid-x not = player-x
                                        or
                                        droid-y not = player-y
                                )
                                and
                                droid-x not < west-limit
                                and
                                droid-x not > east-limit
                                and
                                droid-y not < north-limit
                                and
                                droid-y not > south-limit
                        then
                                if      square ( droid-y, droid-x ) = droid
                                then
                                        move    space to square ( droid-y, droid-x )
                                        move    space to display-square-char
                                        move    droid-x to display-square-x
                                        move    droid-y to display-square-y
                                        move    display-square to output-buffer ( output-buffer-pointer : display-square-s
                                        add     display-square-size to output-buffer-pointer

                                        add     1 to score-this-round
                                        subtract 1 from droid-count
                                        display "" with no advancing
                                end-if
                        end-if
                end-perform
        end-perform
        move    ws-true to sonic-screwdriver-used-up-sw
        .


3-3-2-2-teleport-player.
*       Find new space for human
        perform c-4-generate-random-coordinates
                with    test after
                until   square ( random-y, random-x ) = space

      *         Eliminate human from current square
                move    space to square ( player-y, player-x )

      *         Make it dramatic on screen

                perform 3-3-2-2-1-display-teleportation

      *         Put human into new square
                move    human to square ( random-y, random-x )
                move    random-x to player-x
                move    random-y to player-y
                .

       3-3-2-2-1-display-teleportation.
                if      player-y < random-y
                then
                        move    +1 to teleport-y-increment
                else
                        move    -1 to teleport-y-increment
                end-if

                if      player-x < random-x
                then
                        move    +1 to teleport-x-increment
                else
                        move    -1 to teleport-x-increment
                end-if

      *         These differences are unsigned.
                compute teleport-y-difference = player-y - random-y
                compute teleport-x-difference = player-x - random-x

                if      teleport-y-difference > teleport-x-difference
                then
                        compute teleport-x-increment = teleport-x-increment * ( teleport-x-difference / teleport-y-difference )
                        perform 3-3-2-2-1-1-enteleport-by-y
                        perform 3-3-2-2-1-2-deteleport-by-y
                else
                        if      teleport-x-difference > teleport-y-difference
                        then
                                compute teleport-y-increment = teleport-y-increment * ( teleport-y-difference / teleport-x-differe
                                perform 3-3-2-2-1-3-enteleport-by-x
                                perform 3-3-2-2-1-4-deteleport-by-x
                        else
                                if      teleport-y-difference not = zero
                                then
                                        compute teleport-y-increment = teleport-y-increment * ( teleport-y-difference / teleport-x
                                        perform 3-3-2-2-1-3-enteleport-by-x
                                        perform 3-3-2-2-1-4-deteleport-by-x
                                else
      *                                 ( Both differences = zero )
                                        continue
                                end-if
                        end-if
                end-if
                .

       3-3-2-2-1-1-enteleport-by-y.
                compute teleport-x-current = player-x + teleport-x-increment
                compute teleport-y-current = player-y + teleport-y-increment

                move    human to display-square-char

                perform with test after
                        varying teleport-y-current from teleport-y-current by teleport-y-increment
                        until   teleport-y-current = random-y

                        move    teleport-y-current to display-square-y
                        compute display-square-x rounded = teleport-x-current

      *                 Display human in intermediate square
                        move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                        add     display-square-size to output-buffer-pointer

                        compute teleport-x-current = teleport-x-current + teleport-x-increment


                end-perform
                .

       3-3-2-2-1-3-enteleport-by-x.
                compute teleport-x-current = player-x + teleport-x-increment
                compute teleport-y-current = player-y + teleport-y-increment

                move    human to display-square-char

                perform with test after
                        varying teleport-x-current from teleport-x-current by teleport-x-increment
                        until   teleport-x-current = random-x

                        move    teleport-x-current to display-square-x
                        compute display-square-y rounded = teleport-y-current

      *                 Display human in intermediate square
                        move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                        add     display-square-size to output-buffer-pointer

                        compute teleport-y-current = teleport-y-current + teleport-y-increment
```

```
        end-perform
        .

3-3-2-2-1-2-deteleport-by-y.
        move    player-x to teleport-x-current
        perform with test before
                varying teleport-y-current from player-y by teleport-y-increment
                until   teleport-y-current = random-y

                move    teleport-y-current to display-square-y
                compute display-square-x rounded = teleport-x-current

*               Redisplay actual intermediate-square contents
                move    square ( display-square-y, display-square-x ) to display-square-char
                move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                add     display-square-size to output-buffer-pointer

                compute teleport-x-current = teleport-x-current + teleport-x-increment

        end-perform
        .

3-3-2-2-1-4-deteleport-by-x.
        move    player-y to teleport-y-current
        perform with test before
                varying teleport-x-current from player-x by teleport-x-increment
                until   teleport-x-current = random-x

                move    teleport-x-current to display-square-x
                compute display-square-y rounded = teleport-y-current

*               Redisplay actual intermediate-square contents
                move    square ( display-square-y, display-square-x ) to display-square-char
                move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                add     display-square-size to output-buffer-pointer

                compute teleport-y-current = teleport-y-current + teleport-y-increment

        end-perform
        .

3-3-3-move-droids.
        compute droid-y = player-y - distance
        if      droid-y not < north-limit
        then
                perform 3-3-3-1-move-north-droids
        end-if


        compute droid-x = player-x + distance
        if      droid-x not > east-limit
        then
                perform 3-3-3-2-move-east-droids
        end-if

        compute droid-y = player-y + distance
        if      droid-y not > south-limit
        then
                perform 3-3-3-3-move-south-droids
        end-if

        compute droid-x = player-x - distance
        if      droid-x not < west-limit
        then
                perform 3-3-3-4-move-west-droids
        end-if
        .


3-3-3-1-move-north-droids.
* droid-y was already computed before calling this paragraph.
        compute starting-x = player-x - distance + 1
        if      starting-x < west-limit
        then
                move    west-limit to starting-x
        end-if
        compute ending-x = player-x + distance
        if      ending-x > east-limit
        then
                move    east-limit to ending-x
        end-if
        perform varying droid-x from starting-x by 1
                until droid-x > ending-x

                if      square ( droid-y, droid-x ) = droid
                then
                        if      droid-x < player-x
                        then
                                compute new-droid-x = droid-x + 1
                        else
                                if      droid-x = player-x
                                then
                                        compute new-droid-x = droid-x
                                else
***                                     [ droid-x > player-x ]
                                        compute new-droid-x = droid-x - 1
                                end-if
                        end-if
                        compute new-droid-y = droid-y + 1
                        perform c-2-move-droid
                end-if
        end-perform
```

```
       .

3-3-3-2-move-east-droids.
* droid-x was already computed before calling this paragraph.
       compute starting-y = player-y - distance + 1
       if      starting-y < north-limit
       then
               move    north-limit to starting-y
       end-if
       compute ending-y = player-y + distance
       if      ending-y > south-limit
       then
               move    south-limit to ending-y
       end-if
       perform varying droid-y from starting-y by 1
               until droid-y > ending-y


               if      square ( droid-y, droid-x ) = droid
               then
                       if      droid-y < player-y
                       then
                               compute new-droid-y = droid-y + 1
                       else
                               if      droid-y = player-y
                               then
                                       compute new-droid-y = droid-y
                               else
***                                    [ droid-y > player-y ]
                                       compute new-droid-y = droid-y - 1
                               end-if
                       end-if
                       compute new-droid-x = droid-x - 1
                       perform c-2-move-droid
               end-if
       end-perform
       .

3-3-3-3-move-south-droids.
* droid-y was already computed before calling this paragraph.
       compute starting-x = player-x + distance - 1
       if      starting-x > east-limit
       then
               move    east-limit to starting-x
       end-if
       compute ending-x = player-x - distance
       if      ending-x < west-limit
       then
               move    west-limit to ending-x
       end-if
       perform varying droid-x from starting-x by -1
               until droid-x < ending-x

               if      square ( droid-y, droid-x ) = droid
               then
                       if      droid-x < player-x
                       then
                               compute new-droid-x = droid-x + 1
                       else
                               if      droid-x = player-x
                               then
                                       compute new-droid-x = droid-x
                               else
***                                    [ droid-x > player-x ]
                                       compute new-droid-x = droid-x - 1
                               end-if
                       end-if
                       compute new-droid-y = droid-y - 1
                       perform c-2-move-droid
               end-if
       end-perform
       .

3-3-3-4-move-west-droids.
* droid-x was already computed before calling this paragraph.
       compute starting-y = player-y + distance - 1
       if      starting-y > south-limit
       then
               move    south-limit to starting-y
       end-if
       compute ending-y = player-y - distance
       if      ending-y < north-limit
       then
               move    north-limit to ending-y
       end-if
       perform varying droid-y from starting-y by -1
               until droid-y < ending-y

               if      square ( droid-y, droid-x ) = droid
               then
                       if      droid-y < player-y
                       then
                               compute new-droid-y = droid-y + 1
                       else
                               if      droid-y = player-y
                               then
                                       compute new-droid-y = droid-y
                               else
```

```
***                                             [ droid-y > player-y ]
                                                compute new-droid-y = droid-y - 1
                                        end-if
                                end-if
                                compute new-droid-x = droid-x + 1
                                perform c-2-move-droid
                        end-if
                end-perform
                .


3-4-display-running-score.
                move    room-number to display-room-number
                move    difficulty-level to display-droids-in-room
                move    score-this-round to display-score
                move    total-score to display-total-score
*               The number (8) is:      the number of totals lines (including borders), plus the press-return-prompt, plus one (1)
                if      player-y < north-limit + 8
                then
                        compute totals-y = south-limit - 8
                else
                        compute totals-y = north-limit
                end-if
                if      player-is-dead
                        or
                        ctrlc-entered-flag = ws-true
                then
                        display totals, game-over-message, press-return-prompt, ws-bell, ws-bell, ws-bell, ws-bell with no advanci
                else
                        display totals, press-return-prompt, ws-bell, ws-bell, ws-bell, ws-bell with no advancing
                end-if
                accept  terminal-buffer
                        at end
                        move    ws-true to ctrlc-entered-flag
                end-accept
                .


3-5-increment-difficulty-level.
                add     difficulty-increment to difficulty-level
                .


4-read-top-scores.
                open    i-o     score-file
                move    ws-false to end-of-score-file-sw
*               Read past the title record.
                perform c-7-read-score-record
                if      not end-of-score-file
                then
                        perform c-7-read-score-record
                end-if
                move    zero to score-array-sub
                perform until   score-array-sub = score-array-limit
                                or
                                end-of-score-file
                        add     1 to score-array-sub
                        move    score-record to score-array-item ( score-array-sub )
                        perform c-7-read-score-record
                end-perform

                move    score-array-sub to score-array-size
                close   score-file
                .


5-update-top-scores.
                move    total-score to formatted-total-score
                perform varying score-array-sub from score-array-size by -1
                        until   score-array-sub = zero
                                or
                                score-array-score ( score-array-sub ) not < formatted-total-score
                        if      score-array-sub not = score-array-limit
                        then
                                move    score-array-item ( score-array-sub ) to score-array-item ( score-array-sub + 1 )
                        end-if
                end-perform
                if      score-array-size < score-array-limit
                then
                        add     1 to score-array-size
                        perform 5-1-load-score-into-array
                        perform 5-2-rewrite-score-array
                else
                        if      score-array-sub < score-array-limit
                        then
                                perform 5-1-load-score-into-array
                                perform 5-2-rewrite-score-array
                        end-if
                end-if
                .


5-1-load-score-into-array.
                move    spaces to score-array-item ( score-array-sub + 1 )
                move    total-score to score-array-score ( score-array-sub + 1 )
                call    "lib$date_time"
                        using   by descriptor   score-array-date  ( score-array-sub + 1 )
                call    "ci$get_vax_username"
                        using   by reference    score-array-username ( score-array-sub + 1 )
                .
```

```
5-2-rewrite-score-array.
        open    i-o     score-file
        move    ws-false to end-of-score-file-sw
        perform 5-2-1-position-score-file
        move    score-file-title to score-record
        perform c-8-write-score-record
        perform varying score-array-sub from 1 by 1
                until   score-array-sub > score-array-size
                perform 5-2-1-position-score-file
                move    score-array-item ( score-array-sub ) to score-record
                perform c-8-write-score-record
        end-perform
        close   score-file
        .


5-2-1-position-score-file.
        if      not end-of-score-file
        then
                perform c-7-read-score-record
                if      not end-of-score-file
                then
                        continue
                else
                        close   score-file
                        open    extend  score-file
                end-if
        end-if
        .


c-1-display-entire-room.
        move    clear-screen to output-buffer ( output-buffer-pointer : clear-screen-size )
        add     clear-screen-size to output-buffer-pointer

        move    home-cursor to output-buffer ( output-buffer-pointer : home-cursor-size )
        add     home-cursor-size to output-buffer-pointer

        perform varying room-line-sub from north-limit by 1
                until room-line-sub > south-limit

                move    double-wide to output-buffer ( output-buffer-pointer : double-wide-size )
                add     double-wide-size to output-buffer-pointer

                move    room-line ( room-line-sub ) to output-buffer ( output-buffer-pointer : room-line-size )
                add     room-line-size to output-buffer-pointer

                if      room-line-sub not = south-limit
                then
                        move    crlf to output-buffer ( output-buffer-pointer : crlf-size )
                        add     crlf-size to output-buffer-pointer
                end-if

        end-perform

        move    player-x to display-player-x
        move    player-y to display-player-y
        move    display-player to output-buffer ( output-buffer-pointer : display-player-size )
        add     display-player-size to output-buffer-pointer

        perform c-6-write-buffer
        .


c-2-move-droid.
        add     1 to droids-moved-counter
        move    space to square ( droid-y, droid-x )
        move    space to display-square-char
        move    droid-x to display-square-x
        move    droid-y to display-square-y
        move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
        add     display-square-size to output-buffer-pointer
        evaluate square ( new-droid-y, new-droid-x )
                when    space
                        move    droid to square ( new-droid-y, new-droid-x )
                        move    droid to display-square-char
                        move    new-droid-x to display-square-x
                        move    new-droid-y to display-square-y
                        move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                        add     display-square-size to output-buffer-pointer
                when    droid
                        move    dust to square ( new-droid-y, new-droid-x )
                        move    dust to display-square-char
                        move    new-droid-x to display-square-x
                        move    new-droid-y to display-square-y
                        move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                        add     display-square-size to output-buffer-pointer
                        add     20 to score-this-round
                        subtract 2 from droid-count
                when    dust
                        add     10 to score-this-round
                        subtract 1 from droid-count
                when    dead-human
                        add     10 to score-this-round
                        subtract 1 from droid-count
                when    human
                        move    dead-human to square ( new-droid-y, new-droid-x )
                        move    dead-human to display-square-char
                        move    new-droid-x to display-square-x
                        move    new-droid-y to display-square-y
                        move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                        add     display-square-size to output-buffer-pointer
                        add     10 to score-this-round
                        subtract 1 from droid-count
                        move    ws-true to player-is-dead-sw
                when    other
                        move new-droid-x to display-x
                        move new-droid-y to display-y
                        display "Problem square '"
                                square ( new-droid-y, new-droid-x )
                                "' in c-2-move-droid at ("
                                display-y
                                ","
                                display-x
                                ").  Press RETURN when ready:  " with no advancing
                        accept  terminal-buffer
                                at end
                                move    ws-true to ctrlc-entered-flag
                        end-accept
        end-evaluate
        .


c-3-move-player.
        if      target-player-x < west-limit
        then
                move    ws-false to normal-command-entered-sw
        else
                if      target-player-x > east-limit
                then
                        move    ws-false to normal-command-entered-sw
                end-if
        end-if
        if      target-player-y < north-limit
        then
                move    ws-false to normal-command-entered-sw
        else
                if      target-player-y > south-limit
                then
                        move    ws-false to normal-command-entered-sw
                end-if
        end-if
        if      normal-command-entered
        then
                move    space to square ( player-y, player-x )
                move    space to display-square-char
                move    player-x to display-square-x
                move    player-y to display-square-y
                move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                add     display-square-size to output-buffer-pointer
                evaluate square ( target-player-y, target-player-x )
                        when    space
                                move    human to square ( target-player-y, target-player-x )
                                move    human to display-square-char
                                move    target-player-x to display-square-x
                                move    target-player-y to display-square-y
                                move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                                add     display-square-size to output-buffer-pointer
                        when    droid
                                add     10 to score-this-round
                                subtract 1 from droid-count
                                move    ws-true to player-is-dead-sw
                                move    dead-human to display-square-char
                                move    target-player-x to display-square-x
                                move    target-player-y to display-square-y
                                move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                                add     display-square-size to output-buffer-pointer
                        when    dust
                                move    ws-true to player-is-dead-sw
                                move    dead-human to display-square-char
                                move    target-player-x to display-square-x
                                move    target-player-y to display-square-y

                                move    display-square to output-buffer ( output-buffer-pointer : display-square-size )
                                add     display-square-size to output-buffer-pointer
                        when    other
                                move target-player-x to display-x
                                move target-player-y to display-y
                                display "Problem square '"
                                        square ( target-player-y, target-player-x )
                                        "' in c-3-move-player at ("
                                        display-y
                                        ","
                                        display-x
                                        ").  Press RETURN when ready:  " with no advancing
                                accept  terminal-buffer
                                        at end
                                        move    ws-true to ctrlc-entered-flag
                                end-accept
                end-evaluate
                move    target-player-x to player-x
                move    target-player-y to player-y
        else
                display "" with no advancing
        end-if
        .


c-4-generate-random-coordinates.
        call    "mth$random"
                using   by reference    random-number-seed
                giving  random-number
        compute random-x = ( random-number * ( room-line-size ) ) + 1
        call    "mth$random"
                using   by reference    random-number-seed
                giving  random-number
        compute random-y = ( random-number * ( room-column-size ) ) + 1
        .


c-5-display-changed-squares.
        move    player-x to display-player-x
        move    player-y to display-player-y
        move    display-player to output-buffer ( output-buffer-pointer : display-player-size )
        add     display-player-size to output-buffer-pointer
        perform c-6-write-buffer
        .


c-6-write-buffer.
        compute output-buffer-pointer = output-buffer-pointer - 1
        call    "sys$qiow" using
                by value        terminal-event-flag
                by value        terminal-channel
                by value        special-write-command
                by reference    iosb
                by value        zero
                by value        zero
                by reference    output-buffer
                by value        output-buffer-pointer
                by value        zero
                by value        zero
                by value        zero
                by value        zero

        move    1 to output-buffer-pointer
        .


c-7-read-score-record.
        read    score-file
                at end
                move    ws-true to end-of-score-file-sw
        end-read

        .


c-8-write-score-record.
        if      not end-of-score-file
        then
                rewrite score-record
        else
                write   score-record
        end-if
        .
```