

```
// (C) 2023 Virtual Causality
```

```
dim args, 8 // FUNCTION ARGUMENTS
dim arg2, 8
dim rets, 8 // FUNCTION RETURNS
dim kflg, 12 // KEY INPUT FLAGS
dim kgot, 12 // KEY INPUT GOTS
dim kode, 12 // KEY CODES
dim kcda, 12 // Type A
dim kcdb, 12 // Type B
dim kcla, 12 // TEXTURE A
dim kclb, 12 // TEXTURE B
dim kcle, 12 // TEXTURE
dim comp, 4
dim rate, 8
dim heal, 8
dim cdmg, 8
dim cnt, 8 // CELL_GOT
dim ldnc, 21 // WAIT_TIME
dim ltoc, 21 // BLOCK_LOOT_OC
dim ltt1, 21 // BLOCK_LOOT_T1
dim lsta, 6 // LEVEL_START
dim llim, 6 // LEVEL_LIMIT
dim lnxt, 6
dim sgsw, 9
dim sgdf, 9
dim sglm, 9
dim scsw, 1
dim scdf, 1
dim sclm, 1
dim sosw, 1
dim sodf, 1
dim solm, 1
dim ctsw, 1
dim ctdf, 1
dim ctlm, 1
```

```
ldim klab, 19 // KEY_LABELS
sdim txtv, 6 // TEXT_FORMAT_PROPS
```

```
// KEY BIND
```

```
// MVL/CVL[A] MVR/CVR[D] RTL/RST[J] RTR/NOP[K] PSH/CVU[W] POP/CXL[L] DRP/DEC[SpC.] ACC/CVD[S] SCR[F] LOG[G] ACT[H]
PAU[Esc.]
```

```
kcda = 65, 68, 74, 75, 87, 76, 32, 83, 70, 71, 72, 27 // DEFAULT
```

```
kcla = 0, 24, 144, 168, 264, 192, 408, 240, 72, 96, 120, 384 // DEFAULT
```

```
// MVL/CVL[<-] MVR/CVR[->] RTL/RST[D] RTR/NOP[F] PSH/CVU[^] POP/CXL[G] DRP/DEC[SpC.] ACC/CVD[,.] SCR[Q] LOG[W] ACT[E]
PAU[Esc.]
```

```
kcdb = 37, 39, 68, 70, 38, 71, 32, 40, 81, 87, 69, 27
```

```
kclb = 288, 360, 24, 72, 312, 96, 408, 336, 216, 264, 48, 384
```

```
sgsw = 1, 1, 1, 1, 1, 1, 1, 2, 2 // {0, 1} {0, 1, 2}
```

```
sgdf = 1, 1, 1, 1, 1, 1, 1, 2, 2
```

```
sglm = 2, 2, 2, 2, 2, 2, 2, 3, 3
```

```
scsw = 1 // {0, 1}
```

```
scdf = 1
```

```
sclm = 2
```

```
sosw = 100 // {0, ..., 100}
```

```
sodf = 100
```

```
solm = 101
```

```
ctsw = 0 // {0, 1}
```

```
ctdf = 0
```

```
ctlm = 2
```

```

repeat 12
    kode(cnt) = kcda(cnt)
    kcle(cnt) = kcla(cnt)
loop

// MULTI DELETE BONUS
// xN
comp = 1, 2, 4, 8

// TRIGGERED KEY CONTROL TO
// 0-6 IN GAME 7-14 IN MENU 15-18 IN ALL
klab = *k_mvl, *k_mvr, *k_rtl, *k_rtr, *k_psh, *k_pop, *k_dn, *k_cvl, *k_cvr, *k_rst, *k_nop, *k_cvu, *k_cxl, *k_dec, *k_cvd,
*k_pa, *k_scr, *k_log, *k_act

// WAIT COUNT
ldnc = 201, 191, 181, 171, 161, 151, 141, 131, 121, 111, 101, 91, 81, 71, 61, 51, 41, 21, 121, 121, 2

// BLOCK LOOT
ltoc = 208, 208, 208, 208, 208, 208, 192, 192, 192, 192, 176, 176, 176, 176, 160, 160, 160, 144, 256, 0, 128
ltt1 = 240, 240, 240, 232, 232, 232, 224, 224, 224, 224, 224, 216, 216, 216, 216, 208, 208, 208, 192, 512, 64, 160

// LEVEL INFO
lsta = 0, 3, 6, 9, 0, 20
lim = 6, 10, 14, 18, 22, 21
lnxt = 8, 8, 7, 7, -1, 2

#const TRUE 1
#const FALSE 0

#const QUEUE_0 480
#const QUEUE_1 496
#const QUEUE_2 512
#const QUEUE_3 528
#const QUEUE_4 544
#const QUEUE_G 16
#const STACK_0 560
#const STACK_1 576
#const STACK_2 592
#const CONTROL 0
#const BOARD_H 32
#const BOARD_V 88
#const SEP_W 4
#const OPN_W 14
#const COL_H 56
#const COL_V 112
#const FL_JUMP 10

#const G_DN_DRP 1
#const G_DP_ROT 2
#const G_DP_MOV 1
#const G_DN_ACC 6

#const KEY_MVL 0
#const KEY_MVR 1
#const KEY_RTL 2
#const KEY_RTR 3
#const KEY_PSH 4
#const KEY_POP 5
#const KEY_DRP 6
#const KEY_ACC 7
#const KEY_SCR 8

```

```
#const KEY_LOG 9
#const KEY_ACT 10
#const KEY_PAU 11
#const KEY_CVL 0
#const KEY_CVR 1
#const KEY_RST 2
#const KEY_NOP 3
#const KEY_CVU 4
#const KEY_CXL 5
#const KEY_DEC 6
#const KEY_CVD 7
```

```
#const LAB_MVL 0
#const LAB_MVR 1
#const LAB_RTL 2
#const LAB_RTR 3
#const LAB_PSH 4
#const LAB_POP 5
#const LAB_DRP 6
#const LAB_SCR 16
#const LAB_LOG 17
#const LAB_ACT 18
#const LAB_PAU 15
#const LAB_CVL 7
#const LAB_CVR 8
#const LAB_RST 9
#const LAB_NOP 10
#const LAB_CVU 11
#const LAB_CXL 12
#const LAB_DEC 13
#const LAB_CVD 14
```

```
#const C_VA 0
#const C_OC 1
#const C_T1 2
#const C_T2 3
#const C_TG 4
#const C_BM 5
#const C_BG 6
#const C_BF 7
```

```
// BLOCK RATE
```

```
rate = 0, 100, 500, 50, 8000, 0, 1000, 10
heal = 0, 2, 10, 1, 40, 0, 20, 1
cdmg = 40, 15, 25, 20, 100, 0, 30, 500
```

```
#const S_OC 100
#const S_T1 200
#const S_TG 1600
#const S_BG 800
#const S_BF 10
```

```
#const H_OC 2
#const H_T1 10
#const H_TG 40
#const H_BG 20
#const H_BF 1
```

```
#const D_OF 1000
#const D_BM 3500
```

```
#const PLAYER_HP_X 10000
```

```

// STAT - IN GAME > 0
#const GSTAT_G_S 0x00000001 // START
#const GSTAT_G_X 0x00000002 // DELETE
#const GSTAT_G_N 0x00000003 // NORMAL
#const GSTAT_G_F 0x00000004 // FINAL APPROACH
#const GSTAT_G_J 0x00000005 // JUDGE
// STAT - MENU < 0
#const GSTAT_P_N 0xFFFFFFFF // TITLE
#const GSTAT_P_D 0xFFFFFFFFE // DEATH
#const GSTAT_P_S 0xFFFFFFFFD // SURVIVAL
#const GSTAT_P_P 0xFFFFFFFFC // PAUSE

#const GSTAT_P_N_N 0xFFFF1100 // -MASK-
#const GSTAT_T_L1 0xFFFF1100 // LEVEL
#const GSTAT_T_C1 0xFFFF1101 // CONFIG
#const GSTAT_T_I1 0xFFFF1102 // INFO
#const GSTAT_T_Q 0xFFFF1103 // QUIT

#const GSTAT_T_L1_N 0xFFFF2100 // -MASK-
#const GSTAT_T_L2_C 0xFFFF2100 // COPPER
#const GSTAT_T_L2_S 0xFFFF2101 // SILVER
#const GSTAT_T_L2_G 0xFFFF2102 // GOLD
#const GSTAT_T_L2_P 0xFFFF2103 // PLATINUM
#const GSTAT_T_L2_N 0xFFFF2104 // IRON
#const GSTAT_T_L2_M 0xFFFF2105 // IRIDIUM

#const GSTAT_T_C1_N 0xFFFF2120 // -MASK-
#const GSTAT_T_C2_SEG 0xFFFF2120 // SEGMENT
#const GSTAT_T_C2_GRA 0xFFFF21201 // GRAPHICS
#const GSTAT_T_C2_SOU 0xFFFF21202 // SOUND
#const GSTAT_T_C2_CTR 0xFFFF21203 // CONTROL

#const GSTAT_T_I1_N 0xFFFF2140 // -MASK-
#const GSTAT_T_I2_ABO 0xFFFF2140 // ABOUT
#const GSTAT_T_I2_SYS 0xFFFF21401 // SYSTEM
#const GSTAT_T_I2_LIC 0xFFFF21402 // LICENSE
#const GSTAT_T_I2_STA 0xFFFF21403 // STAFF

#const SEG_CLOCK 0
#const SEG_LEVEL 1
#const SEG_WAIT 2
#const SEG_HP 3
#const SEG_SCORE 4
#const SEG_LINES 5
#const SEG_COUNT 6
#const BAR_HP 7
#const ICO_COUNT 8

#const SCR_CSIZE 0

dim vertices, 2080
dim wsize, 2
dim wsize, 2
dim kposx, 12
dim kposy, 12
dim cposg, 12
dim cposp, 12
wsize = 800, 1600
wsize = 900, 1800
dwsizex = 1600
dwsizex = 1800

```

```
kposx = 535, 583, 654, 678, 559, 630, 583, 559, 630, 654, 678, 535
kposy = 404, 404, 461, 461, 404, 461, 461, 461, 404, 404, 404, 461
cposg = 0, 24, 192, 216, 48, 72, 144, 168, 288, 312, 336, 240
cposp = 0, 24, 408, 432, 96, 384, 360, 120, 288, 312, 336, 264
```

```
args(2) = 0
```

```
// COORD
```

```
args(0) = 28 : args(1) = 14 : args(3) = 156 : args(4) = 205 : args(5) = 23 : args(6) = 24 : gosub *ipt
```

```
// QUEUE & STACK
```

```
/* Q0 */ args(0) = 4 : args(1) = 4 : args(3) = 52 : args(4) = 101 : gosub *ipt
```

```
/* Q1 */ args(4) = 205 : gosub *ipt
```

```
/* Q2 */ args(4) = 309 : gosub *ipt
```

```
/* Q3 */ args(4) = 413 : gosub *ipt
```

```
/* Q4 */ args(4) = 517 : gosub *ipt
```

```
/* S0 */ args(3) = 500 : gosub *ipt
```

```
/* S1 */ args(4) = 621 : gosub *ipt
```

```
/* S2 */ args(4) = 725 : gosub *ipt
```

```
// GAME VARS
```

```
dim cells, 608
```

```
c_sw = 0
```

```
c_x = 0 : c_y = 0 // CONTROL POS LT
```

```
i_flg = 0 : i_f2 = TRUE : i_cnt = 0 : i_idx = 0
```

```
i_v1 = 0 : i_v2 = 0 : i_v3 = 0 : i_v4 = 0 : i_v5 = 0 : i_v6 = 0
```

```
g_v1 = 0 : g_v2 = 0 : g_v3 = 0
```

```
g_stat = GSTAT_P_N
```

```
p_stat = GSTAT_P_N
```

```
dn_cnt = 0
```

```
dn_max = 1
```

```
qg_isvacant = TRUE
```

```
s_cnt = 0 : t_cnt = 0
```

```
hzd = FALSE
```

```
f_v1 = 0 : f_v2 = 255 : f_v3 = 0 : f_v4 = 0 : f_v5 = 0 : f_v6 = 0
```

```
score = 0
```

```
l_cnt = 0 : l_loc = 0
```

```
g_level = 0 : g_mode = 0
```

```
hp = PLAYER_HP_X
```

```
fx_cnt = 1
```

```
s_v1 = ""
```

```
// MENU VARS
```

```
cur_v = 0
```

```
cur_h = 0
```

```
cur_k = 0
```

```
ir_open = 0
```

```
cfig_sel = 0
```

```
// WINDOW INIT
```

```
screen 0, 1600, 1800
```

```
randomize
```

```
buffer 1, 1024, 1024
```

```
gsel 1
```

```
pos 0, 0
```

```
picload "backgroundx.png", 1
```

```
pos 800, 0
```

```
picload "seg_hsp.png", 1
```

```
pos 0, 900
```

```
picload "hpbar_hsp.png", 1
```

```

color 0, 0, 0
boxf 840, 0, 863, 23
color 127, 127, 127
boxf 864, 0, 887, 23

gosub *barcolor

buffer 2, 1024, 1024
gsel 2
pos 0, 0
picload "cells_hsp.png", 1
pos 48, 0
picload "qs_hsp.png", 1
pos 48, 150
gzoom 96, 48, 2, 0, 48, 48, 24
pos 48, 198
gzoom 96, 48, 2, 0, 96, 48, 24
pos 0, 256
picload "logo_v2_hsp.png", 1 // 352 x 80
pos 512, 0
picload "menu16_2_hsp.png", 1 // 32 x 272
pos 560, 0
picload "menu24_2_hsp.png", 1 // 48 x 456
pos 608, 0
picload "menu24_4.png", 1 // 72 x 432
pos 0, 512
picload "p_logo_hsp.png", 1

buffer 3, 1024, 1024
gsel 3
pos 0, 0
picload "p1.png", 1 // 336 x 147
pos 0, 147
picload "menu_bg.png", 1 // 336 x 672
pos 336, 0
picload "levels.png", 1 // 167 x 144

buffer 4, 512, 512
gsel 4
pos 0, 0
picload "p2_nc.png", 1

buffer 6, 1024, 1024
gsel 6
color 255, 255, 255
boxf 0, 0, 243, 1023
color 0, 0, 0
boxf 244, 0, 487, 1023
// 48
notesel nb_1
noteload "ig_text_1.txt"
gosub *rnt

color 255, 255, 255
boxf 488, 0, 571, 1023
color 0, 0, 0
boxf 572, 0, 655, 1023
// 16 LONG
notesel nb_2
noteload "ig_text_2.txt"
gosub *rnt

```

```

color 255, 255, 255
boxf 656, 0, 691, 1023
color 0, 0, 0
boxf 692, 0, 728, 1023
// 16 SHORT
notesel nb_3
noteload "ig_text_3.txt"
gosub *rnt

```

```

buffer 7, 2048, 2048
gsel 7
color 255, 255, 255
boxf 0, 0, 323, 2047
color 0, 0, 0
boxf 324, 0, 647, 2047
// 16 x 40
notesel nb_4
noteload "ig_text_4.txt"
gosub *rnt

```

```

color 255, 255, 255
boxf 648, 0, 971, 2047
color 0, 0, 0
boxf 972, 0, 1295, 2047
// 16 x 40 x 23
notesel nb_5
noteload "ig_text_5.txt"
gosub *rnt

```

```

color 255, 255, 255
boxf 1296, 0, 1619, 2047
color 0, 0, 0
boxf 1620, 0, 1943, 2047
// 16 x 40
notesel nb_6
noteload "ig_text_6.txt"
gosub *rnt

```

```

buffer 5, 800, 900

```

```

*main

```

```

// SYSTEM BODY START

```

```

args(0) = KEY_SCR : args(1) = LAB_SCR : gosub *key_pshdn
args(0) = KEY_LOG : args(1) = LAB_LOG : gosub *key_pshdn
args(0) = KEY_ACT : args(1) = LAB_ACT : gosub *key_pshdn
args(0) = KEY_PAU : args(1) = LAB_PAU : gosub *key_pshdn

```

```

if (g_stat < 0) {
  if (g_stat = GSTAT_T_Q) : end
  // MENU
  repeat 8
    args(0) = cnt : args(1) = cnt + 7 : gosub *key_pshdn
  loop
  if (g_stat < GSTAT_P_N) : i_f2 = FALSE
  if ((g_stat = GSTAT_T_C2_SOU) & (cfg_sel = 1) & (kgot(KEY_NOP) = 0)) {
    if (kgot(KEY_CVL) = 1) : cur_h-
    if (kgot(KEY_CVR) = 1) : cur_h+
    if (cur_h < 0) : cur_h = 0
    if (cur_h > 100) : cur_h = 100
    sosw(cur_v) = cur_h
  }
} else : if (g_stat = GSTAT_G_S) {

```

```

// INGAME - FALLING AWAIT (ZERO) 1F
dn_cnt = G_DN_ACC
dn_max = ldnc(g_level)
gosub *dept
gosub *mov_dn
g_stat = GSTAT_G_N
} else : if (g_stat = GSTAT_G_N) {
// INGAME - FALLING AWAIT (NORMAL)
repeat 7
    args(0) = cnt : args(1) = cnt : gosub *key_pshdn
loop
getkey kgot(KEY_ACC), kcode(KEY_ACC)
if (kgot(KEY_ACC) = 1) {
    if (dn_cnt > G_DN_ACC) {
        dn_cnt = G_DN_ACC
    }
}
dn_cnt-
if (dn_cnt = 0) {
    dn_cnt = dn_max
    gosub *mov_dn
}
} else : if (g_stat = GSTAT_G_F) {
// INGAME - FALLING AWAIT (FINAL)
repeat 7
    args(0) = cnt : args(1) = cnt : gosub *key_pshdn
loop
dn_cnt-
if (dn_cnt = 0) {
    gosub *dr_ctrl
    gosub *judge
    if (rets(4) > 0) {
        fx_cnt = 20
        g_stat = GSTAT_G_X
        repeat 5
            arg2(cnt) = rets(cnt)
        loop
    } else {
        repeat 5
            args(cnt) = rets(cnt)
        loop
        gosub *del_drop
        g_stat = GSTAT_G_S
        if (hzd = TRUE) : gosub *calc_dmg
        hzd = FALSE
        dn_cnt = ldnc(g_level)
        dn_max = ldnc(g_level)
    }
    i_f2 = FALSE
}
} else : if (g_stat = GSTAT_G_X) {
// INGAME - DELETE EFFECT
fx_cnt -= 1
if (fx_cnt <= 0) {
    repeat 5
        args(cnt) = arg2(cnt)
    loop
    gosub *calc_score
    gosub *del_drop
    g_stat = GSTAT_G_S
    if (hzd = TRUE) : gosub *calc_dmg
    hzd = FALSE
}
}

```



```

        if ((lnxt(g_mode) > 0) & (l_loc >= lnxt(g_mode))) {
            l_loc = l_loc - lnxt(g_mode)
            g_level+
            if (g_level = llim(g_mode)) {
                g_stat = GSTAT_P_S
                g_level-
            }
        }
        dn_cnt = ldn(g_level)
        dn_max = ldn(g_level)
    }
    i_f2 = FALSE
}

```

// CONTROL END

// DRAW

gsel 5

pos 0, 0

gcopy 1, 0, 0, 800, 900

// BOX SHADOW

color 95, 95, 95

if ((g_stat > 0) | (kgot(KEY_ACT) = 1)) {

boxf 55, 104, 55+95, 104+95

boxf 55, 208, 55+95, 208+95

boxf 55, 312, 55+95, 312+95

boxf 55, 416, 55+95, 416+95

boxf 55, 520, 55+95, 520+95

boxf 503, 520, 503+95, 520+95

boxf 503, 624, 503+95, 624+95

boxf 503, 728, 503+95, 728+95

}

boxf 159, 208, 159+335, 208+671

boxf 159, 53, 159+335, 53+146

boxf 503, 53, 503+236, 53+457

// PANEL

pos 156, 50

gcopy 3, 0, 0, 336, 147

pos 500, 50

gcopy 4, 0, 0, 237, 458

// NEXT

gmode 7

if (g_stat > GSTAT_P_N) {

pos 54, 50

if (g_stat = GSTAT_G_F) {

gcopy 2, 48, 0, 100, 50

} else {

gcopy 2, 48, 50, 100, 50

}

}

// TIME

if (sgsw(SEG_CLOCK) = TRUE) {

g_v1 = gettime(4)

g_v2 = gettime(5)

g_v3 = gettime(6)

pos 226, 55

if (g_v1 > 9) {

gcopy 1, 800, g_v1 / 10 * 40, 20, 40

```

    } else {
        gcopy 1, 800, 400, 20, 40
    }
    pos 246, 55
    gcopy 1, 800, g_v1 ¥ 10 * 40, 20, 40
    pos 286, 55
    gcopy 1, 800, g_v2 / 10 * 40, 20, 40
    pos 306, 55
    gcopy 1, 800, g_v2 ¥ 10 * 40, 20, 40
    pos 346, 55
    gcopy 1, 800, g_v3 / 10 * 40, 20, 40
    pos 366, 55
    gcopy 1, 800, g_v3 ¥ 10 * 40, 20, 40
} else {
    args(1) = 2
    args(2) = 226
    args(3) = 55
    gosub *dr_seg2
    args(2) = 286
    gosub *dr_seg2
    args(2) = 346
    gosub *dr_seg2
}

// LEVEL
if (sgsw(SEG_LEVEL) = TRUE) {
    args(0) = g_level + 1
    args(1) = 3
    args(2) = 226
    args(3) = 103
    gosub *dr_seg
} else {
    args(1) = 3
    args(2) = 226
    args(3) = 103
    gosub *dr_seg2
}

// WAIT
if (sgsw(SEG_WAIT) = TRUE) {
    g_v1 = dn_cnt * 16
    pos 372, 103
    if (g_v1 > 9999) {
        gcopy 1, 800, g_v1 / 10000 * 40, 20, 40
    } else {
        gcopy 1, 800, 400, 20, 40
    }
    g_v1 = g_v1 ¥ 10000
    pos 392, 103
    gcopy 1, 800, g_v1 / 1000 * 40, 20, 40
    g_v1 = g_v1 ¥ 1000
    pos 432, 103
    gcopy 1, 800, g_v1 / 100 * 40, 20, 40
    g_v1 = g_v1 ¥ 100
    pos 452, 103
    gcopy 1, 800, g_v1 / 10 * 40, 20, 40
} else {
    args(1) = 2
    args(2) = 372
    args(3) = 103
    gosub *dr_seg2
    args(2) = 432

```

```

    gosub *dr_seg2
}

// HP
if (sgsw(SEG_HP) = TRUE) {
    args(0) = hp
    args(1) = 5
    args(2) = 226
    args(3) = 154
    gosub *dr_seg
    pos 346, 154
    gcopy 1, 800, 40, 20, 40
    pos 366, 154
    gcopy 1, 800, 0, 20, 40
    pos 386, 154
    gcopy 1, 800, 0, 20, 40
    pos 406, 154
    gcopy 1, 800, 0, 20, 40
    pos 426, 154
    gcopy 1, 800, 0, 20, 40
} else {
    args(1) = 5
    args(2) = 226
    args(3) = 154
    gosub *dr_seg2
    args(2) = 346
    gosub *dr_seg2
}

// HP BAR
pos 226, 148
gcopy 1, 0, 906, 220, 6
if (sgsw(BAR_HP) > 0) {
    pos 226, 148
    gcopy 1, 0, 912, 220, 6
}

// SCORE
if (sgsw(SEG_SCORE) = TRUE) {
    args(0) = score
    args(1) = 8
    args(2) = 569
    args(3) = 56
    gosub *dr_seg
} else {
    args(1) = 8
    args(2) = 569
    args(3) = 56
    gosub *dr_seg2
}

// LINES
if (sgsw(SEG_LINES) = TRUE) {
    args(0) = l_cnt
    args(1) = 4
    args(2) = 569
    args(3) = 108
    gosub *dr_seg
} else {
    args(1) = 4
    args(2) = 569
    args(3) = 108

```

```

    gosub *dr_seg2
}

// BG
if (sgsw(SEG_COUNT) = TRUE) {
    args(0) = cnt(C_BG)
    args(1) = 4
    args(2) = 569
    args(3) = 163
    gosub *dr_seg
} else {
    args(1) = 4
    args(2) = 569
    args(3) = 163
    gosub *dr_seg2
}

// T BACKGROUND
if (sgsw(ICO_COUNT) = 2) {
    if (f_v4 = 0) {
        f_v1 += 8
        if (f_v1 >= 255) {
            f_v1 = 255
            f_v4 = 1
        }
    } else : if (f_v4 = 1) {
        f_v2 -= 8
        if (f_v2 <= 0) {
            f_v2 = 0
            f_v4 = 2
        }
    } else : if (f_v4 = 2) {
        f_v3 += 8
        if (f_v3 >= 255) {
            f_v3 = 255
            f_v4 = 3
        }
    } else : if (f_v4 = 3) {
        f_v1 -= 8
        if (f_v1 <= 0) {
            f_v1 = 0
            f_v4 = 4
        }
    } else : if (f_v4 = 4) {
        f_v2 += 8
        if (f_v2 >= 255) {
            f_v2 = 255
            f_v4 = 5
        }
    } else {
        f_v3 -= 8
        if (f_v3 <= 0) {
            f_v3 = 0
            f_v4 = 0
        }
    }
}
color f_v1, f_v2, f_v3
} else : if (sgsw(ICO_COUNT) = 1) {
    if (f_v6 = 0) {
        f_v5 += 2
        if (f_v5 >= 255) {
            f_v5 = 255

```

```

        f_v6 = 1
    }
} else {
    f_v5 -= 2
    if (f_v5 <= 0) {
        f_v5 = 0
        f_v6 = 0
    }
}
color f_v5, f_v5, f_v5
} else {
    color 0, 0, 0
}
boxf 505, 211, 552, 258
boxf 505, 263, 552, 310
pos 505, 211
gcopy 2, 48, 150, 48, 48
pos 505, 263
gcopy 2, 48, 198, 48, 48

if (sgsw(SEG_COUNT) = 1) {
    // T1
    args(0) = cnt(C_T1)
    args(1) = 4
    args(2) = 569
    args(3) = 215
    gosub *dr_seg

    // T2
    args(0) = cnt(C_TG)
    args(1) = 4
    args(2) = 569
    args(3) = 267
    gosub *dr_seg
} else {
    args(1) = 4
    args(2) = 569
    args(3) = 215
    gosub *dr_seg2

    args(2) = 569
    args(3) = 267
    gosub *dr_seg2
}

// KEEP
if (g_stat > 0) {
    pos 500, 829
    gcopy 2, 48, 100, 100, 50
}

// CONTROL
if (g_stat > 0) {
    repeat 12
        pos kposx(cnt), kposy(cnt) - 24
        gcopy 2, 560, cposg(cnt), 24, 24
    loop
} else {
    repeat 12
        pos kposx(cnt), kposy(cnt) - 24
        gcopy 2, 560, cposp(cnt), 24, 24
    loop
}

```

```

}
gmode 0
repeat 12
    pos kposx(cnt), kposy(cnt)
    if (kgot(cnt) = 1) {
        gcopy 2, 632, kcle(cnt), 24, 24
    } else {
        gcopy 2, 608, kcle(cnt), 24, 24
    }
}
loop
if ((g_stat ! GSTAT_P_N) & (g_stat >= GSTAT_P_P)) {
    pos 535, 340
    if (g_mode = 0) {
        gcopy 3, 336, 0, 167, 24
    } else : if (g_mode = 1) {
        gcopy 3, 336, 24, 167, 24
    } else : if (g_mode = 2) {
        gcopy 3, 336, 48, 167, 24
    } else : if (g_mode = 3) {
        gcopy 3, 336, 72, 167, 24
    } else : if (g_mode = 4) {
        gcopy 3, 336, 96, 167, 24
    } else : if (g_mode = 5) {
        gcopy 3, 336, 120, 167, 24
    }
}

if (i_f2 = TRUE) {
    gosub *dr_ctrl
}
if ((g_stat > 0) | (kgot(KEY_ACT) = 1)) {
    gosub *dr_board
} else {
    gosub *dr_menu
}
if (g_stat = GSTAT_G_X) {
    gosub *dr_del
}
if (i_f2 = TRUE) {
    gosub *clr_ctrl
} else {
    i_f2 = TRUE
}

```

// SYSTEM BODY END

```

gsel 0
pos 0, 0
gzoom dwsizex, dwsizey, 5, 0, 0, 800, 900

```

```

await 16
goto *main

```

*init

```

c_sw = 0
c_x = 0 : c_y = 0
i_f2 = TRUE
dn_cnt = 0
dn_max = 1
qg_isvacant = TRUE
s_cnt = 0 : t_cnt = 0
l_cnt = 0 : l_loc = 0

```

```

s_sw = 0
p_stat = GSTAT_P_N

score = 0
hp = PLAYER_HP_X
gosub *barcolor
repeat 8
    ccnt(cnt) = 0
loop
repeat 608
    cells(cnt) = C_VA
loop
return

```

```

*add_wait
// ARGS: %4 PLUS
dn_cnt += args(4)
if (dn_cnt > dn_max) {
    dn_cnt = dn_max
}
return

```

```

*key_pshdn
// ARGS: %0 CDE %1 GOTO
getkey kgot(args(0)), kcode(args(0))
if (kgot(args(0)) = 1) {
    if (kflg(args(0)) = 0) {
        kflg(args(0)) = 1
        gosub klab(args(1))
    }
} else {
    if (kflg(args(0)) = 1) {
        kflg(args(0)) = 0
    }
}
return

```

```

*key_stop
repeat 8
    kflg(cnt) = 1
loop
return

```

```

*k_mvl
args(0) = 0
gosub *mov_l
return

```

```

*k_mvr
args(0) = 0
gosub *mov_r
return

```

```

*k_rtl
args(0) = 0
gosub *rot_l
return

```

```

*k_rtr
args(0) = 0
gosub *rot_r
return

```

*k_psh

```
gosub *push
return
```

*k_pop

```
gosub *pop
return
```

*k_dn

```
args(0) = 0
gosub *mov_dn
dn_max = G_DN_DRP
dn_cnt = G_DN_DRP
return
```

*k_cv1

```
if (g_stat = GSTAT_T_L2_N) {
    cur_h-
    if (cur_h = -1) : cur_h = 19
    g_level = cur_h
} else : if (g_stat = GSTAT_T_C2_SEG) {
    if (cfg_sel = 1) {
        cur_h-
        if (cur_h = -1) : cur_h = sg1m(cur_v) - 1
        sgsw(cur_v) = cur_h
        if (cur_v = BAR_HP) : gosub *barcolor
    }
} else : if (g_stat = GSTAT_T_C2_GRA) {
    if (cfg_sel = 1) {
        cur_h-
        if (cur_h = -1) : cur_h = sclm(cur_v) - 1
        scsw(cur_v) = cur_h
        if (cur_v = SCR_CSIZE) {
            screen 0, wsizeh(cur_h), wsizey(cur_h)
            color 0, 0, 0
            boxf
            dwsizex = wsizeh(cur_h)
            dwsizey = wsizey(cur_h)
            await 50
            gsel 5
        }
    }
} else : if (g_stat = GSTAT_T_C2_SOU) {
    if ((kgot(KEY_NOP) = 1) & (cfg_sel = 1)) {
        cur_h-
        if (cur_h < 0) : cur_h = 0
        sosw(cur_v) = cur_h
    }
} else : if (g_stat = GSTAT_T_C2_CTR) {
    if (cfg_sel = 1) {
        cur_h-
        if (cur_h = -1) : cur_h = ct1m(cur_v) - 1
    }
} else : if (g_stat = GSTAT_T_I2_ABO) {
    cur_h-
    if (cur_h = -1) : cur_h = 0
}
return
```

*k_cvr

```
if (g_stat = GSTAT_T_L2_N) {
```



```

cur_h+
if (cur_h = 20) : cur_h = 0
g_level = cur_h
} else : if (g_stat = GSTAT_T_C2_SEG) {
  if (cfg_sel = 1) {
    cur_h+
    if (cur_h = sg1m(cur_v)) : cur_h = 0
    sgsw(cur_v) = cur_h
    if (cur_v = BAR_HP) : gosub *barcolor
  }
} else : if (g_stat = GSTAT_T_C2_GRA) {
  if (cfg_sel = 1) {
    cur_h+
    if (cur_h = sclm(cur_v)) : cur_h = 0
    scsw(cur_v) = cur_h
    if (cur_v = SCR_CSIZE) {
      screen 0, wsize(cur_h), wsize(cur_h)
      color 0, 0, 0
      boxf
      dwsizex = wsize(cur_h)
      dwsizey = wsize(cur_h)
      await 50
      gsel 5
    }
  }
} else : if (g_stat = GSTAT_T_C2_SOU) {
  if ((kgot(KEY_NOP) = 1) & (cfg_sel = 1)) {
    cur_h+
    if (cur_h > 100) : cur_h = 100
    sosw(cur_v) = cur_h
  }
} else : if (g_stat = GSTAT_T_C2_CTR) {
  if (cfg_sel = 1) {
    cur_h+
    if (cur_h = ct1m(cur_v)) : cur_h = 0
  }
} else : if (g_stat = GSTAT_T_I2_ABO) {
  cur_h+
  if (cur_h = 3) : cur_h = 2
}
return

```

*k_rst

```

if (cfg_sel = 1) {
  if (g_stat = GSTAT_T_C2_SEG) {
    cur_h = sgdf(cur_v)
    sgsw(cur_v) = sgdf(cur_v)
    if (cur_v = BAR_HP) : gosub *barcolor
  } else : if (g_stat = GSTAT_T_C2_GRA) {
    cur_h = scdf(cur_v)
    scsw(cur_v) = scdf(cur_v)
    if (cur_v = SCR_CSIZE) {
      screen 0, wsize(cur_h), wsize(cur_h)
      color 0, 0, 0
      boxf
      dwsizex = wsize(cur_h)
      dwsizey = wsize(cur_h)
      await 50
      gsel 5
    }
  } else : if (g_stat = GSTAT_T_C2_SOU) {
    cur_h = sodf(cur_v)
  }
}

```

```

        sosw(cur_v) = sodf(cur_v)
    } else : if (g_stat = GSTAT_T_C2_CTR) {
        cur_h = ctdf(cur_v)
    }
}
return

```

```

*k_nop
return

```

```

*k_cvu
if (g_stat = GSTAT_P_N) {
    cur_v-
    if (cur_v = -1) : cur_v = 3
} else : if (g_stat = GSTAT_P_P) {
    cur_v-
    if (cur_v = -1) : cur_v = 4 // RESUME CONFIG TITLE INFO QUIT
} else : if (g_stat = GSTAT_P_D) {
    cur_v-
    if (cur_v = -1) : cur_v = 1 // TITLE QUIT
} else : if (g_stat = GSTAT_P_S) {
    cur_v-
    if (cur_v = -1) : cur_v = 2 // STAFF TITLE QUIT
} else : if (g_stat = GSTAT_T_L1) {
    cur_v-
    if (cur_v = -1) {
        if (ir_open = 1) {
            cur_v = 5
        } else {
            cur_v = 4
        }
    }
}
g_level = lsta(cur_v)
g_mode = cur_v
} else : if (g_stat = GSTAT_T_C1) {
    cur_v-
    if (cur_v = -1) : cur_v = 3
} else : if (g_stat = GSTAT_T_I1) {
    cur_v-
    if (cur_v = -1) : cur_v = 3
} else : if (g_stat = GSTAT_T_C2_SEG) {
    if (cfg_sel = 0) {
        cur_v-
        if (cur_v = -1) : cur_v = 8
    }
} else : if (g_stat = GSTAT_T_C2_GRA) {
    if (cfg_sel = 0) {
        cur_v-
        if (cur_v = -1) : cur_v = 0
    }
}
return

```

```

*k_cxl
if (g_stat > GSTAT_P_P) {
    cur_v = 0
} else : if ((g_stat & GSTAT_P_N_N) = GSTAT_P_N_N) {
    if (p_stat > 0) {
        g_stat = GSTAT_P_P
    } else {
        g_stat = GSTAT_P_N
    }
}

```

```

    ir_open = 0
    cur_v = 0
    g_level = 0
    g_mode = 0
} else : if ((g_stat & GSTAT_T_L1_N) = GSTAT_T_L1_N) {
    g_stat = GSTAT_T_L1
    if (cur_h = 17) : ir_open = 1
    cur_v = 0
    cur_h = 0
    g_level = 0
    g_mode = 0
} else : if ((g_stat & GSTAT_T_C1_N) = GSTAT_T_C1_N) {
    if (cfg_sel = 0) {
        g_stat = GSTAT_T_C1
        cur_v = 0
        cur_h = 0
    } else {
        if (g_stat = GSTAT_T_C2_SEG) {
            sgsw(cur_v) = cur_k
            if (cur_v = BAR_HP) : gosub *barcolor
        } else : if (g_stat = GSTAT_T_C2_GRA) {
            scsw(cur_v) = cur_k
        } else : if (g_stat = GSTAT_T_C2_SOU) {
            sosw(cur_v) = cur_k
        } else : if (g_stat = GSTAT_T_C2_CTR) {
            ctsw(cur_v) = cur_k
        }
        cfg_sel = 0
    }
} else : if ((g_stat & GSTAT_T_I1_N) = GSTAT_T_I1_N) {
    g_stat = GSTAT_T_I1
    if (p_stat = GSTAT_P_S) {
        g_level = 0
        g_mode = 0
        gosub *init
        g_stat = GSTAT_P_N
    }
    cur_v = 0
    cur_h = 0
}
return

```

*k_dec

```

if (g_stat = GSTAT_P_N) {
    p_stat = g_stat
    g_stat = GSTAT_P_N_N | cur_v
    cur_v = 0
} else : if (g_stat = GSTAT_P_P) {
    if (cur_v = 0) {
        g_stat = p_stat
        cur_h = 0
    } else : if (cur_v = 1) {
        g_stat = GSTAT_T_C1
        cur_v = 0
    } else : if (cur_v = 2) {
        g_stat = GSTAT_P_N
        g_level = 0
        g_mode = 0
        gosub *init
    } else : if (cur_v = 3) {
        g_stat = GSTAT_T_I1
        cur_v = 0
    }
}

```

```

    } else {
        g_stat = GSTAT_T_Q
    }
} else : if (g_stat = GSTAT_P_D) {
    if (cur_v = 0) {
        // TITLE
        g_stat = GSTAT_P_N
        g_level = 0
        g_mode = 0
        gosub *init
    } else {
        g_stat = GSTAT_T_Q
    }
} else : if (g_stat = GSTAT_P_S) {
    if (cur_v = 0) {
        // STAFF
        g_stat = GSTAT_T_I2_STA
        p_stat = GSTAT_P_S
    } else : if (cur_v = 1) {
        // TITLE
        g_stat = GSTAT_P_N
        g_level = 0
        g_mode = 0
        gosub *init
        cur_v = 0
    } else {
        g_stat = GSTAT_T_Q
    }
} else : if (g_stat = GSTAT_T_L1) {
    g_stat = GSTAT_T_L1_N | cur_v
    cur_h = 0
} else : if (g_stat = GSTAT_T_C1) {
    g_stat = GSTAT_T_C1_N | cur_v
    cur_v = 0
} else : if (g_stat = GSTAT_T_I1) {
    g_stat = GSTAT_T_I1_N | cur_v
    cur_v = 0
} else : if ((g_stat & GSTAT_T_L1_N) = GSTAT_T_L1_N) {
    cur_v = 0
    cur_h = 0
    ir_open = 0
    g_stat = GSTAT_G_S // -> GAME MODE
    gosub *init
    gosub *gen_i
} else : if ((g_stat & GSTAT_T_C1_N) = GSTAT_T_C1_N) {
    if (cfg_sel = 1) {
        if (g_stat = GSTAT_T_C2_SEG) {
            sgsw(cur_v) = cur_h
            if (cur_v = BAR_HP) : gosub *barcolor
        } else : if (g_stat = GSTAT_T_C2_GRA) {
            scsw(cur_v) = cur_h
        } else : if (g_stat = GSTAT_T_C2_SOU) {
        } else : if (g_stat = GSTAT_T_C2_CTR) {
            ctsw(cur_v) = cur_h
            if (cur_h = 0) {
                repeat 12
                    kcle(cnt) = kcla(cnt)
                    kode(cnt) = koda(cnt)
                loop
            } else {
                repeat 12
                    kcle(cnt) = kclb(cnt)
            }
        }
    }
}

```

```

        kode(cnt) = kodb(cnt)
    loop
    }
}
cfg_sel = 0
} else {
    if (g_stat = GSTAT_T_C2_SEG) {
        cur_h = sgsw(cur_v)
        cur_k = sgsw(cur_v)
    } else : if (g_stat = GSTAT_T_C2_GRA) {
        cur_h = scsw(cur_v)
        cur_k = scsw(cur_v)
    } else : if (g_stat = GSTAT_T_C2_SOU) {
        cur_h = sosw(cur_v)
        cur_k = sosw(cur_v)
    } else : if (g_stat = GSTAT_T_C2_CTR) {
        cur_h = ctsw(cur_v)
        cur_k = ctsw(cur_v)
    }
    cfg_sel = 1
}
} else : if ((g_stat & GSTAT_T_I1_N) = GSTAT_T_I1_N) {
    g_stat = GSTAT_T_I1
    if (p_stat = GSTAT_P_S) {
        g_level = 0
        g_mode = 0
        gosub *init
        g_stat = GSTAT_P_N
    }
    cur_v = 0
    cur_h = 0
}
return

```

*k_cvd

```

if (g_stat = GSTAT_P_N) {
    cur_v+
    if (cur_v = 4) : cur_v = 0
} else : if (g_stat = GSTAT_P_P) {
    cur_v+
    if (cur_v = 5) : cur_v = 0 // RESUMT CONFIG TITLE INFO QUIT
} else : if (g_stat = GSTAT_P_D) {
    cur_v+
    if (cur_v = 2) : cur_v = 0 // TITLE QUIT
} else : if (g_stat = GSTAT_P_S) {
    cur_v+
    if (cur_v = 3) : cur_v = 0 // STAFF TITLE QUIT
} else : if (g_stat = GSTAT_T_L1) {
    cur_v+
    if (ir_open = 1) {
        if (cur_v = 6) : cur_v = 0
    } else {
        if (cur_v = 5) : cur_v = 0
    }
    g_level = lsta(cur_v)
    g_mode = cur_v
} else : if (g_stat = GSTAT_T_C1) {
    cur_v+
    if (cur_v = 4) : cur_v = 0
} else : if (g_stat = GSTAT_T_I1) {
    cur_v+
    if (cur_v = 4) : cur_v = 0
}

```

```

} else : if (g_stat = GSTAT_T_C2_SEG) {
    if (cfg_sel = 0) {
        cur_v+
        if (cur_v = 9) : cur_v = 0
    }
} else : if (g_stat = GSTAT_T_C2_GRA) {
    if (cfg_sel = 0) {
        cur_v+
        if (cur_v = 1) : cur_v = 0
    }
}
return

```

```

*k_pa
if (g_stat > 0) {
    p_stat = g_stat
    g_stat = GSTAT_P_P
} else : if (g_stat <= GSTAT_P_P) {
    if (p_stat > 0) {
        g_stat = p_stat
        cur_h = 0
        cur_v = 0
    }
}
return

```

```

*k_scr
bmpsave strf("%02d", gettime(4)) + strf("%02d", gettime(5)) + strf("%02d", gettime(6)) + ". bmp"
return

```

```

*k_log
i_v1 = gettime(4)
i_v2 = gettime(5)
i_v3 = gettime(6)
s_v1 = ""
s_v1 += (strf("%02d", i_v1) + " " + strf("%02d", i_v2) + " " + strf("%02d", i_v3) + "¥n")
s_v1 += (strf("%02d", g_level) + " " + strf("%01d", g_mode) + "¥n")
s_v1 += (str(score) + "¥n")
s_v1 += (str(l_cnt) + "¥n")
s_v1 += str(ccnt(C_0C))
repeat 6, 2
    s_v1 += (" " + str(ccnt(cnt)))
loop
s_v1 += "¥n"
i_v4 = 0
repeat 8
    s_v1 += strf("%08X", cells(i_v4))
    i_v4+
    repeat 3
        s_v1 += (" " + strf("%08X", cells(i_v4)))
        i_v4+
    loop
    s_v1 += "¥n"
loop
repeat 32
    s_v1 += strf("%08X", cells(i_v4))
    i_v4+
    repeat 13
        s_v1 += (" " + strf("%08X", cells(i_v4)))
        i_v4+
    loop
    s_v1 += "¥n"

```

```

loop
repeat 32
    s_v1 += strf("%08X", cells(i_v4))
    i_v4+
    repeat 3
        s_v1 += (" " + strf("%08X", cells(i_v4)))
        i_v4+
    loop
    s_v1 += "\n"
loop
notesel s_v1
notesave strf("%02d", gettime(4)) + strf("%02d", gettime(5)) + strf("%02d", gettime(6)) + ".txt"
noteunsel
return

```

```

*k_act
return

```

```

*rot_r
// ARGS: %O MUST 0
c_sw = cells(CONTROL)
cells(CONTROL) = cells(CONTROL + 12)
cells(CONTROL + 12) = cells(CONTROL + 15)
cells(CONTROL + 15) = cells(CONTROL + 3)
cells(CONTROL + 3) = c_sw
c_sw = cells(CONTROL + 1)
cells(CONTROL + 1) = cells(CONTROL + 8)
cells(CONTROL + 8) = cells(CONTROL + 14)
cells(CONTROL + 14) = cells(CONTROL + 7)
cells(CONTROL + 7) = c_sw
c_sw = cells(CONTROL + 2)
cells(CONTROL + 2) = cells(CONTROL + 4)
cells(CONTROL + 4) = cells(CONTROL + 13)
cells(CONTROL + 13) = cells(CONTROL + 11)
cells(CONTROL + 11) = c_sw
c_sw = cells(CONTROL + 5)
cells(CONTROL + 5) = cells(CONTROL + 9)
cells(CONTROL + 9) = cells(CONTROL + 10)
cells(CONTROL + 10) = cells(CONTROL + 6)
cells(CONTROL + 6) = c_sw
rets(0) = 0
if (args(0) = 0) : gosub *req_mr
if (rets(0) > 0) {
    args(0) = 1
    gosub *rot_l
} else : if (args(0) = 0) {
    gosub *check
    args(4) = G_DP_ROT
    gosub *add_wait
}
return

```

```

*rot_l
// ARGS: %O MUST 0
c_sw = cells(CONTROL)
cells(CONTROL) = cells(CONTROL + 3)
cells(CONTROL + 3) = cells(CONTROL + 15)
cells(CONTROL + 15) = cells(CONTROL + 12)
cells(CONTROL + 12) = c_sw
c_sw = cells(CONTROL + 1)
cells(CONTROL + 1) = cells(CONTROL + 7)
cells(CONTROL + 7) = cells(CONTROL + 14)

```

```

cells(CONTROL + 14) = cells(CONTROL + 8)
cells(CONTROL + 8) = c_sw
c_sw = cells(CONTROL + 2)
cells(CONTROL + 2) = cells(CONTROL + 11)
cells(CONTROL + 11) = cells(CONTROL + 13)
cells(CONTROL + 13) = cells(CONTROL + 4)
cells(CONTROL + 4) = c_sw
c_sw = cells(CONTROL + 5)
cells(CONTROL + 5) = cells(CONTROL + 6)
cells(CONTROL + 6) = cells(CONTROL + 10)
cells(CONTROL + 10) = cells(CONTROL + 9)
cells(CONTROL + 9) = c_sw
rets(0) = 0
if (args(0) = 0) : gosub *req_mr
if (rets(0) > 0) {
    args(0) = 1
    gosub *rot_r
} else : if (args(0) = 0) {
    gosub *check
    args(4) = G_DP_ROT
    gosub *add_wait
}
return

```

```

*mov_r
// ARGS: %0 MUST 0
c_x += 1
rets(0) = 0
if (args(0) = 0) : gosub *req_mr
if (rets(0) > 0) {
    args(0) = 1
    gosub *mov_l
} else : if (args(0) = 0) {
    gosub *check
    args(4) = G_DP_MOV
    gosub *add_wait
}
return

```

```

*mov_l
// ARGS: %0 MUST 0
c_x -= 1
rets(0) = 0
if (args(0) = 0) : gosub *req_mr
if (rets(0) > 0) {
    args(0) = 1
    gosub *mov_r
} else : if (args(0) = 0) {
    gosub *check
    args(4) = G_DP_MOV
    gosub *add_wait
}
return

```

```

*check
c_y+
gosub *req_mr
c_y-
if (rets(0) > 0) {
    dn_cnt = ldnc(g_level)
    dn_max = ldnc(g_level)
    g_stat = GSTAT_G_F
}

```



```

} else {
    if (g_stat = GSTAT_G_F) {
        dn_cnt = ldnc(g_level)
        dn_max = ldnc(g_level)
        g_stat = GSTAT_G_N
    }
}
return

```

**mov_dn*

```

if (g_stat = GSTAT_G_F) {
    if (dn_max = G_DN_DRP) {
        dn_max = ldnc(g_level)
        dn_cnt = ldnc(g_level)
    }
} else {
    c_y+
    rets(0) = 0
    gosub *req_mr
    if (rets(0) ! C_VA) {
        c_y-
        g_stat = GSTAT_G_F
        dn_cnt = ldnc(g_level)
        dn_max = ldnc(g_level)
    } else {
        dn_cnt = dn_max
    }
}
return

```

**req_mr*

```

// RETS: %0 RESULT(D/A 1~/0)
i_cnt = CONTROL : i_v2 = c_y
rets(0) = 0
repeat 4
    i_v1 = c_x
    if ((i_v2 >= 0) & (i_v2 < 32)) {
        repeat 4
            if ((i_v1 >= 0) & (i_v1 < 14)) {
                if (cells(i_cnt) ! C_VA) : rets(0) += cells(BOARD_H + i_v2 * OPN_W + i_v1) & 0x000000FF
            } else {
                if (cells(i_cnt) ! C_VA) : rets(0) += 1
            }
            i_cnt+ : i_v1+
        loop
    } else {
        repeat 4
            if (cells(i_cnt) ! C_VA) : rets(0) += 1
            i_cnt+
        loop
        i_v1 += 4
    }
    i_v2+
loop
return

```

**judge*

```

// RETS: %0 %1 %2 %3 RESULT(1 DEL) %4 CNT
rets(0) = 0 : rets(1) = 0 : rets(2) = 0 : rets(3) = 0
i_idx = BOARD_H + c_y * OPN_W
i_v1 = 0
repeat 4

```

```

i_flg = 1
if (c_y + cnt < 32) {
    repeat OPN_W
        if (cells(i_idx) = C_VA) : i_flg = 0
        if (cells(i_idx) = C_BM) : cells(i_idx) = C_BG
        if ((cells(i_idx) & 0x000000FF) = C_T2) : cells(i_idx) = C_T2
        i_idx+
    loop
} else {
    i_flg = 0
}
if (i_flg = 1) {
    rets(cnt) = 1
    l_cnt+
    l_loc+
    i_v1+
}
loop
rets(4) = i_v1
return

```

*gen_c1

```

// RETS: %0 V/0
i_v1 = (rnd(256) << 8) | rnd(256)
i_v2 = rnd(16)
i_cnt = 0
repeat 16
    if (i_v1 & (1 << cnt)) : i_cnt+
loop
if(i_cnt < 8) {
    i_v1 = i_v1 | 1632
}
if(i_v2 < 7) {
    i_v2 = rnd(256)
    if (i_v2 < 16) {
        rets(0) = 65535
    } else : if (i_v2 < 40) {
        rets(0) = 864
    } else : if (i_v2 < 64) {
        rets(0) = 3168
    } else : if (i_v2 < 96) {
        rets(0) = 8800
    } else : if (i_v2 < 128) {
        rets(0) = 17504
    } else : if (i_v2 < 168) {
        rets(0) = 1824
    } else : if (i_v2 < 208) {
        rets(0) = 1632
    } else {
        rets(0) = 8738
    }
} else : if(i_v2 < 10) {
    i_v2 = rnd(128)
    if (i_v2 < 24) {
        rets(0) = 26214
    } else : if (i_v2 < 48) {
        rets(0) = 1911
    } else : if (i_v2 < 88) {
        rets(0) = 8736
    } else {
        rets(0) = 608
    }
}

```

```

} else : if (i_v2 < 15) {
    i_v2 = rnd(4)
    if (i_v2 = 0) {
        i_v1 = i_v1 & 61152 // 1824
    } else : if (i_v2 = 1) {
        i_v1 = i_v1 & 30576 // 8738
    } else : if (i_v2 = 2) {
        i_v1 = i_v1 & 3822 // 1632
    } else {
        i_v1 = i_v1 & 1911 // 1824
    }
    rets(0) = i_v1
} else {
    rets(0) = i_v1 & 28662 // 1824
}
return

```

*gen_c2

```

// ARGS: %0 V/0 %1 OFFSET
i_v2 = rnd(256)
if(i_v2 < 128) {
    i_v3 = C_BM
} else {
    i_v3 = C_T2
}
repeat 16
    if(args(0) & (1 << cnt)) {
        i_v1 = rnd(256)
        if(i_v1 < ltoc(g_level)) {
            cells(args(1) + cnt) = C_OC
        } else : if(i_v1 < ltt1(g_level)) {
            gosub *selcolor
            cells(args(1) + cnt) = C_T1 | rets(0)
        } else {
            rets(0) = 0
            if (i_v3 = C_T2) : gosub *selcolor
            cells(args(1) + cnt) = i_v3 | rets(0)
        }
    } else {
        cells(args(1) + cnt) = C_VA
    }
loop
return

```

*selcolor

```

// RETS: %0 COLOR
i_v4 = rnd(6)
i_v5 = rnd(256)
rets(0) = 0
if (i_v4 = 0) {
    rets(0) = (255 << 24) | (i_v5 << 16) // R g^ *
} else : if (i_v4 = 1) {
    rets(0) = (i_v5 << 24) | (255 << 16) // r, G *
} else : if (i_v4 = 2) {
    rets(0) = (255 << 16) | (i_v5 << 8) // * G b^
} else : if (i_v4 = 3) {
    rets(0) = (i_v5 << 16) | (255 << 8) // * g, B
} else : if (i_v4 = 4) {
    rets(0) = (i_v5 << 24) | (255 << 8) // r^ * B
} else {
    rets(0) = (255 << 24) | (i_v5 << 8) // R * b,
}

```

```
return
```

```
*barcolor
```

```
i_v1 = hp / 2000
i_v2 = hp ¥ 2000 * 16 / 125
gsel 1
gmode 7
if (sgsw(BAR_HP) = 2) {
    if (i_v1 = 5) {
        color 0, 0, 255 // B
    } else : if (i_v1 = 4) {
        color 0, 255 - i_v2, 255 // B -> C
    } else : if (i_v1 = 3) {
        color 0, 255, i_v2 // C -> G
    } else : if (i_v1 = 2) {
        color 255 - i_v2, 255, 0 // G -> Y
    } else : if (i_v1 = 1) {
        color 255, i_v2, 0 // Y -> R
    } else {
        color i_v2, 0, 0 // R -> K
    }
} else {
    color 63, 63, 63
}
boxf 0, 912, 219, 917
pos 0, 912
gcopy 1, 0, 900, 220, 6
gmode 0
color 255, 255, 255
boxf 220, 912, 439, 917
color 0, 0, 0
if (hp < 10000) {
    boxf 220 + (hp * 220 / 10000), 912, 439, 917
}
return
```

```
*gen_i
```

```
gosub *gen_c1
args(0) = rets(0) : args(1) = QUEUE_0
gosub *gen_c2
gosub *gen_c1
args(0) = rets(0) : args(1) = QUEUE_1
gosub *gen_c2
gosub *gen_c1
args(0) = rets(0) : args(1) = QUEUE_2
gosub *gen_c2
gosub *gen_c1
args(0) = rets(0) : args(1) = QUEUE_3
gosub *gen_c2
gosub *gen_c1
args(0) = rets(0) : args(1) = QUEUE_4
gosub *gen_c2
return
```

```
*dept
```

```
c_x = rnd(11) : c_y = 0
i_v1 = CONTROL : i_v2 = QUEUE_0 : i_v3 = QUEUE_1 : i_v4 = QUEUE_2 : i_v5 = QUEUE_3 : i_v6 = QUEUE_4
repeat 16
    cells(i_v1) = cells(i_v2)
    cells(i_v2) = cells(i_v3)
    cells(i_v3) = cells(i_v4)
    cells(i_v4) = cells(i_v5)
```

```

cells(i_v5) = cells(i_v6)
i_v1+ : i_v2+ : i_v3+ : i_v4+ : i_v5+ : i_v6+
loop
if (qg_isvacant = FALSE) {
i_v1 = QUEUE_4 : i_v2 = QUEUE_G
repeat 16
cells(i_v1) = cells(i_v2)
i_v1+ : i_v2+
loop
qg_isvacant = TRUE
} else {
gosub *gen_c1
args(0) = rets(0) : args(1) = QUEUE_4
gosub *gen_c2
}
return

```

*calc_score

```

// ARGS: %0 %1 %2 %3 RESULT %4 CNT
i_v1 = c_y * OPN_W + BOARD_H
i_v2 = args(4) - 1
repeat 4
if (args(cnt) = 1) {
repeat OPN_W
score += rate(cells(i_v1) & 0x000000FF) * comp(i_v2)
hp += heal(cells(i_v1) & 0x000000FF)
ccnt(cells(i_v1) & 0x000000FF)+
i_v1 += 1
loop
} else {
i_v1 += OPN_W
}
loop
if (hp >= PLAYER_HP_X) {
hp = PLAYER_HP_X
}
gosub *barcolor
return

```

*calc_dmg

```

i_v1 = BOARD_H
repeat COL_H
if (cells(i_v1) ! C_VA) {
hp -= cdmg(cells(i_v1) & 0x000000FF)
cells(i_v1) = C_VA
}
i_v1+
loop
repeat COL_V
hp -= cdmg(cells(i_v1) & 0x000000FF)
cells(i_v1) = C_VA
i_v1+
loop
if (hp <= 0) {
hp = 0
g_stat = GSTAT_P_D
}
gosub *barcolor
return

```

*push

```

if (s_cnt < 3) {

```

```

if (s_cnt = 0) {
    i_v1 = STACK_2
} else : if (s_cnt = 1) {
    i_v1 = STACK_1
} else {
    i_v1 = STACK_0
}
i_v2 = CONTROL
i_v3 = FALSE
repeat 16
    cells(i_v1) = cells(i_v2)
    if ((cells(i_v1) & 0x000000FF) = C_T2) {
        cells(i_v1) = (cells(i_v1) & 0xFFFFF00) | C_TG
    } else : if (cells(i_v1) = C_BM) {
        i_v3 = TRUE
        hp -= D_BM
    }
    i_v1+
    i_v2+
loop
if (i_v3 = TRUE) {
    i_v1-
    repeat 16
        if (cells(i_v1) ! C_VA) : cells(i_v1) = C_BF
        i_v1-
    loop
}
s_cnt+
if (hp > 0) {
    g_stat = GSTAT_G_S
} else {
    hp = 0
    g_stat = GSTAT_P_D
}
gosub *barcolor
} else {
    i_v1 = CONTROL : i_v2 = STACK_0
    repeat 16
        c_sw = cells(i_v1)
        cells(i_v1) = cells(i_v2)
        cells(i_v2) = c_sw
        i_v1+ : i_v2+
    loop
    gosub *req_mr
    if (rets(0) > 0) {
        i_v1 = STACK_0 : i_v2 = CONTROL
        repeat 16
            c_sw = cells(i_v1)
            cells(i_v1) = cells(i_v2)
            cells(i_v2) = c_sw
            i_v1+ : i_v2+
        loop
    } else {
        i_v1 = STACK_0
        i_v3 = FALSE
        repeat 16
            if ((cells(i_v1) & 0x000000FF) = C_T2) {
                cells(i_v1) = (cells(i_v1) & 0xFFFFF00) | C_TG
            } else : if (cells(i_v1) = C_BM) {
                i_v3 = TRUE
                hp -= D_BM
            }
        }
    }
}

```

```

        i_v1+
loop
if (i_v3 = TRUE) {
    i_v1-
    repeat 16
        if (cells(i_v1) ! C_VA) : cells(i_v1) = C_BF
        i_v1-
    loop
}
if (hp <= 0) {
    hp = 0
    g_stat = GSTAT_P_D
}
gosub *barcolor
}
}
return

```

*pop

```

// RETS: %0 0/1 F/S
if (s_cnt > 0) {
    if (qg_isvacant = TRUE) {
        i_v1 = CONTROL : i_v2 = QUEUE_G
        if (s_cnt = 1) {
            i_v3 = STACK_2
        } else : if (s_cnt = 2) {
            i_v3 = STACK_1
        } else {
            i_v3 = STACK_0
        }
        repeat 16
            cells(i_v2) = cells(i_v1)
            cells(i_v1) = cells(i_v3)
            i_v1+ : i_v2+ : i_v3+
        loop
        rets(0) = 0
        i_v4 = i_v1 : i_v5 = i_v2
        gosub *req_mr
        i_v1 = i_v4 : i_v2 = i_v5
        if (rets(0) > 0) {
            i_v1- : i_v2- : i_v3-
            repeat 16
                cells(i_v3) = cells(i_v1)
                cells(i_v1) = cells(i_v2)
                i_v1- : i_v2- : i_v3-
            loop
            rets(0) = 0
        } else {
            i_v3-
            repeat 16
                cells(i_v3) = C_VA
                i_v3-
            loop
            qg_isvacant = FALSE
            s_cnt-
            rets(0) = 1
        }
    } else {
        i_v1 = CONTROL
        if(s_cnt = 1) {
            i_v2 = STACK_2
        } else : if (s_cnt = 2) {

```

```

        i_v2 = STACK_1
    } else {
        i_v2 = STACK_0
    }
    repeat 16
        c_sw = cells(i_v1)
        cells(i_v1) = cells(i_v2)
        cells(i_v2) = c_sw
        i_v1+ : i_v2+
    loop
    i_v4 = i_v1 : i_v5 = i_v2
    gosub *req_mr
    i_v1 = i_v4 : i_v2 = i_v5
    if (rets(0) > 0) {
        i_v1- : i_v2-
        repeat 16
            c_sw = cells(i_v1)
            cells(i_v1) = cells(i_v2)
            cells(i_v2) = c_sw
            i_v1- : i_v2-
        loop
        rets(0) = 0
    } else {
        i_v2-
        i_v3 = FALSE
        repeat 16
            if ((cells(i_v2) & 0x000000FF) = C_T2) {
                cells(i_v2) = (cells(i_v2) & 0FFFFFF00) | C_TG
            } else : if (cells(i_v2) = C_BM) {
                i_v3 = TRUE
                hp -= D_BM
            }
            i_v2-
        loop
        if (i_v3 = TRUE) {
            i_v2+
            repeat 16
                if (cells(i_v2) ! C_VA) : cells(i_v2) = C_BF
                i_v2+
            loop
        }
        rets(0) = 1
        if (hp <= 0) {
            hp = 0
            g_stat = GSTAT_P_D
        }
        gosub *barcolor
    }
}
} else {
    rets(1) = 0
}
return

```

*del_drop

```

// ARGS: %0 %1 %2 %3 RESULT(1 DEL)
rets(0) = -1 : rets(1) = -1 : rets(2) = -1 : rets(3) = -1
i_v2 = 0 : i_v3 = c_y
repeat 4
    if (i_v3 >= 32) : i_v2+
    i_v3+
loop

```



```

i_v3 = 3 - i_v2 : i_v4 = 3 - i_v2 : i_v5 = 0
repeat 4 - i_v2
  if (args(i_v3) = 0) {
    rets(i_v4) = i_v3
    if (i_v3 ! i_v4) {
      args(i_v3) = 1
    }
    i_v4-
  } else {
    i_v5+
  }
  i_v3-
loop
i_v1 = BOARD_H + (c_y + 4) * OPN_W - 1
i_v3 = 3 - i_v2
repeat 4 - i_v5
  if (c_y + 3 - cnt < 32) {
    i_v6 = BOARD_H + (c_y + rets(i_v3) + 1) * OPN_W - 1
    repeat OPN_W
      cells(i_v1) = cells(i_v6)
      i_v1- : i_v6-
    loop
      i_v3-
  } else {
    i_v1 -= OPN_W
  }
loop
i_v3 = BOARD_H + c_y * OPN_W - 1
i_v4 = BOARD_H + (c_y + i_v5) * OPN_W - 1
if (i_v3 ! i_v4) {
  repeat c_y * OPN_W
    cells(i_v4) = cells(i_v3)
    cells(i_v3) = C_VA
    i_v4- : i_v3-
  loop
}
rets(0) = 0 : rets(1) = 0 : rets(2) = 0 : rets(3) = 0
i_v6 = BOARD_H
repeat 4 * OPN_W
  if (cells(i_v6) ! C_VA) : hzd = TRUE
  i_v6+
loop
return

```

*clr_ctrl

```

i_v5 = c_y : i_v6 = CONTROL
repeat 4
  if (i_v5 < 32) {
    i_v4 = c_x
    repeat 4
      if ((i_v4 >= 0) & (i_v4 < 14)) {
        if (cells(i_v6) ! C_VA) {
          cells(BOARD_H + i_v5 * OPN_W + i_v4) = C_VA
        }
      }
      i_v4+ : i_v6+
    loop
  }
  i_v5+
loop
return

```

*ipt

```
// ARGS: %0 REPEAT_H %1 REPEAT_W %2 CNT %3 CELL_LX %4 CELL_LY %5 CELL_RB %6 CELL_SIZE
i_v1 = args(4)
repeat args(0)
    i_v2 = args(3)
    repeat args(1)
        vertices(args(2)) = i_v2 : args(2)+ : vertices(args(2)) = i_v1 : args(2)+
        vertices(args(2)) = i_v2 + args(5) : args(2)+ : vertices(args(2)) = i_v1 + args(5) : args(2)+
        i_v2 += args(6)
    loop
    i_v1 += args(6)
loop
return
```

*rnt

```
noteget nl_1, 0
split nl_1, " ", txtv
i_v1 = int(txtv(0))
font msgothic, i_v1
color int(txtv(1)), int(txtv(2)), int(txtv(3))
i_v1 += 4
i_v2 = int(txtv(4)) + 2
i_v3 = int(txtv(5)) + 2
repeat noteinfo(0), 1
    pos i_v2, i_v3
    noteget nl_2, cnt
    mes nl_2
    i_v3 += i_v1
loop
noteunsel
return
```

*dr_ctrl

```
i_v5 = c_y : i_v6 = CONTROL
repeat 4
    if (i_v5 < 32) {
        i_v4 = c_x
        repeat 4
            if ((i_v4 >= 0) & (i_v4 < 14)) {
                if (cells(i_v6) ! C_VA) {
                    cells(BOARD_H + i_v5 * OPN_W + i_v4) = cells(i_v6)
                }
            }
            i_v4+ : i_v6+
        loop
    }
    i_v5+
loop
return
```

*dr_board

```
i_v2 = 0
gmode 7
repeat 520, BOARD_V
    color (cells(cnt) & 0xFF000000) >> 24, (cells(cnt) & 0x00FF0000) >> 16, (cells(cnt) & 0x0000FF00) >> 8
    boxf vertices(i_v2), vertices(i_v2 + 1), vertices(i_v2 + 2), vertices(i_v2 + 3)
    pos vertices(i_v2), vertices(i_v2 + 1)
    gcopy 2, 0, (cells(cnt) & 0x000000FF) * 24, 24, 24
    i_v2 += 4
loop
gmode 0
return
```

*dr_del

```
// ARGS2: %0 %1 %2 %3 DEL %4 CNT
gmode 7
repeat 4
  if (arg2(cnt) = 1) {
    i_v2 = (c_y - 4 + cnt) * OPN_W * 4
    repeat OPN_W
      pos vertices(i_v2), vertices(i_v2 + 1)
      gcopy 1, 840, 0, 24, 24
      i_v2 += 4
    loop
  }
loop
gmode 0
return
```

*dr_seg

```
// ARGS: %0 NUM %1 DIGIT %2 POSX %3 POSY
i_v1 = 10
i_v4 = 9
i_v2 = 20 * (args(1) - 1) + args(2)
pos i_v2, args(3)
gcopy 1, 800, args(0) ¥ 10 * 40, 20, 40
i_v2 -= 20
if (args(1) > 2) {
  repeat args(1) - 2
    pos i_v2, args(3)
    if (args(0) > i_v4) {
      i_v3 = args(0) ¥ (i_v1 * 10) / i_v1
      gcopy 1, 800, i_v3 * 40, 20, 40
      i_v1 = i_v1 * 10
    } else {
      gcopy 1, 800, 400, 20, 40
    }
    i_v2 -= 20
    i_v4 = i_v4 * 10 + 9
  loop
}
if (args(1) > 1) {
  pos i_v2, args(3)
  if (args(0) > i_v4) {
    i_v3 = args(0) / i_v1
    gcopy 1, 800, i_v3 * 40, 20, 40
  } else {
    gcopy 1, 800, 400, 20, 40
  }
}
return
```

*dr_seg2

```
// ARGS: %1 DIGIT %2 POSX %3 POSY
i_v2 = args(2)
repeat args(1)
  pos i_v2, args(3)
  gcopy 1, 800, 400, 20, 40
  i_v2 += 20
loop
return
```

*dr_menu

```
pos 156, 205
```

gcopy 3, 0, 147, 336, 672
gmode 7

color 255, 255, 255

```
if (g_stat = GSTAT_P_N) {
    pos 164, cur_v * 24 + 305 : gcopy 2, 512, 64, 16, 16
    pos 236, 213 : gcopy 2, 0, 256, 176, 80 // TITLE
    pos 186, 303 : gcopy 6, 488, 0, 84, 20 // -PLAY
    pos 186, 327 : gcopy 6, 488, 20, 84, 20 // -CONFIG
    pos 186, 351 : gcopy 6, 488, 40, 84, 20 // -INFO
    pos 186, 375 : gcopy 6, 488, 60, 84, 20 // -QUIT
} else : if (g_stat = GSTAT_P_P) {
    pos 164, cur_v * 24 + 305 : gcopy 2, 512, 64, 16, 16
    pos 162, 243 : gcopy 6, 0, 0, 244, 52 // PAUSE
    pos 186, 303 : gcopy 6, 488, 80, 84, 20 // -RESUME
    pos 186, 327 : gcopy 6, 488, 20, 84, 20 // -CONFIG
    pos 186, 351 : gcopy 6, 488, 120, 84, 20 // -TITLE
    pos 186, 375 : gcopy 6, 488, 40, 84, 20 // -INFO
    pos 186, 399 : gcopy 6, 488, 60, 84, 20 // -QUIT
    pos 162, 853 : gcopy 7, 0, 300, 324, 20 // [ACT]
} else : if (g_stat = GSTAT_P_S) {
    pos 164, cur_v * 24 + 305 : gcopy 2, 512, 64, 16, 16
    pos 162, 243 : gcopy 6, 0, 52, 244, 52 // CLEAR
    pos 186, 303 : gcopy 6, 488, 100, 84, 20 // -STAFF
    pos 186, 327 : gcopy 6, 488, 120, 84, 20 // -TITLE
    pos 186, 351 : gcopy 6, 488, 60, 84, 20 // -QUIT
    pos 162, 853 : gcopy 7, 0, 300, 324, 20 // [ACT]
} else : if (g_stat = GSTAT_P_D) {
    pos 164, cur_v * 24 + 305 : gcopy 2, 512, 64, 16, 16
    pos 162, 243 : gcopy 6, 0, 104, 244, 52 // GAME OVER
    pos 186, 303 : gcopy 6, 488, 120, 84, 20 // -TITLE
    pos 186, 327 : gcopy 6, 488, 60, 84, 20 // -QUIT
    pos 162, 853 : gcopy 7, 0, 300, 324, 20 // [ACT]
} else : if ((g_stat = GSTAT_T_L1) | ((g_stat & GSTAT_T_L1_N) = GSTAT_T_L1_N)) {
    i_v1 = cur_v * 24 + 305
    pos 164, i_v1 : gcopy 2, 512, 64, 16, 16
    if ((g_stat & GSTAT_T_L1_N) = GSTAT_T_L1_N) {
        pos 260, i_v1 : gcopy 2, 512, 160, 16, 16
        pos 452, i_v1 : gcopy 2, 512, 144, 16, 16
    }
    if (cur_v = 0) {
        // COPPER
        pos 162, 543 : gcopy 7, 0, 0, 324, 20 // exp1
        pos 162, 563 : gcopy 7, 0, 20, 324, 20 // exp2
    } else : if (cur_v = 1) {
        // SILVER
        pos 162, 543 : gcopy 7, 0, 40, 324, 20 // exp1
        pos 162, 563 : gcopy 7, 0, 60, 324, 20 // exp2
    } else : if (cur_v = 2) {
        // GOLD
        pos 162, 543 : gcopy 7, 0, 80, 324, 20 // exp1
        pos 162, 563 : gcopy 7, 0, 100, 324, 20 // exp2
    } else : if (cur_v = 3) {
        // PLATINUM
        pos 162, 543 : gcopy 7, 0, 120, 324, 20 // exp1
        pos 162, 563 : gcopy 7, 0, 140, 324, 20 // exp2
    } else : if (cur_v = 4) {
        // IRON
        pos 162, 543 : gcopy 7, 0, 160, 324, 20 // exp1
    } else : if (cur_v = 5) {
        // IRIDIUM
        pos 162, 543 : gcopy 7, 0, 220, 324, 20 // exp1
    }
}
```

```

    pos 162, 563 : gcopy 7, 0, 240, 324, 20 // exp2
}
pos 162, 243 : gcopy 6, 0, 260, 244, 52 // LEVEL
pos 186, 303 : gcopy 6, 488, 420, 84, 20 // -COPPER
if (g_stat = GSTAT_T_L2_C) {
    pos 290, 303 : gcopy 6, 656, 200, 36, 20 // 0
    pos 298, 303 : gcopy 6, 656, 220, 36, 20 // 1
    pos 318, 303 : gcopy 6, 656, 160, 36, 20 // -
    pos 338, 303 : gcopy 6, 656, 200, 36, 20 // 0
    pos 346, 303 : gcopy 6, 656, 320, 36, 20 // 6
    pos 372, 305 : gcopy 2, 512, 16, 16, 16 // *
    pos 388, 305 : gcopy 2, 512, 0, 16, 16 // -
    pos 404, 305 : gcopy 2, 512, 0, 16, 16 // -
    pos 420, 305 : gcopy 2, 512, 0, 16, 16 // -
}
pos 186, 327 : gcopy 6, 488, 440, 84, 20 // -SILVER
if (g_stat = GSTAT_T_L2_S) {
    pos 290, 327 : gcopy 6, 656, 200, 36, 20 // 0
    pos 298, 327 : gcopy 6, 656, 280, 36, 20 // 4
    pos 318, 327 : gcopy 6, 656, 160, 36, 20 // -
    pos 338, 327 : gcopy 6, 656, 220, 36, 20 // 1
    pos 346, 327 : gcopy 6, 656, 200, 36, 20 // 0
    pos 372, 329 : gcopy 2, 512, 16, 16, 16 // *
    pos 388, 329 : gcopy 2, 512, 16, 16, 16 // *
    pos 404, 329 : gcopy 2, 512, 0, 16, 16 // -
    pos 420, 329 : gcopy 2, 512, 0, 16, 16 // -
}
pos 186, 351 : gcopy 6, 488, 460, 84, 20 // -GOLD
if (g_stat = GSTAT_T_L2_G) {
    pos 290, 351 : gcopy 6, 656, 200, 36, 20 // 0
    pos 298, 351 : gcopy 6, 656, 340, 36, 20 // 7
    pos 318, 351 : gcopy 6, 656, 160, 36, 20 // -
    pos 338, 351 : gcopy 6, 656, 220, 36, 20 // 1
    pos 346, 351 : gcopy 6, 656, 280, 36, 20 // 4
    pos 372, 353 : gcopy 2, 512, 16, 16, 16 // *
    pos 388, 353 : gcopy 2, 512, 16, 16, 16 // *
    pos 404, 353 : gcopy 2, 512, 16, 16, 16 // *
    pos 420, 353 : gcopy 2, 512, 0, 16, 16 // -
}
pos 186, 375 : gcopy 6, 488, 480, 84, 20 // -PLATINUM
if (g_stat = GSTAT_T_L2_P) {
    pos 290, 375 : gcopy 6, 656, 220, 36, 20 // 1
    pos 298, 375 : gcopy 6, 656, 200, 36, 20 // 0
    pos 318, 375 : gcopy 6, 656, 160, 36, 20 // -
    pos 338, 375 : gcopy 6, 656, 220, 36, 20 // 1
    pos 346, 375 : gcopy 6, 656, 360, 36, 20 // 8
    pos 372, 377 : gcopy 2, 512, 16, 16, 16 // *
    pos 388, 377 : gcopy 2, 512, 16, 16, 16 // *
    pos 404, 377 : gcopy 2, 512, 16, 16, 16 // *
    pos 420, 377 : gcopy 2, 512, 16, 16, 16 // *
}
pos 186, 399 : gcopy 6, 488, 500, 84, 20 // -IRON
if (g_stat = GSTAT_T_L2_N) {
    pos 276, 401 : gcopy 2, 512, 32, 16, 16 // N1
    if (kgot(KEY_CVL) = 1) {
        pos 292, 401 : gcopy 2, 512, 96, 16, 16 // <<
    } else {
        pos 292, 401 : gcopy 2, 512, 80, 16, 16 // <
    }
    if (kgot(KEY_CVR) = 1) {
        pos 340, 401 : gcopy 2, 512, 128, 16, 16 // >>
    } else {

```

```

        pos 340, 401 : gcopy 2, 512, 112, 16, 16 // >
    }
    pos 356, 401 : gcopy 2, 512, 48, 16, 16 // X20
    pos 370, 399 : gcopy 6, 488, 540, 84, 20 // Training
    if (cur_h = 18) {
        pos 434, 399 : gcopy 6, 656, 180, 36, 20 // ??
        pos 162, 563 : gcopy 7, 0, 180, 324, 20 // exp2
    } else : if (cur_h = 19) {
        pos 434, 399 : gcopy 6, 656, 180, 36, 20 // ??
        pos 162, 563 : gcopy 7, 0, 200, 324, 20 // exp2
    }
    pos 314, 399 : gcopy 6, 656, ((cur_h + 1) / 10) * 20 + 200, 36, 20
    pos 322, 399 : gcopy 6, 656, ((cur_h + 1) % 10) * 20 + 200, 36, 20
}
if (ir_open = 1) {
    pos 186, 423 : gcopy 6, 488, 520, 84, 20 // -IRIDIUM
}
if (g_stat = GSTAT_T_L2_M) {
    pos 314, 423 : gcopy 6, 656, 240, 36, 20 // 2
    pos 322, 423 : gcopy 6, 656, 220, 36, 20 // 1
}
if (g_stat = GSTAT_T_L1) {
    pos 162, 853 : gcopy 7, 0, 260, 324, 20 // expb
} else {
    pos 162, 853 : gcopy 7, 0, 280, 324, 20 // expb
}
} else : if (g_stat = GSTAT_T_C1) {
    pos 164, cur_v * 24 + 305 : gcopy 2, 512, 64, 16, 16
    pos 162, 243 : gcopy 6, 0, 312, 244, 52 // CONFIG
    pos 186, 303 : gcopy 6, 488, 140, 84, 20 // -SEGMENT
    pos 186, 327 : gcopy 6, 488, 160, 84, 20 // -GRAPHICS
    pos 186, 351 : gcopy 6, 488, 180, 84, 20 // -SOUND
    pos 186, 375 : gcopy 6, 488, 200, 84, 20 // -CONTROL
    pos 162, 853 : gcopy 7, 1296, 0, 324, 20 // expb
} else : if (g_stat = GSTAT_T_I1) {
    pos 164, cur_v * 24 + 305 : gcopy 2, 512, 64, 16, 16
    pos 162, 243 : gcopy 6, 0, 364, 244, 52 // INFO
    pos 186, 303 : gcopy 6, 488, 220, 84, 20 // - ABOUT
    pos 186, 327 : gcopy 6, 488, 240, 84, 20 // - SYSTEM
    pos 186, 351 : gcopy 6, 488, 260, 84, 20 // - LICENSE
    pos 186, 375 : gcopy 6, 488, 100, 84, 20 // - STAFF
    pos 162, 853 : gcopy 7, 1296, 0, 324, 20 // expb
} else : if (g_stat = GSTAT_T_C2_SEG) {
    i_v1 = cur_v * 24 + 305
    pos 164, i_v1 : gcopy 2, 512, 64, 16, 16
    repeat 7
        i_v2 = cnt * 24 + 305
        if (sgsw(cnt) = 1) {
            i_v3 = 292
        } else {
            i_v3 = 356
        }
        if ((cfg_sel = 0) | (cnt != cur_v)) {
            pos i_v3, i_v2 : gcopy 2, 512, 224, 16, 16
            pos i_v3 + 16, i_v2 : gcopy 2, 512, 240, 16, 16
            pos i_v3 + 32, i_v2 : gcopy 2, 512, 240, 16, 16
            pos i_v3 + 48, i_v2 : gcopy 2, 512, 256, 16, 16
        } else {
            pos i_v3, i_v2 : gcopy 2, 512, 176, 16, 16
            pos i_v3 + 16, i_v2 : gcopy 2, 512, 192, 16, 16
            pos i_v3 + 32, i_v2 : gcopy 2, 512, 192, 16, 16
            pos i_v3 + 48, i_v2 : gcopy 2, 512, 208, 16, 16
        }
    }
}

```

```

        if (kgot(KEY_CVL) = 1) {
            pos 276, i_v2 : gcopy 2, 512, 96, 16, 16 // <<
        } else {
            pos 276, i_v2 : gcopy 2, 512, 80, 16, 16 // <
        }
        if (kgot(KEY_CVR) = 1) {
            pos 420, i_v2 : gcopy 2, 512, 128, 16, 16 // >>
        } else {
            pos 420, i_v2 : gcopy 2, 512, 112, 16, 16 // >
        }
    }
    pos 314, i_v2 - 2 : gcopy 6, 656, 40, 36, 20 // ON
    pos 374, i_v2 - 2 : gcopy 6, 656, 60, 36, 20 // OFF
loop
repeat 2, 7
    i_v2 = cnt * 24 + 305
    if (sgsw(cnt) = 2) {
        i_v3 = 292
    } else : if (sgsw(cnt) = 1) {
        i_v3 = 388
    } else {
        i_v3 = 340
    }
    if ((cfg_sel = 0) | (cnt ! cur_v)) {
        pos i_v3, i_v2 : gcopy 2, 512, 224, 16, 16
        pos i_v3 + 16, i_v2 : gcopy 2, 512, 240, 16, 16
        pos i_v3 + 32, i_v2 : gcopy 2, 512, 256, 16, 16
    } else {
        pos i_v3, i_v2 : gcopy 2, 512, 176, 16, 16
        pos i_v3 + 16, i_v2 : gcopy 2, 512, 192, 16, 16
        pos i_v3 + 32, i_v2 : gcopy 2, 512, 208, 16, 16
        if (kgot(KEY_CVL) = 1) {
            pos 276, i_v2 : gcopy 2, 512, 96, 16, 16 // <<
        } else {
            pos 276, i_v2 : gcopy 2, 512, 80, 16, 16 // <
        }
        if (kgot(KEY_CVR) = 1) {
            pos 436, i_v2 : gcopy 2, 512, 128, 16, 16 // >>
        } else {
            pos 436, i_v2 : gcopy 2, 512, 112, 16, 16 // >
        }
    }
}
pos 298, i_v2 - 2 : gcopy 6, 656, 0, 36, 20 // HIGH
pos 350, i_v2 - 2 : gcopy 6, 656, 60, 36, 20 // OFF
pos 398, i_v2 - 2 : gcopy 6, 656, 20, 36, 20 // LOW
loop
pos 162, 243 : gcopy 6, 0, 468, 244, 52 // SEGMENT
pos 186, 303 : gcopy 6, 488, 280, 84, 20 // -CLOCK
pos 186, 327 : gcopy 6, 488, 300, 84, 20 // -LEVEL
pos 186, 351 : gcopy 6, 488, 320, 84, 20 // -WAIT
pos 186, 375 : gcopy 6, 488, 340, 84, 20 // -HP
pos 186, 399 : gcopy 6, 488, 360, 84, 20 // -SCORE
pos 186, 423 : gcopy 6, 488, 560, 84, 20 // -LINES
pos 186, 447 : gcopy 6, 488, 600, 84, 20 // -COUNT
pos 186, 471 : gcopy 6, 488, 380, 84, 20 // -HP BAR
pos 186, 495 : gcopy 6, 488, 400, 84, 20 // -COUNT ICON
if (cfg_sel = 0) {
    pos 162, 853 : gcopy 7, 1296, 40, 324, 20 // expb
} else {
    pos 162, 829 : gcopy 7, 1296, 60, 324, 20
    pos 162, 853 : gcopy 7, 1296, 80, 324, 20
}

```

```

if (cur_v < 7) {
    pos 162, 543 : gcopy 7, 1296, 100, 324, 20
} else {
    pos 162, 543 : gcopy 7, 1296, 120, 324, 20
    pos 162, 567 : gcopy 7, 1296, 140, 324, 20
    pos 162, 591 : gcopy 7, 1296, 160, 324, 20
}
} else : if (g_stat = GSTAT_T_C2_GRA) {
    i_v1 = cur_v * 24 + 305
    pos 164, i_v1 : gcopy 2, 512, 64, 16, 16
    repeat 1
        i_v2 = cnt * 24 + 305
        if (scsw(cnt) = 1) {
            i_v3 = 292
        } else {
            i_v3 = 356
        }
        if ((cfg_sel = 0) | (cnt ! cur_v)) {
            pos i_v3, i_v2 : gcopy 2, 512, 224, 16, 16
            pos i_v3 + 16, i_v2 : gcopy 2, 512, 240, 16, 16
            pos i_v3 + 32, i_v2 : gcopy 2, 512, 240, 16, 16
            pos i_v3 + 48, i_v2 : gcopy 2, 512, 256, 16, 16
        } else {
            pos i_v3, i_v2 : gcopy 2, 512, 176, 16, 16
            pos i_v3 + 16, i_v2 : gcopy 2, 512, 192, 16, 16
            pos i_v3 + 32, i_v2 : gcopy 2, 512, 192, 16, 16
            pos i_v3 + 48, i_v2 : gcopy 2, 512, 208, 16, 16
            if (kgot(KEY_CVL) = 1) {
                pos 276, i_v2 : gcopy 2, 512, 96, 16, 16 // <<
            } else {
                pos 276, i_v2 : gcopy 2, 512, 80, 16, 16 // <
            }
            if (kgot(KEY_CVR) = 1) {
                pos 436, i_v2 : gcopy 2, 512, 128, 16, 16 // >>
            } else {
                pos 436, i_v2 : gcopy 2, 512, 112, 16, 16 // >
            }
        }
    }
    pos 314, i_v2 - 2 : gcopy 6, 656, 100, 36, 16 // x2
    pos 378, i_v2 - 2 : gcopy 6, 656, 80, 36, 16 // x1
loop
pos 162, 243 : gcopy 6, 0, 156, 244, 52 // GRAPHICS
pos 186, 303 : gcopy 6, 488, 580, 84, 20 // -RESOLUTION
pos 162, 543 : gcopy 7, 1296, 260, 324, 20
pos 162, 567 : gcopy 7, 1296, 280, 324, 20
if (cfg_sel = 0) {
    pos 162, 853 : gcopy 7, 1296, 40, 324, 20
} else {
    pos 162, 829 : gcopy 7, 1296, 60, 324, 20
    pos 162, 853 : gcopy 7, 1296, 80, 324, 20
}
} else : if (g_stat = GSTAT_T_C2_SOU) {
    i_v1 = cur_v * 24 + 305
    pos 164, i_v1 : gcopy 2, 512, 64, 16, 16
    repeat 1
        i_v2 = cnt * 24 + 305
        if ((cfg_sel = 1) & (cnt = cur_v)) {
            if (kgot(KEY_CVL) = 1) {
                pos 292, i_v2 : gcopy 2, 512, 96, 16, 16 // <<
            } else {
                pos 292, i_v2 : gcopy 2, 512, 80, 16, 16 // <
            }
        }
    }
}

```



```

        if (kgot(KEY_CVR) = 1) {
            pos 340, i_v2 : gcopy 2, 512, 128, 16, 16 // >>
        } else {
            pos 340, i_v2 : gcopy 2, 512, 112, 16, 16 // >
        }
        pos 260, i_v1 : gcopy 2, 512, 160, 16, 16
        pos 452, i_v1 : gcopy 2, 512, 144, 16, 16
    }
    i_v2 -= 2
    if (sosw(cnt) < 10) {
        pos 318, i_v2 : gcopy 6, 656, sosw(cnt) * 20 + 200, 36, 20
    } else : if (sosw(cnt) < 100) {
        pos 314, i_v2 : gcopy 6, 656, (sosw(cnt) / 10) * 20 + 200, 36, 20
        pos 322, i_v2 : gcopy 6, 656, (sosw(cnt) ¥ 10) * 20 + 200, 36, 20
    } else {
        pos 310, i_v2 : gcopy 6, 656, 220, 36, 20
        pos 318, i_v2 : gcopy 6, 656, 200, 36, 20
        pos 326, i_v2 : gcopy 6, 656, 200, 36, 20
    }
loop
pos 162, 243 : gcopy 6, 0, 208, 244, 52 // SOUND
pos 186, 303 : gcopy 6, 488, 640, 84, 20 // -SE
pos 162, 543 : gcopy 7, 1296, 320, 324, 20
pos 162, 567 : gcopy 7, 1296, 400, 324, 20
if (cfg_sel = 0) {
    pos 162, 853 : gcopy 7, 1296, 40, 324, 20
} else {
    pos 162, 829 : gcopy 7, 1296, 60, 324, 20
    pos 162, 853 : gcopy 7, 1296, 80, 324, 20
}
} else : if (g_stat = GSTAT_T_C2_CTR) {
    i_v1 = cur_v * 24 + 305
    pos 164, i_v1 : gcopy 2, 512, 64, 16, 16
    repeat 1
        i_v2 = cnt * 24 + 305
        if (cur_h = 1) {
            i_v3 = 356
        } else {
            i_v3 = 292
        }
        if ((cfg_sel = 0) | (cnt ! cur_v)) {
            pos i_v3, i_v2 : gcopy 2, 512, 224, 16, 16
            pos i_v3 + 16, i_v2 : gcopy 2, 512, 240, 16, 16
            pos i_v3 + 32, i_v2 : gcopy 2, 512, 240, 16, 16
            pos i_v3 + 48, i_v2 : gcopy 2, 512, 256, 16, 16
        } else {
            pos i_v3, i_v2 : gcopy 2, 512, 176, 16, 16
            pos i_v3 + 16, i_v2 : gcopy 2, 512, 192, 16, 16
            pos i_v3 + 32, i_v2 : gcopy 2, 512, 192, 16, 16
            pos i_v3 + 48, i_v2 : gcopy 2, 512, 208, 16, 16
            if (kgot(KEY_CVL) = 1) {
                pos 276, i_v2 : gcopy 2, 512, 96, 16, 16 // <<
            } else {
                pos 276, i_v2 : gcopy 2, 512, 80, 16, 16 // <
            }
            if (kgot(KEY_CVR) = 1) {
                pos 436, i_v2 : gcopy 2, 512, 128, 16, 16 // >>
            } else {
                pos 436, i_v2 : gcopy 2, 512, 112, 16, 16 // >
            }
        }
    }
pos 318, i_v2 - 2 : gcopy 6, 656, 120, 36, 20 // A

```

```

        pos 382, i_v2 - 2 : gcopy 6, 656, 140, 36, 20 // B
    loop
pos 162, 243 : gcopy 6, 0, 416, 244, 52 // CONTROL
pos 186, 303 : gcopy 6, 656, 400, 36, 20 // -TYPE
pos 162, 543 : gcopy 7, 1296, 180, 324, 20
pos 162, 567 : gcopy 7, 1296, 200, 324, 20
pos 162, 591 : gcopy 7, 1296, 220, 324, 20
pos 162, 615 : gcopy 7, 1296, 240, 324, 20
pos 162, 639 : gcopy 7, 1296, 300, 324, 20
    if (cfg_sel = 0) {
        pos 162, 853 : gcopy 7, 1296, 40, 324, 20
    } else {
        pos 162, 829 : gcopy 7, 1296, 60, 324, 20
        pos 162, 853 : gcopy 7, 1296, 80, 324, 20
    }
} else : if (g_stat = GSTAT_T_I2_ABO) {
    pos 162, 243 : gcopy 6, 0, 520, 244, 52 // ABOUT
    if (cur_h = 0) {
        pos 162, 303 : gcopy 7, 648, 0, 324, 460 // p1
        pos 162, 853 : gcopy 7, 648, 460, 324, 20
    } else : if (cur_h = 1) {
        pos 162, 303 : gcopy 7, 648, 480, 324, 460 // p2
        pos 162, 853 : gcopy 7, 648, 940, 324, 20
        pos 164, 325 : gcopy 2, 0, C_OC * 24, 24, 24
        color 255, 175, 0
        boxf 164, 405, 187, 428
        pos 164, 405 : gcopy 2, 0, C_T1 * 24, 24, 24
        color 0, 175, 255
        boxf 164, 485, 187, 508
        pos 164, 485 : gcopy 2, 0, C_T2 * 24, 24, 24
        color 175, 255, 0
        boxf 164, 565, 187, 588
        pos 164, 565 : gcopy 2, 0, C_TG * 24, 24, 24
        pos 164, 645 : gcopy 2, 0, C_BM * 24, 24, 24
    } else : if (cur_h = 2) {
        pos 162, 303 : gcopy 7, 648, 960, 324, 460 // p3
        pos 162, 853 : gcopy 7, 648, 1420, 324, 20
        pos 164, 325 : gcopy 2, 0, C_BG * 24, 24, 24
        pos 164, 405 : gcopy 2, 0, C_BF * 24, 24, 24
    }
} else : if (g_stat = GSTAT_T_I2_SYS) {
    pos 162, 243 : gcopy 6, 0, 572, 244, 52 // SYSTEM
    pos 162, 853 : gcopy 7, 1296, 0, 324, 20
} else : if (g_stat = GSTAT_T_I2_LIC) {
    pos 162, 243 : gcopy 6, 0, 624, 244, 52 // LICENSE
    pos 162, 303 : gcopy 7, 1296, 340, 324, 20
    pos 162, 327 : gcopy 7, 1296, 360, 324, 20
    pos 162, 853 : gcopy 7, 1296, 0, 324, 20
} else : if (g_stat = GSTAT_T_I2_STA) {
    pos 162, 243 : gcopy 6, 0, 676, 244, 52 // STAFF
    pos 162, 303 : gcopy 7, 1296, 380, 324, 20
    pos 162, 853 : gcopy 7, 1296, 20, 324, 20
    pos 180, 329 : gcopy 2, 0, 512, 288, 336 // LOGO
}
gmode 0
return

```