

Sprites Alive para ccz80

La librería SpritesAlive.ccz80 proporciona el interfaz necesario para utilizar las rutinas incluidas en la utilidad Sprites Alive bajo el lenguaje ccz80.

Para utilizarla es necesario tener en cuenta los siguiente puntos:

- Para cargar las rutinas de Sprites Alive y activarlas se debe utilizar la nueva función `sp_load()`, sin argumentos.
- Todos los comandos RSX de Sprites Alive se pueden utilizar bajo ccz80, con los mismos nombres de las órdenes RSX pero con el prefijo `sp_`, para evitar duplicidad con las funciones de las librerías específicas de cada ordenador, por ejemplo `|CLS`, que es `sp_cls()` bajo ccz80.
- Los argumentos que en las órdenes RSX van separados por comas, bajo ccz80 van entre paréntesis y separados por comas. Por ejemplo `|SPUT,0,100,50` se usa como `sp_put(0,100,50)` bajo ccz80.
- Algunas órdenes de Sprites Alive que admiten diferente número de parámetros para realizar diversas funcionalidades se han desdoblado en varias funciones de la librería Sprites Alive para ccz80. Estas órdenes son:

Orden RSX	Número argumentos	Función para ccz80
MOVE	1	sp_move
	2	sp_moverange
REPORT	1	sp_report
	3	sp_reportrange
SPUT	1	sp_sputunique
	2	sp_sputrange
	3	sp_sput

- Las órdenes RSX que necesitan recibir una variable por referencia (siempre es como último argumento) no deben recibirla en la función ccz80 correspondiente, ya que la función devolverá el valor que se asignaría a esa variable. Por ejemplo, si en BASIC se usaría `|READSTIX, 18, @a%`, en ccz80 se haría `a = sp_readstix(18)`, donde `a` estaría declarada de tipo `word`.

Para utilizar este interfaz es necesario:

- En el programa fuente ccz80 utilizar la sentencia siguiente, teniendo el archivo `SpritesAlive.ccz80` en la misma carpeta que el programa fuente:

```
include "SpritesAlive.ccz80"
```

- Antes de utilizar las órdenes de Sprites Alive es necesario cargar las rutinas y activarlas con la función `sp_load()`.
- En el momento de ejecutar el programa sobre un Amstrad real o sobre un emulador, en la unidad A del Amstrad o emulador se debe tener el disquete

de Sprites Alive, o simplemente un disquete con el archivo SPRITE.1 que se encuentra en la cara A de Sprites Alive. Este fichero es el que se carga al utilizar la función sp_load().

Es **muy importante** tener en cuenta que el programa ejecutable no se solape con las rutinas de Sprites Alive que se cargan a partir de la dirección #7000. Se cuenta con unos 12KB para el programa binario ccz80, pero esta cantidad se ve disminuida cuando se cargan los gráficos con la función sp_draw o se crean con sp_dget o bien se cargan nodos con la función sp_node, ya que con ellas la dirección final para el código binario de ccz80 baja de #7000 tanto como ocupen los gráficos y/o nodos. Para saber hasta que dirección puede ocupar el código binario del programa ccz80 se debe utilizar la función sp_mem, que devuelve a partir de qué dirección se encuentra la memoria ocupada por los gráficos o nodos.

A continuación se listan los ejemplos que vienen en el disquete de Sprites Alive, en el formato Basic original y convertidos para ccz80. Sin duda que el código fuente ccz80 traducido de los ejemplos originales podría haber sido escrito con más claridad y más optimizado, pero se ha intentado conservar lo más literal posible para poder comparar ambos códigos. Estos ejemplos se pueden compilar con el siguiente comando desde la línea de órdenes:

```
ccz80.exe <archivo fuente ccz80> /org=#170
```

Ejemplo 1:

```
// 10 ' TANK DEMO (DEM01)
// 20 '
// 30 ' GLENCO SOFTWARE 25 SEPTEMBER 1989
// 40 '
// 50 MEMORY &2F00:|ERASE:|DRAW,"DEMOA":MODE 0:LOCATE 5,1:PRINT"TANK DEMO"
// 60 A%=0:|MISSTYPE,0,40:|MISSDELAY,0,14:|MISSDIST,25:|BULLET,0,3,12
// 70 |MISSILE,32,0,4,7,0,4:|MISSILE,33,0,12,5,4,4:|MISSILE,34,0,14,-4,4,0
// 80 |MISSILE,35,0,14,-14,4,-4:|MISSILE,36,0,4,-19,0,-4
// 90 |MISSILE,37,0,-5,-15,-4,-4:|MISSILE,38,0,-6,-4,-4,0:|MISSILE,39,0,-4,6,-4,4
//100 |SGET,0,32:|STIX,63:|XEDGE,0,4:|YEDGE,0,2:|SGET,2,41:|COLLIDE,2,1
//110 |SPUT,2,140,20:|REPO,2:|SEQUENCE,0,32,32,39,39,38,38,37,37,33,33,34,34,35,35,36,36
//120 |ANIMATE,0,0:|STIXSPEED,1,1,1,1:|SPUT,0,80,100:|SOUND,0,135,1,0,100,31,15,5
//130 ENV 1,15,-2,1:|COLOUR:|SOUND,1,135,2,0,10,15,0,120
//140 |SATTR,0,1,0:|SATTR,2,2,1:ENV 2,15,1,1,40,0,1,15,-1,7
//150 |MOVEALL:|MISSHIT,0,@A%:IF A%=255 THEN 150
//160 |EXPLODE,2,50,10,50,0
//170 X=RND(8)*140:Y=RND(4)*170+20:|SPUT,2,X,Y:|COLLTEST,2,@A%
//180 IF A%=0 THEN |SPUT,2:GOTO 170 ELSE GOTO 150

// TANK DEMO (DEM01)
//
// GLENCO SOFTWARE 25 SEPTEMBER 1989
include "cpc6128.ccz80";
include "SpritesAlive.ccz80";

word x, y;
array byte data_env_1 = { 1, 15, 254, 1 }, data_env_2 = { 3, 15, 1, 1, 40, 0, 1, 15, 255, 7
};

sp_load();
sp_draw("demoa"); mode(0); locate(5, 1); prints("tank demo");
sp_missstype(0, 40); sp_missdelay(0, 14); sp_missdist(25); sp_bullet(0, 3, 12);
sp_missile(32, 0, 4, 7, 0, 4); sp_missile(33, 0, 12, 5, 4, 4); sp_missile(34, 0, 14, -4, 4,
0);
sp_missile(35, 0, 14, -14, 4, -4); sp_missile(36, 0, 4, -19, 0, -4);
sp_missile(37, 0, -5, -15, -4, -4); sp_missile(38, 0, -6, -4, -4, 0); sp_missile(39, 0, -4,
6, -4, 4);
sp_sget(0, 32); sp_stix(63); sp_xedge(0, 4); sp_yedge(0, 2); sp_sget(2, 41); sp_collide(2,
1);
```

```

sp_sput(2, 140, 20); sp_repon(2); sp_sequence(0, 32, 32, 39, 39, 38, 38, 37, 37, 33, 33,
34, 34, 35, 35, 36, 36);
sp_animate(0, 0); sp_stixspeed(1, 1, 1, 1); sp_sput(0, 80, 100); sp_sound(0, 135, 1, 0,
100, 31, 15, 5);
env(1, data_env_1); sp_colour(); sp_sound(1, 135, 2, 0, 10, 15, 0, 120);
sp_sattr(0, 1, 0); sp_sattr(2, 2, 1); env(2, data_env_2);
line150:
sp_moveall(); if (sp_misshitttype(0) == 255) goto line150;
sp_explode(2, 50, 10, 50, 0);
line170:
x = rand() % 141; y = rand() % 170 + 20; sp_sput(2, x, y);
if (sp_colltest(2) == 0) { sp_sputunique(2); goto line170; }
else goto line150;

```

Ejemplo 2:

```

// 10 ' BALLS DEMO (DEMO2)
// 20 '
// 30 ' GLENCO SOFTWARE 13-SEPT-1989
// 40 '
// 41 ' Demonstates edge attributes and automatic sound.
// 42 '
// 50 MEMORY &2F00:|ERASE:|DRAW,"DEMOA":|COLOUR:MODE 0:|SGET,16,24:a%=0
// 60 |SGET,14,25:|SGET,15,26:|SPUT,16,10,50:|SPUT,14,72,120:|SPUT,15,130,160
// 70 N=10:FOR I=2 TO 13:|SGET,I,N:|COLLIDE,I,3:x%=0:y%=0
// 80 WHILE (x%=0 OR y%=0): x%=RND(8)*7-4:y%=RND(8)*7-4:WEND
// 90 |SDIR,i,x%,y%:|REPON,I:|SPUT,I,RND(5)*145,RND(9)*180+10:|COLLTEST,I,@A%
//100 IF A%=0 THEN |SPUT,I:GOTO 90
//110 |REPOFF,I:N=N+1:IF N=14 THEN N=10:|XEDGE,I,4 ELSE IF N=11 THEN |YEDGE,I,4
//120 IF N=12 THEN |XEDGE,I,4:|YEDGE,I,4 ELSE IF N=13 THEN |COLLIDE,I,3:|REPON,I
//130 NEXT:|SOUND,0,1,0,0,30,0,9,5:|SATTR,4,4,0
//140 |SOUND,1,2,0,0,60,0,9,5:|SATTR,8,4,1:|SOUND,2,4,0,0,90,0,9,5:|SATTR,12,4,2
//150 |MOVEALL:GOTO 150

// BALLS DEMO (DEMO2)
//
// GLENCO SOFTWARE 13-SEPT-1989
//
// Demonstates edge attributes and automatic sound.
include "cpc6128.ccz80";
include "SpritesAlive.ccz80";

word i, n, x, y;

sp_load();
sp_draw("demoa"); sp_colour(); mode(0); sp_sget(16, 24);
sp_sget(14, 25); sp_sget(15, 26); sp_sput(16, 10, 50); sp_sput(14, 72, 120); sp_sput(15,
130, 160);
for (n = 10, i = 2; i <= 13; ++i)
{
    sp_sget(i, n); sp_collide(i, 3); x = 0; y = 0; while (x == 0 || y == 0) { x = rand() % 8
- 4; y = rand() % 7 - 4; }
    line90:
    sp_sdir(i, x, y); sp_repon(i); sp_sput(i, rand() % 146, rand() % 181 + 10);
    if (sp_colltest(i) == 0) { sp_sputunique(i); goto line90; }
    sp_repoft(i); ++n; if (n == 14) { n = 10; sp_xedge(i, 4); } else if (n == 11) sp_yedge(i,
4);
    if (n == 12) { sp_xedge(i, 4); sp_yedge(i, 4); } else if (n == 13) { sp_collide(i, 3);
sp_repon(i); }
}
sp_sound(0, 1, 0, 0, 30, 0, 9, 5); sp_sattr(4, 4, 0);
sp_sound(1, 2, 0, 0, 60, 0, 9, 5); sp_sattr(8, 4, 1); sp_sound(2, 4, 0, 0, 90, 0, 9, 5);
sp_sattr(12, 4, 2);
line150:
sp_moveall(); goto line150;

```

Ejemplo 3:

```

// 10 ' BIRD DEMO (DEMO3)
// 20 '
// 30 ' GLENCO SOFTWARE 19-SEPT-1989
// 40 '
// 50 ' Demonstrates individual windows for sprites and automatic animation.
// 60 '
// 70 |ERASE:MEMORY &2F00:|DRAW,"DEMOA":MODE 0:|INKBLACK:a%=0
// 80 A=0:B=4:C=5:D=9:|SEQUENCE,0,A,B,A,B,A,B,A,B,C,D,C,D,C,D,C,D

```

```

// 90 A=14:b=18:c=19:d=23:|SEQUENCE,1,A,B,A,B,A,B,A,B,C,D,C,D,C,D,C,D
//100 |SGET,0,0:|SWINDOW,0,30,159,0,199:|SPUT,0,40,25:|ANIMATE,I,0:|STIX,12
//110 FOR I=2 TO 4:|SGET,I,0:|SWINDOW,I,30,159,0,199:|SPUT,I,I*25+40,I*27
//120 |SDIR,I,RND(3)*3+1,0:|ANIMATE,I,0:NEXT
//130 FOR I=5 TO 6:|SGET,I,14:|SPUT,I,(I-5)*16,(I-5)*40+20
//140 |SDIR,I,0,(I-4)+1:|ANIMATE,I,1:NEXT
//150 |SGET,7,10:|SGET,8,11:|SGET,9,13:|SDIR,7,-1,3:|SDIR,8,-4,2:|SDIR,9,2,2
//160 FOR I=7 TO 9:|SWINDOW,I,30,140,120,199:|REPON,I
//170 |SPUT,I,RND(8)*80+40,RND(9)*40+130
//180 |COLLTEST,I,@A%:IF A%=0 THEN |SPUT,I:GOTO 170
//190 NEXT:|COLOUR
//200 |MOVEALL:GOTO 200

// BIRD DEMO (DEMO3)
//
// GLENCO SOFTWARE 19-SEPT-1989
//
// Demonstrates individual windows for sprites and automatic animation.
include "cpc6128.ccz80";
include "SpritesAlive.ccz80";

word a, b, c, d, i;

sp_load();
sp_draw("demoa"); mode(0); sp_inkblack();
a = 0; b = 4; c = 5; d = 9; sp_sequence(0, a, b, a, b, a, b, a, b, c, d, c, d, c, d, c, d);
a = 14; b = 18; c = 19; d = 23; sp_sequence(1, a, b, a, b, a, b, a, b, c, d, c, d, c, d, c, d);
sp_sget(0, 0); sp_swindow(0, 30, 159, 0, 199); sp_sput(0, 40, 25); sp_animate(i, 0);
sp_stix(12);
for (i = 2; i <= 4; ++i)
{
    sp_sget(i, 0); sp_swindow(i, 30, 159, 0, 199); sp_sput(i, i * 25 + 40, i * 27);
    sp_sdir(i, rand() % 4 + 1, 0); sp_animate(i, 0);
}
for (i = 5; i <= 6; ++i)
{
    sp_sget(i, 14); sp_sput(i, (i - 5) * 16, (i - 5) * 40 + 20);
    sp_sdir(i, 0, (i - 4) + 1); sp_animate(i, 1);
}
sp_sget(7, 10); sp_sget(8, 11); sp_sget(9, 13); sp_sdir(7, - 1, 3); sp_sdir(8, - 4, 2);
sp_sdir(9, 2, 2);
for (i = 7; i <= 9; ++i)
{
    sp_swindow(i, 30, 140, 120, 199); sp_repon(i);
    line170:
    sp_sput(i, rand() % 81 + 40, rand() % 41 + 130);
    if (sp_colltest(i) == 0) { sp_sputunique(i); goto line170; }
}
sp_colour();
line200:
sp_moveall(); goto line200;

```

Ejemplo 4:

```

// 10 ' KNOCKOUT DEMONSTRATION (DEMO4)
// 20 '
// 30 ' GLENCO SOFTWARE
// 40 '
// 50 |ERASE:MEMORY &4400:|DRAW,"DEMOB"
// 60 RESTORE 200:READ hilevel:level=0:|INKBLACK:MODE 0:LOCATE 6,1:PRINT"KNOCKOUT
// 70 FOR I=32 TO 192 STEP 32:|SCENERY,35,12,I:|SCENERY,35,137,i:NEXT
// 80 FOR i=27 TO 120 STEP 22:|SCENERY,34,i,192:NEXT:x=-1:y=0:a=3:a%=0
// 90 hit=0:i=55:FOR y=1 TO 6:READ a$:FOR x=1 TO 9:a%=VAL(MID$(a$,x,1)):IF a%<>0 THEN
|SGET,i+1,a%+36:|COLLIDE,i+1,2:|REPON,i+1:|SPUT,i+1,x*12+17,180-y*14:i=i-1
//100 NEXT x,y:score=55-i
//110 |SGET,0,32:|SGET,2,33:|SPUT,0,60,8:a%=RND*90+30:|SPUT,2,a%,20:|STIX,12
//120 |STIXSPEED,1,1,2,2:|SDIR,2,1,3:|COLLIDE,2,3:|REPON,2:|COLOUR:|WAIT,30
//130 |SOUND,0,135,0,0,50,0,15,5:|SATR,2,4,0
//140 |MOVE,0,2:|SYPOS,2,@a%:IF a%<13 THEN 170
//150 |MOVEHIT,2,@a%:IF a%<>0 THEN 140
//160 |HIT,2,@a%:IF a%=255 THEN 140 ELSE |SPUT,a%:hit=hit+1:IF score=hit THEN
|SPUT,2:|SPUT,0:|RESET:level=level+1:IF level=hilevel THEN GOTO 170 ELSE 90
//165 GOTO 140
//170 LOCATE 1,1:PRINT"GAME ENDED AGAIN Y/N"
//180 A$="" :WHILE A$="" :A$=INKEY$:WEND:IF UPPER$(A$)="Y" THEN |RESET:GOTO 60

```

```

//190 IF UPPER$(A$)="N" THEN |RESET:|CLS:|WP:END ELSE 180
//200 DATA 9,"123454321","123454321","123454321","123454321","123454321","123454321"
//210 DATA "111111111","222222222","333333333","444444444","555555555","666666666"
//220 DATA "000134000","001133400","011133440","001133400","000134000","6000000006"
//230 DATA "124000421","240000042","400222004","002222200","336636633","663363366"
//240 DATA "111000111","100000001","100111001","100111001","100000001","111000111"
//250 DATA "000000000","003636300","006300000","003606300","006303600","003636300"
//260 DATA "000000000","001212100","002101200","001212100","002101200","001202100"
//270 DATA "000000000","004545400","005404500","004545400","005454000","004505400"
//280 DATA "000000000","002505200","005202500","002525200","000002500","002525200"

// KNOCKOUT DEMONSTRATION (DEMO4)
//
// GLENCO SOFTWARE
//
include "cpc6128.ccz80";
include "SpritesAlive.ccz80";

const hilevel = 9;
word level, score, i, a, x, y, hit, data_pointer;
string acode[9];
array word data = {
"123454321","123454321","123454321","123454321","123454321","123454321",
"111111111","222222222","333333333","444444444","555555555","666666666",
"000134000","001133400","011133440","001133400","000134000","6000000006",
"124000421","240000042","400222004","002222200","336636633","663363366",
"111000111","100000001","100111001","100111001","100000001","111000111",
"000000000","003636300","006300000","003606300","006303600","003636300",
"000000000","001212100","002101200","001212100","002101200","001202100",
"000000000","004545400","005404500","004545400","005454000","004505400",
"000000000","002505200","005202500","002525200","000002500","002525200"
};

sp_load();
sp_draw("demob");
line60:
data_pointer = data; level = 0; sp_inkblack(); mode(0); locate(6, 1); prints("knockout ");
for (i = 32; i <= 192; i += 32) { sp_scenery(35, 12, i); sp_scenery(35, 137, i); }
for (i = 27; i <= 120; i += 22) sp_scenery(34, i, 192);
x = -1; y = 0;
line90:
hit = 0; i = 55;
for (y = 1; y <= 6; ++y)
{
strcpy(acode, **data_pointer); data_pointer += 2;
for (x = 1; x <= 9; ++x)
{
a = *(acode + x - 1) - '0';
if (a != 0) { sp_sget(i + 1, a + 36); sp_collide(i + 1, 2); sp_repon(i + 1); sp_sput(i
+ 1, x * 12 + 17, 180 - y * 14); --i; }
}
}
score = 55 - i;
sp_sget(0, 32); sp_sget(2, 33); sp_sput(0, 60, 8); a = rand() % 91 + 30; sp_sput(2, a, 20);
sp_stix(12);
sp_stixspeed(1, 1, 2, 2); sp_sdir(2, 1, 3); sp_collide(2, 3); sp_repon(2); sp_colour();
sp_wait(30);
sp_sound(0, 135, 0, 0, 50, 0, 15, 5); sp_sattr(2, 4, 0);
line140:
sp_moverange(0, 2); if (sp_sypos(2) < 13) goto line170;
if (sp_movehit(2) != 0) goto line140;
a = sp_hit(2); if (a == 255) goto line140;
else { sp_sputunique(a); if (score == ++hit) { sp_sputunique(2); sp_sputunique(0);
sp_reset(); if (++level == hilevel) goto line170; else goto line90; } }
goto line140;
line170:
locate(1, 1); prints("game ended again y/n");
line180:
a = inkey();
if (toupper(a) == 'Y') { sp_reset(); goto line60; }

```

```

if (toupper(a) == 'N') { sp_reset(); sp_cls(); sp_wp(); return; }
goto line180;

```

Ejemplo 5:

```

// 10 ' NODES DEMONSTRATION (DEMO5)
// 20 '
// 30 ' GLENCO SOFTWARE    21 September 1989
// 40 '
// 50 MEMORY &4400:|ERASE:|DRAW,"DEMOB":|NODE,"DEMO5":MODE 0:T=1:|INKBLACK
// 60 |SEQUENCE,1,1,3,1,3,10,12,10,12,7,9,7,9,4,6,4,6:INK 7,26
// 70 FOR I=2 TO 5:|SGET,I,1:|ANIMATE,I,1:|NODESPEED,I,1,1:|NODEATTR,I,T:|NODEON,I
// 80 NEXT:|SPUT,2,3,180:|SPUT,3,146,180:|SPUT,4,99,154:|COLOUR
// 90 |SPUT,5,94,98:|MAZEON
//100 |MOVEALL:GOTO 100

```

```

// NODES DEMONSTRATION (DEMO5)
//
// GLENCO SOFTWARE    21 September 1989
include "cpc6128.ccz80";
include "SpritesAlive.ccz80";

```

```
word t, i;
```

```

sp_load();
sp_draw("demob"); sp_node("demo5"); mode(0); t = 1; sp_inkblack();
sp_sequence(1, 1, 3, 1, 3, 10, 12, 10, 12, 7, 9, 7, 9, 4, 6, 4, 6); ink(7, 26, 26);
for (i = 2; i <= 5; ++i) { sp_sget(i, 1); sp_animate(i, 1); sp_nodespeed(i, 1, 1);
sp_nodeattr(i, t); sp_nodeon(i); }
sp_sput(2, 3, 180); sp_sput(3, 146, 180); sp_sput(4, 99, 154); sp_colour();
sp_sput(5, 94, 98); sp_mazeon();
line100: sp_moveall(); goto line100;

```

Ejemplo 6:

```

// 10 ' NODES DEMONSTRATION (DEMO6)
// 20 '
// 30 ' GLENCO SOFTWARE    21 September 1989
// 40 '
// 50 MEMORY &4400:|ERASE:|DRAW,"DEMOB":|NODE,"DEMO5":MODE 0:|INKBLACK
// 60 M$(1)="RANDOM":M$(2)="CHASE ":M$(3)="FLEE "
// 70 |SEQUENCE,0,17,20,17,20,13,16,13,16,13,16,13,16,13,16,18,20
// 80 |SEQUENCE,1,1,3,1,3,10,12,10,12,7,9,7,9,4,6,4,6
// 90 FOR I=2 TO 5:|SGET,I,1:|ANIMATE,I,1:|NODESPEED,I,1,1:|NODEATTR,I,1:|NODEON,I
//100 NEXT:LOAD "DEMO6.SCR",&C000:|SPUT,2,3,180:|SPUT,3,146,180:|SPUT,4,99,154
//110 |SPUT,5,94,98:|SGET,0,13:|ANIMATE,0,0:|STIX,79:|SPUT,0,3,16:|MAZEON
//120 |COLOUR:T=1:P=0
//130 P=P+1:|MOVEALL:IF P<400 THEN 130
//140 SOUND 7,100,15:T=T+1:IF T>3 THEN T=1
//150 FOR I=2 TO 5:|NODEATTR,I,T:NEXT:LOCATE 15,1:PRINT M$(T);:P=0:GOTO 130

```

```

// NODES DEMONSTRATION (DEMO6)
//
// GLENCO SOFTWARE    21 September 1989
//
include "cpc6128.ccz80";
include "SpritesAlive.ccz80";

```

```
word i, p, t;
array byte m[21]; // Space for 3 strings of 6 characters

```

```

sp_load();
sp_draw("demob"); sp_node("demo5"); mode(0); sp_inkblack();
strcpy(m, "random"); strcpy(m + 7, "chase "); strcpy(m + 14, "flee ");
sp_sequence(0, 17, 20, 17, 20, 13, 16, 13, 16, 13, 16, 13, 16, 13, 16, 18, 20);
sp_sequence(1, 1, 3, 1, 3, 10, 12, 10, 12, 7, 9, 7, 9, 4, 6, 4, 6);
for (i = 2; i <= 5; ++i) { sp_sget(i, 1); sp_animate(i, 1); sp_nodespeed(i, 1, 1);
sp_nodeattr(i, 1); sp_nodeon(i); }
openin("demo6.scr"); load(#C000); closein();
sp_sput(2, 3, 180); sp_sput(3, 146, 180); sp_sput(4, 99, 154);
sp_sput(5, 94, 98); sp_sget(0, 13); sp_animate(0, 0); sp_stix(79); sp_sput(0, 3, 16);
sp_mazeon();
sp_colour(); t = 1; p = 0;
line130: sp_moveall(); if (++p < 400) goto line130;
sound(7, 100, 15, 12, 0, 0, 0); if (++t > 3) t = 1;
for (i = 2; i <= 5; ++i) sp_nodeattr(i, t);

```

```
locate(15, 1); prints(m + (t - 1) * 7); p = 0; goto line130;
```