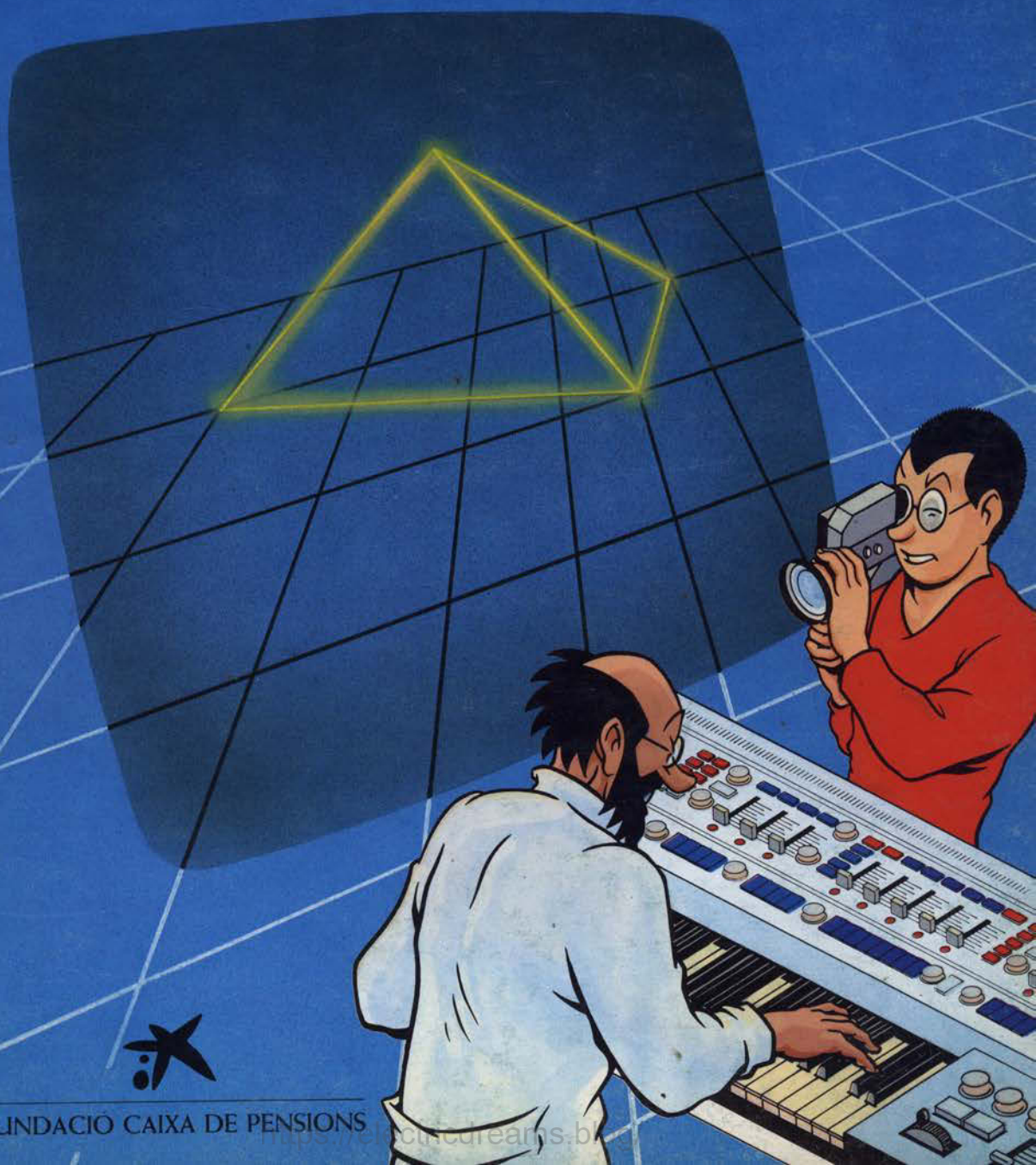


connecta el micro



IMATGES I SONS



connecta el micro

IMATGES I SONS



FUNDACIÓ CAIXA DE PENSIONS

<https://electricdreams.blog/>

Edita:
FUNDACIÓ CAIXA DE PENSIONS

President Executiu de la Fundació
Caixa de Pensions:
JOSEP VILARASAU I SALAT

Vice-President Executiu:
RICARD FORNESA I RIBÓ

Director Executiu de la Fundació
Caixa de Pensions:
JOAN JOSEP CUESTA I TORRES

Cap de Control de l'Obra Social de la Caixa de Pensions:
JOSEP MARIA ARENAS I PASCUAL

Coordinadors de l'edició:
JORDI SALA
ALBERT SÒRIA

Autors:
LLORENÇ HUGUET I ROTGER*, DAMIÀ CASAS I PESSAFERRER*,
FRANCESC FRANCO I JACOBO, MERCÉ GRIERA I FISA*
*U.A.B. Departament d'Informàtica

Disseny de portada i compaginació:
EQUIP 30/53

Dibuix:
ROGER

Fotografia:
ESPARBÉ

Fotocomposició:
CATALANA DE FOTOCOMPOSICION, S.A.

Gestió edició:
MUNDO CIENTÍFICO

Primera edició: maig 1985

© LLORENÇ HUGUET I ROTGER, DAMIÀ CASAS I PESSAFERRER,
FRANCESC FRANCO I JACOBO, MERCÉ GRIERA I FISA. 1985

Tots els drets d'aquesta edició:
FUNDACIÓ CAIXA DE PENSIONS, Via Laietana, 56, 08003 Barcelona

Impressió:
TONSA, Herrera-Alza. Donostia

ISBN: 84-505-1476-2
Dipòsit legal: S.S. 300-85

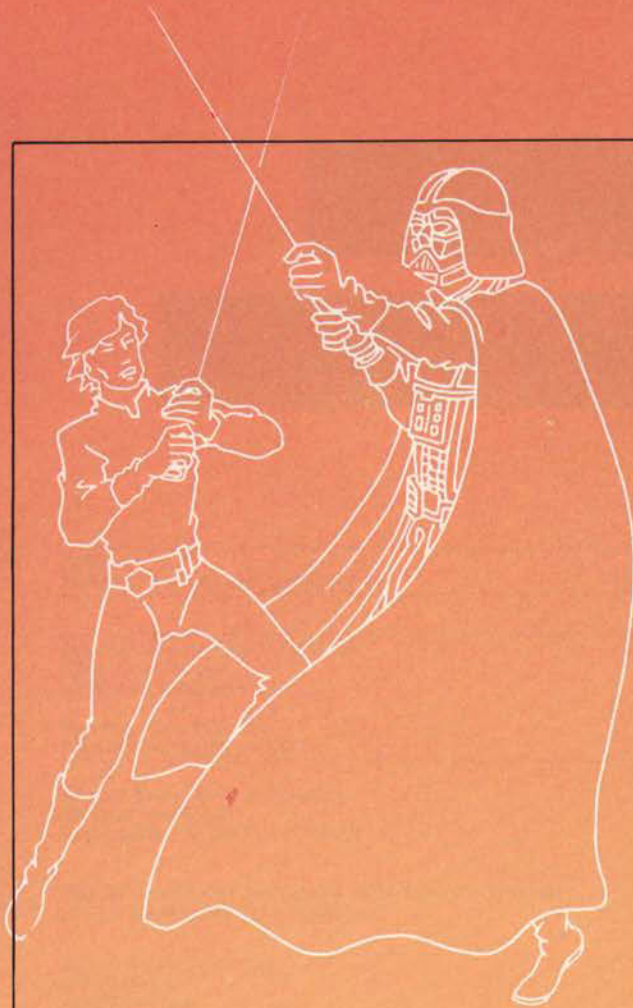
Les imatges i els sons:

una forma d'expressió



3





A la vida de cadascun de nosaltres hi ha moments que no voldríem oblidar mai. Ens agradaria tenir a la nostra disposició no només aquelles imatges sinó també allò que deia la gent i el soroll que ens envoltava. Voldríem disposar físicament d'allò que ara només és un record.

Quan escoltem música i sentim que ens agrada trobarem, potser, la necessitat d'explicar als altres allò que hem sentit i l'impacte que ens ha produït. Però, si provem de fer-ho, de seguit ens adonem que no n'hi ha prou amb la paraula per comunicar allò que hi ha a dins nostre. Si anem al cinema i expliquem la pel·lícula a un amic, per més que ens hi esforcem, hi ha molts detalls que se'ns escapen i molts moments que no sabem com dir. Quan mirem la cara del company i notem que no se'n fa el càrrec, de segur que acabem l'explicació tot dient-li: "Has d'anar a veure-la!".

El llenguatge no pot expressar certes situacions ni certes sensacions. És allò que hom ha dit: "Una imatge, val més que mil paraules!"

Tant per nosaltres mateixos, per trobar la nostra pròpia identitat individual, com per comunicar-nos amb la resta de la societat, necessitem la imatge i del so. Són dos elements que ens són del tot imprescindibles.



Quelcom més que records i imaginacions



La necessitat de perpetuar certes situacions per deixar constància d'un fet, és un problema que començà a preocupar l'home fa milers d'anys. Si anem a Cogull podem veure, a les parets d'aquelles coves, imatges d'animals. Això fa pensar que els nostres avantpassats volien que la resta de la gent que no havia assistit a la cacera sabés quina mena de peça havien matat, la seva grandària i com s'ho havien fet per atrapar-la.

En anar passant el temps, l'home trobà nous suports per guardar les imatges: va passar de la pedra al paper, al paper i a la roba. Si fem una volta pel Museu d'Història de la Ciutat, ens serà fàcil d'imaginar com vivia la gent de fa un grapat d'anys.





La tècnica va anar avançant i es va arribar a la fotografia i al cinema. Aquest dos invents van suposar guanyar "la batalla al temps", perquè ens permeten de veure què passava fa molts anys: podem tenir imatges de qualsevol època. Però la cosa no acabà aquí, i sorgí la televisió. Aquest invent ens va eliminar les distàncies: podem tenir imatges de qualsevol lloc de la terra sense desplaçar-nos.

Els discs, les cintes, la ràdio i el telèfon solucionaren el problema espai-temps amb els sons.

Ara, a més de tot això, la nostra societat disposa de la informàtica. Però, què ens aporta de nou en el camp de les imatges i els sons?

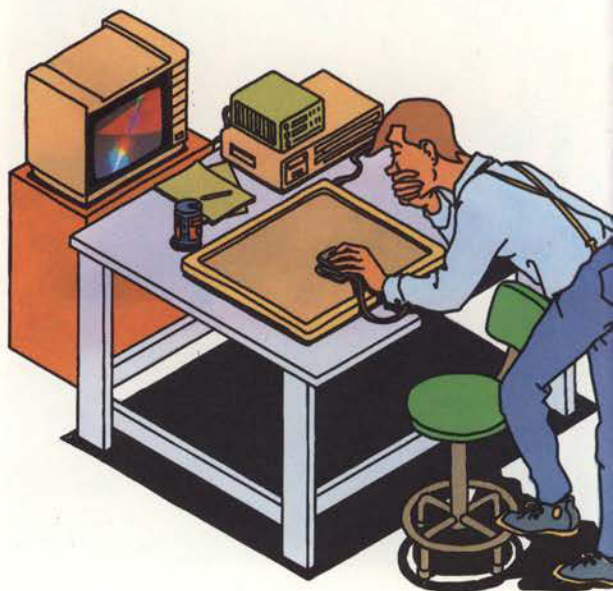
Els ordinadors sorgiren en un principi per solucionar problemes de càlcul i d'ordenació de dades. Per tant, allò que es codificava, s'escrivia amb zeros i uns: eren només informacions numèriques. A mitjan els anys seixanta, el professor Sutherland, a l'Institut Tecnològic de Massachusets (MIT), començà a codificar informacions visuals. També es codificaren (és a dir, s'introduïren en un ordinador) els sons.

En aquell moment, l'ordinador deixà de ser una eina de només ajut a científics, tècnics, administratius, etc., per convertir-se en un instrument al servei de l'art.

A part de les imatges i els sons que provenen de fets reals, l'home, ja fa temps començà a interessar-se per representar imatges imaginades; pensar, per exemple, en Picasso.



Imatge reproduïda per l'ordinador

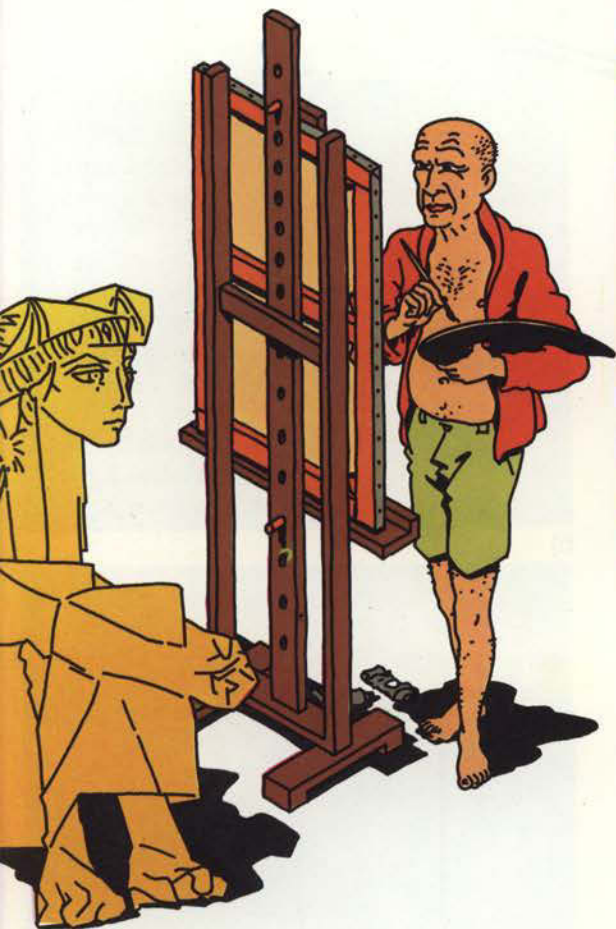


Els ordinadors tenen capacitat d'emmagatzemar imatges tant d'un tipus com de l'altre: només cal canviar els perifèrics d'entrada. Per a les imatges que provenen de fets reals, cal usar una càmera fotogràfica o una càmera de vídeo, mentre que, per a les altres, val un bolígraf òptic sobre una taula digitalitzadora.

Fins ara, només s'ha dit que allò que aporten de nou els ordinadors en aquest terreny és un canvi de suport: es poden tenir imatges i sons dins la memòria d'un ordinador; els tenim tots junts, ocupen poc espai i no es trenquen. Però això no seria pas important sinó fos que a més, ens ofereixen la possibilitat de manipular-ho .



Manipulació d'una imatge



A una imatge dins un ordinador, se li pot canviar el color, la grandària, ampliar-ne un tros, de dues fer-ne una, etc, i tot en un moment. A la música, se li pot canviar el to, el ritme, l'instrument que la toca, etc. Són coses que, sense un ordinador, representarien hores o dies de feina. I això encara no és tot. El que és més sorprenent és que, fins ara, allò que hom

Programem

en BASIC

podia plasmar en un quadre o en un pentagrama era quelcom que havia vist o sentit, o bé quelcom que havia imaginat. Els ordinadors ofereixen (fent variacions sobre les codificacions) la possibilitat de crear un so o una imatge que ni el mateix creador no coneix.

L'ordinador ens dona la possibilitat d'escapar de les nostres percepcions sensorials. Podem crear imatges, colors i sons que no existeixen a la realitat, ni tampoc a la ment de la persona que els ha fet.

Cal parar-se a pensar que, perquè tot això hagi estat possible, ha calgut que un grup de científics (informàtics, matemàtics, físics,...) no es deixessin embadalir per un paisatge o per una música. Hem hagut de menester un grup de gent que renunciés al plaer de les sensacions per tal d'analitzar, i deixar de pensar en la llum, el color i el so com a fenòmens plàstics per tal de veure-ho com a ONES, on només varien les mides (amplitud, freqüència, etc).

A través d'aquestes pàgines, podrem veure que aquest treball d'anàlisi ha estat enormement recompensat, ja que ha permès a l'home de tenir al seu abast una eina nova, amb infinitat de possibilitats, per estimular la seva capacitat de crear.

Aquest llibre té dues parts ben diferenciades: a la primera, es parla d'imatges, i a la segona, de sons. A cadascuna d'elles es pretén fer una presentació dels conceptes i de les principals tècniques que s'usen avui, així com presentar, després, alguna aplicació concreta, i acabar veient què es pot fer amb el microordinador SINCLAIR, tant amb imatges com amb sons. També, tal com s'havia fet en el primer llibre, donarem alguns programes, sense més explicacions, per als microordinadors COMMODORE i el PC d'IBM.

Abans de començar, però, hem col·locat una part de BASIC, que tracta de bucles i variables indexades. Encara que no estigui directament relacionada amb aquests temes, la considerem essencial per a continuar programant.

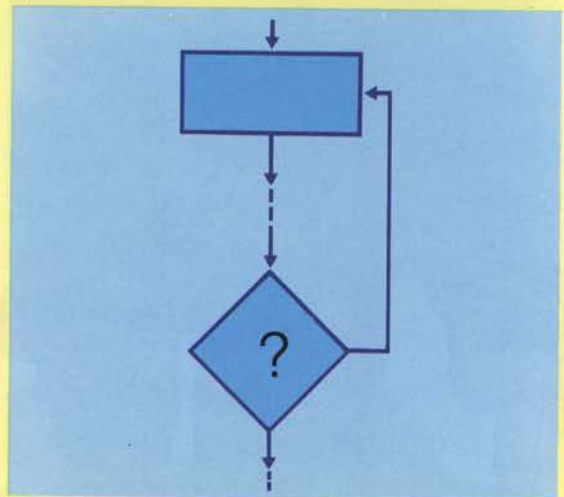
Bucles

Ja coneixem què vol dir "iteratiu". Es tracta de repetir un seguit d'accions sota unes determinades condicions; fins ara n'hem vist dues possibilitats:

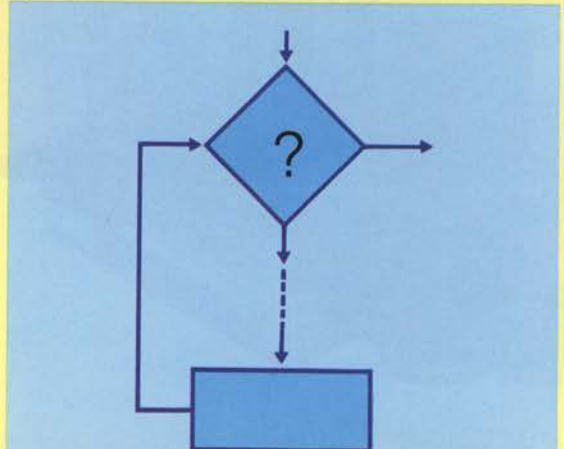
- de primer, actuar, i després, preguntar si hem de repetir o no;
- de primer preguntar, i segons sigui la resposta, actuar o no.

És allò que representàvem a l'ordinograma com:

a)



b)





Aquesta qüestió se'ns plantejava en problemes en els quals no coneixíem quantes vegades s'havia d'executar el grup d'accions a repetir. Aquest grup d'accions és conegut com a BUCLE. Suposem que sabem des del començament la quantitat de vegades que s'ha d'executar un bucle. Per exemple, volem escriure 10 vegades seguides el nostre nom per pantalla. L'algorisme seria:

- 1 - **Escriure** nom
- 2 - **Escriure** nom
- 3 - **Escriure** nom
- "
- "
- 10 - **Escriure** nom

A l'ordinograma passariem 10 vegades, entre l'INICI i el FI



i al programa posariem:

```
10 PRINT "nom"
20 PRINT "nom"
"
```

```
100 PRINT "nom"
```

Quina cosa més avorrida! Havíem quedat que els ordinadors, entre d'altres coses, servien per simplificar els treballs rutinaris. Per fer-ho així, val més plegar!

Per arreglar la situació, cal usar un **COMPTADOR**, això és, una variable numèrica tal que el seu contingut creix, o disminueix, a partir d'un valor determinat, fins a arribar a un altre.

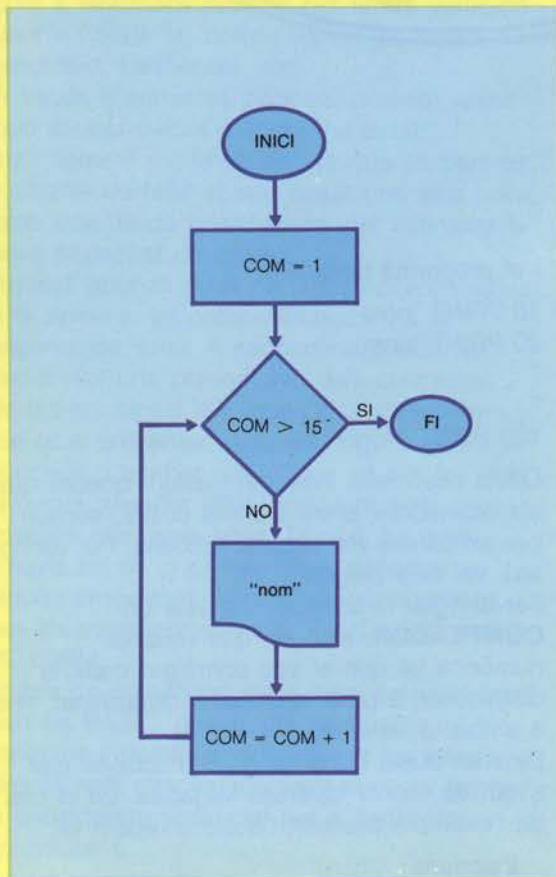
Dins un bucle hi ha un grup d'accions que s'han de repetir diverses vegades. En el cas de l'exemple anterior, l'acció a repetir és:

Escriure "nom"

Farem servir un comptador per controlar si hem escrit, o no, 10 vegades el nom. Prendrem com a comptador, per exemple, la variable COM, i li farem prendre tots els valors entre 1 i 10, de manera que, quan valgui 10, el programa s'acabarà. L'algorisme serà:

- 1 - **Inicialitzar** COM a 1
- 2 - **Si** COM és més gran que 10 aleshores **anar a** 6
- 3 - **Escriure** "nom"
- 4 - **Incrementar** COM en 1
- 5 - **Anar a** 2
- 6 - **Acabar**

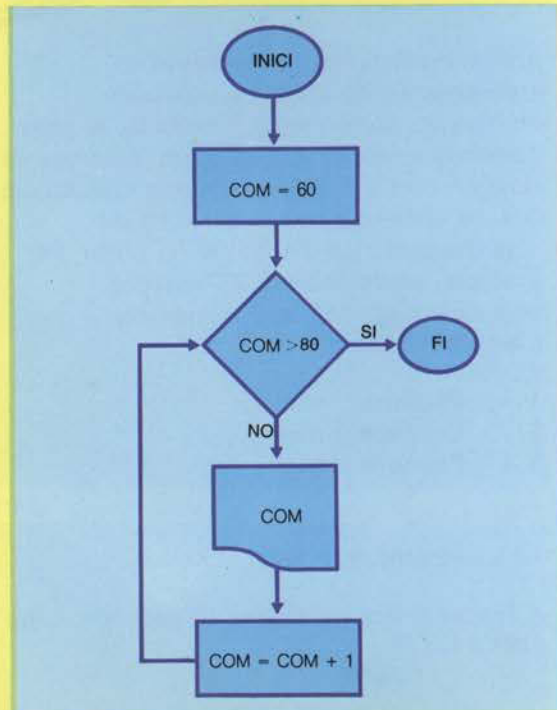
Així el comptador COM prendrà els valors 1,2,3,...,10 i, per a cadascun d'ells, escriurà el nostre nom. L'ordinograma serà:



El mateix problema es planteja si volem escriure tots els nombres entre 60 i 80 (ambdós inclosos). Usarem un comptador que d'entrada valgui 60, escriurem el seu contingut, l'incrementarem en un, tornarem a escriure el seu contingut i així anar fent, fins haver escrit el 80. L'algorisme serà:

- 1 - **Inicialitzar** COM a 60
- 2 - **Si** COM és més gran que 80 aleshores **anar a** 6
- 3 - **Escriure** el contingut de COM*
- 4 - **Incrementar** COM en 1
- 5 - **Anar a** 2
- 6 - **Acabar**

L'ordinograma serà:



Fins aquí, això que hem fet és parlar de bucles dins els ordinogrames i dins els algorismes. Per poder escriure bucles dins un programa es necessiten dues noves sentències:

Sentència **FOR ...TO...** i Sentència **NEXT**

Quan un bucle s'ha de repetir una quantitat de vegades coneguda, hem quedat què faríem servir un comptador.

L'estructura bàsica d'un bucle d'aquest tipus és la següent:

- Assignar al comptador un valor inicial
- Preguntar si el comptador excedeix el valor final
- Fer un seguit d'accions
- Incrementar comptador
- Tornar-ho a preguntar

Això, en un programa, s'escriuria com:

```
90 LET comptador = valor inicial
100 IF comptador > valor final THEN GOTO 210
110 sentències
```

“

```
200 LET comptador = comptador + 1 :
    GOTO 100
210 sentències
```

(Suposant que la primera acció del bucle fos a la línia 110 i l'última a la 190).

Hi ha, però, dues sentències que, combinades, fan el mateix; i, al programa, es posen així:

```
90 FOR comptador = valor inicial TO valor final
100 sentències
```

“

```
190 NEXT comptador
```

La primera sentència (FOR....TO....) equival a :

- assignar a comptador el seu valor inicial
FOR comptador = valor inicial
- establir el valor final del comptador
TO valor final
- Preguntar si el comptador excedeix el valor final

La segona sentència (NEXT...) fa el següent:

- incrementar el valor del comptador
- tornar enrera el flux del programa anant a la pregunta que controla el bucle.

Per tant, els dos exemples anteriors s'escriurien :

```
10 FOR COM = 1 TO 10:
    REM control del bucle i inicialització
20 PRINT "nom":
    REM acció a repetir
30 NEXT COM:
    REM incrementar COM
```

i el segon

```
10 FOR COM = 60 TO 80:
    REM control del bucle i inicialització
20 PRINT COM:
    REM acció a repetir
30 NEXT COM:
    REM incrementar COM
```

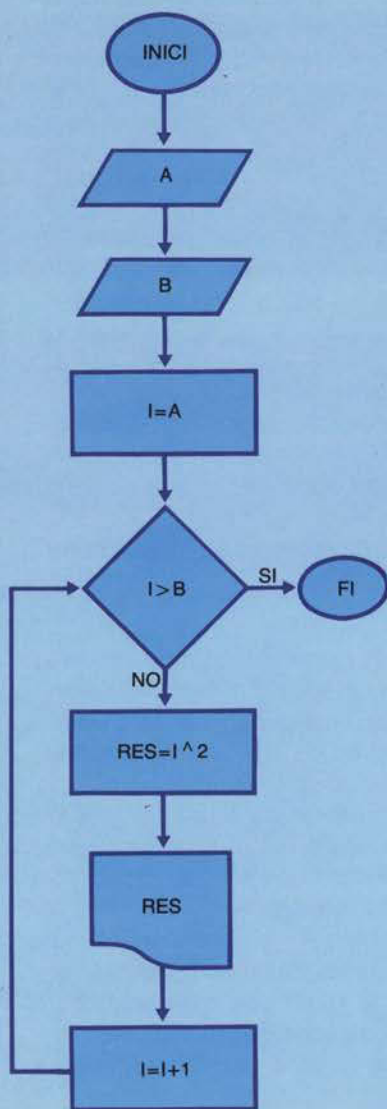
Abans hem dit que el comptador ha de ser una variable numèrica. En canvi, el valor inicial i el valor final poden ser tant una constant (com en els exemples) com una variable. En el ZX Spectrum, el nom de la variable comptador només pot ser una lletra; així, a la línia 10 de l'últim programa només hauríem de posar C en lloc de COM.

Vegem un nou exemple

Tenint dos valors, A i B, que s'entraran per teclat, es tracta de calcular el quadrat de tots els números que hi ha entre ambdós (A i B inclosos). L'algorisme serà ben senzill, farem servir la variable I com a comptador.

- 1 - **Llegir** A
- 2 - **Llegir** B
- 3 - **Inicialitzar** I amb el valor d'A
- 4 - **Si** I és més gran que B
aleshores anar a 9
- 5 - **Calcular** el quadrat d'I
- 6 - **Escriure** el resultat
- 7 - **Incrementar** I en 1
- 8 - **Anar a** 4
- 9 - **Acabar**

l'ordinograma:



i el programa :

```
10 INPUT " VALOR INICIAL"; A
20 INPUT " VALOR FINAL"; B
30 FOR I = A TO B
40 LET RES = I^2
50 PRINT " EL QUADRAT DE ";I;" ES:"; RES
60 NEXT I
70 END
```

Els exemples que hem vist fins ara tenen una limitació: a les sentències FOR...TO... i NEXT... el comptador només es pot incrementar d'un en un. Hi ha problemes, però, en els quals pot ser interessant que el comptador variï de forma diferent: en lloc d'incrementar-lo, podem voler decrementar-lo; o bé que aquesta variació no sigui d'un en un, sinó de tres en tres, o de 0.5 en 0.5. Suposem que desitgem tenir la taula de les arrels quadrades dels números que hi ha entre 0 i 10 però de 0.25 en 0.25 (0, 0.25, 0.5,..., 6.75, 7, 7.25,..., 10). Hi ha una manera de dir a l'ordinador que el comptador, en aquest cas, no varia d'un en un sinó de 0.25 en 0.25 :

FOR ... TO ... **STEP** 0.25

i el programa seria

```
10 FOR I = 0 TO 10 STEP 0.25
20 LET ARREL = I^0.5 : REM càlcul de l'arrel
30 PRINT ARREL
40 NEXT I
```

(Convé recordar que fer l'arrel quadrada d'un número equival a fer la potència d'aquest amb exponent 0.5)

En el cas que el comptador s'hagués de decrementar en un valor constant, per exemple 2, n'hi hauria prou amb posar: FOR comptador = valor inicial TO valor final STEP -2; però, com és lògic, en aquest cas el valor inicial ha de ser més gran que el valor final. Cal tenir present que:

- Per a accions repetitives, que es poden controlar amb un comptador que variï de forma constant, tenim la següent estructura:

FOR comptador = valor inicial TO valor final STEP variació.

- Si la variació és d'un en un, no fa falta posar STEP variació
- Comptador ha de ser una variable numèrica.
- Els valor inicial y final, i la variació, poden ser tant una variable numèrica com una constant.

Calaixos de caixes: variables indexades

Provem de fer el següent:

```
10 INPUT "ENTRA EL VALOR INICIAL ";VI
20 INPUT "ENTRA EL VALOR FINAL ";VF
30 INPUT "ENTRA LA VARIACIO ";VA
40 FOR COM = VI TO VF STEP VA
50 LET ARREL = COM^0.5
60 PRINT "ARREL QUADRADA DE ";
COM;" = "; ARREL
70 NEXT I
```

- En alguns micros, l'estructura d'un FOR no es correspon amb la que hem assenyalat aquí. En lloc de preguntar primer, i, després, si cal, fer el bucle, de primer es fa el bucle i, després, es pregunta. Per tant, ens podrem trobar que, si posem :

```
FOR I = 1 TO 0
"
"
NEXT I
```

el bucle es faci una vegada.

- Es pot posar un bucle dins d'altre, però no es poden entrecruar:

Bé

```
FOR I = 1 TO 10
FOR J = 1 TO 14
"
"
NEXT J
NEXT I
```

Malament

```
FOR I = 1 TO 10
FOR J = 1 TO 14
"
"
NEXT I
NEXT J
```

Hem vist que les variables són molt útils per manipular la informació, tant la numèrica com l'alfanumèrica. Només les hem fet servir, però, per guardar-hi un nom, o bé un resultat, etc...: sempre, petites quantitats d'informació. Si volguéssim tenir a la memòria, per exemple, el telèfon de les persones que viuen en un cert bloc de pisos (un pis per cada planta), podríem fer el següent:

Suposem que el bloc té 10 plantes d'un sol pis i que volem entrar els telèfons per teclat. En principi, necessitarem 10 variables, un per pis. A l'hora de fer la lectura, ens trobarem amb problemes.

No es pot fer:

```
10 FOR I = 1 TO 10
20 INPUT "TELEFON? ";PIS$
30 NEXT I
```

ja que la variable PIS\$, quan s'acabi el programa, només contindrà el telèfon de l'últim pis (a cada nova lectura esborrem el contingut de l'anterior).

Sembla, per tant, que només queda una solució.

- escriure un programa amb 10 sentències de lectura, una darrera l'altra. programa quedaria, més o menys:

```
10 INPUT "PRIMER PIS? ";P1$
20 INPUT "SEGON PIS? ";P2$
"
"
100 INPUT "DESE PIS? ";P10$
```

Això és carregós i incòmode. A més, si volem fer el mateix per un altre bloc de pisos tornarem a trobar-nos amb el mateix problema. El BASIC, per tal de solucionar problemes d'aquest tipus, disposa d'un nou tipus de dades: les VARIABLES INDEXADES.

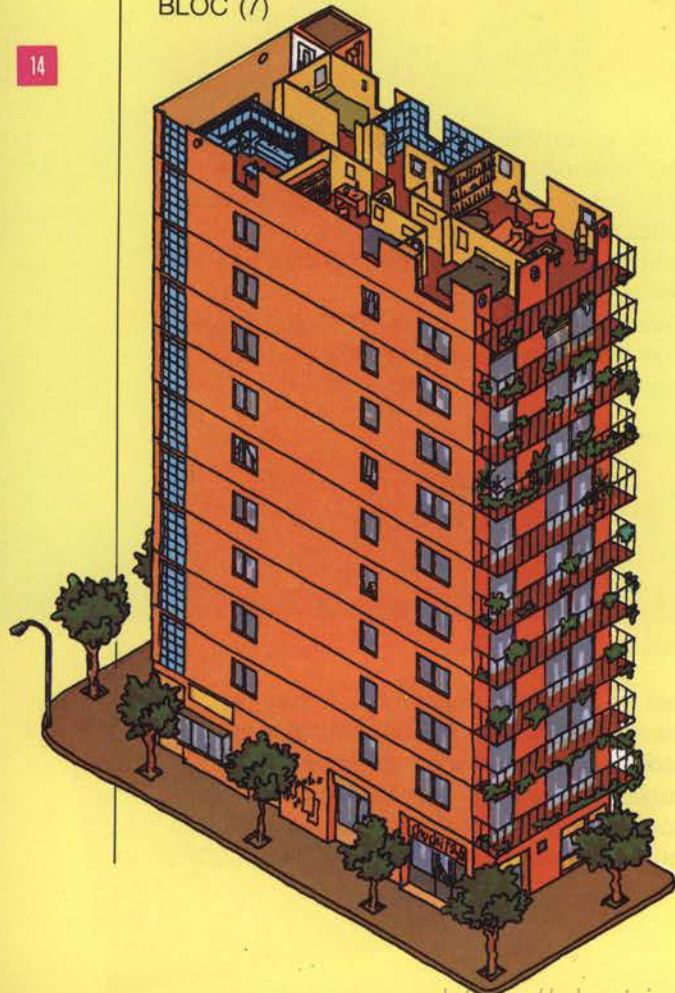
Una variable indexada és com un calaix ple de caixes ordenades i numerades. Per fer referència a la variable indexada (calaix), n'hi ha prou amb donar-li un nom; i per trobar les caixes, n'hi ha prou amb dir el número d'ordre. Una variable indexada, per tant, ens permet d'agrupar un seguit de variables sota el mateix nom, diferenciant-les amb un ordre. A l'exemple, podríem dir que tenim una variable indexada anomenada BLOC que conté deu variables. Per fer referència a un element d'una variable indexada s'ha d'escriure:

nom de la variable (número d'ordre),

on el nom de la variable segueix les mateixes normes que per a les variables habituals, i el número d'ordre pot ésser una variable o una constant numèrica.

A l'exemple:

BLOC (7)



per fer referència al telèfon del veí de la setena planta.

Abans de treballar, dins un programa, amb una variable indexada, haurem d'indicar, al micro, el seu nom i els elements que podrà tenir. Això es fa amb la sentència de dimensió :

DIM nom de la variable (tamany).

En el nostre cas:

DIM BLOC (10)

Sabent que el número d'ordre pot ser una variable numèrica, el programa quedaria així:

```
10 DIM BLOC (10)
20 FOR I = 1 to 10
30 PRINT "TELEFON VEI DEL PIS ";I;" ?=";
40 INPUT BLOC(I)
50 NEXT I
```

La línia 20 fa que el programa escrigui la pregunta, a la primera passada pel FOR:

TELEFON DEL VEI DEL PIS 1 ?=

i deixarà el cursor darrera el signe = perquè hi posem el telèfon d'aquell veí. Tot seguit, com que el bucle s'ha de fer 10 vegades, escriurà:

TELEFON DEL VEI DEL PIS 2 ?=

i així fins a fer-ho 10 cops.

Cada cop que escrivim el telèfon d'un veí, aquest s'emmagatzemarà en un lloc diferent de memòria, ja que la línia 40 dirà , a la primera passada:

40 INPUT BLOC(1)

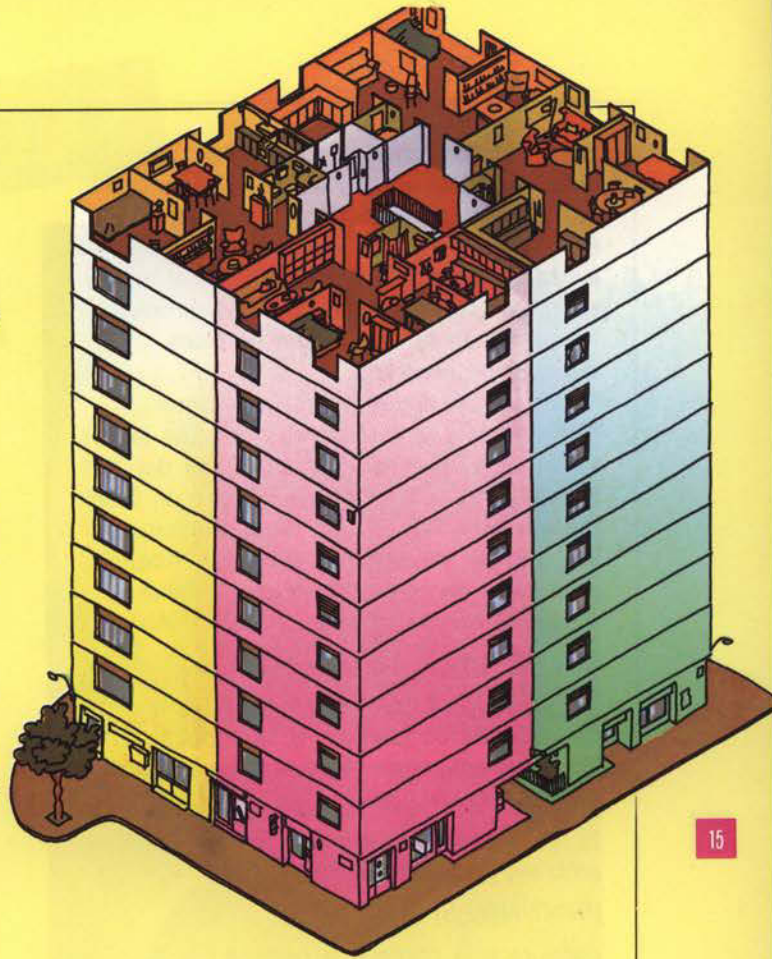
a la segona

40 INPUT BLOC(2)

i així fins l'últim telèfon, que posarà dins BLOC(10).

I si, a cada planta del bloc, hi hagués més d'un pis?

En aquest cas, hi ha un altre tipus de variable indexada que permet ser referenciada per dos números, en lloc de ser-ho per un. Si en el primer tipus era com si tinguéssim un calaix ple de caixes posades una darrera de l'altra, en aquest cas és com si tinguéssim molts d'aquests calaixos un al costat de l'altre, és a dir, les caixes queden col·locades en files i en columnes. Abans només teníem una columna: com que tots els veïns que són al pis 1 de cada planta estan els uns sobre els altres, direm que formen la primera columna; per als del pis 2, tindren la segona columna. En canvi tots els veïns que viuen a la primera planta estan a la primera fila. I així per a totes les plantes. Per tant ens faran falta dos números per localitzar un veí, el de la fila (planta) i el de la columna (pis). En BASIC podem definir variables indexades en les quals, per indicar o referenciar un dels seus elements, seran necessaris dos números (fila i columna) de la manera següent:



DIM nom de variable (quantitat de files, quantitat de columnes)

Si en el nostre cas considerem un bloc de 10 plantes i 4 pisos per planta, fariem:

```
DIM BLOC (10,4)
```

Per fer el programa que serveixi per entrar els telèfons de tots els veïns podem pensar que els entrarem per plantes. Primer entrarem els de la primera planta:

BLOC(1,1) BLOC(1,2) BLOC (1,3) BLOC(1,4)
després els de la segona

BLOC(2,1) BLOC (2,2) BLOC(2,3) BLOC (2,4)

i així fins a la desena planta:

Si representem els dos números de referència mitjançant les variables I i J, veiem que cada cop que que la I pren un valor (entre 1 i 10) la J varia d'1 fins a 4. Això vol dir que haurem de fer servir dos bucles: un que varïi la I i l'altre, dins del primer, que varïi la J.

Amb aquestes consideracions, ja podem escriure el nostre programa d'emmagatzematge dels números de telèfon de tots els veïns del nostre bloc de pisos:

```
10 DIM BLOC(10,4)
20 FOR I = 1 TO 10
30 FOR J = 1 TO 4
40 PRINT "TELEFON DEL PIS ";J;" DE LA
PLANTA ";I;" ?=";
50 INPUT BLOC(I,J)
60 NEXT J
70 NEXT I
```

Aquestes variables indexades s'anomenen TAULES, i els números que fem servir per referenciar els seus elements s'anomenen ÍNDEXS.

Hem vist dos exemples de taules: una, d'un índex; i l'altra, de dos índexs. En els dos casos hem treballat amb valors numèrics (telèfons). El número d'índex que posem indica la dimensió de la taula. En el primer programa tenim una taula d'una dimensió (vector); i en el segon, una de dues dimensions (matriu).

També hi ha taules de cadenes, que funcionen de la mateixa manera que les de valors numèrics, exceptuant el cas del ZX Spectrum on s'ha d'afegir sempre un índex (normalment l'últim) que digui al micro la longitud de les cadenes a tractar. Ademés, en aquest micro el nom d'una variable indexada només pot ser una lletra. Així:

DIM N\$(4,10,8)

voldria dir que hem creat una taula o matriu de cadenes de longitud 8, distribuïda en 4 files i 10 columnes. Aquest últim índex pot semblar molest, però té les seves utilitats:

Si no el poséssim, només podríem treballar amb les cadenes senceres:

PRINT N\$(2,5)

escriurà a la pantalla la cadena (de 8 caràcters) sencera que tenim guardada a l'element (2,5) de la matriu.

– Si el posem, podem tractar un a un els caràcters que formen la cadena.

Així:

N\$(1,3) = "IL·LUMIN"

PRINT N\$(1,3,5)

escriurà la U per pantalla, i

PRINT N\$(1,3,3)

escriurà el punt (·) per pantalla.

– Hi ha una altra possibilitat: tractar només un tros de la cadena. Si ens interessa, per exemple, escriure els tres últims caràcters de la cadena que hi ha a N\$(4,2) podem fer:

PRINT N\$(4,2,6 TO 8)

i l'ordinador ens escriurà els caràcters sisè, setè i vuitè de la cadena referenciada per (4,2) a la matriu N\$.

Les imatges



Les imatges són

una col·lecció

de punts



17

Una imatge és formada per una infinitat de punts lluminosos. Nosaltres, però, a través dels ulls només en percebem uns quants; és el cervell qui fa la resta. A partir d'aquest nombre finit, ens reconstrueix la imatge com si fos una totalitat, una cosa contínua. Pensem, per exemple, en una roba pintada o en un tapís. Es tracta d'una col·lecció de fils entrecruats que formen una trama de punts i forats. L'ull humà ho veu, però, com una cosa contínua. La imaginació elimina automàticament els forats.

També podem pensar en els pintors del segle passat, els impressionistes i els puntillistes. Ells van observar que no els calia dibuixar un contorn i després omplir-lo de color, sinó que n'hi havia prou amb un seguit de taques de color ben repartides, o simplement punts, per donar la sensació d'imatge contínua.

La fotografia funciona d'igual manera. Consta d'un gran nombre de taques de iodur o bromur de plata col·locades sobre una pel·lícula. Són uns granets microscòpics, col·locats l'un al costat de l'altre i acolorits segons sigui la intensitat de la llum. Nosaltres, però, quan mirem una fotografia veiem una imatge contínua.

*Ampliació d'un negatiu.
Podem observar els granets microscòpics.*

Tot això és perquè els nostres ulls són formats per una col·lecció de cèl·lules molt acostades, però independents. La informació captada per cadascuna d'aquestes cèl·lules (un puntet de la realitat) és transmesa al cervell, el qual processa aquesta informació, de manera que en fa una imatge contínua. A l'hora de tractar la imatge amb un ordinador, allò que farem serà intentar reproduir aquest procés; és a dir, de primer col·locarem dins l'ordinador un nombre finit de punts i, a partir d'ells, intentarem refer la imatge completa.

Del continu al discret:

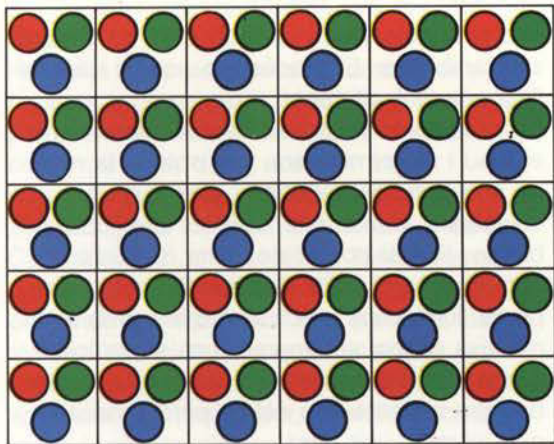
els pixels

Tot objecte il·luminat emet ones lluminoses. Aquestes poden ser transformades en senyals elèctrics usant cèl·lules fotoelèctriques. Aquests senyals poden ser convertits en ones quadrades, usant un convertidor analògic-digital. Per tant, som capaços de transformar un senyal lluminós (color, intensitat) en un seguit de zeros i uns. podem, dir, doncs, que el color és una informació tractable per un ordinador. Si tenim una imatge i la volem guardar en un ordinador, la primera cosa que farem serà quadricular-la. Cadascun d'aquests petits trossos d'imatge entra a l'ordinador de manera seqüencial; és a dir, per línies. I, dins de cada línia, l'un al costat de l'altre sempre amb el mateix ordre. Com que no entrem la

18



imatge tota de cop, sinó a petites parts, ens veiem obligats a descompondre-la en un nombre finit de parts; si no, el temps necessari per emmagatzemar-la seria infinit: mai no tindríem la imatge dins l'ordinador. Allò que es fa és descompondre-la en bandes horitzontals, totes de la mateixa amplada, i cada banda en petits quadrets tots de la mateixa grandària. Aquestes petites parts d'imatge s'anomenen **PIXELS**.



Aquesta paraula ve de la barreja de dos mots anglesos "picture element." Podríem dir, doncs, que pixel vol dir element d'imatge. Les mides dels pixels estan en funció del temps de què disposem per emmagatzemar la imatge, de la quantitat de memòria del nostre ordinador i de la precisió que en vulguem. Per exemple, si tenim una imatge partida en 1.000 pixels, i ara dividim cada costat de quadret per dos, millorem molt les formes, la imatge emmagatzemada és molt més semblant a la imatge real, però multipliquem per quatre el temps necessari per emmagatzemar-la i la quantitat de memòria que ocupa. Així, saber en quantes parts cal descompondre una imatge serà un problema que dependrà no pas només de la precisió, sinó, també de l'ordinador que tinguem. El següent pas és analitzar cada pixel. Es mira quin és el color predominant del

quadret, és a dir, el color de què n'hi ha més, i es considera tot el quadret d'aquest color, menyspreant els altres.

Ara, doncs, només ens cal definir cada pixel per dos valors numèrics, un per la posició i l'altre per al color.

El nombre de posicions i de colors possibles dependrà de la precisió de la imatge i de la capacitat de memòria de l'ordinador.

Aquest procediment s'anomena:

NUMERITZACIÓ:

ja que transforma una imatge en una sèrie de números

DIGITALITZACIÓ:

ja que transforma una imatge en ones digitals (ones quadrades),

DISCRETITZACIÓ:

ja que es passa d'una imatge contínua, formada per un nombre infinit de punts que correspondria a una quadrícula real, a una quadrícula formada per un nombre finit de punts, igualment espaiats, que correspon als nombres sencers, a valors discrets.

Fixem-nos que això ens diu que podem transformar una imatge en un conjunt de números; o, més concretament, en una barreja de zeros i uns. Aquestes imatges discretitzades i "tancades" dins la memòria d'un ordinador s'anomenen **imatges sintètiques**.

A cada col·lecció de números no li correspon només una única imatge sintètica, sinó moltes, una infinitat; perquè hi ha programes que, actuant damunt la codificació de la imatge, permeten de veure desplaçar l'objecte que representa, girar, canviar de grandària, modificar la seva forma i color. Tot això mostra que, per tenir una imatge en un ordinador, no cal tenir un objecte concret; es poden inventar conjunts de xifres, col·locar-les a la memòria i, després, veure què és el que això representa. El procés és sorprenent, ja que permet de crear una imatge sense ni tan sols imaginar-la. L'ordinador ens permet de manipular imatges, ja sigui visualment o numèricament.

Filtrem

imatges

20



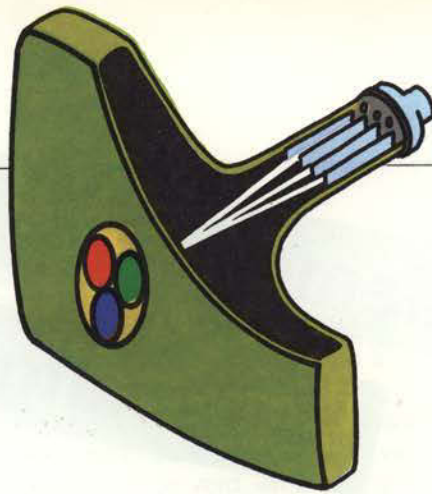
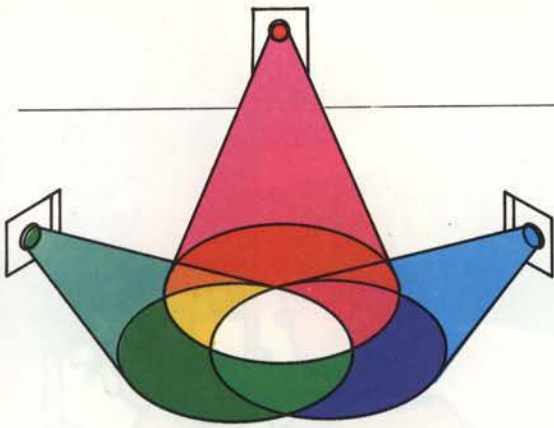
Tots sabem que amb pintura groga, blava i vermella es poden obtenir tots els colors possibles: només cal barrejar-los bé. Aquests tres colors s'anomenen colors bàsics. Per obtenir, per exemple, taronja, cal posar capes de groc i capes de vermell. Segons sigui la proporció, quedarà un taronja més clar o més vermellós. Amb pintura, els diferents colors s'obtenen per sobreposició dels colors bàsics.

Nosaltres, en fer imatges en una pantalla de televisió, allò que fem és construir els diferents colors no pas amb pots de pintura, sinó amb llum. El procés, en aquest cas, és lleugerament diferent.

Per a la llum, els colors bàsics són el verd, el blau i el vermell; ara, per obtenir la resta de colors no sobreposem, sinó que col·loquem punts microscòpics dels colors bàsics, l'un al costat de l'altre, i així l'ull humà capta el color desitjat. Si col·loquem punts lluminosos verds i vermells amb la mateixa proporció, veurem una superfície groga. Per obtenir taronja, amb la llum cal barrejar aquests dos colors però posant-hi més vermell que verd.

Així, per emmagatzemar una imatge dins un ordinador o (allò que és el mateix) per fabricar una imatge habitual de televisió, allò que es fa és **filtrar-la**; és a dir, descompondre-la en tres imatges, una per cada color bàsic. D'una imatge, n'obtenim tres.





Les pantalles de televisió tenen tres tubs, per on surten ones (electrons en moviment): a cadascun d'ells els arriba una imatge filtrada.

Per mitjà d'un sistema electromagnètic, aquests tubs recorren, tots tres a l'hora, la pantalla del televisor fent línies horitzontals, començant per dalt i acabant a baix.

La pantalla té uns 750.000 punts repartits en 250.000 grups de tres, anomenats **triades**.

Cada tríada té un puntet de fòsfor vermell, un de blau i un de verd. Els tubs estan perfectament "apuntats" a la pantalla, de manera que el tub, al qual ha arribat la imatge que ha sortit del filtre vermell, només apunta al punt de fòsfor vermell de cada tríada i, així, igualment amb els altres dos colors.

Les intensitats d'un determinat color depenen de la velocitat dels electrons, quan

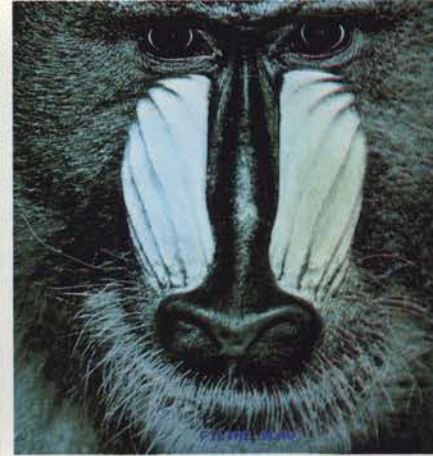
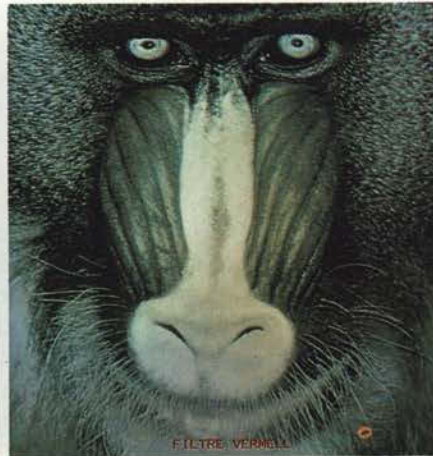
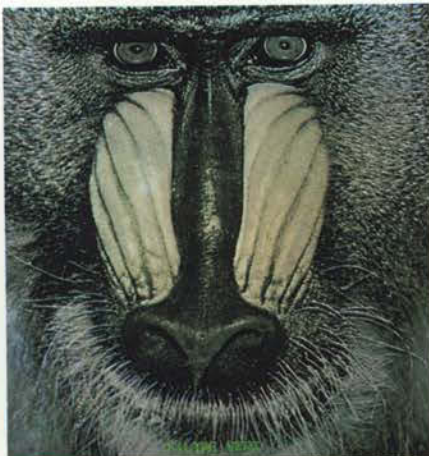
aquests xoquen contra la pantalla; i la brillantor depèn del nombre d'electrons que cauen en un determinat punt.

Per guardar, en un ordinador, una imatge de blanc i negre n'hi ha prou a quadricular-la i guardar-la, mentre que, si hi hem de guardar una imatge de color, el que hem de fer és quadricular-ne i guardar-ne tres, una per a cada color bàsic. Aquestes tres imatges seran emmagatzemades com si es tractés de tons de gris. És la pantalla, la que té capacitat o no de produir color.

Per omplir una pantalla, els tres tubs fan unes 550 passades. Una imatge és composta, doncs, per unes 550 línies. Quant al temps, resulta que omplir 25 vegades la pantalla demana un segon: és per això que veiem la imatge en moviment.

Hi ha pantalles en les quals els tubs permeten de fer 800, 1.024 o fins i tot 2.048 passades per omplir-les; aleshores obtenim imatges molt més exactes i més ben perfilades.

Els tres filtrats de la imatge



Memòria

i imatge

22

Per emmagatzemar una imatge cal guardar, per a cada punt, la intensitat del seu color. Suposem que disposem per a cada punt de 8 intensitats possibles. És a dir, a cada filtre de la imatge podem considerar que els seus punts poden tenir 8 intensitats de color, des de vermell intens, passant per un vermell pàl·lid, fins arribar al punt apagat (intensitat zero) per a la imatge que surt del filtre de vermells; i, així, igual per les altres dues imatges: la del filtre de verds i la del filtre de blaus.

Ja s'ha dit que la imatge vermella té 250.000 punts. Per a cadascun d'ells necessitaríem 3 bits per guardar la intensitat, ja que $8 = 2^3$

111 ----> màxima intensitat
= color molt fort.

110

101

100 ----> intensitat mitjana

011

010

001

000 ----> intensitat zero
= no hi ha color. Punt apagat

Així, per guardar la imatge vermella necessitaríem :

$250.000 * 3 = 750.000$ bits
 $750.000 / 8 = 93.750$ bytes
 $93.750 / 1.024 = 91$ kbytes



També necessitaríem 91 K per a la imatge verda i 91 K per a la imatge blava. Així, emmagatzemar una imatge tal com la veiem per televisió suposa $91K * 3 = 273K$.

En un micro normal de 64 K queden lliures, per al programador unes 40 K. Aleshores no podríem guardar ni una imatge i, a més, per donar la idea de moviment cal enviar 25 imatges per segon a la pantalla.

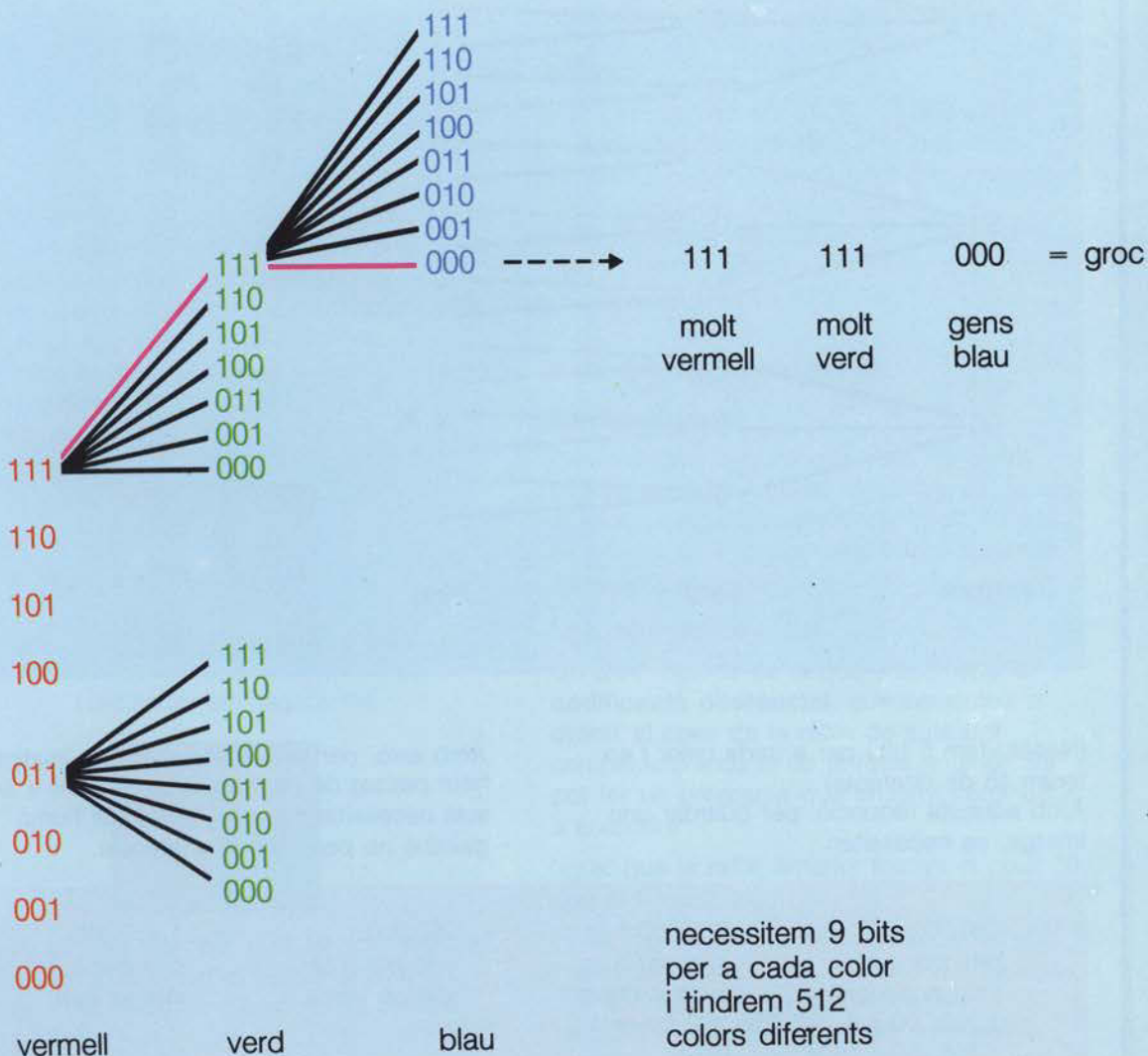
Queda clar que, per treballar les imatges amb un ordinador, s'han de quadruplicar d'una forma més grollera, o bé s'han de trobar noves tècniques d'emmagatzemament. Fent-ho com s'acaba d'explicar, imagineu-vos la quantitat de memòria necessària per guardar una pel·liculeta de 20 minuts. S'ha de reduir la quantitat d'informació que caldrà guardar!

Tècniques

d'emmagatzematge

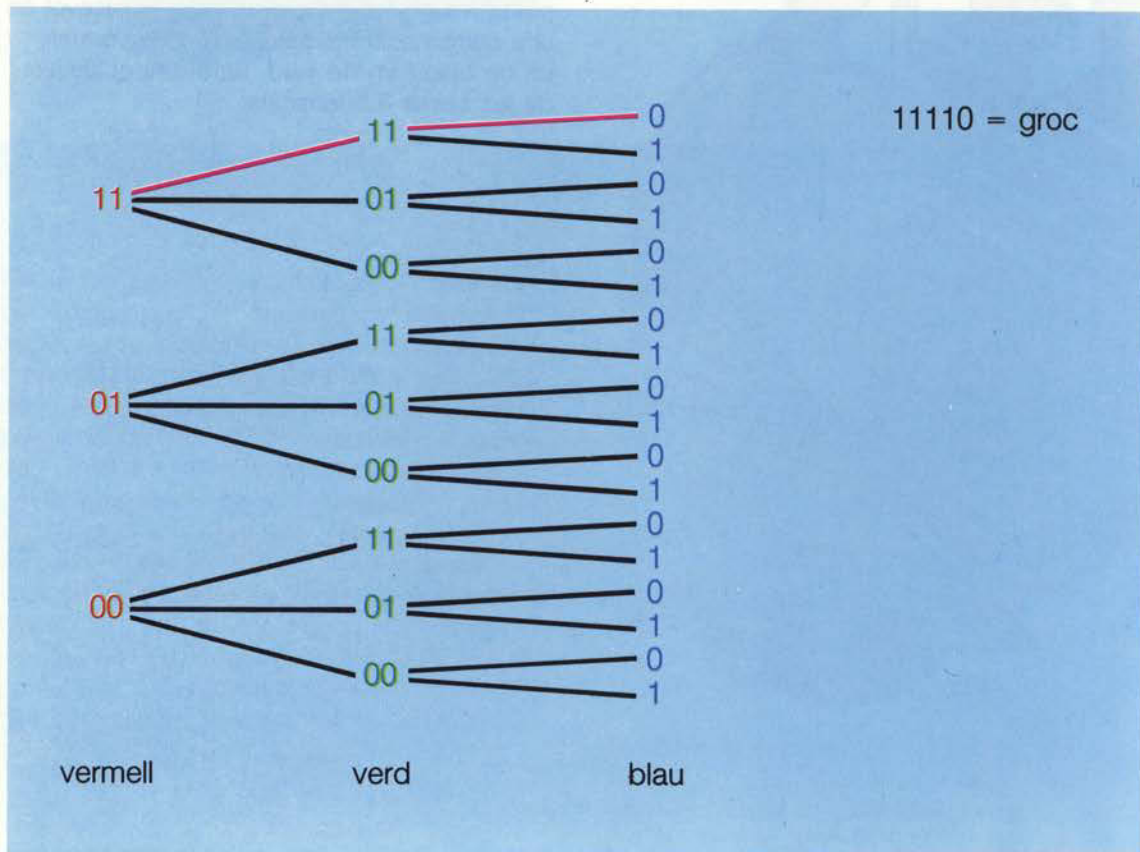
1. Reducció del nombre de colors

Dotar cada punt de la pantalla de 8 possibles intensitats és fabricar 512 possibles colors, ja que un color qualsevol és una combinació de 3 punts: un de vermell, un de blau i un de verd, amb una qualsevol de les seves 8 intensitats.



Se sap que l'ull humà només és capaç de distingir uns 32 colors diferents; aleshores, no cal emmagatzemar-ne 512, perquè tampoc no els podem veure. Allò que es fa en la majoria de micros és considerar tres intensitats per al vermell, tres

per al blau, i dues per al verd. Així, obtenim 18 colors diferents. Aquesta proporció ha estat triada perquè és la que millor s'aproxima al color real. Així, tindriem:



(Necessitem 5 bits per a cada color i en tenim 18 de diferents)
 Amb aquesta reducció, per guardar una imatge, es necessiten :

Amb això, per emmagatzemar una imatge hem passat de necessitar unes 273K a tan sols necessitar-ne unes 152, i l'ull humà gairabé no percebrà la diferència.

$250.000 * 2$ bits per a la imatge vermella	$+ 250.000 * 2$ bits per a la imatge blava	$+ 250.000 * 1$ bits per a la imatge verda	$= 1.250.000$ bits per a la imatge real
$1.250.000 / 8 = 156.250 \text{ bytes.}$			
$156.250 / 1024 = 152 \text{ K}$			

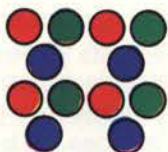
Cal pensar, doncs, a reduir més quantitat de memòria, ja que 152K encara són massa. Continuem sense poder guardar una imatge al micro.

2. Limitació del nombre de pixels.

Fins ara hem fet coincidir els pixels amb les tríades de la pantalla; hem considerat 250.000 pixels, per imatge..., i això ens desbordava.

Si considerem ara els pixels més grans:

1 pixel = 4 tríades = 12 punts



tindrem que cal emmagatzemar només 250.000 tríades / 4 = 62.500 pixels una quarta part del que guardàvem abans. Ara, allò que fa l'ordinador és analitzar cada un d'aquests pixels "grans" i assignar-li un color, que és la mitjana dels colors que hi ha a dins seu.

Per exemple si és



l'ordinador emmagatzema



Observem que, ara, per guardar una imatge real n'emmagatzemarem només una, a la memòria del micro, en comptes de 3, com fèiem abans.

La majoria de micros tenen els que s'anomenen **nivells de resolució**. Aquests

nivells són el nombre de pixels amb què podem dividir la pantalla per tal que "càpiguen" a la memòria.

Nivell 1

Pantalla dividida en $256 \times 176 = 45.056$ pixels, però només podem treballar amb dos colors que acostumen a ésser blanc i negre, o verd i negre.

Nivell 2

Pantalla dividida en $128 \times 96 = 12.288$ pixels. Ara podem treballar amb 4 colors.

Nivell 3

Pantalla dividida en $40 \times 25 = 1.000$ pixels; podem usar tots els 18 colors del micro. Aquestes dades són aproximades, i depenen de la marca del microordinador.

Amb aquestes reduccions, la imatge perd molta realitat: queda amb perfils quadriculats i amb colors estranys.

3. Codificacions especials.

La repetició

Per exemple, si hi ha una colla de pixels seguits de color vermell, en comptes de dir al micro: vermell, vermell..., vermell, 40 cops, és pot fer un programa que li faci entendre: 40 vegades vermell.

Les diferències

Un altre mètode és el que s'anomena la **codificació diferencial**, que consisteix a definir el color de la ratlla de sota per comparació amb el de la ratlla anterior. Es pot fer un programa que ensenyi l'ordinador a entendre:

"igual que la ratlla anterior menys el pixel 30 que el faràs blau".

Hi ha moltes més tècniques de codificació, la major part d'elles molt sotisficades. La cosa que és important és que, aquestes tècniques, redueixen considerablement la quantitat de memòria i no alteren gens la qualitat de la imatge

Gràfics i imatges



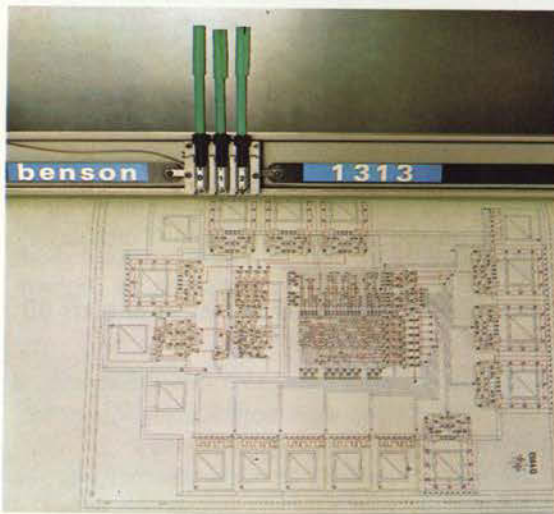
Els informàtics, quan prenen com a base dels seus programes una imatge que ha estat entrada per una taula digitalitzadora, o bé directament per codificació, diuen que estan tractant un **gràfic**; mentre que si treballen amb una imatge entrada per càmera de vídeo, o de fotografia, diuen que tracten una **imatge**.

Els primers són especialistes en **tècniques gràfiques**, mentre que els segons ho són en **tractament d'imatges**

Els mètodes són ben diferents en un i altre camp; per això, són dos camps de recerca. En canvi, passa el contrari amb els resultats. Hi ha imatges que, en veure-les per pantalla, si no s'és un veritable especialista, no se sap distingir si provenen de gràfics o són metamorfosis d'imatges de la realitat.



Equip de tractament d'imatges (U.A.B.)



Plotter de tres plomes



Gràfic realitzat amb micro

Reconstrucció de les imatges



Per una pantalla d'un ordinador es poden veure imatges tan exactes com les imatges de televisió; però passa que, amb aquesta resolució, només es poden veure, no s'hi pot fer res. Per tractar-les, canviar la grandària, el color, girar-les..., cal que siguin les dades d'entrada d'una col·lecció de programes que actuaran sobre elles; per tant, cal tenir-les a la memòria. És en el moment de l'emmagatzematge quan perden precisió: les imatges ens queden fetes a quadrets i amb colors artificials. Hi ha, però, moltes tècniques que permeten, amb un ordinador prou gran, de reconstruir la imatge un cop tractada, donant-li una visió gairabé real. Suposem que tenim una imatge dins la memòria d'un ordinador. Ja sabem que aquesta pot provenir d'una imatge real captada per una càmera de vídeo o de fotografia, d'una imatge imaginada, dibuixada sobre una taula digitalitzadora o d'una imatge sintètica produïda en emmagatzemar directament a la memòria un seguit de zeros i uns.

Aquesta imatge, un cop codificada, és a dir, transformada en un conjunt de zeros i uns, la prendrem com a dada d'uns programes. Aquests, en ser executats donaran com resultat un altre conjunt de zeros i uns, que serà una nova imatge. Una **imatge manipulada**. Ara es tracta de veure-la per pantalla.

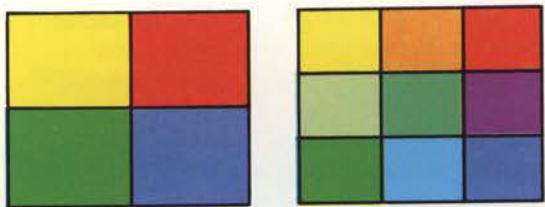
Les ones elèctriques que surten de la memòria cap als tubs envien l'electricitat (electrons en moviment contra la pantalla). Aquesta és una superfície d'un material fotoelèctric que intercepta els electrons (electricitat) i els transforma en fotons (llum).

Ara, allò que veiem per la pantalla és justament allò que hi ha a la memòria: una mala còpia de la realitat; falta, doncs, tot un procés que, sense gastar massa memòria, permeti millorar l'imatge.

La tècnica més usada és la **d'augmentar el nombre de pixels**. Aquest mètode consisteix a col·locar, entre dos pixels seguits, un tercer que tingui per color la mitjana del dos primers.

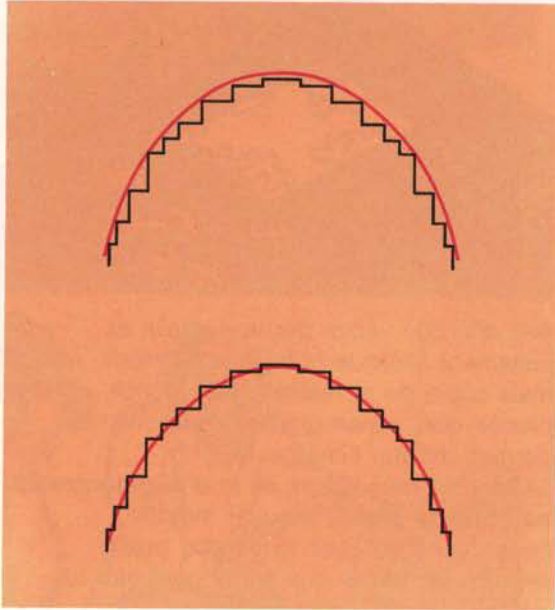
Això es fa al moment de visualitzar la imatge; aquests pixels afegits no són a la memòria però es veuen per pantalla. A la memòria només hi ha un programa que dóna a l'ordinador aquestes ordres:

- disminueix la grandària del pixel
- col·loca un pixel entre cada dos pixels consecutius
- dona-hi el valor mitjà dels dos pixels originals.



Aquest mètode permet de suavitzar els contrastos de color, a l'hora que, en augmentar el nombre de pixels, dóna més sensació de realitat als contorns.

Una altra tècnica és la que consisteix a **allisar les formes**. Hi ha dues maneres de fer-ho: una, és trobant la corba que passa per tots els punts que defineixen el contorn;



28

i l'altre, és buscant la corba que més s'aproxima al traç rectangular que tinguem, encara que no passi pels punts extrems. Aquests dos mètodes, que s'han explicat tan esquemàticament, en veritat són procediments molt complicats, estudiats des de fa temps per enginyers, matemàtics i informàtics. Aquests han estat capaços de crear programes de manera que l'ordinador pugui, ell mateix, arrodonir les figures. A més, hi ha programes, basats tots en tècniques matemàtiques molt sofisticades, que permeten a l'ordinador donar una sensació de profunditat, textura, ombratge, ... Retinguem, doncs, que l'ordinador, prenent una imatge com a dada d'entrada per una gran cadena de programes, la fa passar per tot un seguit de metamorfosis tant de forma com de color, de grandària, de desplaçament... L'ordinador obre en el camp de la imatge una infinitat de possibilitats, avui encara difícils d'imaginar.

Una aplicació en cartografia

Abans hem parlat dels especialistes en tècniques gràfiques. Hom pot pensar que aquestes tècniques encara no corresponen a la nostra societat més pròxima. Doncs, sí! Aquí, a Catalunya (i més concretament, a l'Ajuntament de Barcelona) treballa un grup d'aquests especialistes.

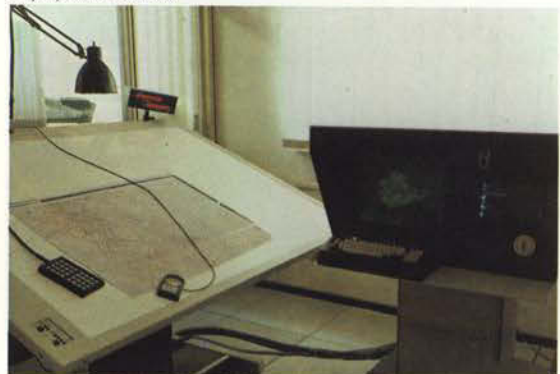
Preparant aquest llibre, hem tingut l'oportunitat de visitar el Centre de Cartografia Automàtica de la Unitat Operativa del Pla de la Ciutat de Barcelona i ens hem adonat del treball d'aquest Centre.

Hi treballen: un cap del servei, quatre analistes del sistema, dos programadors, set operadors i un administratiu.

La composició del material informàtic és:

- un equip CALMA, especialitzat en tractament de gràfics, el qual està en ple rendiment, car és el primer material que va arribar en aquest Centre.
- No obstant ja s'ha quedat petit i és menys versàtil que els altres.

Equip «CALMA»



- un equip INTERGRAPH, suportat per un miniordinador (el que en el primer llibre dèiem ordinador de tipus mitjà) VAX/780 de Digital.

Aquest és l'equip que, en aquests moments, hom posa a punt, mitjançant el seu perfecte coneixement i la introducció de dades (plànols). El perifèrics són: vuit pantalles alfanumèriques, quatre pantalles gràfiques dobles i una de senzilla, a més de les corresponents taules digitalitzadores.

- un equip MATRA, conegut com un restituïdor tridimensional. Aquest, a partir de dues imatges planes, mitjançant uns efectes òptics produïts per un conjunt de lens, permet de veure la imatge tridimensional; és a dir, permet de visualitzar el volum de la imatge.

Una de les característiques del restituïdor tridimensional és que disposa d'una base jeràrquica de dades, per guardar les imatges. Aquest és un equip tot nou, que actualment s'està posant en funcionament i que estarà connectat a l'INTERGRAPH, amb el qual podrà intercanviar informacions.

Les tasques que desenvolupen aquests especialistes en tècniques gràfiques, amb aquest equip, són primordialment les encaminades a gestionar el plànols cartogràfics de Barcelona, substituint les muntanyes de paper per discs i cintes magnètiques manejables pels ordinadors. Una de les tasques que hem vist fer és la d'introduir els plànols, mitjançant una gran taula digitalitzadora i un bolígraf digital, a la memòria de l'INTERGRAPH. Un cop introduïts els pànols, se'n pot disposar a voluntat; per exemple triar una illa de cases d'un barri, fer un "zoom" i apropar-se al pati interior, o a l'inrevés, per veure les illes del voltant; i tot això amb un obrir i tancar d'ulls.

Una característica important d'aquest projecte és que el grafisme utilitzat té una estructuració de capes, fins a 64 nivells; és el que se'n diu una "base de dades gràfica".

Digitalitzador i pantalla gràfica



Centre de Cartografia automatitzada



Digitalitzant



Plànol digitalitzat



Sobre aquesta base cartogràfica introduïda a l'ordinador, s'afegeix:

- grafisme cadastral,
- línies de parcel·lació de la propietat,
- informacions relatives a les edificacions,
- informacions del estat dels vials i trams dels carrers,
- informacions dels serveis, metros, autobusos, llums, aigües...

Una idea de futur és la de lligar totes les feines gràfiques a una base de dades alfanumèrica; de manera que, en visualitzar per pantalla qualsevol parcel·la de Barcelona, es puguin demanar de manera interactiva informacions que l'afecten; com, per exemple, saber qui és el propietari, si està o no en fase d'expropiació, quina superfície té, etc...

No cal insistir en els avantatges que tota aquesta informatització de la cartografia pot reportar als usuaris: tot allò que era guardat en plànols vells i descolorits, (molts són encara pertanyents a la cartografia Martorell), estarà emmagatzemat en ordinadors, la qual cosa comportarà una major facilitat de manteniment, una llarga vida sense enveïllir ni deformar-se, així com permetrà un ràpid accés a qualsevol informació.

Per portar a terme aquest projecte, un dels capdavanters del món, es parla d'uns quatre o cinc anys. I segons ens han dit, una ambició de l'equip humà que hi treballa és la d'arribar a tenir els plànols de Barcelona en imatge tridimensional.

Coratge!

Després de tot el que acabem de dir, és de suposar que ja us han entrat les ganes de començar a dibuixar amb el vostre aparell, de començar a ésser uns "tècnics" en gràfiques. A tots els micros, d'una forma més o menys sofisticada, s'hi poden fer dibuixos per pantalla, donant-los color i moviment. Cada micro té un conjunt d'instruccions que permet de fer tot això. Aquestes instruccions, però, no són les mateixes a tots els aparells, són les que més canvien d'una marca a l'altra. En aquesta part de BASIC s'explicaran les instruccions del ZX Spectrum (Sinclair). Si aquest no és el vostre micro, busqueu el manual si hi ha una instrucció que faci el mateix que us expliquem, encara que no s'escrigui igual.

Abans de començar a parlar d'instruccions, cal recordar que el micro no té gaire memòria. Si es vol treballar amb la quadrícula de pantalla bastant grossa, es podran guardar a la memòria les característiques de cada quadrat: si brilla o no el contingut, el color de fons, i si fa o no pampallugues.

Aquest micro treballa només amb 8 colors; en canvi, el contingut dels quadrets no serà només un color. Podrem guardar-hi dintre una lletra, un nombre, o altres configuracions que prèviament haguem definit.

Si volem que la quadrícula sigui petita, podrem dibuixar amb més precisió línies i figures, però tindrem diverses restriccions en l'ús de colors: un cop dibuixat un quadrat, l'únic que recorda el micro és si hi ha dibuixat o no, però no com ho ha fet.



Instrucció PAPER

Aquesta instrucció, seguida del número de color triat, fa que tot el que continuem escrivint surti amb aquest color de fons. Si fem PAPER 4, tot el que posem a continuació sortirà per pantalla amb fons verd.

Per deixar tota la pantalla d'un color determinat, s'ha d'escriure.

PAPER núm. de color : CLS

Instrucció BORDER

Seguida del número de color triat, pinta d'aquest color el contorn del rectangle de dibuix.

BORDER núm. de color

Per tant, si poseu:

BORDER 2 : PAPER 4 : CLS,

us quedarà el contorn de color vermell i l'interior amb fons verd.

I el color de les lletres i dibuixos?... Vegem, ara, algunes instruccions pròpies del dibuix.

Instrucció INK

La instrucció INK, seguida d'un número de color, fa que tot el que s'escriu o dibuixa surti del color triat.

Si volem fer servir el PAPER i l'INK en un escriptura concreta, els hem de posar entre el PRINT i el que vulguem escriure, separant-ho per punts i comes. Provem-ho:

BORDER 1 : PAPER 6 : CLS
PRINT INK 2 ; PAPER 4 ; "BELLATERRA" : PRINT
INK 5 ; PAPER 1 ; "SANT CUGAT"

Fixem-nos que la segona i la tercera instrucció PAPER només afecten el text que s'ha d'escriure. Això és perquè no hi hem posat el CLS darrera, i perquè són dins d'una sentència d'escriptura.

En el primer cas direm que la pantalla és definida com de baixa resolució, amb 22 files per 32 columnes (704 quadrats), numerades de la 0 a la 21 i de la 0 a la 31, respectivament. El segon cas correspon a una pantalla d'alta resolució, amb 176 files i 256 columnes (45.056 quadrats petits) numerats del 0 al 175 i del 0 al 255; aquests quadrats són allò que abans n'hem dit pixels.

Només cal fer una divisió per veure que cada quadrat de baixa resolució és compost de 64 (8*8) pixels

COLORS

Per indicar al micro el color que es vol fer servir, s'ha de fer amb un número:

0 : Negre	1 : Blau marí	2 : Vermell
3 : Magenta	4 : Verd	5 : Blau cel
6 : Groc	7 : Blanc	

A l'hora de dibuixar amb el micro, no es pot fer servir tota la pantalla del televisor. Normalment, la superfície de què disposem és un rectangle, bastant gran, centrat a la pantalla. per veure'l, farem servir dues instruccions: PAPER i BORDER.

Després d'INK o PAPER, si no són dins un PRINT, tot el que s'hagi d'escriure o dibuixar es farà amb un altre INK o PAPER, respectivament.

El número de color, tant pot ésser una constant com una variable, sempre numèriques.

Instruccions FLASH i BRIGHT

A més de donar un color al fons i a la "tinta" del dibuix, podem fer que les lletres, o paraules que escrivim surtin fent pampallugues o més brillants.

La sensació de pampalluga, el micro l'aconsegueix intercanviant el color de fons amb el de tinta. La instrucció que fa això és **FLASH**. Si volem que un text surti fent aquestes intermitències hem de posar

FLASH 1

Tot allò que el micro hagi d'escriure després de trobar-se amb aquesta instrucció sortirà fent pampallugues. Per desactivar el FLASH 1, hi hem de posar

FLASH 0

i, a partir d'aquest moment, tot el que s'hagi d'escriure sortirà normal.

La instrucció **BRIGHT** funciona de la mateixa manera, però ressaltant el text, fent que brilli. Per activar-lo s'ha de posar **BRIGHT 1**, i, per desactivar-lo, **BRIGHT 0**.

Per veure clar com funcionen aquestes instruccions, farem el següent programa i, després, l'executarem.

```
10 PRINT BRIGHT 1 ; PAPER 4 ; INK 6 ; "BARCELONA"
20 PRINT BRIGHT 0 ; PAPER 4 ; INK 6 ; "BARCELONA"
30 PRINT FLASH 1 ; PAPER 4 ; INK 6 ; "BARCELONA"
40 PRINT FLASH 0 ; PAPER 4 ; INK 6 ; "BARCELONA"
```

A la instrucció 20 i a la 40 no feia falta desactivar el FLASH i el BRIGHT, perquè abans els hem activat dins un PRINT i, per

tant, quan s'acaba la impressió es desactiven sols. Com es pot veure, és bastant diferent el fet de posar PAPER, INK, FLASH i BRIGHT dins o fora d'un PRINT. El programa següent fa la mateixa funció que el d'abans.

```
10 PAPER 4 : INK 6
20 PRINT BRIGHT 1 ; "BARCELONA"
30 PRINT "BARCELONA"
40 PRINT FLASH 1 ; "BARCELONA"
50 PRINT "BARCELONA"
```

Dibuixos de caràcter en caràcter

Anem a parlar de dibuixos en baixa resolució. Cal tenir molt clar que els dibuixos es fan de pixel en pixel, i que els caràcters de què disposem per omplir el pixel són:

- 1) Totes les lletres.
- 2) Tots els números.
- 3) Tots els caràcters especials.
- 4) 16 caràcters gràfics.
- 5) Els definits per nosaltres mateixos.



(The Image Bank)

Els grups 1, 2 i 3 ja els coneixem: són els símbols normals del teclat. El grup 4 és format per una mena de dibuixos, fets amb "taques" blanques i negres. Per veure'ls, premeu la tecla CAPS SHIFT i el 9 alhora; a partir d'aquest moment, si pitgem les tecles de l'1 al 8, s'escriuran a la pantalla vuit d'aquests símbols, i per a veure la resta caldrà prémer la tecla SHIFT alhora que pitgem l'1, el 2, ... fins al 8.

Però no hem de pensar que, per a fer dibuixos s'ha d'anar pitjant tecles tota l'estona. Podem fer-ho dins del programa amb la instrucció PRINT:

```
PRINT "A"
PRINT " "
```

o bé, en lloc d'escriure els caràcters entre cometes, hi podem posar la seva codificació ASCII. Perquè això es correspongui dins un PRINT, hem d'usar la funció CHR\$ (codi). El codi de cada caràcter, en decimal, el teniu al vostre manual, per exemple:

Caràcter	ASCII (decimal)
A	65
T	84
:	59
■	130
■	137

```
PRINT CHR$ (84) escriurà una T
PRINT CHR$ (130) escriurà el símbol gràfic, ■
```

i, si volem un fons de color especial al mateix temps que escrigui en un altre:

```
PRINT PAPER 1; INK 6; CHR$ (137)
```

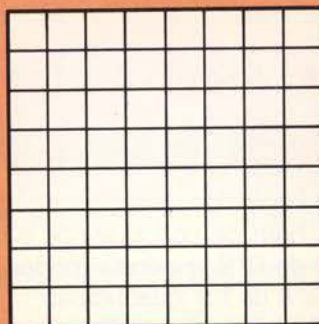
Si es volen veure tots els caràcters i el seu codi per pantalla, cal fer el següent programa

```
5 CLS
10 FOR I=33 TO 143
20 PRINT I, CHR$ (I)
30 NEXT I
```

Anem a parlar del cinquè grup de caràcters.

Instruccions POKE, PEEK

Hi ha un tros de memòria que ens permet definir els caràcters que vulguem. Cal recordar que, per configurar un caràcter per pantalla, usem una matriu de 8 files per 8 columnes de pixels, que es corresponen amb un bit de memòria.



Hi ha 21 matrius d'aquests tipus a la nostra disposició. Si ens hi fixem, cada matriu són vuit files d'un byte cadascuna.

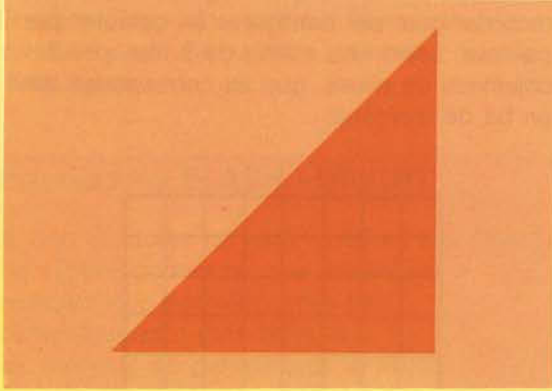
Per adreçar-nos a aquestes matrius s'ha de fer servir el lloc numèric que ocupen dins la memòria, mitjançant la funció:

```
USR "nom de la matriu"
```

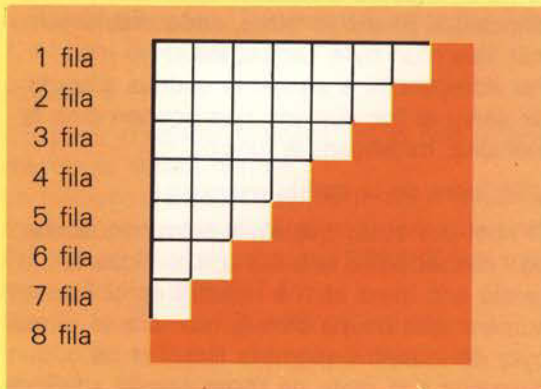
El nom pot ésser qualsevol lletra minúscula de l'alfabet entre la *a* i la *u*, i allò que la funció ens torna com a resultat és el lloc numèric que ocupa dins la memòria el primer byte de la matriu (primera fila). Per tal d'accedir a la resta de bytes només s'han de sumar 1, 2, ..., 7, al resultat de la funció USR. Així, si hem triat la matriu "h" per dibuixar-hi un nou caràcter, hi accedirem:

USR "h"	(primera fila)
USR "h"+1	(segona fila)
USR "h"+2	(tercera fila)
"	"
"	"
"	"
USR "h"+7	(vuitena fila)

Per dir al micro que ens guardi un caràcter de dibuix nou, li hem d'explicar com és. Suposem que volem que tingui la següent forma:



De primer, hem de construir-nos a la llibreta una matriu de 8*8 quadrets (podeu aprofitar els d'un full quadriculat). En segon lloc, anirem omplint els quadradets necessaris perquè ens quedi, més o menys, el caràcter desitjat.



En tercer lloc, cal assignar a cada quadret buit un 0, i als plens, un 1; així, la tercera fila quedarà codificada com:

00000111

I, per finalitzar, hem de guardar el caràcter a la memòria; això es fa byte a byte amb la instrucció POKE, seguida de l'adreça del byte i la codificació; per avisar al micro que la codificació està en binari (0 i 1) hi escriurem BIN al davant.

```
10 POKE USR "h", BIN 000000
20 POKE USR "h"+1, BIN 000000
30 POKE USR "h"+2, BIN 000001
40 POKE USR "h"+3, BIN 000011
50 POKE USR "h"+4, BIN 000111
60 POKE USR "h"+5, BIN 001111
70 POKE USR "h"+6, BIN 011111
80 POKE USR "h"+7, BIN 111111
```

Aquest programa guarda la figura a la matriu "h". Si mirem el manual, veurem que els codis ASCII que van des del número 144 al 164 corresponen als gràfics definits per l'usuari. Com que hem usat la matriu "h", si escrivim al programa anterior

5 CLS

90 PRINT CHR\$(151),

en fer RUN sortirà a la pantalla el nou caràcter gràfic.

L'expressió:

BIN codificació

es pot substituir pel valor decimal del número binari que hem posat per codificar el byte, per exemple:

```
30 POKE USR "h"+2, 7
```

Així com la instrucció POKE serveix per guardar byte a byte un caràcter, donant-li les adreces de cada byte, n'hi ha una altra, la instrucció PEEK, que ens permet mirar allò que hi ha dins els bytes de la memòria. El següent programa escriu a la pantalla el valor dels bytes de la matriu "a"

```
10 FOR I=0 TO 7
20 PRINT PEEK (USR "a" +I)
30 NEXT I
```

Instrucció AT

Tot el que hem dit fins ara ens permet de treballar amb baixa resolució, definir caràcters, escriure'ls a la pantalla, donar-los colors i altres característiques. Encara només sabem escriure caràcters a l'esquerra de tot el quadre de dibuix de la pantalla. Quan vulguem fer un dibuix, potser ens interessarà poder recórrer la pantalla amb facilitat.

Això, ho podem fer amb la sentència de posicionament a baixa resolució:

AT núm. línia, núm. columna

La posem dins d'un PRINT, per indicar al micro a quin lloc ha d'escriure.

El número de línia va entre 0 i 21 i el número de columna entre 0 i 31. Fixem-nos que, així, tractem la pantalla com si fos una matriu (taula de dues dimensions).

Els números de línia i de columna poden ser tant unes constants com unes variables, però sempre numèriques. La posició (0,0) de la pantalla és a dalt i a l'esquerra.

Amb el següent programa col·locarem caràcters a tota la pantalla, en posicions aleatòries entre les línies 2 i 21. A la primera línia sortirà escrit el número de línia que ocupa el caràcter escrit i a la segona el número de columna. Cada cop que escrigui un caràcter, esperarà un cert temps abans de continuar.

La instrucció que provoca aquesta espera és **PAUSE**, seguida d'un número que li indica el temps (50 equival a 1 segon, 100 a 2 segons..., etc.)

Per tallar l'execució, premeu **BREAK**.

```
5 CLS
10 LET X = INT (RND * (19)+2):
   REM CALCUL DEL N. LINIA
20 LET Y = INT (RND * (31)):
   REM CALCUL DEL N. COLUMNA
30 PRINT AT 0,0; "X= "; X ;" "
   REM ESCRIU EL N. LINIA
40 PRINT AT 1,0; "Y= "; Y ;" "
   REM ESCRIU EL N. COLUMNA
50 LET C$ = CHR$ (INT * (RND (112)+32)) :
   REM CALCUL ALEATORI DEL CARACTER
60 PRINT AT X,Y; C$ :
   REM ESCRIURE EL CARACTER
70 PAUSE 100
80 GOTO 10
```

35

Programa de la pilota

Anem a fer un programa que dibuixi un rectangle, amb una pilota que hi corri per dins.
Algorisme:

- 1- **Definir** la pilota.
- 2- **Dibuixar** el rectangle
- 3- **Entrar** els valors de la posició inicial de la pilota X,Y.

- 4- **Dibuixar** la pilota.
- 5- **Moure** la pilota fins que toqui una paret del rectangle.
- 6- **Canviar** la direcció de la pilota.
- 7- **Anar a** 5.

Moure la pilota vol dir canviar el valor de X,Y, esborrar la que hi ha a pantalla i dibuixar-ne una altra a la nova posició X,Y.

Com que les parets del rectangle són les files 0 i 21 i les columnes 0 i 31, per veure si la pilota les toca o no n'hi ha prou amb mirar si la X val 0 o bé 21 o si la Y val 0 o bé 31. Canviar la direcció vol dir això:

- la direcció inicial de la pilota, l'aconsegurem sumant 1 a la X i 1 a la Y. Així, farem que la pilota vagi avall i a la dreta;
- si toca la paret de sota, mourem la pilota

restant 1 a la X, i si toca la de dalt, sumant 1;

- si toca la paret de la dreta, ho farem restant 1 a la Y, i si és la de l'esquerra, sumant 1.

Les variables que s'usaran són:

X,Y: posició de la pilota.

DX,DY: direcció de la pilota.

Dibuixarem la pilota a la matriu de memòria "a."

```
5 PAPER 6 :BORDER 7 : CLS
10 REM CARACTER DE LA PILOTA
20 POKE USR "a", BIN 00011000
30 POKE USR "a"+1,BIN 01111110
40 POKE USR "a"+2,BIN 01111110
50 POKE USR "a"+3,BIN 11111111
60 POKE USR "a"+4,BIN 11111111
70 POKE USR "a"+5,BIN 01111110
80 POKE USR "a"+6,BIN 01111110
90 POKE USR "a"+7,BIN 00011000
100 REM POSICIO INICIAL PILOTA
110 INPUT "FILA ?";X : INPUT "COLUMNA ?";Y
120 REM DIBUIX RECTANGLE
130 INK 1
140 FOR I = 0 TO 21
150 PRINT AT 1,31; CHR$(143)
160 PRINT AT I,0 ; CHR$(143)
170 NEXT I
180 FOR J = 0 TO 31
190 PRINT AT 21,J; CHR$(143)
200 PRINT AT 0,J ; CHR$(143)
210 NEXT J
220 REM PINTAR PILOTA
230 LET DX = 1 : LET DY =1
240 INK 4
250 PRINT AT X,Y; CHR$(144)
260 PAUSE 5
270 REM CONTROLAR LAS PARETS
280 IF X+DX = 0 OR X+DX = 21 THEN LET DX = DX * -1
290 IF Y+DY = 0 OR Y+DY = 31 THEN LET DY = DY * -1
300 REM BORRAR PILOTA I AVANÇAR
310 PRINT AT X,Y; INK 6; CHR$(128)
320 LET X = X+DX : LET Y = Y+DY
330 GOTO 250
```

Dibuixos punt a punt

En aquest apartat, parlarem d'una altra manera de fer dibuixos: amb alta resolució. La pantalla, ara, té 176 files i 256 columnes. En aquest tipus de resolució, només podem dibuixar punts i línies.

En alta resolució, però, hi ha el problema de no disposar de prou memòria per tal d'emmagatzemar el color que té cada pixel. Com veurem més endavant, podem donar el color que vulguem a cada punt; ara bé, per al micro, la pantalla continua quadriculada en baixa resolució. Això vol dir que, dins de cada quadre de 8x8 pixels, només pot haver-hi un color de tinta i un de paper.

Per tant, treballar amb alta resolució vol dir que, respecte a la baixa resolució, hem guanyat quant a facilitat per dibuixar punts i línies, però hem perdut quant a manipulació de colors.

Sí, en alta resolució, pintem amb colors diferents pixels que estiguin un al costat de l'altre, el resultat pot no tenir res a veure amb allò que volíem.

L'única solució que ens queda per dibuixar punts i línies amb diferents colors és vigilar que, si dos pixels tenen diferent color, hi hagi entre ells prou de distància (uns 8 pixels).

La instrucció PAPER, seguida de CLS i BORDER, també servirà aquí per definir el fons i el contorn de tota la pantalla. De la mateixa manera, dins les instruccions de dibuixar també usarem INK per dir al micro el color que volem per a las línies.

Instrucció PLOT

Per dibuixar un punt a la pantalla, hem de donar al micro l'ordre de fer-ho: PLOT, a més de la línia i columna on ha d'estar el punt. Ho farem amb la sentència.

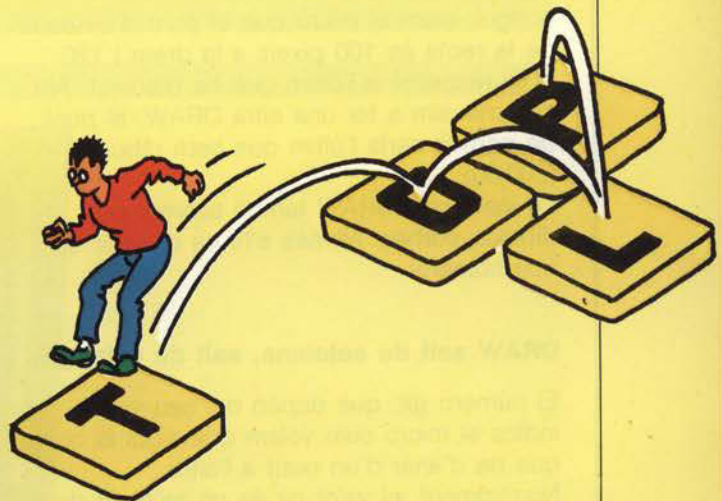
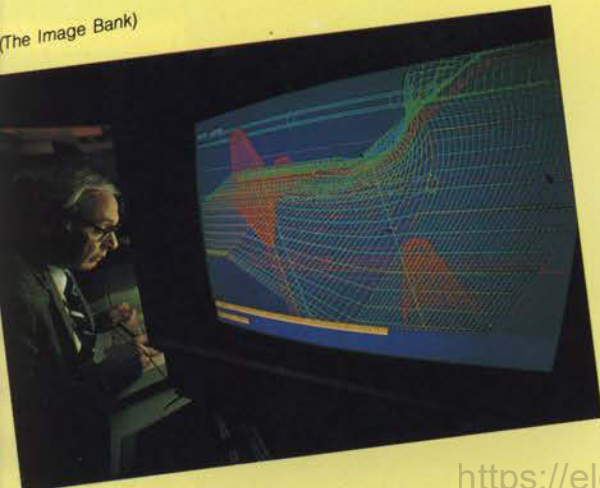
PLOT número de línia, número de columna

En alta resolució, el punt (0,0) és a baix, a l'esquerra.

Per tal de veure millor com funciona, escriurem el programa següent, que ens demanarà el número de línia i el de columna on volem que surti el punt.

```
5 PAPER 7 : CLS
10 BORDER 0
20 INPUT " columna (0-255)? : "; X
30 INPUT " línia (0-175)? : "; Y
40 INPUT " color (0-6)? : -"; Z
50 PLOT INK Z ; X,Y
60 GOTO 20
```

(The Image Bank)



Instrucció DRAW

El BASIC ens dona una facilitat per dibuixar una línia d'una tirada, sense haver de fer-ho pintant els punts, d'un a un.

El micro, en tot moment, recorda quin és el darrer punt que hem dibuixat.

Si encara no hem fet el dibuix, aquest punt serà el (0,0). Per fer una recta des de l'últim punt dibuixat fins a un altre, hi ha una sentència:

DRAW salt de columna, salt de línia

on el "salt de columna" és la diferència entre el número de columna on hi ha el punt al qual ha d'arribar la recta i el número de columna de l'últim punt dibuixat, i això mateix passa amb el "salt de línia".

Així, si volem dibuixar una recta entre el punt (100,150) i el (200,30) la primera cosa que farem serà:

```
10 PLOT 100,150;
```

després, calcularem el salt de columna i de línia:

salt de columna = 200-100 = 100

salt de línia = 30-150 = -120,

i escriurem l'ordre de dibuixar la recta:

```
20 DRAW 100,-120;
```

o sigui, diem al micro que el punt d'arribada de la recta és 100 píxels a la dreta i 120 avall respecte a l'últim que ha dibuixat. Ara, si tornéssim a fer una altra DRAW, el punt de partida seria l'últim que hem dibuixat, el (200,30).

La instrucció DRAW també serveix per dibuixar corbes. Només s'hi ha d'afegir un nou número:

DRAW salt de columna, salt de línia, gir.

El número gir, que depèn del seu valor, indica al micro com volem que sigui la corba que ha d'anar d'un punt a l'altre.

Normalment, el valor gir és un múltiple de la

constant π (és a la tecla on hi ha la M). O sigui, és de la forma:

$\pi * n$,

on n pot ésser:

* $0 < n \leq 1$ per anar del punt inicial al final, ho farà en el sentit invers de les agulles del rellotge, fent poca volta.

* $1 \leq n < 2$ sentit invers a les agulles del rellotge, i molta volta.

* $-2 < n < -1$ sentit de les agulles del rellotge, i molta volta.

* $-1 \leq n < 0$ sentit de les agulles del rellotge, i poca volta.

Perquè això ens quedi més clar, entrem aquest programa a l'ordinador i executem-lo:

```
5 BORDER 1: PAPER 4: INK 0: C
20 LET X = 60:
   LET Y = 80:
   LET T = 110
30 FOR R = 1 TO 2
40 FOR I = -1.5 TO 1.5 STEP
50 PLOT X,Y:
   DRAW T-X,0,I*PI
60 PRINT AT 20,2:
   "GIRO=";I;"*PI"
70 PRINT AT 21,2:
   FLASH 1; PAPER 7;
   "APRETA [ENTER]"
80 PRINT AT 10,17:
   "PUNT INICIAL"
90 PRINT AT 11,17:
   X;";";Y
100 PRINT AT 13,17:
   "PUNT FINAL"
110 PRINT AT 14,17:
   T;";";Y
120 FOR M = 1 TO 4
130 FOR N = 1 TO 10
140 PRINT AT 16+M,16+N:
   PAPER 2; INK 6; CHR$(131)
150 NEXT N: NEXT M
160 IF INKEY$ R<R> CHR$(13)
   THEN GOTO 160
170 CLS
180 NEXT I
190 LET U = T
200 LET T = X
210 LET X = U
220 NEXT R
```


Instrucció CIRCLE

Una altra facilitat per a dibuixar que dóna el BASIC es la instrucció CIRCLE, que permet dibuixar circumferències. Els paràmetres que necessita són:

- el punt central de la circumferència, i
- el radi de la circumferència.

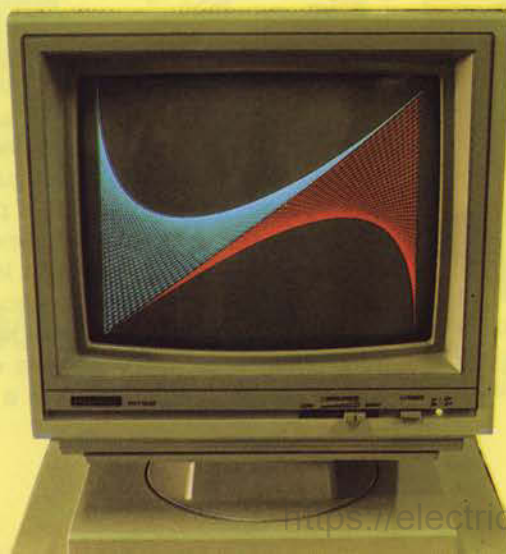
El format de la sentència és:

CIRCLE column, línia, radi.

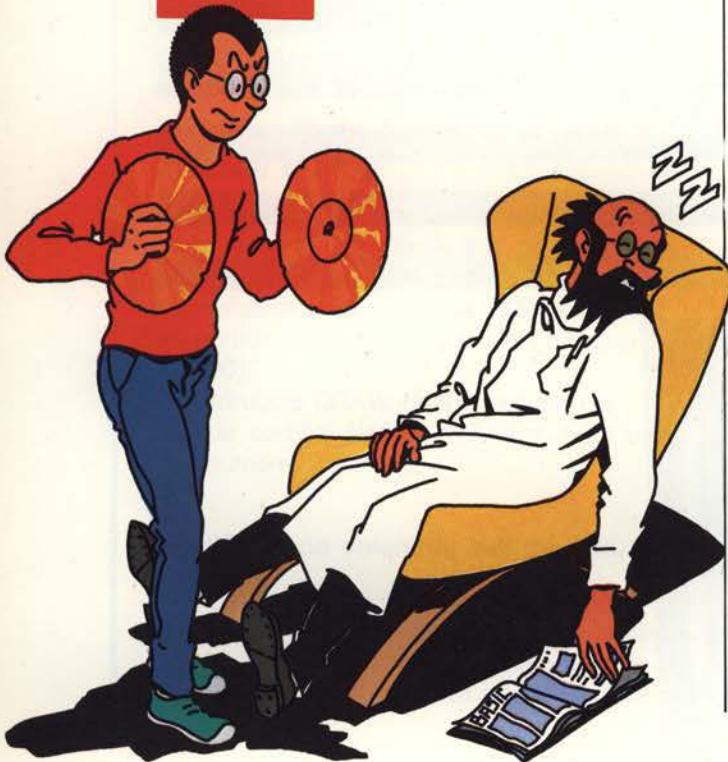
Per tal de veure un joc divertit amb aquesta instrucció, podem escriure el següent programa i fixar-nos com combinant les sentències FOR i CIRCLE, es poden dibuixar cercles fent circumferències consecutives amb el mateix centre (circumcèntriques).

Per acabar l'execució, haurem de prémer la tecla BREAK.

```
5 PAPER 7: CLS: BORDER 0
10 LET Y = INT(RND*140+19)
20 LET X = INT(RND*220+19)
30 LET Z = INT(RND*6)
40 FOR I = 1 TO 15
50 CIRCLE INK Z;X,Y,I
60 NEXT I
65 FOR J = 1 TO 3
70 FOR A = 0 TO 255 STEP 255
80 FOR B = 0 TO 175 STEP 175
90 PLOT A,B: DRAW INK 0; X-A,Y-B:
    BEEP 0.05,40
100 PLOT A,B: DRAW OVER 1; INK 2;
    X-A,Y-B : BEEP 0.10,35
110 NEXT B: NEXT A: NEXT J
120 CLS : PAPER 6: CLS :
    BEEP 0.3,-35: PAPER 2 : CLS:
    BEEP 0.5,-30: PAPER 4 : CLS:
    BEEP 0.5,-40
130 GOTO 5
```



Els sons



Què és

el so

Tots recordem aquella excursió quan, en plena campanya i tots esglaiats, vam sentir la nostra pròpia veu que repetia les paraules que havíem dit feia tan sols uns instants. "És l'eco", ens digué algú. Nosaltres vam insistir a cridar, i l'eco a respondre'ns sempre una mica més tard.

Era evident: la nostra veu acabava de fer un viatge d'anada i tornada.

Però, com viatjava?; quin era el seu vehicle?... En definitiva, per què podem parlar?; i, què és el so?

El so –ho copiem del diccionari de Pompeu Fabra– és la sensació produïda en l'òrgan de l'oïda pel moviment vibratori dels cossos, transmès a l'orella per un medi elàstic, ordinàriament l'aire; especialment la deguda a impulsions que ocorren regularment (oposat a soroll, degut a impulsions irregulars o confuses).

Així, de senzill! El so és la conseqüència d'unes vibracions de l'aire: és l'aire en moviment.

Quan hom parla, expulsa aire dels pulmons i, mitjançant la posició de la llengua i dels llavis, produeix variacions de la pressió de l'aire i provoca un moviment d'expansió i contracció de l'aire que origina les vibracions que arriben a l'orella del qui escolta. És l'òrgan de l'oïda que s'encarrega, llavors, de transformar aquestes vibracions en senyals que el cervell interpretarà com a sons.

Vibracions

i ones

L'aire, a més de ser-nos imprescindible per que ens ofereix l'oxigen que necessita el nostre organisme per a les seves funcions vitals, també ho és perquè puguem xiular, cantar, plorar, riure i, fins i tot, parlar.

El so no només pot viatjar per l'aire. Des de fa temps, se sabia que les balenes es comunicaven entre elles per sons quasi misteriosos; es deia que les balenes "cantaven", i fa ben poc que s'han pogut enregistrar aquests "cants".

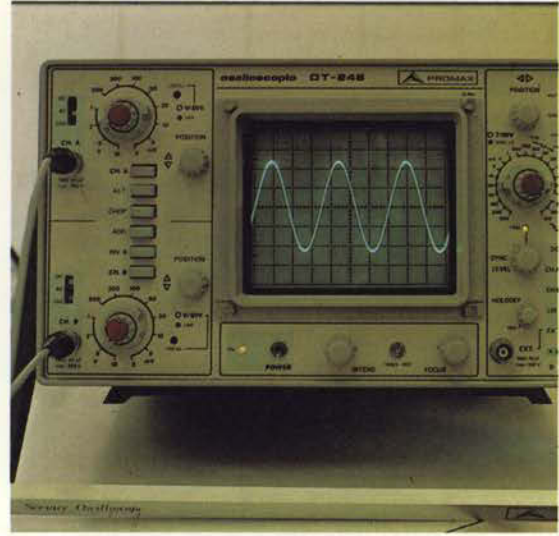
L'home ha volgut dominar aquestes vibracions que produeixen els sons per tal de poder-les canalitzar per altres medis diferents a l'aire; fins i tot somniava de poder-les enregistrar. Avui, l'home no tan sols domina el so, fins al punt que pot enviar-lo per diferents medis (fil telefònic, via satèl·lit, fibra de vidre, etc.) i enregistrar-lo, sinó que, gràcies a l'ordinador, és capaç de crear-ne de nous, fins ara no reproduïts per la natura ni per cap instrument. Una part d'aquests nous sons és la música digital, o allò que trobem al mercat com a música "tecno". Per comprendre el fenomen del so, hem de parlar de les vibracions que el produeixen.

Tothom sap què és una vibració. Per experimentar-la, només cal agafar un cautxú entre les dents, estirar-lo amb una mà i, amb un dit, donar-li un copet al mig. En vibrar, produirà un so. Aquesta mateixa sensació és la que experimenta, encara que no en faci cap cabal, el qui ens ofereix una tonada amb la seva guitarra. En ambdós casos, podem veure vibrar el cautxú, o la corda, malgrat que no podem veure les vibracions que s'han originat a l'aire i que nosaltres percebem com a so.

L'aire va carregat, gairebé saturat, de vibracions. El nostre òrgan de l'oïda pot captar-ne unes quantes, però n'hi ha moltes d'altres que li passen desapercebudes. En general, només podem reconèixer les vibracions; que ni són massa ràpides ni massa lentes, sort que no les podem veure, perquè ens fariem un embolic amb tantes vibracions; es produiria una mena de cortina que ens privaria de veure allò que en realitat ens interessa: les coses que passen al voltant nostre.

Convençuts que el so viatja per l'aire en forma de vibració, allò que ens falta esbrinar és: per què podem distingir tants tipus de sons, i per què hi ha vibracions que produeixen sons que nosaltres no podem percebre, com són ara els infra-sons i els ultra-sons?





Ona sinusoidal en un oscil·loscop

En realitat, podem considerar les vibracions com un seguit de crestes i valls de diferents altures, i més o menys comprimides.

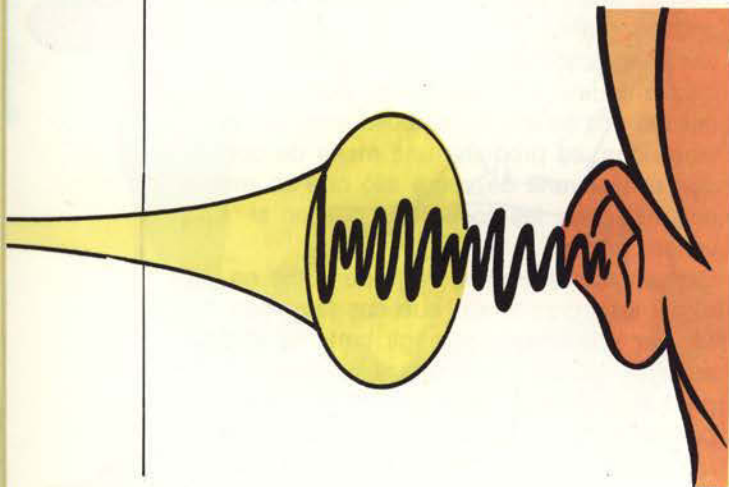
Un símil molt adient és el que s'esdevé quan algú llença una pedra en un llac i l'aigua hi és en repòs. Es formen un seguit d'ones que, partint del punt on cau la pedra, es van escampant al llarg i ample del llac. Aquesta ona és un seguit de pujades i baixades de la superfície de l'aigua que, en anar-se repetint, cada cop disminueixen d'altura i de compressió i abasten més superfície.

L'altura de les crestes i les valls que s'hi han format, així com que estiguin més o menys comprimides, depèn de la força amb què ha estat tirada la pedra i del seu pes.

Si llancem una altra pedra, es formaran també unes altres ones que es barrejaran amb les primeres. S'hi forma un suau marejol d'ones entrecruades. Fixant-nos-hi bé, però, podrem arribar a distingir si són produïdes per un o altre llançament.

Això mateix passa amb les vibracions sonores, que també en direm ones. Quan parlem, parteixen de nosaltres un seguit de crestes i valls fins arribar a l'orella del qui ens escolta. Si hi ha més gent que parla, mentre no siguem molts, l'òrgan de l'oïda del qui escolta és capaç de distingir allò que diu cadascú. És a dir, les ones es barregen, però no es confonen.

Als afeccionats a les matemàtiques, els hem de dir de aquest seguit de crestes i valls que fa l'ona (una cresta i una vall seguides formen el que se'n diu un CICLE) és representable mitjançant una funció del temps. Aquesta funció té una semblança amb la funció trigonomètrica coneguda amb el nom de SINUS, i la corba resultant de la seva representació es coneix amb el nom de SINUSOIDE



Que podem escoltar?

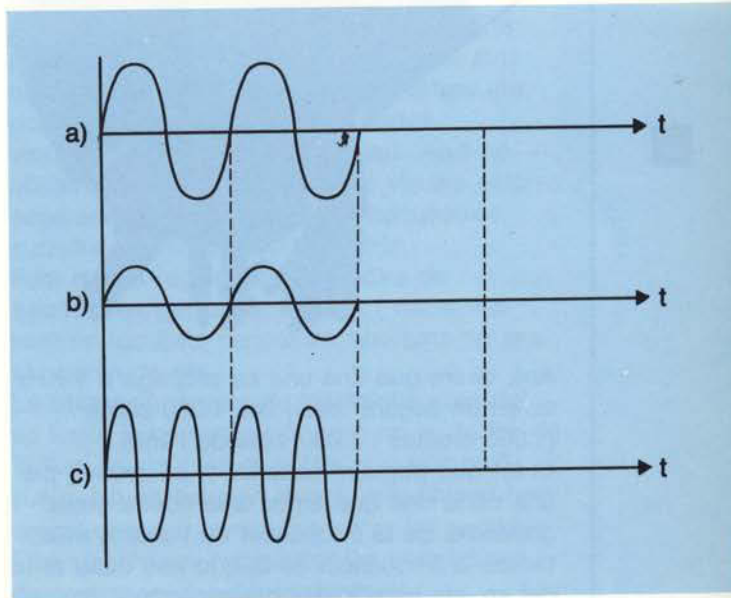


Si prenem un tros de cartó i el movem per fer-nos aire, notarem una frescor que ens arriba a la cara, i sentirem el fregament del cartó amb l'aire; però no hi sentirem cap més soroll especial, llevat que moguéssim el cartó molt ràpidament (com a mínim 15 vegades per segon).

Si estem en silenci y pel nostre voltant es presenta una mosca impertinent, podrem sentir un so finíssim, un brunzit, produït pel bategar de les seves ales ja que les mou molt més ràpidament que nosaltres movem el cartó.

Aquest moviment, més o menys accelerat, és el causant del so i de la seva tonada.

Direm **FREQÜÈNCIA** al nombre de cicles produïts per l'ona en un interval fix de temps.



Les ones a) i b) tenen la mateixa freqüència mentre que l'ona c) té doble freqüència que les anteriors.

La freqüència d'una ona es mesura en Hertz, que abreujarem per Hz: ben segur que aquesta notació és en els vostres aparells de ràdio. A més, s'hi troben les sigles KHz que equival a 1.000 Hz i que es llegeix quilo-cicle o quilo-hertz, o MHz que són 1.000.000 de Hz i que es llegeix mega-cicle o mega-hertz.



Així, direm que una ona es propaga a 1 KHz si, en un segon, tenen lloc 1.000 cicles (1.000 crestes i 1.000 valls de l'ona).

El fet que puguem escoltar el so produït per una certa ona que arriba a la nostra orella dependrà de la freqüència de l'ona. A més, també la freqüència és la que ens dona el to del so: els sons greus corresponen a ones de baixa freqüència, i els aguts, a ones de freqüència més alta.

El conjunt de freqüències que l'home és capaç de reconèixer s'anomena ESPECTRE AUDIBLE. És conegut que l'espectre audible de l'home és comprès entre les freqüències de 15 Hz fins a 16 KHz. Com a cas particular,ensem que la zona de tonades musicals fonamentals (les produïdes per pianos, orgues, etc..) és compresa entre les freqüències de 30 Hz i 8 KHz.

És sorprenent que tant els grans concerts de les orquestres simfòniques, com també la música i la veu dels cantants de rock, no siguin res més que un conjunt d'ones entremesclades, que es propaguen a diferent freqüència.

Per sota dels 15 Hz, l'home només és capaç de percebre vibracions pel tacte: és la zona dels infra-sons; mentre que, per sobre dels 16 KHz, només les pot percebre mitjançant sensors especials: aquesta és la zona dels ultra-sons.

En canvi, hi ha animals que són capaços d'escoltar vibracions sonores per sobre de



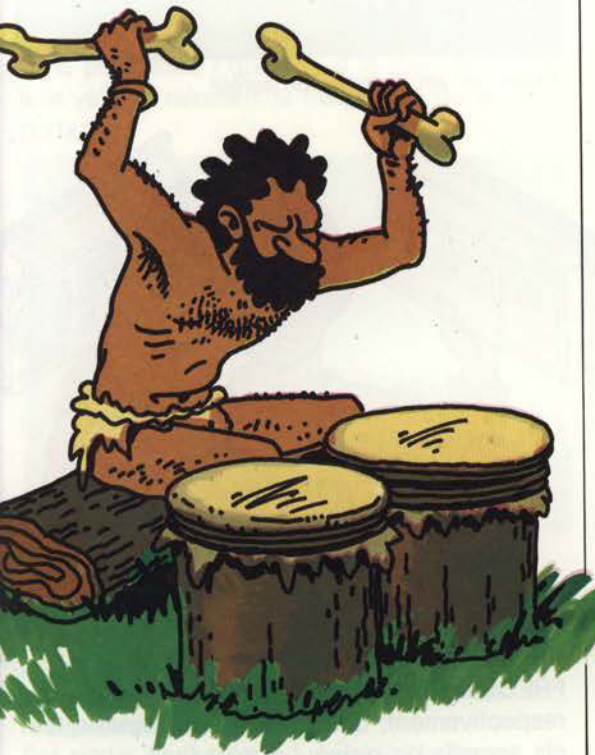
les freqüències dels 16 KHz: en aquest cas, es parla de "sons silenciosos". Les ones supersòniques properes als 50 KHz s'utilitzen en equips de sondeig i prospeccions petrolíferes, així com en l'experimentació bacteriològica i en medicina. Per altra banda, les línies telefòniques treballen amb freqüències entre els 300 Hz i els 3.400 Hz, encara que, en aquest cas, les ones no es transmeten per l'aire sinó per un fil.

L'home no n'ha tingut prou amb saber què és físicament el so i com es propaga, sinó que, amb l'aparició de l'electricitat, de l'electrònica i, després, de la informàtica, ha aconseguit de "domar-lo": el pot transmetre a llarga distància, bé sigui per l'aire (ràdio, TV) bé sigui per fil (telèfon): l'ha pogut emmagatzemar (discs, cintes magnètiques); i ha aconseguit de manipular-lo (ordinadors).

Transmissió:

fem viatjar

el so



Des del seu origen, l'home s'ha valgut de les seves aptituds vocals i dels gestos per comunicar-se amb els qui l'envolten. Això no obstant, a mesura que la societat avançava, anava sorgint la necessitat de comunicar-se a major distància. Els primers intents de comunicació a llarga distància foren els tam-tams, les botzines i els megàfons, però el seu abast era encara molt limitat. Per augmentar la distància de comunicació s'han usat mitjans visuals: com el fum, les torxes i l'oneig de banderes.

El lent i fatigós camí que porta l'home cap als actuals enginyers electrònics de comunicació va passar per l'estudi i el descobriment de les possibilitats de diferents freqüències que hi havia en l'espectre de comunicació, tant del so com de la llum. En particular, l'espectre audible és una petita part de l'espectre de freqüències que avui es coneixen. D'altra banda, els nostres ulls poden convertir certes ones, certes vibracions, en sensacions visuals. Això no obstant, les ones sonores i les visuals estan separades per un interval de freqüències superior als 800.000.000 de MHz.

Fora de les regions de l'espectre de freqüències del tacte, audible i visual, les nostres facultats sensorials inherents no ens serveixen de res.

La utilització directa de les freqüències del so limita la comunicació, però no la de les de la llum (podem veure estels que són molt lluny). No és estrany, doncs, que l'home hagi cercat ones de freqüència entre les audibles i les visuals per augmentar les possibilitats de comunicació a llarga distància.

Avui, ja s'ha aconseguit de dissenyar nombrosos sistemes electrònics de comunicació que, utilitzant diverses porcions de l'espectre, permeten de transmetre i rebre senyals a, i des de, distàncies virtualment il·limitades.

Per poder parlar, però, de la transmissió del so a llarga distància hem de partir del principi que l'aire està plagat de camps magnètics (pensem en el magnetisme terrestre, que permet a l'agulla de la brúixola



d'estar sempre orientada de Nord a Sud, encara que n'estiguem ben lluny) i que es poden crear camps electromagnètics artificialment, els quals es poden propagar molt lluny.

La transmissió del so es farà mitjançant un d'aquests camps electromagnètics, al qual se li aplicarà una vibració molt regular i amb una freqüència molt gran; és a dir, es crea un seguit de crestes i valls de la mateixa altura, que es repeteixen moltes, moltíssimes vegades, en un segon. Ara podem utilitzar aquesta ona que hem creat, que s'en diu PORTADORA o bé de RÀDIO, per transmetre ones sonores a través de l'aire.

En realitat, les ones de ràdio utilitzades per a la transmissió i per a la recepció dels senyals electrònics estan compostes d'un camp electrostàtic i d'un altre camp electromagnètic.

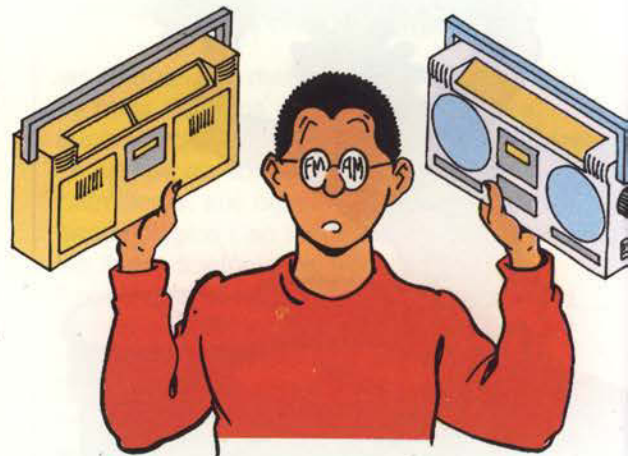
Podrem transmetre les ones sonores després d'haver-les transformades en vibracions d'un senyal elèctric o d'un camp electromagnètic. Tant en un cas com en l'altre, és l'electricitat la que transforma les paraules i la música en "senyals".



Radiotélescopi (Ontario-Canadà)

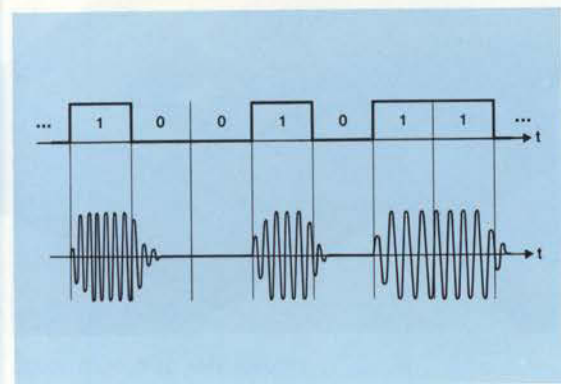


Radar pel seguiment d'un satèl·lit



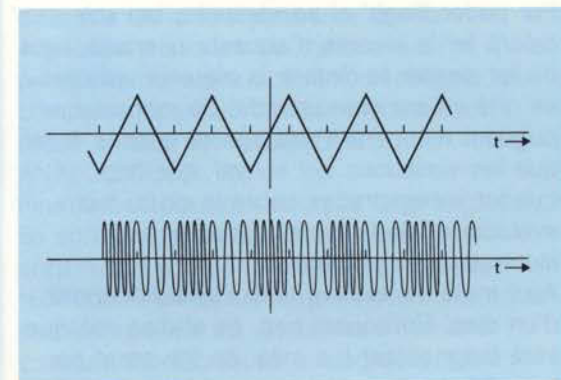
Bàsicament, s'usen dos procediments de transmissió: el procediment de MODULACIÓ D'AMPLADA i el de MODULACIÓ DE FREQUÈNCIA (les sigles AM i FM, respectivament, que trobem en gairebé tots els aparells de ràdio). La modulació

d'amplada (AM) consisteix en el fet que l'ona sonora forma una espècie de recobriment de l'ona portadora, fent variar, així, l'altura de les crestes i valls de l'ona portadora.



Modulació d'amplada

La modulació de freqüència (FM) consisteix a fer variar la freqüència de l'ona portadora, segons siguin les crestes i valls de la sonora; però, en aquest cas, sense variar el valor de l'amplada: quan l'ona sonora és a la fase de pujar la cresta, s'accelera la freqüència de l'ona portadora; i quan es a la fase de baixar a la vall, es ralenteix la freqüència de la portadora.



Modulació de freqüència

En resum, per transmetre les vibracions del so es barreja l'ona sonora amb una altra ona, la portadora, que és la que permetrà de fer viatjar el so a llarga distància i, després, recuperar-lo.

Per poder parlar de recuperació de l'ona

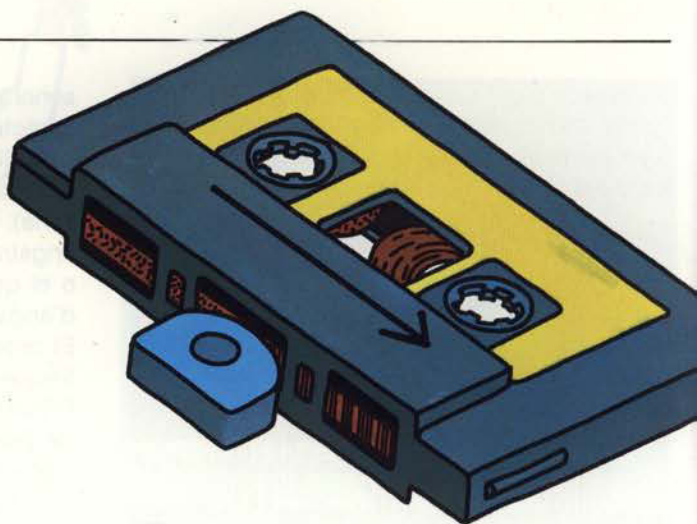
sonora, hem de parlar d'una tercera característica de les ones. Aquesta és la LONGITUD D'ONA, que és la distància que recorré l'ona en un cicle (cresta i vall de l'ona), i és mesurada en ÀNGSTROMS. Un àngstrom equival a 1/10.000.000.000 metres, o el que és el mateix, deu mil milions d'àngstroms són un metre.

El producte de la longitud d'ona per la freqüència és la VELOCITAT DE PROPAGACIÓ del so. Se sap que la velocitat de propagació del so en l'aire és d'uns 331 metres per segon, o, cosa que és el mateix, uns 1.200 km per hora, i que la velocitat de propagació de la llum és d'uns 300.000 km per segon. La velocitat de propagació de les ones electromagnètiques de ràdio coincideix amb la velocitat de la llum.

Per a la recuperació de l'ona sonora de l'ona transmesa, farem servir la longitud d'ona. Cal que el receptor reguli la longitud d'ona que vol captar. Aquesta ha de coincidir amb la longitud de l'ona portadora. Així, el receptor podrà triar aquesta ona d'entre totes les que li arriben, i detectar-ne la forma, que després transformarà en una vibració elèctrica que enviarà als altaveus de l'aparell de ràdio o televisió; o, senzillament, les enregistrarà. En el cas d'una modulació de freqüència, el receptor regularà, a més de la longitud d'ona, l'amplada de la portadora, que és fixa, i interpretarà les variacions de freqüències de la portadora per poder trobar una ona sonora com més semblant possible a la transmesa. Per assolir-ho, només cal un senzill aparell elèctric que faci augmentar la tensió, a mesura que augmenta la freqüència, i que disminueixi la tensió, a mesura que la freqüència ralenteixi.

D'aquesta manera, ja que podem variar la intensitat d'un corrent elèctric, durant la transmissió es pot manipular el so de partida gairebé com es vulgui: amplificar-lo, modificar els tons, filtrar els aguts i els greus...; és a dir, aquelles manipulacions que hom veu fer a les emissores de ràdio i/o televisió, amb aquells taulells tots plens de botons i llums que s'anomenen "taules de so".

Guardem els sons



48

Per a l'emmagatzematge o enregistrament del so ens servirem de l'electromagnetisme: caldrà transformar les vibracions del corrent elèctric en una seqüència de variacions que creï un camp magnètic.

Perquè ens quedi una mica clar això de crear un camp magnètic, podem fer el següent experiment:

Agafar un paper una mica fort i escampar-hi llimadures de ferro. En passar un imant per sota del paper veurem que les llimadures es van col·locant formant una mena de 8. Si anem desplaçant l'imant, les llimadures també s'aniran desplaçant, tot guardant la mateixa forma. La zona d'influència de l'imant rep el nom de "camp magnètic".

Per enregistrar les vibracions del so, usem una cinta o banda impregnada de partícules microscòpiques d'un cert mineral altament magnetitzable. De bon començament, aquestes partícules estan en un ordre qualsevol sobre la cinta; però, quan passen pel camp magnètic que crea el capçal d'enregistrament, s'orienten al mateix temps

que queden impregnades d'un magnetisme permanent, fins que no tornen a passar per un altre camp magnètic. A les partícules que desfilen pel cap d'enregistrament en el moment en que hi ha camp electromagnètic, no els passa res: queden amb el mateix desordre d'origen.

Sobre la banda magnètica enregistrada, les evolucions de les freqüències del pas de partícules magnetitzades es correspon amb les fluctuacions de l'ona sonora.

Per poder "llegir" el so, després, tan sols caldrà fer la inversa d'aquesta operació: hem de fer passar la cinta a la mateixa velocitat en què va ser enregistrada, de manera que puguem recuperar l'ona sonora gràcies al fet que les variacions del senyal, que han quedat enregistrades sobre la cinta, faran evolucionar de la mateixa manera un electroimant d'un altaveu.

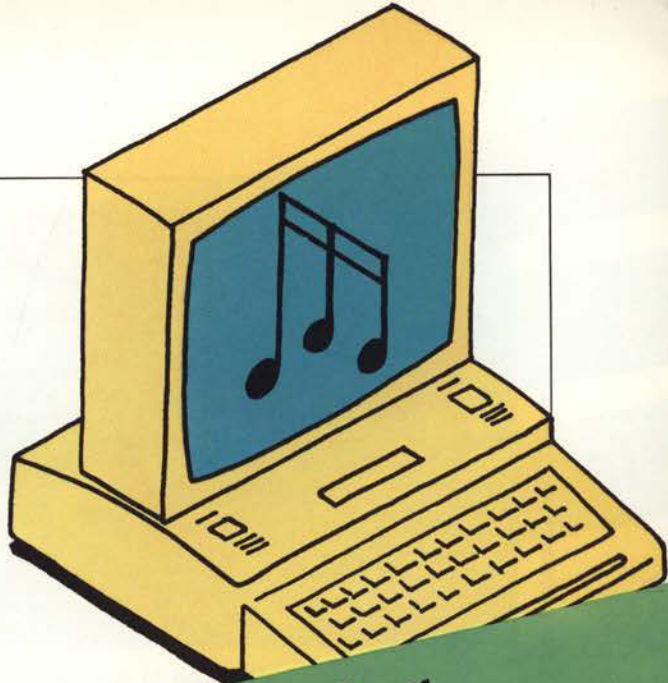
Això mateix s'esdevé amb l'enregistrament d'un disc. En aquest cas, és el disc allò que està magnetitzat i, a més, és travessat per un solc que forma una espiral a l'entorn del centre. Les partícules magnetitzades estan situades al fons d'aquests solcs. El tros de diamant de l'agulla del capçal del tocadiscs serà guiat per aquests solcs i, mentre el disc gira a la mateixa velocitat en que va ésser gravat, vibrarà. Només cal, doncs, tenir l'ajut d'un electroimant per transformar aquestes



Solcs d'un disc «Microsurc»

vibracions en ones electromagnètiques, per enviar-les a un altaveu que reproduirà el so. Com en el cas de la transmissió, un cop dominem la tècnica de l'enregistrament podem manipular aquests sons emmagatzemats, com és ara el cas de "l'estéreo" que permet de reproduir en l'altaveu de l'esquerra els sons enregistrats a unes freqüències i, a la dreta, els enregistrats en altres freqüències, o bé que ens permet de distingir la proximitat o la distància de la font emissora del so. Aquestes tècniques de transmissió i enregistrament que acabem d'esbossar, així com la recuperació del so, en cada cas, reben el nom de **TÈCNiques ANALÒGIQUES**.

Fins fa ben poc, els esforços dels enginyers en so i telecomunicacions estaven encaminats a millorar cada cop més les tècniques i els aparells que intervenen en la manipulació del so. Cada dia s'han anat millorant les tècniques analògiques, però sempre romania algun bri de soroll involuntari que minava la qualitat de l'audició. L'aparició de l'ordinador, i la seva capacitat de càlcul, va fer pensar en la possibilitat de guardar els sons, no en forma analògica sinó en forma digital, numèrica. És a dir, representar els sons mitjançant seqüències de 0 i 1.



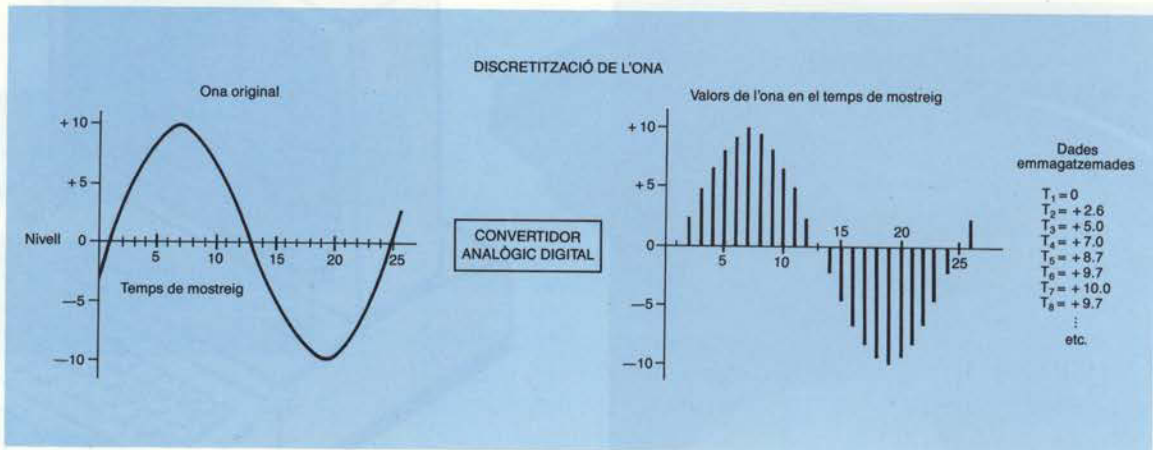
L'ordinador i els sons

Insistim en el fet que l'aparició de l'ordinador ha revolucionat la manera de viure i d'actuar dels homes.

La il·lusió de poder emmagatzemar els sons ja havia estat assolida, però de manera analògica. El repte era, ara, de poder emmagatzemar els sons dins de la memòria d'un ordinador, la qual cosa significaria una nova possibilitat de manipulació per poder crear nous sons.

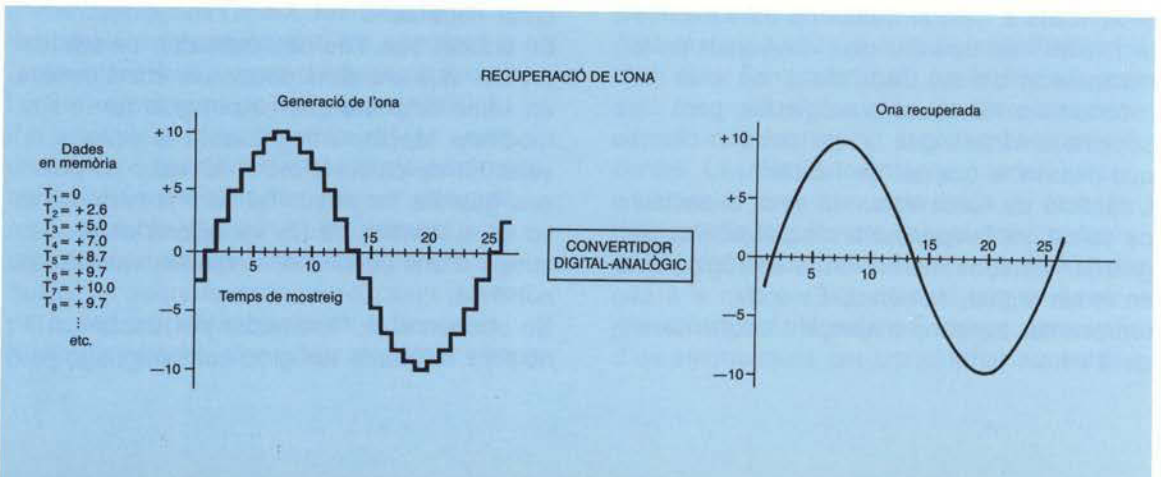
En aquest cas, l'ús de l'ordinador parteix del principi que els sons poden ser transformats en xifres binàries, que ell pot guardar i modificar ràpidament gràcies a la seva velocitat de càlcul. Però l'ordinador no pot pas guardar tot el so. Tal com ja hem dit, el so és el resultat de les variacions en el temps d'una certa ona, i aquesta variació és contínua.

No obstant això, l'ordinador pot tractar un nombre de punts tan gran com vulguem, però



no pot tractar-ne infinits. En conseqüència, si volem utilitzar l'ordinador per a tractar el so, ens haurem de conformar a fer-ho sobre una "mostra" de l'ona. És a dir, agafarem l'ona i la considerarem només sobre un número finit de petits intervals de temps. A mesura que els intervals siguin més petits, la reproducció final del so serà més fiable i real. De la mateixa manera que, en televisió, per reproduir el moviment de les imatges ens fan empassar unes 25 pantalles diferents per segon, és a dir, unes 25 mostres de la realitat (que ens donen la sensació de continuïtat), per poder guardar el so a l'ordinador necessitarem també fer un gran nombre de mostres per segon, si volem una reproducció quasi contínua i quasi real. En particular, en el cas del disc compacte, es poden arribar a fer de 35.000 a 40.000 mostres per segon, seguint el principi de la reproducció del so, que demana que el

nombre de mostres per segon d'una ona superi el doble de la freqüència de la nota més alta que es vol transmetre. I hem dit que hom podia percebre sons de freqüències fins a 16 KHz. Un cop fetes les mostres, que són anomenades IMPULSOS, ens caldrà codificar-les; això és, cada impuls tindrà assignada una seqüència de 0 i 1 que, en definitiva, és el llenguatge que entén l'ordinador. La codificació dels impulsos es fa per un CONVERTIDOR ANALÒGIC-DIGITAL, que consisteix en un aparell electrònic que transforma un impuls d'una ona determinada en una seqüència de 8 bits (en un byte). En aquest cas, podem representar $2^8 = 256$ impulsos diferents. Si la codificació és feta en 16 bits, com és ara el cas dels discs compactes, podem representar fins a $2^{16} = 65.536$ impulsos diferents.



Ordinadors i sintetitzadors: música digital

Es comprensible, doncs que usant 16 bits per impuls, donarà una qualitat sonora molt millor que usant-hi 8 bits.

Un cop s'ha fet la codificació, l'ordinador disposa d'una gran quantitat de xifres sobre les quals pot treballar, segons el programa que s'hagi previst. Podrà transformar els sons i, segons les instruccions del programa, podrà corregir possibles errors. Quan una informació sobre un impuls és incompleta o incorrecta, l'ordinador té a la memòria prou instruccions per poder reconèixer a quin impuls correspon aquesta informació defectuosa, i corregir-la automàticament. I és per això que la reproducció pot semblar perfecte en digital, mentre que en analògic sempre queden algunes restes de sorolls, encara que, de vegades, segons la qualitat de l'aparell, gairebé passen desapercebuts. Si no, pregunteu-ho als qui ja gaudeixen d'un reproductor de discs compactes!

Malauradament però, hom només pot entendre els sons analògics, els sons que són un seguit de vibracions en el temps, i no pas una seqüència de 0 i 1. Si no, proveu de fer que el vostre enregistrator de cassette us reproduxeixi els sons que produeix el programa NINOT que havieu enregistrat, en acabar el primer llibret, o qualsevol altre programa. De sonar, sonarà...; però, és música?...

A l'hora de voler reproduir els sons guardats a l'ordinador necessitarem un CONVERTIDOR DIGITAL-ANALÒGIC; és a dir, un circuit electrònic que, quan rebí una seqüència de 0 i 1, ens la converteixi en un impuls.



Disc compacte

Ja sabem que el so és un conjunt de vibracions de l'aire que afecten el nostre sentit auditiu; però, segons les característiques d'aquestes vibracions, podem distingir els diferents tipus de sons. Les tres característiques principals d'un so són:

- TO: és la freqüència del so. La classificació dels tons és allò que coneixem per escala musical (DO, RE, MI, FA, SOL, LA, SI).
- VOLUM: ens mesura la intensitat del so.
- TIMBRE: és la qualitat d'un so que permet distingir-lo d'altres que tenen idèntics volum i tonada.

En particular, el timbre d'un so vindrà determinat per tot un seguit de paràmetres que podran assolir diferents valors. Alguns d'aquests són: **"Sustain", "Vibrato", "Ressonància", "Attack", "Release", "Delay", "Eco", "Portamento"**, etc. Això ens permet de diferenciar els diversos instruments d'una orquestra, tot i que estiguin tocant la mateixa melodia. La possibilitat de tractament del so mitjançant l'ordinador ha revolucionat, després dels anys 1960, la concepció i la realització musical. A partir d'aquesta dècada, dos mons s'han anat apropant: el de la música i el de la informàtica. L'electrònica és el mitjà que ha fet possible aquesta convergència, i el SINTETITZADOR és l'instrument que ha donat a l'home el control de la producció i manipulació del so.



El britànic ROBERT MOOGE n'és el pioner. La seva aportació és la de permetre controlar mitjançant tensions elèctriques, en comptes d'operacions manuals, els dispositius electrònics (oscil·ladors, filtres, amplificadors...) per a la generació i manipulació del so.

Si podem aconseguir un circuit electrònic que ens generi un so bàsic (per exemple un xiulet) tindrem l'aparell que s'anomena un OSCIL·LADOR. A partir d'aquest so, i a fi d'aconseguir les diferents notes de l'escala musical, utilitzarem uns altres circuits electrònics que ens en varien la freqüència. Així mateix, podem modificar els valors de tots els altres paràmetres per tal d'aconseguir imitar qualsevol instrument. Tot aquest conjunt de circuits disposats de forma convenient constitueixen l'aparell que anomenem un SINTETITZADOR. El que es fa servir per fer variar la freqüència, en un sintetitzador; és, habitualment, un teclat similar al d'un piano. I tot el conjunt de botons, interruptors i reguladors que porta ens possibiliten de variar el timbre del so bàsic. El sintetitzador és l'instrument electrònic per excel·lència: permet de crear tots els sons de la música convencional, i també crear-ne de

nous; però, en canvi, és difícil d'usar. Per emprar-lo bé, s'han de tenir coneixements d'acústica, de les propietats fonamentals del so, de les formes d'ones, així com del funcionament dels diferents mòduls que l'integren (oscil·ladors, filtres, amplificadors, moduladors, etc.).

Encara que sigui un aparell força complicat, fem-ne un petit esbós, per tractar de comprendre què pot fer, i també de comprendre el paper que poden tenir els microordinadors en el món musical. El sintetitzador és un aparell que, en essència, pot memoritzar seqüències de tensions elèctriques (cas ANALÒGIC) o seqüències de zeros i uns (cas DIGITAL), el qual serà capaç de repetir tot allò que es vulgui, segons la seqüència programada, controlat per un SEQÜENCIADOR que regula la velocitat de la seqüència.

En el cas analògic, disposarem d'una unitat generadora de sons: l'oscil·lador. Si a una petita llengua de metall se li aplica una tensió o un voltatge elèctric a través d'un modulador, el metall vibra, i crea la forma més senzilla d'ona sinusoidal. La freqüència de la vibració resultant depèn del voltatge aplicat i de la densitat del metall. El control

d'aquest voltatge ha estat el principal mètode de producció de música sintetitzada. En el cas digital, disposarem d'una memòria ROM (en alguns casos, d'una memòria EPROM = **Erasable Programable Read Only Memory**) que guardarà la codificació binària dels sons produïts per l'instrument musical de què es tracti.

En ambdós casos, el so de l'instrument és captat directament per connexió (guitarres i orgues) o bé mitjançant un micròfon. Si la memorització és analògica, l'ona sonora corresponent es transformarà en un senyal elèctric que, en el moment de la reproducció, serà el que accionarà l'oscil·lador; mentre que, quan la memorització és digital, l'ona sonora és discretitzada i codificada per guardar-la a la ROM (o, en el seu cas, l'EPROM). En aquest darrer cas,

necessitarem un convertidor DIGITAL-ANALÒGIC, el dispositiu electrònic que, a partir d'una representació digital, un conjunt de zeros i uns, ens genera una sèrie d'impulsos elèctrics que podran atacar un amplificador i produir el so corresponent. Les operacions de emmagatzematge de sons són realitzades per molts instruments. Així, queda construït un sintetitzador. Aquest procés és realitzat per les empreses constructores.

Allò que nosaltres podem fer és emprar-lo. La primera cosa que cal fer és introduir una melodia a través d'un teclat. Després se selecciona l'instrument. Si el sintetitzador no rep cap més ordre, reproduirà la melodia en temps real: és a dir, tal com l'hem introduïda. No obstant això, mitjançant el seqüenciador es pot anar variant la velocitat de les seqüències a produir per la melodia d'entrada, però no fer variacions sobre les assignacions de voltatge dels oscil·ladors, en el cas analògic, ni dels continguts de la ROM o EPROM.

Es per poder fer precisament això que podem utilitzar el microordinador. L'aparell que treballa internament en forma digital, i que mitjançant un convertidor digital-analògic ens dona una resposta en

forma de senyal elèctric que representa un so, és un SINTETITZADOR DIGITAL. D'altra banda, podem obtenir tota la informació continguda en forma digital dins del sintetitzador digital i transmetre-la mitjançant una línia d'interconnexió amb altres aparells d'estructura digital.

En particular, si la línia és connectada amb un microordinador, podem emmagatzemar la informació i, al mateix temps, modificar els paràmetres de cadascun dels sons que tenim ara a la memòria del micro. De la mateixa manera, podem crear els valors dels paràmetres dels sons des del micro i transmetre'ls cap al sintetitzador.

Una melodia no és res més que una cadena de sons, lligats, de diferents tonades i duracions. Un cop a dins de la memòria del micro, podem rectificar la tonada i la duració d'unes determinades notes de la melodia, així com les altres característiques dels sons. Una de les utilitzacions que més potencien la versatilitat dels sistemes de música digital mitjançant el microordinador és la possibilitat de simultaneïtzació (i fins i tot de sincronització) de diferents melodies emmagatzemades independentment, així com el tractament de música mitjançant diferents programes.

Com a punt final de l'aplicació del microordinador en el món de la música, enumerarem alguns exemples de configuracions entre diferents tipus de sintetitzadors, microordinador i perifèrics.

– Interconnexió entre dos sintetitzadors digitals de teclat.

Cadascun dels sintetitzadors té la possibilitat d'ésser receptor i emisor; és a dir, poden

Una configuració sintetitzador/microordinador (Vietronic)



interpretar una melodia des d'un dels dos fent servir els sons creats per l'altre, i viceversa. Així mateix, podem fer servir dos sons diferents interpretant simultàniament la mateixa melodia.

– Connexió d'un sintetitzador i un ordinador. Amb aquesta configuració podrem emmagatzemar diferents melodies (pistes) independents i nota a nota, que, posteriorment, podrem gravar en una unitat d'emmagatzematge tal com una cinta o un diskette. Aquesta informació podrà ésser modificada, i posteriorment gravada, sempre que ho desitgem.

– També podrem utilitzar l'ordinador com a eina de composició.

La composició final podrà executar-se enviant la informació cap al sintetitzador (no és d'estranyar que alguns conjunts musicals hagin incorporat el micro entre el seus instruments de reproducció de música). Aquesta configuració es podrà ampliar connectant a la línia de transmissió altres sintetitzadors. Això ens permet d'interpretar una composició de diferents melodies amb diferents instruments i tot sincronitzat a un determinat ritme d'execució.

– Connexió d'una caixa de ritmes a la línia de transmissió.

Una caixa de ritmes és un sintetitzador especialitzat a generar sons similars als produïts per instruments de percussió, tals com la bateria, timbals, platets, etc. Podem,

doncs, sincronitzar una determinada composició provinent d'algun banc de memòria (per exemple, del microordinador) amb el ritme produït per aquesta caixa de ritmes.

– Configuració:

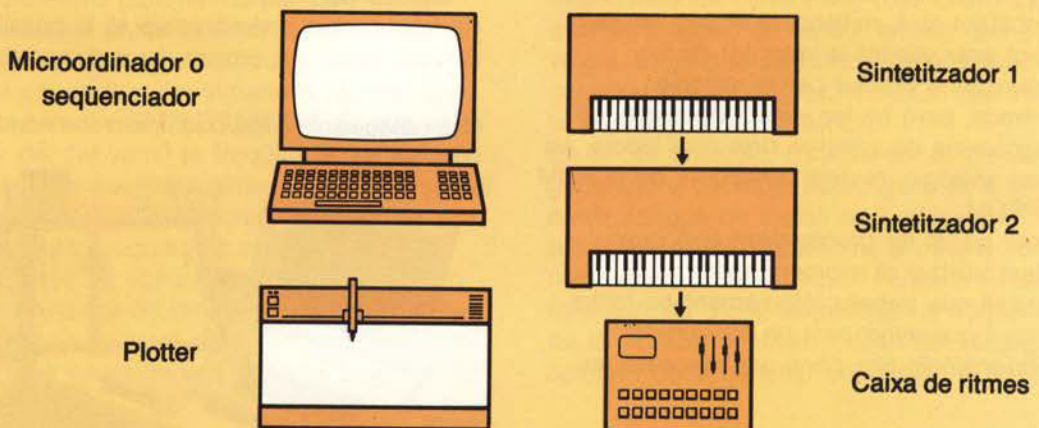
sintetitzador-microordinador-Plotter.

Aquesta configuració ens permet crear o interpretar una composició amb el sintetitzador, gravar-la o modificar-la amb l'ordinador i, mitjançant un programa especial, imprimir posteriorment la partitura musical corresponent amb el Plotter.

– Connexió d'una guitarra sintetitzada a una línia de transmissió.

Una guitarra sintetitzada és un sintetitzador digital en el qual la variació de freqüència del so és produïda mitjançant les diferents posicions dels dits sobre les cordes. Les possibilitats són les mateixes que en el cas d'un sintetitzador de teclat, encara que l'espectacularitat d'aquesta és molt més impressionant. Imaginem-nos que pot representar tocar una guitarra que soni com un saxòfon, o com un violí, o com un piano, o fins i tot com una bateria.

Qualsevol combinació sobre les configuracions anteriors pot ésser realitzada, sempre que la imaginació i els mitjans ho permetin. És important, però, tenir en compte que la música, amb tot allò que representa, només pot néixer de la creativitat de l'ésser humà.



Una configuració possible entre sintetitzadors, microordinador i perifèrics.

Fem música

amb BASIC

Molts microordinadors tenen capacitat per simular instruments musicals.

Els paràmetres que fan servir per definir un so varien d'un micro a l'altre. Mentre el PC d'IBM, el ZX Spectrum (Sinclair) i el Dragon tenen en compte la durada i el to, el COMMODORE té en compte el volum, la velocitat de canvi d'una nota a l'altra, el to i la durada.

En aquest apartat ens dedicarem a programar sons amb el ZX Spectrum; per tant, els paràmetres a utilitzar seran la durada i el to d'una nota.

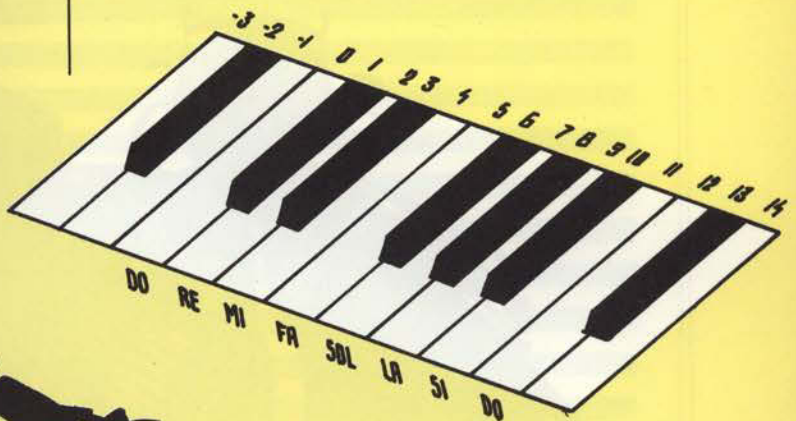
Per tal d'introduir-los al micro, perquè aquest reproduïxi el so desitjat, s'ha d'escriure la següent instrucció:

BEEP durada, to

on, durada i to poden ésser qualsevol tipus d'expressió numèrica. El valor de durada és el temps, en segons, en què la nota està activada (1 = un segon, 2 = dos segons, 0.5 = mig segon, ...) mentre que el valor de to és la quantitat de semitons en què es diferencia la nota que volem sentir del DO central (to = 0). Per sentir el so del DO central hem de posar:

BEEP 1,0

I ens donarà un to molt semblant al DO de l'octava mitjana d'un piano. És sobre aquesta octava que donarem el valor numèric de les notes:



Amb aquesta convenció, l'1 representarà un semitò per sobre del Do central; per tant, correspon amb el DO# (DO sostingut) o amb el REb (RE bemoll) en un piano. De la mateixa manera, el 10 correspon al LA# i el Sib de l'octava central, o el -2, amb les mateixes notes però de l'escala inferior. Si allò que volem és escriure un programa que "toqui" una composició escrita en un pentagrama, s'ha de fer el següent:

- 1.- Mirar en quina clau musical és escrit. Per exemple, si la clau és de SOL, la línia de sota del pentagrama es correspon amb el MI central (el to és 4).
- 2.- Establir el temps en funció del que s'assigni a una negra. Si és d'un segon, a les corxeres els correspondrà mig segon, a les blanques dos segons, etc.
- 3.- Si el pentagrama té definides notes en bemoll o sostingut, mirar quines són; si és bemoll restar 1 al valor del seu to mentre que, si és sostingut, n'hi sumarem 1. Això ho hem de fer també a les mateixes notes de les altres octaves

Així el següent pentagrama,



si assignem 1 segon a les negres i veiem que les notes LA, SI i MI estan bemollitzades, s'escriuria en un programa com:

```
10 REM EXEMPLE MUSICAL
20 BEEP 1,0 : BEEP 1,2
30 BEEP 0,5,3 : BEEP 0,5,2
40 BEEP 1,0
50 BEEP 1,0 : BEEP 1,2
60 BEEP 0,5,3 : BEEP 0,5,2
70 BEEP 1,0
80 BEEP 1,3 : BEEP 1,5
90 BEEP 2,7
100 BEEP 1,3 : BEEP 1,5
110 BEEP 2,7
```

Si ara afegim el següent, i fem RUN, obtindrem una melodia molt coneguda:

```
120 BEEP 0,75,7 : BEEP 0,25,8
130 BEEP 0,5,7 : BEEP 0,5,5
140 BEEP 0,5,3 : BEEP 0,5,2
150 BEEP 1,0
160 BEEP 0,75,7 : BEEP 0,25,8
170 BEEP 0,5,7 : BEEP 0,5,5
180 BEEP 0,5,3 : BEEP 0,5,2
190 BEEP 1,0
200 BEEP 1,0 : BEEP 1,5
210 BEEP 2,0
220 BEEP 1,0 : BEEP 1,5
230 BEEP 2,0
```



Altres

programmes

```
10 REM *****
20 REM *** PROGRAMA TECLAT ***
30 REM *****
40 INK 7
50 CLS
60 FOR I = 1 TO 14
70 FOR J = 0 TO 11
80 PRINT AT 3+J,I*2;CHR$(143);CHR$(144);
90 NEXT I
100 NEXT J
110 INK 2
120 FOR I = 0 TO 30
130 PRINT AT 2,I;CHR$(143);
140 PRINT AT 15,I;CHR$(143);
150 NEXT I
160 FOR A = 2 TO 15
170 INK 2
180 PRINT AT A,0;CHR$(143);
190 PRINT AT A,30;CHR$(143);
200 NEXT A
210 FOR Z = 3 TO 28 STEP 2
220 IF Z = 7 OR Z = 15 OR Z = 21 THEN GOTO 310
230 INK 0
240 FOR A = 3 TO 9
250 PRINT AT A,Z;CHR$(133);INK 7;CHR$(133);
260 NEXT A
310 NEXT Z
320 INK 7
330 PRINT AT 16,2;"q"
340 PRINT AT 16,4;"w"
350 PRINT AT 16,6;"e"
360 PRINT AT 16,8;"r"
370 PRINT AT 16,10;"t"
380 PRINT AT 16,12;"y"
390 PRINT AT 16,14;"u"
400 PRINT AT 16,16;"z"
410 PRINT AT 16,18;"x"
420 PRINT AT 16,20;"c"
430 PRINT AT 16,22;"v"
440 PRINT AT 16,24;"b"
450 PRINT AT 16,26;"n"
460 PRINT AT 16,28;"m"
470 PRINT AT 1,3;"2"
480 PRINT AT 1,5;"3"
490 PRINT AT 1,9;"5"
500 PRINT AT 1,11;"6"
510 PRINT AT 1,13;"7"
520 PRINT AT 1,17;"8"
530 PRINT AT 1,19;"D"
540 PRINT AT 1,23;"G"
550 PRINT AT 1,25;"H"
560 PRINT AT 1,27;"J"
```

```
570 PRINT AT 20,1"ETJEU LA NOTA QUE VOLGUEU "  
580 PRINT AT 22,1"[ O ] PER FINALITZAR "  
590 IF INKEY$="" THEN GOTO 590  
600 IF INKEY$="0" THEN STOP  
610 IF INKEY$="q" THEN LET Z=0 : GOTO 900  
620 IF INKEY$="w" THEN LET Z=2 : GOTO 900  
630 IF INKEY$="e" THEN LET Z=4 : GOTO 900  
640 IF INKEY$="r" THEN LET Z=5 : GOTO 900  
650 IF INKEY$="t" THEN LET Z=7 : GOTO 900  
660 IF INKEY$="y" THEN LET Z=9 : GOTO 900  
670 IF INKEY$="u" THEN LET Z=11 : GOTO 900  
680 IF INKEY$="z" THEN LET Z=12 : GOTO 900  
690 IF INKEY$="x" THEN LET Z=14 : GOTO 900  
700 IF INKEY$="c" THEN LET Z=16 : GOTO 900  
710 IF INKEY$="v" THEN LET Z=17 : GOTO 900  
720 IF INKEY$="b" THEN LET Z=19 : GOTO 900  
730 IF INKEY$="n" THEN LET Z=21 : GOTO 900  
740 IF INKEY$="m" THEN LET Z=23 : GOTO 900  
750 IF INKEY$="2" THEN LET Z=1 : GOTO 900  
760 IF INKEY$="3" THEN LET Z=3 : GOTO 900  
770 IF INKEY$="5" THEN LET Z=6 : GOTO 900  
780 IF INKEY$="6" THEN LET Z=8 : GOTO 900  
790 IF INKEY$="7" THEN LET Z=10 : GOTO 900  
800 IF INKEY$="s" THEN LET Z=13 : GOTO 900  
810 IF INKEY$="d" THEN LET Z=15 : GOTO 900  
820 IF INKEY$="g" THEN LET Z=18 : GOTO 900  
830 IF INKEY$="h" THEN LET Z=20 : GOTO 900  
840 IF INKEY$="j" THEN LET Z=22 : GOTO 900  
850 GOTO 590  
900 BEEP 0.5,Z : GOTO 590
```

```

2 *****
3 *** PROGRAMA PENTAGRAMA ***
4 *****
5 BORDER 0
10 REM CARACTER LINIA DE PENTAGRAMA
15 REM CODI = 144
20 FOR I = 0 TO 7 : POKE USR "a"+I,0
30 NEXT I
40 POKE USR "a"+3,255
50 POKE USR "a"+4,255
60 REM CARACTER NOTA
65 REM CODI = 145
70 POKE USR "b",27
80 POKE USR "b"+1,63
90 POKE USR "b"+2,127
100 POKE USR "b"+3,255
110 POKE USR "b"+4,255
120 POKE USR "b"+5,126
130 POKE USR "b"+6,60
140 POKE USR "b"+7,24
150 REM CARACTER PAL
155 REM CODI = 146
160 FOR I = 0 TO 7 : POKE USR "c"+I,3
170 NEXT I
180 REM CARACTER DELS SEPARADORS
185 REM CODI = 147
190 FOR I = 0 TO 7 : POKE USR "d"+I,16
200 NEXT I
202 DIM N(64)
205 LET CONT=0 : LET IND = 0
210 REM DIBUIX DEL PENTAGRAMA
215 CLS
220 FOR I = 5 TO 13 STEP 2
230 FOR J = 6 TO 31
240 PRINT AT I,J;CHR$(144)
250 NEXT J: NEXT I
260 REM VECTOR DE TONS
270 DIM T(29)
280 LET T(8) = -1 : LET T(9) = 0
290 LET T(10) = 2 : LET T(11) = 4
300 LET T(12) = 5 : LET T(13) = 7
310 LET T(14) = 9 : LET T(15) = 11
320 LET T(16) = 12: LET T(17) = 14
330 LET T(18) = 16: LET T(19) = 17
340 LET T(20) = 19: LET T(21) = 21
350 LET T(22) = 22
360 REM CORRESPONDENCIES ENTRE
370 REM NOTES I LLETRES
380 PRINT AT 3,0;"LA: m"
390 PRINT AT 4,0;"SOL: l"
400 PRINT AT 5,0;"FA: k"
410 PRINT AT 6,0;"MI: j"
420 PRINT AT 7,0;"RE: i"
430 PRINT AT 8,0;"DO: h"
440 PRINT AT 9,0;"SI: g"
450 PRINT AT 10,0;"LA: f"
460 PRINT AT 11,0;"SOL: e"
470 PRINT AT 12,0;"FA: d"
480 PRINT AT 13,0;"MI: c"
490 PRINT AT 14,0;"RE: b"
500 PRINT AT 15,0;"DO: a"
510 REM PREPARACIO DE BEMOLLS
520 REM I SOSTINGUTS
530 INPUT "VOLS SOSTINGUTS O BEMOLLS(S/N)?;C#
540 IF C# = "N" THEN INPUT "PER COMENÇAR PITJA[ENTER]";
S#:GO TO 730

```

```

550 INPUT "NOTA-BEMOLL (ENTRE c i k). [ENTER] PER
    ACABAR:";B#
560 LET BEM = 16-CODE (B#)+97
570 IF BEM < 5 OR BEM >13 THEN GO TO 640
580 IF T(BEM+8) = T(BEM+7) THEN INPUT "CONFIGURACIO
    ERRONEA, PITJEU [ENTER]";S#:GO TO 550
590 PRINT AT BEM-1,7;OVER 1;"b"
600 LET T(BEM) = T(BEM)-1
610 LET T(BEM+8) = T(BEM+8)-1
620 LET T(BEM+16) = T(BEM+16)-1
630 GO TO 550
640 INPUT "NOTA-SOSTINGUT (ENTRE c i k). [ENTER] PER
    ACABAR:";W#
650 LET SOS = 16-CODE (W#)+97
660 IF SOS < 3 OR SOS > 13 THEN GO TO 730
670 IF T(SOS+8) = T(SOS+9) THEN INPUT "CONFIGURACIO
    ERRONEA, PITJEU [ENTER]";S#:GO TO 640
680 PRINT AT SOS-1,9;OVER 1;"#"
690 LET T(SOS) = T(SOS)+1
700 LET T(SOS+8) = T(SOS+8)+1
710 LET T(SOS+16) = T(SOS+16)+1
720 GO TO 640
730 FOR V = 11 TO 31 STEP 2
740 IF V = 11 OR V = 21 OR V = 31 THEN GO TO 760
750 GO TO 800
755 REM DIBUIXAR EL SEPARADOR
760 FOR J = 5 TO 13
770 PRINT AT J,V;OVER 1;CHR$(147)
780 NEXT J
790 GO TO 900
795 REM DIBUIXAR LA NOTA AL SEU LLOC
800 INPUT "ESCRIU LA LLETRE CORRESPONEN A LA NOTA";A#
820 SIT = 15-CODE (A#)+97
830 IF SIT < 3 OR SIT > 15 THEN PRINT AT 20,1;FLAS 1;
    "ERROR, LLETRE INCORRECTA":PAUSE 150:PRINT AT 20,1;
    "":GO TO 800
840 PRINT AT SIT,V;CHR$ (145)
850 PRINT AT SIT-1,V;OVER 1;CHR$ (146)
860 BEEP 0.5,(24-SIT):LET IND=IND+1:LET N(IND)=T(24-SIT)
870 IF SIT <> 3 AND SIT <> 15 THEN GOTO 900
880 PRINT AT SIT,V-1;CHR$(144)
890 PRINT AT SIT,V+1;CHR$(144)
900 NEXT V
903 LET CONT=CONT+1
907 IF CONT > 7 THEN GOTO 930
910 INPUT "VOLS SEGUIR POSSANT NOTES? (S/N)";C#
940 IF C# <> "S" THEN GOTO 990
945 CLS
950 FOR I = 1 TO IND: BEEP 0.5,N(I)
960 NEXT I
970 INPUT "VOLS TORNAR A ESCOLTAR-LA ? (S/N) ";C#
980 IF C#="S" THEN GOTO 950
990 INPUT "VOLS ESCRIUREN UNA ALTRA ? (S/N) ";C#
1000 IF C#="S" THEN GOTO 205
1010 STOP

```

Versió pel ZX-Spectrum

```
*****
*** PROGRAMA FORMIGUES *****
*****
```

```
1 REM *** AUTOR:FERRAN ANDREU BUSTAMENTE *****
2 V=53248:POKEV+21,0
3 FOR B=0TO15:POKEV+B,0:NEXTB
5 PRINT "<1CLR>"
6 POKE 53280,6: POKE 53281,1
7 PRINT "<10CSRD><8CSRR><CTRL-1RVSON><CTRL-BLU>"
LA GUERRA DE LES FORMIGUES"
9 GET T$: IFT$ <>" " THEN?
10 PRINT "<1CLR>"
20 POKE 53280,0: POKE 53281,14
25 PRINT "<16CSRD>"
30 FOR I = 1 TO 639 :
40 PRINT "<CTRL-1RVSON><CTRL-GRN>";:
50 NEXT I
500 POKE 2040,192: POKE 2041,193: POKE 2042,194:
POKE 2043,195: POKE 2044,196
501 POKE 2045,197: POKE 2046,198: POKE 2047,199
510 FOR A = 0 TO 62: READ X : POKE 12289+A,X: NEXT A
520 FOR B = 0 TO 62: READ X : POKE 12352+B,X: NEXT B
530 FOR C = 0 TO 62: READ X : POKE 12416+C,X: NEXT C
540 FOR D = 0 TO 62: READ X : POKE 12480+D,X: NEXT D
550 FOR E = 0 TO 62: READ X : POKE 12544+E,X: NEXT E
560 FOR F = 0 TO 62: READ X : POKE 12608+F,X: NEXT F
570 FOR G = 0 TO 62: READ X : POKE 12672+G,X: NEXT X
580 FOR H = 0 TO 62: READ X : POKE 12736+H,X: NEXT X
600 V=53248 : GOSUB 15000
650 GOSUB 16000
700 GOSUB 7000
800 GOSUB 8000
850 GOSUB 9000
900 GOSUB 9500
950 GOSUB 20000
960 FOR H = 0 TO 150: POKEV+8,65+H: POKEV+9,150+SIN(H):
NEXT H
961 FOR S = 0 TO 35:
962 POKEV+8,215+S+((-1)^S): POKEV+9,150-(1.5*S):
963 FOR G = 0 TO 200 : NEXT G
964 NEXT S
1000 FOR P = 0 TO 62: READ W: POKE 12480+P,0:
POKE 12480+P,W : NEXT P
1001 POKEV+6,250: POKEV+7,99: POKEV+21,231: POKEV+42,0:
POKEV+21,239
2000 FOR J = 0 TO 250: POKEV+6,250-J: POKEV+7,99+SIN(J)
2010 IF J = 67 OR J = 115 OR J = 164 THEN GOSUB 25000
2020 NEXT J
2999 POKEV+21,224
3000 RESTORE: POKEV+21,0: GOTO 10
7000 POKEV+10,180: POKEV+11,214: POKEV+44,1: POKEV+29,56:
7010 POKEV+23,56: POKEV+21,55
7020 RETURN
8000 POKEV+6,0: POKEV+7,0: POKEV+21,63:
8010 FOR S = 0 TO 80
8020 POKEV+6,255-S: POKEV+7,150+SIN(S)
8030 NEXT S
8040 RETURN
9000 POKEV+21,47: POKEV+23,248: POKEV+29,248: POKEV+12,120:
9010 POKEV+13,195: POKEV+45,0: POKEV+21,111:
9020 RETURN
9500 POKEV+8,0: POKEV+9,0: POKEV+21,127:
9510 FOR G = 0 TO 64
9520 POKEV+8,G: POKEV+9,150+SIN(G):
9530 POKEV+6,175-G: POKEV+7,150+SIN(G)
9540 NEXT G
```


Les imatges i els sons: una forma d'expressió 3

Quelcom més que records i imaginacions 5

Programem en BASIC 9

Les imatges

Les imatges son una col·lecció de punts 17

Del continu al discret: els pixels 18

Filtrem imatges 20

Memòria i imatge 22

Tècniques d'emmagatzematge 23

Gràfics i imatges 26

Reconstrucció de les imatges 27

Una aplicació en cartografia 29

Dibuixem en BASIC 31

Els sons

Què és el so 40

Vibracions i ones 41

Què podem escoltar? 43

Transmissió: fem viatjar el so 45

Guardem els sons 48

L'ordinador i els sons 49

Ordinadors i sintetitzadors: música digital 51

Fem música amb BASIC 55

Altres programes 57

Volem agrair al Departament de Planificació Urbanística de l'Ajuntament de Barcelona i a la casa VIETRONIC S.A., la informació i material gràfic que ens han facilitat de forma desinteressada.

**L'Obra Social de la Caixa de Pensions
posa a la vostra disposició,
dins la seva xarxa de Biblioteques
el nou servei de**

MICROTEQUES

La biblioteca,
entesa com un espai de recursos per a la informació i la formació,
amb aquest servei incorpora noves tècniques i mitjans
que permetran l'usuari
interactuar amb la documentació i les dades.

A la MICROTECA trobareu:

- Uns microordinadors i uns perifèrics que han estat triats en funció dels objectius del servei.
 - Uns programes i uns paquets informàtics seleccionats en funció de la seva qualitat, autonomia explicativa i llengua.
 - Una informació escrita i documentació variada que permetrà usar eficaçment les eines informàtiques.
 - Els serveis normals de la biblioteca i dels seus professionals que us posaran a disposició bibliografia especialitzada i l'eficàcia organitzativa del servei.
 - Un programa de dinamització que promourà accions de reflexió, informació i estudi a l'entorn dels temes informàtics de major interès.
-

**Les primeres Microteques
començaran a funcionar properament a les següents Biblioteques:**

TARRAGONA
C. Colom, 2

LLEIDA
C. Bisbe Torres, 2

GIRONA
C. Migdal, 32

BARCELONA
C. Clot, 21-25

MANRESA
Guimerà, 1

Podeu demanar més informació al telèfon (93) 302 54 04 ext. 206

Col·lecció

connecta el micro

- 1 FEM INFORMÀTICA
- 2 IMATGES I SONS
- 3 TRACTAMENT DE LA INFORMACIÓ
- 4 ROBOTS I INTEL·LIGÈNCIA ARTIFICIAL
- 5 NOVES TECNOLOGIES

Servei de Publicacions de la



FUNDACIÓ CAIXA DE PENSIONS

El programa «connecta el micro, pica l'start» és una iniciativa conjunta de Caixa de Pensions «La Caixa» i TV3

275 Ptes.



<https://electrodreams.blog/>