

Se una volta esisteva il «cuneo chiamato Giulietta» che doveva il suo nome appunto alla sua spiccata aerodinamicità che gli permetteva di fendere facilmente l'aria innanzi a sé quando correva velocemente, oggi esiste un cuneo umano (di nome appunto Cunei) che ama inCunearsi nell'altrui operato spulciando difetti, errori ed omissioni che puntualmente ci comunica con le sue letterine rompiscatole (così almeno le chiama lui). Ci ha provato anche col sottoscritto quando in Appunti di Informatica mi venne la malsana idea di vagheggiare in computabilità logica, con macchine di Turing, funzioni calcolabili e non, infiniti, cardinalità e idempotenza, innestando poi una perversa reazione a catena per la quale ha continuato a leccarsi le ferite per alcuni mesi dopo l'accaduto. Con questo non voglio dire che disdegnamo interventi di questo genere, anzi ben vengano quanti più Cunei possibile (quello attuale è anche particolarmente simpatico...) ma che la critica (per usare la solita frase fatta) sia costruttiva...

Fortunatamente questo è proprio il caso nostro. La letterina di protesta del Cunei questa volta silura gli articoli di Paolo Russo che ci ha «cantato» del Blitter per circa tre mesi. La diatriba, come era da aspettarselo, riguarda il procedimento adottato per ottenere una data cosa da un determinato ambiente. Vi rigiriamo pari pari la sua missiva che, lungi dall'essere una sfida a noi, a voi o al Russo, rappresenta comunque un importante intervento sulla grafica di Amiga che non smetterà mai di stupirci. Prima di lasciare a lui la parola (ricordiamoci che anche se non morde è pur sempre pericoloso...) ci auguriamo di ascoltare presto altre autorevoli voci in capitolo ma soprattutto di vedere quanto prima applicati tutti questi consigli in almeno un programma opera dei lettori che vorranno cimentarsi nella realizzazione di una simile impresa (il livello dei programmi che ci giungono in redazione continua ad essere, ahimè, troppo basso...) adp

## HAM, HAM

di Antonio Cunei - Monfalcone (GO)

Salve a tutti! Vi ricordate di me? Sono Cunei, quello del giochino delle sequenze... non vi ricordate di me!?!...

Sic transit gloria mundis...

Vabbé, procediamo. Orbene, mi è dato di leggere sul numero 69 di MCmicrocomputer un interessante articolo di Paolo Russo, il quale ha creato un LIFE incredibilmente veloce per Amiga grazie all'aiuto del Blitter. In mezzo a varie altre elucubrazioni si può leggere la seguente frase (testuale): «Infine mi rendo conto che per predisporre uno schermo in modo HAM sono necessarie trentacinque (!) chiamate alla ROM: su qualunque altro computer sarebbe bastato pokare qualche locazione».

Ohibò! Quale terribile maledizione impedisce agli Amighevoli smanettoni di ottenere uno schermo HAM con trentaquattro, o trentatré chiamate? E quale mezzo magico consente dunque di ottenere il famigerato modo HAM alle ROM che, alla stessa stregua del programmatore (si suppone), non hanno possibilità di ottenerlo «pokando qualche locazione»? Dobbiamo dunque arguire che i sadici progettisti del sistema abbiano impedito all'utente di accedere ai registri hardware, costringendolo dunque ad ardui funambolismi in mezzo alle routine del sistema operativo?...

Tranquilli, cari colleghi, sono qui io a levarvi da tali turbe esistenziali. Non sarete dunque più costretti all'«acquisto di un ST» (sic) per soddisfare la vostra sete di grafica. (Nota: è altresì vero che una buona dose di sadismo effettivamente pare esserci nei progettisti del S.O.).

### ROM & Libraries

Donde nasce l'equivoco? È presto detto: come molti sapranno, nel sistema operativo di Amiga esistono diverse librerie di routine già belle e pronte da usare, alcune delle quali risiedono su

ROM, e altre si trovano invece nel device logico "LIBS:", dal quale vengono caricate in caso di necessità. Le succitate librerie non corrispondono però, come si potrebbe credere, ad un unico livello di astrazione; alcune (Exec gestiscono funzioni centrali del sistema (gestione dei task e dei loro messaggi, della memoria, dell'I/O etc.), altre gestiscono le primitive grafiche del sistema (Graphics), altre ancora gestiscono un evolutissimo sistema di comunicazione interattivo con l'utente (Intuition) e consentono di creare schermi, finestre negli schermi, gadget nelle finestre, menu, etc., etc., etc., il tutto con una facilità incredibile (sempre relativamente alla media di Amiga). Così, capita che per creare uno schermo si possano chiamare routine di Exec per allocare memoria per la palette dei colori, per i Bitplanes, per le strutture del sistema, etc., e si debbano chiamare routine assortite di Graphics per creare Screen, Rastport, View, Viewport, RasInfo, e via impazzendo. Si può facilmente capire come si possa giungere a trentacinque (folli!) chiamate. D'altronde si può anche decidere di fare ricorso ad Intuition, e crearsi così uno schermo HAM in modo veramente chic con UNA chiamata ad una routine che si chiama OpenScreen, e che si può parimenti chiamare da C come da AmigaBasic (stupiti?). Vi prego dunque di notare la pregevole fattura del programma IntuiHam.c, scritto in Lattice C, il quale crea uno schermo HAM alquanto ciociottello (340 per 275), vi disegna un po' dentro, quindi lo porta un po' a spasso per lo schermo e infine lo cancella. Né d'altronde pare da disdegnare un'occhiatina al programmino in AmigaBasic che crea un analogo schermo in HAM e vi schiaccia dentro rettangoli pieni fino alla pressione del tasto sinistro del mouse nella finestra principale del Basic. Ah, quasi dimenticavo, lo schermo così generato è sempre uno schermo di Intuition, e come tale può essere spostato su e giù e mandato davanti e dietro agli altri schermi grazie agli invisibili gadget nell'angolo in alto a destra.

Come dite? Vorreste ulteriori dettagli su come ottenere tali schermi? Ah, curiosità insaziabile! Vabbé, ma solo perché siete voi.

### Apriamo gli schermi

Allora: se desiderate che Intuition crei uno schermo per voi tutto quello che dovete fare è:

- comunicare al sistema che intendete usare Intuition con una OpenLibrary;
- creare una struttura NewScreen e riempirla con le caratteristiche che desiderate per il vostro schermo;
- chiamare la routine che crea lo schermo (OpenScreen), che vi ritorna un puntatore ad una struttura Screen che rispecchia lo stato attuale del vostro schermo;
- usare lo schermo come più vi piace (aprendoci dentro finestre, attaccandoci

menu o semplicemente disegnandoci dentro passando alle routine di Graphics il puntatore alla struttura RastPort che è contenuta nella struttura Screen (il valore del puntatore alla struttura Screen incrementato di 84));

— chiudere lo schermo chiamando CloseScreen;

— comunicare al sistema che non avete più bisogno di Intuition mediante una chiamata a CloseLibrary.

Ora tutto quello che vi serve ancora è conoscere cosa dovete mettere dentro alla struttura NewScreen. Vediamo insieme come tale struttura è definita in intuition/intuition.h per quanto riguarda il C:

```
struct NewScreen
{
    SHORT LeftEdge, TopEdge, Width, Height,
    Depth;
```

```
    UBYTE DetailPen, BlockPen;
    USHORT ViewModes;
    USHORT Type;
    struct TextAttr * Font;
    UBYTE * DefaultTitle;
    struct Gadget * Gadgets;
    struct BitMap * CustomBitMap;
};
```

I primi cinque valori sono (interi di due byte con segno): posizione a sinistra e in alto dell'angolo in alto a sinistra (per l'appunto); il manuale suggerisce di porli a zero; la larghezza e l'altezza dello schermo e il numero di BitPlane. Seguono due byte che indicano i colori da usare per le superfici e per le linee (per il titolo). Vanno bene zero e uno, in generale. Segue un intero a due byte che dovrà essere costruito con OR dei seguenti valori:

```
#define PFBA 0x40
#define DUALPF 0x400
#define HIRES 0x8000
#define LACE 4
#define HAM 0x800
#define SPRITES 0x4000
#define VP_HIDE 0x2000
#define GENLOCK_AUDIO 0x100
#define GENLOCK_VIDEO 2
#define EXTRA_HALFBRITE 0x80
```

Naturalmente non tutte le combinazioni sono possibili. (Nota: nelle prime versioni del sistema operativo la routine OpenScreen non supportava tutte le possibilità e infatti il manuale non fa cenno della possibilità di usarla per schermi HAM e simili. Che sia questo che ha tratto in inganno il nostro Russo?).

L'intero Type dovrà similmente essere ricavato «ORando» alcuni a piacere fra i seguenti:

```
#define SCREENTYPE 0x000F
#define WBENCHSCREEN 0x0001
#define CUSTOMSCREEN 0x000F
#define SHOWTITLE 0x0010
#define BEEPING 0x0020
#define CUSTOMBITMAP 0x0040
#define SCREENBEHIND 0x0080
#define SCREENQUIET 0x0100
```

Per ragioni di spazio non posso illustrarvi tutte le combinazioni di ViewModes e Type: non è comunque difficile ottenerle con il ben noto meccanismo del prova e riprova (tanto caro a noi hacker).

Infine quattro puntatori, rispettivamente per il font da usare per lo schermo (NULL per il Topaz 8), uno per il titolo (NULL per nessun titolo), uno per i

#### HAM da Basic

```

/
/ HAM da BASIC? Si, grazie!
/
/ Tony's creation 20-Mar-88
/
////////////////////////////////////
DECLARE FUNCTION OpenScreen& LIBRARY

LIBRARY "graphics.library"
LIBRARY "intuition.library"
OPTION BASE 0
DIM newscr%(15)
FOR x=0 TO 15: READ newscr%(x):NEXT

scr& = OpenScreen& (VARPTR(newscr%(0)))
IF scr&=0 THEN PRINT "Poca memoria!";GOTO NienteSchermo

rp&=scr&+84&
WHILE MOUSE(0)=0
    x&=RND*48+16
    CALL SetAPen (rp&,x&)
    z&=RND*335
    y&=CLNG(RND*240/30)*30&
    CALL RectFill (rp&,z&,y&,z&+1&,y&+29&)
WEND
FOR x=1 TO 260
    CALL MoveScreen&(scr&,0&,1&)
NEXT
CALL CloseScreen&(scr&)

NienteSchermo:
LIBRARY CLOSE

DATA 0,0,340,275,6,1,2048: 'HAM
DATA 271,0,0,0,0,0,0,0,0
/ CUSTOMSCREEN : SCREENQUIET
```

## IntuiHAM.c

```

/*
   File name: IntuiHAM.c
   ~~~~~
   Ovvero: come usare TUTTE le risorse di Amiga e vivere felici.
   ~~~~~
   (Tony's creation 20-Mar-88)

*/

#include <intuition/intuition.h>
#include <graphics/gfxbase.h>
#define Mx 340
#define My 275

extern struct GfxBase *GfxBase;
extern struct IntuitionBase *IntuitionBase;

struct Screen *scr;
struct RastPort *rp;

struct NewScreen nscr =
{
  0,0,Mx,My,6,
  0,1,HAM,
  CUSTOMSCREEN:SCREENQUIET,
  NULL,
  (UBYTE *) "Schermo nuovo di zecca!",
  NULL,NULL
};

void quit(errno)
uns errno;
{
  switch (errno)
  {
    case 3: CloseScreen (scr);
    case 2: CloseLibrary (IntuitionBase);
    case 1: CloseLibrary (GfxBase);
    case 0: abort ();
  }
}

void _main() /* (non serve l'interfaccia standard Lattice) */
{
  long a,b,x=0L;

  if ((GfxBase=OpenLibrary ("graphics.library",0))==NULL) quit (0);
  if ((IntuitionBase=OpenLibrary ("intuition.library",0))==NULL) quit (1);
  if ((scr=OpenScreen (&nscr))==NULL) quit (2);

  rp=&scr->RastPort;

  for (b=1; b<10; b++)
  {
    for (a=0; a<My; a++)
    {
      SetAPen (rp,(x++)%63);
      Move (rp,0,a);
      Draw (rp,(long)(Mx-1-(float)Mx/My*a),0);
      Draw (rp,Mx,My-1-a);
      Draw (rp,(long)((float)Mx/My*a),My-1);
      Draw (rp,0,a);
    }
  }

  for (a=1; a<100000; a++);

  for (a=0; a>20; a--)
  {
    scr->ViewPort.DxOffset=a;
    MakeVPort (GfxBase->ActiView,&scr->ViewPort);
    MrgCop (GfxBase->ActiView);
    LoadView (GfxBase->ActiView);
    for (b=1; b<40000;b++);
  }

  for (a=1; a<200000; a++);

  quit (3);
}

```

Gadgets (NULL sempre. In effetti non si possono attaccare Gadgets agli schermi, ma bisogna farlo con le BackDrop Windows... beh, di questo parleremo un'altra volta) e uno se eventualmente volete fornire voi il BitMap (per esigenze particolari; anche qui piazzate un bel NULL).

**Ultimi dettagli**

Anf, mi pare che sia tutto!

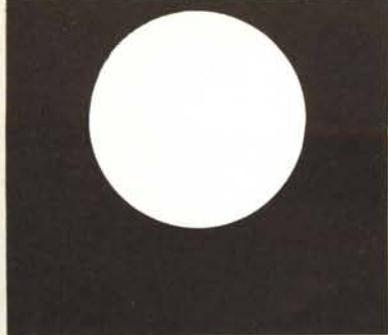
Ancora una nota finale per quanto riguarda il programma in Basic: per poterlo usare dovete mettere nella directory corrente i file «Graphics.bmap» e «Intuition.bmap» che potete ricavare con il programma ConvertFD che trovate sul disco Extras; i dettagli per l'operazione sono alla pagina 777 di Tel..., volevo dire, nell'appendice F del manuale del Basic.

Ah, quasi mi dimenticavo, voi vorrete senza dubbio avere la vostra palette di colori personalizzata; esistono diversi sistemi per cambiare i colori di uno schermo, ma il più facile è usare la routine SetRGB4 (vp,n,r,g,b), dove vp è il puntatore alla struttura ViewPort (cioè quello dello schermo più 44), n è il numero del colore che desiderate cambiare ed r,g e b sono le componenti di rosso, blu e verde del colore che desiderate (da 0 a 15). Se lavorate in Basic ricordatevi di usare sempre variabili e valori interi (per i puntatori interi lunghi).

Detto tutto, un paio di notiziole di bonus, visto che mi avete letto fino a qui: è in giro UN ALTRO VIRUS! Quest'ultimo è di «Byte Bandits» ed ha come effetto quello di fare impiantare il computer un certo tempo dopo il reset (cosa che d'altra parte lo rende facilmente individuabile). Giusto a proposito di virus: volete sapere con assoluta certezza se avete il virus SCA annidato in memoria? È sufficiente che teniate il pulsante sinistro del mouse premuto durante il reset: se il virus è in memoria lo schermo diventerà verdolino per circa mezzo secondo.

E con questo ho veramente finito. Ciao!

**Post Scriptum:** se, dopo tutto quanto avete letto, desiderate ancora «pokare in qualche locazione», potete provare a settare il bit 11 della word \$DFF100. Peccato che il sistema si affretti a resettarvelo, altrimenti potreste probabilmente vedere uno splendido esempio di schermo in modo HAM... 



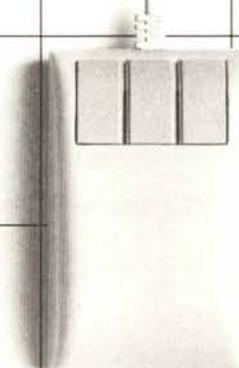
## ANKO MOUSE MODEL AK-6000

- CUSTOM C-MOS TECHNOLOGY
- SCANNER OTTICO PER MASSIMA SENSIBILITÀ
- DESIGN ERGONOMICO
- MOUSE SYSTEM E MICROSOFT COMPATIBILE
- RISOLUZIONE 200 DOT/INC
- NON RICHIEDE ALIMENTAZIONE ESTERNA
- PER IBM PC/XT/AT E COMPATIBILI
- SOFTWARE IN DOTAZIONE: TEST MOUSE E DRIVERS
- DISPONIBILE ADATTATORE 25/9 PINS PER PC/AT
- PAD MOUSE ANTISTATICO

### EXECUTIVE DISTRIBUISCE:

- SUPER EGA CARD 800 X 600
- HANDY SCANNER
- MODEMS
- GRUPPI DI CONTINUITÀ
- SCHEDE ANALOGICO DIGITALI
- RETI LOCALI
- JOYSTICK ANKO
- GAME I/O CARD

IBM, MICROSOFT, ANKO  
sono marchi registrati



Lit. 59.000 + IVA

# ANKO MOUSE

### EXECUTIVE COMPUTER DEALER

Via Bovara, 16  
22053 LECCO (CO)  
Tel. 0341/364706  
Fax 0341/365646

Per informazioni:

NOME \_\_\_\_\_

DITTA \_\_\_\_\_

INDIRIZZO \_\_\_\_\_

TEL. \_\_\_\_\_

ATTIVITA' \_\_\_\_\_