

Programmare in C su Amiga (40)

di Dario de Judicibus

Con questa puntata chiudiamo la serie sui controlli proporzionali e, almeno per il momento, anche le puntate dedicate ad Intuition. Vedremo inoltre la penultima serie di schede relative alle funzioni della **graphics.library**, versione 2

Abbiamo visto nella scorsa puntata le tre funzioni base che permettono la creazione, la visualizzazione e la rimozione delle barre semplici da una finestra o da un quadro. Vediamo ora come vanno gestite tali barre durante l'esecuzione di un programma.

La gestione delle barre

Supporteremo come al solito di estendere il *programma scheletro* con l'aggiunta di un nuovo menu per la creazione e la rimozione dei vari tipi di barra, come già fatto in passato per altri controlli. In pratica avremo un menu chiamato «**Potenzimetri**» con una serie di voci, di cui la prima sarà appunto «**Barre semplici**». Con questo termine ci riferiamo appunto a semplici rettangoli in cui scorre un cursore, senza alcun controllo addizionale associato, come potrebbero essere ad esempio i pulsanti di scorrimento (o frecce).

Supporteremo inoltre di avere sottovoci per ogni voce (vedi figura 1) per visualizzare e rimuovere i controlli dalla finestra. Se quindi l'utente selezionerà in sequenza «**Potenzimetri**», «**Barre semplici**» e «**Crea**», compariranno nella finestra del programma dimostrativo due barre semplici, una verticale ed una orizzontale. A questo punto, selezionando di nuovo «**Potenzimetri**» e «**Barre semplici**» e, successivamente, «**Cancel**», le due barre saranno rimosse dalla finestra.

In figura 2 è riportato un estratto del codice che viene attivato quando l'evento **MENUPICK** è emesso da Intuition a fronte della selezione da parte dell'uten-

te delle voci in questione. In entrambi i casi viene chiamata la funzione **SimpleBars()** riportata in figura 3.

Questa funzione, analoga a quelle già viste nelle scorse puntate per i vari tipi di pulsanti e di campi, accetta un solo parametro in ingresso. Se questi ha come un valore **TRUE**, viene chiamata la **CreateBar()** per creare due barre semplici, una orizzontale ed una verticale. se le barre sono già state create in precedenza, invece, si limita ad emettere un messaggio d'errore. Se viceversa il parametro in questione assume come valore **FALSE**, eventuali barre create in precedenza vengono rimosse dalla finestra. Anche in questo caso, una doppia chiamata di rimozione genera un messaggio di errori.

I parametri passati alla **CreateBar()** sono definiti in figura 4 come variabili globali e costanti per il preprocessore. Da notare che la struttura **BarInfo**, o più semplicemente **BINFO**, contiene i valori logici associati alla barra, e non quelli fisici. Ad esempio, nel caso della barra orizzontale, il cursore può assumere valore in un arco che va da meno venti unità a centoventi, con incrementi di venti unità per volta, mentre il valore iniziale è zero. Tali valori non hanno niente a che vedere con i valori effettivamente assunti dal cursore per Intuition, il quale ragiona sempre in termini di frazioni di **MAXPOT**. Infatti, a parità di dimensioni fisiche e forma del potenziometro, avremmo potuto benissimo definire una scala da 300 a 4000, con passi di 50 unità e valore iniziale 2550. Per Intuition non sarebbe cambiato niente. Sono le funzioni di interfaccia che abbiamo visto nella scorsa puntata, e quelle che vedremo più avanti in questa, ad occuparsi delle conversioni tra valori logici e valori reali.

Torniamo ora al programma dimostrativo. Se a questo punto l'utente fa *click* con il mouse nel contenitore di una delle due barre, oppure aggancia il cursore di una barra e lo fa scorrere con il bottone sinistro del mouse premuto e poi rilascia

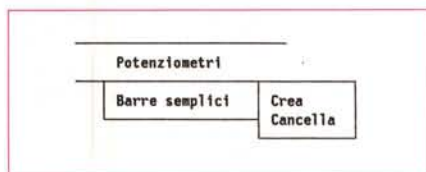


Figura 1 - Menu per le barre.

```

/*****
** H_MenuPick: gestisce l'evento MENU PICK
**
*****/
int H_MenuPick(msg)
    INMSG *msg;
{
    MENU *strip;
    ITEM *select;
    struct Window *whichwin;
    USHORT id, menunum, itemnum, subnum;

    /*
    ** E' stata fatta effettivamente una selezione? Se si, allora
    ** determiniamo a quale finestra si riferisce, e quindi a quale
    ** menù, così possiamo usare questa procedura per gestire più menù
    ** appartenenti a più finestre.
    */
    id = msg->Code;          /* Identificativo selezione */
    while (id != MENU NULL) /* id = sssssiiiiimmm (16 bits) */
    {
        menunum = MENUNUM(id); /* Identificativo del menù */
        itemnum = ITEMNUM(id); /* Identificativo della voce */
        subnum = SUBNUM(id);   /* Identificativo della sottovoce */

        whichwin = msg->IDCMPWindow; /* Indirizzo della finestra */
        strip = whichwin->MenuStrip; /* Indirizzo della barra dei menù */
        select = ItemAddress(strip, id); /* Indirizzo della voce selezionata */

        /*
        ** BLOCCO PER LA GESTIONE DEI CODICI
        */
        switch (menunum)
        {
            :
            case M_PROBAR:
                switch (itemnum)
                {
                    case I_BARSMP:
                        switch (subnum)
                        {
                            case S_SMPON : SimpleBars(TRUE); break;
                            case S_SMPOFF: SimpleBars(FALSE); break;
                        }
                        break;
                }
                break;
        }

        /*
        ** Vediamo ora se l'utente ha effettuato una selezione multipla
        */
        id = select->NextSelect;
    }
    return(GOAHEAD);
}

```

Figura 3
SimpleBars().

il bottone con il cursore ancora posizionato nell'area di selezione del potenziometro, Intuition emette un messaggio di tipo **GADGETUP**. Vi ricordo che ciò avviene solo perché abbiamo deciso, nella **CreateBar()**, di assegnare al campo **Gdg->Activation** solo il valore **RELVE-RIFY**. Nulla ci avrebbe impedito di aggiungere anche **GADGETIMMEDIATE**, ed intercettare così anche il messaggio **GADGETDOWN**. A questo punto, tuttavia, sarebbe stato necessario definire un meccanismo di sincronizzazione tra le due funzioni che gestiscono tali messaggi, per loro natura asincroni. Questo ci avrebbe offerto probabilmente possibilità più sofisticate di gestione dei controlli, al prezzo di una maggiore occupazione di memoria ed un'ulteriore serie di verifiche incrociate da effettuare. Chi

Figura 2
H_MenuPick().

vuole, può provare a disegnare un algoritmo a tale scopo. Si tratta indubbiamente di un esercizio interessante, e non solo in relazione ad Intuition ed alla programmazione su Amiga.

Tornando ora al codice in figura, il messaggio viene intercettato come al solito dalla funzione **HandleEvent()** che chiama l'**H_GadgetUp()**. Questa, riportata in figura 5, almeno per quella parte che interessa le barre semplici, utilizza le due funzioni di lettura **HBarValue()** e **VBarValue()** per visualizzare il valore assunto dal cursore quando l'utente ha finito di operare con il potenziometro. Vediamo in dettaglio queste funzioni.

Le funzioni di lettura

Abbiamo chiamato queste due funzioni di lettura, in quanto esse «leggono» il valore corrente della posizione del cursore in un potenziometro dai campi della struttura **PropInfo** associata ad esso gestiti direttamente da Intuition. Tale valore sarà quindi successivamente con-

```

/*****
** SimpleBars: crea o cancella barre semplici (senza frecce)
**
*****/
void SimpleBars (onoff)
    BOOL onoff;
{
    if (onoff)
    {
        if (mask & MSK_SHP)
            ShowMsgReq(w, "Barre già visibili");
        else
        {
            /*
            ** Crea le barre
            */
            bars[BAR0001] = CreateBar(HSMPBAR, &c, HSMPXTX, &binfo, NULL,
                &pbbar, NOBARARROWS|HORIZONTALBAR);
            if (bars[BAR0001] == NULL) CloseAll(EMSG_NOMEMORY);

            /*
            ** Attenzione: basta una sola barra per attivare la maschera
            */
            mask |= MSK_SHP;

            bars[BAR0002] = CreateBar(VSMPBAR, &c, VSMPXTX, NULL, &binfo,
                &pvbar, NOBARARROWS|VERTICALBAR);
            if (bars[BAR0002] == NULL) CloseAll(EMSG_NOMEMORY);

            /*
            ** Visualizza i campi
            */
            DisplayBars(bars[BAR0001], &c);
            DisplayBars(bars[BAR0002], &c);
        }
    }
    else
    {
        if (!(mask & MSK_SHP))
            ShowMsgReq(w, "Barre già cancellate");
        else
        {
            if (bars[BAR0001]) DeleteBar(bars[BAR0001], &c);
            if (bars[BAR0002]) DeleteBar(bars[BAR0002], &c);
            mask &= ~MSK_SHP;
        }
    }
}

```


Figura 4
Definizione per le
barre.

```

/*
** Barre
*/
#define NUMBARS 8
IGDG *bars[NUMBARS];      /* Vettore dei puntatori alle barre */

/*
** Definizioni per le barre semplici
*/
#define BAR0001 1
#define HSMPTXT 80
#define HSMPTXT "Orizzontale semplice"
BINFO hbinfo = { 20, 100, 60, 10 };
PRECT phbar = { 50, 50, 120, 12 };

#define BAR0002 2
#define VSMPTXT 82
#define VSMPTXT "Verticale semplice"
BINFO vbinfo = { -20, 120, 0, 20 };
PRECT pvbar = { -90, 30, 24, 60 };

```

vertito nel corrispondente valore logico associatogli, grazie alle informazioni salvate dalla **CreateBar()** nella struttura di servizio **UsrBar**.

Sia la **HBarValue()** che la **VBarValue()** accettano in ingresso un solo parametro, e precisamente il puntatore alla struttura **Gadget** corrispondente al controllo interessato. Per il resto sono praticamente identiche, almeno come logica. Si differenziano solo per i campi e le variabili delle strutture **PropInfo** e **UsrBar** che utilizzano: relativi allo scor-

gimento orizzontale del cursore la prima, e relativi allo scorrimento verticale, la seconda. Ovviamente, nel caso di un potenziometro a *piastra*, andranno usate entrambe le funzioni. Il valore logico è ritornato come **SHORT** in entrambi i casi.

Anche a queste funzioni si applica il discorso fatto nella scorsa puntata relativamente all'algoritmo di conversione dai valori reali a quelli logici. In particolare, qualunque algoritmo più o meno sofisticato si sia usato nella **CreateBar()**

per evitare la propagazione degli errori dovuti agli arrotondamenti nei calcoli parziali effettuati dal computer, sarà necessario adottare qui la tecnica inversa. È importante capire che, al contrario di quanto avviene in matematica pura, nel caso di algoritmi di calcolo, così come nel caso dell'elaborazione di dati sperimentali, le tecniche di calcolo inverse possono presentare problemi differenti da quelle normali, come si può vedere da alcuni esempi in figura 8.

Nella figura in questione, la formula inversa sembra presentare rischi minori di quella diretta, qualora si utilizzi la rozza tecnica del calcolo a parziali interi. Per altre formule succede il contrario. Ogni caso va analizzato a sé.

Se e quanto effettivamente un algoritmo è sicuro, è funzione di molti fattori. Non esiste una regola unica per calcolare l'ambito di validità o comunque l'area di sicurezza di un algoritmo generico. Una tecnica molto utile tuttavia, è quella basata sul calcolo differenziale, utilizzato moltissimo nella teoria della propagazione degli errori nella elaborazione di dati sperimentali.

Data infatti una funzione di una variabile **x**, l'errore sulla funzione è dato dalla derivata della stessa rispetto ad **x** moltiplicato per l'errore su **x**. Il risultato può essere a sua volta una funzione di **x**, e di conseguenza l'errore si può propagare in modo più o meno *drammatico* a seconda del valore stesso di **x**. Un calcolo dei punti in cui si annulla la derivata seconda, o, come nei casi riportati in figura 9 e figura 10, l'individuazione di eventuali punti di discontinuità, può fornire un'utile indicazione delle aree a rischio.

Attenzione però. Nel caso dei calcoli

```

/*****
** H_GadgetUp: gestisce l'evento GADGETUP
**
*****/
int H_GadgetUp(msg)
    INMSG *msg;
{
    IGDG *gdg;
    USHORT gid;
    char *temp;

    /*
    ** E' stata fatta effettivamente una selezione? Se sì, allora
    ** determiniamo a quale controllo si riferisce.
    */
    gdg = (IGDG *)msg->IAddress;      /* Puntatore al controllo */
    gid = gdg->GadgetID;              /* Identificativo selezione */

    /*
    ** Alcune definizioni per la stampa
    */
    :
    #define PRT_SMP(g,v) printf("Barra [%s] con valore [%d]\n", (g), (v))

    /*
    ** BLOCCO PER LA GESTIONE DEI CODICI
    */
    switch (gid)
    {
    :
    case HSMPTXT: temp = (gdg->GadgetText)->IText;
        PRT_SMP(temp, HBarValue(gdg));
        break;
    case VSMPTXT: temp = (gdg->GadgetText)->IText;
        PRT_SMP(temp, VBarValue(gdg));
        break;
    defaults : break;
    }

    return(GOAHHEAD);
}

```

Figura 5
GadgetUp().

Figura 6
HBarValue().

```

/*****
** HBarValue()          FUNZIONE          Versione 1.00
**
** Funzione fornita:  restituisce il valore logico corrente del cursore
**                    relativo al movimento orizzontale dello stesso
**
** Dati in ingresso:  bar          puntatore alla barra
**
** Dati in uscita:   lval         valore logico
**
*****/
SHORT HBarValue(bar)
    IGDG *bar;
{
    PROP *pro;
    UBAR *usr;
    USHORT pot;
    USHORT rng;

    pro = (PROP *) (bar->SpecialInfo);
    usr = (UBAR *) (bar->UserData);

    pot = pro->HorizPot;
    rng = (usr->hi).max - (usr->hi).min;

    return ((pot * rng / MAXPOT) + (usr->hi).min);
}

```


al computer, l'approssimazione non è data da un errore di misura sulla variabile **x**, bensì dagli errori introdotti nelle approssimazioni che vengono effettuate nei vari calcoli parziali. Tali approssimazioni esistono sempre, anche nel caso di calcoli con reali a doppia precisione, dato che esiste sempre un limite sul numero

di decimali gestibili da un computer. Non vi illudiate quindi di poter evitare un'analisi approfondita delle formule che usate nei vostri programmi, solo perché avete deciso di usare reali di tipo **double**. Anzi, questi sono i casi più rischiosi, dato che si forma nello sviluppatore un'illusione di precisione che prima o poi vie-

ne disattesa. Un classico è la divisione per due numeri molto vicini: qualunque sia il numero di decimali utilizzati, si tratta sempre di un'operazione da evitare, magari con tecniche speciali.

Ricapitolando.

- Analizzate sempre con attenzione le formule che usate.

La scheda tecnica: Inside 2.0

Nella scorsa puntata abbiamo visto sette schede relative alle funzioni della nuova **graphics.library**, a partire dalla **GfxAssociate()**, fino alla **OpenMonitor()**.

In questa vedremo altre sette schede, e precisamente dalla **ReadPixelArray8()** alla **TextxExtent()**.

Nella prossima vedremo le ultime schede, per poi passare ad una libreria completamente nuova: la **gadtools.library**.

ReadPixelArray8

Legge il numero di penna di ogni singolo pixel di un'area rettangolare di un certo *RastPort*, di cui sono date le coordinate dell'origine e dell'angolo diagonalmente opposto.

Vi ricordo che il numero di penna permette di ricavare dalla tavolozza dei colori associata a quello specifico *RastPort*, il colore del pixel stesso.

```

prototipo
LONG ReadPixelArray8 // Ritorna il numero di pixel letti
(
  struct RastPort *rp      , // Puntatore alla struttura RastPort
  UWORD          xstart   , // Ascissa del punto di partenza
  UWORD          ystart   , // Ordinata del punto di partenza
  UWORD          xstop    , // Ascissa del punto di arrivo
  UWORD          ystop    , // Ordinata del punto di arrivo
  UBYTE          *array   , // Ordinata del punto di arrivo
  struct RastPort *tempRp // Puntatore ad una RastPort temporanea
) ;

// Allocazione di memoria:
// -----
// array almeno (((xdelta+15)>>4)<<4)*(ydelta+1) bytes
// dove          xdelta = xstop - xstart
//               ydelta = ystop - ystart
// -----
// Assicuratevi che xstop >= xstart ed ystop >= ystart

```

Novità rispetto alla versione precedente
Si tratta di una nuova funzione.

ReadPixelLine8

Legge il numero di penna di ogni singolo pixel di un segmento orizzontale in un certo *RastPort*, di cui sono date le coordinate dell'origine ed il numero di pixel da leggere verso destra.

```

prototipo
LONG ReadPixelLine8 // Ritorna il numero di pixel letti
(
  struct RastPort *rp      , // Puntatore alla struttura RastPort
  UWORD          xstart   , // Ascissa del punto di partenza
  UWORD          ystart   , // Ordinata del punto di partenza
  UWORD          width    , // Numero di pixel da leggere
  UBYTE          *array   , // Ordinata del punto di arrivo
  struct RastPort *tempRp // Puntatore ad una RastPort temporanea
) ;

// Allocazione di memoria:
// -----
// array almeno (((width+15)>>4)<<4) bytes
// -----
// Assicuratevi che width > 0

```

Novità rispetto alla versione precedente
Si tratta di una nuova funzione.

ScalerDiv

Calcola lo stesso fattore di scala che verrebbe calcolato da **BitMapScale** a partire dalle stesse informazioni, e lo applica al valore fornito come **factor**.

Fate riferimento alla scheda sulla **BitMapScale()** riportata nella 38ª puntata, pubblicata su *MC 112* dello scorso novembre, ed alla struttura **BitScaleArgs** riportata in figura 8 della stessa puntata.

```

prototipo
UWORD ScalerDiv // Ritorna il risultato della "scalatura"
(
  UWORD factor      , // Valore da scalare
  UWORD numerator   , // In BitMapScale sarebbe XDestFactor
  UWORD denominator // In BitMapScale sarebbe XSrcFactor
) ;

// Limiti di validità:      factor      0..16383
//                          numerator, denominator 1..16383

```

Da notare che questa funzione è necessaria proprio per quei motivi riportati nel testo di questo e dell'articolo precedente, per i quali lo stesso calcolo, e cioè in questo caso:

$$\text{result} = \text{factor} * \frac{\text{numerator}}{\text{denominator}}$$

può dare risultati differenti a seconda dell'algoritmo adottato e delle approssimazioni effettuate. Se quindi avete scalato una mappa di bit, e dovete scalare in accordo un altro oggetto, non calcolate voi direttamente il fattore di scala, ma utilizzate questa funzione.

Novità rispetto alla versione precedente
Si tratta di una nuova funzione.

ScrollRaster

Muove i pixel all'interno di una certa area rettangolare di un *raster*, di un certo *delta* in direzione dell'origine del *raster* stesso.

I pixel fuori dell'area rettangolare in questione non sono interessati dallo spostamento.

- Effettuate sempre l'analisi in relazione all'arco di valori che le variabili possono assumere (sempre ben definito in un programma), e non in generale sulla funzione teorica.
- Utilizzate la tecnica della propagazione degli errori per individuare *potenziali* aree a rischio, salvo restando il fatto che

ulteriori errori sono introdotti dalle approssimazioni sui parziali.

- Tenete presente che funzioni inverse possono presentare problematiche differenti da quelle dirette, che non possono essere risolte semplicemente usando l'algoritmo *inverso*.

A volte basta solo invertire l'ordine

delle operazioni parziali, per ridurre i rischi, od anche semplicemente dividere l'espressione in più sottoespressioni più sicure. E non fate l'errore di pensare che non ci sono risultati parziali solo perché scrivete l'intera formula in una singola istruzione: il computer comunque la esegue in più passi.

prototipo

```
void ScrollRaster // Non ritorna informazioni di alcun genere
(
  struct RastPort *rp , // Puntatore alla struttura RastPort
  WORD dx , // Spostamento orizzontale
  WORD dy , // Spostamento verticale
  UWORD xmin , // Ascissa dell'origine del rettangolo.
  UWORD ymin , // Ordinata dell'origine del rettangolo
  UWORD xmax , // Ascissa dell'angolo opposto all'origine
  UWORD ymax // Ordinata dell'angolo opposto all'origine
);
```

Novità rispetto alla versione precedente

Nelle versioni 1.2 ed 1.3 del sistema operativo, se una **SUPERBITMAP** veniva fatta scorrere solo verso destra o verso sinistra, e non c'era nessun *raster* temporaneo associato alla struttura **RASTPORT**, il sistema ne allocava automaticamente una, ma non la rilasciava alla fine, né ne manteneva il puntatore. Per evitare questo era necessario associare un *raster* temporaneo grande almeno quanto **MAXBYTESPERROW** prima di chiamare lo **ScrollRaster()**. Questo problema è stato risolto con la versione 2.

Nella 2.0, inoltre, **ScrollRaster()** aggiunge le aree spostate alla lista del «danni» nel caso di finestre **SIMPLE_REFRESH**, ma non per quelle **SMART_REFRESH**.

SetFont

Imposta il tipo e le caratteristiche del testo associato ad una *RastPort*.

prototipo

```
void SetFont // Non ritorna informazioni di alcun genere
(
  struct RastPort *rp , // Puntatore alla struttura RastPort
  struct TextFont *font // Puntatore al nuovo "font" da impostare
);
// Il puntatore (font) è quello ritornato da OpenFont() o OpenDiskFont()
```

Novità rispetto alla versione precedente

La documentazione relativa alle versioni 1.2 ed 1.3, affermava che era accettato un valore **NULL** per il secondo parametro. Questo è da sconsigliare, dato che può addirittura fare andare in *guru* il sistema. Non farlo comunque mai con la 2.0.

Nella versione 2, le seguenti varianti non sono più supportate direttamente:

- **tf_CharSpace** nullo, e **tf_CharKern** non nullo,
- **tf_CharSpace** non nullo, e **tf_CharKern** nullo,
- **tf_CharSpace** e **tf_CharKern** nulli, e la seconda componente di **tf_CharLoc** maggiore di **tf_XSize**.

Il sistema modifica automaticamente le caratteristiche dei caratteri, nel caso una variante non valida venga richiesta.

Attenzione. Se si chiama la **SetFont()** con un *font* interno, non

ottenuto cioè da **OpenFont()**, ad esempio, verranno persi 24 byte di memoria a meno che non si chiami prima la **StripFont()** (vedi la scheda successiva).

StripFont

Rimuove la parte estesa di un *font* (**tf_Extension**) dallo stesso.

prototipo

```
VOID StripFont // Non ritorna informazioni di alcun genere
(
  struct TextFont *font // Puntatore al "font" da ripulire
);
```

Novità rispetto alla versione precedente.

Si tratta di una nuova funzione.

TextExtent

Determina l'estensione di uno specifico *font* in un certo *raster* in modo più completo di quanto faccia la **TextLength()**.

prototipo

```
void TextExtent // Non ritorna alcuna informazione "come funzione"
(
  struct RastPort *rp , // Puntatore alla struttura RastPort
  STRPTR string , // Stringa da misurare
  WORD count , // Numero di caratteri nella stringa
  struct TextExtent *textExtent // Struttura che riceve il risultato
);
```

La struttura **TextExtent** è riportata in figura 11, insieme alle indicazioni su come essa è riempita dalla **TextExtent()**.

Novità rispetto alla versione precedente.

Si tratta di una nuova funzione.

```
struct TextExtent
{
  UWORD te_Width ; // Lunghezza della stringa, in pixel
  UWORD te_Height ; // Altezza del "font", in pixel
  struct Rectangle te_Extent ; // Rettangolo del testo. Vedi sotto...
};

// La lunghezza in pixel è la stessa che riporterebbe TextLength()
// L'altezza è quella riportata in tf_YSIZE

/*
** Il rettangolo contenente il testo è misurato relativamente alla
** linea "guida" del testo, o base:
**
** te_Extent.MinX Solitamente 0
** te_Extent.MinY Sempre come - tf_Baseline
** te_Extent.MaxX Solitamente te_Width - 1
** te_Extent.MaxY Sempre come te_Height - tf_Baseline - 1
**
*/
```

Figura 11 - Struttura *TextExtent*.

```

/*****
** VBarValue()          FUNZIONE          Versione 1.00      **
**                                                              **
** Funzione fornita:  restituisce il valore logico corrente del cursore
**                   relativo al movimento verticale dello stesso
**                                                              **
** Dati in ingresso:  bar      puntatore alla barra
**                                                              **
** Dati in uscita:   lval     valore logico
**                                                              **
*****/
SHORT VBarValue(bar)
IGDG *bar ;
{
  PROP *pro ;
  UBAR *usr ;
  USHORT pot ;
  USHORT rng ;

  pro = (PROP *) (bar->SpecialInfo) ;
  usr = (UBAR *) (bar->UserData) ;

  pot = pro->VertPot ;
  rng = (usr->vi).max - (usr->vi).min ;

  return ((pot * rng / MAXPOT) + (usr->vi).min) ;
}
    
```

Figura 7 - VBarValue().

Consideriamo $y = \frac{x}{50 - (x/3)}$ con $x = 148$

x	148
148 / 3	49
50 - 49	1
148 / 1	148
cioè y = 148		

x	148
148 / 3	49.(3)
50 - 49.(3)	0.(6)
148 / 0.(6)	222
cioè y = 222		

La formula opposta è $x = \frac{50y}{1 + (y/3)}$ con $y = 222$

y	222
222 / 3	74
1 + 74	75
50y / 75	148
cioè x = 148		

y	222
222 / 3	74
1 + 74	75
50y / 75	148
cioè x = 148		

Figura 8 - Calcoli diretti ed inversi.

FORMULA $y = \frac{x}{50 - (x/3)}$

CALCOLO DELLA PROPAGAZIONE DELL'ERRORE

$$dy = d\left(\frac{x}{50 - (x/3)}\right) = d(x) \cdot \frac{1}{50 - (x/3)} + x \cdot d\left(\frac{1}{50 - (x/3)}\right)$$

RISULTATO $dy = \frac{50}{(50 - (x/3))^2} dx$ Per $x = 148$ si ha $dy = 112.5 dx$

Figura 9 - Propagazione dell'errore (formula diretta).

FORMULA $x = \frac{50y}{1 + (y/3)}$

CALCOLO DELLA PROPAGAZIONE DELL'ERRORE

$$dx = d\left(\frac{50y}{1 + (y/3)}\right) = d(y) \cdot \frac{50}{1 + (y/3)} + y \cdot d\left(\frac{50}{1 + (y/3)}\right)$$

RISULTATO $dx = \frac{50}{(1 + (y/3))^2} dy$ Per $y = 222$ si ha $dx = 0.009 dy$

Figura 10 - Propagazione dell'errore (formula inversa).

Conclusioni

Terminiamo così la parte dedicata ad i controlli di Intuition. Per dare un po' di respiro alla rubrica, ho deciso di interrompere per il momento la serie dedicata a questo componente del sistema operativo dell'Amiga per affrontare in modo sistematico un nuovo argomento, e cioè quello delle mappe di tastiera e della gestione dei caratteri.

Sono passate ben **quaranta** puntate e più di tre anni da quando questa rubrica è apparsa per la prima volta su *MCmicrocomputer*, e tuttavia non posso fare a meno di rimanere impressionato nel constatare quante cose ancora questa stupenda macchina nasconda al suo interno: file IFF, multi-tasking, animazio-

ne, musica. E le nuove versioni continuano ad aggiungere sempre di più. A conoscerla a fondo, l'Amiga è sicuramente una macchina di una potenza eccezionale, capace di dare al programmatore soddisfazioni che ben pochi sistemi sono in grado di offrire.

Proprio per questo, ho cercato di impostare questa serie di articoli con lo scopo di cercare di dare a tutti la possibilità di venire a conoscenza dei vari componenti del sistema da una parte, cercando nel contempo di approfondire gli stessi al di là di quello che la documentazione ufficiale od i vari libri disponibili sul mercato sono in grado di dare.

I due obiettivi sono ovviamente in contrasto l'uno con l'altro, dato che il primo porterebbe a sviluppare nuovi argomenti

ad ogni puntata, mentre il secondo comporta l'approfondimento di ogni argomento, cosa che richiederebbe più puntate per argomento. La soluzione da me adottata è stata quella di approfondire un certo argomento, integrandolo con esercizi, proposte, persino piccole provocazioni, per poi passare ad un altro, una volta che sia stata data una certa *spinta* sul precedente, e lasciare poi al lettore la decisione se continuare su quella strada, o passare ad un argomento per lui o lei più interessante.

Spero che questa *formula* abbia trovato il vostro gradimento. Se no, fatemelo sapere... e buon 1992 a tutti. MB

Dario de Judicibus è raggiungibile tramite MC-link alla casella MC2120.

miniPC[®] ELOX[®]

IL PERSONAL COMPUTER IN 580 GRAMMI

DIMENSIONI 230x110x29 mm / IBM PC/XT COMPATIBILE
RAM 640 K - ROM 640 K / MEMORIA DI MASSA = MEMORY CARD FINO A 2 MB (8 MB NEL 1992)
È POSSIBILE UTILIZZARE SOFTWARE STANDARD MS-DOS

PREZZO PUBBLICO
Lit. 980.000
IVA ESCLUSA



Schermo LCD bianco e nero,
80 colonne per 25 righe,
640x200 punti, CGA

Tastiera 79 tasti
tipo dattilo

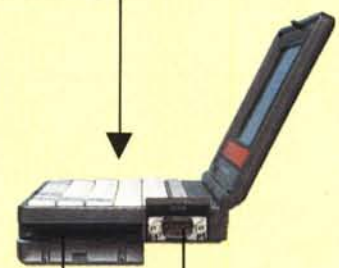
Regolatore intensità schermo



Porta per Disk Drive

Porta parallela standard
per stampante

Memory Card
L'HARD DISK
DEL FUTURO



Slot per Memory Card

Porta seriale RS232C standard

CARATTERISTICHE DI SISTEMA

CPU: 80C88 (CMOS) IBM PC/XT COMPATIBILE
LOW-POWER: CHIP AVANZATO SUPER INTEGRATO

MEMORIA

ROM: 640 K
RAM: 640 K
MEMORY CARD SLOT PER UTENTE (CARD, RAM, ROM, OTP)

SOFTWARE • ROM-DOS: DR-DOS V.5.0 • SOFTWARE APPLICATIVO INCLUSO:

- FILE LINK (per trasferire dati e programmi da/per altri Computer) - WORD PROCESSOR - DOS FILE - VIEWMAX - SCHEDULE REMINDER (con allarme) - CALENDARIO - DIARIO DIGITALE - NOTE PAD - DIARIO CARD / INDEX CARD - NAME CARD / INDEX CARD - CALCOLATRICE 12 DIGIT

ALIMENTAZIONE

BATTERIA "AA" x 4
DURATA DELLA BATTERIA CIRCA 20 ORE

DOTAZIONE

4 PILE ALCALINE AA
BORSA
ALIMENTATORE RETE 220 VOLT 50 Hz

▼ IL COMPLETO



▼ L'UFFICIO NELLA 24 ORE



▼ IN VIAGGIO



EVART S.r.l. - Via Rossetti, 17 - 20145 Milano - Tel. 02/4814619 - Fax 02/48006714

RICHIEDERE IL RIVENDITORE PIÙ VICINO

**Il software MS-DOS, Amiga e Macintosh
di Pubblico Dominio e Shareware
distribuito da**



**in collaborazione con
Microforum**

Questo software non può essere venduto a scopo di lucro ma solo distribuito dietro pagamento delle spese vive di supporto, confezionamento, spedizione e gestione del servizio. I programmi classificati Shareware comportano da parte dell'utente l'obbligo morale di corrispondere all'autore un contributo indicato al lancio del programma.

CODICE	TITOLO&DESCRIZIONE	REC. HARDWARE	CODICE	TITOLO&DESCRIZIONE	REC. HARDWARE	CODICE	TITOLO&DESCRIZIONE	REC. HARDWARE
MSDOS								
COMUNICAZIONE								
COM/01	ONE TO ONE	mc104	GIO/16	EGA GOLF Gioco del Golf	EGA/VGA	SPD/02	EXPRESS-CALC	mc104
COM/02	PROCOMM Noto programma di comunicazione	Hard disk	GIO/17	EGA TREK Star Trek	EGA/VGA	SPD/03	EZ-SPREADSHEET Calcoli di budget	
COM/03	OMEGA LINK	mc106	GIO/18	JOUST VGA Gioco da bar	VGA	SPD/04	INSTACALC	mc107
COM/04	BACKCOMM	mc103	GIO/19	MINER VGA	mc104 VGA	SPD/05	QUEBECALC Spreadsheet 3D	
COM/05	ZIP	mc110	GIO/21	MOSAIX Puzzle	VGA	UTILITY		
COM/06	FOSSIL DRIVER & .TPU	mc110	GIO/22	OTHELLO EGA	mc103 EGA/VGA	UTI/01	PC-DESK-TEAM	mc107
COM/07	MAXIHOST	mc110	GIO/23	POKER SOLITAIRE Poker da soli	EGA/VGA	UTI/02	HARD DISK UTILITIES Per gestire l'Hard Disk	Hard Disk
DATABASE			GIO/24	QUATRI Tetris con Bombe ecc.	EGA/VGA	UTI/03	DOS HELP	mc104
DBS/01	EASY LABELS Per creare etichette		GIO/25	SHARKS Giocate ai sommozzatori	EGA/VGA	UTI/04	DISK SPOOL II	mc103
DBS/02	VIDEO DATABASE	mc105 Hard disk	GIO/26	SLOT EGA Slot Machine	EGA/VGA	UTI/05	LOCKTITE Protegge i file con password	
DBS/03	HOME MANAGER DataBase, calcolatrice e calendario	Hard disk	GIO/27	BASSTOUR	EGA/VGA	UTI/07	LHARC	mc105
DBS/04	MAIL-MONSTER	mc103	GIO/28	BLACKJACK Gioco da Casinò	EGA/VGA	UTI/08	ARJ	mc105
DBS/05	MAKE MY DAY Per organizzare il lavoro		GIO/29	GALACTIC BATTLE Clone di Invaders con sonoro	EGA/VGA	UTI/09	LZEKE	mc105
DBS/06	PC-FILE+	mc106	GIO/30	HOUSE OF HORRORS Casa degli orrori	EGA/VGA	UTI/10	DIET	mc105
DBS/07	TASK MASTER Projet Plaining		GIO/31	NOID Consegnate la pizza all'ultimo piano	EGA/VGA	UTI/11	PKLITE	mc105
DBS/08	RELIANCE MAILING LIST Mailing per associazioni culturali		GIO/32	PINBALL EGA Super Flipper	EGA/VGA	UTI/12	NEWSPACE	mc105
DBS/09	DMS	mc107	GIO/33	STARDEF Missili distruggono la terra...		UTI/13	CATDISK	mc105
DBS/10	ARCHIVIO PARROCCHIALE	mc109	GIO/34	MAHJONG EGA Gioco di società orientale	EGA/VGA	UTI/14	POINT&SHOOT	mc105
EDUCATIVO			GIO/35	MR.SPOCK	mc105 EGA/VGA	UTI/15	SHEZ	mc106
EDU/01	ABC FUN KEYS	mc103	GIO/36	MONUMENTS OF MARS	mc106	UTI/16	ZZAP	mc106
EDU/02	COMPUTER TUTOR Auto-apprendimento del computer		GIO/37	PHARAOH'S TOMB	mc106	UTI/17	GUARDIAN ANGEL	mc107
EDU/03	PC-FASTYPE	CGA	GIO/38	POKER	mc107 EGA/VGA	UTI/18	STORE	mc107
EDU/04	Imparare professionalmente ad usare la tastiera GEOBASE ARCH. GEOGRAFICO		GIO/39	NIM	mc108 CGA	UTI/19	TXT	mc107
GIOCO			GIO/40	TESORI	mc108 CGA	UTI/20	xSET	mc108
GIO/02	2BIT POKER Poker Canadian	EGA/VGA	GIO/41	TOMBOLA	mc108 CGA	UTI/21	ZAPDIR	mc108
GIO/03	ASTRO BLASTER Clone di Space Invaders	PC-AT/286	GIO/42	SMILE	mc109 VGA	UTI/22	UTILITY COLLECTION	mc109
GIO/04	ALDO'S ADVENTURE	mc103 EGA/VGA	GIO/43	CHINESE SOLITARIE	mc111 VGA	UTI/23	DIR	mc109
GIO/05	CAESAR Strategia	BASIC+EGA/VGA	GIO/44	TRETRIX	mc111 VGA	UTI/24	CLEANUP	mc111
GIO/07	CLONE INVADERS Clone di Space Invaders		GIO/45	SICHUAN	mc112 VGA	UTI/25	SAB DISKETTE UTILITY	mc111
GIO/08	EGAI NT	mc104 EGA/VGA	GIO/46	EGAWALLS	mc113 EGA	UTI/26	TIF2GRAY	mc111
GIO/09	PC-JIGSAW Puzzle		GIO/47	GRID HER	mc113 VGA	UTI/27	FILLDISK	mc111
GIO/10	MAHJONG Solitario orientale	EGA/VGA	GIO/48	BANDIERE!	mc114	UTI/28	ORASCO	mc111
GIO/11	SUPER PINBALL Super Flipper		GIO/49	PETWORLD	mc114	UTI/29	XDIR	mc111
GIO/12	ARK Clone di Arkanoid	EGA/VGA	GIO/50	FORZA4	mc114	UTI/30	WINCOMMANDER	mc112
GIO/13	BANYON WARS Strategia	EGA/VGA	GRAFICA			UTI/31	MOUSE FORMATTER	mc112
GIO/14	CAPTAIN COSMIC Gioco grafico	EGA/VGA	GRF/01	FINGER PAINT Programma di disegno		UTI/32	WINZIP	mc112
			GRF/02	PC-KEY-DRAW	mc107 CGA	UTI/33	MOUSE EDITOR	mc113
			GRF/03	H&P CALENDAR	mc103	UTI/34	DEPURA	mc113
			GRF/04	PC-DEMO SYSTEM	mc105	UTI/35	DISK FATTER	mc113
			GRF/05	GRAPHICWORKSHOP	mc106	UTI/36	POWER DOS	mc114
			GRF/06	SOLAI & TRAVI	mc112	UTI/37	SIM_LIB	mc114
			GRF/07	GOSTPAINT	mc112	UTI/38	UTILITY PC	mc114
			SPREADSHEET			VARIE		
			SPD/01	AS-EASY-AS	mc103	VAR/01	COMPOSER Per suonare al computer e stampare lo spartito	
						VAR/02	CHECK-MATE Controllo delle finanze personali	
						VAR/03	PIANO-MAN	mc104
						VAR/04	BARTENDER Tutti i cocktail	mc103
						VAR/05	DIET DISK La dieta al computer	
						VAR/06	ELEMENTARY C Per programmatori in C	
						VAR/07	RECIPES	mc104
						VAR/08	PERSONAL C COMPILER	mc105
						VAR/09	MOUSE.TPU & NEWEXEC	mc106

CODICE	TITOLO&DESCRIZIONE	REC. HARDWARE
VAR/10	TSR, PRINT & GESTECC	mc106
VAR/11	ARIANNA	mc106
VAR/12	TOTOPROJET	mc108 CGA
VAR/13	COVER	mc108
VAR/14	CODICE FISCALE	mc109 Hard disk
VAR/15	FLIGHT	mc109
VAR/16	DIZIONARIO INFORMATICO	mc109
VAR/17	ITALIA90	mc110
VAR/18	TATA-BIGNOMIX UTILITY	mc110
VAR/19	QUICK BASIC ROUTINES	mc110
VAR/20	MICROGESTS	mc113
VAR/21	CALCOLO INDICE ELO	mc113
VAR/22	MENU	mc113
VAR/23	PROMETEO	mc114

WORDPROCESSOR

WPR/01	W.P.FOR CHILDREN Per insegnare ai bambini il WP	
WPR/02	FREEWORD	mc103
WPR/03	PC-WRITE	mc106
WPR/04	THESAURUS PLUS Sinonimi in inglese (TSR)	
WPR/05	GALAXY	mc104
WPR/06	EDITOR	mc110
WPR/07	NOTEBOOK	mc112
WPR/08	WORDY	mc113
WPR/09	VIX	mc114

AMIGA

COMUNICAZIONE

AMCO/01	AMIPAC	mc110
AMCO/02	FC FREE COMMUNICATION	mc113

DATABASE

AMDB/01	BADGER	mc113
---------	--------	-------

GIOCO

AMGI/02	WELLTRIX	mc105
AMGI/03	SYS	mc105
AMGI/04	SCOPONE SCIENTIFICO	mc108
AMGI/05	LA FINE DI UN TIRANNO	mc109
AMGI/06	LA PANTERA SIAMO NOI	mc109
AMGI/07	MEGABALL	mc110
AMGI/08	REVERSI	mc114

GRAFICA

AMGR/01	PRINTSTUDIO	mc104
AMGR/02	TEXTPOINT	mc105
AMGR/03	SCREENX	mc105
AMGR/04	SETPAL	mc105
AMGR/05	FREEPAINT	mc113
AMGR/06	LABEL MAKER	mc114
AMGR/07	PICTSAVER	mc114

SPREADSHEET

AMSP/01	SPREAD	mc104
AMSP/02	EQUATIONWRITER	mc110

UTILITY

AMUT/01	MACH III	mc104
AMUT/02	RULER	mc104
AMUT/03	HEX	mc104
AMUT/04	MOM	mc104
AMUT/05	CB	mc104
AMUT/06	ZETAVIRUS	mc104
AMUT/07	DIRMASTER	mc105
AMUT/08	KDC	mc105
AMUT/09	XCOPYIII	mc105
AMUT/10	CD2TAPE	mc105
AMUT/11	BBS & LOG	mc106
AMUT/12	UTILITIES	mc106
AMUT/13	VIEW80 II	mc106
AMUT/14	MATCALC	mc106
AMUT/15	ICONMASTER	mc106
AMUT/16	HERMIT	mc106
AMUT/17	TURBO IMPLODER	mc106
AMUT/18	FONTSPRINTER	mc107
AMUT/19	SVD	mc107
AMUT/20	MC-PROGRAMS	mc107
AMUT/21	CHP&SAVE-PREFS	mc107
AMUT/22	CIDITEIP	mc108
AMUT/23	DISKEDITOR	mc108
AMUT/24	5 UTILITY	mc108
AMUT/25	OROLOGIO PARLANTE	mc108
AMUT/26	LSLAB	mc110
AMUT/27	DIRWORK	mc111
AMUT/28	SCREENMOD	mc111

CODICE	TITOLO&DESCRIZIONE	REC. HARDWARE
AMUT/29	SYSINFO	mc111
AMUT/30	SUPERDUPER	mc111
AMUT/31	PRFONT	mc113
AMUT/32	TG	mc113
AMUT/33	ICONS	mc113
AMUT/34	TURBOGIF	mc113

VARIE

AMVR/01	FRACTUS	mc108
AMVR/02	RUBRICA, DACIA & GESTFAT	mc109
AMVR/03	FUNZ3D	mc109
AMVR/04	PLAYSMUS	mc110
AMVR/05	MULTI PLAYER	mc111
AMVR/06	DRAWMAP	mc111
AMVR/07	TOTAMIGA	mc112
AMVR/08	AUTO	mc112
AMVR/09	SOUNDMASTER	mc112
AMVR/10	AMIGA L8	mc112
AMVR/11	FRACTAL	mc112
AMVR/12	SPECTROGRAM	mc114
AMVR/13	CHEMESTHETICS	mc114
AMVR/14	DAY2DAY	mc114

MACINTOSH

COMUNICAZIONE

MICO/01	RED RYDER	mc110
---------	-----------	-------

EDUCATIVO

MIED/01	KID PIX	mc107
MIED/02	NUMBER TALK	mc107
MIED/03	ALPHA TALK	mc107

GIOCO

MIGI/01	STELLA OSCURA	mc106
MIGI/02	PARARENA	mc106
MIGI/03	VIDEO POKER FOR FUN	mc106
MIGI/04	SPACE STATION PHETA	mc106
MIGI/05	STRATEGO	mc106
MIGI/06	THE LAWNZAPPER	mc107
MIGI/07	MACTRIS	mc107
MIGI/08	CANFIELD	mc107
MIGI/09	YAHTZEE	mc108
MIGI/10	GLIDER	mc108
MIGI/11	MACNINJA	mc108
MIGI/12	GLIPHA	mc108

CODICE	TITOLO&DESCRIZIONE	REC. HARDWARE
MIGI/13	MONOPOLY	mc109
MIGI/14	GOLF	mc109
MIGI/15	WHEEL	mc109
MIGI/16	GUNSHY	mc109
MIGI/17	MEGARIDS	mc110
MIGI/18	SHUFFLEPUCK	mc110
MIGI/19	CRIMINALS	mc111
MIGI/20	SQUIX	mc112
MIGI/21	HOTEL CAPER	mc112

GRAFICA

MIGR/01	CALENDAR MAKER	mc106
---------	----------------	-------

SPREADSHEET

MISP/01	BIPLANE	mc112
---------	---------	-------

STACK

MISK/01	FOOD 1	mc111
MISK/02	BUSINESS 1	mc111
MISK/03	SOUND 1	mc111

UTILITY

MIUT/01	OLIVER'S BUTTONS	mc107
MIUT/02	POPCHAR	mc107
MIUT/03	RAMDISK	mc108
MIUT/04	SCROLL2	mc109
MIUT/05	DECK EDITOR	mc109
MIUT/06	BANNER MAKER	mc110
MIUT/07	SPEEDOMETER	mc110
MIUT/08	LOODLE	mc112
MIUT/09	FAST FORMAT	mc112
MIUT/10	SOUND MASTER	mc112
MIUT/11	STUFFIT CLASSIC	mc112
MIUT/12	DISKDUPL+	mc114
MIUT/13	DTPPRINTER	mc114
MIUT/14	FOLDER FROM HELL	mc114
MIUT/15	NUMBERCRUNCH	mc114
MIUT/16	PASTE-IT	mc114
MIUT/17	SAVE A TREE	mc114
MIUT/18	MACBINARY	mc114

VARIE

MIVR/01	RIDICOLO	mc108
MIVR/02	ELIZA	mc109
MIVR/03	HYPERSTAR	mc113

Compilare e spedire a: MCmicrocomputer

Desidero acquistare il software di seguito elencato al prezzo di **L. 8.000 a titolo (ordine minimo: tre titoli)**. Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla: Technimedia srl, Via Carlo Perrier 9, 00157 Roma.

dischetti da	<input type="checkbox"/> 3.5"	<input type="checkbox"/> 5.25"
Codici:	_____	

Totale dischi <input type="checkbox"/> x 8.000=Lire _____		
Manuali in italiano:		
<input type="checkbox"/> TSPD/01 AS EASY AS	<input type="checkbox"/> TUTI/01 HARD DISK UTILITIES	
<input type="checkbox"/> TVAR/02 CHEKMATE	<input type="checkbox"/> TWPR/05 GALAXY	
Totale manuali <input type="checkbox"/> x 8.000=Lire _____		

Nome e Cognome _____

Indirizzo _____

CAP/Città _____

Telefono _____

MCmicrocomputer non offre alcuna garanzia e non si assume alcuna responsabilità sugli eventuali danni diretti o indiretti derivanti dall'utilizzo del software distribuito



NEWEL[®] srl

NUOVO CATALOGO PC-MS-DOS & COMPATIBILI/DICEMBRE 1991

ATTENZIONE!!! TUTTI I PREZZI SONO IVA 19% COMPRESA!!!

La NEWEL dispone di un completo assortimento di sistemi di elaborazione, la cui linea di assemblaggio è ubicata in Italia: facilitando così notevolmente qualsiasi problema di assistenza con estrema celerità. Ciascun sistema è formato da pezzi di ottima qualità che sottolineano il grado di affidabilità. Ciascun prodotto viene venduto con una garanzia integrale di 12 mesi.

L'assortimento consiste in 4 modelli: il desktop, il mini Tower, il Big Tower ed il Super Torre. Disponibili in varie configurazioni con nuovissima piastra madre di ottima qualità basata sui microprocessori 80286, 80386 e i nuovissimi 80486. Tutte le nostre macchine vengono fornite senza sovrapprezzo alcuno nelle seguenti configurazioni base: cabinet Desktop o mini tower, 1 MB Ram, Hard Disk 1 Drive 3" 1/2 (1,44MB) o 5"1/4 (1,2MB), una scheda grafica VGA 256K Ram video 640 x 480 256 colori, 800 x 600 16 colori. Un controller per 2 drive e 2 Hard Disk, una scheda Multi I/O con 2 seriali e 1 parallela, una tastiera estesa 102 tasti italiana e USA, manuali delle schede, zoccolo per coprocessore 80287. Le nostre macchine vengono fornite con Hard Disk formattato, pronte all'uso! Ecco qui a seguire alcuni esempi di configurazioni.



NUOVO PC

386 SX20

HARD DISK 60MB

2MB RAM

PORTATILE NOTEBOOK

+BORSA £3.490.000

DISPONIBILI: MEMORIE IN OFFERTA SPECIALE!

OFFERTA DEL MESE!

PC-NOTEBOOK 286 COMMODORE

piccolissimo come una agenda!!! Meno di 3 kg.

286 12 MHz schermo VGA retroilluminato
disk drive 3" 1/2 (1,44 MB) +
hard disk 20 MB (23 ms.)

1 MB RAM espandibile a 5 possibilità attacco
monitor esterno borsa,
batterie e alimentatore

in offerta NEWEL a **L. 2.490.000**
DOS & MANUALI IN ITALIANO

TUTTI CON MOUSE IN OMAGGIO E CON DR-DOS 5,0 ORIGINALE IN ITALIANO

286 12 MHz.

L. 990.000

1 MB on board espandibile a 4 - 0 Wait States + HD 20 + 1 Drive 3" 1/2 (1,44 MB) + Scheda Grafica VGA 800 x 600 + 2 Seriali e 1 Parallela + Tastiera estesa 102 tasti + Monitor monocromatico VGA + Garanzia 12 mesi. Il tutto Iva 19% compresa

286 16 MHz

L. 1.090.000

1 MB Ram on board espandibile a 4 - 0 Wait States + HD 40 MB AT-BUS 25 ms. + 1 Disk Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificati di garanzia 12 mesi.

386-SX 20 MHz

L. 1.390.000

1 MB Ram on board espandibile a 8 - 0 Wait States + HD 40 MB AT-BUS 25 ms. + 1 Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificato di garanzia 12 mesi.

386-25 MHz

L. 1.890.000

4 MB Ram on board espandibile a 8 - 0 Wait States + HD 40 MB AT-BUS 25 ms. + 1 Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificati di garanzia 12 mesi.

ATTENZIONE!

Possibilità Hard Disk 130 MB aggiungendo L. 500.000

386-33 MHz + 64 KB CACHE M. L. 2.190.000

4 MB Ram on board espandibile a 8 - 0 Wait States + HD 40 MB AT-BUS 25 ms. + 1 Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificati di garanzia 12 mesi.

ATTENZIONE!

Possibilità HD 130 MB aggiungendo L. 500.000

386-40 MHz + 64 KB CACHE M. L. 2.390.000

4 MB Ram on board espandibile a 8 - 0 Wait States + Hard Disk 40 MB AT-BUS 25 ms. + 1 Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificati di garanzia 12 mesi.

ATTENZIONE!

Possibilità Hard Disk 130 MB aggiungendo L. 500.000

486 - SX 20

L. 2.490.000

4 MB Ram on board espandibile a 8 - 0 Wait States + Hard Disk 40 MB AT-BUS 25 ms. + 1 Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificati di garanzia 12 mesi + Coprocessore matematico 487 fornito di serie. ATTENZIONE! NEI *Possibilità Hard Disk 130 MB* aggiungendo L. 500.000

486-33 MHz + 64 KB CACHE M. L. 3.290.000

4 MB Ram on board espandibile a 8 - 0 Wait States + Hard Disk 40 MB AT-BUS 25 ms. + 1 Drive + Scheda Grafica VGA 256K (800 x 600) + 2 Seriali + 1 Parallela + Tastiera estesa 102 tasti + Cavi & Manuali tecnici delle schede + Certificati di garanzia 12 mesi + Coprocessore matematico 487 fornito di serie. ATTENZIONE!

Possibilità Hard Disk 130 MB aggiungendo L. 500.000

CONFIGURAZIONI AGGIUNTIVE A PREZZI SPECIALI VALIDI SOLO PER ACQUISTI DEI PC

Con secondo Drive	aggiungere	L. 130.000
Con scheda super VGA 1024 x 768 (1 MB) 256 col.	*	L. 200.000
Con Q-tec Mouse 100% Microsoft compatibile	*	L. 50.000
Con Hard Disk 130 MB 19 ms. AT-BUS	*	L. 500.000
Con Hard Disk 340 MB 15 ms. AT-BUS	*	L. 1.900.000
Per ogni MB di Ram in più	*	L. 99.000

NUOVI PRODOTTI OFFERTA

MUSIC CARD (AD-LIB comp.)

L. 198.000

SOUND BLASTER

L. 298.000

SOUND BLASTER PROFESSIONAL

L. 498.000

SCHEDA TELEVIDEO PC L. 299.000

SCHEDA GRAFICA VGA 256K

L. 99.000

RAM MOD. SIMM 1MB 70ns

L. 99.000

MONITOR VGA-MONO L. 249.000

MONITOR VGA COLOR L. 549.000

MONITOR VGA COLOR 1024x768

L. 699.000

MONITOR NEC 3FG color Multisync

1024x768 Novità L.1.190.000

NUOVO IBM NOTEBOOK

386 sx HD 40 MB L.3.500.000

OFFERTE STAMPANTI

PPC 001 stampante Mannesmann MT-81 (9 aghi 120 cps.)	L. 299.000
PPC 002 stampante Commodore MPS 1230 (9 aghi 120 cps.)	L. 299.000
PPC 003 stampante Commodore MPS 1270 (Inkjet, silenziosissima)	L. 299.000
PPC 004 stampante STAR LC-20 "Super" (9 aghi alta Qualità)	L. 360.000
PPC 005 stampante STAR LC-200 colori (9 aghi alta Qualità)	L. 499.000
PPC 006 stampante CITIZEN 124D (straordinaria 24 aghi)	L. 599.000
PPC 007 stampante STAR LC-24/200 (24 aghi multifonti)	L. 699.000
PPC 008 stampante STAR LC-24/200 colori (24 aghi a colori new!)	L. 790.000
PPC 009 stampante NEC P20 Plus (24 aghi alta Qualità)	L. 690.000
PPC 010 stampante NEC P30 Plus (24 aghi 132 colonne)	L. 890.000
PPC 011 stampante NEC P60 Plus (24 aghi 360x360 punti)	L. 1.100.000
PPC 012 stampante NEC P70 Plus (come sopra ma 132 col.)	L. 1.350.000
PPC 013 kit colore per trasformare NECX P60 & P70 a colori	L. 265.000
PPC 014 stampante laser varie marche come Star, HP, Epson da	L. 1.950.000

RICHIEDI IL CATALOGO

TUTTI I MARCHI CITATI SONO PROPRIETA' DEGLI AVENTI DIRITTO

20155 MILANO - via Mac Mahon, 75 - Telefono negozio 02/32.34.92 Telefono uffici 02/32.70.226 - Telefax 24 ore 02/33.00.00.35 - Ufficio Spedizioni 33000036

QUANDO OLTRE AL PREZZO C'E' LA QUALITA' & L'ASSISTENZA

TUTTI I NOSTRI PREZZI SONO IVA 19% COMPRESA

IL PIU' VASTO ASSORTIMENTO DI ACCESSORI PER IL TUO COMPUTER

PAGAMENTI PERSONALIZZATI RATEALI
SENZA CAMBIALI
FINO A 3 ANNI

SU TUTTI I COMPUTER AMSTRAD SCONTO 10%

RIVENDITORE AUTORIZZATO



ATTENZIONE!

Ricorda, un computer non può essere acquistato ovunque, ma solo presso una organizzazione specializzata che è in grado di consigliare e assistere il cliente prima e soprattutto dopo la vendita!!!

La NEWEL è una società specializzata che opera nel settore da oltre 10 anni. Noi trattiamo tutte le migliori marche, e quindi siamo in grado di offrirvi il computer che si addice di più alle vostre esigenze.

Da noi non trovi soltanto il computer ma anche tutto ciò che ti può servire successivamente, hardware & software. Pensaci prima di comprare un computer!