La grafica è sempre stata il sogno di molti programmatori, nel senso che anche programmatori esperti, nel momento di mettere le mani su un programma «pesantemente» grafico, finiscono con l'imbattersi in un gran numero di problemi sempre nuovi ed affascinanti. A volte serve la velocità, altre volte occorre trovare un equilibrio grafico particolare, altre volte occorre immagazzinare grandi quantità di immagini con i problemi che ne conseguono. Ecco quindi perché accolgo sempre volentieri su queste pagine qualsiasi strumento (e in qualsiasi linguaggio) possa aiutare a risolvere, e spero a comprendere meglio, i problemi tipici della grafica ad alto livello. Il programma presentato permette di gestire delle Shape in movimento e dei fondali, anch'essi dinamici: con notevole semplicità e soprattutto con un listato breve e pubblicabile. Non è però possibile pubblicare le immagini di prova né i programmi di Edit che ne permettono la creazione, materiale che comunque troverete sul disco in vendita presso la redazione

Shape Tool-Kit

di Giancarlo Del Sordo

Shape non è un Cad. Da un lato non mi voglio nemmeno permettere di fare una brutta copia di alcuni programmi in circolazione da diversi anni, dall'altro non è nemmeno mia intenzione creare un editor di disegni tecnici.

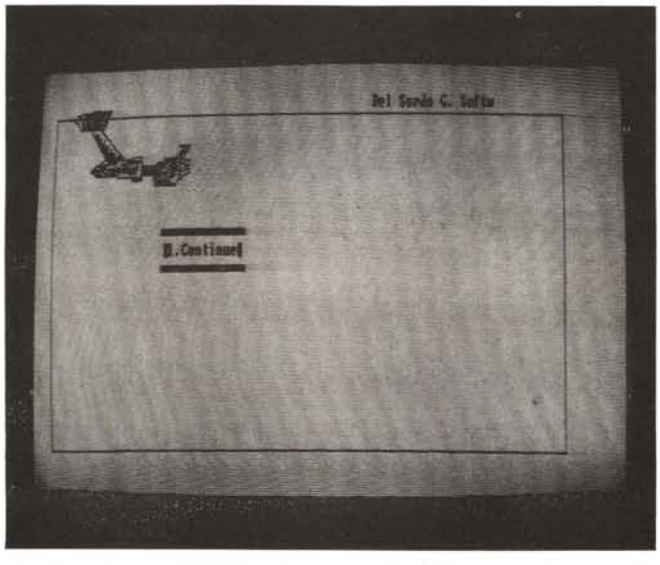
Anzi, se non si parte da questi presupposti si avranno non poche delusioni: mia intenzione era quella di trasferire il concetto di background e sprite anche al non poco spoglio Turbo Pascal.

Con una settimana di applicazione con Shape si possono ottenere degli effetti spettacolari con il minimo della fatica.

Creare giochi di buona grafica non sarà più un miraggio per il povero Turbo e, cosa più importante, richiamando una direttiva di include scritta in Pascal, quindi suscettibile a modifiche semplici di qualsiasi tipo.

In qualsiasi momento della utilizzazione deve essere presente nel drive B: il disco con i file creati, per questo consiglio di fare una bella copia del file di esempio del dischetto (contrassegnati da estensione .?CM) e di sistemarlo prima di qualsiasi operazione in B: (chi non ha il disco B: deve fare una ASSIGN B=A oppure modificare SHAPE.SYS.

Altro appunto, anzi la nota dolente di tutta la carriola, è il fatto che Shape lavora solo su CGA.



Qui non si paga tanto in termini di risoluzione quanto alla mancanza di colori. I migliori risultati si possono ottenere con una CGA e un monitor monocromatico dove il sacrificio è minimo.

Creati i vari disegni e sfondi e memorizzatili in vari file, si possono richiamare da Turbo usando i comandi aggiuntivi di SHAPE.SYS che in effetti ha una struttura molto elementare ma che può mostrare ai più esperti l'organizzazione e la gestione dei dati tanto da permettergli di modificarlo con estrema facilità. Il file gestisce il movimento e la selezione dei vari sprite e background.

EDIT1

Indubbiamente è l'editor più importante: serve per le creazioni di nuove figure o forme da considerarsi come sprite o parti di background.

Per dirla in breve può lavorare a piani indipendenti, sfruttare fino a 5 griglie, zoomare le figure fino al singolo pixel (dunque REAL DIMENSION ZOOM), avere un piano che crea colore invisibile opaco (per coprire background o sprite sottostanti), riempimento di aree con punti di intensità desiderata (ottimo per dare l'idea di volume), possibilità di linee semplici, doppie, triple, curve (per le curve ho coniato un algoritmo particolarissimo...).

Vi domanderete perché i comandi sono scritti tutti in inglese: in effetti comandi come LAYER, GRATE, EDIT sono insostituibili e scrivere i nomi metà in italiano e metà in inglese non mi piaceva: così ho finito di tradurre quelli meno ricorrenti come RUBBER, FILL, SHADOW ecc...

Partiamo dal cursore: coloro che hanno la tastiera avanzata è meglio usino il tastierino numerico con Num Lock attivo. Per cambiare velocità usino il tasto al centro, quello contrassegnato dal numero 5.

Per chi ha la tastiera vecchio tipo con Num Lock disattivo si hanno in meno i movimenti obliqui (usare * + − per modificare velocità).

Prima di cominciare qualsiasi operazion con EDIT1 bisogna visualizzare il cursore o si finisce di far credere al

Si prega il Sig. Giancarlo Del Sordo di mettersi in contatto con la redazione.

computer di non averne bisogno. I comandi ad impatto diretto sono pochi ma efficaci: **insert** per richiamare la tavola di comando vera e propria, **delete** per attivare o disattivare le griglie, **home** per attivare o disattivare le scritte dei comandi diretti, **end** per disattivare momentaneamente tutte le griglie, **page up** per fissare tutte le immagini dei vari layer sullo schermo principale, **page down** per selezionare e lavorare con i layer.

I comandi ad impatto indiretto sono molti e anche difficili da spiegare: cominciamo da quelli che si ottengono a livello 0, premendo insert. Questi comandi interessano tutti i tasti di funzione da 1 a 10: faccio notare che premere insert serve solo all'operatore per ricordargli i comandi che avrebbe potuto ottenere premendo i tasti di funzione durante l'operazione di disegno.

F1 serve per pulire la tavola e rappresentare il disegno qualora ve ne sia bisogno o qualora vi sia bisogno di una buona pulita dai comandi di Rectangle.

F2 serve per rappresentare la Fill Table.

F3 è indubbiamente il più lungo e il più noioso. Permette di svolgere delle operazioni su una area delimitata. Per delimitare una determinata area basta pensare a due vertici opposti del rettangolo che racchiuda tutta la superficie. Uno deve essere l'ultimo punto preso in considerazione da una line statement o da un F5 o da un F6. L'altro invece si raggiunge col cursore e si fissa con F3.

F4 serve per scegliere lo stile delle linee.

F5 e **F6** servono per fissare un punto, uno non disegnandolo, l'altro disegnandolo.

F7 crea una linea dall'ultimo punto preso in considerazione o da un'altra linea o da F5 o da F6 al punto corrente del cursore. Giostrando con i tasti del cursore e F7 si può creare una sagoma continua: si varia di poco la posizione del cursore e si preme F7, un altro poco e ancora F7 (sistema veloce e semplice per creare gli sprite, dove non è richiesta tanto la precisione quanto la semplicità e la velocità).

F8 serve per richiamare da disco (premendo poi F1) o una figura prememoriz-

È disponibile, presso la redazione, il disco con il programma pubblicato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 263.

```
{Shape.sys                                              Del Sordo G. Softw
type
matrix1=array [0..16383] of byte;
matrix2=array [0..8191 ] of integer;
matrix3=array [1..100  ] of integer;
matrix4=string [7];
var
A,A1,A2,Saveb1,Saveb2:^matrix1;
Field:array [1..9] of ^matrix1;
Savei1,savei2,savei3,savei4:^matrix2;
FileS,FileB:file of matrix1;
FileI:file of matrix2;
Contia,contfa,contpa,contre:matrix3;
Filec:file of matrix3;
File1:file of integer;
i,cont,figure,contref,last:integer;
Modovideo:byte absolute $0000:$0465;
fname1,fname2:string [9];

Procedure GroupSprites (name :matrix4);begin hires;
if Modovideo=7 then a:=ptr ($B000,$0000) else a:=ptr ($B800,$0000);
a1:=ptr ($4C00,$0000);a2:=ptr ($5000,$0000);
field [1]:=ptr ($5400,$0000);field [2]:=ptr ($5800,$0000);
field [3]:=ptr ($5C00,$0000);field [4]:=ptr ($6000,$0000);
field [5]:=ptr ($6400,$0000);field [6]:=ptr ($6800,$0000);
field [7]:=ptr ($6C00,$0000);field [8]:=ptr ($7000,$0000);
field [9]:=ptr ($7400,$0000);saveb1:=ptr ($7800,$0000);
saveb2:=ptr ($7C00,$0000);savei1:=ptr ($8000,$0000);
savei2:=ptr ($8400,$0000);savei3:=ptr ($4800,$0000);
savei4:=ptr ($4400,$0000);fname1:='B:'+name;delay (3000);
a1^:=a^;a2^:=a^;saveb1^:=a^;saveb2^:=a^;
for i:=1 to 9 do field [i]^:=a^;
for i:=0 to 8191 do begin savei1^ [i]:=0;savei2^ [i]:=0;savei3^ [i]:=0;
savei4^ [i]:=0      end;
Assign (fileb,fname1+'.BCM');Assign (file1,fname1+'.ICM');
Assign (filec,fname1+'.DCM');Assign (file1,fname1+'.LCM');
reset (fileb);reset (file1);reset (filec);reset (file1);seek (file1,0);
read (file1,last);seek (filec,0);read (filec,contia);seek (filec,1);
read (filec,contfa);seek (filec,2);read (filec,contpa);seek (filec,3);
read (filec,contre);
Seek (fileb,0);read (fileb,Saveb1^);
if contpa [last]>2 then begin seek (fileb,1);read (fileb,Saveb2^) end;
Seek (file1,0);read (file1,savei1^);
if contpa [last]>1 then begin seek (file1,1);read (file1,savei2^) end;
if contpa [last]>2 then begin seek (file1,2);read (file1,savei3^) end;
if contpa [last]=4 then begin seek (file1,3);read (file1,savei4^) end    end;

Procedure GroupBkGrounds (name:matrix4);begin
Fname2:='B:'+name;Assign (files,fname2+'.SCM');reset (files) end;

Procedure SetBkGrounds    (first,second:integer);begin
cont:=0;for i:=first to second do begin Seek (files,i-1);cont:=cont+1;
read (files,field [cont]^) end end;

Procedure BkGround        (first:integer);begin
a2^:=field [first]^ end;

Procedure Recall1;begin
for i:=contia [figure] to contra [figure] do
a1^ [savei1^ [i]+contref]:=saveb1^ [i] end;
Procedure Recall2;begin
for i:=contia [figure] to contfa [figure] do
a1^ [savei2^ [i]+contref]:=saveb1^ [i+8192] end;
Procedure Recall3;begin
for i:=contia [figure] to contfa [figure] do
a1^ [savei3^ [i]+contref]:=saveb2^ [i] end;
Procedure Recall4;begin
for i:=contia [figure] to contfa [figure] do
a1^ [savei4^ [i]+contref]:=saveb2^ [i+8192] end;

Procedure Sprite          (first:integer);begin
figure:=first end;

Procedure ToXY            (moreX,moreY:integer);begin
contre [figure]:=contre [figure]+moreX+80*moreY;
contref:=        contre [figure];case contpa [figure] of
1:recall1;2:recall2;3:recall3;4:recall4 end end;

Procedure InXY            (moreX,moreY:integer);begin
ToXY (moreX,moreY);a2^:=a1^ end;

Procedure Video1;begin
a^:=a1^ end;

Procedure Video2;begin
a^:=a2^ end;

Procedure NewTo;begin
a1^:=a2^ end;
```

zata in memoria o presente semplicemente in memoria tramite F2: a questo punto si usa F1 F2 per selezionare il numero dello sprite e F9 per concludere le operazioni e tornare a livello 0.

F9 serve in generale in ogni livello per concludere le operazioni e scendere di livello. Qui, a livello 0 abbandona il programma non senza prima aver memorizzato su disco tutte le figure salvate precedentemente.

F10 serve per sopprimere la visualizzazione di finestre a qualsiasi livello. F10 e Home insieme sopprimono completamente qualsiasi forma di aiuto data dalla visualizzazione del nome dei comandi in atto.

Adesso passiamo ad approfondire la spiegazione con la descrizione dei comandi a livello superiore.

Partiamo dai comandi di Rectangle, quelli per l'esattezza ottenuti con F3. In questo livello abbiamo i seguenti comandi:

F1 serve per memorizzare l'area prescelta: rappresenta a memorizzazione effettuata il numero della pagina nella quale ha memorizzato la figura (in tutto sono 4), il numero dei settori liberi (in tutto sono 8) e, unico dato di fatto indispensabile in numero associato alla figura.

In caso di insufficienza di memoria sarà dato un Memory over. Non si possono creare figure troppo grandi: se dà un errore tipo Too big to fit in the memory bisogna spezzare la figura in due parti (tipo il caso di un background tutto schermo).

F2 serve per cancellare l'area prescelta nello schermo attivo: i layer rimarranno pertanto immutati.

F3 serve per riempire un'area di un numero determinato di punti.

F1 F2 per scegliere il numero.

F4 serve per zoomare un'area discretamente piccola. Serve per vedere lo sprite nel dettaglio e capire magari un risultato errato in Fill.

Scegliere un'area veramente piccola (10*5) per non invecchiare di fronte al monitor.

F5 serve per creare nell'area prescelta una grata e renderla attiva o disattiva. Completa la funzione di Del a livello 0.

F6 serve per tracciare un riempimento di colore opaco trasparente (Fill); l'area deve contenere appena la sagoma per risparmiare tempo. La sagoma non deve presentare disegni all'interno, non deve avere interruzione. Se non è riempita completamente, ripetere il comando nelle aree non riempite. In caso di altri errori usare le opzioni F3 e F4 di layer.

Adesso passiamo agli stili di linea (F4 a livello 0); con F1 si sceglie se fare linee curve o linee normali, con F2 e F3 si regola l'ispessimento delle linee, con F4 il colore.

```
program x;
{$I Shape.sys}
{       Semplice  routine che chiama Urana.?cm
      In effetti il file  e'  adibito a creazioni spaziali e contiene
      1-astronave mia
      2-astronave mia con ruota
      3-astronave nemica 1
      4-astronave nemica 2
      5-astronave nemica 2 semidistrutta
      6-polvere cosmica concentrata
      7-polvere cosmica poco concentrata
      8-astronave mia semidistrutta
      9-astronave nemica 3 che spunta dallo schermo
     10-astronave nemica 3 intera
     11-astronave nemica 3 semidistrutta
     12-missile mio
     13-missile nemico
In questo esempio non ho pensato agli sfondi (ci pensate voi)
Gustatevelo:si potrebbe usare il file come parte di un game spaziale }
var c:integer;
begin
GroupSprites ('Urana');
{Come si potrebbe presentare la mia astronave ferma e dopo un attacco}
sprite (1);toxy (50,10);sprite (2);toxy (50,10);
for c:=1 to 30 do begin sprite (1);toxy (-1,0);videol;newto;
                        sprite (2);toxy (-1,0);videol;newto end;
sprite (8);toxy (20,10);videol;newto;
sprite (6);toxy (10,5);videol;newto;
sprite (7);toxy (10,15);videol;newto;videol;delay (1000);
sprite (9);toxy (0,0);videol;newto;
for c:=1 to 30 do begin sprite (10);toxy (1,1);videol;newto end;
sprite (11);toxy (30,30);videol;newto;delay (250);
sprite (7);toxy (-10+30,-15+30);videol;delay (100);newto;videol end.
```

Per l'editazione di linee dovrebbe essere tutto chiaro se si tratta di rette. Se invece si tratta di curve si passa all'editazione della curva dalla posizione del cursore usando F1 F2 per regolare la x massima, F3 F4 per regolare il raggio, F5 F6 per regolare la posizione e F7 F8 per regolare l'eccentricità.

Si ricorda che Save (F3-F1) registra solo quanto sta sullo schermo attivo, quindi usare Fix (Page up) per il fissaggio di quanto si vede sullo schermo allo schermo attivo.

Per i layer F1 F2 servono per imprimere quanto sta sullo schermo in un layer e per selezionare i layer attivi. F3 serve per trasferire Fill Table sullo schermo attivo. A questo punto si fanno le modifiche e si usa F4 per riportare tutto alla normalità. A questo punto si può sopprimere il layer 5 usato da EDIT1 per l'operazione.

EDIT2

Questo editor si usa per trasformare un'area selezionata in EDIT1 in uno sprite. Non si può cambiare il nome del file, tutto quello che vi era prima sarà sostituito da quello richiamato da EDIT2. Quello che basta per operare questa trasformazione è di richiamare l'area corrispondente. Le altre opzioni non sono obbligatorie.

F1 permette di operare questo richiamo: con F1 F2 si opera una selezione del numero dello sprite. Le modalità di rappresentazione sono quelle descritte in EDIT1.

F2 permette di posizionare uno sprite in un posto ben preciso dello schermo: è obbligatorio usare Num Lock attivo, dunque attenersi a quanto detto all'inizio del paragrafo EDIT1. La velocità del cursore non è visualizzata. Le modalità di richiamo della figura riguardano il numero con il quale EDIT2 ha memorizzato le figure richiamate.

F3 permette di salvare due figure presenti nello schermo attivo in una terza, fondendole.

F4 opera una formattazione della memoria.

F9 torna a sistema operativo.

EDIT3

Quando si devono creare degli sfondi si passa sotto il torchio di questo editor.

F1 come F1 di EDIT2 richiama entità da disco.

F2 richiama da disco interi sfondi creati da un EDIT3 precedente e li fissa sullo schermo attivo.

F3 come F2 di EDIT2.

F4 permette di fare operazioni con uno dei 9 schermi utilizzati da EDIT3. F1

```
program x;
{$I shape.sys}
var c,t,d:integer;
begin
groupsprites ('Prova');
groupbkgrounds ('Prova3');
setbkgrounds (1,2);
bkground (1);newto;
sprite (1);d:=2;
toxy (45,50);video1;sprite (2);toxy (45,50);
for t:=1 to 2 do begin
for c:=1 to 90 do begin
bkground (d);d:=d+1;if d=3 then d:=1;
newto;sprite (d);toxy (1-t,1-t);video1;end;end;
hires;
 end.
```

permette di selezionare lo Screen con il quale si vogliono fare delle operazioni. F3 permette di trasferire dalla memoria lo Screen corrente allo schermo attivo. F4 trasferisce dallo schermo attivo allo Screen corrente. F5 resetta tutto lo Screen corrente. F6 permette di decidere se lo Screen corrente debba essere salvato. Assicurarsi che sia posizionato su On ogni Screen che si vuole registrare prima di abbandonare il sistema.
F5 nella versione 1.0 resetta le entità richiamate con F1.
F9 torna a sistema operativo.

EDIT 4

Visto che SHAPE.SYS può richiamare anche 18 Screen mentre EDIT3 ne controlla al massimo 9, EDIT4 permette di concatenare più Screen in un unico file. All'inizio vengono richiesti il nome dei due file di immissione e quello del file di uscita.

Viene domandato l'insieme di Screen continui dei primi due file e se essi eccedono le capacità vengono prese in esame le capacità massime.

SHAPE.SYS

Si richiama con direttiva di include {$I SHAPE.SYS}
GroupSprites ('nome')
Comando obbligatorio da porsi all'inizio del programma in Turbo.
È indicato il nome del file che contiene gli sprite.
GroupBKGrounds ('nome')
Comando non obbligatorio. Il nome riguarda il file che contiene gli sfondi.
SetBKGrounds (intero1, intero2)
Setta un insieme di sfondi dal file nominato da GroupBKGrounds. Intero1 è il numero di campo di inizio, Intero2 è il numero di campo di fine.
Sistema il blocco intero1-intero2 a partire da BKGround (1) in poi.
BKGround (intero)
Setta lo sfondo che porta quel numero.
Sprite (numero)
Setta lo sprite che porta quel numero.
Inxy (X-in-piu', Y-in-piu')
Fissa lo sprite corrente nello sfondo alle coordinate aggiuntive nominate. Finché non si resetta BKGround la figura rimarrà sovrapposta come se lo sprite avesse sempre fatto parte dello sfondo.
Toxy (X-in-piu', Y-in-piu')
Stesso effetto di InXY ma fissa lo sprite temporaneamente fino a un NewTo.
Video1
Mostra lo schermo corrente con gli Toxy nominati.
Video2
Mostra lo schermo del background corrente con in più le modifiche degli Inxy.
NewTo
Azzera lo schermo 1 come spiegato a Toxy.

Esempi

Come avrete potuto notare il dischetto contiene dei programmi in Pascal (Sprova1, Sprova2.pas) e dei file .?CM.

Sono esempi che ho creato per darvi una idea delle tre diverse potenzialità di Shape.

Sprova1 è un file molto semplice che mostra una parte di un file chiamato Urana che contiene potenzialmente gli Sprite pronti pronti per un giochino spaziale. Se vi sembra lento è solo una impressione: ho fatto muovere le astronavi di poco alla volta (in gioco possono fare in una volta, per esempio, balzi più grossi e dare l'effetto di un balzo velocissimo).

Sprova2 dà un'idea delle capacità di simulazione del movimento: un elicottero F41 tenta di alzarsi in volo durante la tempesta di un violento temporale che impedisce quasi la visibilità. A stento dopo essere stato a lungo fermo e con le pale in crescente velocità si alza in volo mentre la pioggia continua ad infuriare. Il file Prova2.?CM contiene uno sprite di numero 1 richiamabile da EDIT1 o EDIT2 che permette di impratichirsi con le operazioni su disco di Shape. Lo sprite è un elicottero tipo Supercopter e dà una buona idea invece sulle capacità grafiche di Shape (pensate a un gioco con delle figure modellate così!). Ci rimarrete molto male quando saprete che io (ovviamente il programma lo conosco bene, ma anche voi potreste impratichirvi a questo punto) ho disegnato quelle figure in 15 minuti!

Certo, voi ce ne potrete mettere 30, ma anche così facendo in una giornata buttate giù le basi di un game impensabile persino da Turbo!

Dati tecnici

Il sistema con il quale si registrano i file lo ho chiamato ?CM e come si può vedere dal listato di SHAPE.SYS è semplice e veloce da richiamare.

Presenta l'unico inconveniente di mangiarvi tutti i dischi: utilizzando una pagina consumate 32K, due pagine 48K, tre pagine 80K, quattro pagine addirittura 96K!

Gli sprite per un buon giochino possono occupare anche più di un disco (18 backgrounds compresi)!!

Conclusioni

Se siete arrivati fin qui vi sono due probabili casi. O siete come quelle splendide modelle francesi che assaggiucchiano i piatti e poi vanno avanti per vedere cosa c'è, o avete acceso il vostro MEGA-AT, vi siete seduti sulla vostra poltrona con rifiniture da guerre stellari e vi siete decisi a rovinarvi il pomeriggio per vedere «Che diavolo ci ha combinato 'sto qua», magari sbuffando quando vi siete resi conto delle limitazioni della gestione delle memorie di massa o della mancanza di gestione del mouse o dell'utilizzo confinato a CGA.

In effetti questo è un programma molto strano: alcuni potrebbero rimanere contenti (dopotutto a creare sprite in Turbo non ci aveva pensato nessuno), altri potrebbero dire «Che schifo» e magari sbarazzarsene nel cestino confinante alla propria poltrona.

Infine un ringraziamento da parte mia a Alessandro Carpitella e Alessandro Biavasco per avermi dato una spinta morale a continuare a scrivere i vari programmi tutte le volte che mi è venuta la voglia di piantare tutto e di darmi all'ippica.