

"POLI_MENU" - UN SOTTOPROGRAMMA PER
APPLICAZIONI INTERATTIVE SU PERSONAL COMPUTER
CHE OPERANO SU ARCHIVI STRUTTURATI A "DATA-BASE"

A.Ficarra

IRA 171/92

INTRODUZIONE

Con questo rapporto l'autore presenta un sottoprogramma, di nome POLI MENU, scritto in linguaggio CLIPPER. Com'è noto, il CLIPPER è un sistema per lo sviluppo di applicazioni su Personal Computer che utilizzano archivi strutturati a "database" sotto sistema operativo MS-DOS, e si compone di un compilatore e di alcune librerie di funzioni. Inoltre POLI MENU accede a una libreria, di nome FUNCKY, la quale è particolarmente indicata per applicazioni di tipo interattivo, consentendo lo sviluppo di interfacce-utenti che sfruttano a fondo le potenzialità dell'elaboratore senza la necessità di fare ricorso a linguaggi maggiormente orientati alla "macchina", quali il C o l'Assembler. Infatti la FUNCKY consiste di alcune centinaia di funzioni le quali, da un lato, ampliano ed arricchiscono le potenzialità già insite nelle librerie di CLIPPER, in particolar modo per quello che riguarda la gestione dello schermo e del colore e l'accesso agli archivi, dall'altro introducono nuove possibilità, come per esempio l'utilizzo del "mouse", che attualmente le librerie standard di CLIPPER non consentono.

POLI MENU è un sottoprogramma, richiamabile da un'applicazione CLIPPER sotto forma di funzione, che consente di fornire al programma principale l'informazione richiesta, selezionandola fra le "opzioni" elencate in una "catena di menu", comunque strutturata e complessa. Partendo da un "menu principale" e facendo uso del mouse, l'operatore può attivare (o disattivare) dei sottomenù, percorrere avanti e indietro tutte le possibili diramazioni della catena, eseguire operazioni di scorrimento delle opzioni all'interno di ogni menu ("scrolling"), e infine selezionare l'opzione che gli interessa. Rispetto ad altri sottoprogrammi di CLIPPER o della stessa FUNCKY che eseguono le medesime operazioni, POLI MENU presenta il vantaggio di non porre limiti né alle dimensioni dei menu né alla complessità della catena, consentendo, contemporaneamente, sia l'organizzazione gerarchica dei menu che la possibilità di "scrolling" all'interno di ciascuno di essi.

Questo rapporto si rivolge, genericamente parlando, agli "utenti" di POLI MENU ed è articolato in due parti nettamente distinte, in ragione del significato particolare che viene attribuito al termine "utenti". Esistono infatti gli "utenti-programmatori", cioè coloro che scrivono programmi applicativi in CLIPPER con chiamate a POLI MENU (che, ricordiamo, è un sottoprogramma-funzione e pertanto necessita sempre di un "main"), ed esistono "utenti-operatori", cioè coloro che utilizzano applicativi già fatti e quindi non sono interessati alla struttura del programma ma solo al suo uso. Questa distinzione è importante, in quanto gli operatori possono in tal modo evitare la lettura dei capitoli dedicati alla programmazione e consultare esclusivamente la parte di loro interesse. Infatti, qualunque sia l'applicazione che faccia ricorso a POLI MENU, esistono norme operative comuni, indispensabili da conoscere per un completo e corretto utilizzo delle possibilità offerte dal programma.

ISTRUZIONI PER GLI OPERATORI

L'utente che, per la soluzione di un suo problema particolare, intende far girare un programma già compilato e "link-editato" (cioè in formato eseguibile), il quale a sua volta è stato costruito con chiamate alla funzione POLI_MENU, deve anzitutto assicurarsi che il P.C. su cui lavora sia provvisto di un "mouse" a tre bottoni. La presenza del mouse è infatti obbligatoria. Tuttavia, a beneficio di quegli utenti a cui l'uso del mouse non è congeniale, tutte le sue funzioni possono essere simulate dalla tastiera: precisamente, per quello che riguarda il movimento, si può ottenere lo stesso risultato utilizzando i tasti con la "freccia", e, per quello che riguarda le selezioni, i bottoni di sinistra, di centro e di destra del mouse sono simulati rispettivamente dai tasti "funzione" F5, F6 e F7; il bottone di sinistra del mouse è simulato anche dal tasto "Enter". Va precisato, però, che il programma agisce in ogni caso sul cursore del mouse e pertanto, affinché questo sia visibile, il driver di un mouse a tre bottoni deve essere comunque installato in memoria, e il mouse stesso, anche se inutilizzato, deve essere fisicamente collegato al P.C.

Cio' premesso, l'operatore utilizzerà il mouse (o le sue funzioni simulate su tastiera) per "navigare" all'interno dei menu' e per effettuare le selezioni delle opzioni prescelte.

Si è detto che POLI_MENU permette di lavorare su una catena di menu' comunque complessi, organizzati gerarchicamente, cioè ad "albero", con un menu' principale e vari eventuali sottomenu', ciascuno dei quali può attivare a sua volta altri sottomenu' e così via. Ogni menu' può essere, a seconda di come è stato definito nel programma, di tipo "orizzontale", cioè con le opzioni elencate sulla medesima riga, oppure di tipo "verticale", cioè con le opzioni disposte su righe diverse e incolonnate. In ogni caso il menu' appare all'interno di una figura rettangolare ("finestra"), con colore e dimensioni definiti dal programma, attraverso la quale sono visibili, tutte o in parte, le stringhe di caratteri che definiscono le diverse opzioni.

Muovendo il cursore del mouse all'interno della finestra del menu' principale, l'operatore può effettuare la prima selezione, senza spingere alcun bottone del mouse, ma semplicemente fermando per un attimo il movimento del cursore sull'opzione prescelta; questa appare quindi in "evidenza" (cioè con colori di foreground e background invertiti), con due diverse possibili conseguenze, a seconda dei casi:

- 1) L'opzione selezionata attiva a sua volta un altro menu': compare sullo schermo una nuova finestra, con nuove opzioni, che permettono di specificare con maggiore dettaglio la scelta più generale effettuata nel menu' precedente; l'operatore può eseguire tale scelta posizionando il cursore del mouse all'interno della nuova finestra e così proseguendo in altri eventuali sottomenu' finché non seleziona un'opzione finale, cioè una opzione che non attiva ulteriori menu'. Esiste anche la possibilità di "tornare indietro" su menu' precedenti: questo si realizza spostando il cursore su un'altra opzione del menu' che ha generato il sottomenu', oppure, se il cursore si trova già all'interno della finestra dell'ultimo menu', riportandolo fuori da essa; in quest'ultima situazione il cursore è ricollocato automaticamente nel menu' superiore e, in ogni caso, la finestra dell'ultimo menu' attivato si "spegne", cioè scompare dal video. Si consiglia, almeno in fase di prova, di sfruttare tale possibilità, percorrendo avanti e indietro tutte le possibili ramificazioni dei menu', per rendersi conto di quante e quali operazioni il programma consenta di eseguire.

- 2) L'opzione evidenziata non accende altri menu': si tratta, come s'e' detto, di un'opzione "finale": per effettuare la selezione bisogna premere il bottone di sinistra del mouse (mantenendo il cursore posizionato sull'opzione prescelta); questo fa si' che il programma "ritorni" dall'esecuzione di POLI MENU ed esegua le operazioni corrispondenti all'opzione "finale" selezionata.

Quando il numero delle opzioni possibili di un menu' supera le dimensioni della finestra definita per quel menu,' si accende una "freccetta" (su uno dei due lati opposti della finestra, o su entrambi, a secondo dei casi). Se il menu' in questione e' "attivo", cioe' se e' l'ultimo selezionato, e' possibile far scorrere le opzioni "dietro" la finestra ("scrolling"), operando a scelta con la tastiera o con il mouse, nel seguente modo:

- 1) "Scrolling" mediante tastiera (ovviamente le operazioni vanno eseguite prima di effettuare la selezione):
 - a) premendo il tasto "Page-down" si ottiene lo scorrimento delle opzioni in avanti, nella misura pari ad una "pagina", cioe' alle dimensioni della finestra;
 - b) premendo il tasto "Page-up" le opzioni scorrono indietro di una pagina;
 - c) premendo il tasto "Home" si riporta il menu' all'inizio;
 - d) premendo il tasto "End" si riporta il menu' alla fine.
- 2) "Scrolling" mediante mouse :
 - a) premendo il bottone di destra si attiva l'operazione di "scrolling", evidenziata dall'accendersi di numerose freccette sui bordi della finestra;
 - b) da questo momento, ogni volta che si preme il bottone di sinistra, si provoca uno scorrimento pari a una "pagina";
 - c) ogni volta che si preme il bottone centrale si riporta il menu' all'inizio o alla fine;
 - d) in entrambi i casi precedenti lo scorrimento avviene automaticamente in avanti o all'indietro se esiste una sola direzione possibile; se invece lo "scrolling" puo' essere eseguito in entrambe le direzioni, lo spostamento e' all'indietro se l'operatore ha posizionato il cursore entro la meta' superiore della finestra (o di sinistra nel caso di menu' orizzontale), mentre e' in avanti nel caso contrario;
 - e) premendo nuovamente il bottone di destra si disattiva la funzione di "scrolling" e si riabilita lo stato di attesa di input da menu'.

A ogni menu' e' associato un messaggio informativo, il cui contenuto e' definito dal programmatore. Nel caso che piu' menu' siano contemporaneamente visibili, il messaggio si riferisce all'ultimo menu' selezionato e appare (con lo stesso colore) sull'ultima riga in basso dello schermo (riga 24). Il messaggio puo' essere anche piu' lungo di 80 caratteri (per l'esattezza, fino a 512) e, se cio' avviene, una freccetta a destra ne indica la continuazione; l'operatore puo' leggere la parte rimanente, utilizzando, anche in questo caso, una funzione di "scrolling" (orizzontale), nel seguente modo:

- a) premendo il bottone centrale del mouse attiva lo "scrolling", evidenziato dall'accendersi di freccette verticali su tutta la riga soprastante;
- b) con la funzione di "scrolling" attivata, premendo il bottone di sinistra (o di destra), fa scorrere il messaggio verso sinistra (o verso destra) di 40 posizioni, cioe' di mezzo schermo;
- c) premendo nuovamente il bottone centrale disattiva la funzione di "scrolling" e riabilita lo stato di attesa di input da menu'.

Il messaggio di riga 24 e' molto importante e si raccomanda all'operatore di leggerlo sempre con estrema attenzione. Infatti, per la sua possibile lunghez-

za, il messaggio e' in grado di fornire informazioni anche molto dettagliate, e per tale motivo POLI_MENU non prevede (in quanto non ne ha bisogno) di associare messaggi alle singole opzioni o di abbinare ai menu' altre funzioni di "help".

ISTRUZIONI PER I PROGRAMMATORI

Informazioni generali

Si e' detto che POLI_MENU e' un sottoprogramma-funzione, cioe' deve essere "chiamato" da un programma applicativo (anch'esso scritto in CLIPPER) con una sequenza di parametri e restituisce al programma chiamante un valore di ritorno.

POLI_MENU non "pulisce" lo schermo del video; questo, per dare modo al programma di combinare la visualizzazione dei menu' con quella di altre informazioni generate precedentemente; in caso cio' non interessi, spetta al programma chiamante effettuare il "clear" dello schermo prima di accedere a POLI_MENU.

Sia i menu' che le singole opzioni sono identificate nel programma da un valore numerico. In particolare le opzioni sono identificate da un numero "assoluto", cioe' la numerazione procede dalla prima opzione all'ultima senza riazzerarsi fra un menu' e l'altro.

La funzione POLI_MENU restituisce il numero di identificazione dell'opzione selezionata. In condizioni normali solo il bottone di sinistra del mouse e' abilitato a provocare il ritorno al programma chiamante (quando, come abbiamo detto, il cursore del mouse e' posizionato su un'opzione "finale"). Tuttavia, utilizzando un parametro della chiamata, si possono opzionalmente abilitare, per il ritorno, anche tasti selezionati della tastiera (oltre al tasto <ESC> che e' sempre attivo). In questo caso POLI_MENU restituisce un numero negativo, che equivale al codice ASCII, cambiato di segno, del tasto premuto. Quando si preme <ESC> la funzione restituisce in ogni caso il valore zero.

Per queste sue caratteristiche POLI_MENU si presta a essere utilizzata non solo per gestire dei menu', ma anche per altri scopi, come per esempio l'esame visivo o la ricerca e selezione di un'informazione all'interno di un file di testo anche molto lungo: in questo caso POLI_MENU potrebbe essere chiamata ricorsivamente all'interno di un ciclo di "do", in cui, se il ritorno e' da tastiera, il carattere trasmesso potrebbe essere il primo di una stringa "gettata" (facendo uso del comando "keyboard" per la memorizzazione del carattere) e utilizzata per restringere o spostare il campo di ricerca nella chiamata successiva a POLI_MENU; se invece il ritorno e' a seguito di selezione effettuata tramite mouse, questo potrebbe causare l'interruzione del ciclo di "do", con l'informazione selezionata a disposizione del programma chiamante.

Archivi di appoggio

Per lavorare, la funzione POLI_MENU si appoggia a due archivi, strutturati a

"database", la cui generazione e' a cura del programma chiamante. Il primo archivio contiene le informazioni su i menu' mentre il secondo archivio riguarda le opzioni. La loro struttura e' la seguente:

Primo archivio (un record per ogni menu'):

NUMMENU	N	3	Indicizzato su	NUMMENU
HV	C	1		
ROC	N	2		
CLORT	N	2		
CRORB	N	2		
MENUCOL	N	3		
MESSMENU	C	512		

Dove :

- NUMMENU e' il numero d'ordine del menu';
- HV definisce se il menu' e' orizzontale (H) o verticale (V);
- ROC definisce la riga (se il menu' e' H) o la colonna (se il menu' e' V) del top-left corner della finestra;
- CLORT definisce la colonna (se il menu' e' H) o la riga (se il menu' e' V) del top-left corner della finestra;
- CRORB definisce la colonna (se il menu' e' H) o la riga (se il menu' e' V) del bottom-right corner della finestra; la quarta coordinata, cioe' la riga (se il menu e' H) o la colonna (se il menu' e' V) del bottom-right corner della finestra viene determinata automaticamente: e' univoca nel caso di menu' orizzontale, mentre, se il menu' e' verticale, dipende dalla lunghezza massima delle "prompt-strings" delle opzioni;
- MENUCOL e' un numero che identifica il colore utilizzato (con foreground e background invertiti sull'opzione selezionata), secondo il seguente codice standard:

0 = Nero	8 = Grigio
1 = Blu	9 = Blu brillante
2 = Verde	10 = Verde brillante
3 = Ciano	11 = Ciano brillante
4 = Rosso	12 = Rosso brillante
5 = Violetto	13 = Violetto brillante
6 = Marrone	14 = Giallo
7 = Bianco	15 = Bianco brillante

Il valore da attribuire a MENUCOL si compone delle parti di foreground (f) e di background (b) secondo la seguente formula : $f+b*16$. Per esempio, se si vuole che il menu' sia scritto in giallo su sfondo blu, MENUCOL deve essere: $14+1*16 = 30$. Ovviamente, la gestione del colore e' possibile solo se il tipo di monitor utilizzato lo consente; in caso di monitor in bianco e nero, gli unici colori possibili sono: 7 (bianco su nero) o il suo inverso 112 (nero su bianco). A questo proposito si suggerisce di inserire all'inizio del programma un'interrogazione all'operatore sul tipo di monitor utilizzato e di definire un'array di indici di colori in funzione della risposta dell'operatore; successivamente, tutti gli indici di colori utilizzati dal programma (e quindi anche i valori di MENUCOL) dovrebbero essere definiti solo tramite gli elementi dell'array.

NOTA : gli indici compresi fra 8 e 15, se usati per definire il background, producono diverso risultato a secondo del parametro di chiamata della funzione BLINKBIT (della libreria FUNCKY). Precisamente:

se si inizializza il programma con la chiamata: BLINKBIT(.F.) tutti i numeri di colore hanno il significato indicato in tabella; se non si chiama la funzione BLINKBIT o si pone: BLINKBIT(.T.), ogni volta che b e' maggiore di 7 si comporta come se fosse: $b-8$ (cioe' il colore di fondo che appare e' il corrispondente non "brillante"); in compenso tutte le scritte appaiono "blinkanti", il che e' poco indicato e poco gradevole quando si

lavora con dei menu'. Si consiglia pertanto di inserire, come abitudine generale, l'istruzione BLINKBIT(.F.) all'inizio di ogni programma che usa POLI MENU.

- MESSMENU e' il messaggio associato al menu', che, come si e' detto, appare alla riga 24 dello schermo.

Secondo archivio (un record per ogni opzione):

NUMMENU	N	3	Indicizzato su :
ID_OPTION	N	4	str(NUMMENU,3)+str(ID_OPTION,4)
PROMMEN	C	70	
CALLED	N	4	

Dove :

- NUMMENU e' il numero del menu' a cui l'opzione si riferisce;
- ID_OPTION e' il numero che identifica l'opzione;
- PROMMEN e' la "prompt-string" dell'opzione;
- CALLED e' il numero del sottomenu' chiamato dall'opzione (se non e' una opzione "finale"); altrimenti: CALLED=0.

Il nome del primo archivio e' a scelta del programma chiamante (massimo 6 caratteri). Il secondo archivio deve chiamarsi come il primo, con l'aggiunta di una "S" finale. Gli indici sono di tipo .NTX e devono avere lo stesso nome dei rispettivi .DBF. In questo modo, fornendo, in un parametro della chiamata a POLI_MENU, il nome del primo archivio, restano automaticamente determinati i nomi dei due files .DBF e dei due files .NTX utilizzati dalla funzione.

Talvolta gli archivi devono essere generati dinamicamente, all'interno dello stesso programma chiamante. In questo caso il programma deve provvedere a definire tutti i menu' con posizioni, colori e dimensioni delle finestre. A questo proposito e' a disposizione la funzione POLI_CREA che ammette come parametro il nome del primo archivio e crea i due archivi anzidetti (con contenuto vuoto) e i rispettivi indici. Se invece gli archivi possono essere determinati a priori, cioe' non dipendono dal flusso del programma, esiste un programma gia' fatto, di nome DBMENU, che permette di costruire gli archivi interattivamente, eseguendo tutti i controlli di compatibilita' necessari. Un altro programma, di nome MENUTEST, consente di testare gli archivi (generati da DBMENU o in qualunque altro modo), prima di utilizzare gli stessi nel programma applicativo. Entrambi i programmi chiamano a loro volta la funzione POLI_MENU e quindi utilizzano tutte le tecniche sopra descritte (vedi sezioni seguenti).

Sequenza di chiamata

```
ret = POLI_MENU(dbn,niniz,contasto,idiniz,numex,arrex)
```

dove:

dbn e' il nome del primo archivio (senza .DBF) (obbligatorio)
niniz e' il numero del menu' principale della catena (obbligatorio)
contasto e' una stringa opzionale contenente un'espressione logica che indica quali tasti della keyboard (identificati dai loro codici ASCII) sono abilitati per il ritorno. Esempio: "(tasto>47.and.tasto<58)" significa che sono abilitati i soli caratteri numerici. Tre osservazioni: 1) racchiudere l'espressione fra virgolette e fra parentesi, come nell'esempio; 2) il nome della variabile di confronto all'interno dell'espressione logica non e' di fantasia, ma deve proprio essere: "tasto" come nell'esempio; 3) tasto=0 (mouse) e tasto=27 (ESC) non sono da includere in quanto di default; come pure non sono da includere: tasto=18 (Page Up), tasto=3 (Page Down), tasto=1 (Home) e tasto=6 (End), in quanto il programma li usa per eseguire le operazioni di "scrolling"

idiniz e' il numero di identificazione dell'opzione che deve inizialmente apparire in cima al primo menu'. Opzionale; se e' omesso o se e' zero, appare la prima opzione del menu'

numex e' il numero di opzioni da escludere. Opzionale; se e' omesso o se zero sono incluse tutte le opzioni dei menu' selezionati

arrex e' un array contenente i numeri di identificazione delle opzioni da escludere. Opzionale (obbligatorio se numex e' definito e maggiore di zero). Se un'opzione e' memorizzata con segno meno, cio' indica che sono da escludere tutte le opzioni a partire dalla stessa (in valore assoluto) fino a quella indicata dall'elemento successivo dell'array (o fino alla fine dell'archivio, se il numero negativo e' in ultima posizione). Esempio:

numex=5 arrex[1]=4 arrex[2]=-6 arrex[3]=9 arrex[4]=15 arrex[5]=-20

(sono da escludere le opzioni: 4,6,7,8,9,15 e dalla 20 alla fine)

ret : se > 0, e' il numero di identificazione dell'opzione "finale" selezionata con il mouse

se = 0, e' stato premuto il tasto <ESC>

se < 0, e' stato premuto uno dei tasti abilitati, il cui codice ASCII e': -ret

Programma: DBMENU

Il programma DBMENU serve per generare interattivamente gli archivi contenenti tutte le informazioni sui menu' e le opzioni, necessarie per l'uso della funzione POLI MENU. L'utente deve solo preparare un testo, con nome dbn.INP (dove dbn e' il nome generico del primo dei due archivi da creare), con le seguenti informazioni (una per riga):

Messaggio associato al primo menu'

"Prompt-string" della prima opzione del primo menu'

"Prompt-string" della seconda opzione del primo menu'

ecc....

< una o piu' righe bianche >

Messaggio associato al secondo menu'

"Prompt-string" della prima opzione del secondo menu'

ecc....

Nel preparare questo file, l'unico criterio d'ordine da tenere presente e' che le informazioni relative a un sottomenu' devono essere successive a quelle relative al suo menu' principale.

Il programma lavora in varie fasi, ciascuna delle quali produce un output gia' completo e riutilizzabile nella fase successiva. Queste fasi sono :

- 1) creazione della struttura degli archivi e lettura del file di input, con produzione (a scelta dell'utente) di un file di testo, di nome dbn.OUT. Il suo contenuto e' molto simile a quello del file di input, con la differenza che tutte le opzioni e i menu' sono numerati. L'utente potrebbe, a questo punto, uscire dal programma, stampare il file, e consultare lo stampato per eseguire le fasi successive.
- 2) creazione dei legami fra i menu': l'utente deve selezionare da un menu' che presenta la lista di tutte le opzioni, quelle non "finali" (utilizzando il mouse o digitando il numero dell'opzione) e, per ogni opzione, selezionare da un altro menu' contenente la lista dei menu' successivi, il menu' "called" dall'opzione stessa (anche in questo caso la selezione puo' essere fatta con il mouse oppure digitando il numero del menu'). Alla fine del processo (determinata premendo <ESC>) viene di nuovo prodotto il file di testo dbn.OUT, contenente, in piu', il valore del campo "CALLED" di ciascuna opzione.

- 3) rilettura (opzionale) del file di input: se la fase successiva fallisce, in quanto le "prompt" sono troppo lunghe e le "finestre" si sovrappongono troppo le une con le altre, l'utente puo' modificare il file di input (solo il contenuto delle stringhe, non il loro numero o il loro ordine), riducendo in particolare la lunghezza delle stringhe e rientrando in DBMENU in questo punto, cioe' senza ridefinire i legami, gia' costruiti nella fase precedente.
- 4) definizione delle finestre: vengono determinati interattivamente: il tipo di menu' (orizzontale o verticale), la scelta del colore, la posizione e le dimensioni della finestra. Per quest'ultima operazione, eseguita con il mouse, il programma forza automaticamente le dimensioni minime della finestre in funzione della lunghezza delle "prompt".
- 5) esame visivo di ogni menu' costruito: se il menu' "piace" si va avanti, altrimenti si ricomincia dalla fase 4 (ovviamente per il solo menu' in esame: le fasi 4 e 5 riguardano un menu' alla volta).

L'esecuzione di DBMENU puo' essere interrotta (per esempio per correggere un menu' precedentemente definito) e ripresa esattamente dallo stesso punto, o da punti precedenti, con una grande flessibilita' che consente di eseguire il lavoro a piu' riprese e di modificare o ricostruire parti gia' fatte senza alterare il resto.

Programma: MENUTEST

Il programma MENUTEST permette di verificare visivamente il funzionamento di una catena di menu' utilizzando l'archivio associato. L'utente seleziona ciclicamente:

- 1) l'archivio;
 - 2) il menu' iniziale;
 - 3) le opzioni da escludere (con il mouse o digitando il numero, anche negativo, in tal caso per escludere tutte le opzioni comprese entro un "range");
 - 4) l'opzione da cui partire (idem);
- eseguite queste operazioni, puo' "navigare" fra i menu', verificare che i menu' sono stati costruiti correttamente, e infine selezionare un'opzione.

Funzioni associate a POLI_MENU

Nello stesso file in cui e' memorizzata la funzione POLI_MENU sono state inserite altre funzioni, che servono da appoggio a POLI_MENU, ma che potrebbero anche essere usate indipendentemente da un programma applicativo.

Oltre alla gia' citata POLI_CREA(dbn), troviamo:

- Funzione CENTLINE(rig, frase, colore)

Centra un messaggio in una determinata riga dello schermo.

Parametri :

rig : riga in cui deve essere scritto il messaggio

frase : messaggio, costituito da una stringa alfanumerica, lunga al massimo 80 caratteri

colore : indice di colore (foreground+background) del messaggio (il background riempie l'intera riga)

Ritorno : nullo

- Funzione CONT_WAIT(funz, condiz)

Mantiene il programma in stato di attesa fino al verificarsi di determinate condizioni di input da mouse o da tastiera

Parametri :

funz : stringa, che puo' essere: "m_inbutton(n)" o "m_release(n)";
fa si' che il programma attenda che venga premuto (o rilasciato)
il bottone del mouse specificato dal numero n (0=bottone di sini-
stra, 1=bottone di destra, 2=bottone centrale) o i rispettivi ta-
sti simulati sulla keyboard

condiz : stringa con la stessa struttura e significato del parametro
"contasto" di POLI_MENU (con due differenze: 1) e' sufficiente
racchiudere l'espressione logica fra virgolette (senza parente-
si); 2) nessun "tasto" e' di default)

Ritorno : il codice ASCII del tasto premuto (0 = mouse)

Esempio : ret = CONT_WAIT("m_release(2)","tasto=0.or.(tasto>=65.and.
tasto<=90)")

Resta in attesa che venga rilasciato il bottone centrale del mou-
se oppure premuto un tasto con lettera alfabetica maiuscola.

Valori possibili di ret : 0,65,66,...90

- Funzione **MYMESSAGE**(arram,colore)

Crea una "finestra" al cui interno scrive un messaggio, costituito al mas-
simo da 19 righe, ciascuna delle quali lunghe al massimo 69 caratteri. Ogni
riga viene centrata automaticamente. Eseguita la scrittura del messaggio,
il programma resta in attesa che si prema il bottone centrale del mouse
oppure il tasto ESC; dopodiche' la finestra scompare e lo schermo viene
ripristinato nelle condizioni precedenti

Parametri :

arram : array di stringhe del messaggio (max. 19 elementi). Ogni elemento
corrisponde a una riga

colore : indice di colore (foreground+background) del messaggio
(il background riempie l'intera finestra)

Ritorno : 0 (e' stato premuto il bottone centrale del mouse), oppure
27 (e' stato premuto il tasto ESC)

- Funzione **MYROLOC**(drit)

Calcola l'indice del colore inverso di un colore dato (la analoga fun-
zione ROLOC della libreria FUNCKY e' errata!), cioe' inverte il colore di
foreground con quello di background

Parametri :

drit = indice del colore da invertire

Ritorno : indice del colore invertito

- Funzione **MYSETTRAP**(proc0,proc1,proc2)

Definisce delle "trappole", cioe' delle procedure, scritte dall'utente, da
mandare in esecuzione ogni volta che viene premuto un bottone del mouse
mentre il programma e' in stato di attesa di input. Le "trappole" sono poi
attivate con la funzione (della libreria FUNCKY) M_TRAPSET() e rilasciate
con la funzione M_TRAPFREE(). MYSETTRAP fornisce la possibilita' di asse-
gnare al programma percorsi diversi a secondo di quale bottone del mouse
venga premuto. In POLI_MENU, per esempio, questa tecnica e' usata per atti-
vare (via mouse) le funzioni di "scrolling".

Parametri :

proc0 : se e' una stringa nulla, azzerà la "trappola" relativa al bottone
di sinistra del mouse; altrimenti definisce il nome di una proce-
dura-utente da eseguire se viene premuto il bottone di sinistra
del mouse mentre il programma e' in stato di attesa di input. La
procedura deve prevedere, per motivi di sistema, tre parametri
"dummy".

proc1 : idem, come sopra, ma relativamente al bottone di destra

proc2 : idem, come sopra, ma relativamente al bottone centrale

Ritorno : nullo

- Funzione **MYTRAPSAVE()**

Salva la situazione delle "trappole" esistenti in una stringa alfanumerica. E' consigliabile eseguire questa funzione prima di ridefinire le trappole con la MYSETTRAP.

Parametri : nessuno

Ritorno : la stringa alfanumerica in cui deve essere salvata la situazione esistente

- Funzione **MYTRAPREST(salvat)**

Ripristina una situazione di "trappole" precedentemente salvata

Parametri : la stringa in cui e' stata salvata la situazione precedente

Ritorno : nullo

Esempio di utilizzo combinato delle ultime tre funzioni :

```
saved = MYTRAPSAVE()
MYSETTRAP("left_proc","right_proc","")
M_TRAPSET()
M_CSRON()      (accende il cursore del mouse)
CONT_WAIT("m_inbutton(2)","tasto=0")
M_CSROFF()     (spegne il cursore del mouse)
M_TRAPFREE()
MYTRAPREST(saved)
.....
.....
PROCEDURE LEFT PROC
parameters d1,d2,d3
.....
return
.....
.....
PROCEDURE RIGHT PROC
parameters d1,d2,d3
.....
return
```

Spiegazione:

mentre il programma e' in attesa che si prema il bottone centrale del mouse, e' possibile fargli eseguire la procedura LEFT_PROC se si preme il bottone di sinistra, o la procedura RIGHT_PROC se si preme il bottone di destra.