a cura di Valter Di Dio

LS

di Pasquale d'Andreti - Riardo (CE)

Questa volta accontentiamo chi si ostina ad usare i comandi di UNIX anche quando lavora sul suo PC sotto MS-Dos (con le prevedibili conseguenze). Il programma che pubblichiamo permette infatti di emulare il comando «ls» di UNIX (con i relativi switch) per leggere le nostre abituali directory in un modo più completo e versatile. Chissà che non sia il primo mattone di una serie di utility che ci porterà alla fine ad avere una specie di interprete dei comandi stile UNIX?
Per chi invece usa il proprio computer per cose più serie (leggi giochi!) ecco una bellissima simulazione di guerra ambientata durante la Grande Guerra (con qualche concessione per migliorare il gioco). Al solito non è possibile pubblicare il listato in quanto troppo lungo; inoltre mancherebbero le immagini e i font.
Si gioca da soli contro il computer, mentre sarebbe stato bello poter avere più concorrenti in competizione, magari con uno gestito dalla macchina.
Comunque c'è il sorgente e non dovrebbe essere difficile implementarne una versione «multi-utente»

Sono disponibili, presso la redazione, i dischi con i programmi presentati in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 263

Il programma ls.c serve a mostrare a video, oppure su qualsiasi periferica che supporti la redirezione dell'output, il contenuto della directory specificata come argomento nello stile dell'ls UNIX, compatibilmente con le caratteristiche peculiari dell'MS-DOS.

Il programma interpreta l'argomento che specifica la directory allo stesso modo del DOS, cosa che invece non fa l'ls originale UNIX. Non intendo entrare nei dettagli del flusso del programma in quanto non presenta particolari difficoltà ed è comunque ampiamente commentato nel listato del sorgente stesso.

Mi soffermerò, invece, sugli aspetti più prettamente distintivi del sistema operativo in quanto ho codificato ls più per migliorare la mia conoscenza delle strutture dati utilizzate dal DOS che perché mancassero in giro procedure di questo tipo.

Proprio per questo motivo è stato d'obbligo scegliere il C in quanto, essendo il linguaggio con cui è stata scritta gran parte del codice delle ultime versioni di MS-DOS, è il più adatto per utilizzare il DOS senza perdere la potenza dell'Assembly ma continuando ad avere la chiarezza di un linguaggio ad alto livello. In particolare, ho utilizzato il Turbo C 2.0 per le sue notevolissime doti di velocità e facilità d'uso ma dovrebbe essere molto facile approntarne una versione per il compilatore Microsoft C 5.1.

In un programma del genere servono tre classi di sottoprocedure: nella prima vi sono quelle atte alla manipolazione dei percorsi con i quali vengono individuati i file, nella seconda si trovano le funzioni DOS che presiedono alla effettiva lettura dei contenuti delle directory mentre nella terza troviamo quelle relative alla lettura dei valori correnti attribuiti ai parametri di sistema.

Queste procedure sono fornite dal DOS principalmente per due motivi: per uniformare i protocolli di utilizzo delle funzioni di sistema e per fornire un ulteriore grado di astrazione nella scrittura dei programmi.

La prima classe di procedure si rende necessaria in quanto nel sistema operativo MS-DOS vi sono diversi modi per fare riferimento ad un file: c'è un modo «assoluto» nel quale il drive, il percorso assoluto, il nome e l'estensione del file sono specificati; vi sono poi tutta una serie di modi nei quali una o più delle componenti sopra citate dell'indirizzo del file sono omesse o specificate parzialmente.

Tanto per sintetizzare, anche se pochi non conoscono ancora queste cose, se voglio far riferimento al file il cui nome assoluto è:

```
C: \ USR \ SRC \ LS.C
```

e la directory corrente è C: \ potrò scrivere in uno dei seguenti modi:

```
USR \ SRC \ LS.C
C:USR \ SRC \ LS.C
C:\ USR \ SRC \ LS.C
```

mentre se la directory corrente è D: \C51 \ EXAMPLES mentre sul drive C: è C:\USR potrò scrivere:

```
C:SRC \ LS.C
C: \ USR \ SRC \ LS.C
```

Proprio per questa varietà di modi si è resa necessaria l'utilizzazione delle procedure fnmerge ed fnsplit che, rispettivamente, costruiscono un percorso completo a partire dalle singole componenti (ripeto, drive, percorso della directory, nome ed estensione del file) e, viceversa, suddividono un percorso completo nelle varie componenti. La seconda funzione indica anche quale componente è riuscita ad isolare e quale no. Ad esempio, nella seguente specifica:

```
C:LS
```

sono presenti solo le componenti relative al drive ed al nome del file mentre mancano l'estensione e il percorso della directory nella quale è contenuto LS. Questa prima classe di funzioni serve per interpretare l'argomento fornito come specifica dei file da elencare e per sostituire le componenti eventualmente mancanti con i valori correnti.

Le specifiche e le strutture dati utilizzate da queste funzioni sono nei commenti del sorgente.

La lettura fisica delle directory viene espletata dalla seconda classe di procedure, comprensiva di findfirst e findnext. La prima si occupa di cercare il primo (in ordine di memorizzazione) file contenuto nella directory specificata e i cui attributi siano consistenti con quelli forniti come argomento memorizzando le informazioni ad esso relative nella struttura di tipo ffblk commentata, come al solito, nel sorgente. La seconda trova i restanti file. Non ho detto ancora che, trattandosi della ricerca di più file, è possibile specificare i cosiddetti caratteri-jolly «*» e «?» il cui significato, com'è ben noto a tutti, è quello di sostituire, rispettivamente, una intera stringa qualsiasi (anche nulla) ed un solo carattere (non nullo).

La memorizzazione in strutture delle informazioni relative ai file si rende necessaria in quanto queste contengono dati eterogenei.

La terza classe è composta da getdisk e getdfree. Sono funzioni molto semplici delle quali la prima rileva il nome del drive corrente mentre la seconda restituisce, nella apposita struttura di tipo dfree, tutte le informazioni necessarie al computo della quantità di memoria totale e ancora libera disponibile sul drive dato come argomento. Come sempre i riferimenti sono sul listato del sorgente. Ribadisco che ls non è stato concepito come campione di flessibilità e potenza anche se garantisce sicuramente il funzionamento di tutte le opzioni ed il rispetto con la compatibilità, a livello di argomento, con la dir del DOS.

I margini di miglioramento ed ampliamento sono molti, a partire da un numero massimo di file trattabili maggiore (in questo caso dovrebbe esse compilato con il modello small), oppure si potrebbero inserire opzioni che ordinano su campi diversi dal nome o, ancora, informazioni più utili come l'occupazione totale dei file specificati e tante altre modifiche.

Si potrebbero studiare delle modifiche anche sul livello delle strutture dati per migliorarne l'occupazione in memoria e la duttilità d'impiego. Il programma è stato compilato con il compilatore di linea tcc in quanto produce un codice sensibilmente migliore di quello generato col compilatore interattivo.

Se si sono già impostate le variabili

LS

```
/*
        ls [-adlsOr] [nome file]:       elenca i file contenuti nelle directory
        Versione 1.02

        opzioni:
        -a:     include nella lista anche i file che hanno gli attributi di
                sistema, solo-lettura, etichetta impostati;
        -d: aggiunge, alla fine della lista dei file, lo spazio disponibile
                sul disco;
        -l: include nell' elenco la dimensione, la data, l' ora e gli attributi
                dei file;
        -s: elenca i file in ordine alfabetico normale;
        -O: elenca i file in ordine alfabetico inverso;
        -r: elenca anche i file nelle directory sottostanti alla data
                ricorsivamente.

        Pasquale D' Andreti MCMLXXXIX

*/

#include <dos.h>
#include <dir.h>
#include <stdio.h>
#include <string.h>
#include <process.h>
#include <ctype.h>
#include <stdlib.h>

#define         MAX_EL                  512
#define         MAX_DIR                 64
#define         VERO                    0x01
#define         FALSO                   0x00
#define         GIORNO                  0x001f
#define         MESE                    0x01e0
#define         ANNO                    0xfe00
#define         ORA                             0xf800
#define         MINUTO                  0x07e0
#define         SECONDO                 0x001f

/*
                MAXPATH, MAXDRIVE, MAXDIR, MAXFILE, MAXEXT, WILDCARDS, EXTENSION,
        FILENAME, DIRECTORY  e DRIVE sono definite in dir.h mentre FA_RDONLY,
        FA_HIDDEN, FA_SYSTEM, FA_LABEL, FA_DIREC e FA_ARCH sono definite in
        dos.h
*/

typedef         unsigned char   BOOL;

/*
                Definizione delle strutture FCB atte a contenere le informazioni dei
        file contenuti nelle directory.

                struct ffblk
                        {
                                char            ff_reserved[21];        riservati al DOS;
                                char            ff_attrib;                      byte contenente gli attributi
                                                                                        dei file;
                                unsigned        ff_fdate;                       parola contenente la data della
                                                                                        ultima modifica del file
                                unsigned        ff_ftime;                       parola contenente l' ora della
                                                                                        ultima modifica del file
                                long            ff_fsize;                       dimensione del file;
                                char            ff_name[13];            nome del file (comprensivo di
                                                                                        estensione);
                                };

                ff_fdate e ff_ftime sono le due parole di 16 bit contenenti le
        informazioni impaccate sulla data e l' ora dell' ultima operazione di
        scrittura effettuata sul file. Il formato è il seguente:

                                ff_fdate                                        :                               ff_ftime
        F E D C B A 9 8 7 6 5 4 3 2 1 0 : F E D C B A 9 8 7 6 5 4 3 2 1 0
         anno - 1980 : mese : giorno    :     ora    : minuto  : secondo / 2
*/

struct ffblk    cib[MAX_EL];
struct ffblk    cibo;

/*
                La struttura di tipo dfree contiene le informazioni sullo spazio
        libero disponibile sul drive specificato da getdfree.

                struct dfree
                        {
                                unsigned        df_avail;               cluster disponibili;
                                unsigned        df_total;               cluster totali sul drive;
                                unsigned        df_bsec;                byte per settore;
                                unsigned        df_sclus;               settori per cluster;
                                };
*/
struct dfree    bytes_rim;

int                             ordine;

/*
                Prototipi delle funzioni.
*/
void    main(int, char **);
void    ctoa(char *, char);
int             confr_el(struct ffblk *, struct ffblk *);


/*
                main
*/

void main(argc, argv)
int             argc;
char    **argv;
{
register unsigned int   num_el, i, file_tot;
BOOL                                    tutti, lista_est, err_arg;
BOOL                                    spazio_lib, ordinati, lista_ric;
int                                             el_pres, p_pila;
char                                    drive[MAXDRIVE];
char                                    percorso[MAXDIR];
char                                    nome_f[MAXFILE];
char                                    est[MAXEXT];
char                                    pila_perc[MAX_DIR][MAXPATH];
char                    perc_t[MAXPATH];
char                                    attributi[7];

tutti           =       FALSO;
lista_est       =       FALSO;
spazio_lib      =       FALSO;
ordinati        =       FALSO;
err_arg         =       FALSO;
lista_ric       =       FALSO;
```

(continua a pag. 252)

d'ambiente LIB e INCLUDE basterà dare il comando:

tcc -GaZkO -mt -f- -lxndct ls.c

mentre se tali variabili non sono state impostate e, ad esempio, le librerie risiedono in c: \ tc \ lib e i file da includere in c: \ tc \ include si dovrà digitare

tcc -GaZkO -mt -f- -lxndct -I<include directory>-L<lib directory> ls.c

Così facendo si specifica la massima ottimizzazione in velocità, l'utilizzo del modello di memoria tiny (64 Kb per dati e programma) è necessario per rendere il codice in formato non rilocabile (.COM), l'esclusione dell'emulatore in virgola mobile e si passano al link le istruzioni per non generare un file di mappa (.MAP), per non utilizzare le librerie usuali, per rispettare le minuscole e per generare direttamente un eseguibile non rilocabile senza passare per il programma di sistema EXE2BIN. Possedendo un computer con almeno un 80286, si può istruire il compilatore a generare codice specifico per tale processore sostituendo le opzioni '-GaZkO' con '-1GaZkO'. Un'ultima cosa: la lista completa dei 568 file presenti sul mio disco fisso ottenuta con

ls -r \

è stata visualizzata in 19 secondi mentre la stessa lista, ma con tutte le opzioni impostate

ls -adlor \

ha impiegato 55 secondi; il tutto su un AT a 6 MHz su cui gira il DOS 3.30 con il disco fisso da 60 ms.

(segue da pag. 251)

```
file_tot = 0;
ei_pres = 0;
/*
                Controlla la presenza dei parametri.
*/
if(argc > 1)
/*
                Se i parametri esistono e sono preceduti dal segno '-' ne prende atto
        modificando gli opportuni flag.
*/
        if(argv[1][0] == '-')
                {
                i = 0;
                while(argv[1][++i])
                        {
                        if(argv[1][i] == 'l')
/*
                Flag lista_est: si richede la presentazione estesa della directory.
*/
                                {
                                lista_est = VERO;
                                continue;
                                }
                        if(argv[1][i] == 'a')
/*
                Flag tutti: si richede la inclusione dei file con gli attributi di
        sistema, di 'nascosto' e di volume nell' elenco.
*/
                                {
                                tutti = VERO;
                                continue;
                                }
                        if(argv[1][i] == 'd')
/*
                Flag spazio_lib: si richede il conteggio dei byte rimanenti sul disco.
*/
                                {
                                spazio_lib = VERO;
                                continue;
                                }
                        if(argv[1][i] == 'o')
/*
                Flag ordinati: si richede l' ordinamento dell' elenco in senso
        ascendente.
*/
                                {
                                ordinati = VERO;
                                ordine = 1;
                                continue;
                                }
                        if(argv[1][i] == 'O')
/*
                Flag lista_est: si richede l' ordinameto dell' elenco in senso
        discendente.
*/
                                {
                                ordinati = VERO;
                                ordine = -1;
                                continue;
                                }
                        if(argv[1][i] == 'r')
/*
                Flag lista_ric: si richede l' elenco dei file risiedenti in tutte le
        sottodirectory presenti in quella specificata.
*/
                                {
                                lista_ric = VERO;
                                continue;
                                }
/*
                Flag err_arg: viene impostato se i parametri specificati non sono
        permessi.
*/
                        err_arg = VERO;
                        break;
                        }
                tutti = tutti ? FA_HIDDEN | FA_SYSTEM | FA_LABEL : tutti;
/*
                Suddivide l' eventuale percorso fornito come secondo argomento sulla
        linea di comando nei nome dei drive, suffisso del percorso, nome del
        file e sua estensione restituendo un intero nel quale i primi 5 bit
        rappresentano la presenza o meno delle varie componenti nella stringa da
        sezionare.
*/
                if(argc > 2)
                        ei_pres = fnsplit(argv[2], drive, percorso, nome_f, est);
                }
        else
/*
                Vedere commento precedente (primo argomento).
*/
                ei_pres = fnsplit(argv[1], drive, percorso, nome_f, est);
/*
                Se i parametri sono errati, emette un messaggio di aiuto ed esce.
*/
if(err_arg)
        {
        fputs("Uso: ls [-adlOor][modello]", stderr);
        exit(-1);
        }
/*
                Rimpiazza le parti di percorso non trovate da fnsplit con valori di
        default: il drive corrente è ottenuto con getdisk che restituisce il
        valore numerico associato ad ogni drive (0 = A, 1 = B ...).
*/
if(!(ei_pres & DRIVE))
        {
        drive[0] = getdisk() + 'a';
        drive[1] = ':';
        drive[2] = '\0';
        }
/*
                La directory corrente è ottenuta con getcurdir che restituisce una
        stringa SENZA backslash iniziale e finale.
*/
if(!(ei_pres & DIRECTORY))
        {
        getcurdir((int) (toupper(drive[0]) - 'A' + 1), percorso);
        strcpy(pila_perc[0], "\\");
        strcat(pila_perc[0], percorso);
        if(percorso[0] != '\0')
                strcat(pila_perc[0], "\\");
        strcpy(percorso, pila_perc[0]);
        }
/*
                Assume come default la 'wild-card' '*' per il nome del file e per
        la sua estensione.
```

War

di Luca Cammisa - Catania

Il gioco che vi propongo è una fedele, dettagliata e realistica riproduzione del primo conflitto mondiale, ristretto, impacchettato ed immagazzinato in «una breve, ma efficace e gradevole simulazione». Simile alquanto al classico seppur indimenticabile Risiko, svelta nell'elaborazione dei dati, compatta e versatile nella riproduzione geopolitica dell'Europa, ed efficace nel contatto con l'utente, si tratta or dunque di una guerra

```
*/
if(!(el_pres & FILENAME))
        strcpy(nome_f, "*");
if(!(el_pres & EXTENSION))
        strcpy(est, ".*");

/*
                Inizializza il puntatore alla cima della pila ed inserisce, come
        primo elemento, il percorso costruito finora. fnmerge si comporta in
        modo opposto a fnsplit riunendo le varie componenti del percorso in
        una sola stringa.
*/
p_pila = 0;
fnmerge(pila_perc[p_pila++], drive, percorso, nome_f, est);
/*
                Ciclo principale di lettura delle directory ricorsivamente. Nel caso
        non si fosse specificato il paramentro -r questo ciclo viene effettuato
        una sola volta.
*/
while(p_pila > 0 && p_pila < MAX_DIR)
        {
        num_el = 0;
        strcpy(perc_t, pila_perc[--p_pila]);
/*
                Ciclo che ricerca i file da elencare presenti nella directory e li
        memorizza nel vettore di strutture cib che contiene tutte le informazioni
        necessarie alla identificazione dei singoli file. findfirst ricerca il
        primo file presente nella directory specificata che rispetta i parametri
        forniti e findnext trova i seguenti.
*/
        if(findfirst(perc_t, &cibo,
                        FA_RDONLY | FA_DIREC | FA_ARCH | tutti) == 0)
                do
                        {
                        cib[num_el++] = cibo;
                        if((cibo.ff_attrib & FA_DIREC) && lista_ric &&
                                        cibo.ff_name[0] != '.')
                                {
                                fnsplit(perc_t, drive, percorso, nome_f, est);
                                strcat(percorso, cibo.ff_name);
                                strcat(percorso, "\\");
                                fnmerge(pila_perc[p_pila++], drive, percorso, nome_f, est);
                                }
                        }
                while(num_el < MAX_EL && findnext(&cibo) == 0);
/*
                Se è stato specificato il parametro -o oppure -O provvede ad ordinare
        i file memorizzati per nome. L' ordine è specificato dal flag ordine
        (-1 = discendente, +1 = ascendente).
*/
        if(ordinati)
                qsort(cib,(size_t) num_el,(size_t) (sizeof(cib[0])), confr_el);
/*
                Se è stato specificato il parametro -r premette all' elenco dei file
        la directory dove essi risiedono.
*/
        if(lista_ric)
                printf("\nDirectory di %s\n", perc_t);
/*
                Ciclo di stampa dei file trovati nella directory.
*/
        for(i = 0; i < num_el; i++)
                {
                printf("%-12s", cib[i].ff_name);

                Se è stato specificato il parametro -l fornisce informazioni
        aggiuntive sui file utilizzando ctoa che formatta in maniera leggibile
        i suoi attributi DOS (h = hidden, s = system, a = archive, l = label,
        r = read-only, d = directory).
*/
                if(lista_est == VERO)
                        {
                        ctoa(attributi, cib[i].ff_attrib);
                        printf("%10ld  %02u/%02u/%04u  %02u:%02u:%02u  %6s",
                                cib[i].ff_fsize,
                                (unsigned int) (cib[i].ff_fdate & GIORNO),
                                (unsigned int) ((cib[i].ff_fdate & MESE) >> 5),
                                (unsigned int) ((cib[i].ff_fdate & ANNO) >> 9) + 1980,
                                (unsigned int) ((cib[i].ff_ftime & ORA) >> 11),
                                (unsigned int) ((cib[i].ff_ftime & MINUTO) >> 5),
                                (unsigned int) ((cib[i].ff_ftime & SECONDO) << 1),
                                attributi);
                        }
                putchar('\n');
                file_tot++;
                }
        }
/*
                Se è stato specificato il parametro -d, viene visualizzato lo spazio
        rimanente (in bytes) del disco dal quale sono stati elencati i file.
*/
if(spazio_lib)
        {
        getdfree(toupper(drive[0]) - 'A' + 1, &bytes_rim);
        if((signed int) (bytes_rim.df_sclus) != -1)
                printf("\nSpazio disponibile sul drive %s %9ld bytes\n",
                drive,
                ((long) bytes_rim.df_avail) * ((long) bytes_rim.df_bsec) *
                        ((long) bytes_rim.df_sclus));
        }
/*
                Per finire, esce a DOS restituendo nel codice di uscita del processo
        il numero totale dei file elencati. Può sempre essere utilizzata da un
        processo chiamante oppure da DOS con ERRORLEVEL.
*/
exit(file_tot);
}

void ctoa(str_dest, attr)
char *str_dest;
char attr;
{
register int i;

strcpy(str_dest, "adlshr");
for(i = 0; i < 6; i++)
        if(!((attr >> i) & 1))
                str_dest[5 - i] = '-';
}

int confr_el(el1, el2)
struct ffblk *el1;
struct ffblk *el2;
{
        return(strcmp(el1->ff_name, el2->ff_name) * ordine);
}

/* eof ls.c */
```

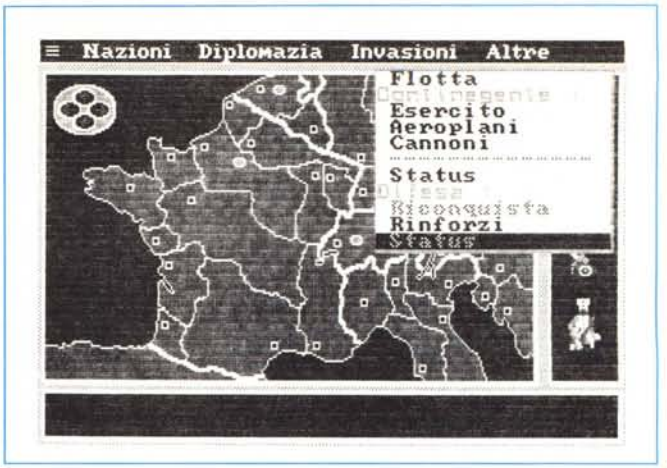
simulata in un «vile» ma notevole calcolatore quale un IBM, che si svolge secondo le concezioni tattiche e strategiche d'inizio secolo, con le macchine belliche dell'epoca, ed altresì con la mentalità cavalleresca del tempo, ben diversa dalle direttive che condussero la politica dei tempi successivi.

Il titolo di questa magnificente passerella di invasioni, trattati, tregue e rivolte, di per sé stesso è già tutto dire: Scenario di guerra in Europa. Ciò significa che la protagonista per antonomasia è l'Europa, la culla della nostra civiltà occidentale: la Triplice Intesa e l'Alleanza degli Imperi Centrali, le nazioni della nostra amata terra in cui decenni or sono si svolsero quei titanici scontri di armate, guidate dal solo patriottismo nazionale, se non pur dalla vanagloria dell'avversario.

L'ambiente in cui si svolge tale simulazione è una vasta e particolareggiata mappa dell'Europa, suddivisa in imperi, nazioni e staterelli di minore importanza, ognuno dei quali minuziosamente schematizzato in regioni, città e capitali, con centinaia di corpi d'armata, cannoni, macchine belliche d'ogni sorta, ed aeroplani stanziati in ognuna di esse.

Tale carta geopolitica è incastonata in un preziosissimo pannello di controllo, su cui gestire ogni spedizione bellica in territorio nemico, ogni relazione diplomatica con l'alleato o con l'avversario, per mezzo del quale ottenere qualsiasi informazione circa l'Europa in guerra su larga scala, nonché nel particolare, fino, addirittura, a penetrare nella più intima frazione del continente. Il tutto è gestito dall'immancabile barra di menu, piacevolmente dotata, sulla scia dello stile Amiga, dei più arditi attributi, classici del mondo Macintosh e Commodore; dalla agile freccia grazie alla quale in poche frazioni di secondo siamo in grado di spostarci nel continente coprendo incredibili distanze; da una esigua sezione delle schermo in cui una vasta gamma di informazioni ci viene propinata spesso con una sconvolgente rapidità; ed infine da un abile e veloce sistema di finestre, adibite, il più delle volte, a qualsiasi uso immaginabile ed ottenibile.

Lo scopo del gioco senz'altro è vincere: sbaragliare ed annichilire le forze dell'avversario, demolire le sue più ardite e fantasiose prospettive, conquistare ogni capitale sottomettendo popoli interi, offrire miseri trattati di pace dalle condizioni più umilianti, incitare rivolte e guerre intestine in territorio nemico, inviare gli agenti più dotati per avere le informazioni più preziose circa i piani strategici dell'avversario, di modo che si possa dedurre se non prevenire ogni



Due significative immagini di War.

sua imponente invasione. Questa è la linea politico-bellica che conduce le sorti dell'Europa in questa realistica simulazione; ed in questo caso tale destino è nelle mani di chi, abile nel calcolo delle perdite, scaltro nello scegliere le terre da conquistare o da cedere all'avanzare del nemico, si accinge a ricostruire idealmente la storia del nostro continente.

Descrizione del programma

Lo schermo di gioco è suddiviso in quattro sezioni, di cui solo tre, in vero, sono realmente funzionanti. La prima è adibita ai menu, cinque per l'esattezza; la seconda, come ho già accennato, è dedicata all'Europa, sezionata, nel gioco, in sei finestre; invece la terza accoglie una minuta finestra di «Remark», che serve, appunto, alle brevi informazioni circa l'intervento del nemico (che si scandiscono spesso con una rapidità «invadente»), od anche a rari ma preziosi sotto-menu su stampo didascalico.

I menu

Il primo menu (rappresentato con uno strano ma pur loquace simbolo) oltre a servire da «Info», permette altresì all'utente di memorizzare sul disco del gioco la situazione attuale e, immancabilmente, di richiamarla per continuare lo svolgersi del conflitto; di cambiare i colori dello schermo; di abbandonare il programma per tornare in ambiente MS DOS, ed infine di «iniziare il conflitto». Il gioco, infatti, si scandisce in due momenti: all'inizio le proprie armate sono disposte uniformemente nei territori nazionali, senza, appunto, tener conto dei piani del nemico e, di conseguenza, delle regioni che sono più a rischio; è quindi compito dell'utente disporre le armate nei territori più bisognosi grazie alle informazioni delle spie; quando, invece, il giocatore ritiene di aver posizionato le proprie armate in maniera equa ed opportuna, può selezionare la su citata opzione, notando, quindi, dalla spia del drive acceso che l'avversario sta facendo la medesima cosa...

Il secondo menu (Nazioni) è di uso prevalentemente «prebellico». Permette di avere informazioni sull'Europa o su una singola nazione alleata o avversa riguardanti le probabilità di belligeranza, l'esercito, l'aviazione, la flotta, le armi belliche, e via dicendo. Scopo secondario, ma non per questo in scala di importanza, è quello di selezionare una nazione allo scopo di intraprendere una procedura diplomatica pro o contro di essa (vedi menu Diplomazia).

Il terzo menu (Diplomazia), breve ed altrettanto complesso, può far sì che il giocatore (1) dichiari guerra a una nazione, (2) le offra un trattato di pace, (3) inciti ribellioni all'interno di essa, (4) esorti un alleato non belligerante ad entrare in guerra ed altresì un avversario a passare dalla propria parte, (5) di avere importanti informazioni strategiche sui piani del nemico (nel pre-guerra) o di conoscere le azioni dell'avversario non altrimenti ottenibili.

Il quarto menu (Invasioni) è il più frequentato. Dopo aver selezionato sulla cartina geografica una regione, per invaderla bisogna innanzitutto selezionare la regione da cui invadere (confinante!) (1' opz.), successivamente scegliere il contingente bellico (2,3,4' opz.), ed infine attaccare. Il gioco gestisce fino a 30 Invasioni contemporaneamente, calcolando battaglia per battaglia le perdite del giocatore e dell'avversario ed informando il primo solo in caso di vittoria o di sconfitta. Pertanto è consigliabile rendersi conto dell'evolversi della situazione attraverso la 6' opzione, mandare rinforzi dalla regione da cui è partita l'invasione in corso con la 7' opzione, o, in caso di perdite gravosissime, ritirarla con l'8' opzione. Per selezionare un'invasione o si sceglie prima la regione invasa e poi con la 1' opzione la regione che invade, oppure si sfrutta la 7' opzione che presenta un quadro completo delle invasioni in corso.

Il quinto menu (Altre), infine, permette di scegliere quale sia la parte della flotta militare da impegnare in guerra (sempre che lo stato selezionato ne abbia una!), di avere informazioni sul contingente bellico stanziato nella regione selezionata, di riconquistare regioni sconfitte, di mandare rinforzi a una regione presidiata (il procedimento è simile a quello dell'invasione), ed infine, «dulcis in fundo» , di avere uno «status quo» della zona di Europa in memoria (Teste di ponte, Presidi, Possedimenti, Conquiste e Sconfitte).

La freccia e la carta geografica

Ci si può muovere attraverso l'Europa grazie alle quattro frecce in alto allo schermo, si può selezionare una regione «clickando» sul quadratino chiaro che la contrassegna, ed infine si può entrare in un menu premendo il tasto Alt seguito dalla lettera iniziale del nome del menu (per il primo, visto che è simbolicamente rappresentato da un carattere bizzarro quantomai raro, basta solo la pressione del tasto Esc).

L'avversario

L'avversario, cioè il computer, simula una sorta di intelligenza artificiale che, calcolando lo stato geopolitico dell'Europa centinaia e centinaia di volte durante una sola partita, prende ogni decisione non secondo uno schema predefinito, bensì in base ai dati ottenuti dalla schematizzazione politica dell'Europa che,

come ho già detto, avviene ogni qual volta viene ristampata la freccia sullo schermo dopo aver intrapreso qualsivoglia azione, vale a dire un numero indefinito di volte.

Questo è uno degli aspetti più interessanti del gioco, in quanto l'utente può star sempre certo che l'avversario è in realtà sempre pronto ad approfittare di ogni sua svista; che il computer, mentre il suo oppositore spesso agisce senza un piano razionale, realizza ogni sua strategia bellica il più delle volte dopo aver valutato con prudenza ogni probabilità di vittoria e di sconfitta; e che l'avversario, inoltre, fa sempre il possibile per salvare ogni suo ettaro di terreno dall'oppressione del nemico. Il programma, pertanto, cerca con ogni mezzo a disposizione e di frenare l'avanzare delle milizie antagoniste, e di stroncare le forze del nemico ovunque esso sia debole ed indifeso...

D'altronde il computer può fare tutto ciò che il giocatore può intraprendere, tranne trasgredire ogni legge che non vada al di fuori del «codice cavalleresco», come, ad esempio, invadere uno stato neutrale; non conosce reazioni emotive irrazionali, pertanto quando si vede sconfitto non manda allo sbaraglio le proprie milizie, bensì accetta la pace; e così via.

La rapidità con cui agisce e/o reagisce dipende dal livello di gioco (Principiante, Medio, Avanzato) che viene scelto insieme all'Alleanza per cui giocare nel programma di «Intro» del gioco.

Particolarità

Il programma fa uso rarissimo del font di sistema, ed invece sfrutta a pieno le capacità del compilato stampando «pixel per pixel» ogni carattere secondo una matrice propria del font in memoria. Quindi, rilocando un font qualsiasi in una zona libera della RAM, è possibile stampare qualsiasi testo con diverse «calligrafie».

Il gioco occupa appena un disco, in quanto funziona con l'incartapecorita seppur veloce e maneggevole CGA, e credo che ciò sia un punto a favore per l'utente.

Una caratteristica peculiare del gioco è che è interamente in italiano. Quantunque il suo nome possa ingannare, vi assicuro che ogni parola prodotta dal programma è squisitamente italiana, ad eccezione, ad onor del vero, del semplice ma loquace «QUIT» la cui sola brevità mi ha indotto ad introdurlo.

Per motivi di simmetria all'interno del programma ho dovuto introdurre l'intervento in guerra della Svezia e della Norvegia (che eresia!), ed eliminare quello degli USA, aggiungendo persino ribellioni in Francia, Austria-Ungheria e Turchia oltre che, come è d'uopo, in Russia. MK

Note

I sorgenti sono stati compilati con il **QUICKBASIC** v. 3.00 con le sole opzioni **MINIMIZE STRING DATA, ON ERROR, COMPILE** with **BCOM30.LIB**, e **OPTIMIZATION OPTIONS: SPEED**. Successivamente sono stati linkati con il linker della stessa versione. Il batch file di start è **WAR.BAT**.