

**EMILIANO VEZZOLI**

**IL PROGETTO**





Autore:..... **Ing. Emiliano Vezzoli**  
Versione: ..... **1.1 – Settembre 2003**  
Copyright: ..... **EVGames - © 2003 – Tutti i diritti sono riservati**

Successive versioni:

2002   **2003**   2004   2005   2006

**“Bisogna sempre giocare lealmente quando si hanno in mano le carte vincenti”**  
(Oscar Wilde)

## PREFAZIONE

Il Progetto "Gameisland" nasce nel secondo semestre del 2002 con l'obiettivo di raggruppare in un unico prodotto tutti i videogiochi realizzati con il marchio EVGames a partire dal 1995.

In questo documento si può ripercorrere la storia di ogni videogame attraverso la corrispondente scheda tecnica, corredata di immagini e informazioni su software utilizzati, trucchi e suggerimenti, le regole, le istruzioni, i codici sorgente degli oggetti principali. Rappresenta un valido strumento di riferimento per chi, come me, ha intrapreso o intende farlo la strada dello sviluppo di software dedicato all'intrattenimento.

*Emiliano Vezzoli*

## INDICE

### Capitolo 1: Gameisland

1.	Gameisland .....	1
1.2	Il modulo di registrazione online .....	1
1.2.2	La comunicazione attraverso la rete .....	2
1.2.3	Client-Side Server-Side programming .....	3
1.2.4	La tecnologia Common Gateway Interface (CGI) .....	5
1.2.5	La tecnologia ASP (Active Server Pages) .....	8
1.3	Il programma di installazione .....	18

### Capitolo 2: Star Trek Giovani Ufficiali

2.	Scheda tecnica .....	20
2.1	Regole del gioco.....	20
2.2	Trucchi e suggerimenti .....	20
2.3	La realizzazione (The Making Of).....	20
	□ La storia .....	20
	□ I software utilizzati .....	21
	□ Bozze e screenshot.....	21
2.4	Il codice.....	26

### Capitolo 3: Acquatica

3.	Scheda tecnica .....	45
3.1	Regole del gioco.....	45
3.2	Trucchi e suggerimenti .....	45
3.3	La realizzazione (The Making Of).....	45
	□ La storia .....	45
	□ I software utilizzati .....	46
	□ Bozze e screenshot.....	46
3.4	Il codice.....	50

### Capitolo 4: Toon Music Game

4.	Scheda tecnica .....	57
4.1	Regole del gioco.....	57
4.2	Trucchi e suggerimenti .....	58
4.3	La realizzazione (The Making Of).....	58
	□ La storia .....	58
	□ I software utilizzati .....	59
	□ Bozze e screenshot.....	59
4.4	Il codice.....	62

## Capitolo 5: Java Games Virtual Console

5.	Scheda tecnica .....	70
5.1	Regole del gioco.....	73
5.2	Trucchi e suggerimenti .....	73
5.3	La realizzazione (The Making Of).....	73
	□ La storia .....	73
	□ I software utilizzati .....	74
	□ Bozze e screenshot.....	74
5.4	Il codice.....	78

## Capitolo 6: Cinemusik III

6.	Scheda tecnica .....	82
6.1	Regole del gioco.....	82
6.2	Trucchi e suggerimenti .....	83
6.3	La realizzazione (The Making Of).....	83
	□ La storia .....	83
	□ I software utilizzati .....	84
	□ Bozze e screenshot.....	84
6.4	Il codice.....	87

## Capitolo 7: Caccia all'Anello di Sauron

7.	Scheda tecnica .....	95
7.1	Regole del gioco.....	95
7.2	Trucchi e suggerimenti .....	95
7.3	La realizzazione (The Making Of).....	95
	□ La storia .....	95
	□ I software utilizzati .....	96
	□ Bozze e screenshot.....	96
7.4	Il codice.....	98

## Capitolo 8: SuperSecchione

8.	Scheda tecnica .....	107
8.1	Regole del gioco.....	107
8.2	Trucchi e suggerimenti .....	107
8.3	La realizzazione (The Making Of).....	108
	□ La storia .....	108
	□ I software utilizzati .....	109
	□ Bozze e screenshot.....	109
8.4	Il codice.....	113

## Capitolo 9: Up & Down 3 – Guerre Stellari

9.	Scheda tecnica .....	120
9.1	Regole del gioco.....	120
9.2	Trucchi e suggerimenti .....	120
9.3	La realizzazione (The Making Of).....	121
	□ La storia .....	121
	□ I software utilizzati .....	122
	□ Bozze e screenshot.....	122
9.4	Il codice.....	124

## Capitolo 10: TV-Series Musical Quiz

10.	Scheda tecnica.....	133
10.1	Regole del gioco .....	133
10.2	Trucchi e suggerimenti .....	134
10.3	La realizzazione (The Making Of) .....	134
	□ La storia .....	134
	□ I software utilizzati .....	135
	□ Bozze e screenshot.....	135
10.4	Il codice .....	138

## Capitolo 11: Voyager Shuttle Memory

11.	Scheda tecnica.....	146
11.1	Regole del gioco .....	146
11.2	Trucchi e suggerimenti .....	146
11.3	La realizzazione (The Making Of) .....	146
	□ La storia .....	146
	□ I software utilizzati .....	147
	□ Bozze e screenshot.....	147
11.4	Il codice .....	150

## Capitolo 12: Lader – Lo Sciacallo

12.	Scheda tecnica.....	156
12.1	Regole del gioco .....	156
12.2	Trucchi e suggerimenti .....	156
12.3	La realizzazione (The Making Of) .....	156
	□ La storia .....	156
	□ I software utilizzati .....	157
	□ Bozze e screenshot.....	157
12.4	Il codice .....	160

## Capitolo 13: L'Oracolo del Nilo

13.	Scheda tecnica.....	179
13.1	Regole del gioco .....	179
13.2	Trucchi e suggerimenti .....	179
13.3	La realizzazione (The Making Of) .....	179
	□ La storia .....	179
	□ I software utilizzati .....	180
	□ Bozze e screenshot.....	180
13.4	Il codice .....	183

## Capitolo 14: Il Cervellone - Mastermind

14.	Scheda tecnica.....	188
14.1	Regole del gioco .....	188
14.2	Trucchi e suggerimenti .....	188
14.3	La realizzazione (The Making Of) .....	188
	□ La storia .....	188
	□ I software utilizzati .....	189
	□ Bozze e screenshot.....	189
14.4	Il codice .....	191

## Capitolo 15: The Dragon Simon

15.	Scheda tecnica.....	205
15.1	Regole del gioco .....	205
15.2	Trucchi e suggerimenti .....	205
15.3	La realizzazione (The Making Of) .....	205
	□ La storia .....	205
	□ I software utilizzati .....	206
	□ Bozze e screenshot.....	206
15.4	Il codice .....	208

## Capitolo 16: Guardie e Ladri

16.	Scheda tecnica.....	222
16.1	Regole del gioco .....	222
16.2	Trucchi e suggerimenti .....	222
16.3	La realizzazione (The Making Of) .....	223
	□ La storia .....	223
	□ I software utilizzati .....	223
	□ Bozze e screenshot.....	223
16.4	Il codice .....	226

## Capitolo 17: Il Bruco

17.	Scheda tecnica.....	231
17.1	Regole del gioco .....	231
17.2	Trucchi e suggerimenti .....	231
17.3	La realizzazione (The Making Of) .....	232
	□ La storia .....	232
	□ I software utilizzati .....	232
	□ Bozze e screenshot.....	232
17.4	Il codice .....	235

## Capitolo 18: MicroTetris

18.	Scheda tecnica.....	241
18.1	Regole del gioco .....	241
18.2	Trucchi e suggerimenti .....	241
18.3	La realizzazione (The Making Of) .....	242
	□ La storia .....	242
	□ I software utilizzati .....	242
	□ Bozze e screenshot.....	242
18.4	Il codice .....	245

## Capitolo 19: SpiderBuster

19.	Scheda tecnica.....	252
19.1	Regole del gioco .....	252
19.2	Trucchi e suggerimenti .....	252
19.3	La realizzazione (The Making Of) .....	253
	□ La storia .....	253
	□ I software utilizzati .....	254
	□ Bozze e screenshot.....	254
19.4	Il codice .....	257

## 1. GAMEISLAND



Fig. 1.1: il menu di navigazione Gameisland

Il menu di navigazione "Gameisland" (vedi fig.1.1) è stato ideato sotto forma di isola virtuale dove selezionare i vari videogiochi della raccolta. Al passaggio del mouse sopra un titolo, questo si attiva visualizzando sul display inferiore la scheda del videogame associato, con la descrizione ed uno screenshot.

Per giocare basta cliccare sulla voce desiderata. Per le informazioni su autore e release occorre premere il tasto "INFO", mentre per tornare al sistema operativo occorre premere "ESCI". Il tasto "I SEGRETI DI GAMEISLAND" permette di accedere a questo documento.

Anche Gameisland, come tutti i videogame della collezione, è stato implementato servendosi dell'ambiente di programmazione Multimedia Builder (vedi fig. 1.2).

### 1.2 IL MODULO DI REGISTRAZIONE ON LINE

Al solo scopo di redigere un censimento delle copie distribuite di "Gameisland" è stato studiato ed implementato un apposito modulo da compilare e spedire via internet

Inizialmente si era pensato di servirsi di una pagina web dedicata a questa funzione ma, in seguito, si è ritenuto più interessante creare un semplice client personalizzato. Il modulo di registrazione, in fig. 1.3, sfruttando le tecnologie CGI ed ASP (descritte nei paragrafi successivi), consente all'utente di inserire i propri dati nel modulo speciale "offline" e di spedirli solo in un secondo momento.

I dati ricevuti dal server e processati da uno script dinamico ASP vengono memorizzati in un database Microsoft Access poco prima che l'utente riceva conferma dell'avvenuta registrazione (vedi Fig. 1.6).

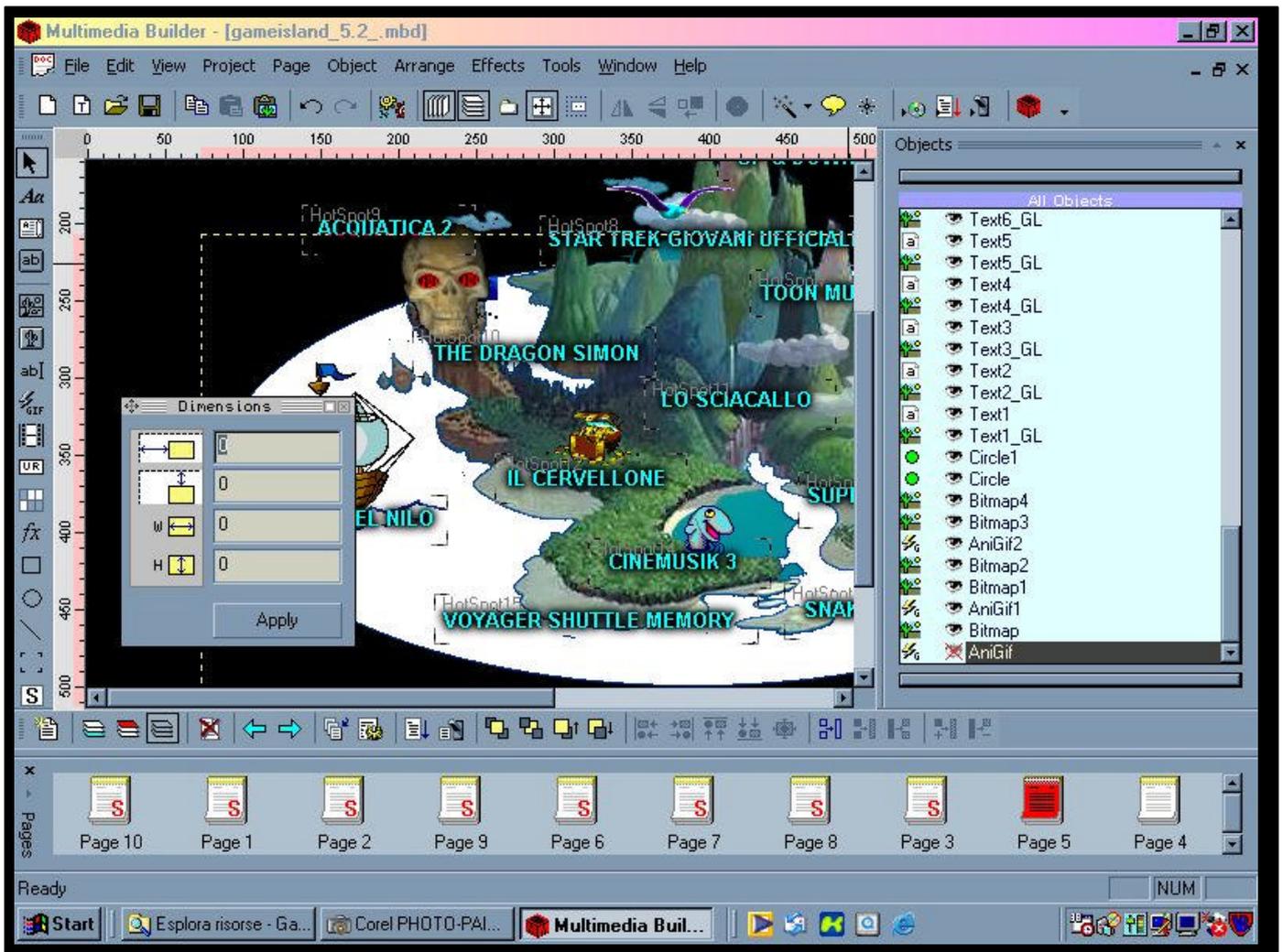


Fig. 1.2: un'immagine dell'ambiente di programmazione Multimedia Builder.

### 1.2.2 LA COMUNICAZIONE VIA WEB

Il modulo è stato implementato e compilato con l'ambiente di programmazione Multimedia Builder della società Mediachance, servendosi del "metodo GET" e dell'interfacciamento CGI, come descritto nel prossimo paragrafo. Di seguito riportiamo la parte di codice essenziale al suo funzionamento in rete:

```
Contact_Servizio$ = 'Gameisland'
url$='http://www.evweb.it/cgi-bin/registerGI.asp?Nome=' + Contact_Nome$
+ '&Cognome=' + Contact_Cognome$ + '&Indirizzo='
+ Contact_Indirizzo$ + '&Cap=' + Contact_Cap$ + '&Localita='
+ Contact_Localita$ + '&Provincia=' + Contact_Provincia$
+ '&Telefono=' + Contact_Telefono$ + '&Email=' + Contact_Email$
+ '&CodiceFiscale=' + Contact_Fiscale$ + '&Servizio='
+ Contact_Servizio$
Messaggio$='Verificare che la connessione internet sia attiva e premere invio'
Message("Attenzione: ", "Messaggio$")
Run("iexplore.exe", "url$")
```

 <span style="float: right;"><b>MODULO REGISTRAZIONE ONLINE</b> <b>GAMEISLAND</b></span>	
Nome	Mario
Cognome	Rossi
Indirizzo	Via Dante Alighieri, 5
Cap	
Località	
Provincia	
Telefono	
E-mail	
Codice Fiscale	
<p>In ottemperanza alla legge italiana 675/96 (Privacy), i dati forniti sono da considerarsi confidenziali e coperti da garanzia di riservatezza. Pertanto verranno utilizzati solo per l'espletamento delle procedure di registrazione, il cui unico scopo è quello di redigere un censimento del numero di copie distribuite del software. La registrazione non prevede l'invio di email pubblicitarie o di altra natura.</p>	
<input type="button" value="Invia"/>	
<input type="button" value="Esci"/>	

Fig. 1.3: Il modulo di registrazione online di Gameisland.

### 1.2.3 CLIENT-SIDE SERVER-SIDE PROGRAMMING

Il funzionamento classico di un'architettura Client-Server prevede che il Client, attraverso un software denominato Browser, effettui una richiesta ad un Server, il quale, dopo aver avuto accesso al proprio File System, risponde. Per sviluppare applicazioni web dinamiche esistono due tecniche di programmazione, chiamate: *Server-Side* e *Client-Side*.

- **Server-Side:** in questa modalità il Server, che riceve sempre la richiesta dal Client, cerca il programma nel suo File System, lo esegue (i file eseguibili sono chiamati *Back-End*), prende l'output del programma e lo invia come risposta al Client. Da notare che nel Server side il programma può essere scritto con qualsiasi linguaggio. E' evidente come eseguendo un'applicazione, ci sia la possibilità di interagire con i file richiesti e come questi acquisiscano dinamicità, con le più svariate possibilità che un programma può offrire, come per esempio l'aggiornabilità immediata, la consultazione di database, ecc.
- **Client-Side:** ci sono due diverse possibilità per questo tipo di architettura: nella prima il Client invia la richiesta al Server, questi risponde inviando un file contenente un programma che, quando la pagina viene visualizzata dal browser, è capace di attivare un'applicazione che già risiede sul Client, un elaboratore testi, un foglio elettronico o qualsiasi altro. Questa potrebbe

essere una strada percorribile ma, di fatto, è pressochè inutilizzata. Nella seconda le pagine HTML inviate al Client nascondono del software che viene eseguito ed il cui risultato viene mostrato dal browser. Ovviamente è necessario che il browser stesso sia in grado di eseguire il software accluso.

Nella Tabella 1 riportata di seguito vengono riassunte le caratteristiche tipiche della programmazione *Client-Side* e *Server-Side*.

Java viene spesso descritto come un linguaggio per programmazione client side, con riferimento al suo bytecode "sicuro" e scaricabile dai browsers. Vediamo ora come Java rappresenti anche una buona soluzione (sotto opportune condizioni) per l'altra faccia della medaglia, la programmazione server side.

La Web Programming è nata con la server side programming. Quelli che in origine erano statici documenti HTML, sono divenuti, grazie al server side programming, pagine dinamiche costruite on the fly per processare moduli e visualizzare i risultati di interrogazioni di database. Il primo linguaggio usato per rendere interattive le pagine Web è stato il Perl, sfruttando la specifica nota come Common Gateway Interface ( vedi paragrafo successivo).

Client-Side	Server-Side
Le pagine appaiono dinamiche direttamente nel browser.	Le pagine sono generate dinamicamente dal server, ma una volta inviate al browser sono statiche.
L'elaborazione del programma richiede risorse del calcolatore su cui il browser viene eseguito.	Il back-end è eseguito sul server e quindi non impegna risorse del calcolatore su cui il browser viene eseguito.
L'architettura complessiva del sistema browser-server rimane invariata	L'architettura complessiva del sistema browser-server deve essere modificata, per introdurre il back-end.
Il browser deve essere abilitato per eseguire il programma associato alla pagina.	Qualsiasi browser può essere utilizzato.
Il programma associato alla pagina deve essere trasferito dal server al browser, impegnando la linea di comunicazione	Il back-end rimane sul server e solo i risultati dell'elaborazione vengono trasferiti lungo la linea.
Soluzione non adatta per sistemi di consultazione di database che non possono essere trasferiti al browser per ragioni di riservatezza, o sicurezza, o dimensioni.	Soluzione adatta per back-end con funzione di sistemi di consultazione di database.

**Tabella 1:** Le caratteristiche della programmazione Client-Side e Server-Side.

La CGI è rimasta per anni il principale strumento per la server side web programming. Anche se oggi gli script CGI sono i più diffusi, sono da tempo comparse soluzioni tecnicamente più solide e pratiche per lo sviluppo di applicazioni Web di qualità professionale e respiro industriale. Citiamo ad esempio il *Server JavaScript* di Netscape, le *Active Server Pages* di Microsoft o l'*IntraBuilder* di Borland. Java deve gran parte del suo iniziale successo alle *applet*, che insieme ai plugin/ActiveX e al JavaScript/VBScript rappresentano la parte client side del web programming.

Tuttavia, sviluppare servizi accessibili via web significa nella maggior parte dei casi dover gestire aspetti di programmazione riguardanti tanto il lato server quanto il lato client. Facciamo un esempio: supponiamo di dover generare un grafico utilizzando dei dati contenuti in un database. Una buona soluzione è generare una pagina HTML contenente i parametri per una applet che visualizza il grafico. La parte server quindi interroga il database e costruisce una

pagina HTML contenente la descrizione del grafico, ma la visualizzazione vera e propria viene gestita dalla applet che gira sul client.

Data la necessità di dover considerare entrambe le facce della medaglia, è naturale che molti sviluppatori desiderino utilizzare un solo linguaggio per lo sviluppo di applicazioni sia lato client che lato server. Chiunque abbia tentato di fare server side programming in Java avrà scoperto che sebbene Java sia un ottimo linguaggio per molti usi, di per sé non si presta particolarmente allo sviluppo di applicazioni CGI, perché non fornisce nessuno supporto specifico.

#### 1.2.4 LA TECNOLOGIA COMMON GATEWAY INTERFACE (CGI)

Data la natura stessa del progetto e l'obiettivo di realizzare un servizio accessibile tramite un comune browser (Microsoft Internet Explorer, Netscape Navigator,...) la tecnica di interfacciamento client/server risultata vincente si basa sullo standard CGI.

CGI (Common Gateway Interface) è uno standard, un insieme di regole, per l'interfacciamento di applicazioni client/server.

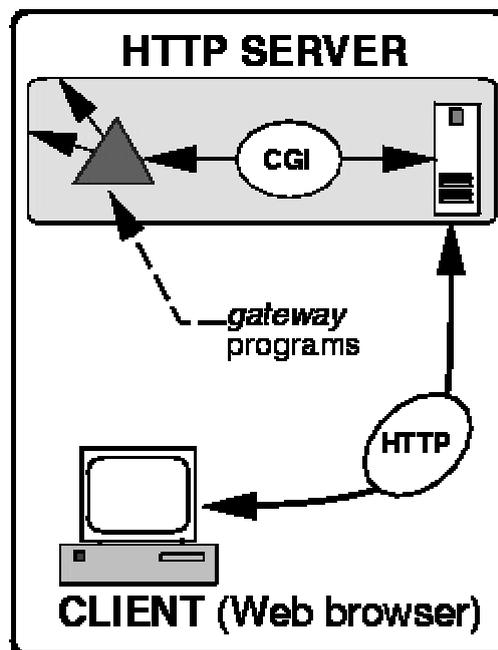


Fig. 1.4 : Common Gateway Interface.

Un semplice documento HTML, che il server restituisce quando sopraggiunge una chiamata dal browser client, è un oggetto statico. Un programma CGI, invece, viene eseguito sul server in risposta alle richieste ricevute, in modo tale da restituire un'informazione dinamica.

L'interfaccia CGI è stata definita allo scopo di standardizzare i metodi di decodifica dei dati in arrivo al server e di uniformare tutta la produzione software futura..

Ad un programma CGI, per assolvere alle funzioni per le quali è stato progettato, devono avere associati dei permessi tali che chiunque possa richiederne l'esecuzione sul sistema sul quale il Web server è installato. Questo, chiaramente, implica di dover ricorrere ad alcune precauzioni, in termini di configurazione del server e di implementazione del programma, al fine di garantire la sicurezza del sistema stesso.

I programmi CGI devono risiedere in una directory speciale, in modo che il server Web possa decidere di procedere nell'esecuzione del file eseguibile piuttosto che limitarsi a visualizzarne il contenuto sul navigatore (browser). Questa directory è, solitamente, sotto il diretto controllo del webmaster, che inibisce all'utente generico la creazione di programmi CGI in essa. In generale, la directory nella quale risiedono i programmi CGI viene denominata "/cgi-bin". Un altro modo con il quale il server identifica un file come eseguibile, invece che come immagine, testo o codice html, è l'estensione .cgi nel nome del file. Essa non è quindi usata, come potrebbe sembrare, per

denotare il tipo di linguaggio con il quale il codice sorgente del programma è stato scritto; un programma CGI può essere infatti scritto in qualsiasi linguaggio, che ne consenta l'esecuzione sul sistema.

La dinamica di funzionamento è la seguente:

- Il client richiede al server l'esecuzione di uno specifico programma CGI.
- Il server riceve i dati e li passa al programma CGI.
- Il software li elabora e li restituisce al browser sotto forma di documento HTML, testo, ecc.

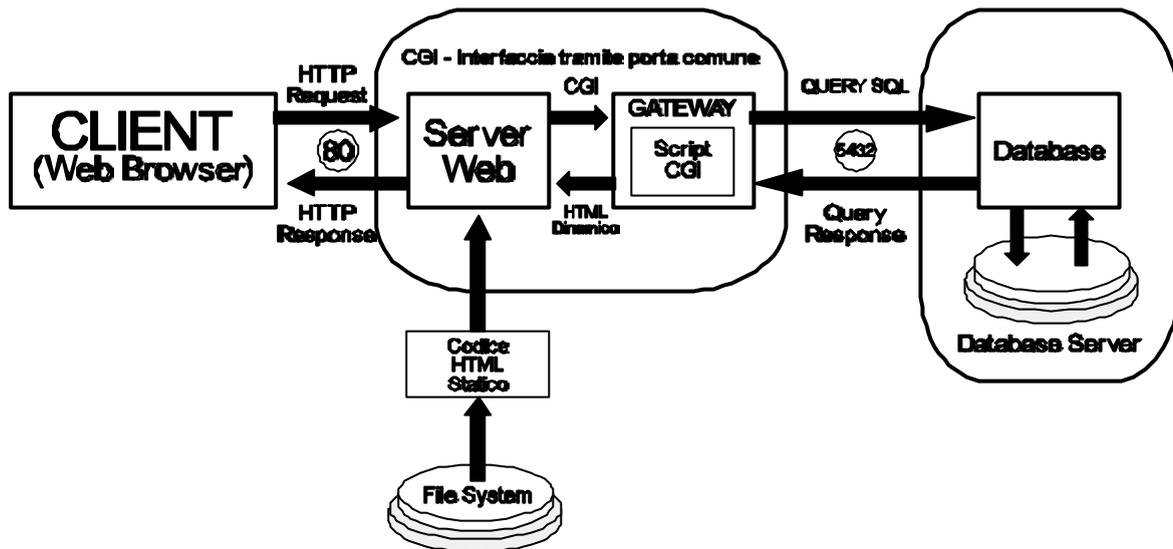


Fig. 1.5: Esempio di uso dell'Interfaccia Common Gateway per l'accesso a database attraverso la rete.

All'interno dei documenti HTML, sono presenti delle strutture attraverso le quali è possibile richiedere l'esecuzione di programmi CGI. Queste sono:

1. i moduli (form),
2. la struttura Isindex,
3. il link diretto al nome del programma CGI.

Per permettere che l'output del programma CGI sia dinamico è necessario consentire al client di specificare, unitamente all'invocazione del programma, dei parametri di ingresso. Ci sono due famiglie di parametri che il programma riceve e può utilizzare:

1. le variabili ambientali: parametri definiti dai sistemi client e server in uso,
2. i parametri specificati dall'utente, i cui valori vengono inseriti in strutture appositamente definite nell'ambito della pagina Web.

**Variabili Ambientali (Environment Variables):** Il server definisce delle speciali variabili: sono memorizzate nell'ambiente del sistema, e per questo risultano facilmente accessibili. Tali variabili vengono ricavate dalle informazioni inviate dal browser e rese disponibili al programma CGI in esecuzione sul server. Esse danno informazioni sul browser (tipo, files supportati, nome dell'host client, e altri), sul server (nome, versione, porta del socket, e altri) ed anche sullo stesso programma CGI (nome del programma e path). Le più importanti variabili d'ambiente CGI sono:

- SERVER\_SOFTWARE: il nome e la versione del server http che risponde alla richiesta cgi;
- SERVER\_NAME: il nome del server http che può essere un alias di DNS o un indirizzo IP numerico;
- GATEWAY\_INTERFACE: la versione delle specifiche CGI;
- SERVER\_PROTOCOL: il nome e la versione del protocollo di comunicazione HTTP;
- SERVER\_PORT: il numero di porta a cui è indirizzata la richiesta (porta 80, generalmente);
- REQUEST\_METHOD: il metodo usato per la richiesta (GET, POST, ed altri);
- SCRIPT\_NAME: il path completo del programma CGI;
- QUERY\_STRING: contiene i dati passati dalle richieste GET, ISINDEX, ISMAP;
- REMOTE\_HOST: il nome dell'host che invia la richiesta;
- REMOTE\_ADDR: l'indirizzo IP dell'host che invia la richiesta;
- AUTH\_TYPE: il metodo usato dal protocollo di autenticazione dell'utente. Questa variabile è settata solo se il server supporta tale protocollo e solo se lo script cgi è protetto;
- REMOTE\_USER: il nome dell'utente autenticato. Questa variabile è settata solo se il server supporta il protocollo di autenticazione e se lo script cgi è protetto;
- CONTENT\_TYPE: per le richieste POST e PUT questa variabile contiene il tipo MIME dei dati passati al cgi;
- CONTENT\_LENGTH: la lunghezza dei dati passati con il metodo POST;
- HTTP\_ACCEPT: i tipi MIME accettati dal client;
- HTTP\_USER\_AGENT: il nome e la versione del browser che ha inviato la richiesta.

**Parametri Utente:** Esistono due metodi che possono essere usati per passare dei parametri in ingresso al programma CGI:

1. il metodo GET,
2. il metodo POST.

Per determinare quale metodo viene usato, il programma CGI controlla la variabile ambientale REQUEST\_METHOD, che può essere impostata con valore pari a GET oppure a POST. Se è impostata a POST, allora la lunghezza dell'informazione trasmessa è contenuta nella variabile d'ambiente CONTENT\_LENGTH.

**Metodo GET** : I dati inseriti dall'utente nel modulo (form) , appositamente definito all'interno della pagina HTML che l'utente aveva precedentemente richiamato, vengono inviati dal browser come parte dell'URL richiesto; ovvero vengono "attaccati" (appended) all'URL originale (specificato nel modulo dall'attributo ACTION) e comunicati in questo modo al server.

Il server li memorizza nella variabile d'ambiente QUERY\_STRING, che verrà poi messa a disposizione del programma CGI. Quando, ad esempio, si seleziona "Submit" nella pagina HTML, il contenuto del form viene elaborato dal browser, e viene inviata una query che si presenta nella forma:

<http://nome.del.server/cgi-bin/programma?nome1=valore1&nome2=valore2&nome3=valore3...>

**Metodo POST:** Il contenuto del modulo viene codificato esattamente come per il metodo GET; quello che cambia è il modo in cui i dati vengono inviati al server. Anzichè aggiungere il contenuto del modulo all'URL, esso viene spedito in un blocco di dati e-mail: ovvero tutti i campi (e relativi valori presenti nel form) vengono riuniti ed elaborati per formare uno o più messaggi e-mail.

L'attributo ACTION, se specificato, conterrà l'URL al quale il blocco di dati verrà spedito.

**Esecuzione:** Ogni volta che un client richiede l'URL corrispondente ad un programma CGI, il server lo esegue in tempo reale. L'uscita del programma viene poi inviata, più o meno direttamente, al client. Esistono varie tecniche per eseguire programmi CGI sul server web. Le più note sono:

- **Chiamata Diretta.** Tramite una linea di comando virtuale (la riga nel browser dove si specifica l'URL da visitare) è possibile eseguire direttamente un programma CGI. Attraverso gli argomenti resi disponibili con la linea di comando le informazioni vengono decodificate dal server e passate al programma, che le può utilizzare, ad esempio, per la ricerca in una banca-dati.
- **Isindex.** Questo è il secondo modo per poter eseguire un programma CGI sul server Web. Esso non ha bisogno di essere inserito all'interno di un modulo, nè di specificare il nome o il valore. Permette, infatti, il passaggio di un solo parametro, senza nome, ma solo con il suo valore. La forma dell'URL che viene composto all'invio dei dati è: <http://nome.del.server/cgi-bin/programma?valore>. Con l'elemento ISINDEX, l'invio del comando di esecuzione del programma CGI avviene premendo il tasto invio della tastiera, dopo l'immissione del parametro nella finestra creata da ISINDEX.
- **Moduli HTML.** I moduli (form) rappresentano una delle possibili strutture presenti all'interno di un documento HTML. Sono dei raccoglitori nei quali vengono inserite istruzioni che definiscono dei campi per l'inserimento di dati. All'interno dei moduli possono essere inseriti campi di input, tasti e caselle di testo. L'invio del comando di esecuzione del programma CGI avviene "cliccando" il tasto "Submit", dopo aver immesso i vari parametri nelle diverse finestre visualizzate nel form stesso. Il tasto Submit è presente convenzionalmente alla fine del modulo, dentro la finestra del navigatore.

**Uscite Prodotte:** Il programma CGI trasmette i risultati al client (browser) visualizzandoli, secondo parametri convenzionali (http/html), sullo standard output (stdout) del programma stesso. Può anche creare delle intestazioni proprie o tipiche del server risparmiando la fatica a quest'ultimo.

I programmi CGI hanno, pertanto, un ruolo di Response. Possono restituire una grande varietà di tipi di documenti (immagini, documenti HTML, semplici files di testo, ...), oppure riferimenti ad altri documenti. Il client deve, in ogni caso, sapere che tipo di documento gli è stato restituito, per poterlo trattare opportunamente. Il programma CGI deve quindi comunicare al server quale tipo di documento sta restituendo: il protocollo CGI richiede, a questo scopo, la presenza di un'intestazione (header) nell'output.

### 1.2.5 LA TECNOLOGIA ASP (ACTIVE SERVER PAGES)

Superare i limiti dell'html per creare dei siti sempre più rispondenti alle esigenze dei visitatori è stata una delle mete a cui i programmatori di linguaggi di scripting hanno puntato nel corso della storia del web. Dalle prime pagine statiche, manifesto di un sito, si è progressivamente arrivati non solo all'esplosione del multimediale, ma, soprattutto, al diffondersi di pagine interattive, in grado non solo di affascinare, ma di fornire un utile strumento a chi le volesse usare.

Di linguaggi scripting ne esistono parecchi, alcuni più simili a veri e propri linguaggi di programmazione, altri più facili. Il PERL, ad esempio, è il tipico caso di linguaggio complesso, ma molto efficace, utilizzato per la creazione di CGI (Common Gateway Interface) a livello professionale e poco diffuso fra gli utenti medi, a causa della difficoltà di apprendimento e di gestione della sua sintassi; inoltre, essendo nato in ambiente Unix ha trovato non poche difficoltà ad affermarsi al di fuori di una pur vasta cerchia di provider e professionisti del settore che utilizzano questo sistema operativo. Un'altra difficoltà notevole per l'utente medio è da sempre

rappresentata dalla generale impossibilità di eseguire i programmatori che fruttano l'interfacciamento CGI al di fuori della directory cgi-bin del web server del quale ci si serve, directory il cui accesso è limitato al webmaster. Solo in rari casi l'amministratore di sistema consente l'esecuzione di script PERL (che hanno la possibilità di eseguire operazioni di lettura e scrittura su disco) da una qualsiasi directory. Se ciò ha una giustificazione a livello di sicurezza di un sito web, certo non ha invogliato gli utenti ad utilizzare questo linguaggio che richiede una "interazione" con provider e amministratori a volte troppo indaffarati per dare ascolto alle insolite richieste di un cliente.

Superare la staticità delle pagine web, mantenendo al contempo una semplicità di programmazione che consenta a tutti di intervenire senza prima dovere leggere voluminosi manuali è ora possibile grazie ai nuovi linguaggi di scripting ed alle nuove tecniche di interfacciamento. Una tecnologia recente è rappresentata dall'ASP (Active Server Pages) per la rapidità e flessibilità di utilizzo che la caratterizzano, che però sono controbilanciate da uno svantaggio non indifferente; l'utilizzo di questo linguaggio è confinato ai server Microsoft, come ad esempio a IIS (Internet Information Server), e non funziona quindi con tutti gli altri server che popolano il web. La sempre maggiore diffusione dei server Windows contribuisce però a rendere meno limitante questo ostacolo e, tutto sommato, non è difficile vedere diversi provider abbandonare il mondo Unix per le nuove possibilità offerte da Windows NT. Dato che il sito internet [www.evweb.it](http://www.evweb.it) è ospitato da un mantainer con server dotati di sistema operativo Windows, si è deciso di adottare questa tecnologia.

Grazie all'utilizzo delle pagine ASP, si possono creare dei documenti che possono fornire informazioni, rispondendo in modo diverso alle differenti richieste dei navigatori. Ma quali sono, in breve, i vantaggi nell'utilizzo di questo linguaggio di scripting?

- 1) Le pagine ASP sono completamente integrate con i file html.
- 2) Sono facili da creare e non necessitano di compilazione.
- 3) Sono orientate agli oggetti e usano componenti server ActiveX.

Visti i vantaggi, e viste anche le limitazioni cui abbiamo accennato in precedenza, riassumiamo le tecnologie coinvolte nello sviluppo e funzionamento delle Active Server Pages:

- 1) Windows NT
- 2) Protocollo TCP/IP
- 3) Un web server che supporti Active Server, come IIS
- 4) ODBC (Open DataBase Connectivity) e un server database.

Esaminando più da vicino l'"anatomia" di questo genere di pagine possiamo constatare che esse sono costituite da tre differenti parti:

- 1) Testo
- 2) Marcatori html
- 3) Comandi script

In un documento con estensione .asp è consentito utilizzare variabili, cicli, istruzioni di controllo, etc., grazie alla possibilità di richiamare la sintassi un linguaggio di scripting, come ad esempio il vbscript e il javascript, ma anche perl e rexx. La scelta del linguaggio dipende in primo luogo dalle necessità del programmatore e dal tipo di esecuzione che si vuole avere: se si vogliono eseguire gli script dal lato server è preferibile utilizzare il vbscript, mentre se ci si vuole affidare alla potenza degli "scripting engine" (motore che interpreta i comandi dei linguaggi di scripting e li esegue) dei singoli navigatori è sicuramente meglio utilizzare il javascript, semplice ed efficace.

Dato che per tutte le procedure di registrazione online di EVWeb.it Alternative Network si è utilizzata la tecnologia ASP scrivendo codice in JScript, anche per il progetto Gameisland si è adottato lo stesso linguaggio. Di seguito viene riportato il codice sorgente della pagine di registrazione REGISTERGI.ASP a cui il modulo di invio dati utente descritto nei paragrafi precedenti fa riferimento.

```
<%@ LANGUAGE = JScript %>
<%

// Ricevo dati dal FORM applicativo PC MMB GAMEISLAND
var nome = Request.QueryString("Nome");
var cognome = Request.QueryString("Cognome");
var indirizzo = Request.QueryString("Indirizzo");
var localita = Request.QueryString("Localita");
var cap = Request.QueryString("Cap");
var provincia = Request.QueryString("Provincia");
var telefono = Request.QueryString("Telefono");
var email = Request.QueryString("Email");
var codicefiscale = Request.QueryString("Codicefiscale");
var servizio = Request.QueryString("Servizio");
var d = new Date();
var g = "" + d.getDate() + "/" + (d.getMonth()+1) + "/" + d.getYear();

if (nome == "") Response.Redirect("errore.asp?error=1");
if (nome == " ") Response.Redirect("errore.asp?error=77");
.
.
.
if (nome == "@") Response.Redirect("errore.asp?error=77");

if (cognome == "") Response.Redirect("errore.asp?error=2");
if (cognome == " ") Response.Redirect("errore.asp?error=77");
.
.
.
if (cognome == "@") Response.Redirect("errore.asp?error=77");

if (email == "") Response.Redirect("errore.asp?error=8");
if (email == " ") Response.Redirect("errore.asp?error=77");
.
.
.
if (email == "@") Response.Redirect("errore.asp?error=77");

//Istanzio l'oggetto Connection
var ADOConn = new ActiveXObject("ADODB.Connection");

//Creo la stringa di connessione
var strConn = "";
strConn += "driver={Microsoft Access Driver (*.mdb)};";
strConn += "dbq=" + Server.MapPath("/mdb-database/gameisland.mdb");

//Apro la connessione al Database Gameisland
ADOConn.Open(strConn);

//Istanzio RecordSet
var RecSet = new ActiveXObject("ADODB.Recordset");
```

```
//Collego RecordSet alla connessione aperta
RecSet.Open("clienti",ADODConn,3,3);
```

```
//Formulo il nuovo record
RecSet.AddNew();
RecSet("cognome") = cognome;
RecSet("nome") = nome;
RecSet("indirizzo") = indirizzo;
RecSet("cap") = cap;
RecSet("localita") = localita;
RecSet("provincia") = provincia;
RecSet("telefono") = telefono;
RecSet("codicefiscale") = codicefiscale;
RecSet("email") = email;
RecSet("servizio") = servizio;
RecSet("data") = g;
```

```
// Aggiorno la Tabella
RecSet.Update();
```

```
// Chiudo RecordSet
RecSet.Close();
```

```
// Chiudo la connessione
ADODConn.Close();
```

```
%>
```

```
<html>
```

```
<head>
```

```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
```

```
<meta name="GENERATOR" content="Microsoft FrontPage 3.0">
```

```
<base target="principale">
```

```
<title>Download</title>
```

```
</head>
```

```
<body bgcolor="fixed" bgcolor="#FFFFFF">
```

```
<div align="left">
```

```
<table border="0" cellpadding="0" cellspacing="0" bordercolor="#000000"
width="530">
```

```
<tr>
```

```
<td></td>
```

```
</tr>
```

```
<tr>
```

```
<td><div align="center"><center><table border="2" cellpadding="0"
cellspacing="3"
```

```
bordercolor="#AABBCE" width="530">
```

```
<tr>
```

```
<td bgcolor="#FFFFFF"><div align="center"><center><table border="0"
cellpadding="3"
```

```
cellspacing="0">
```

```
<tr>
```

```
<td><a href="OrdinaCDROM.asp" target="_top"></a></td>
```

```
</tr>
```

```
</table>
```

```
</center></div></td>
```







```
if error1="3" then
  Response.Write("ATTENZIONE: Non hai inserito il CODICE FISCALE")
end if
if error1="5" then
  Response.Write("ATTENZIONE: Non hai inserito la LOCALITA")
end if
if error1="4" then
  Response.Write("ATTENZIONE: Non hai inserito il CAP")
end if
if error1="6" then
  Response.Write("ATTENZIONE: Non hai inserito la PROVINCIA")
end if
if error1="7" then
  Response.Write("ATTENZIONE: Non hai inserito il TELEFONO")
end if
if error1="8" then
  Response.Write("ATTENZIONE: Non hai inserito l indirizzo E-MAIL")
end if
if error1="9" then
  Response.Write("ATTENZIONE: Non hai inserito il TELEFONO di casa")
end if
if error1="10" then
  Response.Write("ATTENZIONE: Non hai inserito il NUMERO di iscrizione all
albo professionale")
end if
if error1="11" then
  Response.Write("ATTENZIONE: Non hai inserito la PARTITA IVA")
end if
if error1="12" then
  Response.Write("ATTENZIONE: Non hai inserito la tua LOGIN")
end if
if error1="13" then
  Response.Write("ATTENZIONE: Non hai inserito la PASSWORD")
end if
if error1="14" then
  Response.Write("ATTENZIONE: la PASSWORD e la sua CONFERMA devono essere
uguali")
end if
if error1="15" then
  Response.Write("ATTENZIONE: Non hai inserito in numero di Tessera
Sanitaria")
end if
if error1="16" then
  Response.Write("ATTENZIONE: Non hai inserito il nome della Societa'")
end if
if error1="17" then
  Response.Write("ATTENZIONE: Non hai inserito il numero di telefono
cellulare")
end if
if error1="18" then
  Response.Write("ATTENZIONE: UserID o Password errati")
  Response.Write("Se non e' registrato al servizio, deve prima effettuare
l'iscrizione!!")
end if
if error1="77" then
  Response.Write("ATTENZIONE: Non hai compilato i campi in modo corretto!")
  Response.Write(" ")
  Response.Write("In ottemperanza alla legge 675/96 (Privacy), i dati forniti
sono ")
end if
```



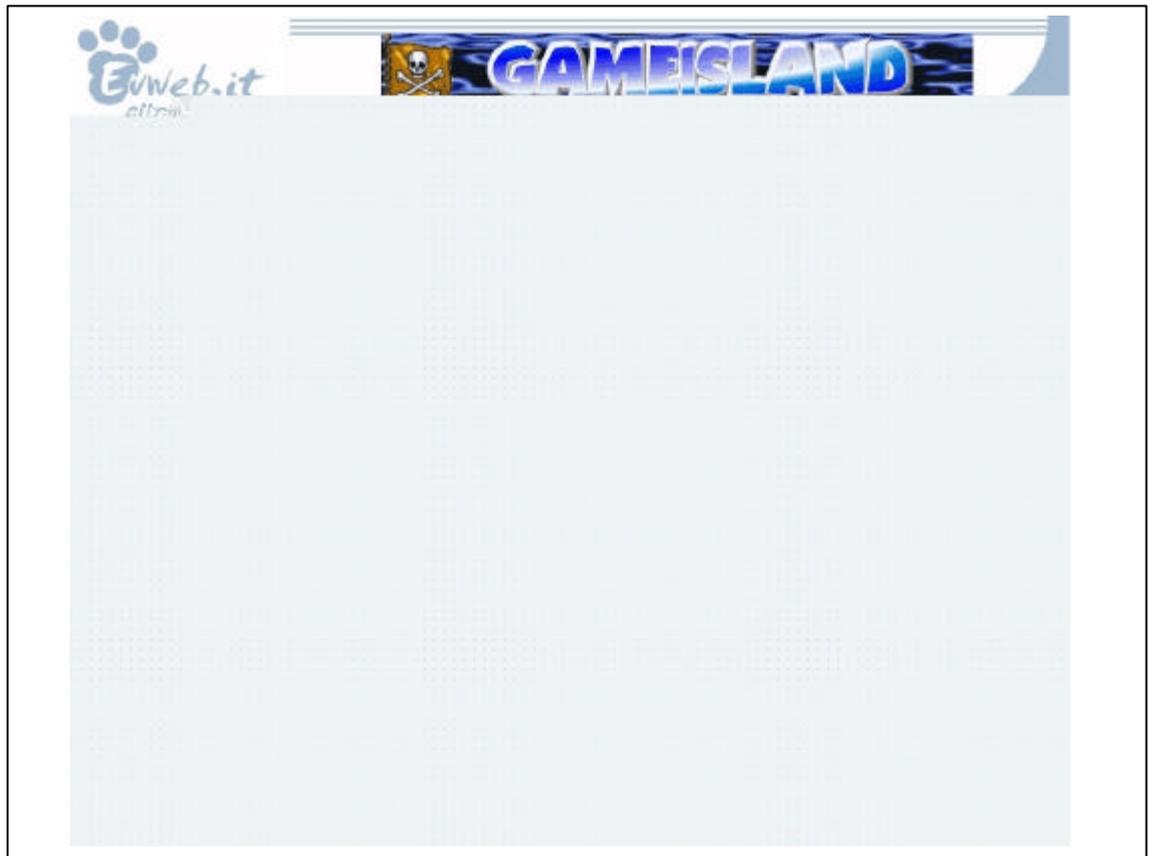


Fig. 1.6-a: Pagina web creata dallo script ASP di registrazione utente.

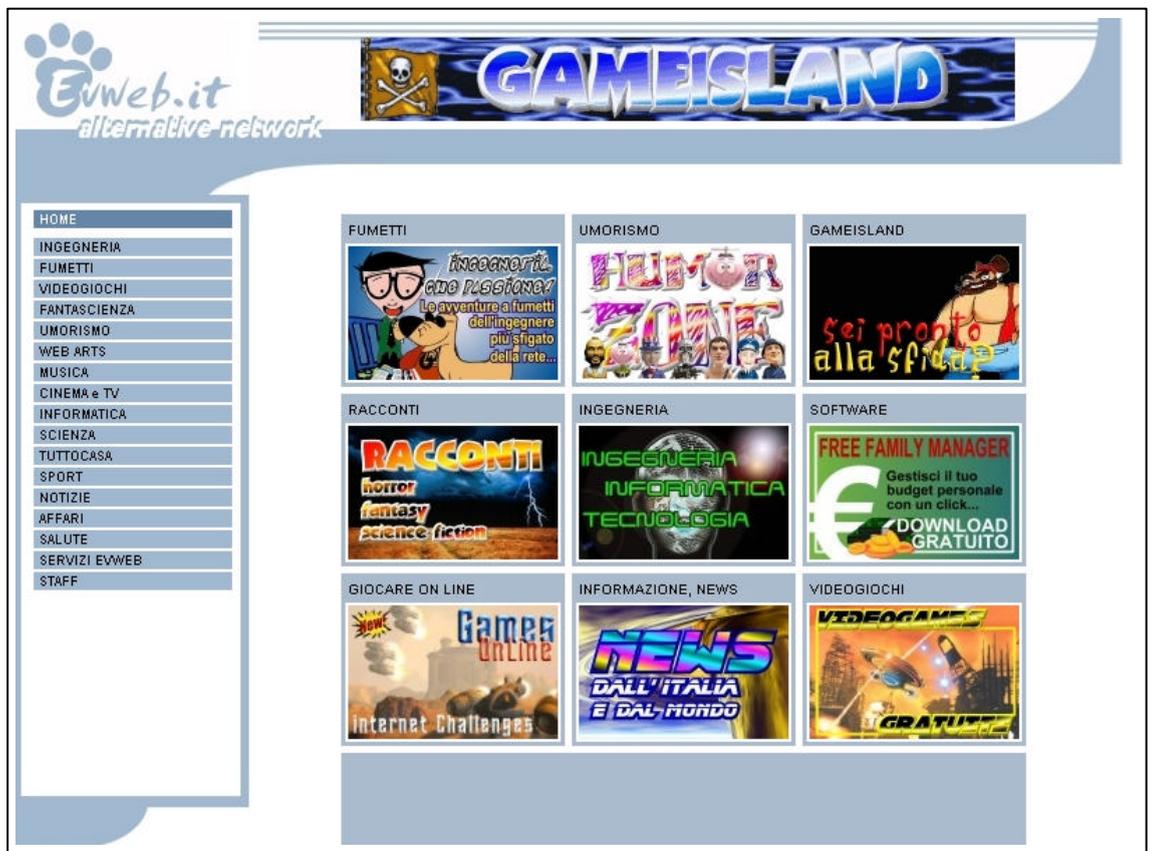


Fig. 1.6-b: Home Page del sito internet EVWeb.it Alternative Network.

### 1.3 IL PROGRAMMA DI INSTALLAZIONE

In fig. 1.7 un'immagine dell'applicativo "Setup Generator" della società Gentee, utilizzato per la creazione del software di installazione di Gameisland su Hard Disk. "Setup Generator" è freeware e può essere prelevato gratuitamente da un qualsiasi sito web di risorse per programmatori. E' uno dei pochi disponibili in lingua italiana. In fig. 1.8 come appare all'utente il programma di installazione guidata.

L'installazione non è indispensabile. L'utente può giocare direttamente dal CD-ROM (vedi fig. 1.9). Una caratteristica interessante è che la procedura di setup non prevede l'installazione di file nelle cartelle del sistema operativo, nessuna libreria dinamica DLL (quelle necessarie sono incorporate nei file dei giochi) e nessun file di registro. Questo al fine di concentrare i file necessari in una sola cartella, facilitando l'eventuale e futura procedura di disinstallazione.

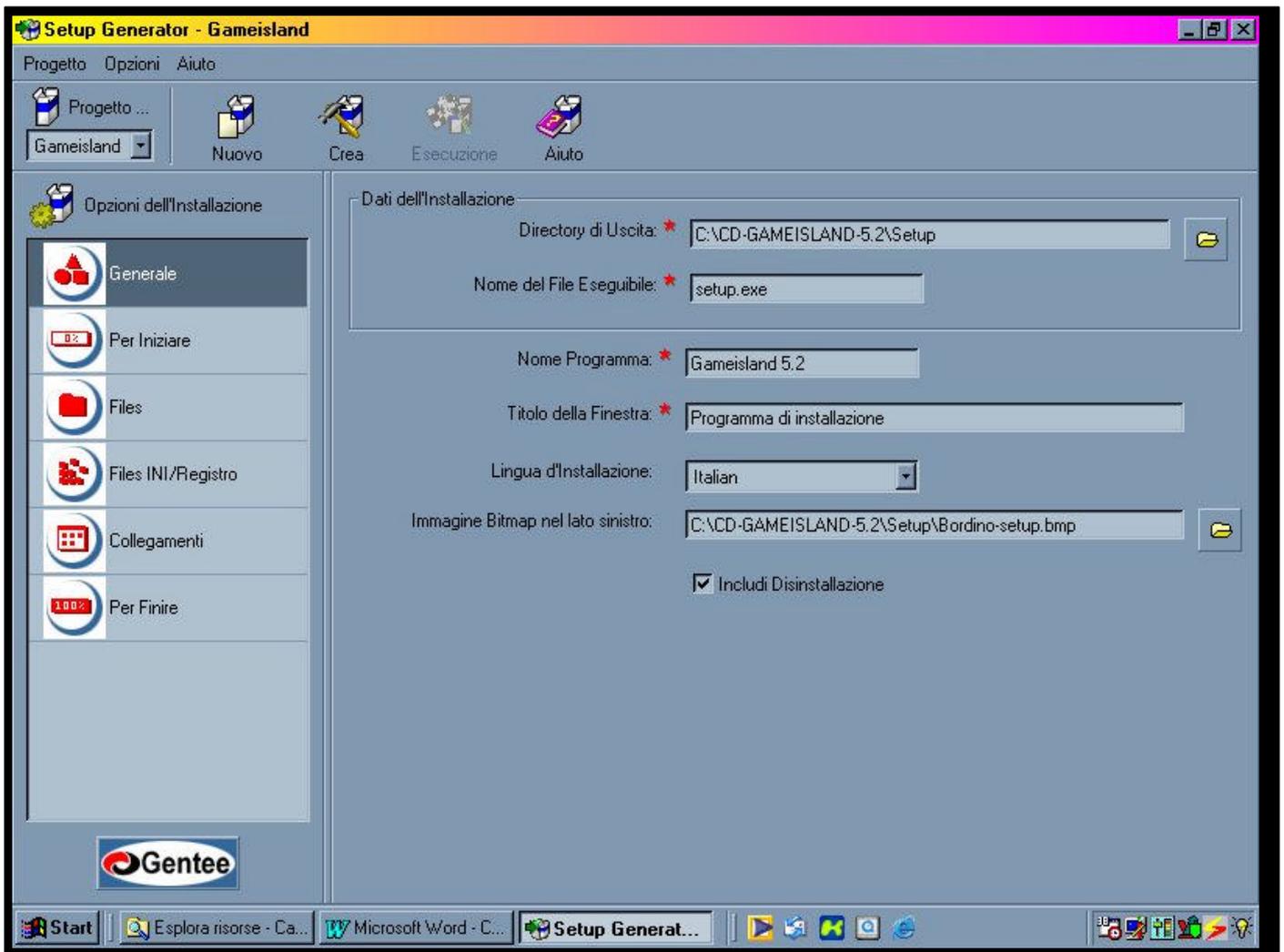


Fig. 1.7: Interfaccia principale di "Setup Generator".

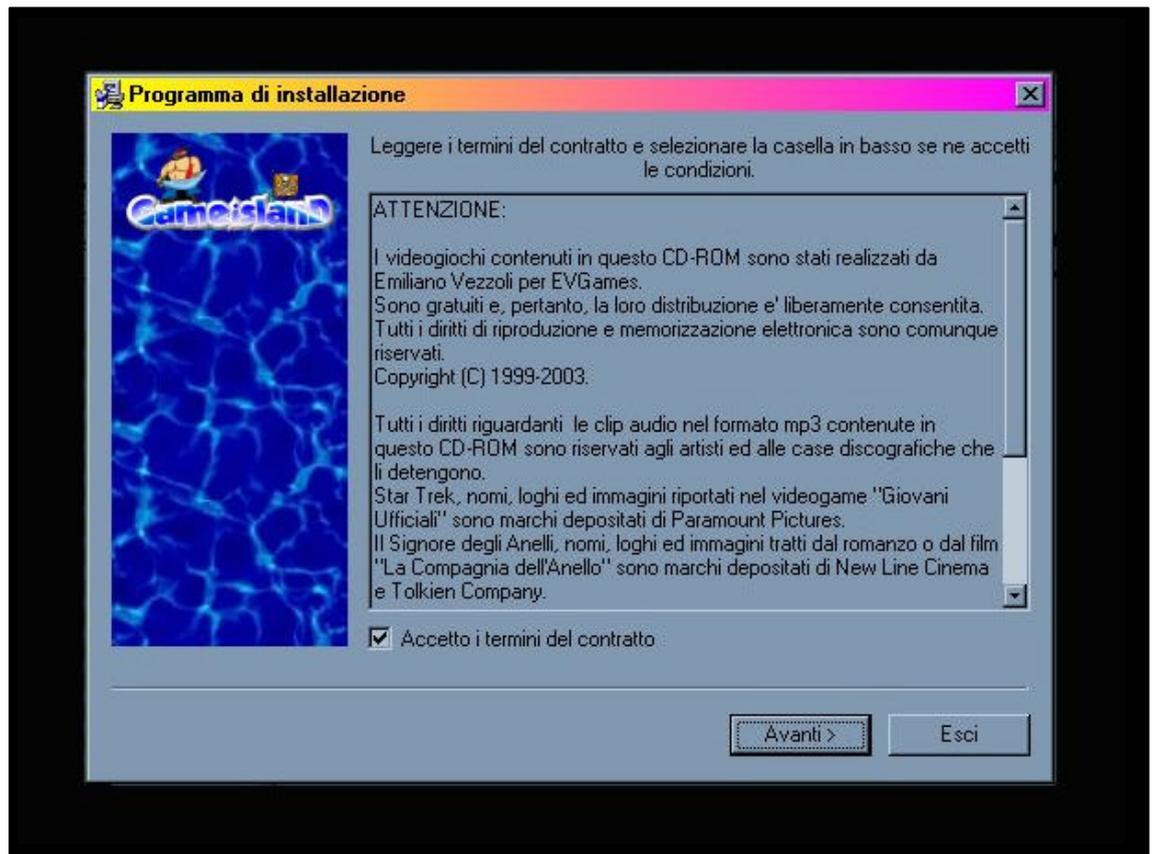


Fig. 1.8: Screenshot del programma di installazione di Gameisland.

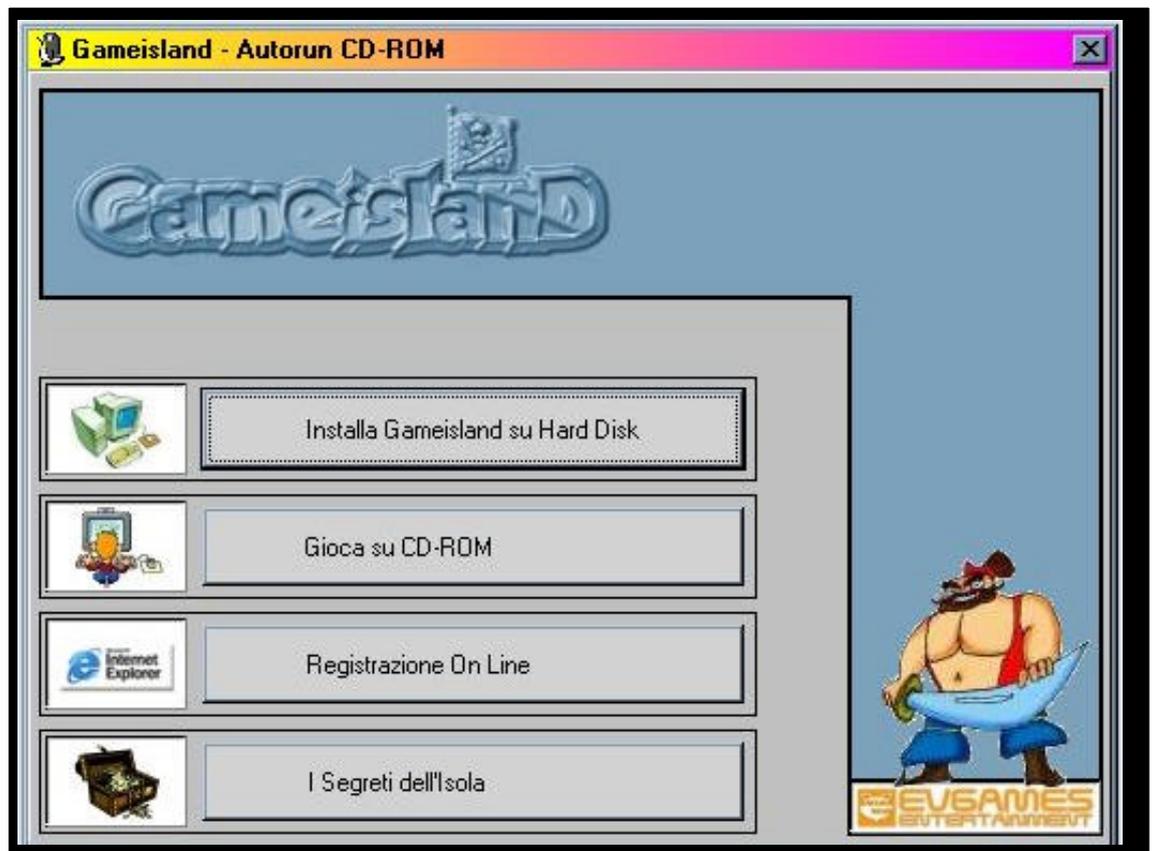


Fig. 1.9: Screenshot del programma di autorun del CD-ROM.

## 2. STAR TREK GIOVANI UFFICIALI

- ❑ Versione 2.0 Beta 3 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2001 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Star Trek, immagini, loghi, nomi, suoni sono marchi registrati di Paramount Pictures. Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 300Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 6,97 Mb

### 2.1 REGOLE DEL GIOCO

Star Trek Giovani Ufficiali è un'avventura grafica ambientata nel mondo dell'omonima e celebre saga televisiva. In particolare è dedicata alla serie " *The Next Generation*".

Il giocatore veste i panni di un giovane tenente della Flotta Stellare che deve sostenere l'esame per la promozione al grado di comandante. Tutto questo servendosi di un simulatore di missione che riproduce, fedelmente ed in perfetto stile trek (LCARS), la plancia di comando della nave spaziale Enterprise NCC-1701-D (vedi figura 2.1). Il giocatore ha a sua disposizione: visore principale, visore per comunicazioni interne con gli ufficiali superiori, display tattico ed ingegneria, velocità curvatura, distacco sezione a disco, teletrasporto, scudi deflettori, nonché un ricco database della Flotta, con la cartografia stellare, le navi, la strumentazione e le armi in dotazione all'equipaggio. Il giocatore può fare la sua mossa solo quando l'insegna " *comandante*" è accesa.

### 2.2 SUGGERIMENTI E TRUCCHI

Una delle difficoltà riscontrate maggiormente dai giocatori (soprattutto chi non conosce il mondo Trek) riguarda l'interazione con i personaggi del gioco, e cioè con gli ufficiali di plancia. Il pannello comandi alla sinistra del simulatore contiene tutti gli strumenti necessari al gioco. Per esempio: se un ufficiale rivela che qualcuno si trova in difficoltà in una certa zona dello spazio, è presumibile pensare che il comandante darà disposizioni per recarsi sul posto. In questo caso, dunque, il giocatore deve scegliere una delle velocità warp sulla tastierina numerica. In caso di emergenza, e cioè se il gioco risulta bloccato, premere il tasto di "reset manuale" posto in basso a destra sullo schermo.

### 2.3 LA REALIZZAZIONE (THE MAKING OF)

#### ❑ LA STORIA

La storia di questo gioco risulta molto complessa e sofferta. L'idea iniziale, risalente al 1999, era quella di realizzare un'avventura grafica che potesse competere con quelle più famose distribuite sul mercato. Progetto ambizioso, ma di difficile realizzazione. Non avendo a disposizione grafici, musicisti e programmatori per una "game engine" complessa, si è deciso di sfruttare le numerose librerie grafiche ed audio disponibili gratuitamente in rete. Dopo un

paio di mesi di lavoro una prima versione (la 1.0) denominata "Le Insegne del Comando" è stata resa disponibile al pubblico sul sito [www.evweb.it](http://www.evweb.it), con un discreto successo. Più di 1000 download in poche settimane.



Fig. 2.1: Interfaccia principale del simulatore.

Nel 2001, grazie al numero sempre maggiore di utenti che decidono di provare il videogioco, viene presa la decisione di migliorarlo e di effettuare un restyling grafico. Il risultato è la versione odierna, la 2.0 Beta 3. Ben lontano dall'essere priva di bachi, a causa dello storyboard particolarmente complesso, si rivela superiore alla versione precedente.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 2.10).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 2.2 un'immagine della cartografia stellare, con la mappa dei quadranti della nostra galassia. In fig. 2.3 c'è il menu di navigazione del database della Flotta Stallare; in fig. 2.4 i dati tecnici della nave, con animazioni e Over Screen; in fig. 2.5 fotografia dell'equipaggio della nave; in fig. 2.6 Help On Line e informazioni sulla release; nelle figg. 2.7 e 2.8 database delle navi stellari, con animazioni e fotografie. Infine, in fig. 2.9 database della strumentazione tecnica, medica e delle armi.

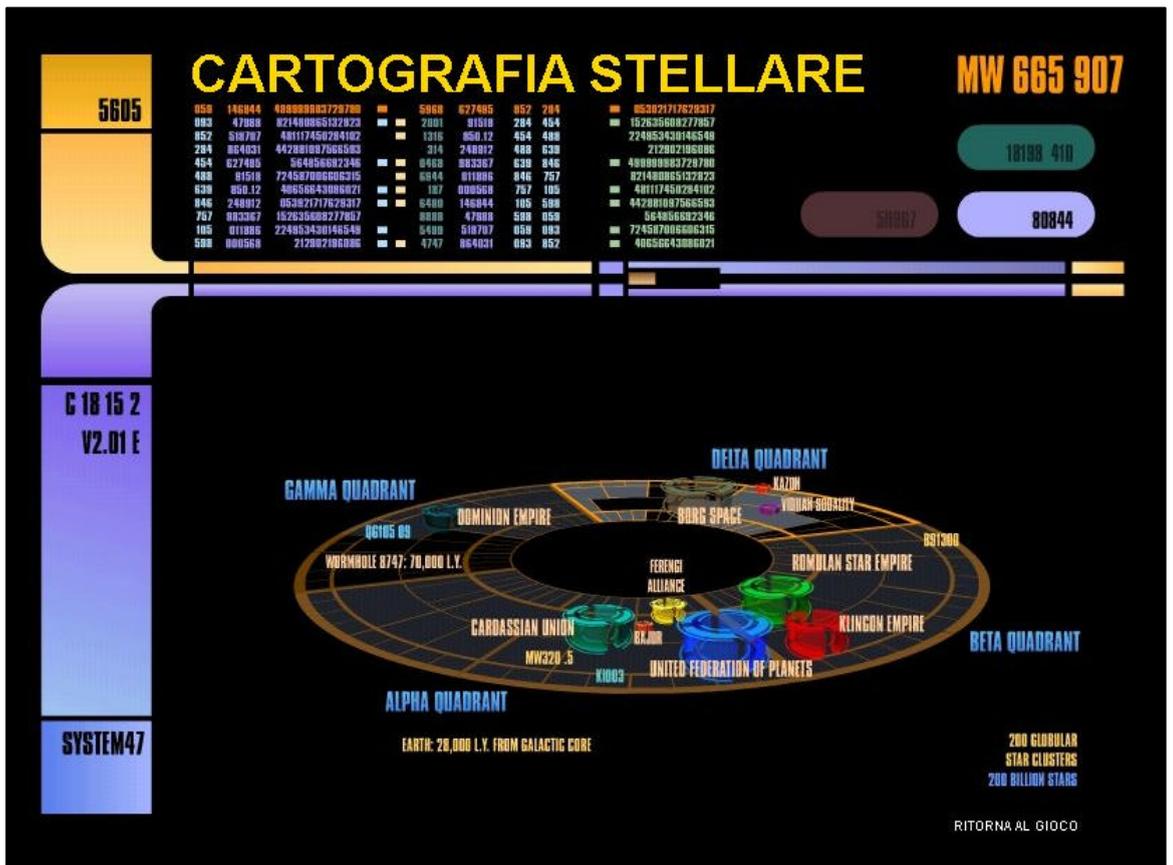


Fig. 2.2: Cartografia Stellare

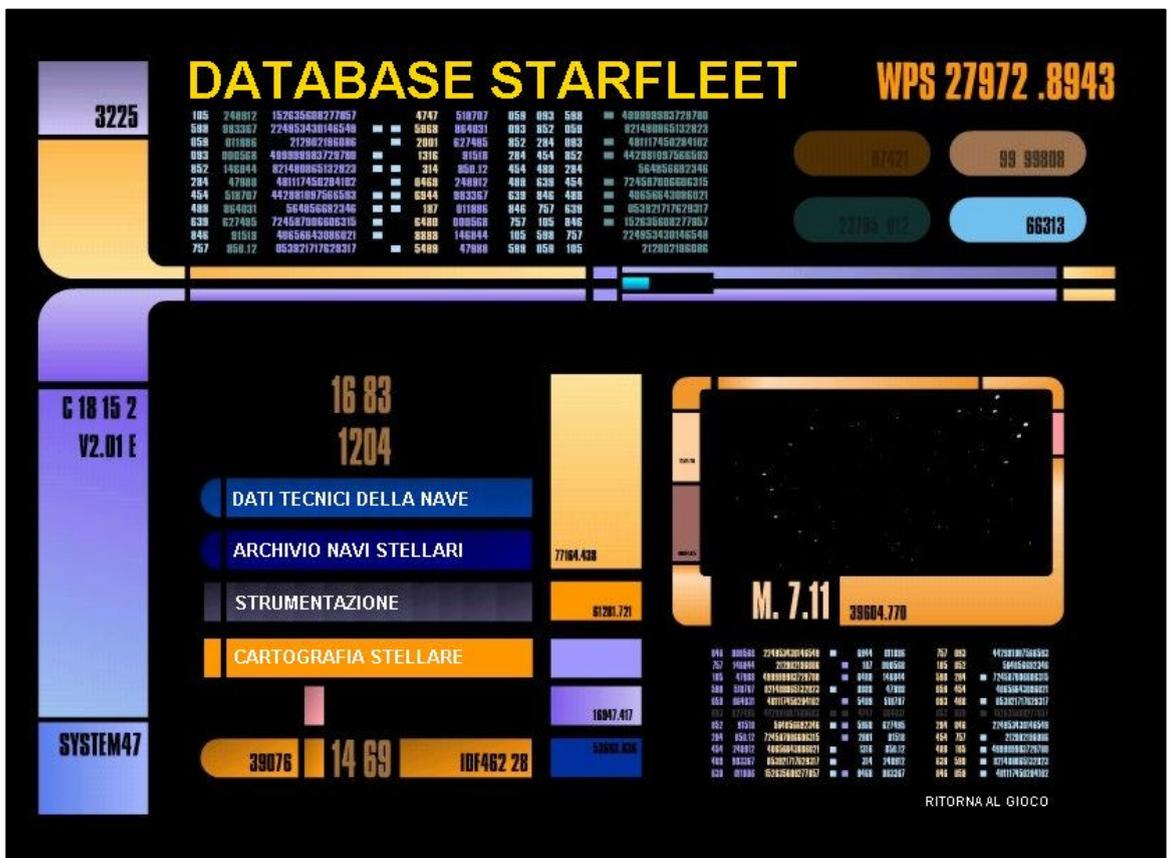


Fig. 2.3: Menu di navigazione Database della Flotta Stellare.

91131	4 79
11 60	GAMMA 521010
85622	4 22708 4
95775	4 30 DELTA 271018 2
62642	886 DELTA 28143 25
85494	1267 3.8784 1
	15 56
40106	
61285	1564
	4 45 82 971410 14

## DATA

OVER SCREEN



**E' la quinta nave a portare questo nome. E' un vascello di classe Galaxy, il piu' grande e potente della Federazione ( dopo l'ultima nata, la classe Sovereign, di cui fa parte l'Enterprise - E ). E' lunga circa 640 metri e pesa oltre cinque milioni di tonnellate. La struttura di base dello scafo e' formata da una lega di tritanio/duranio. L'Equipaggio consta di 1012 persone, inclusi i membri in servizio e le loro famiglie. La nave e' capace di operare**



58527.57	0558	EQUIPAGGIO	0058
28364.01			47007
		RITORNA AL GIOCO	77008

Fig. 2.4: Dati tecnici della nave, con animazioni e Over Screen.

91131	4 79
11 60	GAMMA 521010
85622	4 22708 4
95775	4 30 DELTA 271018 2
62642	886 DELTA 28143 25
85494	1267 3.8784 1
	15 56
40106	
61285	1564
	4 45 82 971410 14

## TEAM



58527.57	0558	0058	
28364.01		47007	
		RITORNA AL GIOCO	77008

Fig. 2.5: Fotografia dell'equipaggio della nave.

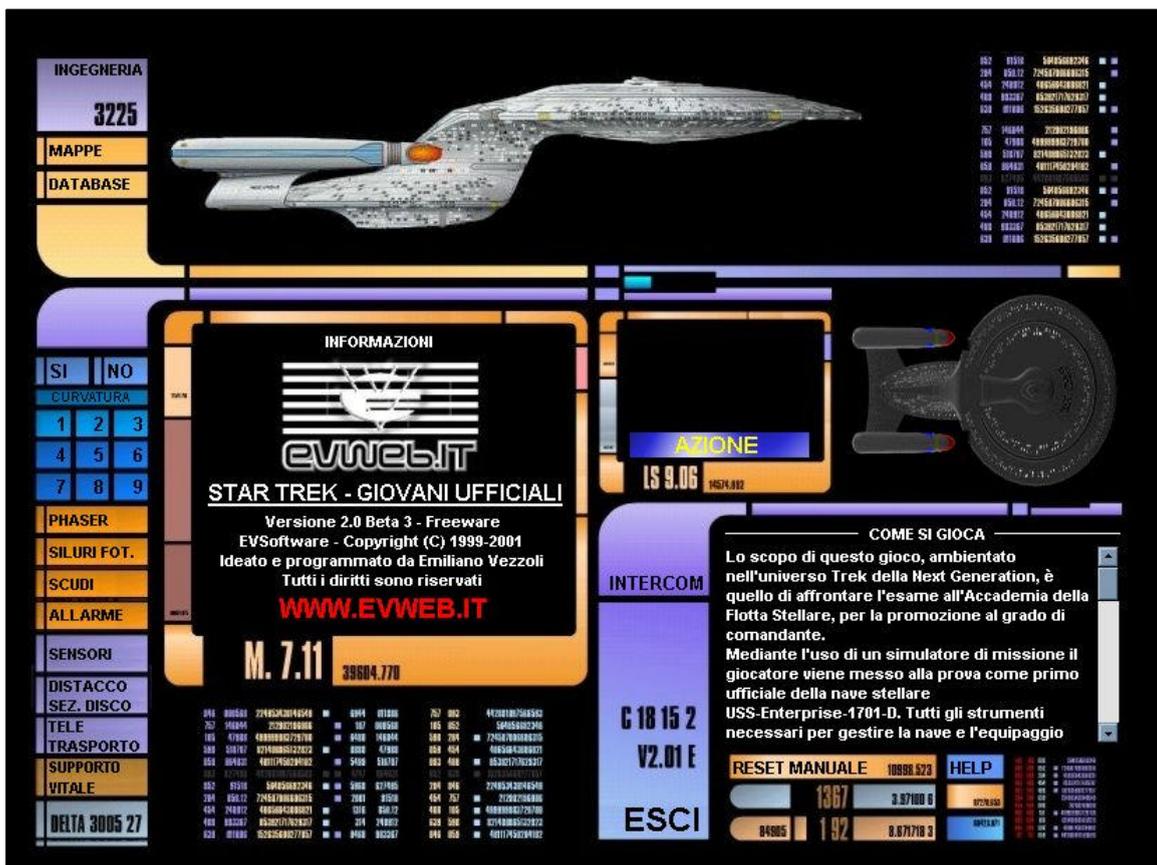


Fig. 2.6: Help On Line e informazioni sulla release.

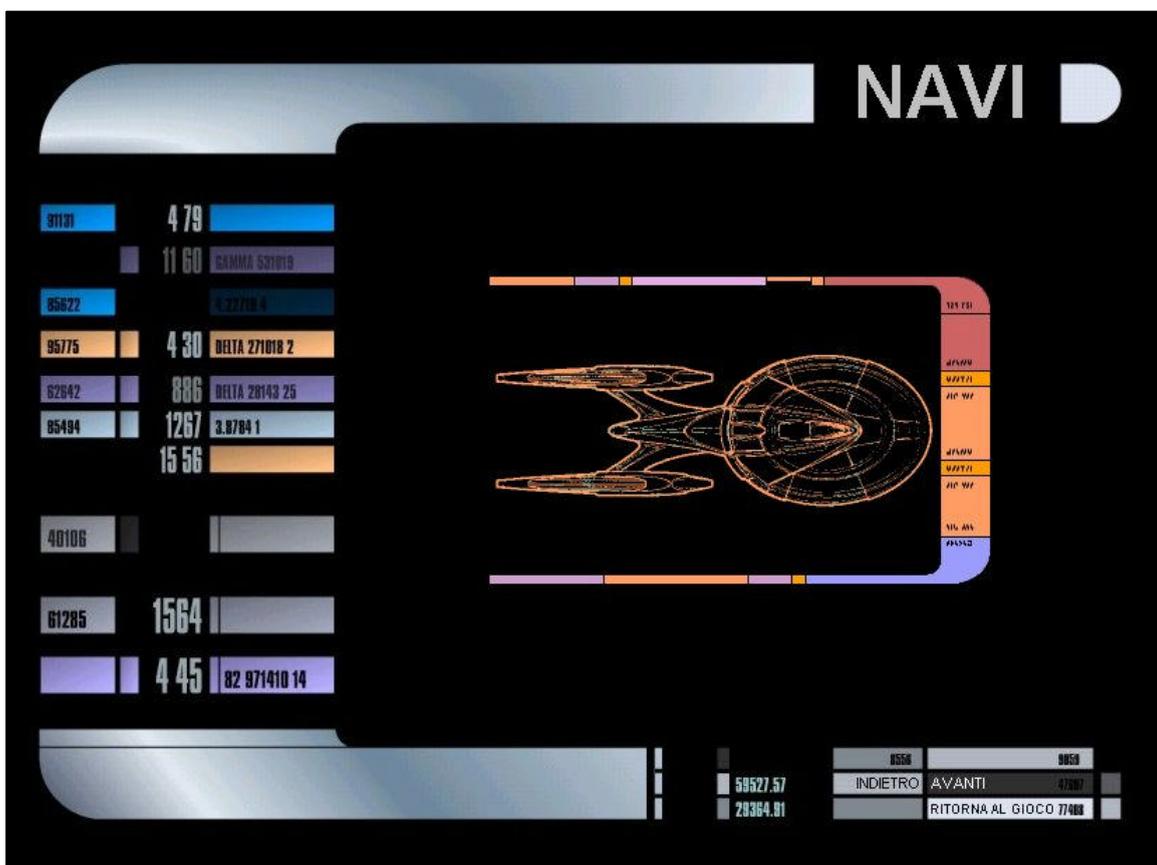


Fig. 2.7: Database delle navi stellari, con animazioni e fotografie.

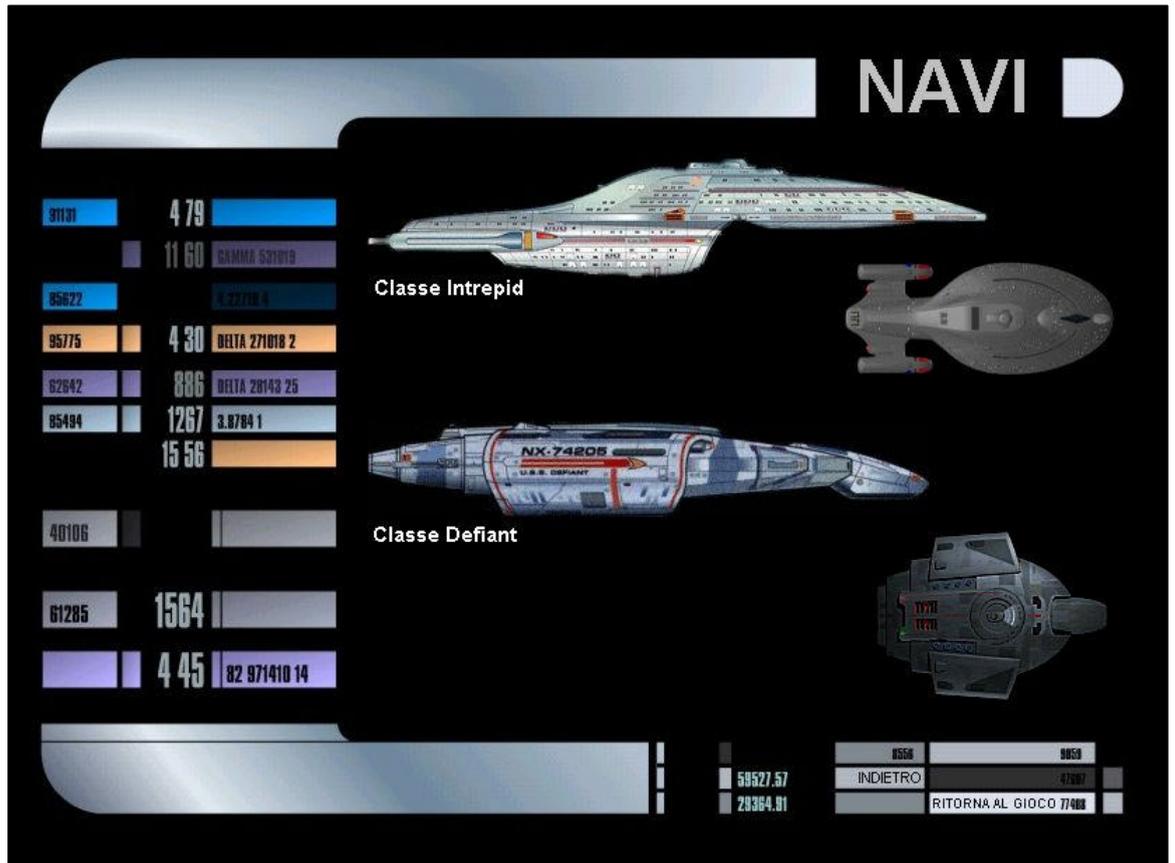


Fig. 2.8: Database delle navi stellari: la Voyager e la Defiant.

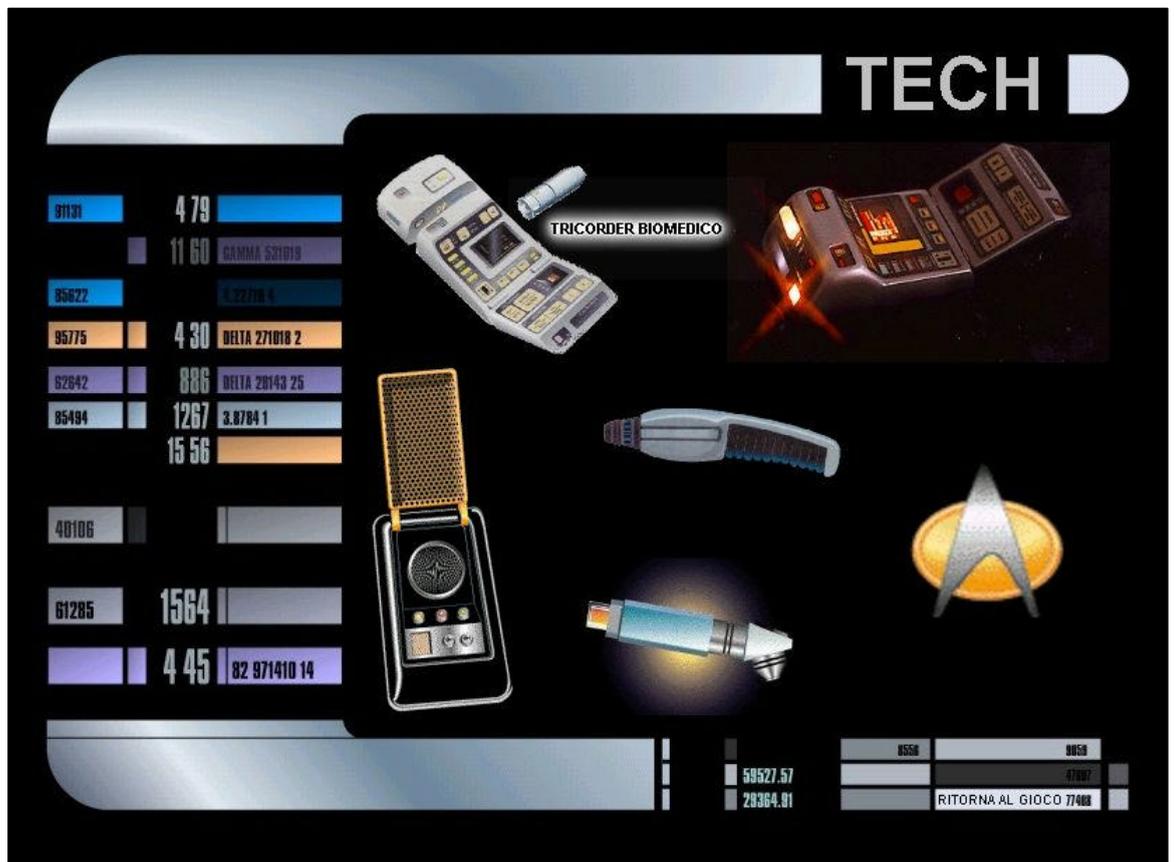


Fig. 2.9: Database della strumentazione tecnica, medica e delle armi

## 2.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

MP3Open("<SrcDir>\Suoni\Tng.mp3")
InstallFont("<Embedded>\TT1240M_.ttf")
InstallFont("<Embedded>\Arial.ttf")
Hide("Dialoghi")
Hide("Schermo")
Hide("Text58")
Hide("Text59")
Hide("Text60")
if (scudi = 1) then
  Show("Scudi1")
end
if (scudi = 2) then
  Show("Scudi2")
end
if (scudi = 3) then
  Show("Scudi3")
end
if (scudim = 100) then
  Show("Scudi100")
end
if (scudim = 50) then
  Show("Scudi50")
end
if (scudim = 20) then
  Show("Scudi20")
end
if (danno = 1) then
  Show("Danno")
end
if (dannoscafo = 1) then
  Show("DannoScafo")
end
if (dannodeflettore = 1) then
  Show("DannoDeflettore")
end
if (dannoprora = 1) then
  Show("DannoProra")
end
if (dannoimpulso = 1) then
  Show("DannoImpulso")
end
if (dannoscafo2 = 1) then
  Show("DannoScafo2")
end
if (data = 1) then
  Show("Com")
end
if (im = 1) then
  Show("Bit_Omicron")
end
if (im = 2) then
  Show("Bit_CuboBattaglia")
end
if (im = 3) then
  Show("Bit_Klingon")
end
if (im = 4) then

```

```

    Show("Bit_Esplosione")
end
if (im = 5) then
    Show("Bit_Phaser")
end
if (im = 6) then
    Show("Bit_CuboNave")
end
if (im = 7) then
    Show("Bit_CuboBorg")
end
if (im = 8) then
    Show("Bit_Warp")
end
if (im = 9) then
    Show("Bit_UPF")
end
if (im = 11) then
    Show("Bit_BaseStellare")
end
if (im = 12) then
    Show("Bit_Disturbo")
end
control5 = 1
if (inizio = 0) then
    AGifStop("AniGif")
    Hide("Dialoghi")
    Hide("Schermo")
    Show("HotSpot3")
    Show("Inizia")
    inizio = 1
end
AGifStop("AniGif")

```

#### □ Script 18

```

Hide("Dialoghi")
Show("Text48")
Hide("U_Cast")
Show("U_Worf")
Show("DannoScafo")
Hide("Scudi1")
Hide("Scudi100")
Show("Scudi2")
Show("Scudi50")
scudi=2
dannoscafo=1
scudim=50
danno2=1
PlaySound("<SrcDir>\Suoni\borg2.wav")
Show("Com")
control5=1

```

#### □ Script 20

```

Hide("Dialoghi")
Show("Text49")
Hide("U_Cast")
Show("U_Worf")
Hide("Scudi2")
Hide("Scudi50")
Show("Scudi3")
Show("Scudi10")
scudi=3
scudim=10
siluril=1

```

```
PlaySound("<SrcDir>\Suoni\Emergenza.wav")
Show("Com")
control5=1
```

□ **Script 19**

```
Hide("Dialoghi")
Hide("Schermo")
Show("Text51")
Hide("U_Cast")
Show("U_Worf")
Show("Bit_Klingon")
im=3
ScriptTimer("Script20", "12000")
PlaySound("<SrcDir>\Suoni\Siluri.wav")
```

□ **Phaser**

```
AGifStop("Anigif9")
Show("Com")
control5=1
```

□ **Warp 9**

```
Hide("RedAlert")
if (vittorial=0 & siluril=0 & scappol=0 & danno2=0 & guail=0 & borg1=0 & ds9=0) then
  Hide("Dialoghi")
  Show("Text22")
  Hide("U_Cast")
  Show("U_Paris")
  Show("Com")
  control5=1
  PlaySound("<SrcDir>\Suoni\Loudbell.wav")
end
if (vittorial=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Bit_Warp")
  Show("Text18_9")
  Hide("U_Cast")
  Show("U_Paris")
  im=8
  vittorial=0
  ScriptTimer("ScriptFinale", "8000")
  PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (danno2=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text19_9")
  Hide("U_Cast")
  Show("U_Paris")
  Show("Bit_Warp")
  danno2=0
  propostal=1
  ScriptTimer("Script5", "10000")
  PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (guail=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text19_9")
  Hide("U_Cast")
  Show("U_Paris")
  Show("Bit_Warp")
  guail=0
  propostal=1
```

```

    ScriptTimer("Script5", "10000")
    PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (danno3=1) then
    Hide("Dialoghi")
    Hide("Schermo")
    Show("Text19_9")
    Hide("U_Cast")
    Show("U_Paris")
    Show("Bit_Warp")
    danno2=0
    propostal=1
    ScriptTimer("Script5", "10000")
    PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (borg1=1) then
    Hide("Dialoghi")
    Hide("Schermo")
    Show("Text20_9")
    Hide("U_Cast")
    Show("U_Paris")
    Show("Bit_Warp")
    borg1=0
    reil=1
    rei2=1
    siluril=0
    ScriptTimer("Script3a", "10000")
    PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (scappol=1) then
ncl=r1=dannol=phaser1=ar1_1=scudil_1=dannol_1=scudil=no=nr1=ar1=klington1_1=klington1_
2=klington1_3=0
    Hide("Dialoghi")
    Hide("Schermo")
    Show("Text21_9")
    Hide("U_Cast")
    Show("U_Paris")
    Show("Bit_Warp")
    scappol=0
    borg1=1
    ScriptTimer("Script9", "10000")
    PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (ds9=1) then
    Hide("Dialoghi")
    Hide("Schermo")
    Show("Text18_9")
    Hide("U_Cast")
    Show("U_Paris")
    Show("Bit_Warp")
    ds9=0
    tunnell=1
    ScriptTimer("Script1", "10000")
    PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (siluril=1) then
    Hide("Dialoghi")
    Hide("Schermo")
    Show("Text18_9")
    Hide("U_Cast")
    Show("U_Paris")
    Show("Bit_Warp")
    siluril=0
    borg1=1
    ScriptTimer("Script9", "10000")

```

```

    PlaySound( "<SrcDir>\Suoni\Turbo.wav" )
end

```

#### □ TastoPhaser

```

if (scudi3=0 & ar3=0 & guail=0 & danno2=0 & scappol=0 & klingon1_3=0 & t2=0 &
siluril=0 & phaser1=0 & danno1=0 & scudil_1=0 & arl_1=0) then
    PlaySound( "<SrcDir>\Suoni\Comunicazioni3.wav" )
    control5=1
    Show( "Com" )
end
if (scudi3=1) then
    Hide( "Dialoghi" )
    Hide( "Schermo" )
    Show( "Text47" )
    Hide( "U_Cast" )
    Show( "U_Worf" )
    Show( "Bit_Phaser" )
    ScriptTimer( "Script18", "6000" )
    im=5
    scudi3=0
    danno3=1
    PlaySound( "<SrcDir>\Suoni\torpedo.wav" )
end
if (ar3=1) then
    Hide( "Dialoghi" )
    Hide( "Schermo" )
    Show( "Text47" )
    Hide( "U_Cast" )
    Show( "U_Worf" )
    Show( "Bit_CuboBattaglia" )
    im=2
    ScriptTimer( "Script18", "6000" )
    ar3=0
    danno3=1
    PlaySound( "<SrcDir>\Suoni\torpedo.wav" )
end
if (guail=1) then
    Hide( "Dialoghi" )
    Show( "Text44" )
    Hide( "U_Cast" )
    Show( "U_Worf" )
    guail=0
    Show( "Com" )
    control5=1
    PlaySound( "<SrcDir>\Suoni\Comunicazioni5.wav" )
end
if (danno2=1) then
    Hide( "Schermo" )
    Hide( "Dialoghi" )
    Show( "Text44" )
    Hide( "U_Cast" )
    Show( "U_Worf" )
    Hide( "Scudi1" )
    Hide( "Scudi2" )
    Show( "Scudi3" )
    Show( "Scudi10" )
    Hide( "Scudi50" )
    Hide( "Scudi100" )
    Hide( "NaveMoto" )
    im=5
    scudi=3
    scudim=10
    danno2=0
    guail=1

```

```

Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Comunicazioni4.wav")
end
if (scappol=1) then
Hide("Dialoghi")
scappol=0
PageTimer("10000", "Page 19")
PlaySound("<SrcDir>\Suoni\Comunicazioni5.wav")
end
if (klingon1_3=1) then
Hide("Schermo")
Show("Bit_Esplosione")
Hide("Dialoghi")
Show("Text42")
Hide("U_Cast")
Show("U_Worf")
im=4
klingon1_3=0
ScriptTimer("ScriptNON", "5000")
PlaySound("<SrcDir>\Suoni\Esplosione1.wav")
end
if (phaser1=1) then
Hide("Schermo")
Hide("Dialoghi")
Show("Bit_Esplosione")
im=4
Show("Text42")
Hide("U_Cast")
Show("U_Worf")
phaser1=0
scappol=1
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Esplosione1.wav")
end
if (t2=1) then
Hide("Schermo")
Hide("Dialoghi")
Show("Bit_Phaser")
im=5
Show("Text1")
Hide("U_Cast")
Show("U_Worf")
phaser1=1
t2=0
ScriptTimer("Phaser", "8000")
PlaySound("<SrcDir>\Suoni\Phaser.wav")
end
if (dannol=1) then
Hide("Schermo")
Hide("Dialoghi")
Show("Bit_Phaser")
im=5
Show("Text1")
Hide("U_Cast")
Show("U_Worf")
dannol=0
phaser1=1
ScriptTimer("Phaser", "10000")
PlaySound("<SrcDir>\Suoni\Phaser.wav")
end
if (ar1_1=1) then
Hide("Schermo")
Hide("Dialoghi")

```

```

Show("Bit_Phaser")
im=5
Show("Text1")
Hide("U_Cast")
Show("U_Worf")
arl_1=0
phaser1=1
ScriptTimer("Phaser", "10000")
PlaySound("<SrcDir>\Suoni\Phaser.wav")
end
if (scudil_1=1) then
Hide("Schermo")
Hide("Dialoghi")
Show("Bit_Phaser")
Show("Text1")
Hide("U_Cast")
Show("U_Worf")
im=5
scudil_1=0
phaser1=1
ScriptTimer("Phaser", "10000")
PlaySound("<SrcDir>\Suoni\Phaser.wav")
end
if (siluril=1) then
Hide("Schermo")
Hide("Dialoghi")
Hide("Scudil")
Hide("Scudi2")
Hide("Scudi3")
Hide("Scudi10")
Hide("Scudi50")
Hide("Scudi100")
Hide("NaveMoto")
Show("DannoScafo")
Show("DannoImpulso")
Show("DannoScafo2")
Show("DannoProra")
Show("DannoDeflettore")
Show("Danno")
Show("Bit_Disturbo")
scudi=0
scudim=0
dannoscafo=0
dannoimpulso=0
dannoscafo2=0
dannoprora=0
dannodeflettore=0
danno=0
im=12
Show("Text43")
Hide("U_Cast")
Show("U_Worf")
PageTimer("10000", "Page 19")
siluril=0
PlaySound("<SrcDir>\Suoni\Comunicazioni3.wav")
end

```

#### □ TastoSiluri

```

if (scudi2=0 & ar2=0 & guail=0 & danno3=0 & phaser1=0 & scappol=0) then
PlaySound("<SrcDir>\Suoni\Comunicazioni3.wav")
control5=1
Show("Com")
end
if (scudi2=1) then

```

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text47")
Hide("U_Cast")
Show("U_Worf")
Show("Bit_CuboBattaglia")
im=2
ScriptTimer("Script18", "6000")
scudi2=0
danno2=1
PlaySound("<SrcDir>\Suoni\torpedo.wav")
end
if (guail=1) then
Hide("Dialoghi")
Show("Text46")
Hide("U_Cast")
Show("U_Worf")
guail=0
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Ingegneria.wav")
end
if (danno3=1) then
Hide("Dialoghi")
Show("Text44")
Hide("U_Cast")
Show("U_Worf")
guail=1
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Ingegneria.wav")
end
if (ar2=1) then
Hide("Dialoghi")
Hide("Schermo")
Show("Text47")
Hide("U_Cast")
Show("U_Worf")
Show("Bit_CuboBattaglia")
im=2
ScriptTimer("Script18", "6000")
ar2=0
danno2=1
PlaySound("<SrcDir>\Suoni\torpedo.wav")
end
if (phaser1=1) then
Hide("Dialoghi")
Show("Text50")
Hide("U_Cast")
Show("U_Worf")
AGifStop("AniGif9")
phaser1=0
siluril=1
ScriptTimer("Script19", "6000")
PlaySound("<SrcDir>\Suoni\Siluro.wav")
end
if (scappol=1) then
Hide("Dialoghi")
PageTimer("6000", "page 19")
scappol=0
PlaySound("<SrcDir>\Suoni\Comunicazioni6.wav")
end
end

```

## □ TastoScudi

```

klingon1_3=0
if (reil=0 & rei2=0 & tunnel2=0 & klingon1_2 & klingon1_1=0) then
  Hide("Dialoghi")
  Show("Text6b")
  Hide("U_Cast")
  Show("U_Worf")
  ScriptTimer("pippo", "5000")
  PlaySound("<SrcDir>\Suoni\Comunicazioni5.wav")
end
if (klingon1_2=1) then
  Hide("Dialoghi")
  Show("Scudi1")
  Show("Scudi100")
  Show("Text3")
  Hide("U_Cast")
  Show("U_Worf")
  scudi=1
  scudim=100
  Klingon1_2=0
  scudil=1
  scudil_1=1
  PlaySound("<SrcDir>\Suoni\Scudi.wav")
end
if (rei2=1) then
  Hide("Dialoghi")
  Show("Scudi1")
  Show("Scudi100")
  Show("Text4")
  Hide("U_Cast")
  Show("U_Worf")
  scudi=1
  scudim=100
  rei2=0
  scudi2=1
  scudi3=1
  PlaySound("<SrcDir>\Suoni\Allarme.wav")
end
Show("Com")
control5=1

```

## □ TastoAllarme

```

AGifStop("AniGif1")
AGifStop("NaveMoto")
klingon1_3=0
if (rei2=0 & reil=0 & tunnel2=0 & klingon1_1=0 & klingon1_2=0) then
  Hide("Dialoghi")
  Show("Text6")
  Hide("U_Cast")
  Show("U_Esaminatore")
  PageTimer("10000", "Page 19")
  PlaySound("<SrcDir>\Suoni\Comunicazioni5.wav")
end
if (klingon1_1=1) then
  Hide("Dialoghi")
  Show("RedAlert")
  AGifStop("AniGif")
  AGifStop("AniGif1")
  Show("Scudi1")
  Show("Scudi100")
  Show("Text3")
  Hide("U_Cast")
  Show("U_Worf")
  scudi=1

```

```

scudim=100
klignonl_1=0
ar1=1
ar1_1=1
PlaySound("<SrcDir>\Suoni\Allarme.wav")
end
if (reil=1) then
Hide("Dialoghi")
Show("RedAlert")
Show("Scudi1")
Show("Scudi100")
AGifStop("AniGif")
AGifStop("AniGif1")
Show("Text4")
Hide("U_Cast")
Show("U_Worf")
scudi=1
scudim=100
reil=0
ar2=1
ar3=1
PlaySound("<SrcDir>\Suoni\Allarme.mp3")
end
ScriptTimer("ScriptAllarme", "6000")

```

□ **Script5a**

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text28")
Hide("Text59")
Hide("U_Cast")
Show("U_Worf")
Show("Bit_CuboBorg")
im=7
ScriptTimer("Script6", "8000")
PlaySound("<SrcDir>\Suoni\borg2.wav")

```

□ **Script2**

```

Hide("Dialoghi")
Show("Text27")
Hide("U_Cast")
Show("U_Data")
tunnell=1
PlaySound("<SrcDir>\Suoni\comunicazioni.wav")
Show("Com")
control5=1

```

□ **Script1**

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text26")
Hide("U_Cast")
Show("U_Paris")
Show("Bit_Ds9")
im=10
ScriptTimer("Script2", "10000")
PlaySound("<SrcDir>\Suoni\Comunicazioni6.wav")

```

□ **Script4**

```

Hide("Dialoghi")
Show("Text25")
Hide("U_Cast")
Show("U_Worf")
reil=1

```

```
rei2=1
PlaySound("<SrcDir>\Suoni\Comunicazioni5.wav")
Show("Com")
control5=1
```

□ **Script3**

```
Hide("Dialoghi")
Hide("Schermo")
Show("Text24")
Hide("U_Cast")
Show("U_Worf")
Show("Bit_CuboNave")
im=6
Hide("Text56")
ScriptTimer("Script4", "10000")
PlaySound("<SrcDir>\Suoni\Emergenza.wav")
```

□ **Script6**

```
Hide("Dialoghi")
Hide("Schermo")
Show("Text23")
Hide("U_Cast")
Show("U_Geordi")
Show("Bit_CuboBattaglia")
im=2
ScriptTimer("Script7", "10000")
PlaySound("<SrcDir>\Suoni\Loudbell.wav")
```

□ **Script7**

```
Hide("Dialoghi")
Show("Text21")
Hide("U_Cast")
Show("U_Paris")
ScriptTimer("Script8", "8000")
PlaySound("<SrcDir>\Suoni\Comunicazioni4.wav")
```

□ **Script8**

```
Hide("Dialoghi")
Show("Text20")
Hide("U_Cast")
Show("U_Beverly")
propostal=1
PlaySound("<SrcDir>\Suoni\Comunicazioni6.wav")
Show("Com")
control5=1
```

□ **Script9**

```
Hide("Dialoghi")
Hide("Schermo")
Hide("Scudi1")
Hide("Scudi2")
Hide("Scudi3")
Hide("Scudi10")
Hide("Scudi50")
Hide("Scudi100")
Hide("Danno")
Hide("DannoScafo")
Hide("DannoScafo2")
Hide("DannoDeflettore")
Hide("DannoProra")
Hide("DannoImpulso")
scudi=0
scudim=0
```

```

danno=0
dannoscafo=0
dannodeflettore=0
dannoprora=0
dannoimpulso=0
dannoscafo2=0
Show("Text19")
Hide("U_Cast")
Show("U_Geordi")
Show("Bit_BaseStellare")
im=11
ScriptTimer("Script10", "12000")
PlaySound("<SrcDir>\Suoni\Loudbell.wav")

```

#### □ Script10

```

Hide("Dialoghi")
Show("Text18")
Hide("U_Cast")
Show("U_Data")
borg1=1
PlaySound("<SrcDir>\Suoni\Emergenza.wav")
Show("Com")
control5=1

```

#### □ TastoSensori

```

if (tunnell1=0 & tunnel2=0) then
  Hide("Dialoghi")
  Show("Text10")
  Hide("U_Cast")
  Show("U_Data")
  Show("Com")
  control5=1
end
if (tunnel2=1) then
  Hide("Dialoghi")
  Show("Text9")
  Hide("U_Cast")
  Show("U_Data")
  ScriptTimer("Script13", "7000")
  tunnel2=0
  klingon1_1=1
  klingon1_2=1
  klingon1_3=1
  PlaySound("<SrcDir>\Suoni\Scienza2.wav")
end
if (tunnell1=1) then
  Hide("Dialoghi")
  Show("Text8")
  Hide("U_Cast")
  Show("U_Data")
  ScriptTimer("Script11", "10000")
  tunnell1=0
  tunnel2=1
  PlaySound("<SrcDir>\Suoni\Scienza2.wav")
end

```

#### □ TastoDisco

```

Hide("Dialoghi")
Show("Text17")
Hide("U_Cast")
Show("U_Geordi")
PlaySound("<SrcDir>\Suoni\Scienza1.mp3")

```

### □ TastoTrasporto

```
Hide("Dialoghi")
Show("Text15")
Hide("U_Cast")
Show("U_OBrian")
PlaySound("<SrcDir>\Suoni\Scienza2.wav")
```

### □ TastoVitale

```
Hide("Dialoghi")
Show("Text16")
Hide("U_Cast")
Show("U_OBrian")
PlaySound("<SrcDir>\Suoni\Loudbell.wav")
```

### □ Script11

```
Hide("Dialoghi")
Show("Text11")
Hide("U_Cast")
Show("U_Data")
tunnel2=1
ScriptTimer("Script12", "12000")
PlaySound("<SrcDir>\Suoni\Comunicazioni.wav")
```

### □ Script12

```
Hide("Dialoghi")
Show("Text14")
Hide("U_Cast")
Show("U_Data")
Hide("AniGif4")
PlaySound("<SrcDir>\Suoni\Comunicazioni3.wav")
Show("Com")
control5=1
```

### □ Script13

```
Hide("Dialoghi")
Hide("Schermo")
Show("Text12")
Hide("U_Cast")
Show("U_Data")
Show("Bit_Klingon")
im=3
klingon1_1=1
klingon1_2=1
klingon1_3=1
ScriptTimer("Script14", "12000")
PlaySound("<SrcDir>\Suoni\Siluri.wav")
```

### □ Script14

```
Hide("Dialoghi")
Show("Text13")
Hide("U_Cast")
Show("U_Geordi")
Show("Danno")
Show("DannoProra")
danno=1
dannoprora=1
PlaySound("<SrcDir>\Suoni\Emergenza.wav")
Show("Com")
control5=1
```

## □ TastoSi

```

if (vittorial=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text18_6")
  Hide("U_Cast")
  Show("U_Paris")
  Show("Bit_Warp")
  im=8
  vittorial=0
  ScriptTimer("ScriptFinale", "8000")
  PlaySound("<SrcDir>\Suoni\Turbo.wav")
end
if (curl=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text36")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Disturbo")
  Hide("NaveMoto")
  Hide("Scudi50")
  Hide("Scudi100")
  Hide("Scudi1")
  Hide("Scudi2")
  Show("Scudi3")
  Show("Scudi10")
  im=12
  scudi=3
  scudim=10
  curl=0
  PageTimer("7000", "Page 19")
  PlaySound("<SrcDir>\Suoni\Scudi.wav")
end
if (propostal=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text35")
  Hide("U_Cast")
  Show("U_Data")
  Show("Bit_Phaser")
  im=5
  propostal=0
  vittorial=1
  ScriptTimer("Script16", "8000")
  PlaySound("<SrcDir>\Suoni\Scudi.wav")
end
if (ncl=1) then
  Hide("Dialoghi")
  Show("Text33")
  Hide("U_Cast")
  Show("U_Worf")
  scappol=1
  ncl=0
  Show("Com")
  control5=1
  PlaySound("<SrcDir>\Suoni\Comunicazioni5.wav")
end
if (r1=1) then
  Hide("Dialoghi")
  Show("Text32")
  Hide("U_Cast")
  Show("U_Worf")
  ncl=1

```

```

r1=0
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Comunicazioni2.wav")
end
if (nr1=1) then
Hide("Dialoghi")
Show("Text31")
Hide("U_Cast")
Show("U_Worf")
no=1
r1=1
nr1=0
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Comunicazioni4.wav")
end
if (ar1=1) then
Hide("Dialoghi")
Show("Text30")
Hide("U_Cast")
Show("U_Data")
ar1=0
nr1=1
ScriptTimer("Script15", "10000")
PlaySound("<SrcDir>\Suoni\Comunicazioni3.wav")
end
if (scudil=1) then
Hide("Dialoghi")
Show("Text30")
Hide("U_Cast")
Show("U_Data")
scudil=0
nr1=1
ScriptTimer("Script15", "6000")
PlaySound("<SrcDir>\Suoni\Comunicazioni3.wav")
end

```

□ **Script16**

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text37")
Hide("U_Cast")
Show("U_Worf")
Show("Bit_Esplosione")
im=4
vittorial=1
PlaySound("<SrcDir>\Suoni\Esplosione2.wav")
Show("Com")
control5=1

```

□ **Script15**

```

Hide("Dialoghi")
Show("Text29")
Hide("U_Cast")
Show("U_Worf")
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Loudbell.wav")

```

## □ TastoNo

```

if (curl=0 & propostal=0 & nrl=0 & rl=0 & ar=0 & ncl=0) then
  Hide("Dialoghi")
  Show("Text41")
  Hide("U_Cast")
  Show("U_Data")
  Show("Com")
  control5=1
  PlaySound("<SrcDir>\Suoni\Comunicazioni6.wav")
end
if (curl=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Hide("Scudi1")
  Hide("Scudi2")
  Hide("Scudi100")
  Hide("Scudi50")
  Show("Scudi10")
  Show("Scudi3")
  Show("Text36")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Disturbo")
  scudi=3
  scudim=10
  im=12
  curl=0
  PageTimer("7000","Page 19")
  PlaySound("<SrcDir>\Suoni\Emergenza.wav")
end
if (propostal=1) then
  Hide("Dialoghi")
  Show("Text39")
  Hide("U_Cast")
  Show("U_Geordi")
  curl=1
  Show("Com")
  control5=1
  PlaySound("<SrcDir>\Suoni\Scienza2.wav")
end
if (nrl=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text38")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Klingon")
  im=3
  nrl=0
  ScriptTimer("Script17","12000")
  PlaySound("<SrcDir>\Suoni\Siluri.wav")
end
if (rl=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text38")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Klingon")
  im=3
  rl=0
  ScriptTimer("Script17","12000")
  PlaySound("<SrcDir>\Suoni\Siluri.wav")
end

```

```

if (ncl=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Show("Text38")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Klingon")
  im=3
  ncl=0
  ScriptTimer("Script17", "12000")
  PlaySound("<SrcDir>\Suoni\Siluri.wav")
end
if (arl=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Hide("Scudil")
  Hide("Scudi2")
  Hide("Scudi100")
  Hide("Scudi50")
  Show("Scudi10")
  Show("Scudi3")
  Show("Text36")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Disturbo")
  im=12
  scudi=3
  scudim=10
  arl=0
  ScriptTimer("ScriptNON", "5000")
  PlaySound("<SrcDir>\Suoni\Emergenza.wav")
end
if (scudil=1) then
  Hide("Dialoghi")
  Hide("Schermo")
  Hide("Scudil")
  Hide("Scudi2")
  Hide("Scudi100")
  Hide("Scudi50")
  Show("Scudi10")
  Show("Scudi3")
  Show("Text36")
  Hide("U_Cast")
  Show("U_Worf")
  Show("Bit_Disturbo")
  im=12
  scudi=3
  scudim=10
  scudil=0
  ScriptTimer("ScriptNON", "5000")
  PlaySound("<SrcDir>\Suoni\Emergenza.wav")
end

```

#### □ Script17

```

Hide("Dialoghi")
Hide("Scudil")
Hide("Scudi3")
Hide("Scudi10")
Hide("Scudi100")
Show("Scudi50")
Show("DannoDeflettore")
Show("Danno")
Show("Scudi2")
Show("Text40")

```

```

Hide("U_Cast")
Show("U_Geordi")
scudi=2
scudim=50
dannodeflettore=1
danno=1
dannol=1
PlaySound("<SrcDir>\Suoni\Scudi.wav")
Show("Com")
control5=1

```

#### □ Script5

```

Hide("Schermo")
Show("Bit_BaseStellare")
Hide("Dialoghi")
Show("Text59")
Hide("U_Cast")
Show("U_Paris")
im=1
ScriptTimer("Script5a", "8000")
PlaySound("<SrcDir>\Suoni\comunicazioni.wav")

```

#### □ Scrip3a

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text56")
Hide("U_Cast")
Show("U_Paris")
Show("Bit_Omicron")
im=1
ScriptTimer("Script3", "8000")
PlaySound("<SrcDir>\Suoni\Emergenza.wav")

```

#### □ Pippo

```

Hide("Text6b")
Hide("Text6c")
Show("Com")
control5=1

```

#### □ ScriptInizio

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text7")
Hide("U_Cast")
Show("U_Data")
Show("Com")
control5=1
PlaySound("<SrcDir>\Suoni\Comunicazioni4.wav")

```

#### □ ScriptFinale

```

Hide("Dialoghi")
Hide("Schermo")
Show("Text34")
Hide("U_Cast")
Show("U_Esaminatore")
Show("Bit_Ds9")
im=10
PageTimer("8000", "Page 20")
MP3Open("<SrcDir>\Suoni\Tng.mp3")

```

❑ **ScriptAllarme**

```
AGifReset ("RedAlert")
AGifStop ("RedAlert")
AGifPlay ("NaveMoto")
AGifPlay ("AniGif1")
Show ("Com")
control5=1
```

❑ **ScriptNON**

```
Hide ("Dialoghi")
Show ("Text60")
Hide ("U_Cast")
Show ("U_Esaminatore")
PageTimer ("9000", "Page 19")
```

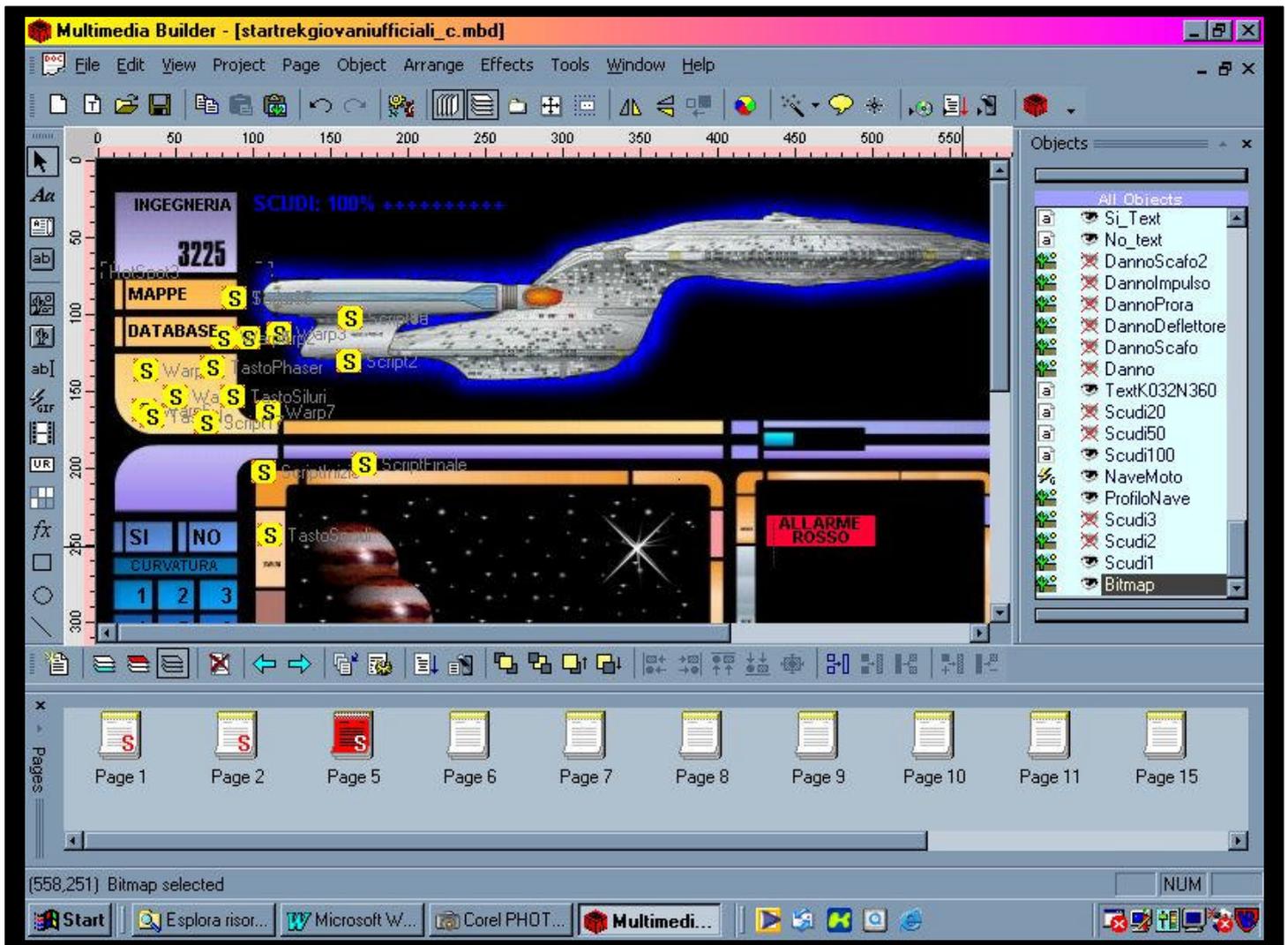


Fig. 2.10: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

### 3. ACQUATICA

- ❑ Versione 2.0 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2002 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



#### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,75 Mb

#### 3.1 REGOLE DEL GIOCO

"Acquatica 2" è un videogioco che abbina logica, strategia e fortuna. Studiato per un solo giocatore l'obiettivo da raggiungere consiste nell'accendere tutte le spie del tabellone. Ogni volta che si preme su una delle spie spente, si accendono o si spengono alcune di quelle adiacenti, secondo modalità e configurazioni proprie del livello che si sta giocando. Ad ogni spia accesa/spenta corrisponde una mossa. All'incremento del numero delle mosse corrisponde una diminuzione del livello energetico, terminato il quale il gioco finisce. Per accedere ai livelli successivi è necessario accendere tutte le spie.

#### 3.2 SUGGERIMENTI E TRUCCHI

Esiste un solo modo per riuscire a vincere a questo gioco: cominciare accendendo le spie poste agli spigoli del tabellone e procedendo regolarmente verso il centro. In tutti gli altri casi sperimentati, le strategie scelte non si sono dimostrate efficaci.

#### 3.3 LA REALIZZAZIONE (THE MAKING OF)

##### ❑ LA STORIA

La prima versione del videogame risale al 2000, come esercizio tecnico nella programmazione e gestione grafica di matrici ed effetti speciali vari (vedi figg. 3.7 e 3.8). Il nome, "Acquatica 2000", è legato al tema delle immagini, dei suoni e delle animazioni, che richiamano l'acqua e le onde del mare.

La difficoltà e, in un certo senso, la monotonia della trama non hanno aiutato il videogame a riscontrare il consenso del pubblico della rete al punto tale che, anche la seconda e, probabilmente, ultima release non ha ottenuto il successo sperato.



Fig. 3.2: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 3.9).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. Nelle figg. 3.2 e 3.3 ci sono degli screenshot dell'interfaccia principale del videogame. Nelle figg. 3.4 c'è la finestra con le informazioni su autore e release del gioco.

In fig. 3.5 c'è l'Help On Line con le istruzioni per giocare. In fig. 3.6: schermata di game over in caso di sconfitta. In fig. 1.8: schermata di game over in caso di sconfitta.

In fig. 3.7 e 3.8 si sono due immagini tratte dalla prima versione di Acquatica.

Infine, in fig. 3.9 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



Fig. 3.3: Interfaccia principale del videogioco.



Fig. 3.4: Informazioni su autore e release del gioco.



Fig. 3.5: Help On Line - Istruzioni per giocare.



Fig. 3.6: Schermata di game over.



Fig. 3.7: Uno screenshot tratto dalla versione 1.0 di Acquatica.

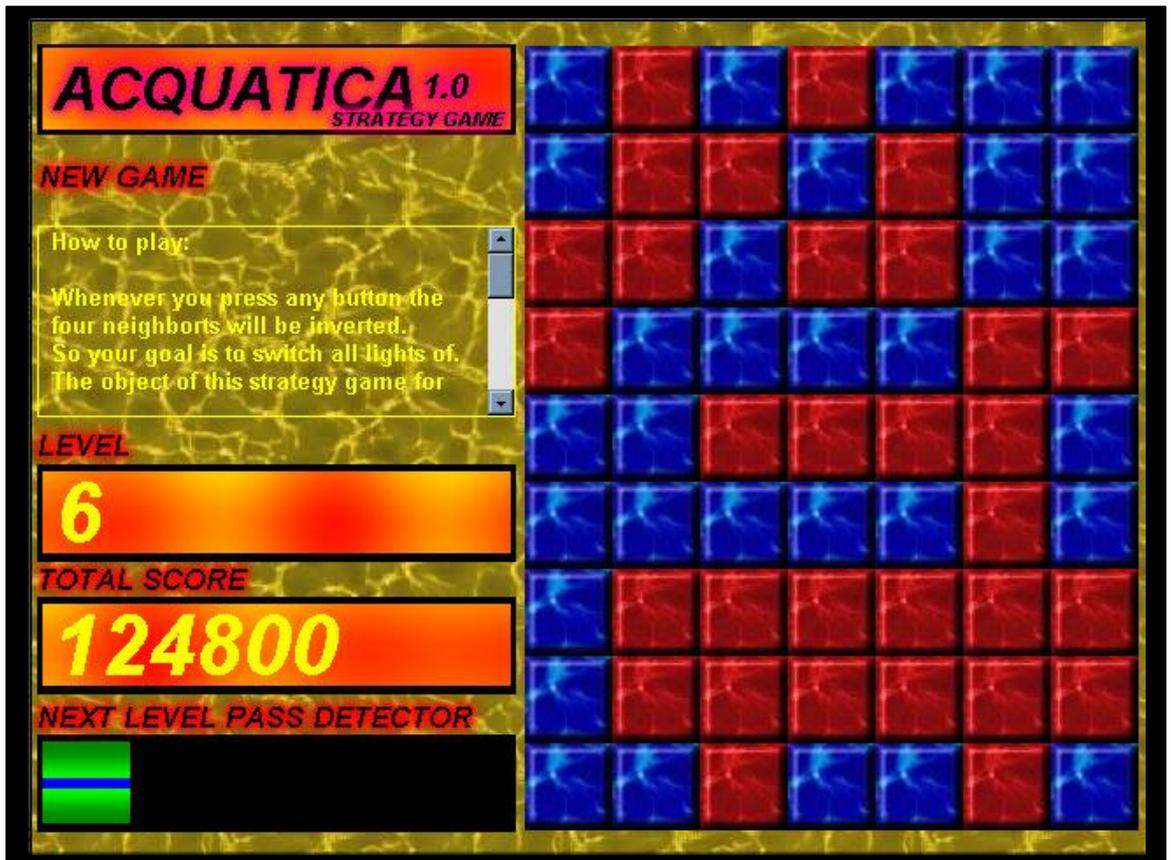


Fig. 3.8: Uno screenshot tratto dalla versione 1.0 di Acquatica.

### 3.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

#### □ Main Page

```
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\BlackWolf.ttf")
InstallFont("<Embedded>\Arial.ttf")
PageTimer("3000", "Page 2")
WavePlay("<SrcDir>\Acqua.wav", "[LOOP]")
punteggio=0
pp=0
WaveStop()
```

#### □ Tasto Nuovo

```
pp=0
punteggio=0
livello=1
LevelControl = 1
NumeroMosse = 0
Show("Matrix")
Hide("Group5")
DisplayValue("Text5", "punteggio")
DisplayValue("Text6", "livello")
DisplayValue("Mosse", "NumeroMosse")
MatrixSet("Matrix[0,0]", "1")
for a=0 to 10
  col = RND(9)+1
  row = RND(9)+1
  MatrixSet("Matrix[col,row]", "2")
next a
Hide("HotSpot4")
Hide("HaiPerso")
Hide("HaiVinto")
Hide("Help")
Hide("Info")
WaveStop()
Show("ImgFX_1")
```

#### □ Tasto Info

```
Hide("Help")
Hide("Perso")
Hide("Vinto")
```

#### □ Script help

```
Hide("Info")
Hide("Perso")
Hide("Vinto")
Show("Help")
```

#### □ Script Matrix – Mouse Down

```
MatrixGet("Matrix[MXCOL,MXROW]", "last")
MatrixSet("Matrix[MXCOL,MXROW]", "3")
lastX = MXCOL
lastY = MXROW
```

#### □ Script Matrix – Mouse Up

```
if (LevelControl = 1) then
  MatrixSet("Matrix[lastX,lastY]", "last")
  MatrixGet("Matrix[MXCOL,MXROW]", "idx")
```

```

MatrixSet ("Matrix[MXCOL,MXROW]", "3-idx")
if (MXCOL>1) then
    MatrixGet ("Matrix[MXCOL-1,MXROW]", "idx")
    MatrixSet ("Matrix[MXCOL-1,MXROW]", "3-idx")
end
MatrixGet ("Matrix[MXCOL+1,MXROW]", "idx")
MatrixSet ("Matrix[MXCOL+1,MXROW]", "3-idx")
if (MXROW>1) then
    MatrixGet ("Matrix[MXCOL,MXROW-1]", "idx")
    MatrixSet ("Matrix[MXCOL,MXROW-1]", "3-idx")
end
MatrixGet ("Matrix[MXCOL,MXROW+1]", "idx")
MatrixSet ("Matrix[MXCOL,MXROW+1]", "3-idx")
MatrixGet ("Matrix[MXCOL-1,MXROW-1]", "v1")
MatrixGet ("Matrix[MXCOL-1,MXROW]", "v2")
MatrixGet ("Matrix[MXCOL-1,MXROW+1]", "v3")
MatrixGet ("Matrix[MXCOL,MXROW-1]", "v4")
MatrixGet ("Matrix[MXCOL,MXROW]", "v5")
MatrixGet ("Matrix[MXCOL,MXROW+1]", "v6")
MatrixGet ("Matrix[MXCOL+1,MXROW-1]", "v7")
MatrixGet ("Matrix[MXCOL+1,MXROW]", "v8")
MatrixGet ("Matrix[MXCOL+1,MXROW+1]", "v9")
end
if (LevelControl = 2) then
    MatrixSet ("Matrix[lastX,lastY]", "last")
    MatrixGet ("Matrix[MXCOL,MXROW]", "idx")
    d=2-idx
    if (d=2) then
        MatrixSet ("Matrix[MXCOL,MXROW]", "d")
    end
    if (d=3) then
        MatrixSet ("Matrix[MXCOL,MXROW]", "d-2")
    end
    if (MXCOL>1) then
        MatrixGet ("Matrix[MXCOL-1,MXROW]", "idx")
        MatrixSet ("Matrix[MXCOL-1,MXROW]", "3-idx")
    end
    MatrixGet ("Matrix[MXCOL+1,MXROW]", "idx")
    MatrixSet ("Matrix[MXCOL+1,MXROW]", "3-idx")
    if (MXROW>1) then
        MatrixGet ("Matrix[MXCOL,MXROW-1]", "idx")
        MatrixSet ("Matrix[MXCOL,MXROW-1]", "3-idx")
    end
    MatrixGet ("Matrix[MXCOL,MXROW+1]", "idx")
    MatrixSet ("Matrix[MXCOL,MXROW+1]", "3-idx")
    MatrixGet ("Matrix[MXCOL-1,MXROW-1]", "v1")
    MatrixGet ("Matrix[MXCOL-1,MXROW]", "v2")
    MatrixGet ("Matrix[MXCOL-1,MXROW+1]", "v3")
    MatrixGet ("Matrix[MXCOL,MXROW-1]", "v4")
    MatrixGet ("Matrix[MXCOL,MXROW]", "v5")
    MatrixGet ("Matrix[MXCOL,MXROW+1]", "v6")
    MatrixGet ("Matrix[MXCOL+1,MXROW-1]", "v7")
    MatrixGet ("Matrix[MXCOL+1,MXROW]", "v8")
    MatrixGet ("Matrix[MXCOL+1,MXROW+1]", "v9")
end
if (v1=v2=v3 | v4=v5=v6 | v7=v8=v9 | v1=v4=v7 | v2=v5=v8 | v3=v6=v9) then
    punteggio=punteggio+100
    pp=pp+100
    PlaySound ("<SrcDir>\M100.wav")
end
if (v1=v2=v3=v4=v5=v6=v7=v8=v9) then
    punteggio=punteggio+1000
    pp=pp+1000
    PlaySound ("<SrcDir>\M1000.wav")
end
end

```

```

if (v1=v5=v9 | v3=v6=v9 | v1=v2=v3=v5 | v7=v8=v9=v5 | v4=v5=v6=v8 | v4=v5=v6=v2)
then
  punteggio=punteggio+500
  pp=pp+500
  PlaySound ("<SrcDir>\M500.wav")
end
DisplayValue ("Text5", "punteggio")
if (NumeroMosse > 70) then
  Show ("HotSpot4")
  Show ("Perso")
  WavePlay ("<SrcDir>\Acqua.wav", "[LOOP]")
  Hide ("HaiVinto")
  Hide ("Help")
  Hide ("Info")
  DisplayValue ("Text6", "livello")
  Show ("Group5")
end
if (NumeroMosse >= 10) then
  Show ("Rectangle20")
end
if (NumeroMosse >= 13) then
  Show ("Rectangle19")
end
if (NumeroMosse >= 16) then
  Show ("Rectangle18")
end
if (NumeroMosse >= 19) then
  Show ("Rectangle17")
end
if (NumeroMosse >= 22) then
  Show ("Rectangle16")
end
if (NumeroMosse >= 25) then
  Show ("Rectangle15")
end
if (NumeroMosse >= 28) then
  Show ("Rectangle14")
end
if (NumeroMosse >= 31) then
  Show ("Rectangle13")
end
if (NumeroMosse >= 34) then
  Show ("Rectangle12")
end
if (NumeroMosse >= 37) then
  Show ("Rectangle11")
end
if (NumeroMosse >= 40) then
  Show ("Rectangle10")
end
if (NumeroMosse >= 43) then
  Show ("Rectangle9")
end
if (NumeroMosse >= 46) then
  Show ("Rectangle8")
end
if (NumeroMosse >= 49) then
  Show ("Rectangle7")
end
if (NumeroMosse >= 52) then
  Show ("Rectangle6")
end
if (NumeroMosse >= 55) then
  Show ("Rectangle5")
end
end

```

```

if (NumeroMosse >= 58) then
  Show("Rectangle4")
end
if (NumeroMosse >= 70) then
  Show("Rectangle3")
end
MatrixGet("Matrix[1,1]", "a1")
MatrixGet("Matrix[1,2]", "a2")
MatrixGet("Matrix[1,3]", "a3")
MatrixGet("Matrix[1,4]", "a4")
MatrixGet("Matrix[1,5]", "a5")
MatrixGet("Matrix[1,6]", "a6")
MatrixGet("Matrix[1,7]", "a7")
MatrixGet("Matrix[1,8]", "a8")
MatrixGet("Matrix[1,9]", "a9")
MatrixGet("Matrix[1,10]", "a10")
MatrixGet("Matrix[2,1]", "a11")
MatrixGet("Matrix[2,2]", "a12")
MatrixGet("Matrix[2,3]", "a13")
MatrixGet("Matrix[2,4]", "a14")
MatrixGet("Matrix[2,5]", "a15")
MatrixGet("Matrix[2,6]", "a16")
MatrixGet("Matrix[2,7]", "a17")
MatrixGet("Matrix[2,8]", "a18")
MatrixGet("Matrix[2,9]", "a19")
MatrixGet("Matrix[2,10]", "a20")
MatrixGet("Matrix[3,1]", "a21")
MatrixGet("Matrix[3,2]", "a22")
MatrixGet("Matrix[3,3]", "a23")
MatrixGet("Matrix[3,4]", "a24")
MatrixGet("Matrix[3,5]", "a25")
MatrixGet("Matrix[3,6]", "a26")
MatrixGet("Matrix[3,7]", "a27")
MatrixGet("Matrix[3,8]", "a28")
MatrixGet("Matrix[3,9]", "a29")
MatrixGet("Matrix[3,10]", "a30")
MatrixGet("Matrix[4,1]", "a31")
MatrixGet("Matrix[4,2]", "a32")
MatrixGet("Matrix[4,3]", "a33")
MatrixGet("Matrix[4,4]", "a34")
MatrixGet("Matrix[4,5]", "a35")
MatrixGet("Matrix[4,6]", "a36")
MatrixGet("Matrix[4,7]", "a37")
MatrixGet("Matrix[4,8]", "a38")
MatrixGet("Matrix[4,9]", "a39")
MatrixGet("Matrix[4,10]", "a40")
MatrixGet("Matrix[5,1]", "a41")
MatrixGet("Matrix[5,2]", "a42")
MatrixGet("Matrix[5,3]", "a43")
MatrixGet("Matrix[5,4]", "a44")
MatrixGet("Matrix[5,5]", "a45")
MatrixGet("Matrix[5,6]", "a46")
MatrixGet("Matrix[5,7]", "a47")
MatrixGet("Matrix[5,8]", "a48")
MatrixGet("Matrix[5,9]", "a49")
MatrixGet("Matrix[5,10]", "a50")
MatrixGet("Matrix[6,1]", "a51")
MatrixGet("Matrix[6,2]", "a52")
MatrixGet("Matrix[6,3]", "a53")
MatrixGet("Matrix[6,4]", "a54")
MatrixGet("Matrix[6,5]", "a55")
MatrixGet("Matrix[6,6]", "a56")
MatrixGet("Matrix[6,7]", "a57")
MatrixGet("Matrix[6,8]", "a58")
MatrixGet("Matrix[6,9]", "a59")

```

```

MatrixGet ("Matrix[6,10]", "a60")
MatrixGet ("Matrix[7,1]", "a61")
MatrixGet ("Matrix[7,2]", "a62")
MatrixGet ("Matrix[7,3]", "a63")
MatrixGet ("Matrix[7,4]", "a64")
MatrixGet ("Matrix[7,5]", "a65")
MatrixGet ("Matrix[7,6]", "a66")
MatrixGet ("Matrix[7,7]", "a67")
MatrixGet ("Matrix[7,8]", "a68")
MatrixGet ("Matrix[7,9]", "a69")
MatrixGet ("Matrix[7,10]", "a70")
MatrixGet ("Matrix[8,1]", "a71")
MatrixGet ("Matrix[8,2]", "a72")
MatrixGet ("Matrix[8,3]", "a73")
MatrixGet ("Matrix[8,4]", "a74")
MatrixGet ("Matrix[8,5]", "a75")
MatrixGet ("Matrix[8,6]", "a76")
MatrixGet ("Matrix[8,7]", "a77")
MatrixGet ("Matrix[8,8]", "a78")
MatrixGet ("Matrix[8,9]", "a79")
MatrixGet ("Matrix[8,10]", "a80")
MatrixGet ("Matrix[9,1]", "a81")
MatrixGet ("Matrix[9,2]", "a82")
MatrixGet ("Matrix[9,3]", "a83")
MatrixGet ("Matrix[9,4]", "a84")
MatrixGet ("Matrix[9,5]", "a85")
MatrixGet ("Matrix[9,6]", "a86")
MatrixGet ("Matrix[9,7]", "a87")
MatrixGet ("Matrix[9,8]", "a88")
MatrixGet ("Matrix[9,9]", "a89")
MatrixGet ("Matrix[9,10]", "a90")
MatrixGet ("Matrix[10,1]", "a91")
MatrixGet ("Matrix[10,2]", "a92")
MatrixGet ("Matrix[10,3]", "a93")
MatrixGet ("Matrix[10,4]", "a94")
MatrixGet ("Matrix[10,5]", "a95")
MatrixGet ("Matrix[10,6]", "a96")
MatrixGet ("Matrix[10,7]", "a97")
MatrixGet ("Matrix[10,8]", "a98")
MatrixGet ("Matrix[10,9]", "a99")
MatrixGet ("Matrix[10,10]", "a100")
if
(a1=a2=a3=a4=a5=a6=a7=a8=a9=a10=a11=a12=a13=a14=a15=a16=a17=a18=a19=a20=a21=a22=a23=
a24=a25=a26=a27=a28=a29=a30=a31=a32=a33=a34=a35=a36=a37=a38=a39=a40=a41=a42=a43=a44=
a45=a46=a47=a48=a49=a50=a51=a52=a53=a54=a55=a56=a57=a58=a59=a60=a61=a62=a63=a64=a65=
a66=a67=a68=a69=a70=a71=a72=a73=a74=a75=a76=a77=a78=a79=a80=a81=a82=a83=a84=a85=a86=
a87=a88=a89=a90=a91=a92=a93=a94=a95=a96=a97=a98=a99=a100) then
  Hide ("Matrix")
  pp=0
  punteggio=20000
  livello = livello + 1
  if (LevelControl = 1) then
    LevelControl = 2
  end
  if (LevelControl = 2) then
    LevelControl = 1
  end
  Show ("HaiVinto")
  Hide ("Help")
  Hide ("Info")
  SfondoAnimato = livello
  if (SfondoAnimato > 4) then
    SfondoAnimato = 1
  end
  Hide ("SfondiAnimati")

```

```

if (livello = 1) then
  Show("ImgFX_1")
end
if (livello = 2) then
  Show("ImgFX_2")
end
if (livello = 3) then
  Show("ImgFX_3")
end
if (livello = 4) then
  Show("ImgFX_4")
end
DisplayValue("Text6","livello")
Show("Group5")
Show("Matrix2")
end
NumeroMosse = NumeroMosse + 1
DisplayValue("Mosse","NumeroMosse")

```

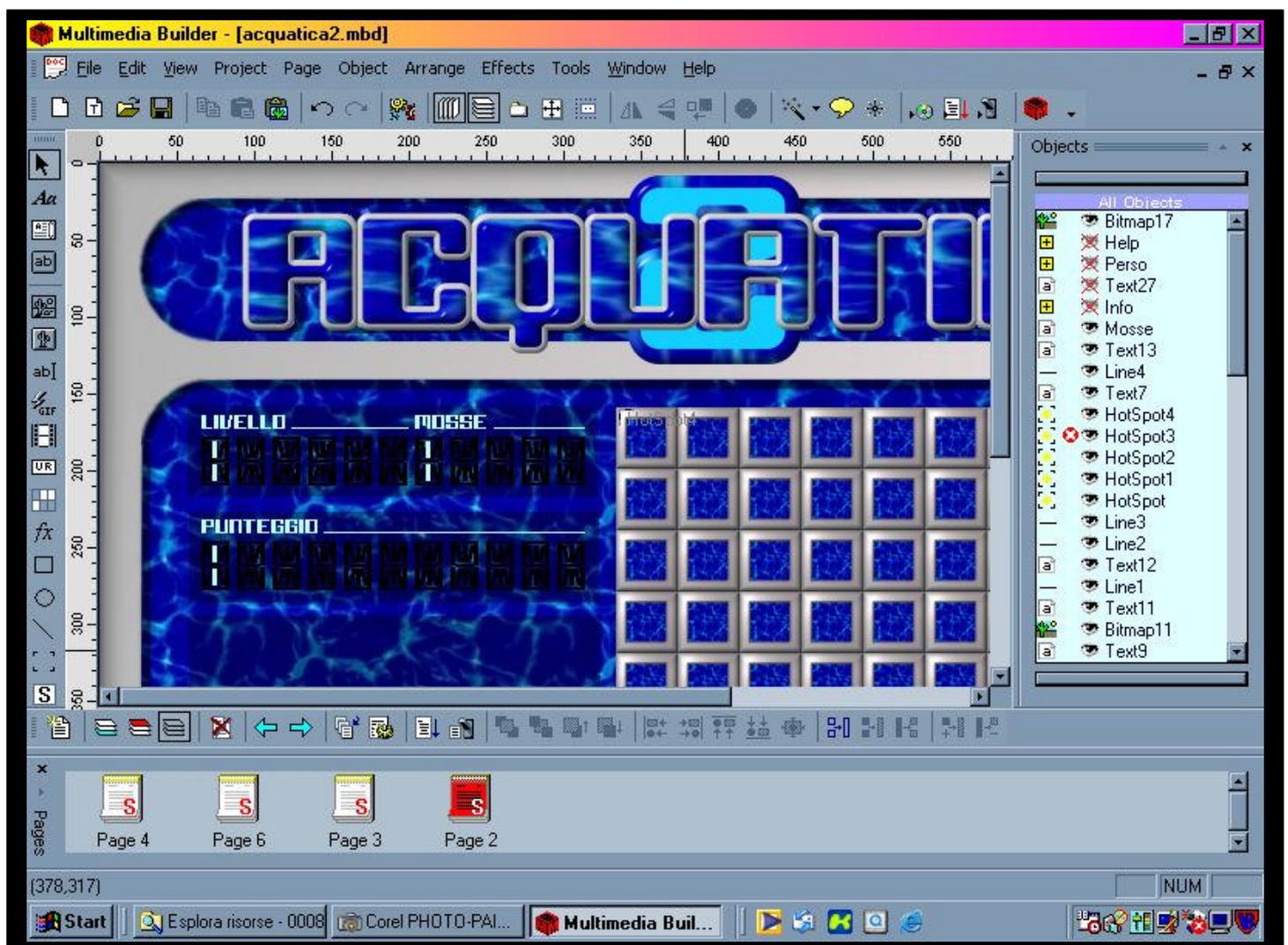


Fig. 3.9: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

□ **Script Script**

```
colore=RND(2)+1
x=RND(6)+1
y=RND(8)+1
MatrixSet("MatrixC[x,y]","colore")
i=i+1
if (i<200) then
  ScriptTimer("Script","50")
end
if (i>=200) then
  Hide("MatrixC")
end
```

## 4. TOON MUSIC GAME

- ❑ Versione 2.7 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2001 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti riguardanti le clip audio nel formato mp3 sono riservati agli artisti ed alle case discografiche che li detengono.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 3,03 Mb

### 4.1 REGOLE DEL GIOCO

Toon Music Game è un gioco a quiz musicale per un solo giocatore. L'obiettivo è di indovinare il maggior numero di sigle dei cartoni animati tra quelle proposte. L'archivio comprende ben 50 clip audio della durata di 20 secondi, con le più famose sigle dei cartoni animati trasmessi dalle reti televisive italiane negli anni 70-80. Da Ufo Robot a Heidi, passando per Chobin e Mazinga.

Per giocare una partita è sufficiente premere il tasto "NUOVO". Inizia così la riproduzione di una clip musicale e viene richiesto di selezionare a quale cartone animato associarla. Ci sono tre possibilità: A, B o C. Per effettuare la propria scelta basta premere il tasto corrispondente. Le impostazioni di default prevedono una durata dell'ascolto pari a 10 secondi, seguiti da altri 10 per consentire al giocatore di fare la propria scelta. In totale ci sono, quindi, 20 secondi per rispondere.

L'aspetto grafico consente al giocatore di tenere sotto controllo tutti gli aspetti della partita. Il primo elemento è costituito dalla barra di stato giallo/rossa. La barra gialla indica la fase d'ascolto mentre quella rossa (scandita da un tic-tac d'orologio) la fase di risposta. Per interrompere brevemente lo svolgimento della partita selezionare/deselezionare la funzione di PAUSA premendo sulla barra di stato.

Ogni risposta esatta fa guadagnare 100 punti mentre una errata ne fa perdere 50. Due suoni diversi identificano l'esito della risposta. Rispondendo esattamente viene riprodotta l'intera clip audio della sigla corrispondente (circa 20 secondi) prima di passare alla domanda successiva. Sul display a destra vengono riassunte tutte le informazioni relative a livello, numero di domanda, punteggio, nonché messaggi vari (risposta corretta/errata, pausa, game over, ...). Ogni livello prevede 10 domande e cinque Livelli. Attraverso il tasto "OPZIONI" è possibile modificare la durata dell'ascolto clip (fino ad un massimo di 20 secondi). Il numero di giocatori non è, invece, modificabile. Le istruzioni per giocare vengono mostrate a video premendo il tasto "COME". Per le informazioni su autore, release, sito web, ecc. basta premere il tasto "INFO".

## 4.2 SUGGERIMENTI E TRUCCHI

E' consigliabile non essere precipitosi nel rispondere al quiz. Attendere la fine della fase d'ascolto e solo dopo fare la propria scelta. Nessun bonus, infatti, viene assegnato per la velocità della risposta. Uno stratagemma che si può adottare, quando si conosce la sigla ma, al momento, la memoria non fa ricordare il nome del cartone animato, consiste nell'utilizzare la funzione di pausa. Questo permette di guadagnare un po' di tempo per pensare.

## 4.3 LA REALIZZAZIONE (THE MAKING OF)

### □ LA STORIA

E' sempre stata un'idea nel cassetto quella di realizzare un videogame che avesse per protagonisti gli eroi dei cartoni animati giapponesi degli anni '70-'80. L'occasione si presenta alla fine degli anni novanta, con la notevole diffusione del formato di compressione audio Mp3 e con la realizzazione, da parte degli studenti del Politecnico di Torino, del famoso "Progetto Prometeo". Questo progetto prevede la raccolta in formato mp3 di tutte le colonne sonore dei cartoni animati trasmessi in Italia, dagli anni '70 fino ad oggi. Canzoni che poi sono state rese disponibili, gratuitamente, in internet. Nonostante le notevoli difficoltà legate ai diritti d'autore, il progetto è ancora attivo e vanta una collezione di oltre 400 brani musicali.

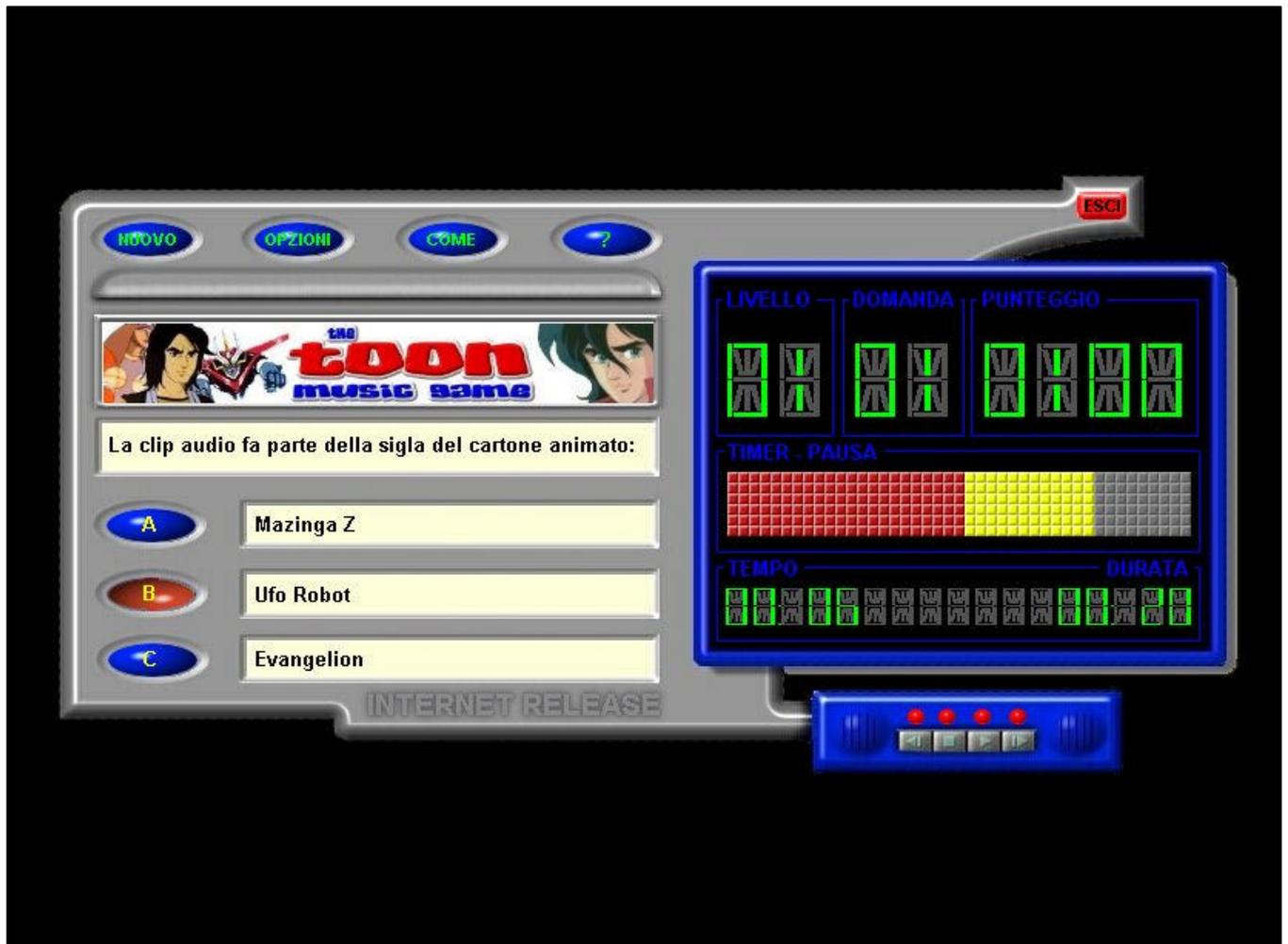


Fig. 4.1: Interfaccia principale del simulatore.

Sfruttando la "Game Engine" sviluppata per il videogioco "Super Secchione" e 50 Brani Mp3 tratti dal Progetto Prometeo nasce, nel 1999, la prima versione di Toon Music Game. La prima release prevede brani mp3 completi ed in qualità medio-alta (Stereo, 44100 Hz, 128b Kbit/sec). Per consentirne il download tramite internet e, quindi, la sua distribuzione capillare,

nonché per evitare problemi di copyright, le canzoni vengono rimpiazzate nelle versioni successive da clip di soli 20 secondi ed in bassa qualità.

Dal 1999 il videogame è stato prelevato dal sito web [www.evweb.it](http://www.evweb.it) più di 5000 volte.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 4.10).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 4.2 un'immagine della prima versione del videogame. In fig. 4.3 l'interfaccia principale del videogioco. In fig. 4.4 come appaiono le informazioni su autore e release del gioco. In fig. 4.5 c'è l' Help On Line con le istruzioni per giocare. In fig. 4.6: Schermata di game over in caso di vittoria. In fig. 4.7 e 4.8 le due schermate di game over in caso di sconfitta. Infine, in fig. 4.9 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

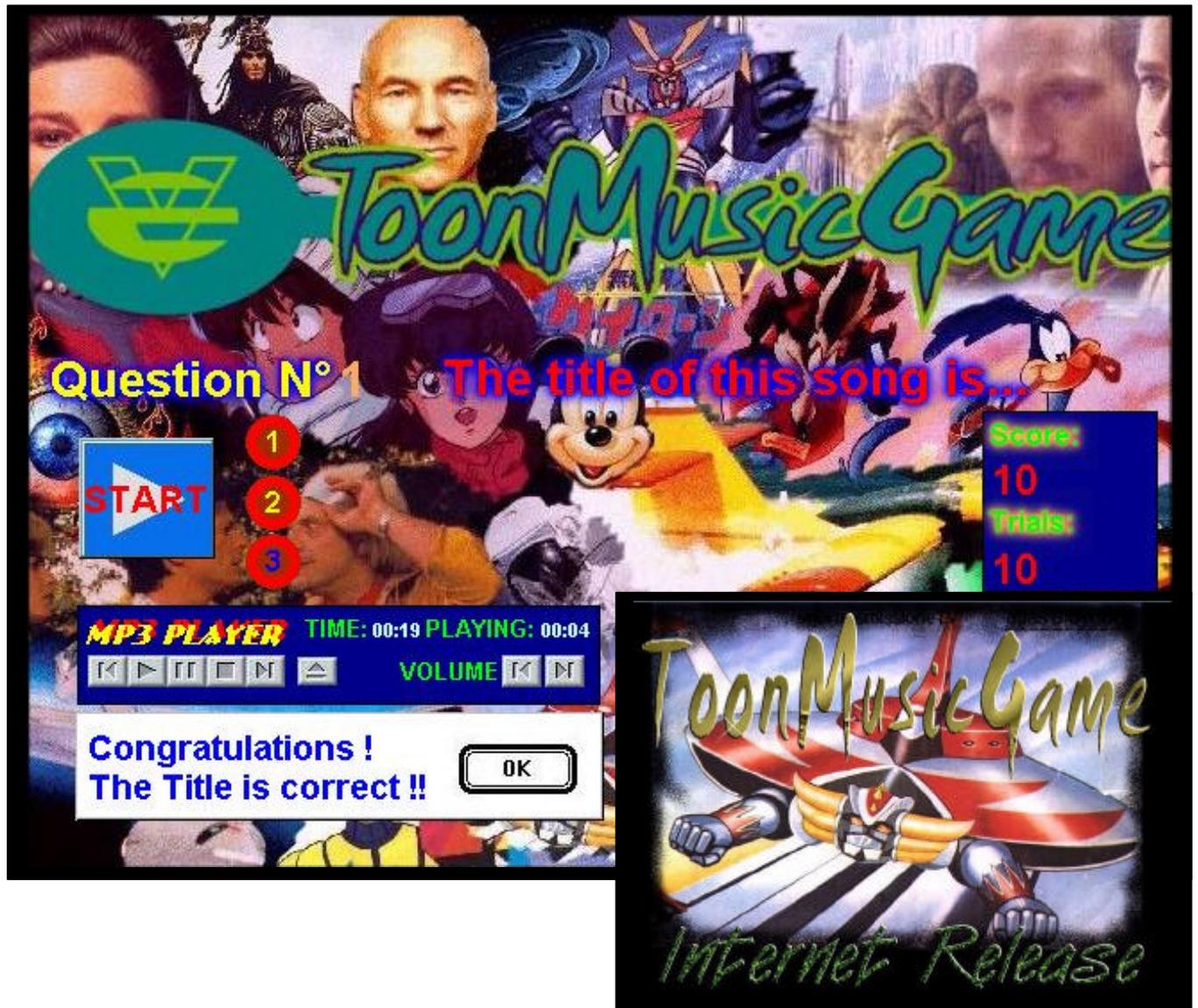


Fig. 4.2: Un'immagine della prima versione del videogame.



Fig. 4.3: La finestra con le opzioni di gioco.



Fig. 4.4: Informazioni su autore e release del gioco.



Fig. 4.5: Help On Line - Istruzioni per giocare.



Fig. 4.6: Schermata di game over in caso di vittoria.



Fig. 4.7: Schermata di game over in caso di sconfitta.



Fig. 4.8: Schermata di game over in caso di sconfitta.

## 4.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

InstallFont ("<Embedded>\Led_real.ttf")
InstallFont ("<Embedded>\Arial.ttf")
PageTimer ("1000", "Page 3")
MP3ListLoad ("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
NDom=1
Click=0
riga=0
for i=1 to 22
  for j=1 to 6
    MatrixSet ("Matrix[i,j]", "3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet ("Matrix[i,j]", "2")
  next j
next i
for ElementoLista=1 to 50
  LoadText ("NumeDoman$", "<List>ElementoLista")
  NumeroDomanda [ElementoLista]=VAL (NumeDoman$)
next ElementoLista
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText ("Text8", "Testo_NLivello$")
LoadText ("Text9", "Testo_NDomanda$")
MP3ListLoad ("<Embedded>\ToonA.txt")
LoadText ("RispostaA", "<List>NumeroDomanda [NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonB.txt")
LoadText ("RispostaB", "<List>NumeroDomanda [NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonC.txt")
LoadText ("RispostaC", "<List>NumeroDomanda [NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonE.txt")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda [NumeroDomandeTotali]")
N=NumeroDomanda [NumeroDomandeTotali]
Nu$=CHAR (N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
Show ("R_ascolta")
PlaySound ("File$")
RunScript ("Timer")

```

### □ Blu Control Botton

```

pausa=0
Hide ("Pic_Box")
Show ("PIC_VIEWS")
Hide ("Animations")
Hide ("R_pausa")
Hide ("AniGif")
Hide ("AniGif_Rack")
Risposta$='B'
RunScript ("Script")

```

```
Click=1
Show("TB")
Show("BloccaTasti")
Hide("R_ascolta")
```

#### □ Tasto Pausa

```
if (pausa=0) then
  pausa=1
  Show("R_pausa")
  Show("HotSpot9")
  Hide("R_ascolta")
  MP3Pause()
End
```

#### □ Script Timer

```
if (pausa=0) then
  riga=riga+1
  for k=1 to 6
    MatrixSet("Matrix[44-riga,k]","1")
  next k
  WavePlay("tone","")
  if (riga<>43) then
    ScriptTimer("Timer","tempo")
  end
  if (riga=43 & Click=0) then
    RunScript("Script")
  end
  if (riga=20) then
    MP3Stop()
    Hide("R_ascolta")
  end
end
end
```

#### □ Script Fine1

```
Show("Congratulazioni")
WaveStop()
WavePlay("Bravo","")
```

#### □ Script Fine2

```
WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver")
Click=1
```

#### □ Script Fine3

```
WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver2")
Click=1
```

#### □ Script Mp3Control

```
Show("Mp3Data")
Hide("R_corretta")
pausa=0
Hide("R_pausa")
Hide("HotSpot9")
Show("HotSpot8")
pausa=0
ScriptTimer("NuovaDomanda","20000")
```

## □ Script NuovaDomanda

```

MP3Stop()
if (NDom=10) then
  NDom=0
  DomandeGiuste=0
  NumeroLivello=NumeroLivello+1
end
riga=0
Click=0
Risposta$=''
Hide("Mp3Data")
Hide("R_errata")
Hide("R_corretta")
Hide("R_temposcaduto")
Hide("Evide")
Hide("R_pausa")
Show("HotSpot14")
for i=1 to 22
  for j=1 to 6
    MatrixSet("Matrix[i,j]","3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet("Matrix[i,j]","2")
  next j
next i
NumeroDomandeTotali=NumeroDomandeTotali+1
NDom=NDom+1
if (NumeroLivello=1) then
  Testo_NLivello$='01'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=2) then
  Testo_NLivello$='02'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=3) then
  Testo_NLivello$='03'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=4) then
  Testo_NLivello$='04'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=5) then
  Testo_NLivello$='05'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=6) then
  Testo_NLivello$='06'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=7) then
  Testo_NLivello$='07'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=8) then
  Testo_NLivello$='08'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=9) then
  Testo_NLivello$='09'
  LoadText("Text8","Testo_NLivello$")

```

```

end
if (NumeroLivello>=10) then
    Testo1$=CHAR(NumeroLivello)
    LoadText ("Text8", "Testo1$")
end
if (NDom=1) then
    Testo_NDomanda$='01'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=2) then
    Testo_NDomanda$='02'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=3) then
    Testo_NDomanda$='03'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=4) then
    Testo_NDomanda$='04'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=5) then
    Testo_NDomanda$='05'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=6) then
    Testo_NDomanda$='06'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=7) then
    Testo_NDomanda$='07'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=8) then
    Testo_NDomanda$='08'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=9) then
    Testo_NDomanda$='09'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom>=10) then
    Testo_NDomanda$=CHAR(NDom)
    LoadText ("Text9", "Testo_NDomanda$")
end
Hide ("BloccaTasti")
MP3ListLoad ("<Embedded>\ToonA.txt")
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonB.txt")
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonC.txt")
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonE.txt")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
PlaySound ("File$")
RunScript ("Timer")
Show ("R_ascolta")

```

## □ Script Script

```

MP3Stop()
Hide ("Info")

```

```

Hide("Come")
Hide("GameOver")
Hide("GameOver2")
Hide("Congratulazioni")
if (Click=0 & Risposta$='') then
  Show("R_temposcaduto")
  WaveStop()
  WavePlay("Errore","")
  punteggio=punteggio-5
end
if (Risposta$=RispostaEsatta$) then
  Show("R_corretta")
  WaveStop()
  PlaySound("File$")
  Hide("HotSpot14")
  punteggio=punteggio+100
  DomandeGiuste=DomandeGiuste+1
end
if (Risposta$<>RispostaEsatta$ & Risposta$<>'') then
  Show("R_errata")
  WaveStop()
  WavePlay("Errore","")
  punteggio=punteggio-50
end
if (punteggio<=0) then
  punteggio=0
  Testo_Punteggio$='0000'
end
if (punteggio>=0 & punteggio<10) then
  Testo_Punteggio$='000'+CHAR(punteggio)
end
if (punteggio>=10 & punteggio<100) then
  Testo_Punteggio$='00'+CHAR(punteggio)
end
if (punteggio>=100 & punteggio<1000) then
  Testo_Punteggio$='0'+CHAR(punteggio)
end
if (punteggio>=1000) then
  Testo_Punteggio$=CHAR(punteggio)
end
LoadText("Text10","Testo_Punteggio$")
if (NumeroDomandeTotali<50 & Risposta$=RispostaEsatta$) then
  if (NDom<>10 | DomandeGiuste>=6) then
    ScriptTimer("Mp3Control","2000")
  end
end
if (NumeroDomandeTotali<50 & Risposta$<>RispostaEsatta$) then
  if (NDom<>10 | DomandeGiuste>=6) then
    ScriptTimer("NuovaDomanda","2000")
  end
end
if (punteggio=5000 & NumeroLivello=5) then
  MP3Stop()
  Hide("R_corretta")
  ScriptTimer("Finel","2000")
end
if (NDom=10 & DomandeGiuste<6 & NumeroLivello<>5) then
  MP3Stop()
  Hide("R_corretta")
  Hide("R_errata")
  Hide("R_temposcaduto")
  ScriptTimer("Fine2","2000")
end
if (NDom=10 & punteggio>=2000 & punteggio<5000 & NumeroLivello=5) then
  ScriptTimer("Fine3","2000")

```

```

Hide("R_corretta")
Hide("R_errata")
Hide("R_temposcaduto")
MP3Stop()
end

```

## □ Tasto Nuovo

```

if (pausa=1) then
    pausa=0
    Hide("R_pausa")
    Hide("HotSpot9")
    Show("HotSpot8")
end
Hide("GameOver")
Hide("GameOver2")
Hide("Congratulazioni")
Hide("Info")
Hide("Come")
Hide("BloccaTasti")
Hide("Evide")
Hide("Opzioni")
NumeroLivello=1
punteggio=0
DomandeGiuste=0
Testo_Punteggio$='0000'
LoadText("Text10","Testo_Punteggio$")
NDom=1
Click=0
Risposta$=''
Hide("R_errata")
Hide("R_corretta")
Hide("R_temposcaduto")
riga=0
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
for i=1 to 22
    for j=1 to 6
        MatrixSet("Matrix[i,j]","3")
    next j
next i
for i=23 to 44
    for j=1 to 6
        MatrixSet("Matrix[i,j]","2")
    next j
next i
for ElementoLista=1 to 50
    LoadText("NumeDoman$","<List>ElementoLista")
    NumeroDomanda[ElementoLista]=VAL(NumeDoman$)
next ElementoLista
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText("Text8","Testo_NLivello$")
LoadText("Text9","Testo_NDomanda$")
MP3ListLoad("<Embedded>\ToonA.txt")
LoadText("RispostaA","<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonB.txt")
LoadText("RispostaB","<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonC.txt")

```

```

LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonE.txt")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
PlaySound ("File$")
RunScript ("Timer")
Show ("R_ascolta")
    
```

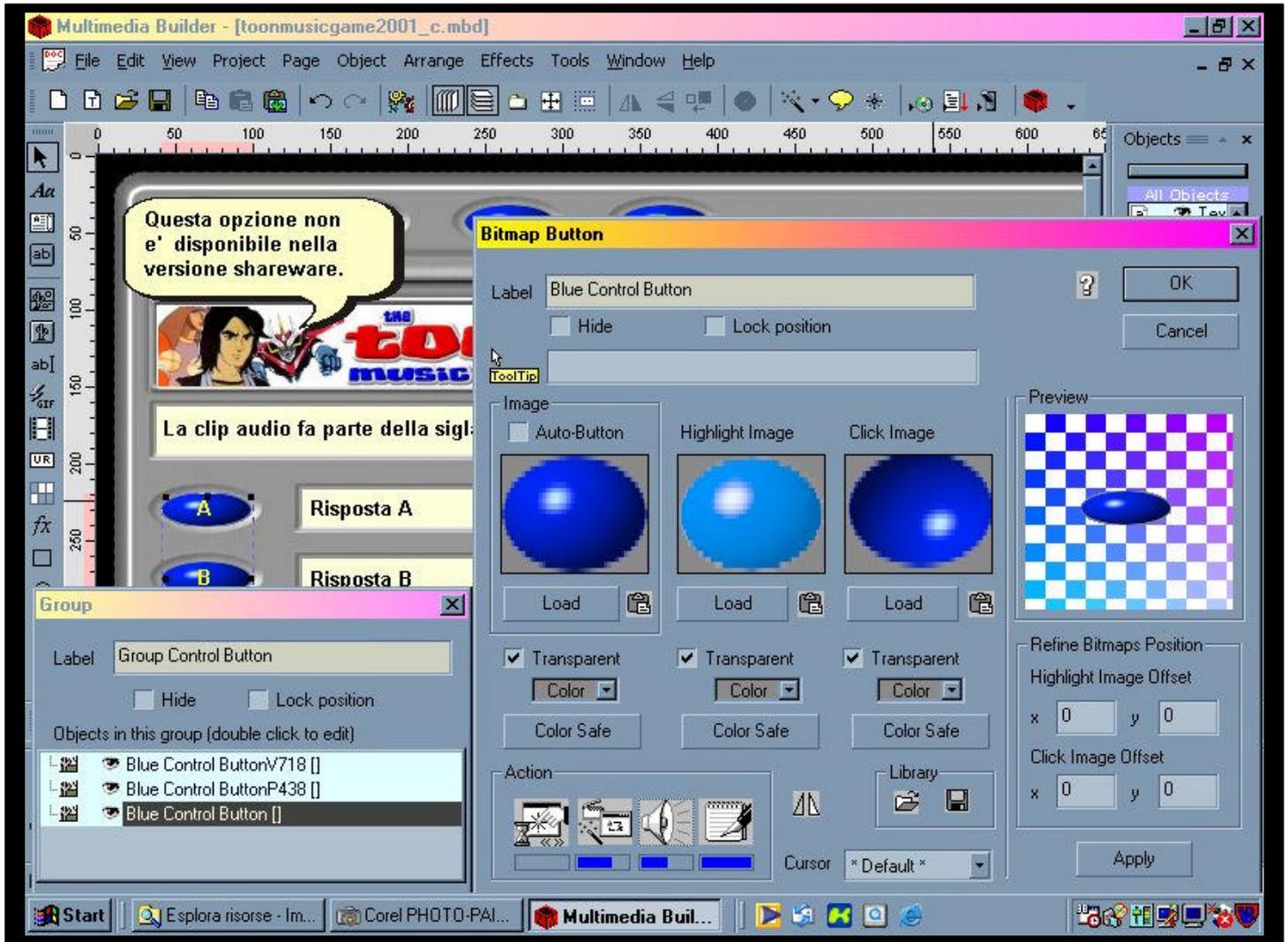


Fig. 4.9: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

## 5. JAVA GAMES VIRTUAL CONSOLE

- ❑ Versione 2.0 - Freeware.
- ❑ EVSoftware - Copyright © 2001-2003 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.
- ❑ I videogames java della collezione sono di proprietà esclusiva dei rispettivi autori.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 4,84 Mb

### 5.1 REGOLE DEL GIOCO

JGames 2.0 è una console virtuale per videogiochi JAVA che raccoglie 28 giochi freeware in un'unica collezione. L'interfaccia grafica è molto intuitiva. Dalla barra di menu in basso è possibile selezionare i vari sottomenu disponibili: Giochi 1, Giochi 2,...

Ogni menu è composto da quattro giochi. Per giocare è sufficiente cliccare sulle immagini. Ogni immagine descrive il corrispettivo videogame, la cui categoria è visualizzata nell'etichetta laterale inserita nella cornice (esempio: Strategia, Guerra, Sport,...). Per il corretto funzionamento, JGames 2.0 richiede Microsoft Internet Explorer 4 o superiore.

Di seguito sono riportati i videogame della collezione:



**1. Torpedo Alley:** gioco di guerra sottomarina in cui bisogna affondare le navi nemiche da un sommergibile, evitando di colpire le imbarcazioni civili e della Croce Rossa.

**2. Cross Section:** il dinosauro protagonista deve raggiungere la sua meta scavalcando gli ostacoli ed accumulando punti.

**3. Tail Gunner:** un videogame di fantascienza dove si pilota un'astronave e si devono distruggere i velivoli nemici. Notevole la grafica vettoriale realizzata.



4. **Basket Ball:** semplice versione del gioco del basket. L'obiettivo è definire una traiettoria e la forza da imprimere alla palla per fare canestro.



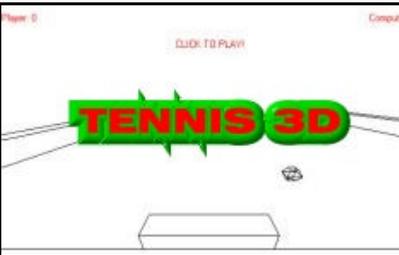
5. **Bustout:** il famosissimo passatempo dei mattoncini da colpire e distruggere con la pallina e la racchetta.



6. **Iceblox:** bisogna aiutare il pinguino a superare i cubetti di ghiaccio che intralciano la sua strada.



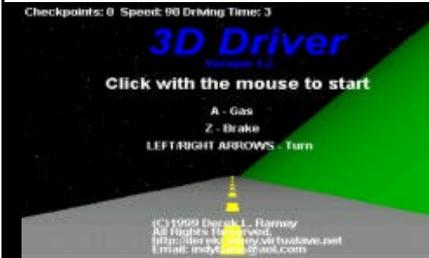
7. **Lunar Lander:** il più famoso tra i classici. Si deve far atterrare il modulo lunare su degli altipiani non proprio facili da raggiungere, prima che finisca il carburante.



8. **Tennis 3D:** una sorta di ping pong virtuale dalla grafica piuttosto "essenziale".



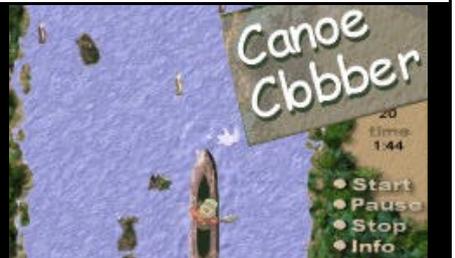
9. **Timepilot 3D:** gioco tridimensionale di guerra aerea nel migliore stile "Arcade".



10. **3D Driver:** simulazione di una corsa automobilistica lungo una scogliera e attraverso improbabili gallerie.



11. **Warp:** videogame di fantascienza con visione dall'alto del campo di battaglia spaziale.



12. **Canoe Clobber:** l'obiettivo è guidare il canoista attraverso le rapide, raccogliendo oggetti bonus lungo il percorso ed evitando gli ostacoli.



13. **Crosswords:** interessante rivisitazione virtuale di un classico dell'enigmistica cartacea, il cruciverba. Rigorosamente in lingua inglese.



14. **Defender Of The Moon:** bisogna impedire la conquista della luna distruggendo tutte le astronavi aliene che tentano l'atterraggio.



15. **Forza 4:** si vince allineando 4 pedine dello stesso colore in verticale, orizzontale e diagonale.



**16. Labyrinth:** sfida contro il computer che genera virtualmente un labirinto nei formati 3D "Square" e "Hexagonal".



**17. Memory:** per vincere basta ricordare la posizione delle carte e formare le coppie di carte uguali.



**18. Missile Commando 2:** Gioco di guerra in cui occorre gestire la contraerea per distruggere i missili nemici sparati contro la città.



**19. Carpet Golf 3D:** ottimo gioco di simulazione di minigolf su tappeto. Con diversi livelli, pratica e tornei.



**20. Santa Claus Snowracer:** Babbo Natale in snowboard deve evitare alberi e rocce per arrivare in fondo alla pista.



**21. Star Battle:** gioco di battaglie spaziali



**22. PacMan:** semplice ma divertente: evitare i fantasmini lungo il percorso e mangiare più che si può per avanzare di livello



**23. Wastem:** tiro al piattello



**24. Zapper:** Gioco di logica



**25. Dugout:** aiuta il protagonista attraverso i vari livelli, accumulando punti, raccogliendo oggetti ed evitando gli ostacoli.



**26. Urbanoids:** gioco di simulazione vagamente ispirato al famoso "Simcity"



**27. Slot Machine:** un classico dei casinò in una versione riveduta e corretta per PC.



28. **Ufo Attack:** grafica essenziale e regole elementari per questo classico dei giochi d'annata.

## 5.2 SUGGERIMENTI E TRUCCHI

La maggior parte dei videogame di questa collezione è un classico o una sua rielaborazione. Non ci sono trucchi o suggerimenti significativi o degni di nota.

## 5.3 LA REALIZZAZIONE (THE MAKING OF)

### □ LA STORIA

Nel 2001, dopo aver utilizzato molte applet java con videogames freeware nel sito web [www.evweb.it](http://www.evweb.it), è venuta l'idea di sfruttare il materiale raccolto per una consolle di videogame java che li raccogliesse tutti.

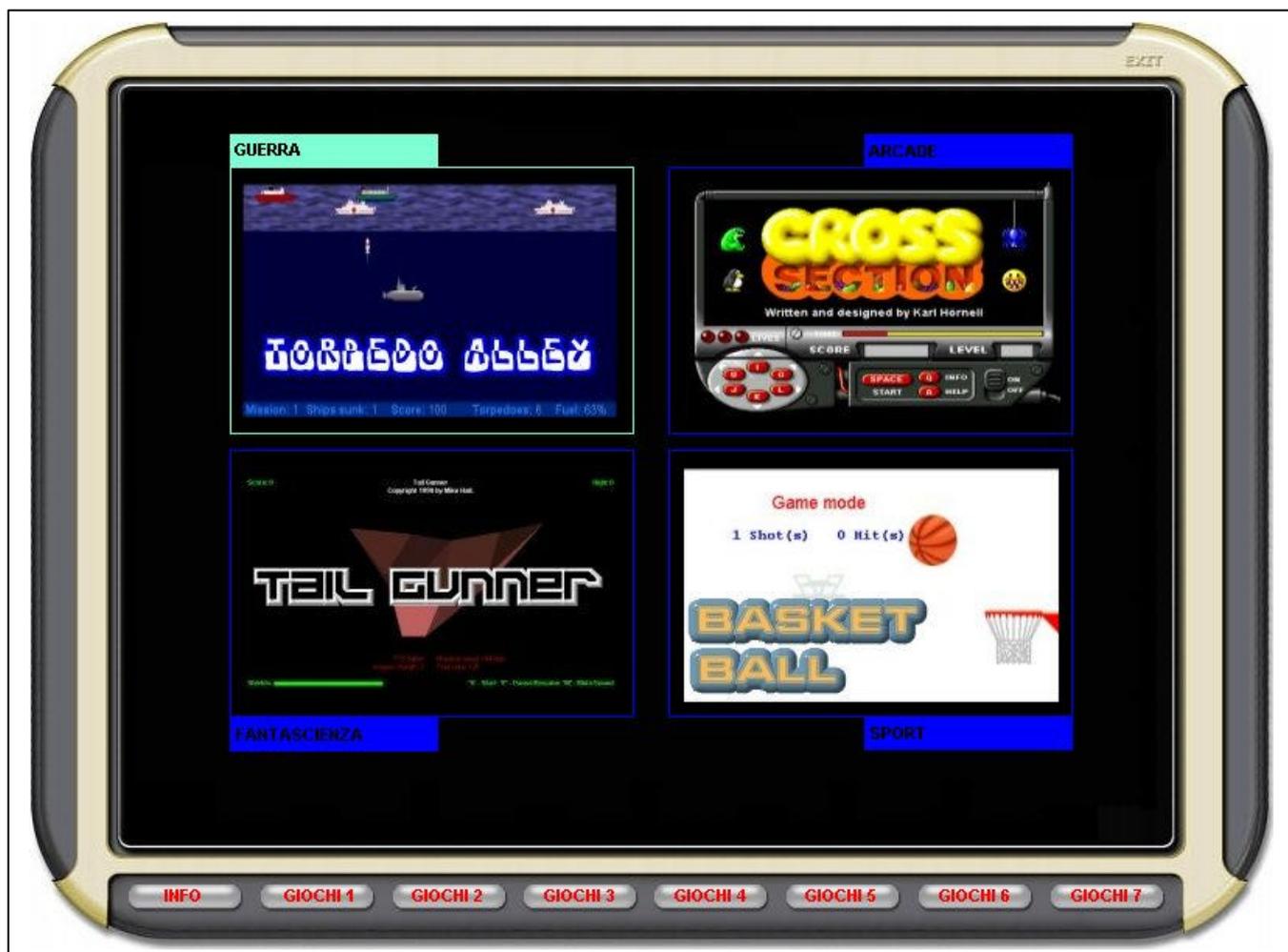


Fig. 5.1: Interfaccia principale del videogame.

Di fatto il progetto ha previsto la realizzazione di una semplice interfaccia utente con dei menu grafici per la selezione del gioco. L'aspetto più interessante di questo software è stato l'integrazione di un oggetto come il browser web "Microsoft Internet Explorer" all'interno di un'applicazione multimedia Builder. L'esperimento ha manifestato fin dall'inizio le sue potenzialità, ma anche i suoi limiti. La versione del browser condiziona il corretto funzionamento dei videogiochi, esattamente come accade quando si gioca in rete.

□ **I SOFTWARE UTILIZZATI**

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine:* Mediachance Multimedia Builder (vedi fig. 5.10).
- *Grafica e Immagini:* Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica:* Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

□ **BOZZE E SCREENSHOT**

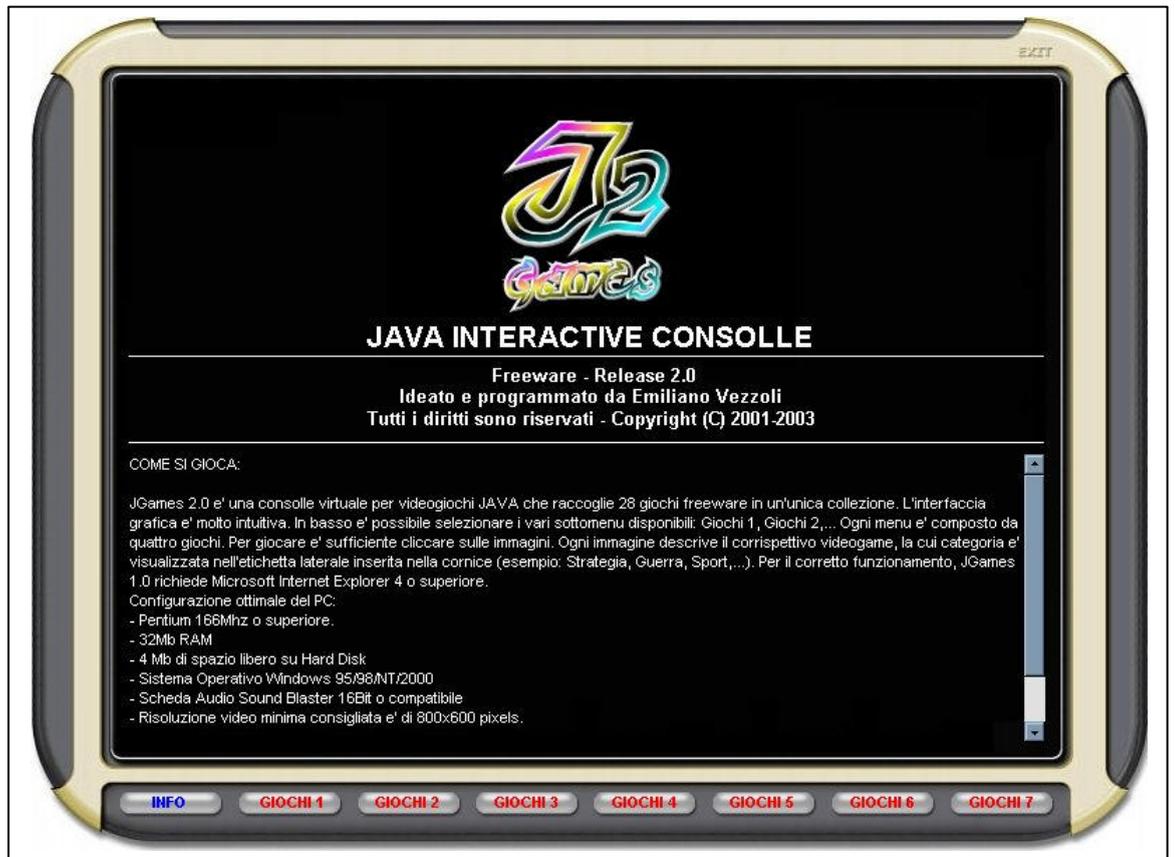
Ecco alcuni screenshot tratti dall'ultima versione del gioco.

In fig. 5.2 un'immagine tratta dell'interfaccia principale del videogame.

In fig. 5.3 la finestra con le informazioni su autore, release, sito internet e le istruzioni del gioco.

Nelle figg. 5.4, 5.5, 5.6, 5.7, 5.8 e 5.9 le finestre con i vari sottomenu di selezione videogame.

Infine, in fig. 5.10 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



**Fig. 5.3:** Al centro la finestra con le istruzioni per giocare, informazioni su autore, release e sito web.



Fig. 5.4: Il menu "Giochi 2".



Fig. 5.5: Il menu "Giochi 3".

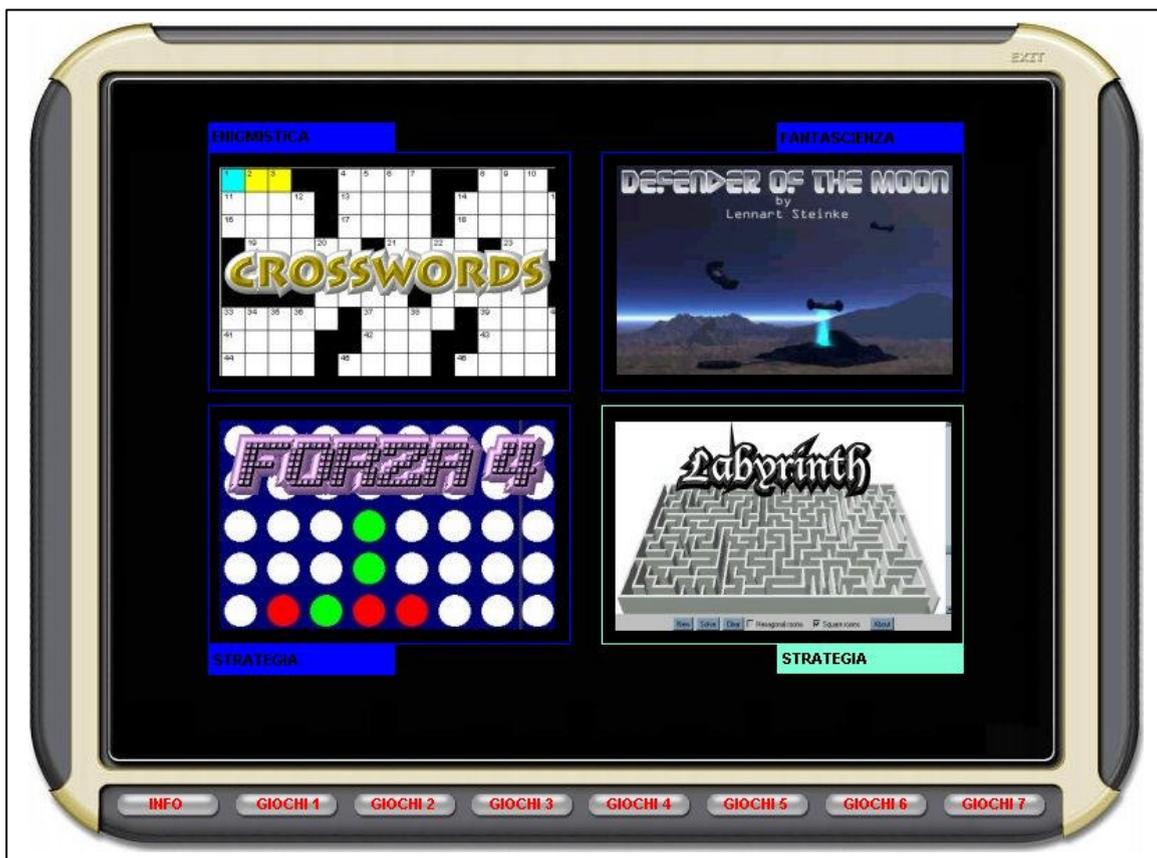


Fig. 5.6: Il menu "Giochi 4".

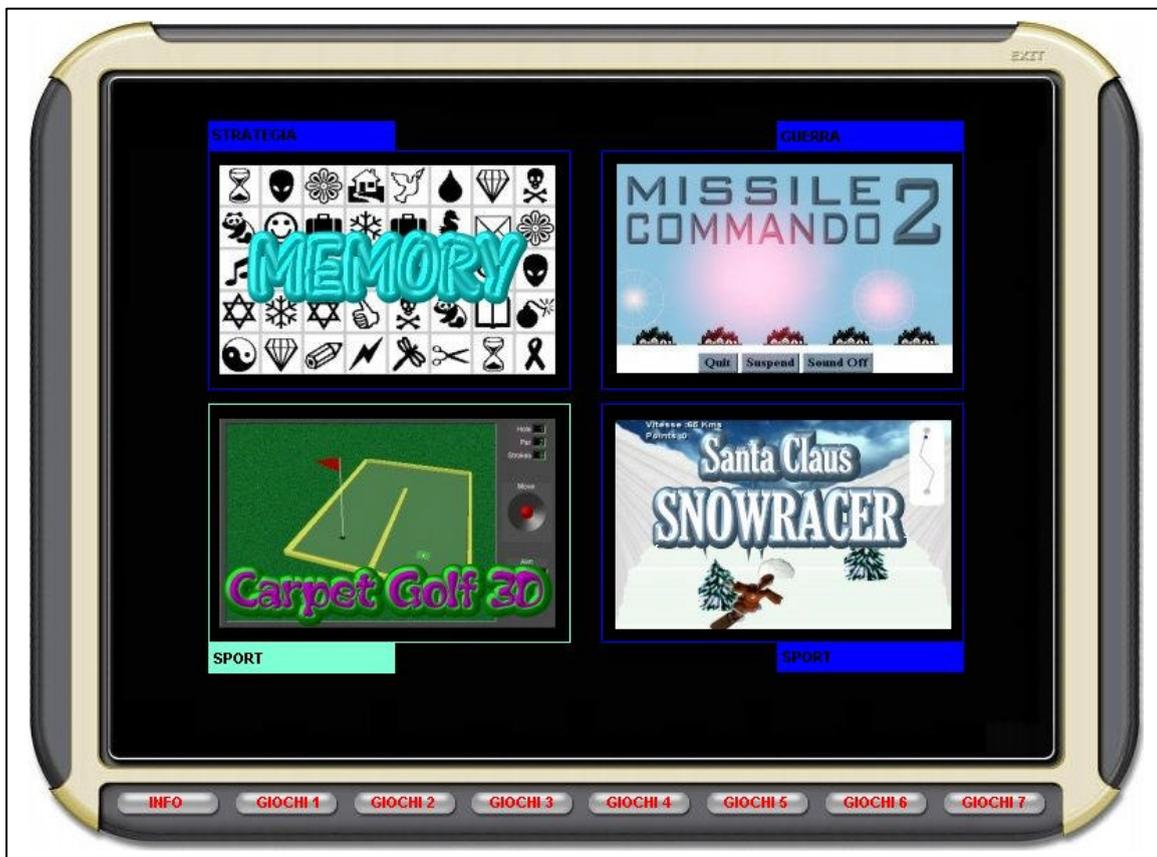


Fig. 5.7: Il menu "Giochi 5".



Fig. 5.8: Il menu "Giochi 6".



Fig. 5.9: Il menu "Giochi 7".

## 5.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Script Tasto Info

```
Hide("Browser")
Browser("Browser", "Stop")
Browser("Browser", "Close")
Show("GroupInfo")
Hide("GroupLogo")
Hide("GroupMenu")
Show("Infooff")
Hide("Infoon")
```

### □ Script Tasto Giochi1

```
Hide("Browser")
Browser("Browser", "Stop")
Browser("Browser", "Close")
Show("GroupMenu")
Hide("RECl1a")
Hide("REC2a")
Hide("REC3a")
Hide("REC4a")
Hide("GroupLogo")
Hide("GroupInfo")
menu = 1
ReplaceImage("Bitmap1", "<SrcDir>\Data\game01\Demo01.jpg")
ReplaceImage("Bitmap2", "<SrcDir>\Data\game02\Demo02.jpg")
ReplaceImage("Bitmap3", "<SrcDir>\Data\game03\Demo03.jpg")
ReplaceImage("Bitmap4", "<SrcDir>\Data\game04\Demo04.jpg")
Categoria_1_$='GUERRA'
Categoria_2_$='ARCADE'
Categoria_3_$='FANTASCIENZA'
Categoria_4_$='SPORT'
LoadText("Categorial", "Categoria_1_$")
LoadText("Categoria2", "Categoria_2_$")
LoadText("Categoria3", "Categoria_3_$")
LoadText("Categoria4", "Categoria_4_$")
```

### □ Script Tasto Giochi2

```
Hide("Browser")
Browser("Browser", "Stop")
Browser("Browser", "Close")
Show("GroupMenu")
Hide("RECl1a")
Hide("REC2a")
Hide("REC3a")
Hide("REC4a")
Hide("GroupLogo")
Hide("GroupInfo")
menu = 2
ReplaceImage("Bitmap1", "<SrcDir>\Data\game05\Demo05.jpg")
ReplaceImage("Bitmap2", "<SrcDir>\Data\game06\Demo06.jpg")
ReplaceImage("Bitmap3", "<SrcDir>\Data\game07\Demo07.jpg")
ReplaceImage("Bitmap4", "<SrcDir>\Data\game08\Demo08.jpg")
Categoria_1_$='ARCADE'
Categoria_2_$='ARCADE'
Categoria_3_$='FANTASCIENZA'
Categoria_4_$='SPORT'
LoadText("Categorial", "Categoria_1_$")
LoadText("Categoria2", "Categoria_2_$")
LoadText("Categoria3", "Categoria_3_$")
```

```
LoadText ("Categoria4", "Categoria_4_$")
```

### □ Script Tasto Giochi3

```
Hide ("Browser")
Browser ("Browser", "Stop")
Browser ("Browser", "Close")
Show ("GroupMenu")
Hide ("REC1a")
Hide ("REC2a")
Hide ("REC3a")
Hide ("GroupLogo")
Hide ("GroupInfo")
Hide ("REC4a")
menu = 3
ReplaceImage ("Bitmap1", "<SrcDir>\Data\game09\Demo09.jpg")
ReplaceImage ("Bitmap2", "<SrcDir>\Data\game10\Demo10.jpg")
ReplaceImage ("Bitmap3", "<SrcDir>\Data\game11\Demo11.jpg")
ReplaceImage ("Bitmap4", "<SrcDir>\Data\game12\Demo12.jpg")
Categoria_1_$='GUERRA'
Categoria_2_$='SPORT'
Categoria_3_$='GUERRA'
Categoria_4_$='SPORT'
LoadText ("Categoria1", "Categoria_1_$")
LoadText ("Categoria2", "Categoria_2_$")
LoadText ("Categoria3", "Categoria_3_$")
LoadText ("Categoria4", "Categoria_4_$")
```

### □ Script Tasto Giochi4

```
Hide ("Browser")
Browser ("Browser", "Stop")
Browser ("Browser", "Close")
Show ("GroupMenu")
Hide ("REC1a")
Hide ("REC2a")
Hide ("REC3a")
Hide ("GroupLogo")
Hide ("GroupInfo")
Hide ("REC4a")
menu = 4
ReplaceImage ("Bitmap1", "<SrcDir>\Data\game13\Demo13.jpg")
ReplaceImage ("Bitmap2", "<SrcDir>\Data\game14\Demo14.jpg")
ReplaceImage ("Bitmap3", "<SrcDir>\Data\game15\Demo15.jpg")
ReplaceImage ("Bitmap4", "<SrcDir>\Data\game16\Demo16.jpg")
Categoria_1_$='ENIGMISTICA'
Categoria_2_$='FANTASCIENZA'
Categoria_3_$='STRATEGIA'
Categoria_4_$='STRATEGIA'
LoadText ("Categoria1", "Categoria_1_$")
LoadText ("Categoria2", "Categoria_2_$")
LoadText ("Categoria3", "Categoria_3_$")
LoadText ("Categoria4", "Categoria_4_$")
```

### □ Script Tasto Giochi5

```
Hide ("Browser")
Browser ("Browser", "Stop")
Browser ("Browser", "Close")
Show ("GroupMenu")
Hide ("REC1a")
Hide ("REC2a")
Hide ("REC3a")
Hide ("GroupLogo")
Hide ("GroupInfo")
Hide ("REC4a")
```

```

menu = 5
ReplaceImage ("Bitmap1", "<SrcDir>\Data\game17\Demo17.jpg")
ReplaceImage ("Bitmap2", "<SrcDir>\Data\game18\Demo18.jpg")
ReplaceImage ("Bitmap3", "<SrcDir>\Data\game19\Demo19.jpg")
ReplaceImage ("Bitmap4", "<SrcDir>\Data\game20\Demo20.jpg")
Categoria_1_$= 'STRATEGIA'
Categoria_2_$= 'GUERRA'
Categoria_3_$= 'SPORT'
Categoria_4_$= 'SPORT'
LoadText ("Categorial", "Categoria_1_$")
LoadText ("Categoria2", "Categoria_2_$")
LoadText ("Categoria3", "Categoria_3_$")
LoadText ("Categoria4", "Categoria_4_$")

```

## □ Script Tasto Giochi6

```

Hide ("Browser")
Browser ("Browser", "Stop")
Browser ("Browser", "Close")
Show ("GroupMenu")
Hide ("RECl a")
Hide ("REC2a")
Hide ("REC3a")
Hide ("REC4a")
Hide ("GroupLogo")
Hide ("GroupInfo")
menu = 6
ReplaceImage ("Bitmap1", "<SrcDir>\Data\game21\Demo21.jpg")
ReplaceImage ("Bitmap2", "<SrcDir>\Data\game22\Demo22.jpg")
ReplaceImage ("Bitmap3", "<SrcDir>\Data\game23\Demo23.jpg")
ReplaceImage ("Bitmap4", "<SrcDir>\Data\game24\Demo24.jpg")
Categoria_1_$= 'FANTASCIENZA'
Categoria_2_$= 'ARCADE'
Categoria_3_$= 'SPORT'
Categoria_4_$= 'STRATEGIA'
LoadText ("Categorial", "Categoria_1_$")
LoadText ("Categoria2", "Categoria_2_$")
LoadText ("Categoria3", "Categoria_3_$")
LoadText ("Categoria4", "Categoria_4_$")

```

## □ Script Tasto Giochi7

```

Hide ("Browser")
Browser ("Browser", "Stop")
Browser ("Browser", "Close")
Show ("GroupMenu")
Hide ("RECl a")
Hide ("REC2a")
Hide ("REC3a")
Hide ("REC4a")
Hide ("GroupLogo")
Hide ("GroupInfo")
menu = 7
ReplaceImage ("Bitmap1", "<SrcDir>\Data\game25\Demo25.jpg")
ReplaceImage ("Bitmap2", "<SrcDir>\Data\game26\Demo26.jpg")
ReplaceImage ("Bitmap3", "<SrcDir>\Data\game27\Demo27.jpg")
ReplaceImage ("Bitmap4", "<SrcDir>\Data\game28\Demo28.jpg")
Categoria_1_$= 'ARCADE'
Categoria_2_$= 'STRATEGIA'
Categoria_3_$= 'ARCADE'
Categoria_4_$= 'FANTASCIENZA'
LoadText ("Categorial", "Categoria_1_$")
LoadText ("Categoria2", "Categoria_2_$")
LoadText ("Categoria3", "Categoria_3_$")
LoadText ("Categoria4", "Categoria_4_$")

```

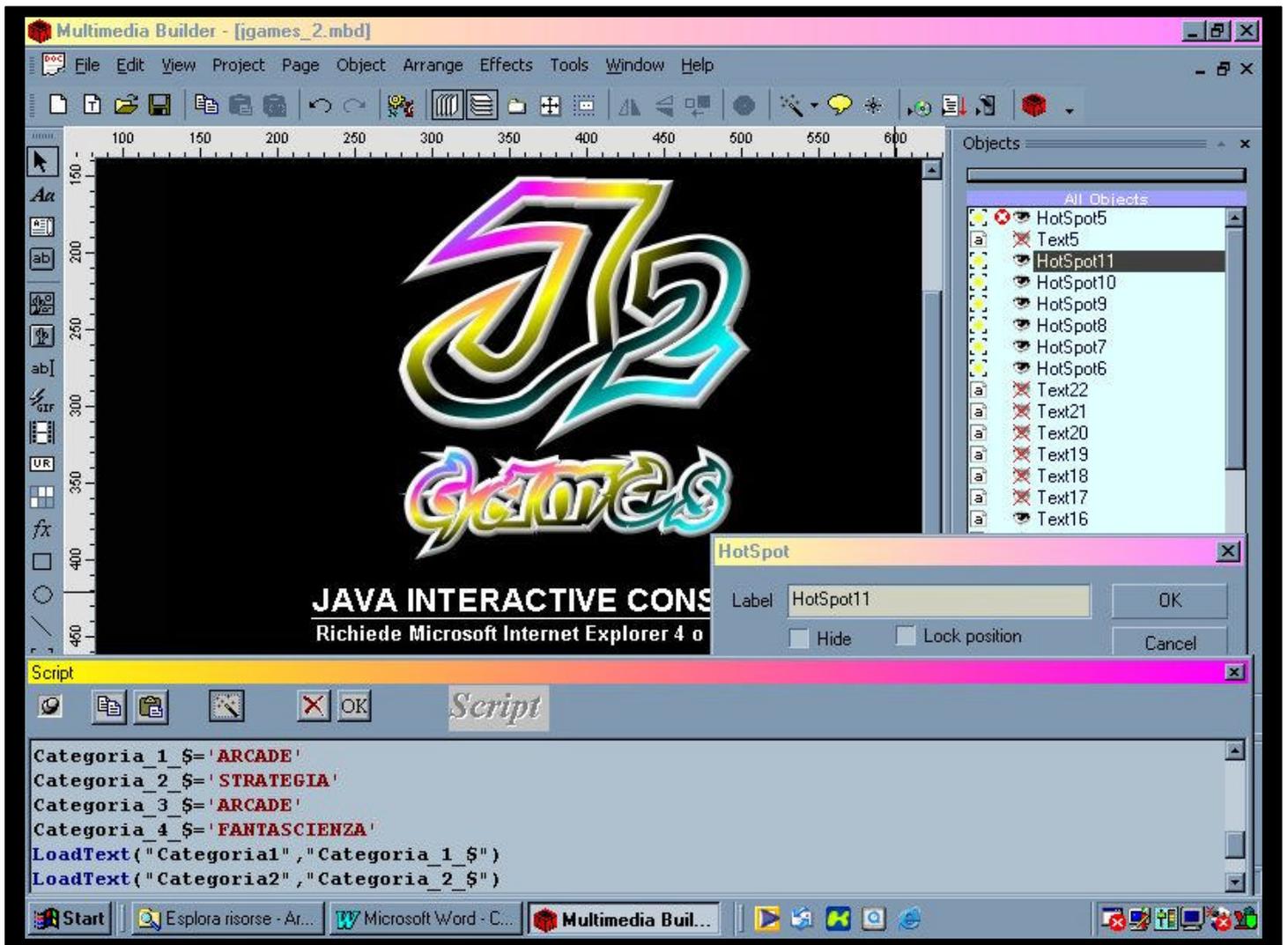


Fig. 5.10: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

## 6. CINEMUSIK III

- ❑ Versione 3.0 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2002 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti riguardanti le clip audio nel formato mp3 sono riservati agli artisti ed alle case discografiche che li detengono.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 3,22 Mb

### 6.1 REGOLE DEL GIOCO

Cinemusik III è un gioco a quiz musicale per un solo giocatore. L'obiettivo è di indovinare il maggior numero di colonne sonore di famosi film tra quelle proposte. L'archivio comprende ben 50 clip audio della durata di 10 secondi, con le colonne sonore dei film più famosi degli ultimi quarant'anni. Da "Indiana Jones" a "Lo Squalo", da "Superman" a "Guerre Stellari" e "Pretty Woman".

Per giocare una partita è sufficiente premere il tasto "NUOVO". Inizia così la riproduzione di una clip musicale e viene richiesto di selezionare a quale lungometraggio associarla. Ci sono tre possibilità: la A, B e C. Per effettuare la propria scelta basta premere il tasto corrispondente. Le impostazioni di default prevedono una durata dell'ascolto pari a 10 secondi, seguiti da altri 10 per consentire al giocatore di fare la propria scelta. In totale ci sono, quindi, 20 secondi per rispondere.

L'aspetto grafico consente al giocatore di tenere sotto controllo tutti gli aspetti della partita. Il primo elemento è costituito dalla barra di stato giallo/verde. La barra verde indica la fase d'ascolto mentre quella gialla (scandita da un tic-tac d'orologio) la fase di risposta. Per interrompere brevemente lo svolgimento della partita selezionare/deselezionare la funzione di PAUSA premendo sulla barra di stato (vedi fig. 6.1).

Ogni risposta esatta fa guadagnare 100 punti mentre una errata ne fa perdere 50. Due suoni diversi identificano l'esito della risposta. Se si risponde esattamente viene riprodotta l'intera clip audio della sigla corrispondente (circa 10 secondi) prima di passare alla domanda successiva. Sul display a destra vengono riassunte tutte le informazioni relative a livello, numero di domanda, punteggio, nonché messaggi vari (risposta corretta/errata, pausa, game over, ...). Ogni livello prevede 10 domande, per un totale di cinque livelli. Attraverso il tasto "OPZIONI" è possibile modificare la durata dell'ascolto clip (fino ad un massimo di 10 secondi). Il numero di giocatori non è, invece, modificabile. Le istruzioni per giocare vengono mostrate a video premendo il tasto "COME". Per le informazioni su autore, release, sito web, ecc. basta premere il tasto "INFO".

Nella finestra "Random Poster" sulla destra vengono mostrate le locandine cinematografiche di numerosi film, allo scopo di distrarre il giocatore. Non esiste, infatti, alcuna relazione tra locandina e colonna sonora riprodotta.

## 6.2 SUGGERIMENTI E TRUCCHI

E' consigliabile non essere precipitosi nel rispondere al quiz. Attendere la fine della fase d'ascolto e solo dopo fare la propria scelta. Nessun bonus, infatti, viene assegnato per la velocità della risposta. Uno stratagemma che si può adottare, quando si conosce la sigla ma, al momento, la memoria non fa ricordare il nome del cartone animato, consiste nell'utilizzare la funzione di pausa. Questo permette di guadagnare un pò di tempo per pensare.

## 6.3 LA REALIZZAZIONE (THE MAKING OF)

### □ LA STORIA

Cinemusik III fa parte della trilogia di quiz musicali dedicata alle colonne sonore (cartoni animati, telefilm e film). Servendosi della stessa "Game Engine" sviluppata per "Toon Music Game" ma adattata ad una nuova interfaccia grafica nasce, alla fine del 1999, la prima versione del gioco, chiamata "Cinemia 2000". La prima release prevede brani mp3 completi ed in qualità medio-alta (Stereo, 44100 Hz, 128b Kbit/sec). Per consentirne il download tramite internet e, quindi, la sua distribuzione capillare, nonché per evitare problemi di copyright, i brani vengono rimpiazzati nella versioni successive da clip di soli 10 secondi ed in bassa qualità.

Dal 2000 il videogame è stato prelevato dal sito web [www.evweb.it](http://www.evweb.it) più di 2000 volte.

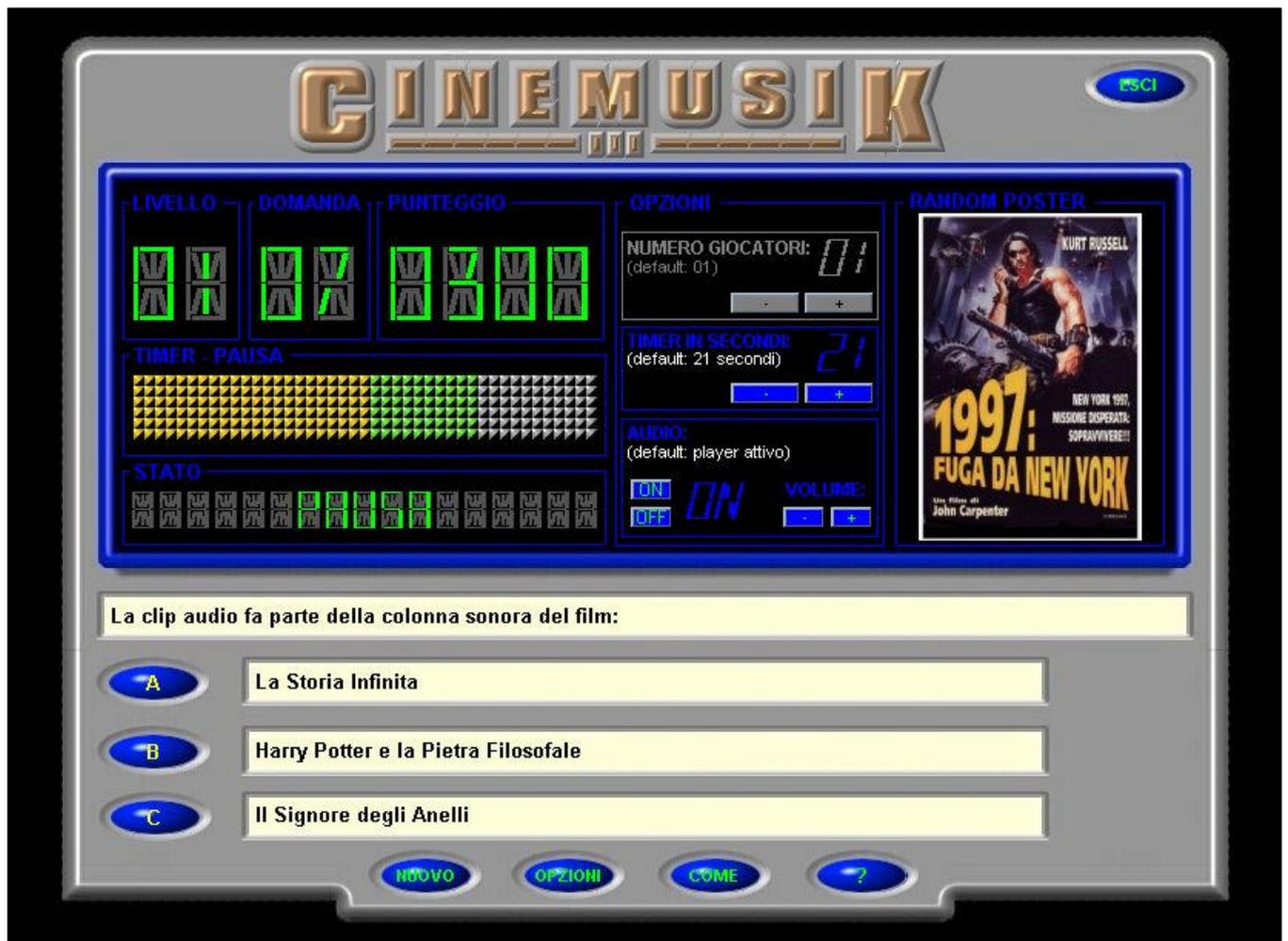


Fig. 6.1: interfaccia principale del videogame.

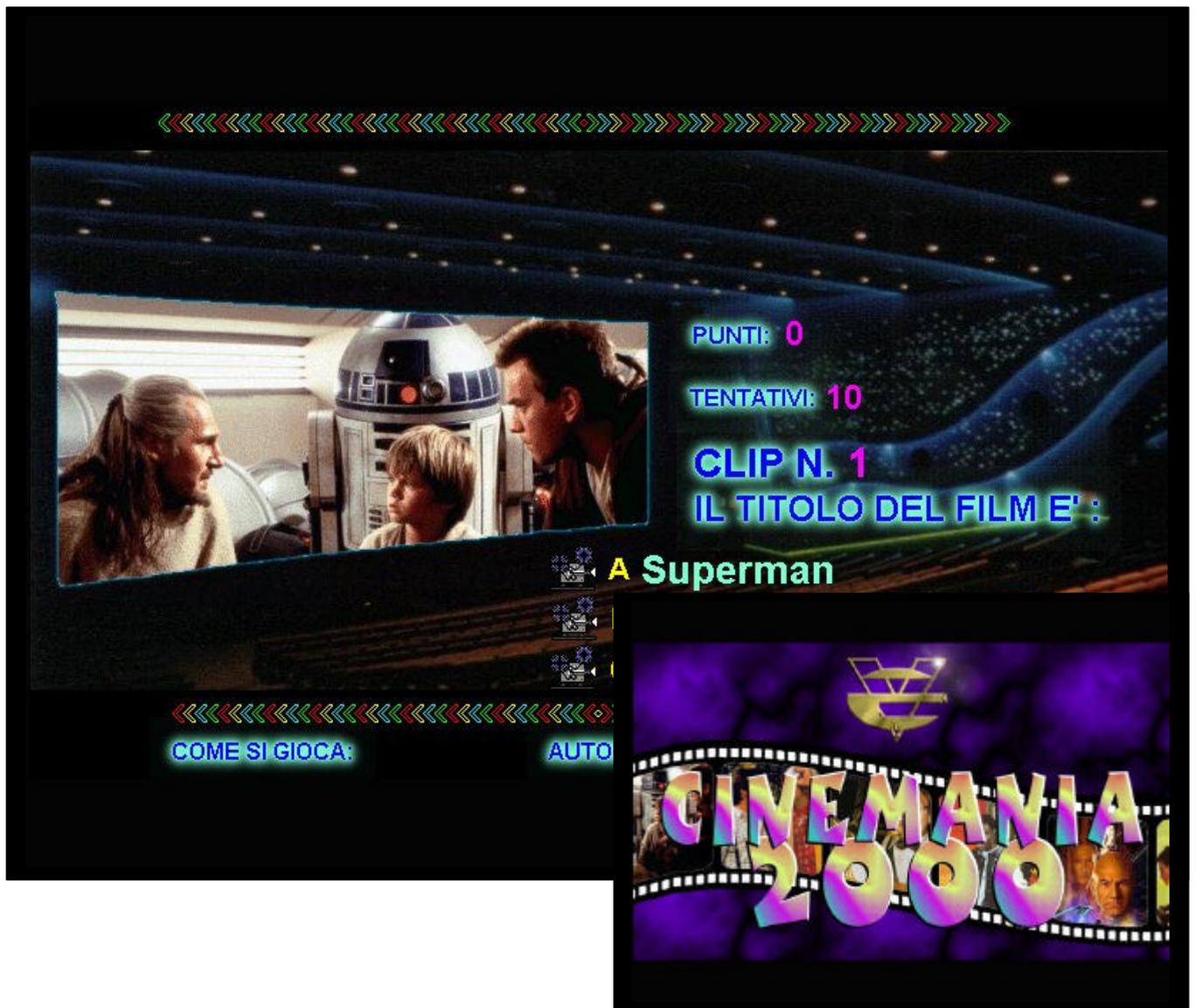
□ **I SOFTWARE UTILIZZATI**

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine:* Mediachance Multimedia Builder (vedi fig. 6.8).
- *Grafica e Immagini:* Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica:* Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

□ **BOZZE E SCREENSHOT**

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 6.2 un'immagine della prima versione del videogame. In fig. 6.3 l'interfaccia principale del videogioco (in primo piano la finestra dei parametri e delle opzioni di gioco). In fig. 6.4 come appaiono le informazioni su autore e release del gioco. In fig. 6.5 c'è l'Help On Line con le istruzioni per giocare. In fig. 6.6: Schermata di game over in caso di vittoria. In fig. 6.7 schermata di game over in caso di sconfitta. Infine, in fig. 6.8 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



**Fig. 6.2:** un'immagine della prima versione del videogame, realizzata alla fine del 1999, caratterizzata da una grafica che riproduce una sala cinematografica e da immagini sullo schermo tratte da famosi film.

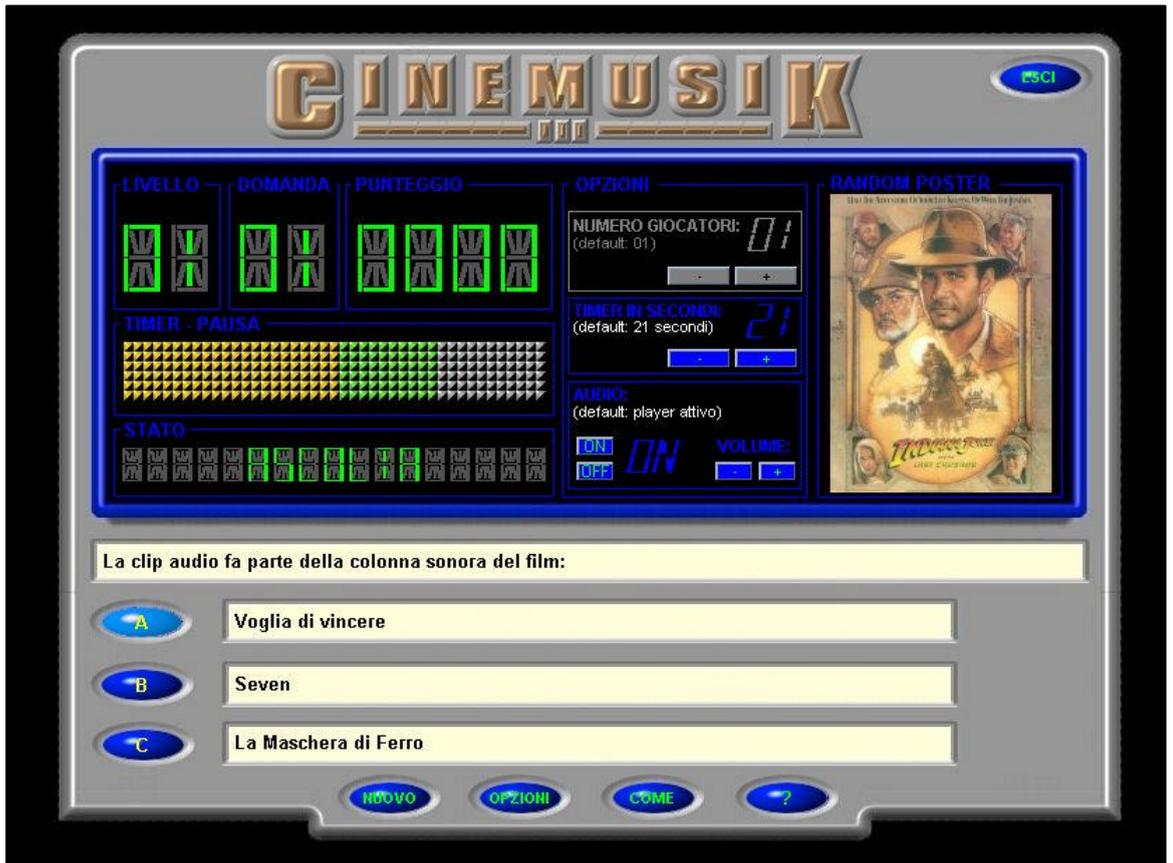


Fig. 6.3: opzioni e parametri di gioco.



Fig. 6.4: informazioni su autore e release del gioco.

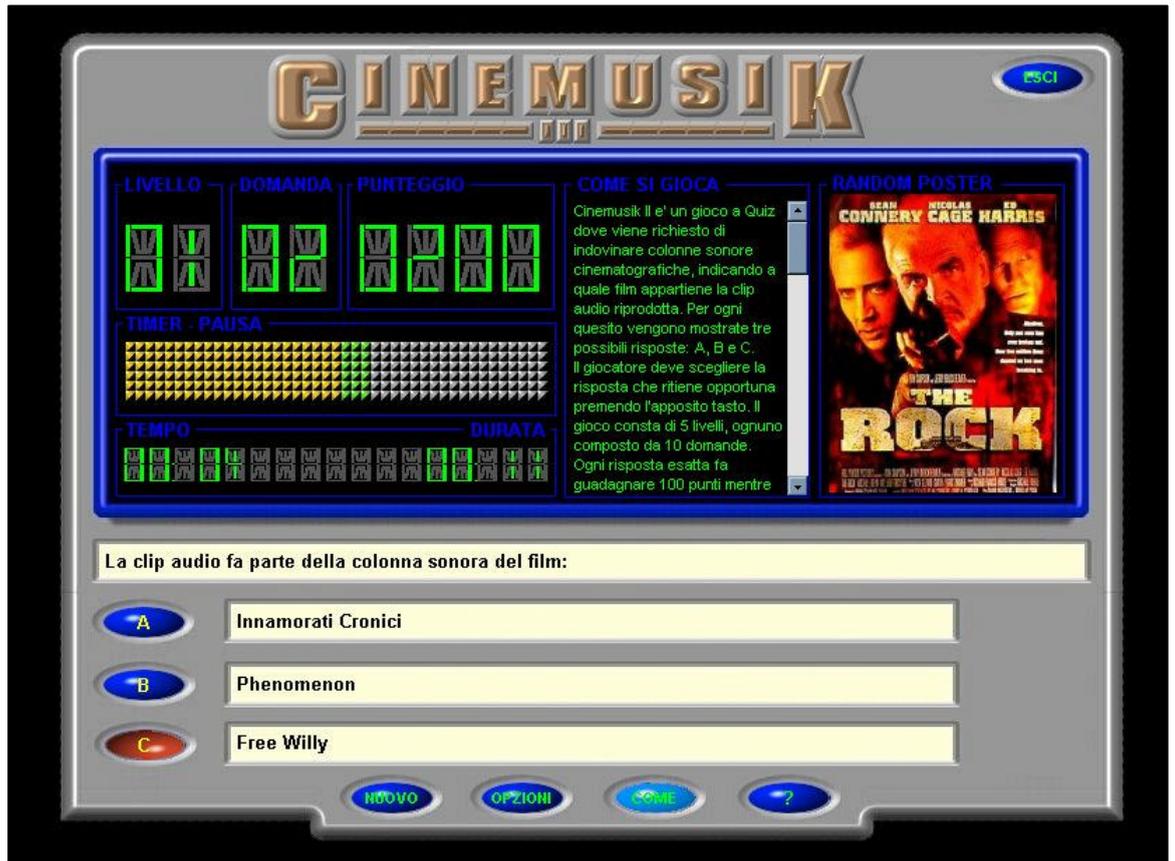


Fig. 6.5: Help On Line - Istruzioni per giocare.



Fig. 6.6: schermata di game over in caso di vittoria.



Fig. 6.7: schermata di game over in caso di sconfitta.

## 6.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

PageTimer("1000", "Page 3")
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
NDom=1
Click=0
riga=0
for i=1 to 22
  for j=1 to 6
    MatrixSet("Matrix[i,j]","3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet("Matrix[i,j]","2")
  next j
next i
for ElementoLista=1 to 50

```

```

    LoadText ("NumeDoman$", "<List>ElementoLista")
    NumeroDomanda[ElementoLista]=VAL(NumeDoman$)
next ElementoLista
NumeroImg = NumeroDomanda[NumeroDomandeTotali]
NumeroImmagine$ = CHAR(NumeroImg)
FileImmagine$ = '<SrcDir>\Img\img' + NumeroImmagine$ + '.jpg'
ReplaceImage ("Locandina", "FileImmagine$")
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText ("Text8", "Testo_NLivello$")
LoadText ("Text9", "Testo_NDomanda$")
MP3ListLoad("<Embedded>\ToonA.txt")
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonB.txt")
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonC.txt")
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonE.txt")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
Show("R_ascolta")
PlaySound("File$")
RunScript("Timer")

```

#### ❑ Script TimerPiu

```

tempo=tempo+24
display=display+1
DisplayValue("Text39", "display")

```

#### ❑ Script TimerMeno

```

tempo=tempo-24
display=display-1
if (tempo<100) then
    tempo=5
end
if (display<6) then
    display=6
end
DisplayValue("Text39", "display")

```

#### ❑ Blu Control Button

```

pausa=0
Hide("Animations")
Hide("Pic_Box")
Hide("R_pausa")
Show("PIC_Animation")
Hide("AniGif")
Show("AniGif_Rack")
Risposta$='C'
RunScript("Script")
Click=1
Show("TC")
Show("BloccaTasti")
Hide("R_ascolta")

```

#### ❑ Tasto Pausa

```

if (pausa=0) then
    pausa=1
    Show("R_pausa")
    Show("HotSpot9")
    Hide("R_ascolta")

```

```
MP3Pause()
End
```

#### □ Script Timer

```
if (pausa=0) then
  riga=riga+1
  for k=1 to 6
    MatrixSet("Matrix[44-riga,k]","1")
  next k
  WavePlay("tone","")
  if (riga<>43) then
    ScriptTimer("Timer","tempo")
  end
  if (riga=43 & Click=0) then
    RunScript("Script")
  end
  if (riga=20) then
    MP3Stop()
    Hide("R_ascolta")
  end
end
end
```

#### □ Script Fine1

```
Show("Congratulazioni")
WaveStop()
WavePlay("Bravo","")
```

#### □ Script Fine2

```
WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver")
Click=1
```

#### □ Script Fine3

```
WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver2")
Click=1
```

#### □ Script Mp3Control

```
Show("Mp3Data")
Hide("R_corretta")
pausa=0
Hide("R_pausa")
Hide("HotSpot9")
Show("HotSpot8")
pausa=0
ScriptTimer("NuovaDomanda","20000")
```

#### □ Script NuovaDomanda

```
MP3Stop()
if (NDom=10) then
  NDom=0
  DomandeGiuste=0
  NumeroLivello=NumeroLivello+1
end
riga=0
Click=0
Risposta$=''
Hide("Mp3Data")
```

```

Hide("R_errata")
Hide("R_corretta")
Hide("R_temposcaduto")
Hide("Evide")
Hide("R_pausa")
Show("HotSpot14")
for i=1 to 22
  for j=1 to 6
    MatrixSet("Matrix[i,j]","3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet("Matrix[i,j]","2")
  next j
next i
NumeroDomandeTotali=NumeroDomandeTotali+1
NDom=NDom+1
if (NumeroLivello=1) then
  Testo_NLivello$='01'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=2) then
  Testo_NLivello$='02'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=3) then
  Testo_NLivello$='03'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=4) then
  Testo_NLivello$='04'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=5) then
  Testo_NLivello$='05'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=6) then
  Testo_NLivello$='06'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=7) then
  Testo_NLivello$='07'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=8) then
  Testo_NLivello$='08'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello=9) then
  Testo_NLivello$='09'
  LoadText("Text8","Testo_NLivello$")
end
if (NumeroLivello>=10) then
  Testo1$=CHAR(NumeroLivello)
  LoadText("Text8","Testo1$")
end
if (NDom=1) then
  Testo_NDomanda$='01'
  LoadText("Text9","Testo_NDomanda$")
end
if (NDom=2) then
  Testo_NDomanda$='02'
  LoadText("Text9","Testo_NDomanda$")

```

```

end
if (NDom=3) then
    Testo_NDomanda$='03'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=4) then
    Testo_NDomanda$='04'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=5) then
    Testo_NDomanda$='05'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=6) then
    Testo_NDomanda$='06'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=7) then
    Testo_NDomanda$='07'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=8) then
    Testo_NDomanda$='08'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=9) then
    Testo_NDomanda$='09'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom>=10) then
    Testo_NDomanda$=CHAR(NDom)
    LoadText ("Text9", "Testo_NDomanda$")
end
Hide("BloccaTasti")
NumeroImg = NumeroDomanda[NumeroDomandeTotali]
NumeroImmagine$ = CHAR(NumeroImg)
FileImmagine$ = '<SrcDir>\Img\img' + NumeroImmagine$ + '.jpg'
ReplaceImage("Locandina", "FileImmagine$")
MP3ListLoad("<Embedded>\ToonA.txt")
LoadText("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonB.txt")
LoadText("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonC.txt")
LoadText("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonE.txt")
LoadText("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
PlaySound("File$")
RunScript("Timer")
Show("R_ascolta")

```

#### □ Script Script

```

MP3Stop()
Hide("Info")
Hide("Come")
Hide("GameOver")
Hide("GameOver2")
Hide("Congratulazioni")
if (Click=0 & Risposta$='') then
    Show("R_temposcaduto")
    WaveStop()
    WavePlay("Errore", "")

```

```

    punteggio=punteggio-5
end
if (Risposta$=RispostaEsatta$) then
    Show("R_corretta")
    WaveStop()
    PlaySound("File$")
    Hide("HotSpot14")
    punteggio=punteggio+100
    DomandeGiuste=DomandeGiuste+1
end
if (Risposta$<>RispostaEsatta$ & Risposta$<>'') then
    Show("R_errata")
    WaveStop()
    WavePlay("Errore","")
    punteggio=punteggio-50
end
if (punteggio<=0) then
    punteggio=0
    Testo_Punteggio$='0000'
end
if (punteggio>=0 & punteggio<10) then
    Testo_Punteggio$='000'+CHAR(punteggio)
end
if (punteggio>=10 & punteggio<100) then
    Testo_Punteggio$='00'+CHAR(punteggio)
end
if (punteggio>=100 & punteggio<1000) then
    Testo_Punteggio$='0'+CHAR(punteggio)
end
if (punteggio>=1000) then
    Testo_Punteggio$=CHAR(punteggio)
end
LoadText("Text10","Testo_Punteggio$")
if (punteggio<5000 & Risposta$=RispostaEsatta$) then
    if (NDom<>10 | DomandeGiuste>=6) then
        ScriptTimer("Mp3Control","2000")
    end
end
if (punteggio<5000 & Risposta$<>RispostaEsatta$) then
    if (NDom<>10 | DomandeGiuste>=6) then
        ScriptTimer("NuovaDomanda","2000")
    end
end
if (punteggio=5000 & NumeroLivello=5) then
    MP3Stop()
    Hide("R_corretta")
    ScriptTimer("Finel","2000")
end
if (NDom=10 & DomandeGiuste<6 & NumeroLivello<>5) then
    MP3Stop()
    Hide("R_corretta")
    Hide("R_errata")
    Hide("R_temposcaduto")
    ScriptTimer("Fine2","2000")
end
if (NDom=10 & DomandeGiuste<6 & NumeroLivello=5) then
    ScriptTimer("Fine3","2000")
    Hide("R_corretta")
    Hide("R_errata")
    Hide("R_temposcaduto")
    MP3Stop()
end
End

```

## □ Tasto Nuovo

```

if (pausa=1) then
    pausa=0
    Hide("R_pausa")
    Hide("HotSpot9")
    Show("HotSpot8")
end
Hide("GameOver")
Hide("GameOver2")
Hide("Congratulazioni")
Hide("Info")
Hide("Come")
Hide("BloccaTasti")
Hide("Evide")
Hide("Opzioni")
NumeroLivello=1
punteggio=0
DomandeGiuste=0
Testo_Punteggio$='0000'
LoadText("Text10","Testo_Punteggio$")
NDom=1
Click=0
Risposta$=''
Hide("R_errata")
Hide("R_corretta")
Hide("R_temposcaduto")
riga=0
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
for i=1 to 22
    for j=1 to 6
        MatrixSet("Matrix[i,j]","3")
    next j
next i
for i=23 to 44
    for j=1 to 6
        MatrixSet("Matrix[i,j]","2")
    next j
next i
for ElementoLista=1 to 50
    LoadText("NumeDoman$","<List>ElementoLista")
    NumeroDomanda[ElementoLista]=VAL(NumeDoman$)
next ElementoLista
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText("Text8","Testo_NLivello$")
LoadText("Text9","Testo_NDomanda$")
MP3ListLoad("<Embedded>\ToonA.txt")
LoadText("RispostaA","<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonB.txt")
LoadText("RispostaB","<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonC.txt")
LoadText("RispostaC","<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonE.txt")
LoadText("RispostaEsatta$","<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'

```

```

PlaySound("File$")
RunScript("Timer")
Show("R_ascolta")
    
```

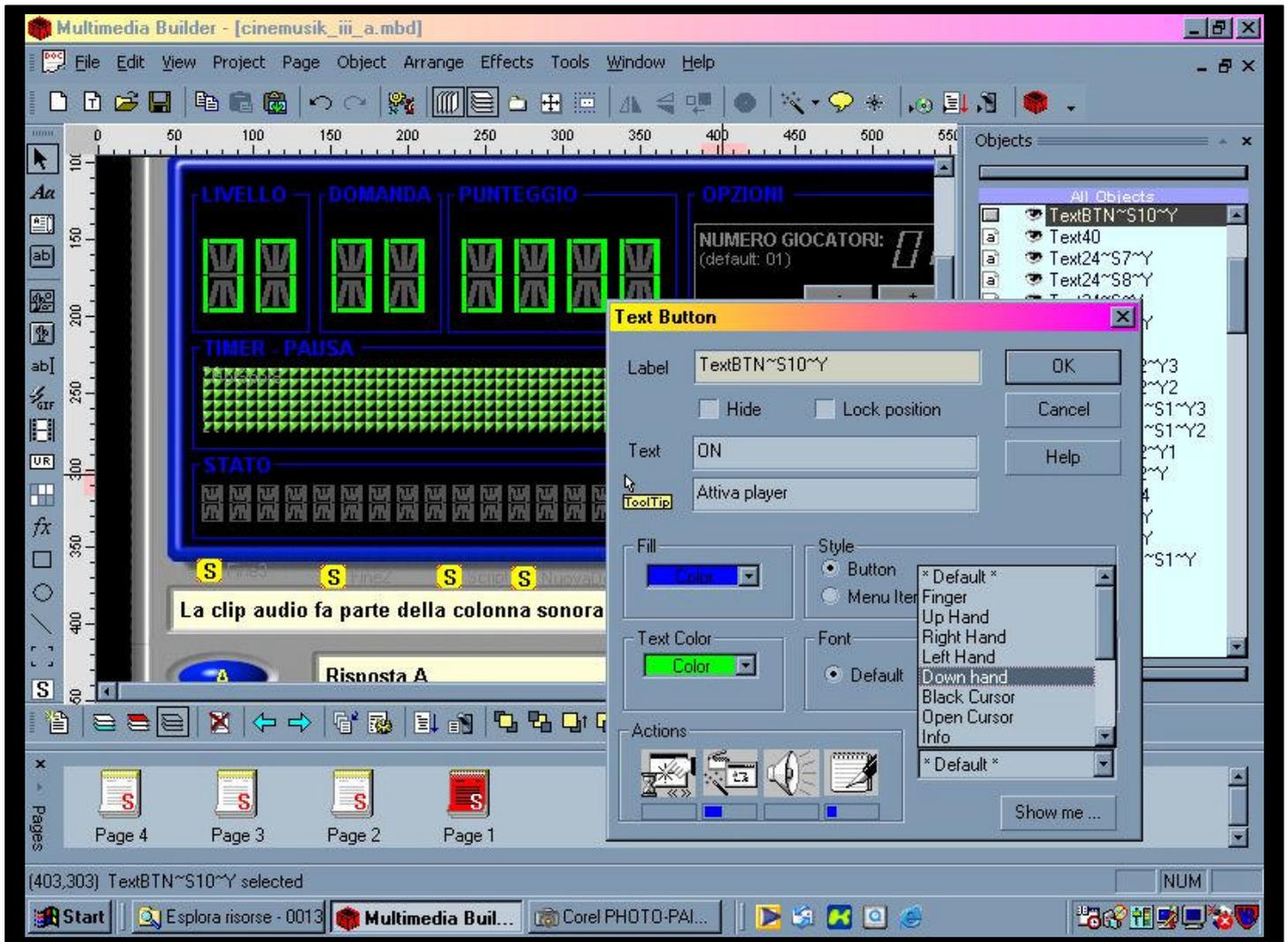


Fig. 6.8: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

## 7. CACCIA ALL'ANELLO DI SAURON

- ❑ Versione 1.0 - Freeware.
- ❑ EVSoftware - Copyright © 2002 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sul romanzo "Il Signore Degli Anelli" sono detenuti da Tolkien Company.  
Le immagini tratte dal film "La Compagnia dell'Anello" sono di proprietà esclusiva di New Line Cinema.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 2,09 Mb

### 7.1 REGOLE DEL GIOCO

Videogioco per un giocatore ispirato al film: Il Signore Degli Anelli "La compagnia dell'Anello". Con l'aiuto della memoria e un pò di rapidità la missione del giocatore è quella di trovare l'Anello Unico, forgiato da Sauron, Signore di Mordor, e distruggerlo. Per far ciò devono essere individuate le coppie di anelli magici scoprendo le caselle dei campi da gioco.

Il campo di gioco è diviso in due parti, una sul lato sinistro e una sul lato destro dello schermo. Alternativamente il giocatore deve cliccare su una delle postazioni di sinistra e poi una di quelle a destra, allo scopo di formare la coppia (come nel classico "Memory"). Se viene scoperta una coppia di anelli uguali si guadagnano punti. Il gioco prosegue fino a che viene trovata la coppia di immagini raffiguranti l'Anello Unico. Ad ogni errore tutte le caselle del tabellone vengono coperte e bisogna ricominciare da capo. Al centro dello schermo un'ampolla con un timer che scandisce il tempo disponibile al giocatore per fare le sue mosse. Il liquido dell'ampolla diminuisce di livello man mano passa il tempo, come una sorta di clessidra.

I due "Message Display" in basso segnalano uno le varie fasi di gioco, mentre l'altro il punteggio e le mosse effettuate. Per giocare basta premere sul tasto "NUOVO". La finestra con le istruzioni di gioco e informazioni su autore e release si apre premendo il tasto "HELP", mentre il tasto "ESC" consente di chiudere il programma e tornare a Windows.

### 7.2 SUGGERIMENTI E TRUCCHI

Un solo suggerimento può tornare utile per giocare a "Caccia all'Anello di Sauron": allenare la memoria e sforzarsi di ricordare sempre anche le posizioni degli anelli scoperti nelle varie fasi del gioco. Ad ogni errore, infatti, tutte le caselle vengono ricoperte, costringendo il giocatore a ricominciare da capo. E' veramente difficile riuscire a ricordare le varie immagini. Tutti gli anelli sono molto simili fra loro. Bisogna aguzzare la vista!

### 7.3 LA REALIZZAZIONE (THE MAKING OF)

#### ❑ LA STORIA

Dopo il successo mondiale del film "Il Signore degli Anelli – La Compagnia dell'Anello", del regista Peter Jackson, nel Gennaio 2002 si è pensato di sfruttarne la popolarità per creare un

nuovo videogame che facesse da catalizzatore d'utenti per il sito web [www.evweb.it](http://www.evweb.it). L'idea di base è stata quella di sfruttare un gioco classico come "Memory" e di adattarlo al nuovo tema, complicando un po' le regole.

Il risultato non è però stato all'altezza delle aspettative, a causa della grafica non particolarmente definita e di regole di gioco troppo complesse, tali da non divertire il giocatore. I download del videogame dal sito internet sono comunque stati numerosi, forse giustificati solo dal richiamo "altisonante" del titolo. Allo stato attuale non sono previste nuove versioni.



Fig. 7.1: Interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachanche Multimedia Builder (vedi fig. 7.5).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco.

Nelle figg. 7.2 e 7.3 immagini tratte dell'interfaccia principale del videogame.

In fig. 7.4 la finestra con le informazioni su autore, release, sito internet e istruzioni del gioco. Infine, in fig. 7.5 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

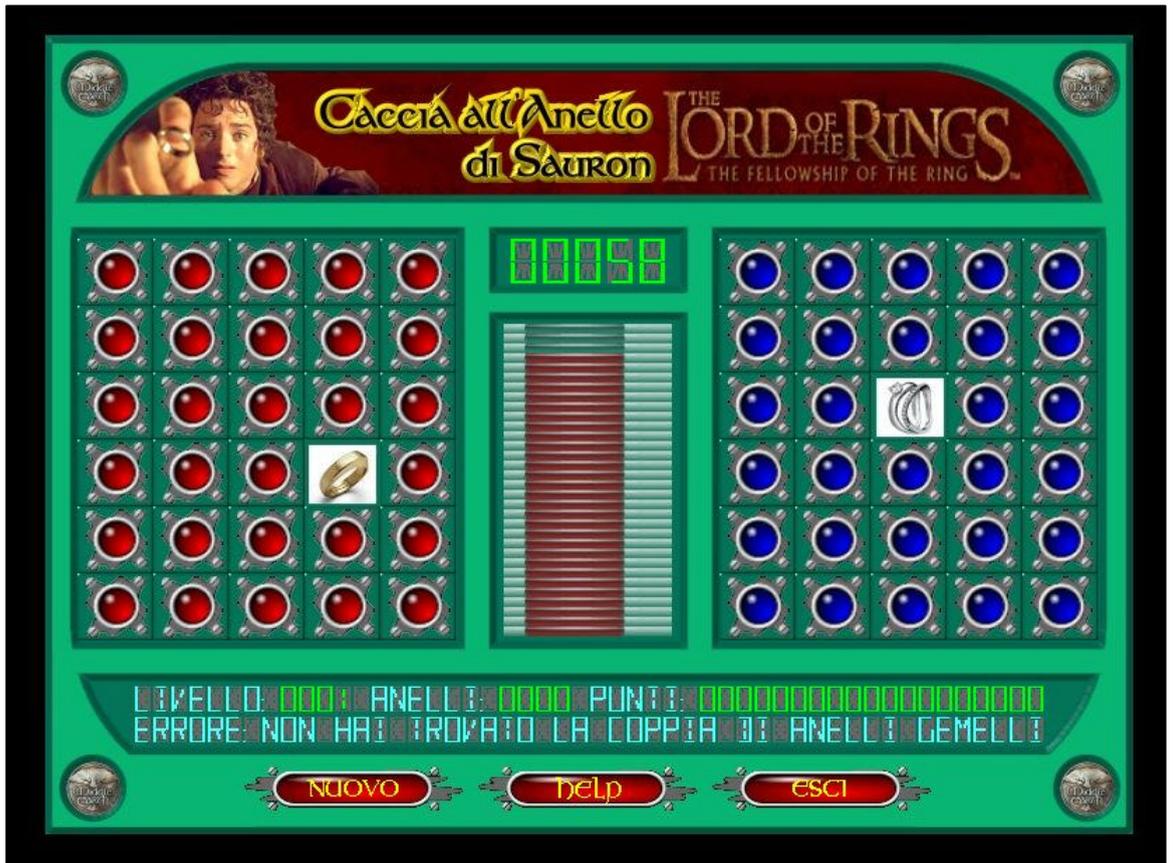


Fig. 7.3: immagine della schermata principale del videogame.



Fig. 7.4: informazioni su autore, release, sito internet, con le istruzioni su come si gioca.

## 7.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
InstallFont("<Embedded>\Keltn____.ttf")
WavePlay("thunder", "")
```

### □ Script Tasto NUOVO

```
WavePlay("Tone1", "")
NumeroLivello=1
Punteggio=0
Tempo=2000
SecondiBonus = 0
Click=0
riga=0
Show("TabelloneA")
Show("TabelloneB")
Hide("MascheraInizioA")
Hide("MascheraInizioB")
Hide("MascheraA")
Hide("Help")
Hide("Messaggi")
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
for ElementoLista=1 to 30
  LoadText("NumeRing$", "<List>ElementoLista")
  NumeroAnelloA[ElementoLista]=VAL(NumeRing$)
  NumeImmagine$ = 'ImgA' + CHAR(ElementoLista)
  TipoImmagine$ = '<Embedded>Ring' + NumeRing$ + '.jpg'
  ReplaceImage("NumeImmagine$", "TipoImmagine$")
next ElementoLista
MP3ListRND()
for ElementoLista=1 to 30
  LoadText("NumeRing$", "<List>ElementoLista")
  NumeroAnelloB[ElementoLista]=VAL(NumeRing$)
  NumeImmagine$ = 'ImgB' + CHAR(ElementoLista)
  TipoImmagine$ = '<Embedded>Ring' + NumeRing$ + '.jpg'
  ReplaceImage("NumeImmagine$", "TipoImmagine$")
next ElementoLista
Livello = 1
if (Livello<10 & Livello>0) then
  Livello$ = '000' + CHAR(Livello)
end
LoadText("Livello", "Livello$")
Anelli = 0
if (Anelli=0) then
  Anelli$ = '0000'
end
if (Anelli<10 & Anelli>0) then
  Anelli$ = '000' + CHAR(Anelli)
end
LoadText("Anelli", "Anelli$")
Punti = 0
if (Punti<10 & Punti>=0) then
  Punti$ = '000000000000000000' + CHAR(Punti)
end
LoadText("Punti", "Punti$")
TastoPremutoA = 0
TastoPremutoB = 0
```

```
ScriptTimer("Timer", "Tempo")
MatrixSet("Matrix1[0,0]", "1")
MatrixSet("Matrix2[0,0]", "1")
MatrixSet("Matrix3[0,0]", "1")
MatrixSet("Matrix4[0,0]", "1")
MatrixSet("Matrix5[0,0]", "1")
MatrixSet("Matrix6[0,0]", "1")
MatrixSet("Matrix7[0,0]", "1")
MatrixSet("Matrix8[0,0]", "1")
MatrixSet("Matrix9[0,0]", "1")
MatrixSet("Matrix10[0,0]", "1")
MatrixSet("Matrix11[0,0]", "1")
MatrixSet("Matrix12[0,0]", "1")
MatrixSet("Matrix13[0,0]", "1")
MatrixSet("Matrix14[0,0]", "1")
MatrixSet("Matrix15[0,0]", "1")
MatrixSet("Matrix16[0,0]", "1")
MatrixSet("Matrix17[0,0]", "1")
MatrixSet("Matrix18[0,0]", "1")
MatrixSet("Matrix19[0,0]", "1")
MatrixSet("Matrix20[0,0]", "1")
MatrixSet("Matrix21[0,0]", "1")
MatrixSet("Matrix22[0,0]", "1")
MatrixSet("Matrix23[0,0]", "1")
MatrixSet("Matrix24[0,0]", "1")
MatrixSet("Matrix25[0,0]", "1")
MatrixSet("Matrix26[0,0]", "1")
MatrixSet("Matrix27[0,0]", "1")
MatrixSet("Matrix28[0,0]", "1")
MatrixSet("Matrix29[0,0]", "1")
MatrixSet("Matrix30[0,0]", "1")
MatrixSet("Matrix31[0,0]", "1")
MatrixSet("Matrix32[0,0]", "1")
MatrixSet("Matrix33[0,0]", "1")
MatrixSet("Matrix34[0,0]", "1")
MatrixSet("Matrix35[0,0]", "1")
MatrixSet("Matrix36[0,0]", "1")
MatrixSet("Matrix37[0,0]", "1")
MatrixSet("Matrix38[0,0]", "1")
MatrixSet("Matrix39[0,0]", "1")
MatrixSet("Matrix40[0,0]", "1")
MatrixSet("Matrix41[0,0]", "1")
MatrixSet("Matrix42[0,0]", "1")
MatrixSet("Matrix43[0,0]", "1")
MatrixSet("Matrix44[0,0]", "1")
MatrixSet("Matrix45[0,0]", "1")
MatrixSet("Matrix46[0,0]", "1")
MatrixSet("Matrix47[0,0]", "1")
MatrixSet("Matrix48[0,0]", "1")
MatrixSet("Matrix49[0,0]", "1")
MatrixSet("Matrix50[0,0]", "1")
MatrixSet("Matrix51[0,0]", "1")
MatrixSet("Matrix52[0,0]", "1")
MatrixSet("Matrix53[0,0]", "1")
MatrixSet("Matrix54[0,0]", "1")
MatrixSet("Matrix55[0,0]", "1")
MatrixSet("Matrix56[0,0]", "1")
MatrixSet("Matrix57[0,0]", "1")
MatrixSet("Matrix58[0,0]", "1")
MatrixSet("Matrix59[0,0]", "1")
MatrixSet("Matrix60[0,0]", "1")
MatrixSet("MatrixTimer[0,0]", "1")
```

❑ **Script Tasto tabellone A – MOUSE DOWN**

```
MatrixSet("Matrix1[MXCOL,MXROW]", "3")
```

❑ **Script Tasto tabellone A – MOUSE UP**

```
TastoPremutoA = 1
Hide("Matrix1")
Show("MascheraA")
Hide("MascheraB")
```

❑ **Script Tasto tabellone B – MOUSE DOWN**

```
MatrixSet("Matrix1[MXCOL,MXROW]", "3")
```

❑ **Script Tasto tabellone B – MOUSE UP**

```
TastoPremutoB = 27
Hide("Matrix57")
Hide("MascheraA")
Show("MascheraB")
RunScript("Game")
```

❑ **Script TIMER**

```
if (pausa = 0) then
  riga = riga + 1
  MatrixSet("MatrixTimer[1,riga]", "2")
  WavePlay("Tone1", "")
  if (riga <> 32) then
    ScriptTimer("Timer", "Tempo")
  end
  if (riga = 32 & Click = 0) then
    RunScript("ScriptFine")
  end
end
Sec = (32 - riga)*Tempo/1000
if (Sec<10 & Sec>=0) then
  Secondi$ = '0000' + CHAR(Sec)
end
if (Sec<100 & Sec>=10) then
  Secondi$ = '000' + CHAR(Sec)
end
if (Sec<1000 & Sec>=100) then
  Secondi$ = '00' + CHAR(Sec)
end
if (Sec<10000 & Sec>=1000) then
  Secondi$ = '0' + CHAR(Sec)
end
if (Sec>10000) then
  Secondi$ = CHAR(Sec)
end
LoadText("TextTimer", "Secondi$")
```

❑ **Script GAME**

```
Show("MascheraA")
Show("MascheraB")
if (NumeroAnelloA[TastoPremutoA] = NumeroAnelloB[TastoPremutoB] &
NumeroAnelloB[TastoPremutoB] <> 1) then
  Punti = Punti + 1000
  Anelli = Anelli + 1
  SecondiBonus = 10
  WavePlay("drum", "")
  ScriptTimer("RipristinaDisplay", "2000")
  Show("MsgCoppia")
end
```

```

if (NumeroAnelloA[TastoPremutoA] = NumeroAnelloB[TastoPremutoB] &
NumeroAnelloB[TastoPremutoB] = 1) then
    Punti = Punti + 10000
    Anelli = Anelli + 1
    SecondiBonus = 20
    WavePlay("bell", "")
    Show("MsgVinto")
    ScriptTimer("NuovoLivello", "2000")
    Livello = Livello + 1
end
if (NumeroAnelloA[TastoPremutoA] <> NumeroAnelloB[TastoPremutoB]) then
    Punti = Punti - 50
    WavePlay("thunder", "")
    Show("MsgErrore")
    ScriptTimer("RipristinaTabelloni", "2000")
end
Show("MascheraB")
if (Livello<10 & Livello>=0) then
    Livello$ = '000' + CHAR(Livello)
end
if (Livello<100 & Livello>=10) then
    Livello$ = '00' + CHAR(Livello)
end
if (Livello<1000 & Livello>=100) then
    Livello$ = '0' + CHAR(Livello)
end
if (Livello>=1000) then
    Livello$ = CHAR(Livello)
end
LoadText("Livello", "Livello$")
if (Anelli<10 & Anelli>=0) then
    Anelli$ = '000' + CHAR(Anelli)
end
if (Anelli<100 & Anelli>=10) then
    Anelli$ = '00' + CHAR(Anelli)
end
if (Anelli<1000 & Anelli>=100) then
    Anelli$ = '0' + CHAR(Anelli)
end
if (Anelli>=1000) then
    Anelli$ = CHAR(Anelli)
end
LoadText("Anelli", "Anelli$")
if (Punti<10 & Punti>=0) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<100 & Punti>=10) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<1000 & Punti>=100) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<10000 & Punti>=1000) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<100000 & Punti>=10000) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<1000000 & Punti>=100000) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<10000000 & Punti>=1000000) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
if (Punti<100000000 & Punti>=10000000) then

```

```

Punti$ = '0000000000' + CHAR(Punti)
end
if (Punti<1000000000 & Punti>=100000000) then
    Punti$ = '0000000000' + CHAR(Punti)
end
if (Punti<10000000000 & Punti>=1000000000) then
    Punti$ = '000000000' + CHAR(Punti)
end
if (Punti<100000000000 & Punti>=10000000000) then
    Punti$ = '00000000' + CHAR(Punti)
end
if (Punti<1000000000000 & Punti>=100000000000) then
    Punti$ = '0000000' + CHAR(Punti)
end
if (Punti<10000000000000 & Punti>=1000000000000) then
    Punti$ = '000000' + CHAR(Punti)
end
if (Punti<100000000000000 & Punti>=10000000000000) then
    Punti$ = '00000' + CHAR(Punti)
end
if (Punti<1000000000000000 & Punti>=100000000000000) then
    Punti$ = '0000' + CHAR(Punti)
end
if (Punti<10000000000000000 & Punti>=1000000000000000) then
    Punti$ = '000' + CHAR(Punti)
end
if (Punti<100000000000000000 & Punti>=10000000000000000) then
    Punti$ = CHAR(Punti)
end
LoadText ("Punti", "Punti$")
TastoPremutoA = 0
TastoPremutoB = 0

```

□ **Script RipristinaTabelloni**

```

Show("TabelloneA")
Show("TabelloneB")
Hide("MascheraA")
Hide("Messaggi")
ScriptTimer("Timer", "Tempo")
MatrixSet("Matrix1[0,0]", "1")
MatrixSet("Matrix2[0,0]", "1")
MatrixSet("Matrix3[0,0]", "1")
MatrixSet("Matrix4[0,0]", "1")
MatrixSet("Matrix5[0,0]", "1")
MatrixSet("Matrix6[0,0]", "1")
MatrixSet("Matrix7[0,0]", "1")
MatrixSet("Matrix8[0,0]", "1")
MatrixSet("Matrix9[0,0]", "1")
MatrixSet("Matrix10[0,0]", "1")
MatrixSet("Matrix11[0,0]", "1")
MatrixSet("Matrix12[0,0]", "1")
MatrixSet("Matrix13[0,0]", "1")
MatrixSet("Matrix14[0,0]", "1")
MatrixSet("Matrix15[0,0]", "1")
MatrixSet("Matrix16[0,0]", "1")
MatrixSet("Matrix17[0,0]", "1")
MatrixSet("Matrix18[0,0]", "1")
MatrixSet("Matrix19[0,0]", "1")
MatrixSet("Matrix20[0,0]", "1")
MatrixSet("Matrix21[0,0]", "1")
MatrixSet("Matrix22[0,0]", "1")
MatrixSet("Matrix23[0,0]", "1")
MatrixSet("Matrix24[0,0]", "1")
MatrixSet("Matrix25[0,0]", "1")

```

```

MatrixSet("Matrix26[0,0]", "1")
MatrixSet("Matrix27[0,0]", "1")
MatrixSet("Matrix28[0,0]", "1")
MatrixSet("Matrix29[0,0]", "1")
MatrixSet("Matrix30[0,0]", "1")
MatrixSet("Matrix31[0,0]", "1")
MatrixSet("Matrix32[0,0]", "1")
MatrixSet("Matrix33[0,0]", "1")
MatrixSet("Matrix34[0,0]", "1")
MatrixSet("Matrix35[0,0]", "1")
MatrixSet("Matrix36[0,0]", "1")
MatrixSet("Matrix37[0,0]", "1")
MatrixSet("Matrix38[0,0]", "1")
MatrixSet("Matrix39[0,0]", "1")
MatrixSet("Matrix40[0,0]", "1")
MatrixSet("Matrix41[0,0]", "1")
MatrixSet("Matrix42[0,0]", "1")
MatrixSet("Matrix43[0,0]", "1")
MatrixSet("Matrix44[0,0]", "1")
MatrixSet("Matrix45[0,0]", "1")
MatrixSet("Matrix46[0,0]", "1")
MatrixSet("Matrix47[0,0]", "1")
MatrixSet("Matrix48[0,0]", "1")
MatrixSet("Matrix49[0,0]", "1")
MatrixSet("Matrix50[0,0]", "1")
MatrixSet("Matrix51[0,0]", "1")
MatrixSet("Matrix52[0,0]", "1")
MatrixSet("Matrix53[0,0]", "1")
MatrixSet("Matrix54[0,0]", "1")
MatrixSet("Matrix55[0,0]", "1")
MatrixSet("Matrix56[0,0]", "1")
MatrixSet("Matrix57[0,0]", "1")
MatrixSet("Matrix58[0,0]", "1")
MatrixSet("Matrix59[0,0]", "1")
MatrixSet("Matrix60[0,0]", "1")

```

#### □ Script ScriptFine

```

Show("MascheraInizioB")
Show("MascheraInizioA")
Show("MascheraB")
Show("MascheraA")
Show("MsgGameOver")

```

#### □ Script RipristinaDisplay

```

Hide("Messaggi")
Hide("MascheraA")
ScriptTimer("Timer", "Tempo")
Sec = Sec + SecondiBonus
riga = riga - (SecondiBonus*1000/Tempo)
if (riga >= 32) then
  riga = 32
end
if (riga <= 0) then
  riga = 1
end
MatrixSet("MatrixTimer[0,0]", "1")
for i = 1 to riga
  MatrixSet("MatrixTimer[1,riga]", "2")
next i

```

□ **Script NuovoLivello**

```

SecondiBonus = 0
Click=0
riga=0
Show("TabelloneA")
Show("TabelloneB")
Hide("MascheraInizioA")
Hide("MascheraInizioB")
Hide("MascheraA")
Hide("Help")
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
for ElementoLista=1 to 30
    LoadText("NumeRing$", "<List>ElementoLista")
    NumeroAnelloA[ElementoLista]=VAL(NumeRing$)
    NumeImmagine$ = 'ImgA' + CHAR(ElementoLista)
    TipoImmagine$ = '<Embedded>Ring' + NumeRing$ + '.jpg'
    ReplaceImage("NumeImmagine$", "TipoImmagine$")
next ElementoLista
MP3ListRND()
for ElementoLista=1 to 30
    LoadText("NumeRing$", "<List>ElementoLista")
    NumeroAnelloB[ElementoLista]=VAL(NumeRing$)
    NumeImmagine$ = 'ImgB' + CHAR(ElementoLista)
    TipoImmagine$ = '<Embedded>Ring' + NumeRing$ + '.jpg'
    ReplaceImage("NumeImmagine$", "TipoImmagine$")
next ElementoLista
if (Livello<10 & Livello>0) then
    Livello$ = '000' + CHAR(Livello)
end
LoadText("Livello", "Livello$")
if (Anelli=0) then
    Anelli$ = '0000'
end
if (Anelli<10 & Anelli>0) then
    Anelli$ = '000' + CHAR(Anelli)
end
LoadText("Anelli", "Anelli$")
if (Punti<10 & Punti>=0) then
    Punti$ = '000000000000000000' + CHAR(Punti)
end
LoadText("Punti", "Punti$")
TastoPremutoA = 0
TastoPremutoB = 0
ScriptTimer("Timer", "Tempo")
MatrixSet("Matrix1[0,0]", "1")
MatrixSet("Matrix2[0,0]", "1")
MatrixSet("Matrix3[0,0]", "1")
MatrixSet("Matrix4[0,0]", "1")
MatrixSet("Matrix5[0,0]", "1")
MatrixSet("Matrix6[0,0]", "1")
MatrixSet("Matrix7[0,0]", "1")
MatrixSet("Matrix8[0,0]", "1")
MatrixSet("Matrix9[0,0]", "1")
MatrixSet("Matrix10[0,0]", "1")
MatrixSet("Matrix11[0,0]", "1")
MatrixSet("Matrix12[0,0]", "1")
MatrixSet("Matrix13[0,0]", "1")
MatrixSet("Matrix14[0,0]", "1")
MatrixSet("Matrix15[0,0]", "1")
MatrixSet("Matrix16[0,0]", "1")
MatrixSet("Matrix17[0,0]", "1")
MatrixSet("Matrix18[0,0]", "1")

```

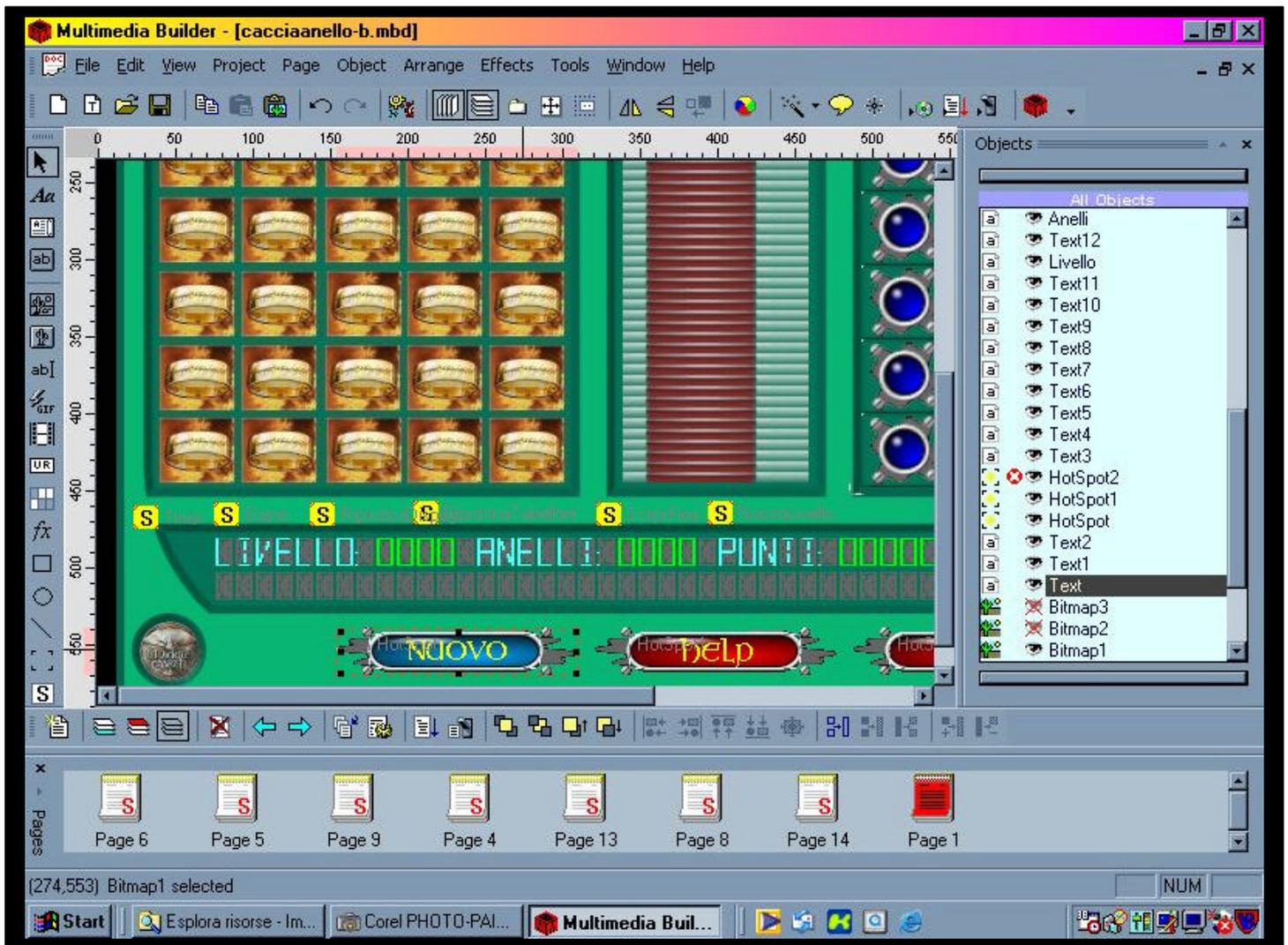


Fig. 7.5: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

```

MatrixSet ("Matrix19[0,0]", "1")
MatrixSet ("Matrix20[0,0]", "1")
MatrixSet ("Matrix21[0,0]", "1")
MatrixSet ("Matrix22[0,0]", "1")
MatrixSet ("Matrix23[0,0]", "1")
MatrixSet ("Matrix24[0,0]", "1")
MatrixSet ("Matrix25[0,0]", "1")
MatrixSet ("Matrix26[0,0]", "1")
MatrixSet ("Matrix27[0,0]", "1")
MatrixSet ("Matrix28[0,0]", "1")
MatrixSet ("Matrix29[0,0]", "1")
MatrixSet ("Matrix30[0,0]", "1")
MatrixSet ("Matrix31[0,0]", "1")
MatrixSet ("Matrix32[0,0]", "1")
MatrixSet ("Matrix33[0,0]", "1")
MatrixSet ("Matrix34[0,0]", "1")
MatrixSet ("Matrix35[0,0]", "1")
MatrixSet ("Matrix36[0,0]", "1")
MatrixSet ("Matrix37[0,0]", "1")
MatrixSet ("Matrix38[0,0]", "1")
MatrixSet ("Matrix39[0,0]", "1")
MatrixSet ("Matrix40[0,0]", "1")
MatrixSet ("Matrix41[0,0]", "1")
MatrixSet ("Matrix42[0,0]", "1")
MatrixSet ("Matrix43[0,0]", "1")

```

```
MatrixSet("Matrix44[0,0]", "1")
MatrixSet("Matrix45[0,0]", "1")
MatrixSet("Matrix46[0,0]", "1")
MatrixSet("Matrix47[0,0]", "1")
MatrixSet("Matrix48[0,0]", "1")
MatrixSet("Matrix49[0,0]", "1")
MatrixSet("Matrix50[0,0]", "1")
MatrixSet("Matrix51[0,0]", "1")
MatrixSet("Matrix52[0,0]", "1")
MatrixSet("Matrix53[0,0]", "1")
MatrixSet("Matrix54[0,0]", "1")
MatrixSet("Matrix55[0,0]", "1")
MatrixSet("Matrix56[0,0]", "1")
MatrixSet("Matrix57[0,0]", "1")
MatrixSet("Matrix58[0,0]", "1")
MatrixSet("Matrix59[0,0]", "1")
MatrixSet("Matrix60[0,0]", "1")
MatrixSet("MatrixTimer[0,0]", "1")
```

## 8. SUPERSECCHIONE

- ❑ Versione 4.2 - Shareware.
- ❑ EVSoftware - Copyright © 1995-2001 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,02 Mb

### 8.1 REGOLE DEL GIOCO

Super Secchione è un gioco a quiz per un solo giocatore. L'obiettivo è di rispondere correttamente al maggior numero di domande tra quelle proposte. L'archivio comprende ben 500 quesiti, dalla chimica alla matematica, dal cinema alla musica, dalla storia alla geografia. Per giocare una partita è sufficiente premere il tasto "NUOVO". Viene mostrata sul display la domanda alla quale bisogna rispondere. Ci sono tre possibilità: A, B o C. Per effettuare la propria scelta basta premere il tasto corrispondente.

L'aspetto grafico consente al giocatore di tenere sotto controllo tutti gli aspetti della partita. Il primo elemento è costituito dalla barra cronometro gialla. Tale barra è composta da una serie di elementi, inizialmente accesi, che si spengono in successione e segnalano, con un tic-tac d'orologio, il tempo a disposizione per rispondere. Ogni risposta esatta fa guadagnare 10 punti mentre una risposta errata ne fa perdere cinque. Due suoni diversi identificano l'esito della risposta. Un timer scandisce il tempo massimo disponibile per effettuare la propria scelta. Se viene lasciato scadere il tempo senza aver risposto si perdono altri cinque punti. Per interrompere temporaneamente la partita basta premere sulla barra gialla del Timer. Per riprendere il gioco è sufficiente premere nuovamente tale barra.

Sul display a destra vengono riassunte tutte le informazioni relative a livello, numero di domanda, punteggio, nonché messaggi vari (risposta corretta/errata, pausa, game over, ...). Ogni livello prevede 20 domande, per un totale di 25 Livelli. Attraverso il tasto "OPZIONI" è possibile modificare la durata del timer. Il numero di giocatori non è, invece, modificabile. Le istruzioni per giocare vengono mostrate a video premendo il tasto "COME". Per le informazioni su autore, release, sito web, ecc. basta premere il tasto "INFO".

### 8.2 SUGGERIMENTI E TRUCCHI

Considerando che nessun bonus viene assegnato per la rapidità della risposta, conviene sempre leggere attentamente il testo della domanda e sfruttare strategicamente il tempo a disposizione per rispondere. Una buona tattica prevede di non badare al tic-tac dell'orologio ma di pensare alla risposta tenendo sotto controllo l'inevitabile spegnimento della barra cronometro.

### 8.3 LA REALIZZAZIONE (THE MAKING OF)

#### □ LA STORIA

Super Secchione è il primo videogame della serie EVGames. E' il capostipite. La sua nascita risale al 1995 quando, come esercizio di programmazione in TurboPascal 7, viene implementato un rudimentale videogioco a quiz per sistema operativo Microsoft DOS. La grafica è piuttosto geometrica e realizzata con semplici effetti cromatici e caratteri ASCII ed il mouse non è abilitato. Si gioca digitando sulla tastiera, comprese le risposte A, B o C, che vengono date premendo il tasto del carattere corrispondente. Dalla prima rudimentale versione, denominata "GENIUS" (alcuni screenshot nelle figg. 8.2, 8.11 e 8.12), caratterizzata anche da un archivio di domande piuttosto povero, si passa a quelle successive nel 1996 e 1997, "Super Quiz II" e "Cleverdick". In queste ultime versioni viene di volta in volta incrementato il database delle domande e migliorata la grafica (con animazioni e suoni), nonché la giocabilità, con l'introduzione di diversi livelli di difficoltà.



**Fig. 8.2:** Screenshot della prima versione del gioco, GENIUS, datata 1995.

Il progetto non subisce evoluzioni fino al 2001 quando, dopo l'esperienza positiva delle "Game Engine" sviluppate nel 1999 per "Las Vegas" e "Toon Music Game", si decide di riprenderlo per i sistemi operativi Windows. Dopo aver recuperato i vecchi file di testo contenenti l'archivio di 500 quesiti e definito un prototipo di algoritmo, viene ridisegnata l'interfaccia grafica, semplice ma particolarmente "user friendly" ed inizia il duro lavoro di restyling. Dopo alcune settimane il risultato è un videogioco completamente nuovo, che non assomiglia minimamente, nè graficamente nè per funzionamento, alle vecchie versioni. Unico elemento di continuità: il database delle domande. Il nome doveva essere speciale e

richiamare l'attenzione, così come il logo. Anziché scegliere un simbolo grafico, quindi, si è preferito un personaggio cartoon, Armando, il secchione per eccellenza e protagonista della serie a fumetti "Ingegneria, che passione!", pubblicata dalla rivista universitaria "La Scatola Nera". Il nome del gioco si è imposto di conseguenza: "Super Secchione", come a sottolineare l'estrema difficoltà degli argomenti trattati e delle domande proposte.

"Super Secchione 2001", prelevabile gratuitamente dal sito internet [www.evweb.it](http://www.evweb.it) e dal portale di software freeware della Mondadori, è stato prelevato più di 1500 volte.



Fig. 8.3: Interfaccia principale del videogioco.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachanche Multimedia Builder (vedi fig. 4.10).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. Nelle figg. 8.2, 8.11 e 8.12 ci sono degli screenshot della prima versione del gioco, chiamata "GENIUS", datata 1995. Nelle figg. 8.3 e 8.4 la schermata principale di Super Secchione. In fig. 8.5 Informazioni su autore e release del gioco. In fig. 8.6 c'è l'Help On Line con le istruzioni per giocare. In fig. 8.7: Schermata di game over in caso di vittoria. In fig. 8.8: Schermata di game over in caso di sconfitta. In fig. 8.9: Finestra con le opzioni ed i parametri di gioco. Infine, in fig. 8.10 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



Fig. 8.4: Interfaccia principale del videogioco – funzione pausa.

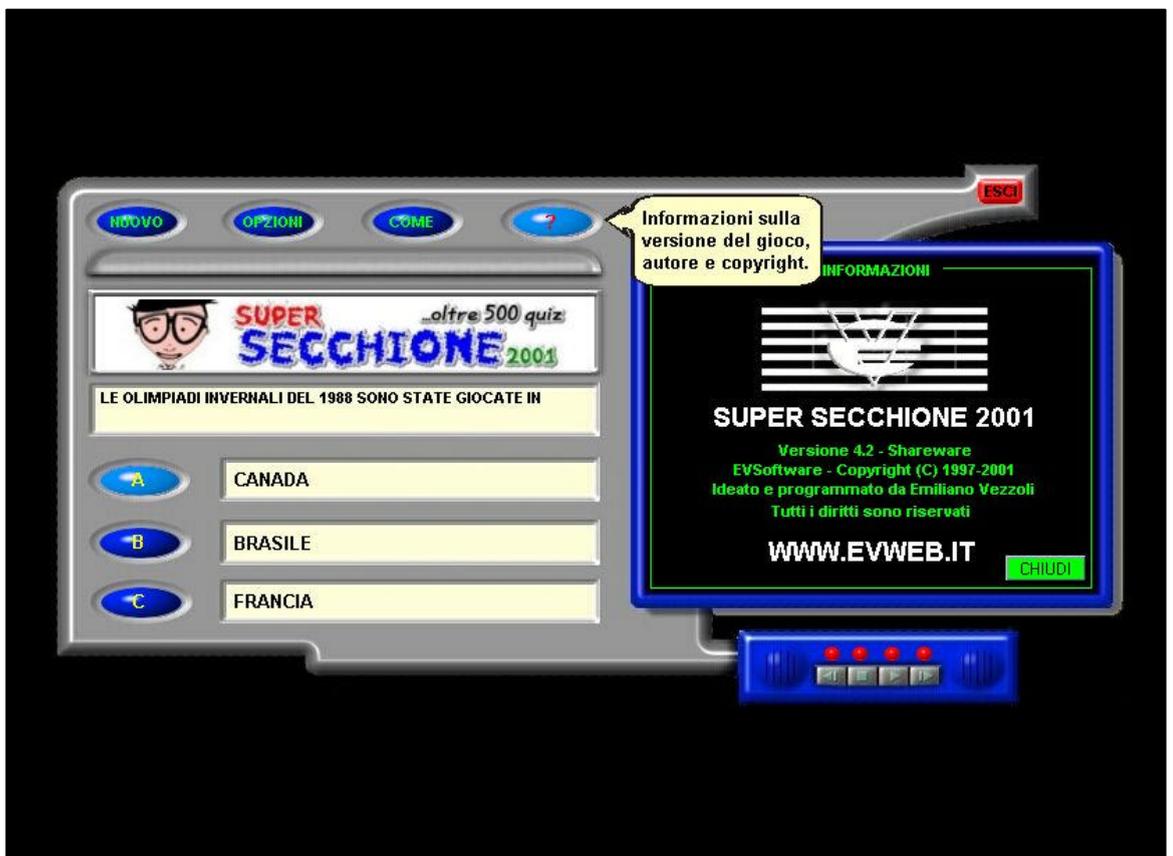


Fig. 8.5: Informazioni su autore e release del gioco.

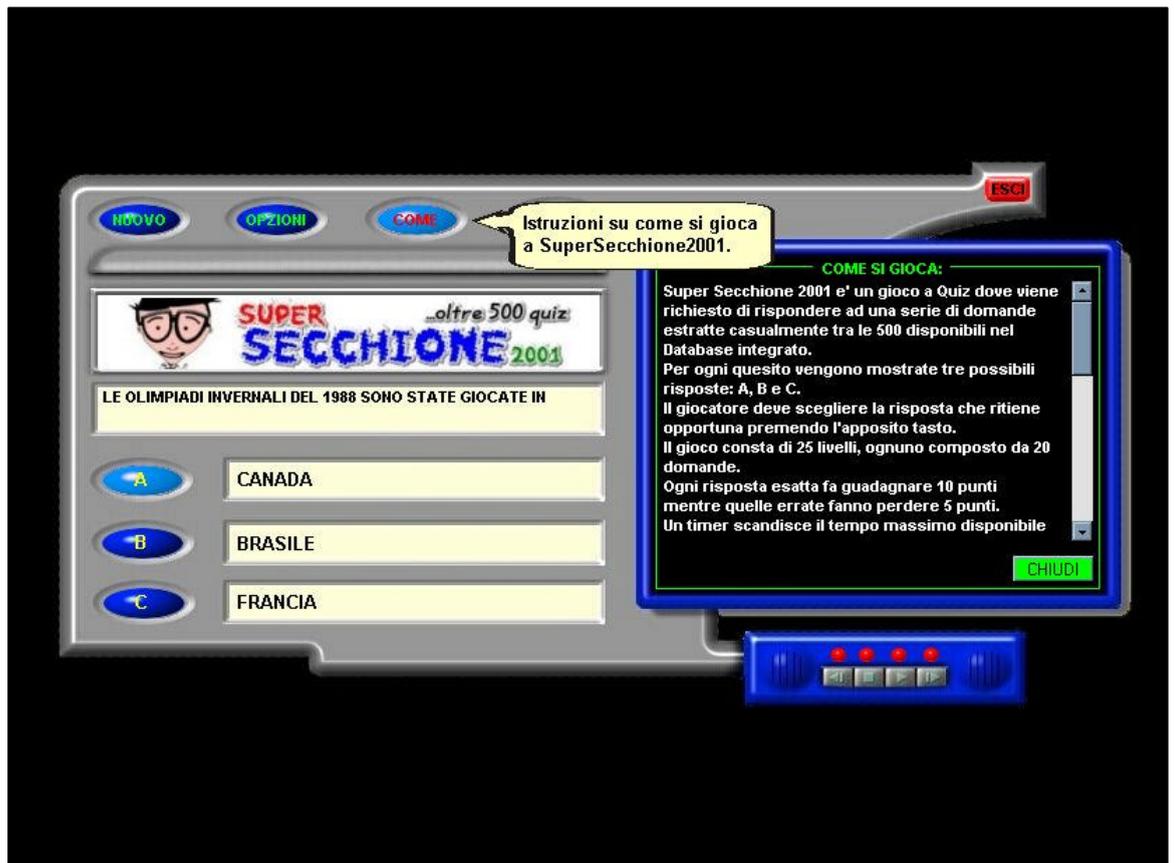


Fig. 8.6: Help On Line - Istruzioni per giocare.

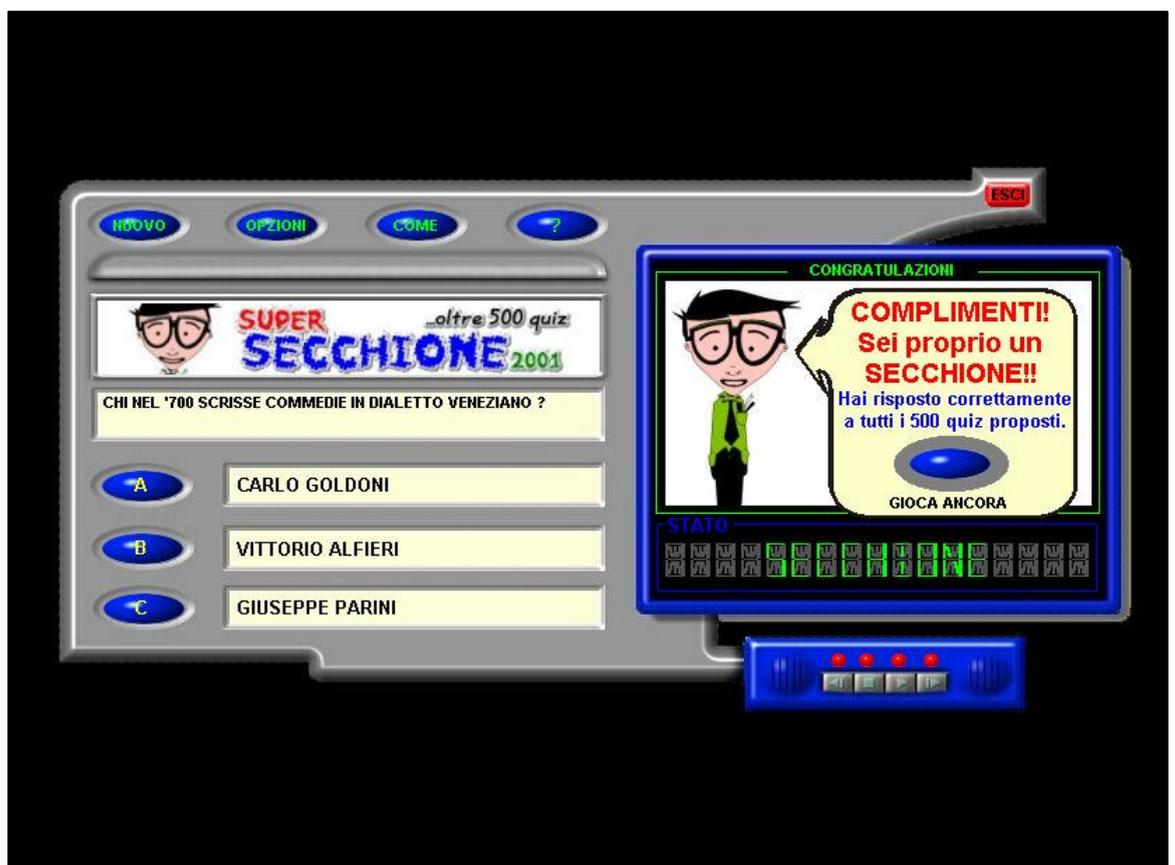


Fig. 8.7: Schermata di game over in caso di vittoria.

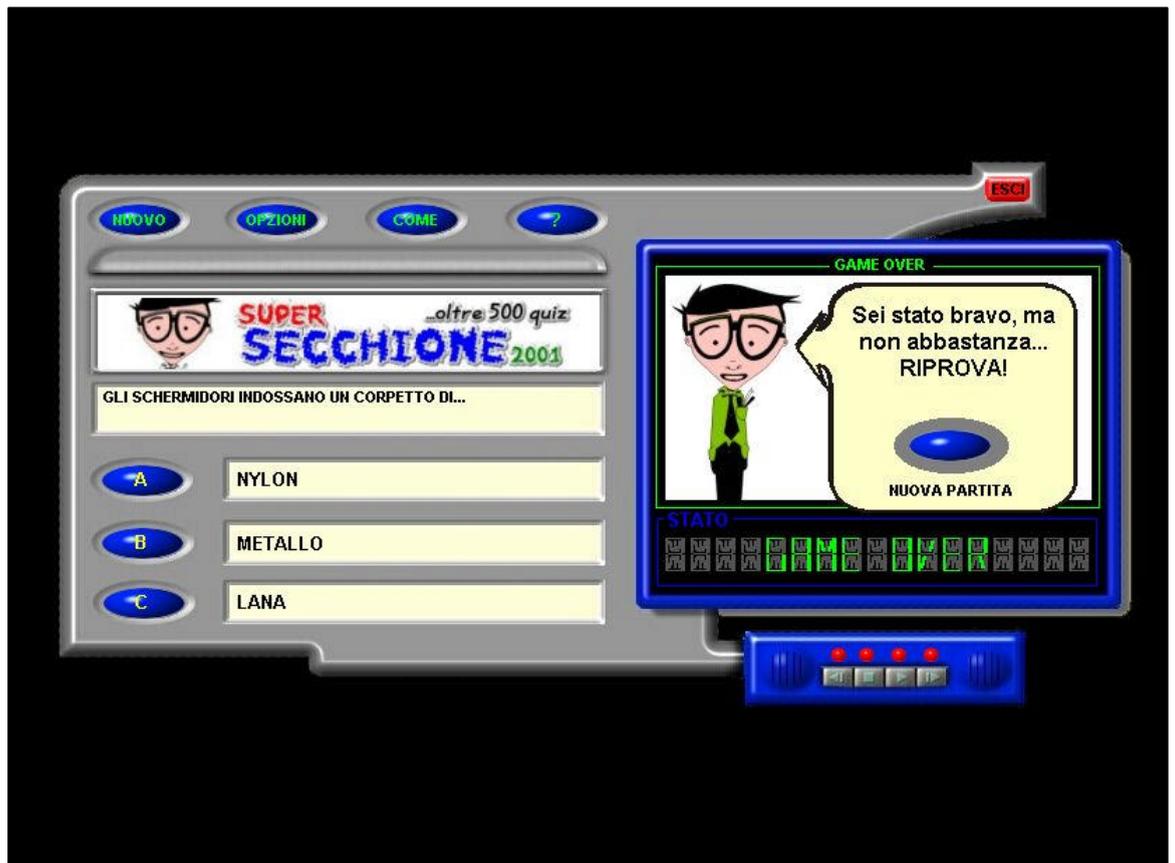


Fig. 8.8: Schermata di game over in caso di sconfitta.



Fig. 8.9: Finestra con le opzioni ed i parametri di gioco.

## 8.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
NDom=1
Click=0
riga=0
for ElementoLista=1 to 500
    LoadText("NumeDoman$", "<List>ElementoLista")
    NumeroDomanda[ElementoLista]=VAL(NumeDoman$)
next ElementoLista
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText("Text8", "Testo_NLivello$")
LoadText("Text9", "Testo_NDomanda$")
MP3ListLoad("<Embedded>\DataD.eqz")
LoadText("Domanda", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataA.eqz")
LoadText("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataB.eqz")
LoadText("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataC.eqz")
LoadText("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataE.eqz")
LoadText("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")

```

### □ Tasto BlueControlButton

```

pausa=0
Hide("Pic_Box")
Show("PIC_Views")
Hide("Animations")
Hide("R_pausa")
Hide("AniGif")
Hide("AniGif_Rack")
Risposta$='B'
RunScript("Script")
Click=1
Show("TB")
Show("BloccaTasti")

```

### □ Tasto Pausa

```

if (pausa=1) then
    pausa=0
    Hide("R_pausa")
    Hide("HotSpot9")
    Show("HotSpot8")
    RunScript("Timer")
end
if (pausa=0) then
    pausa=1
    Show("R_pausa")

```

```

    Show("HotSpot9")
End

```

#### □ Script Timer

```

if (pausa=0) then
    riga=riga+1
    for k=1 to 6
        MatrixSet("Matrix[44-riga,k]","1")
    next k
    WavePlay("tone","")
    if (riga<>43) then
        ScriptTimer("Timer","tempo")
    end
    if (riga=43 & Click=0) then
        RunScript("Script")
    end
end
end

```

#### □ Script Fine1

```

Show("Congratulazioni")
Hide("R_corretta")
Hide("R_errata")
Hide("R_temposcaduto")
WaveStop()
WavePlay("Bravo","")

```

#### □ Script Fine2

```

WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver")
Click=1
Hide("R_temposcaduto")

```

#### □ Script Fine3

```

WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver2")
Click=1

```

#### □ Script NuovaDomanda

```

if (NDom=20) then
    NDom=0
    DomandeGiuste=0
    NumeroLivello=NumeroLivello+1
end
riga=0
Click=0
Risposta$=''
Hide("R_errata")
Hide("R_corretta")
Hide("R_temposcaduto")
Hide("Evide")
for i=1 to 44
    for j=1 to 6
        MatrixSet("Matrix[i,j]","2")
    next j
next i
NumeroDomandeTotali=NumeroDomandeTotali+1
NDom=NDom+1
if (NumeroLivello=1) then

```

```

Testo_NLivello$='01'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=2) then
Testo_NLivello$='02'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=3) then
Testo_NLivello$='03'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=4) then
Testo_NLivello$='04'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=5) then
Testo_NLivello$='05'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=6) then
Testo_NLivello$='06'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=7) then
Testo_NLivello$='07'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=8) then
Testo_NLivello$='08'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=9) then
Testo_NLivello$='09'
LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello>=10) then
Testo1$=CHAR(NumeroLivello)
LoadText ("Text8", "Testo1$")
end
if (NDom=1) then
Testo_NDomanda$='01'
LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=2) then
Testo_NDomanda$='02'
LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=3) then
Testo_NDomanda$='03'
LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=4) then
Testo_NDomanda$='04'
LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=5) then
Testo_NDomanda$='05'
LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=6) then
Testo_NDomanda$='06'
LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=7) then
Testo_NDomanda$='07'

```

```

    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=8) then
    Testo_NDomanda$='08'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=9) then
    Testo_NDomanda$='09'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom>=10) then
    Testo_NDomanda$=CHAR(NDom)
    LoadText ("Text9", "Testo_NDomanda$")
end
Hide ("BloccaTasti")
MP3ListLoad("<Embedded>\DataD.eqz")
LoadText ("Domanda", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataA.eqz")
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataB.eqz")
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataC.eqz")
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\DataE.eqz")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
RunScript ("Timer")

```

#### □ Script Script

```

Hide ("Info")
Hide ("Come")
Hide ("GameOver")
Hide ("GameOver2")
Hide ("Congratulazioni")
if (Click=0 & Risposta$='') then
    Show("R_temposcaduto")
    WaveStop()
    WavePlay("Errore", "")
    punteggio=punteggio-5
end
if (Risposta$=RispostaEsatta$) then
    Show("R_corretta")
    WaveStop()
    WavePlay("Power", "")
    punteggio=punteggio+10
    DomandeGiuste=DomandeGiuste+1
end
if (Risposta$<>RispostaEsatta$ & Risposta$<>'') then
    Show("R_errata")
    WaveStop()
    WavePlay("Errore", "")
    punteggio=punteggio-5
end
if (punteggio<=0) then
    punteggio=0
    Testo_Punteggio$='0000'
end
if (punteggio>=0 & punteggio<10) then
    Testo_Punteggio$='000'+CHAR(punteggio)
end
if (punteggio>=10 & punteggio<100) then
    Testo_Punteggio$='00'+CHAR(punteggio)
end
if (punteggio>=100 & punteggio<1000) then
    Testo_Punteggio$='0'+CHAR(punteggio)
end

```

```

end
if (punteggio>=1000) then
    Testo_Punteggio$=CHAR(punteggio)
end
LoadText ("Text10", "Testo_Punteggio$")
if (punteggio<5000) then
    if (NDom<>20 | DomandeGiuste>=15) then
        ScriptTimer ("NuovaDomanda", "2000")
    end
end
if (punteggio=5000 & NumeroLivello=25) then
    ScriptTimer ("Fine1", "2000")
end
if (NDom=20 & DomandeGiuste<15 & NumeroLivello<>25) then
    ScriptTimer ("Fine2", "2000")
end
if (NDom=20 & DomandeGiuste<15 & NumeroLivello=25) then
    ScriptTimer ("Fine3", "2000")
end

```

## □ Tasto Nuovo

```

Hide ("GameOver")
Hide ("GameOver2")
Hide ("Congratulazioni")
Hide ("Info")
Hide ("Come")
Hide ("BloccaTasti")
Hide ("Evide")
Hide ("Opzioni")
NumeroLivello=1
punteggio=0
DomandeGiuste=0
Testo_Punteggio$='0000'
LoadText ("Text10", "Testo_Punteggio$")
NDom=1
Click=0
Risposta$=''
Hide ("R_errata")
Hide ("R_corretta")
Hide ("R_temposcaduto")
for i=1 to 44
    for j=1 to 6
        MatrixSet ("Matrix[i,j]", "2")
    next j
next i
riga=0
MP3ListLoad ("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
for ElementoLista=1 to 500
    LoadText ("NumeDoman$", "<List>ElementoLista")
    NumeroDomanda[ElementoLista]=VAL(NumeDoman$)
next ElementoLista
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText ("Text8", "Testo_NLivello$")
LoadText ("Text9", "Testo_NDomanda$")
MP3ListLoad ("<Embedded>\DataD.eqz")
LoadText ("Domanda", "<List>NumeroDomanda[NumeroDomandeTotali]")

```

```

MP3ListLoad ("<Embedded>\DataA.eqz")
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\DataB.eqz")
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\DataC.eqz")
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\DataE.eqz")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
    
```

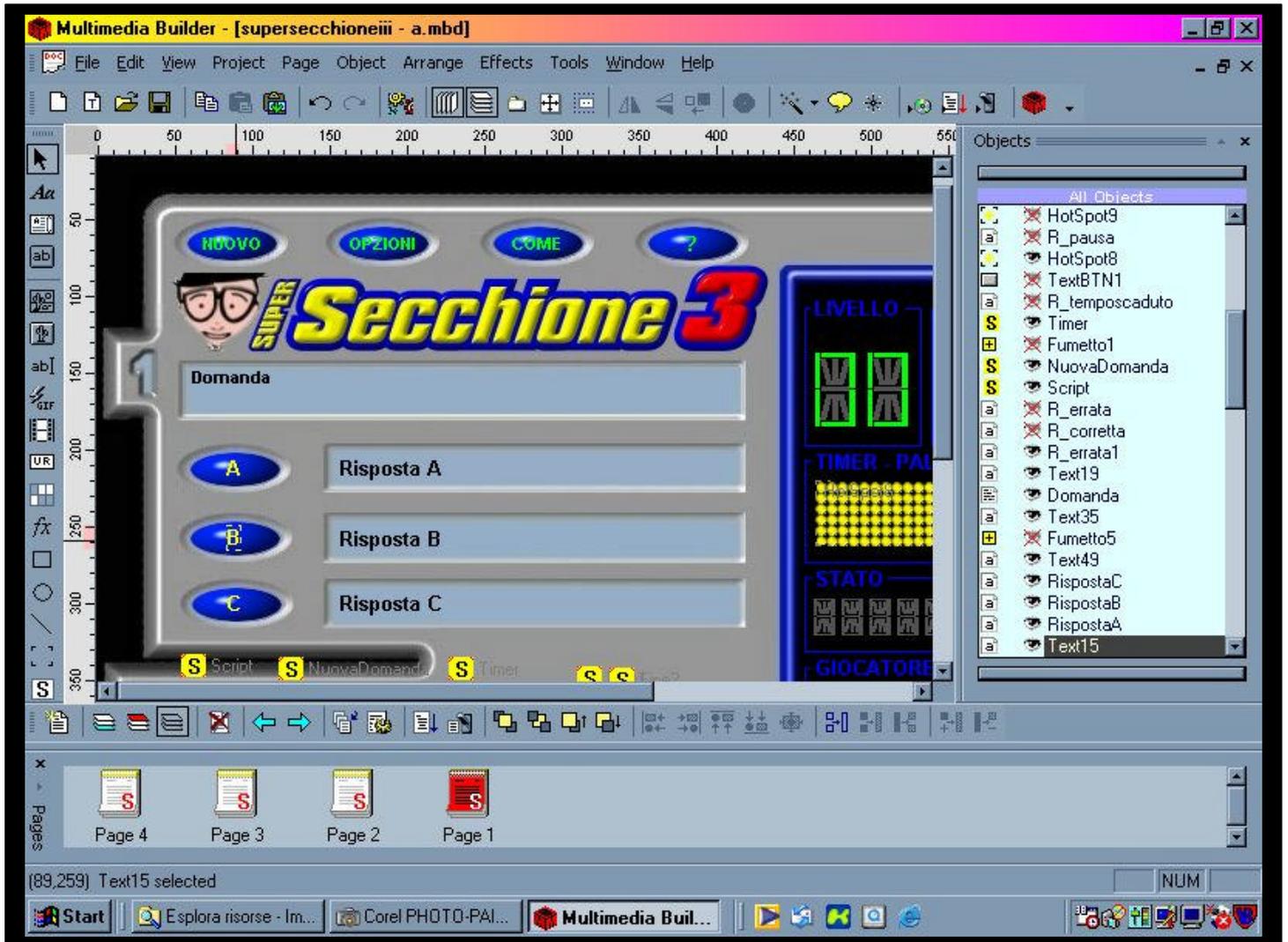


Fig. 8.10: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

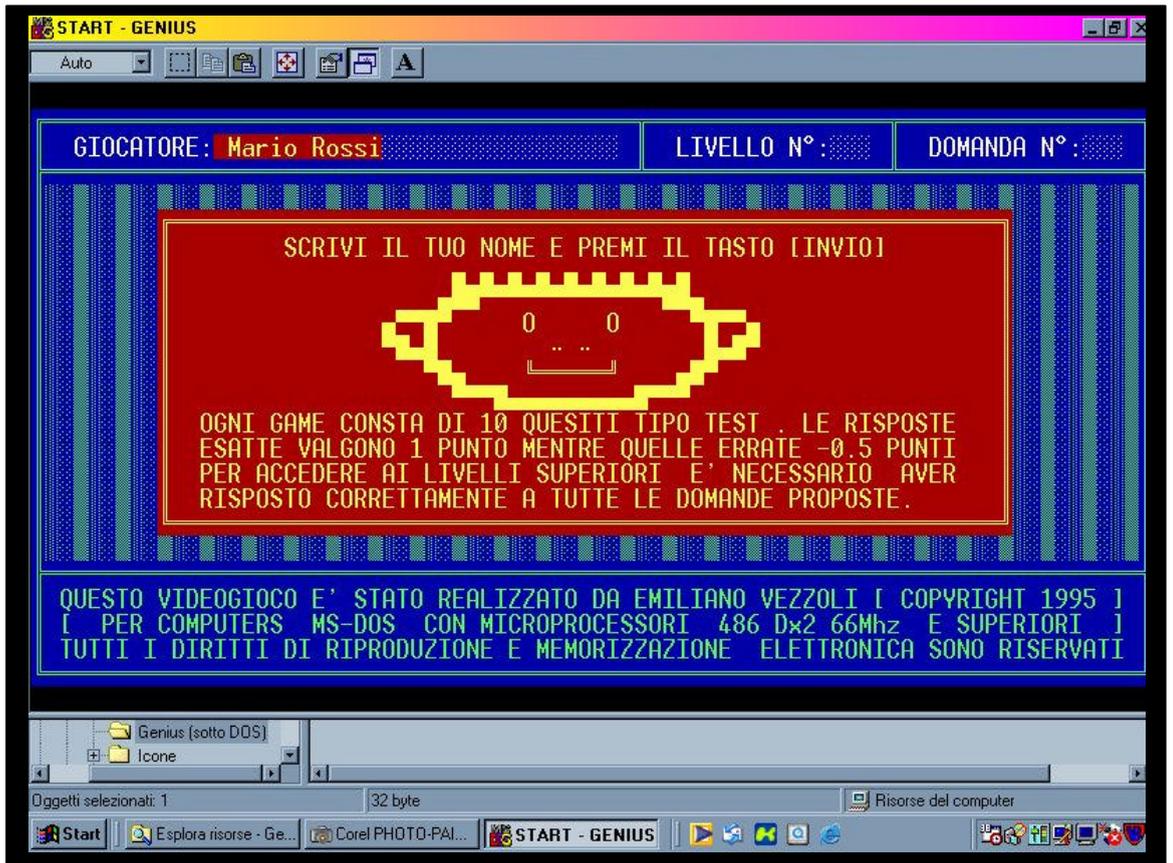


Fig. 8.11: Menu di accesso al gioco GENIUS del 1995.

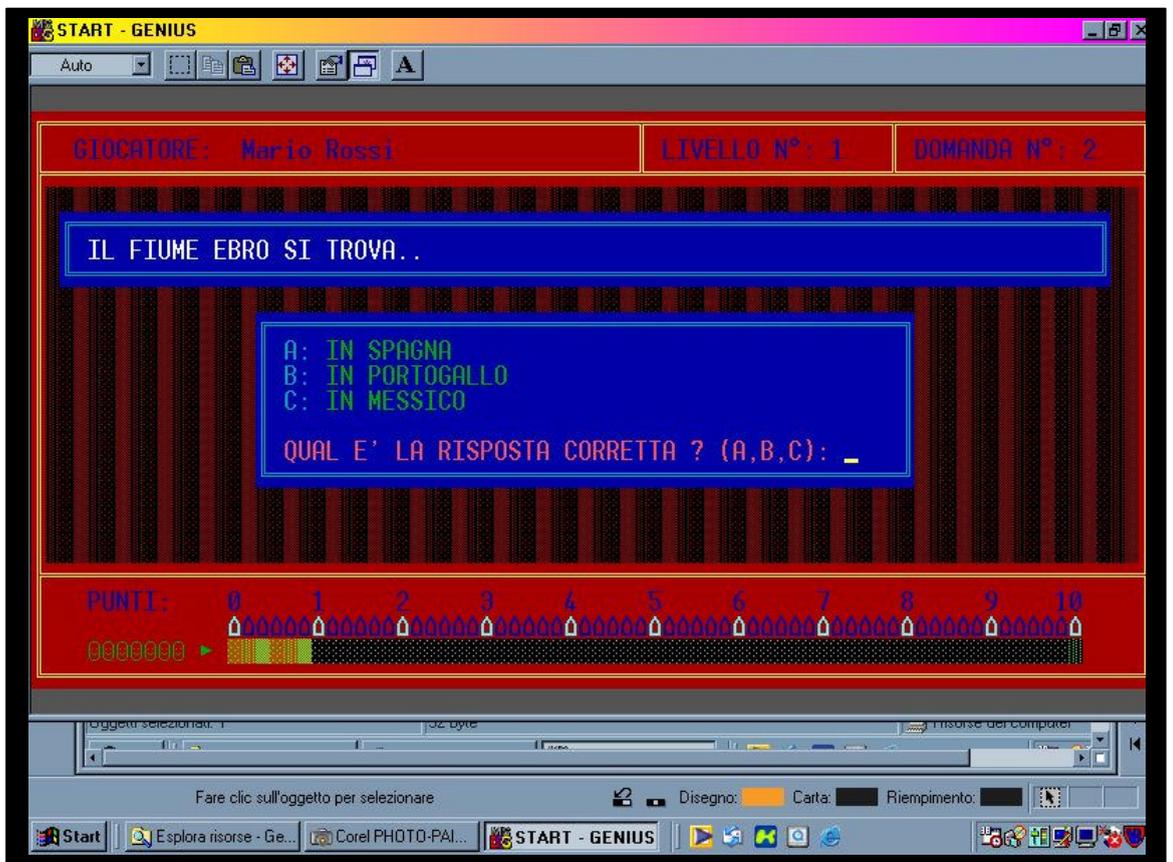


Fig. 8.12: schermata di gioco principale di GENIUS del 1995.

## 9. UP & DOWN – GUERRE STELLARI

- ❑ Versione 3.0 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2003 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti di riproduzione e memorizzazione elettronica sono riservati.
- ❑ Guerre Stellari, nomi, loghi, suoni ed immagini sono marchi depositati di Lucasfilm Ltd.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 2,270 Mb

### 9.1 REGOLE DEL GIOCO

Up & Down è un gioco per un solo giocatore ed è basato sulla fortuna. Lo scopo è di indovinare se la carta estratta casualmente dal mazzo virtuale ha un valore superiore o inferiore a quella corrente. I valori possibili sono 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K. Il giocatore deve soltanto premere sui tasti UP (su) oppure DOWN (giu).

Per iniziare una nuova partita basta premere sul tasto "START". All'inizio della partita vengono assegnati al giocatore 10 tentativi. Questo significa che si può sbagliare al massimo 10 volte. In base al punteggio raggiunto, vengono assegnati dei bonus del valore di 10000, 50000 e 100000 punti, oppure vengono incrementati i tentativi a disposizione.

Caratteristica peculiare di questo videogame è il tema delle fotografie stampate sulle carte. Nella release numero 3 il tema è "Guerre Stellari" e le immagini raffigurano i protagonisti della celebre saga cinematografica. Il tasto "AUD ON/OFF" serve ad abilitare/disabilitare l'audio. Durante la partita, infatti, è possibile ascoltare uno dei celebri brani della colonna sonora di "Star Wars": "La cantina", tratto da "Il ritorno dello Jedi".

Premendo sul tasto "INFO" si apre, invece, la finestra con le informazioni su autore e release del videogame, alternandosi alla finestra attiva di default con le istruzioni per giocare. Il numero di giocatori non è, invece, modificabile. Nessun'altra opzione di gioco è disponibile nella corrente release.

### 9.2 SUGGERIMENTI E TRUCCHI

Non ci sono particolari strategie per giocare a questo videogame. Le carte, infatti, vengono estratte casualmente dal mazzo, quindi è difficile prevedere l'esito finale di una giocata. In ogni caso ci si può affidare al calcolo delle probabilità. Per esempio: se la carta attuale è un "2", la probabilità che la carta estratta abbia un valore superiore a 2 (3,4, ... fino a K) è notevolmente più alta che non un valore inferiore (1).

### 9.3 LA REALIZZAZIONE (THE MAKING OF)

#### □ LA STORIA

Up & Down vede la luce nel 1999, con una prima versione un po' particolare e concepita appositamente per attirare nuove utenze nel sito web [www.evweb.it](http://www.evweb.it). La sua caratteristica è la natura erotica delle immagini raffigurate sulle carte, con numerose fotografie di modelle in parte senza veli. Il successo del gioco è scontato, nonostante l'interfaccia grafica piuttosto mediocre.

A questo difetto si pone rimedio nel 2001 quando viene distribuita la versione 2. Nuova grafica e nuove immagini, sempre più maliziose. A causa del tema delle carte la diffusione del gioco si rivela, però, rivolta ad un solo pubblico adulto.

Il desiderio di inserire questo gioco nel "Progetto Gameisland" spinge quindi verso una modifica radicale: sostituire le fotografie delle modelle con altre. Ma quali? Che tema scegliere? Dopo una breve riflessione si sceglie un tema molto amato dai navigatori della Rete: "Guerre Stellari". Questo per assicurare un favorevole riscontro del pubblico, amante dei videogame amatoriali e, contemporaneamente, della famosa saga creata da George Lucas.

Dal 1999 il videogame è stato prelevato, nelle sue tre versioni, dal sito web [www.evweb.it](http://www.evweb.it) più di 2000 volte.

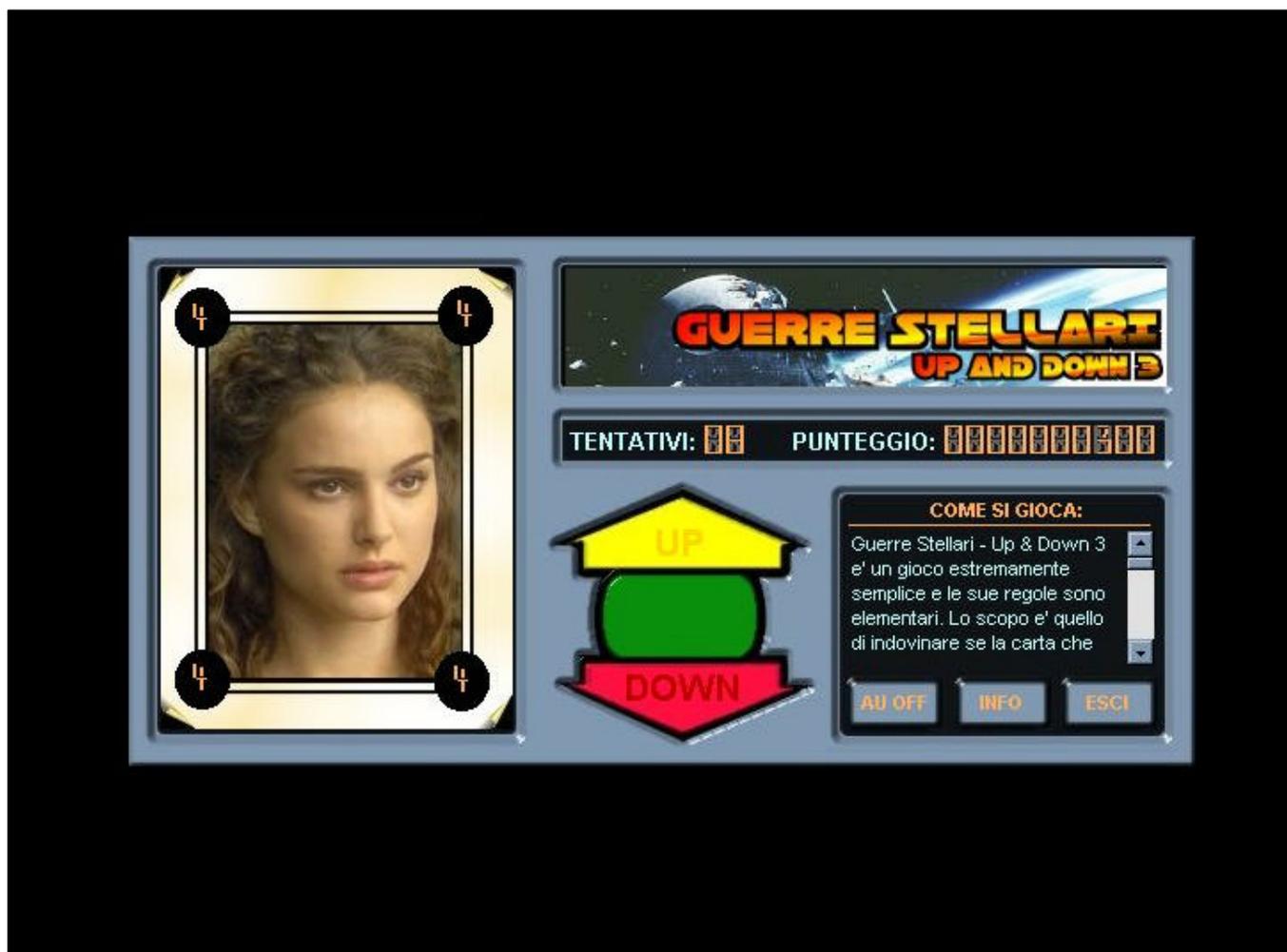


Fig. 9.1: interfaccia principale del videogame.

## □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 9.6).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

## □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 9.2 e 9.3 due immagini dell'interfaccia principale del videogioco. In fig. 9.4 come appaiono le informazioni su autore e release del gioco. In fig. 9.5: Finestra con le istruzioni per giocare.

Infine, in fig. 9.6 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

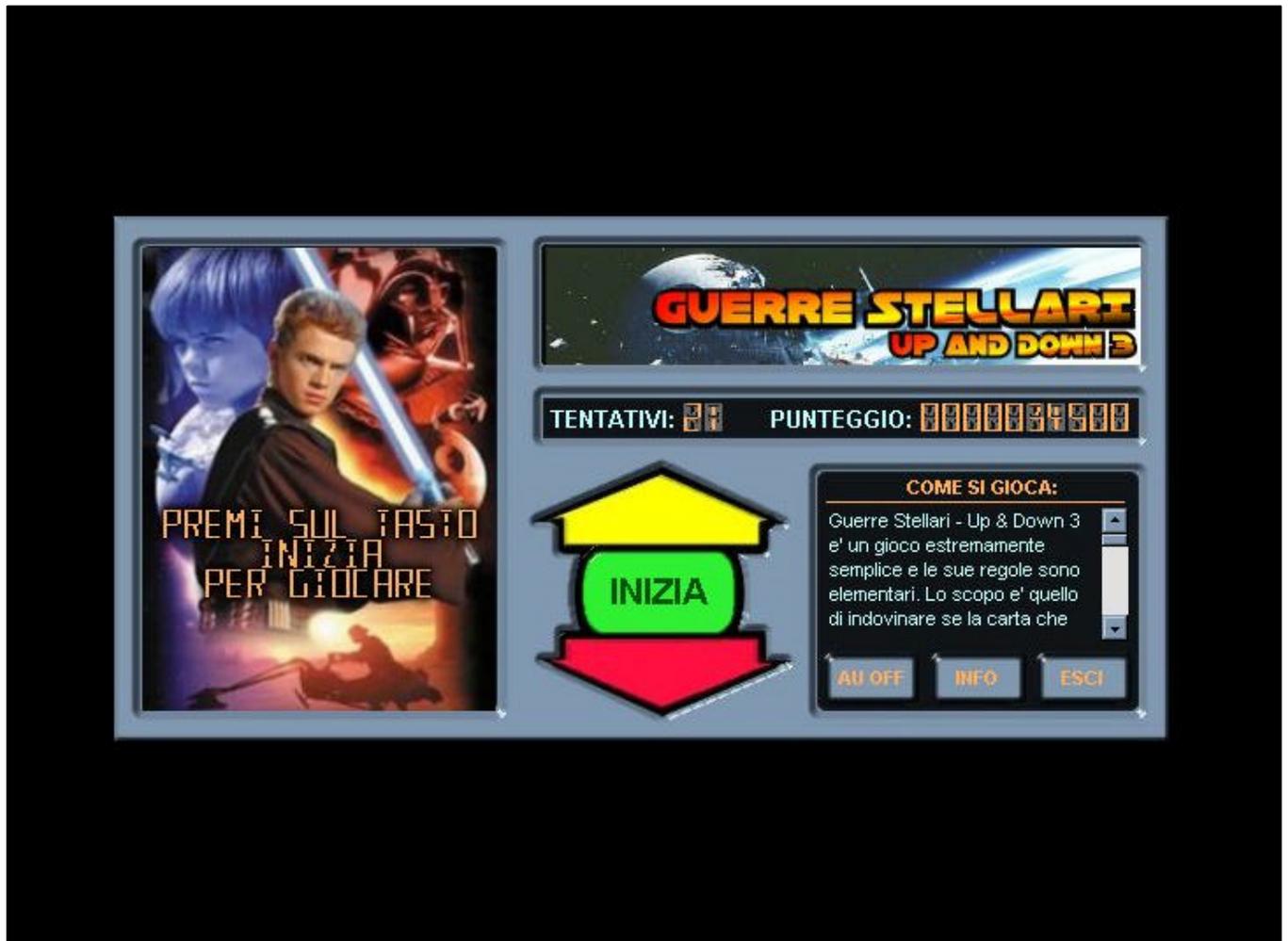


Fig. 9.2: screenshot principale di Up & Down 3 – Guerre Stellari.

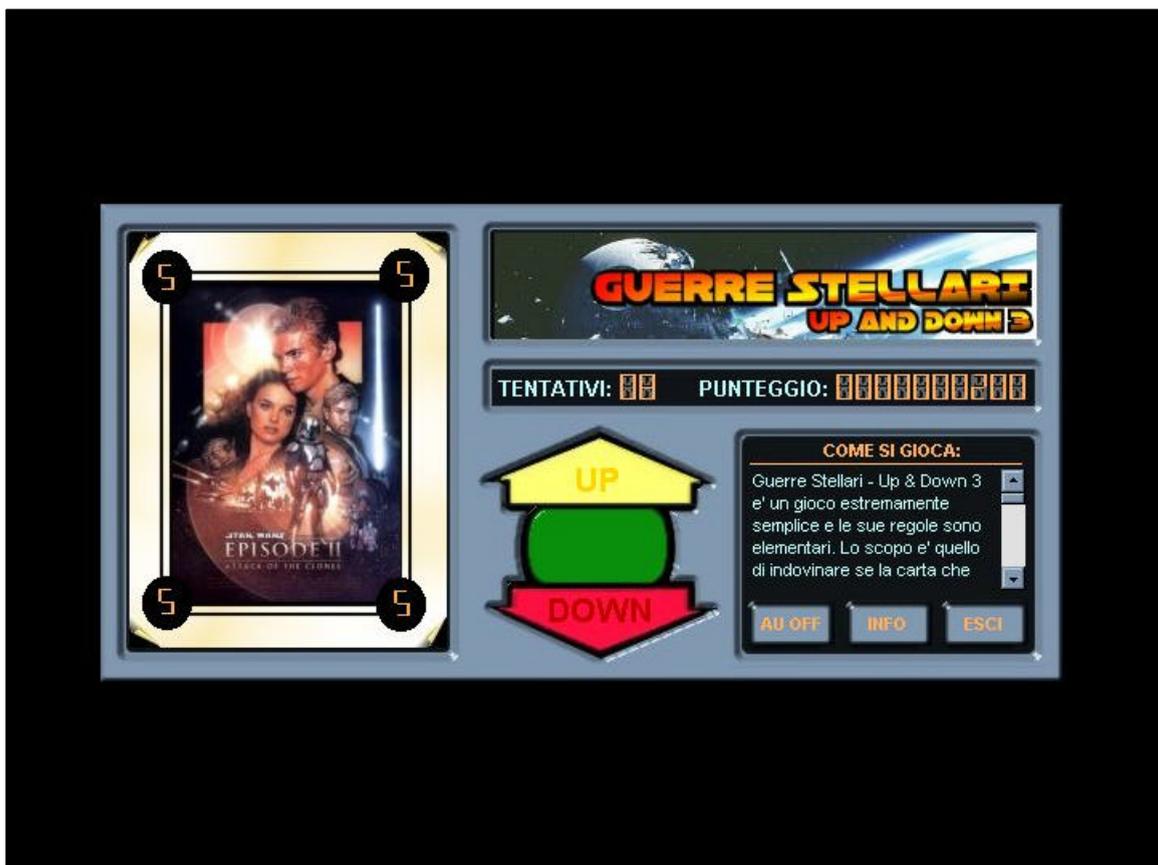


Fig. 9.3: schermata principale del gioco.

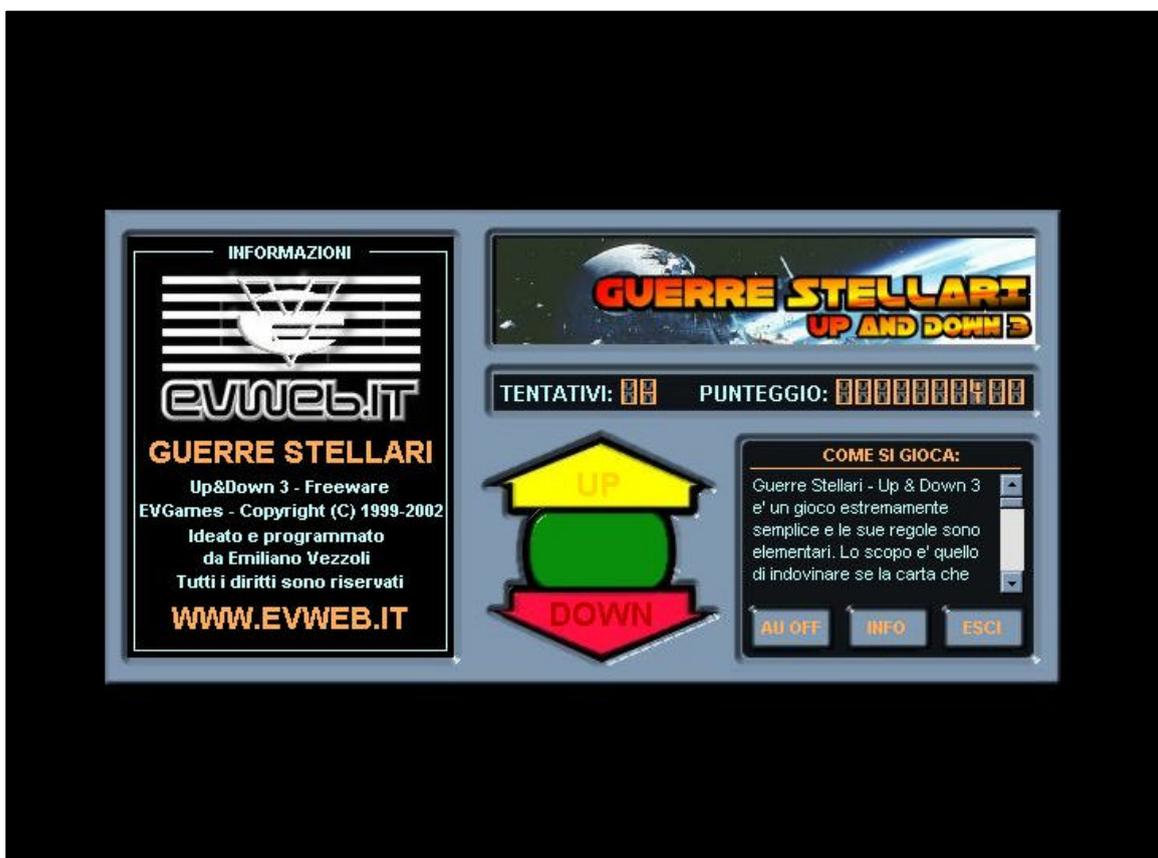


Fig. 9.4: informazioni su autore e release del gioco.

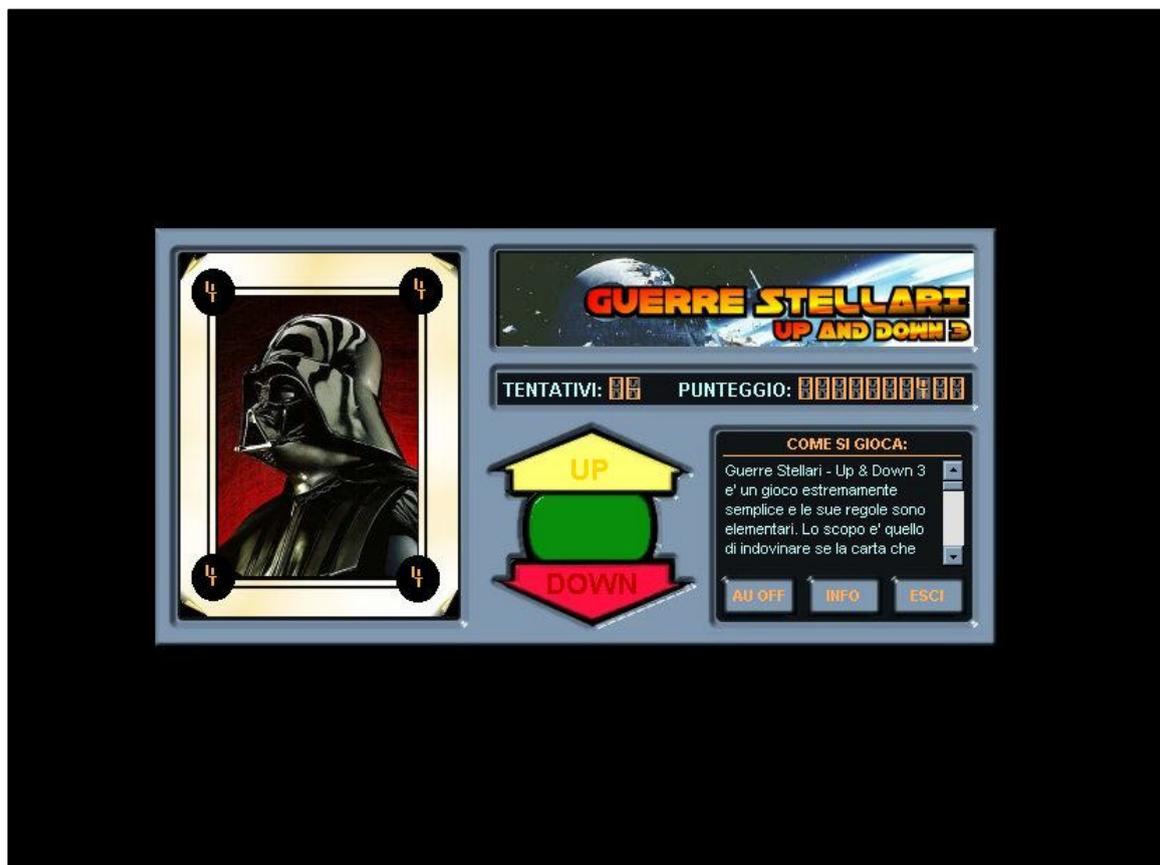


Fig. 9.5: in basso a destra c'è la finestra con le istruzioni per giocare.

## 9.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
PageTimer("1000", "Page 3")
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
punti=0
tentativi=10
Testo_Punteggio$='0000000000'
Testo_Tentativi$='10'
LoadText("Punteggio", "Testo_Punteggio$")
LoadText("Tentativi", "Testo_Tentativi$")
MidiPlay("Cantina", "[LOOP]")
```

### □ HotSpot1 – Start

```
carta=RND(12)+1
D=carta
punti=0
tentativi=10
Testo_Punteggio$='0000000000'
Testo_Tentativi$='10'
LoadText("Punteggio", "Testo_Punteggio$")
LoadText("Tentativi", "Testo_Tentativi$")
riga=0
Show("Matrix")
MatrixSet("Matrix[0,0]", "2")
if (tentativi>=1) then
```

```

    RunScript( "Game" )
end
Show( "Up" )
Show( "Down" )
Show( "HotSpot3" )
Hide( "HotSpot4" )
Hide( "HotSpot5" )
NBonus=1
NBonus1=1
NBonus2=1

```

#### □ HotSpot2 – Down

```

Control=0
riga=0
Show( "Matrix" )
Show( "HotSpot4" )
Show( "HotSpot5" )
MatrixSet( "Matrix[0,0]", "2" )
carta=RND(12)+1
if (carta<=D) then
    punti=punti+100
    Control=1
    D=carta
end
if (carta>D) then
    tentativi=tentativi-1
    D=carta
end
RunScript( "Game" )

```

#### □ Hotspot – Up

```

Control=0
riga=0
Show( "Matrix" )
Show( "HotSpot4" )
Show( "HotSpot5" )
MatrixSet( "Matrix[0,0]", "2" )
carta=RND(12)+1
if (carta>=D) then
    punti=punti+100
    Control=1
    D=carta
end
if (carta<D) then
    tentativi=tentativi-1
    D=carta
end
RunScript( "Game" )

```

#### □ Tasto AU ON

```

MidiPlay( "Cantina", "[LOOP]" )
Hide( "Auon" )
Show( "Auoff" )

```

#### □ Tasto Info ON

```

Show( "info" )
Show( "Infooff" )
Hide( "Infoon" )

```

#### □ Tasto AU OFF

```

MidiStop( )
Hide( "Auoff" )
Show( "Auon" )

```

## □ Tasto Info OFF

```
Hide("info")
Hide("Infooff")
Show("Infoon")
```

## □ Script Game2

```
Hide("Bonus10000")
Hide("Bonus50000")
Hide("Bonus100000")
Hide("Hard")
Hide("info")
Hide("info")
Hide("GameOver")
if (tentativi<=0) then
  Show("GameOver")
  Show("Start")
  Hide("Up")
  Hide("Down")
  Hide("HotSpot3")
  Show("HotSpot4")
  Show("HotSpot5")
end
if (carta<>10) then
  Hide("Group10")
  Show("NAS")
  Show("NAD")
  Show("NBS")
  Show("NBD")
end
Hide("Promo")
if (punti = 10000*NBonus ) then
  punti=punti+10000
  Show("Bonus10000")
  NBonus=Nbonus+1
  tentativi=tentativi+1
end
if (punti = 50000*NBonus1 ) then
  punti=punti+50000
  Show("Bonus50000")
  NBonus1=Nbonus1+3
  tentativi=tentativi+3
end
if (punti = 100000*NBonus2 ) then
  punti=punti+100000
  Show("Bonus100000")
  NBonus2=Nbonus2+5
  tentativi=tentativi+5
end
if (punti = 1000000000) then
  Show("Hard")
end
if (tentativi<10) then
  Testo_Tentativi$='0'+CHAR(tentativi)
end
if (punti<=0) then
  punti=0
  Testo_Punteggio$='0000000000'
end
if (punti>=0 & punti<10) then
  Testo_Punteggio$='00000000'+CHAR(punti)
end
if (punti>=10 & punti<100) then
  Testo_Punteggio$='0000000'+CHAR(punti)
end
```

```

if (punti>=100 & punti<1000) then
  Testo_Punteggio$='000000'+CHAR(punti)
end
if (punti>=1000 & punti<10000) then
  Testo_Punteggio$='000000'+CHAR(punti)
end
if (punti>=10000 & punti<100000) then
  Testo_Punteggio$='00000'+CHAR(punti)
end
if (punti>=100000 & punti<1000000) then
  Testo_Punteggio$='0000'+CHAR(punti)
end
if (punti>=1000000 & punti<10000000) then
  Testo_Punteggio$='000'+CHAR(punti)
end
if (punti>=10000000 & punti<100000000) then
  Testo_Punteggio$='00'+CHAR(punti)
end
if (punti>=100000000 & punti<1000000000) then
  Testo_Punteggio$='0'+CHAR(punti)
end
if (punteggio>=1000000000) then
  Testo_Punteggio$=CHAR(punteggio)
end
LoadText ("Punteggio", "Testo_Punteggio$")
LoadText ("Tentativi", "Testo_Tentativi$")
if (carta=1 & tentativi>=1) then
  carta$=CHAR(carta)
  LoadText ("NBD", "carta$")
  LoadText ("NBS", "carta$")
  LoadText ("NAD", "carta$")
  LoadText ("NAS", "carta$")
end
if (carta=2 & tentativi>=1) then
  carta$=CHAR(carta)
  LoadText ("NBD", "carta$")
  LoadText ("NBS", "carta$")
  LoadText ("NAD", "carta$")
  LoadText ("NAS", "carta$")
end
if (carta=3 & tentativi>=1) then
  carta$=CHAR(carta)
  LoadText ("NBD", "carta$")
  LoadText ("NBS", "carta$")
  LoadText ("NAD", "carta$")
  LoadText ("NAS", "carta$")
end
if (carta=4 & tentativi>=1) then
  carta$=CHAR(carta)
  LoadText ("NBD", "carta$")
  LoadText ("NBS", "carta$")
  LoadText ("NAD", "carta$")
  LoadText ("NAS", "carta$")
end
if (carta=5 & tentativi>=1) then
  carta$=CHAR(carta)
  LoadText ("NBD", "carta$")
  LoadText ("NBS", "carta$")
  LoadText ("NAD", "carta$")
  LoadText ("NAS", "carta$")
end
if (carta=6 & tentativi>=1) then
  carta$=CHAR(carta)
  LoadText ("NBD", "carta$")
  LoadText ("NBS", "carta$")

```

```

    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
if (carta=7 & tentativi>=1) then
    carta$=CHAR(carta)
    LoadText("NBD", "carta$")
    LoadText("NBS", "carta$")
    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
if (carta=8 & tentativi>=1) then
    carta$=CHAR(carta)
    LoadText("NBD", "carta$")
    LoadText("NBS", "carta$")
    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
if (carta=9 & tentativi>=1) then
    carta$=CHAR(carta)
    LoadText("NBD", "carta$")
    LoadText("NBS", "carta$")
    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
if (carta=10 & tentativi>=1) then
    Show("Group10")
    Hide("NAS")
    Hide("NAD")
    Hide("NBS")
    Hide("NBD")
end
if (carta=11 & tentativi>=1) then
    carta$='J'
    LoadText("NBD", "carta$")
    LoadText("NBS", "carta$")
    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
if (carta=12 & tentativi>=1) then
    carta$='Q'
    LoadText("NBD", "carta$")
    LoadText("NBS", "carta$")
    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
if (carta=13 & tentativi>=1) then
    carta$='K'
    LoadText("NBD", "carta$")
    LoadText("NBS", "carta$")
    LoadText("NAD", "carta$")
    LoadText("NAS", "carta$")
end
soggetto=RND(24)+1
if (soggetto=1) then
    ReplaceImage("CartaVuota", "<Embedded>carta01.jpg")
    if (punti>1000000000) then
        ReplaceImage("CartaVuota", "<Embedded>carta01x.jpg")
    end
end
if (soggetto=2) then
    ReplaceImage("CartaVuota", "<Embedded>carta02.jpg")
    if (punti>1000000000) then
        ReplaceImage("CartaVuota", "<Embedded>carta02x.jpg")
    end
end
end

```

```

if (soggetto=3) then
  ReplaceImage ("CartaVuota", "<Embedded>carta03.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta03x.jpg")
  end
end
if (soggetto=4) then
  ReplaceImage ("CartaVuota", "<Embedded>carta04.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta04x.jpg")
  end
end
if (soggetto=5) then
  ReplaceImage ("CartaVuota", "<Embedded>carta05.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta05x.jpg")
  end
end
if (soggetto=6) then
  ReplaceImage ("CartaVuota", "<Embedded>carta06.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta06x.jpg")
  end
end
if (soggetto=7) then
  ReplaceImage ("CartaVuota", "<Embedded>carta07.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta07x.jpg")
  end
end
if (soggetto=8) then
  ReplaceImage ("CartaVuota", "<Embedded>carta08.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta08x.jpg")
  end
end
if (soggetto=9) then
  ReplaceImage ("CartaVuota", "<Embedded>carta09.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta09x.jpg")
  end
end
if (soggetto=10) then
  ReplaceImage ("CartaVuota", "<Embedded>carta10.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta10x.jpg")
  end
end
if (soggetto=11) then
  ReplaceImage ("CartaVuota", "<Embedded>carta11.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta11x.jpg")
  end
end
if (soggetto=12) then
  ReplaceImage ("CartaVuota", "<Embedded>carta12.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta12x.jpg")
  end
end
if (soggetto=13) then
  ReplaceImage ("CartaVuota", "<Embedded>carta13.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta13x.jpg")
  end
end

```

```

end
if (soggetto=14) then
  ReplaceImage ("CartaVuota", "<Embedded>carta14.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta14x.jpg")
  end
end
if (soggetto=15) then
  ReplaceImage ("CartaVuota", "<Embedded>carta15.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta15x.jpg")
  end
end
if (soggetto=16) then
  ReplaceImage ("CartaVuota", "<Embedded>carta16.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta16x.jpg")
  end
end
if (soggetto=17) then
  ReplaceImage ("CartaVuota", "<Embedded>carta17.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta17x.jpg")
  end
end
if (soggetto=18) then
  ReplaceImage ("CartaVuota", "<Embedded>carta18.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta18x.jpg")
  end
end
if (soggetto=19) then
  ReplaceImage ("CartaVuota", "<Embedded>carta19.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta19x.jpg")
  end
end
if (soggetto=20) then
  ReplaceImage ("CartaVuota", "<Embedded>carta20.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta20x.jpg")
  end
end
if (soggetto=21) then
  ReplaceImage ("CartaVuota", "<Embedded>carta21.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta21x.jpg")
  end
end
if (soggetto=22) then
  ReplaceImage ("CartaVuota", "<Embedded>carta22.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta22x.jpg")
  end
end
if (soggetto=23) then
  ReplaceImage ("CartaVuota", "<Embedded>carta23.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta23x.jpg")
  end
end
if (soggetto=24) then
  ReplaceImage ("CartaVuota", "<Embedded>carta24.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta24x.jpg")
  end
end

```

```

end
end
if (soggetto=25) then
  ReplaceImage ("CartaVuota", "<Embedded>carta25.jpg")
  if (punti>1000000000) then
    ReplaceImage ("CartaVuota", "<Embedded>carta25x.jpg")
  end
end
if (tentativi>99) then
  tentativi=99
end
ScriptTimer ("Game3", "200")

```

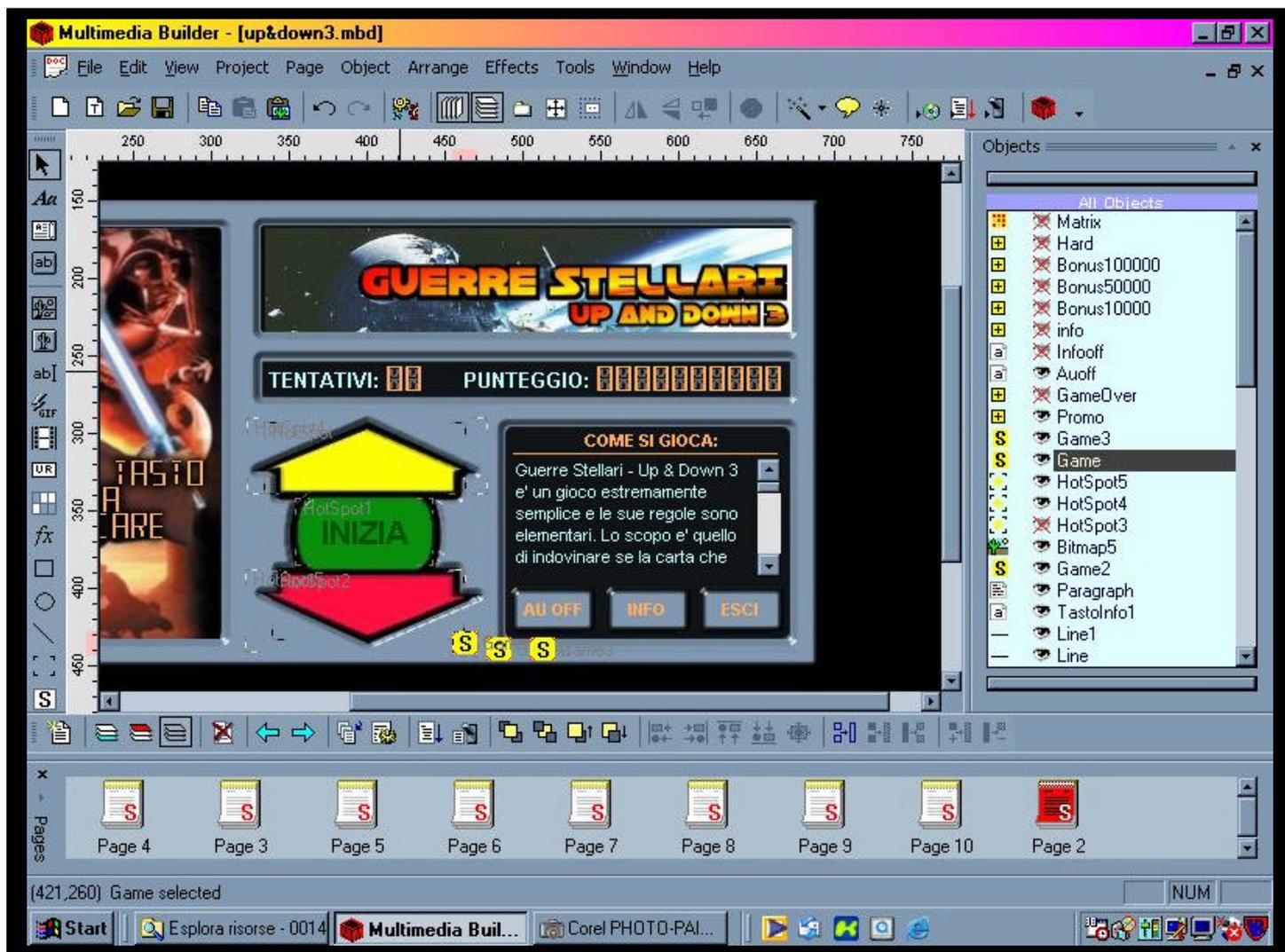


Fig. 9.6: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

□ **Script Game**

```

riga=riga+1
for k=1 to 38
  MatrixSet ("Matrix[30-riga,k]", "1")
next k
if (riga<>29) then
  ScriptTimer ("Game", "5")
end
if (riga=29) then
  riga=0
  RunScript ("Game2")
end

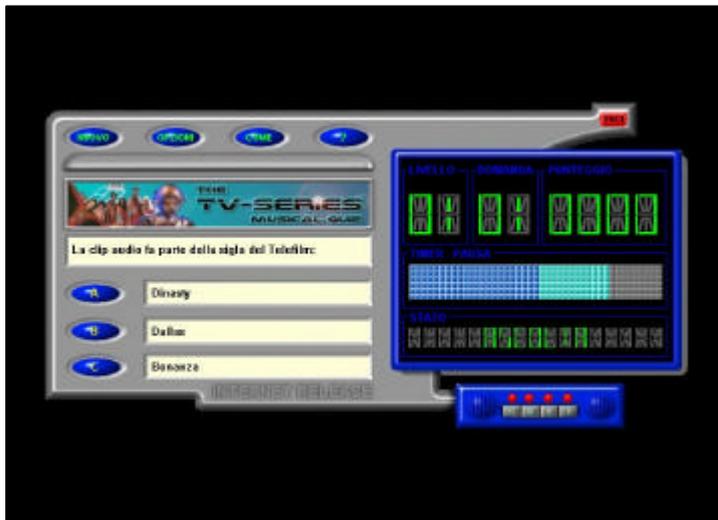
```

□ **Script Game 3**

```
riga=riga+1
for k=1 to 38
  MatrixSet("Matrix[30-riga,k]","2")
next k
if (riga<>29) then
  ScriptTimer("Game3","5")
end
if (riga=29) then
  riga=0
end
Hide("HotSpot4")
Hide("HotSpot5")
```

## 10. TV-SERIES MUSICAL GAME

- ❑ Versione 2.5 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2001 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti riguardanti le clip audio nel formato mp3 sono riservati agli artisti ed alle case discografiche che li detengono.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 4,74 Mb

### 10.1 REGOLE DEL GIOCO

TV-Series Musical Quiz è un gioco a quiz musicale per un solo giocatore. L'obiettivo è di indovinare il maggior numero di sigle di telefilm tra quelle proposte. L'archivio comprende ben 50 clip audio della durata di 20 secondi, con le più famose sigle dei telefilm trasmessi dalle reti televisive italiane negli anni '80-'90. Da "Star Trek" a "Mork e Mindy", da "Beverly hills" a "Magnum P.I."

Per giocare una partita è sufficiente premere il tasto "NUOVO". Inizia così la riproduzione di una clip musicale e viene richiesto di selezionare a quale telefilm associarla. Ci sono tre possibilità: A, B o C. Per effettuare la propria scelta basta premere il tasto corrispondente. Le impostazioni di default prevedono una durata dell'ascolto pari a 10 secondi, seguiti da altri 10 per consentire al giocatore di fare la propria scelta. In totale ci sono, quindi, 20 secondi per rispondere.

L'aspetto grafico consente al giocatore di tenere sotto controllo tutti gli aspetti della partita. Il primo elemento è costituito dalla barra di stato azzurro/blu. La barra azzurra indica la fase d'ascolto mentre quella blu (scandita da un tic-tac d'orologio) la fase di risposta. Per interrompere brevemente lo svolgimento della partita selezionare/deselezionare la funzione di PAUSA premendo sulla barra di stato.

Ogni risposta esatta fa guadagnare 100 punti mentre una errata ne fa perdere 50. Due suoni diversi identificano l'esito della risposta. Se si risponde esattamente viene riprodotta l'intera clip audio della sigla corrispondente (circa 20 secondi) prima di passare alla domanda successiva. Sul display a destra vengono riassunte tutte le informazioni relative a livello, numero di domanda, punteggio, nonché messaggi vari (risposta corretta/errata, pausa, game over, ...). Ogni livello prevede 10 domande, per un totale di 5 Livelli. Attraverso il tasto "OPZIONI" è possibile modificare la durata dell'ascolto clip (fino ad un massimo di 20 secondi). Il numero di giocatori non è, invece, modificabile. Le istruzioni per giocare vengono mostrate a video premendo il tasto "COME". Per le informazioni su autore, release, sito web, ecc. basta premere il tasto "INFO".

## 10.2 SUGGERIMENTI E TRUCCHI

E' consigliabile non essere precipitosi nel rispondere al quiz. Attendere la fine della fase d'ascolto e solo dopo fare la propria scelta. Nessun bonus, infatti, viene assegnato per la velocità della risposta. Uno stratagemma che si può adottare, quando si conosce la sigla ma, al momento, la memoria non fa ricordare il nome del telefilm, consiste nell'utilizzare la funzione di pausa. Questo permette di guadagnare un po' di tempo per pensare.

## 10.3 LA REALIZZAZIONE (THE MAKING OF)

### □ LA STORIA

TV-Series Musical Quiz fa parte della trilogia di quiz musicali dedicata alle colonne sonore (cartoni animati, telefilm e film). Servendosi della stessa "Game Engine" sviluppata per "Toon Music Game" e di 50 Brani Mp3 tratti dal Progetto Geiar (è il duale del Progetto Prometeo, ma dedicato ai telefilm anni '70-'80) nasce, nel 1999, la prima versione del gioco. La prima release prevede brani mp3 completi ed in qualità medio-alta (Stereo, 44100 Hz, 128b Kbit/sec). Per consentirne il download tramite internet e, quindi, la sua distribuzione capillare, nonché per evitare problemi di copyright, le canzoni vengono rimpiazzate nelle versioni successive da clip di soli 20 secondi ed in bassa qualità.

Dal 1999 il videogame è stato prelevato dal sito web [www.evweb.it](http://www.evweb.it) più di 3000 volte.

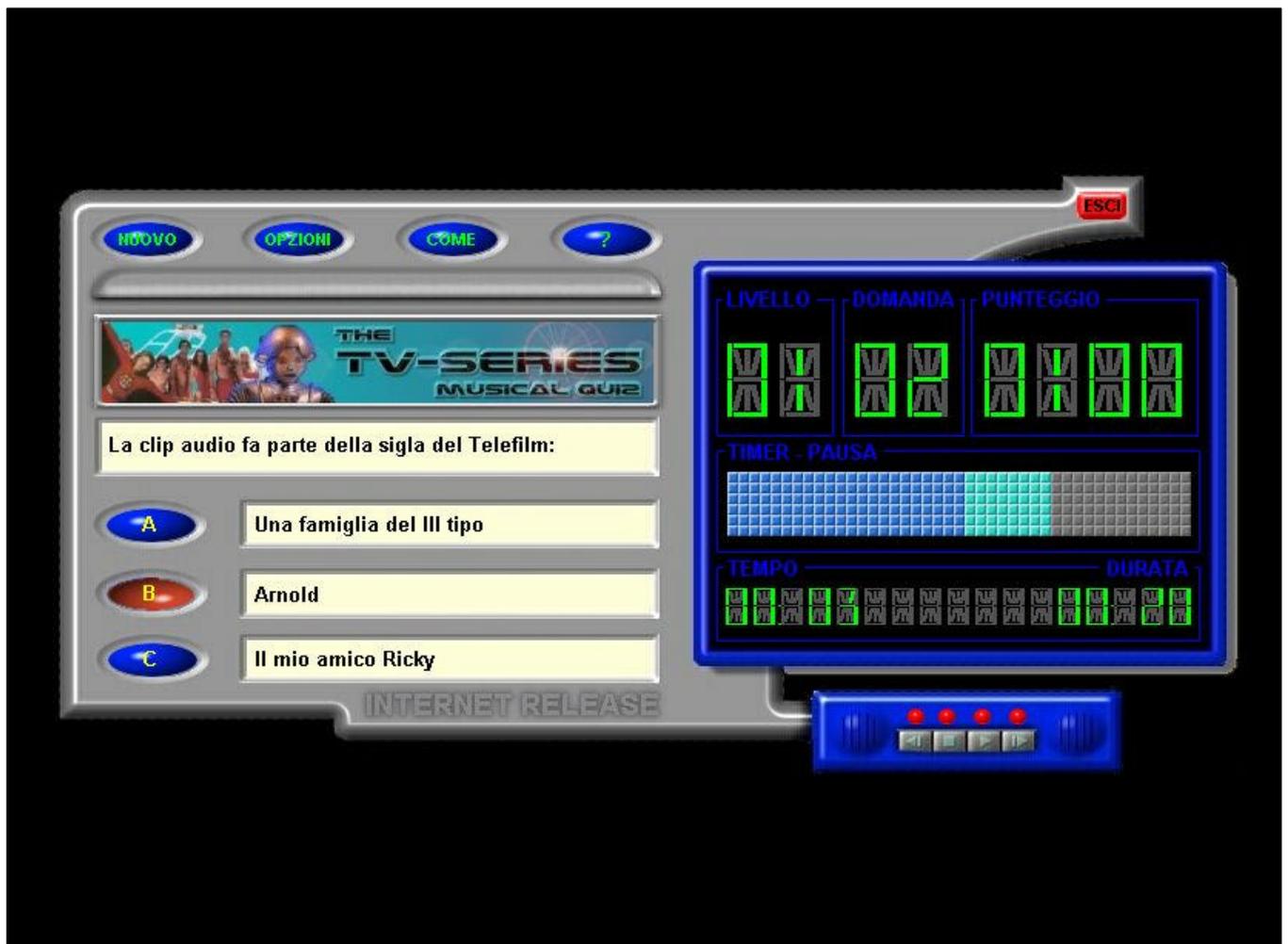


Fig. 10.1: interfaccia principale del videogioco.

## □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 10.8).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

## □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 10.2 un'immagine che decreta la vittoria della partita. In fig. 10.3 l'interfaccia principale del videogioco, con in primo piano la finestra delle opzioni configurabili. In fig. 10.4 come appaiono le informazioni su autore e release del gioco. In fig. 10.5 c'è l'Help On Line con le istruzioni per giocare. In fig. 10.6 e 10.7 le due schermate di game over in caso di sconfitta. Infine, in fig. 10.8 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



Fig. 10.2: un'immagine che mostra la vincita della partita.



Fig. 10.3: le opzioni ed i parametri configurabili del videogame.



Fig. 10.4: informazioni su autore e release del gioco.



Fig. 10.5: Help On Line - Istruzioni per giocare.



Fig. 10.6: schermata di game over..



Fig. 10.7: schermata di game over in caso di sconfitta.

## 10.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
PageTimer("1000", "Page 3")
MP3ListLoad("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
NDom=1
Click=0
riga=0
for i=1 to 22
  for j=1 to 6
    MatrixSet("Matrix[i,j]","3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet("Matrix[i,j]","2")
  next j
next i
for ElementoLista=1 to 50

```

```

    LoadText ("NumeDoman$", "<List>ElementoLista")
    NumeroDomanda[ElementoLista]=VAL(NumeDoman$)
next ElementoLista
Testo_NLivello$='01'
Testo_NDomanda$='01'
LoadText ("Text8", "Testo_NLivello$")
LoadText ("Text9", "Testo_NDomanda$")
MP3ListLoad ("<Embedded>\ToonA.txt")
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonB.txt")
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonC.txt")
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad ("<Embedded>\ToonE.txt")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
Show ("R_ascolta")
PlaySound ("File$")
RunScript ("Timer")

```

#### □ Blu Control Botton

```

pausa=0
Hide ("Pic_Box")
Show ("PIC_Views")
Hide ("Animations")
Hide ("R_pausa")
Hide ("AniGif")
Hide ("AniGif_Rack")
Risposta$='B'
RunScript ("Script")
Click=1
Show ("TB")
Show ("BloccaTasti")
Hide ("R_ascolta")

```

#### □ Tasto Pausa

```

if (pausa=0) then
    pausa=1
    Show ("R_pausa")
    Show ("HotSpot9")
    Hide ("R_ascolta")
    MP3Pause()
End

```

#### □ Script Timer

```

if (pausa=0) then
    riga=riga+1
    for k=1 to 6
        MatrixSet ("Matrix[44-riga,k]", "1")
    next k
    WavePlay ("tone", "")
    if (riga<>43) then
        ScriptTimer ("Timer", "tempo")
    end
    if (riga=43 & Click=0) then
        RunScript ("Script")
    end
    if (riga=20) then
        MP3Stop()
        Hide ("R_ascolta")
    end
end

```

```
end
```

#### ❑ Script Fine1

```
Show("Congratulazioni")
WaveStop()
WavePlay("Bravo", "")
```

#### ❑ Script Fine2

```
WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver")
Click=1
```

#### ❑ Script Fine3

```
WaveStop()
Hide("R_errata")
Hide("R_corretta")
Show("GameOver2")
Click=1
```

#### ❑ Script Mp3Control

```
Show("Mp3Data")
Hide("R_corretta")
pausa=0
Hide("R_pausa")
Hide("HotSpot9")
Show("HotSpot8")
pausa=0
ScriptTimer("NuovaDomanda", "20000")
```

#### ❑ Script NuovaDomanda

```
MP3Stop()
if (NDom=10) then
  NDom=0
  DomandeGiuste=0
  NumeroLivello=NumeroLivello+1
end
riga=0
Click=0
Risposta$=''
Hide("Mp3Data")
Hide("R_errata")
Hide("R_corretta")
Hide("R_temposcaduto")
Hide("Evide")
Hide("R_pausa")
Show("HotSpot14")
for i=1 to 22
  for j=1 to 6
    MatrixSet("Matrix[i,j]","3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet("Matrix[i,j]","2")
  next j
next i
NumeroDomandeTotali=NumeroDomandeTotali+1
NDom=NDom+1
if (NumeroLivello=1) then
  Testo_NLivello$='01'
```

```

    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=2) then
    Testo_NLivello$='02'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=3) then
    Testo_NLivello$='03'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=4) then
    Testo_NLivello$='04'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=5) then
    Testo_NLivello$='05'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=6) then
    Testo_NLivello$='06'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=7) then
    Testo_NLivello$='07'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=8) then
    Testo_NLivello$='08'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello=9) then
    Testo_NLivello$='09'
    LoadText ("Text8", "Testo_NLivello$")
end
if (NumeroLivello>=10) then
    Testo1$=CHAR(NumeroLivello)
    LoadText ("Text8", "Testo1$")
end
if (NDom=1) then
    Testo_NDomanda$='01'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=2) then
    Testo_NDomanda$='02'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=3) then
    Testo_NDomanda$='03'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=4) then
    Testo_NDomanda$='04'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=5) then
    Testo_NDomanda$='05'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=6) then
    Testo_NDomanda$='06'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=7) then
    Testo_NDomanda$='07'
    LoadText ("Text9", "Testo_NDomanda$")

```

```

end
if (NDom=8) then
    Testo_NDomanda$='08'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom=9) then
    Testo_NDomanda$='09'
    LoadText ("Text9", "Testo_NDomanda$")
end
if (NDom>=10) then
    Testo_NDomanda$=CHAR(NDom)
    LoadText ("Text9", "Testo_NDomanda$")
end
Hide ("BloccaTasti")
MP3ListLoad("<Embedded>\ToonA.txt")
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonB.txt")
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonC.txt")
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")
MP3ListLoad("<Embedded>\ToonE.txt")
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")
N=NumeroDomanda[NumeroDomandeTotali]
Nu$=CHAR(N)
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'
PlaySound ("File$")
RunScript ("Timer")
Show("R_ascolta")

```

#### □ Script Script

```

MP3Stop()
Hide("Info")
Hide("Come")
Hide("GameOver")
Hide("GameOver2")
Hide("Congratulazioni")
if (Click=0 & Risposta$='') then
    Show("R_temposcaduto")
    WaveStop()
    WavePlay("Errore", "")
    punteggio=punteggio-5
end
if (Risposta$=RispostaEsatta$) then
    Show("R_corretta")
    WaveStop()
    PlaySound ("File$")
    Hide("HotSpot14")
    punteggio=punteggio+100
    DomandeGiuste=DomandeGiuste+1
end
if (Risposta$<>RispostaEsatta$ & Risposta$<>'') then
    Show("R_errata")
    WaveStop()
    WavePlay("Errore", "")
    punteggio=punteggio-50
end
if (punteggio<=0) then
    punteggio=0
    Testo_Punteggio$='0000'
end
if (punteggio>=0 & punteggio<10) then
    Testo_Punteggio$='000'+CHAR(punteggio)
end
if (punteggio>=10 & punteggio<100) then

```

```

    Testo_Punteggio$='00'+CHAR(punteggio)
end
if (punteggio>=100 & punteggio<1000) then
    Testo_Punteggio$='0'+CHAR(punteggio)
end
if (punteggio>=1000) then
    Testo_Punteggio$=CHAR(punteggio)
end
LoadText ("Text10", "Testo_Punteggio$")
if (NumeroDomandeTotali<50 & Risposta$=RispostaEsatta$) then
    if (NDom<>10 | DomandeGiuste>=6) then
        ScriptTimer ("Mp3Control", "2000")
    end
end
if (NumeroDomandeTotali<50 & Risposta$<>RispostaEsatta$) then
    if (NDom<>10 | DomandeGiuste>=6) then
        ScriptTimer ("NuovaDomanda", "2000")
    end
end
if (punteggio=5000 & NumeroLivello=5) then
    MP3Stop()
    Hide ("R_corretta")
    ScriptTimer ("Fine1", "2000")
end
if (NDom=10 & DomandeGiuste<6 & NumeroLivello<>5) then
    MP3Stop()
    Hide ("R_corretta")
    Hide ("R_errata")
    Hide ("R_temposcaduto")
    ScriptTimer ("Fine2", "2000")
end
if (NDom=10 & punteggio>=2000 & punteggio<5000 & NumeroLivello=5) then
    ScriptTimer ("Fine3", "2000")
    Hide ("R_corretta")
    Hide ("R_errata")
    Hide ("R_temposcaduto")
    MP3Stop()
end
end

```

#### □ Tasto Nuovo

```

if (pausa=1) then
    pausa=0
    Hide ("R_pausa")
    Hide ("HotSpot9")
    Show ("HotSpot8")
end
Hide ("GameOver")
Hide ("GameOver2")
Hide ("Congratulazioni")
Hide ("Info")
Hide ("Come")
Hide ("BloccaTasti")
Hide ("Evide")
Hide ("Opzioni")
NumeroLivello=1
punteggio=0
DomandeGiuste=0
Testo_Punteggio$='0000'
LoadText ("Text10", "Testo_Punteggio$")
NDom=1
Click=0
Risposta$=''
Hide ("R_errata")
Hide ("R_corretta")

```

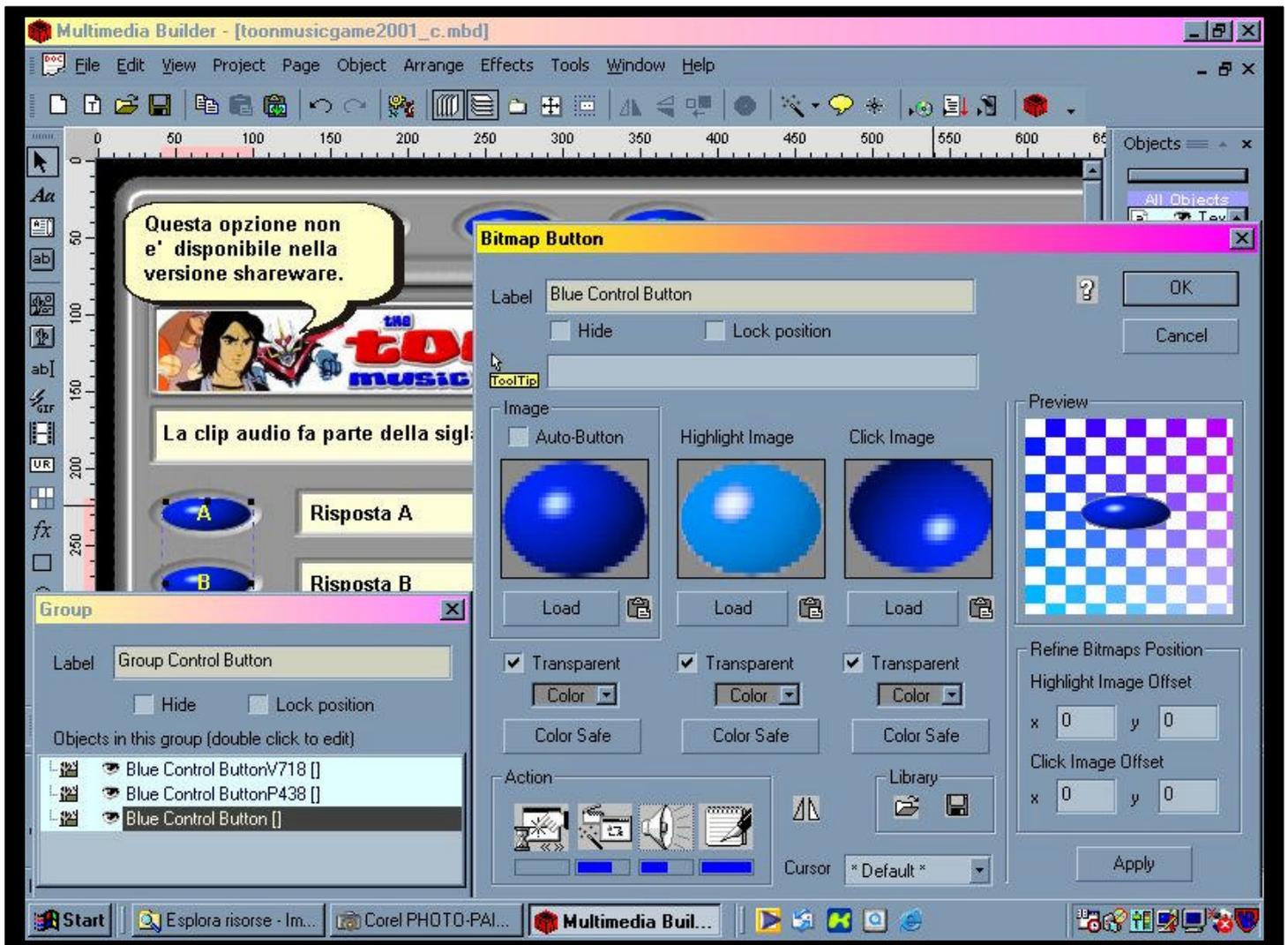


Fig. 10.8: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

```

Hide ("R_temposcaduto")
riga=0
MP3ListLoad ("<Embedded>\Array.num")
MP3ListRND()
NumeroLivello=1
punteggio=0
display=21
tempo=500
DomandeGiuste=0
NumeroDomandeTotali=1
for i=1 to 22
  for j=1 to 6
    MatrixSet ("Matrix[i,j]", "3")
  next j
next i
for i=23 to 44
  for j=1 to 6
    MatrixSet ("Matrix[i,j]", "2")
  next j
next i
for ElementoLista=1 to 50
  LoadText ("NumeDoman$", "<List>ElementoLista")
  NumeroDomanda [ElementoLista]=VAL (NumeDoman$)
next ElementoLista
Testo_NLivello$='01'

```

```
Testo_NDomanda$='01'  
LoadText ("Text8", "Testo_NLivello$")  
LoadText ("Text9", "Testo_NDomanda$")  
MP3ListLoad ("<Embedded>\ToonA.txt")  
LoadText ("RispostaA", "<List>NumeroDomanda[NumeroDomandeTotali]")  
MP3ListLoad ("<Embedded>\ToonB.txt")  
LoadText ("RispostaB", "<List>NumeroDomanda[NumeroDomandeTotali]")  
MP3ListLoad ("<Embedded>\ToonC.txt")  
LoadText ("RispostaC", "<List>NumeroDomanda[NumeroDomandeTotali]")  
MP3ListLoad ("<Embedded>\ToonE.txt")  
LoadText ("RispostaEsatta$", "<List>NumeroDomanda[NumeroDomandeTotali]")  
N=NumeroDomanda[NumeroDomandeTotali]  
Nu$=CHAR(N)  
File$='<SrcDir>\Audio\\'+Nu$+'.mp3'  
PlaySound ("File$")  
RunScript ("Timer")  
Show ("R_ascolta")
```

## 11. STAR TREK VOYAGER SHUTTLE MEMORY

- ❑ Versione 1.0 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2002 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 4,60 Mb

### 11.1 REGOLE DEL GIOCO

Vogare Shuttle Memory è una rivisitazione ispirata al mondo di Star Trek del famoso gioco Memory. In particolare il tema è rappresentato dallo shuttle dell'ammiraglio Janeway, apparso nell'ultimo episodio della serie Tv Star Trek Voyager. I simboli delle pedine riprendono loghi, astronavi ed oggettistica Trek. Lo schermo è distinto in diverse zone. In basso a destra c'è il tabellone di gioco principale, con le caselle da scoprire. A sinistra c'è il menu con le varie opzioni di gioco. In alto a destra una vista laterale dello shuttle, a puro scopo decorativo. Per giocare una nuova partita bisogna premere il tasto "NUOVO" e successivamente selezionare "1 GIOCATORE" oppure "2 GIOCATORI", indicandone i nomi nelle opportune caselle di testo. Una delle peculiarità di questo gioco, infatti, è che si può gareggiare con un amico oppure in solitario. A turno, se si gioca in due, si devono scoprire coppie di caselle sul tabellone, individuando quello con la stessa immagine. Il turno di gioco viene mostrato dalle targhette, che identificano il giocatore, sopra il tabellone ed illuminate, se attive. Vince chi individua più coppie.

Per accedere alla finestra con le informazioni su autore e release occorre servirsi del pulsante "INFO", mentre, per le istruzioni di gioco, del tasto "HELP". Il tasto "ESC", come di consueto, consente di chiudere l'applicazione e ritornare al sistema operativo Windows. Il tasto speciale "SHUTTLE" consente di accedere ad un'area con le immagini, le animazioni ed i dati tecnici dello shuttle dell'ammiraglio Janeway.

### 11.2 SUGGERIMENTI E TRUCCHI

Trattandosi del classico gioco "Memory", non ci sono particolari accorgimenti per vincere, se non allenare la memoria e informarsi sul mondo Trek...

### 11.3 LA REALIZZAZIONE (THE MAKING OF)

#### ❑ LA STORIA

Dopo l'enorme successo del videogame "Star Trek Giovani Ufficiali", nel secondo semestre del 2002 viene presa la decisione di realizzare un nuovo gioco dedicato alla famosa saga televisiva. L'idea di una nuova avventura grafica è stata subito accantonata, a causa degli eccessivi costi di progettazione e sviluppo. Così si è optato per un gioco classico, con un

restyling Trek particolarmente curato nei minimi dettagli. La nuova grafica ispirata al sistema LCARS delle astronavi della Flotta Stellare risulta ben congegnata e con un notevole impatto visivo. La scelta dell'immagine del dorso dello shuttle dell'ammiraglio Janeway della serie "Star Trek Voyager" è avvenuta quasi per caso, navigando tra i siti d'appassionati. Nonostante il progetto iniziale prevedesse di riadattare la "Game Engine" sviluppata per il videogame "Caccia all'Anello di Sauron", la necessità di permettere sfide a uno o due giocatori ha costretto ad una programmazione ex novo. Il videogioco non è mai stato reso disponibile gratuitamente sul sito internet [www.evweb.it](http://www.evweb.it) ma è stato inserito, come bonus game, nel cd-rom "Gameisland".

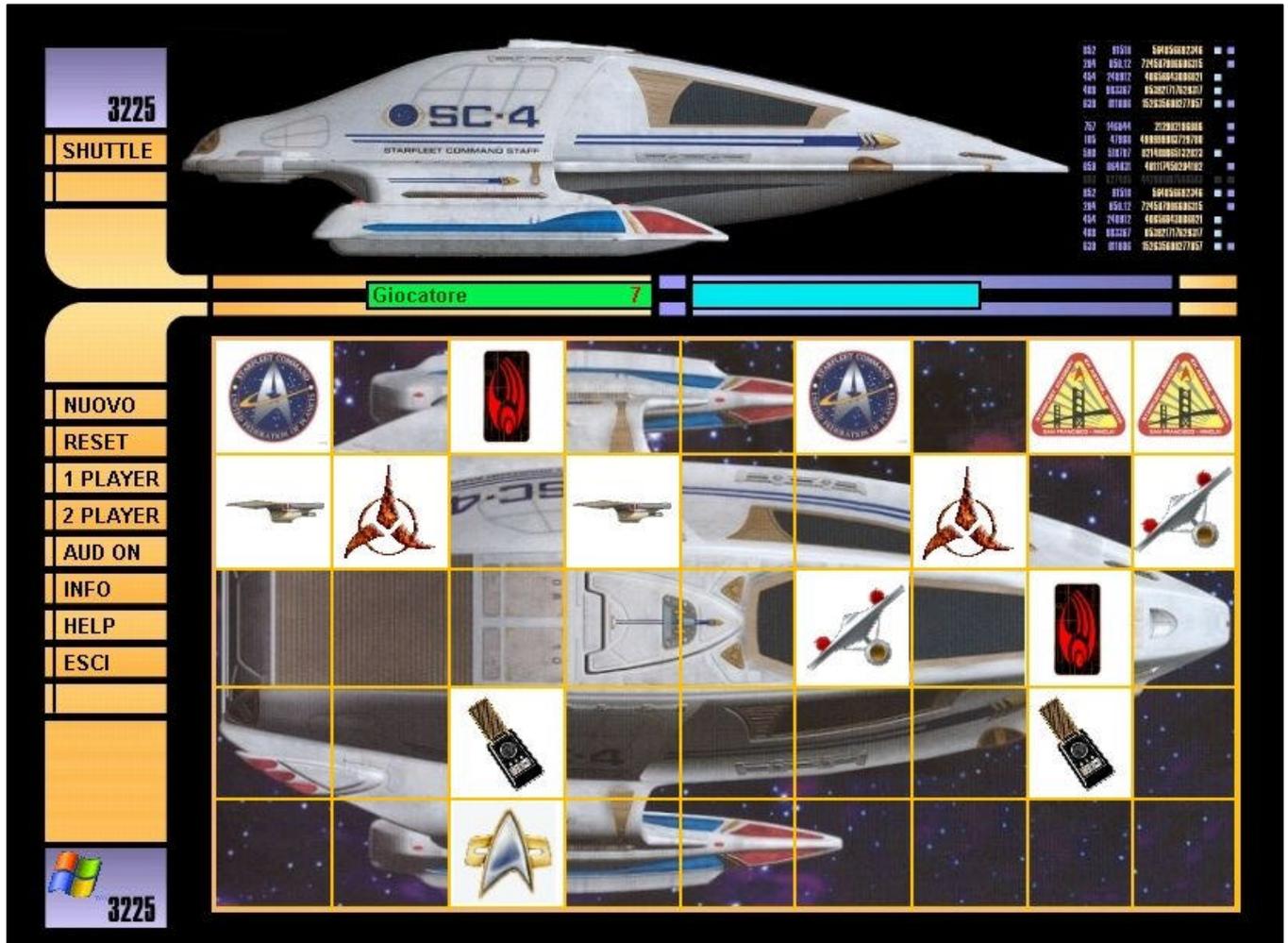


Fig. 11.2: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 11.8).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. I fig. 11.2 una schermata dell'interfaccia principale del videogame. Nelle figg. 11.3 e 11.4 ci sono, rispettivamente, la finestra con le informazioni su autore e release del gioco e l'Help On Line. In fig. 11.5: menu inserimento dati dei giocatori. Nelle figg. 11.6 e 11.7 immagini tratte dal database della navetta. Infine, in fig. 11.8 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

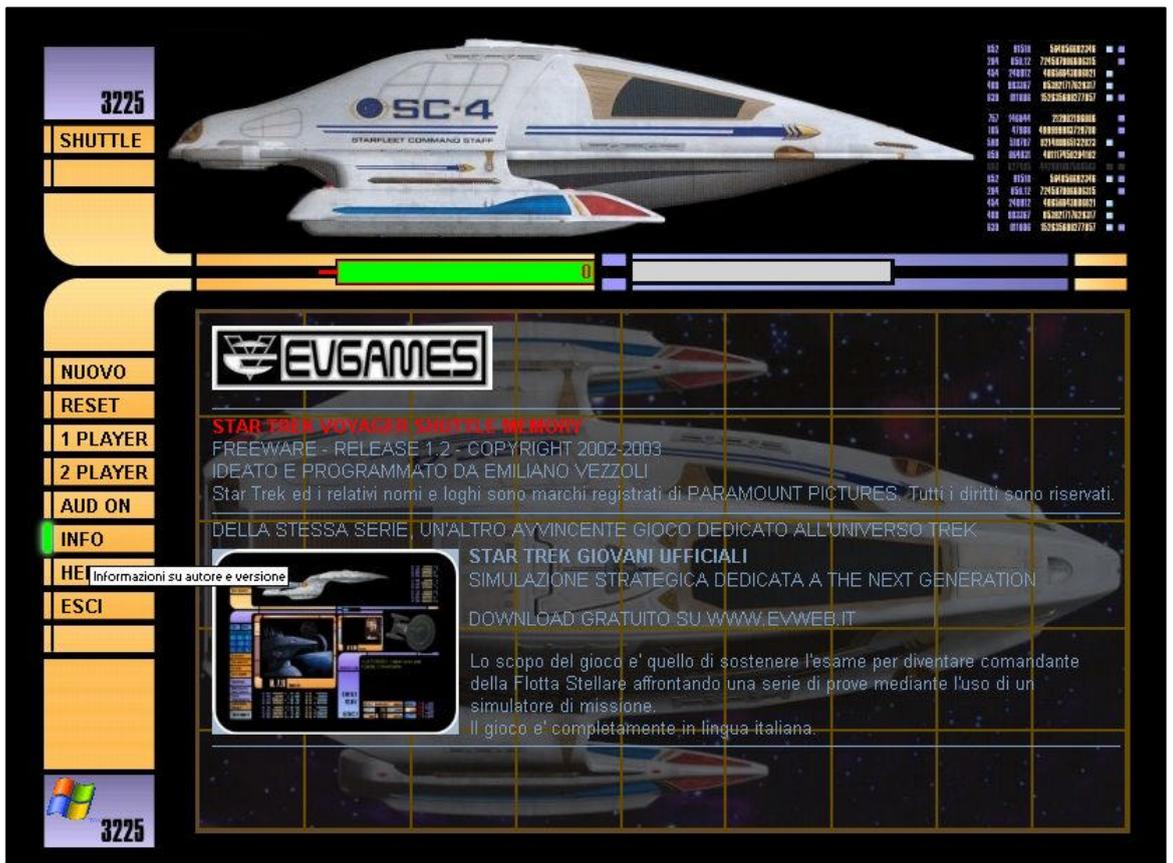


Fig. 11.3: informazioni su autore e release del gioco.

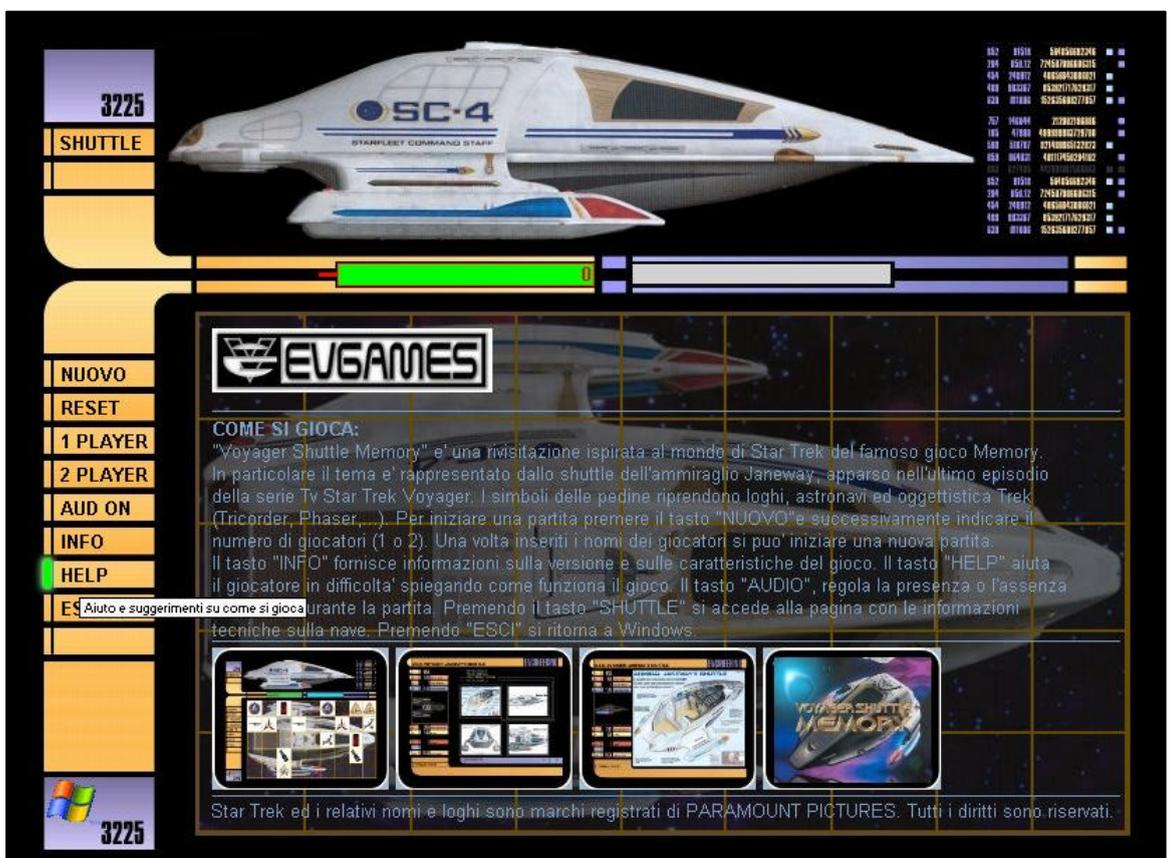


Fig. 11.4: Help On Line - Istruzioni per giocare.

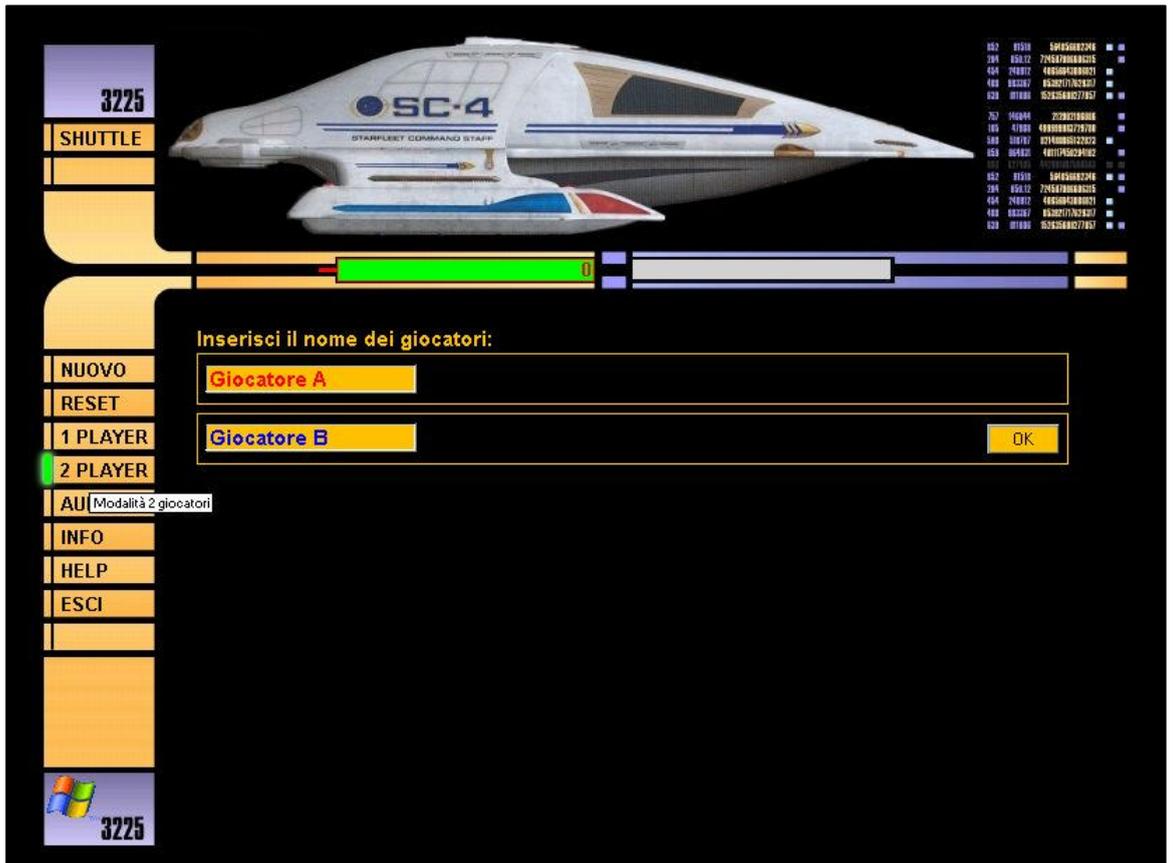


Fig. 11.5: menu inserimento dati per due giocatori.

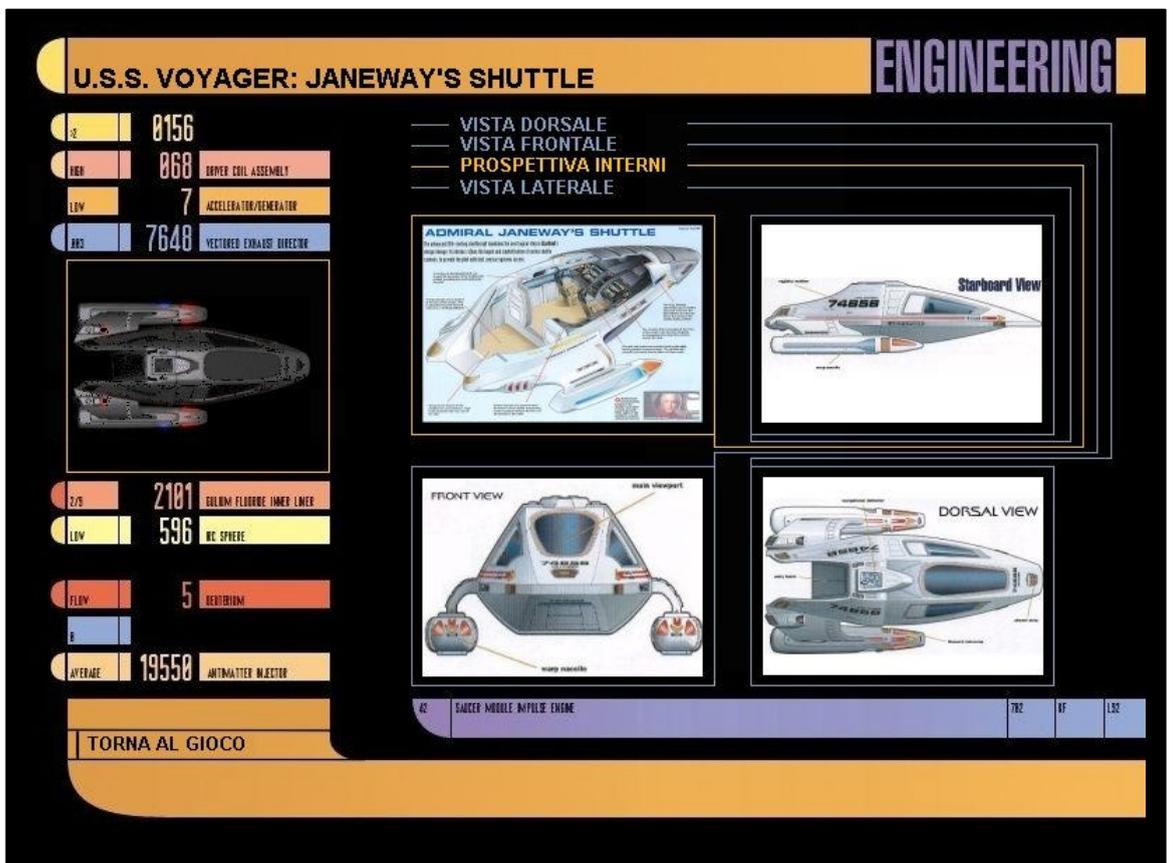


Fig. 11.6: menu principale dedicato ai dati tecnici dello shuttle.

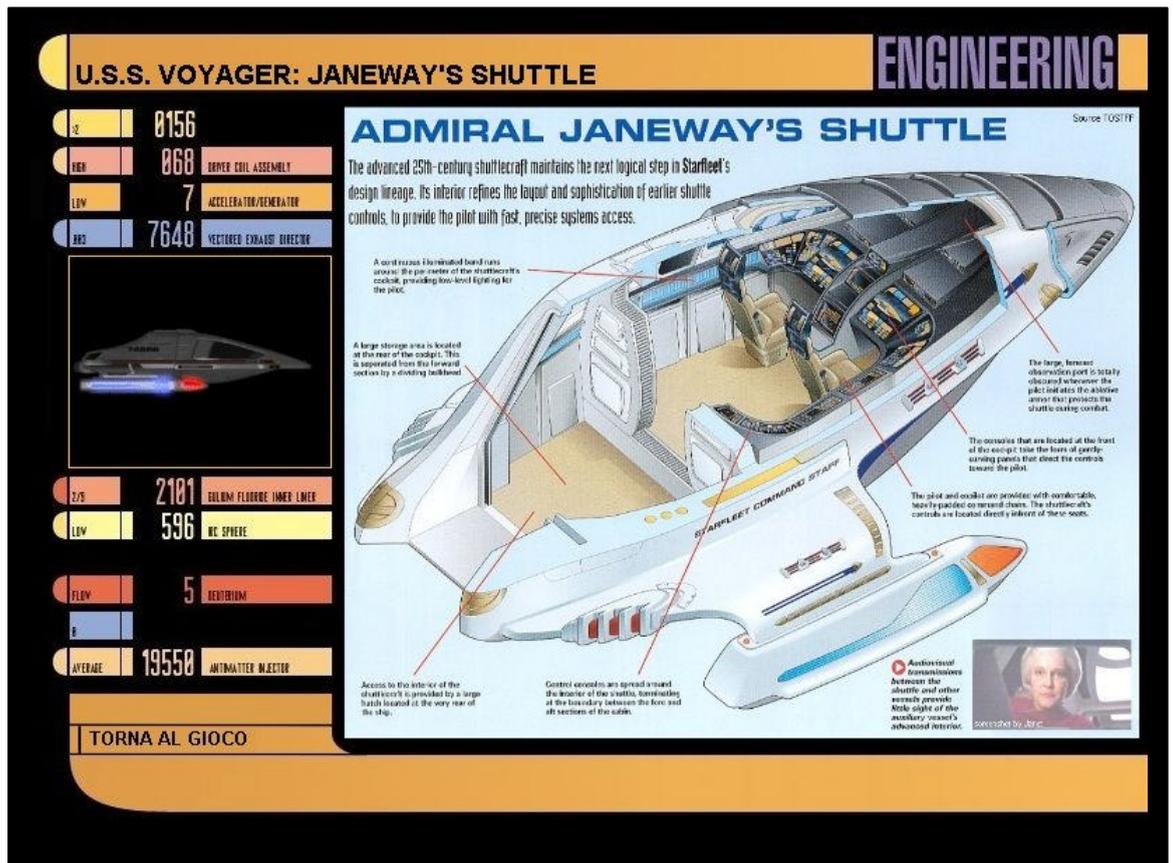


Fig. 11.7: uno screenshot tratto dall'archivio dei dati tecnici dello shuttle.

### 11.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

#### □ Main Page

```
PageTimer ("2000", "Page 1")
Audio = 1
InstallFont (<Embedded>\Arial.ttf")
InstallFont (<Embedded>\FinalFrontier.TTF")
InstallFont (<Embedded>\Trekbats.ttf")
PlaySound (<SrcDir>\Musica.mp3")
```

#### □ Tasto ScriptBTNUm

```
NJogadores=1
Hide ("GroupAjuda")
Hide ("GroupDois")
Show ("GroupPainel")
Show ("GroupUnico")
```

#### □ Tasto ScriptBTNDois

```
NJogador=2
Hide ("GroupUnico")
Hide ("GroupAjuda")
Show ("GroupDois")
Show ("GroupPainel")
```

#### □ Script ScriptBTNAjuda

```
Hide ("GroupUnico")
Hide ("GroupDois")
Show ("GroupPainel")
```

```
Show("GroupAjuda")
```

#### □ Script ScriptTestar

```

if (TA=1) then
  Show("Obj3$")
  Val1 = NumeroCasella[TastoPremuto]
  Obj1$=NA$
  Hide("Obj1$")
end
if (TA=2) then
  Val2 = NumeroCasella[TastoPremuto]
  Obj2$=NA$
  Hide("Obj2$")
end
if (Val1=Val2) then
  if (TA=2) then
    Hide("Obj1$")
    Hide("Obj2$")
    if (Jogador=1) then
      PontosA=PontosA+1
      ScoreA=ScoreA+1
    end
    if (Jogador=2) then
      PontosB=PontosB+1
      ScoreB=ScoreB+1
    end
    TA=0
    Obj3$=' '
    ** RunScript("ScriptCerto")
    if (Audio=0) then
      WavePlay("Bravo", "")
    end
  end
end
if (Val1<>Val2) then
  if (TA=2) then
    Show("Obj1$")
    **Show("Obj2$")
    Obj3$=Obj2$
    TA=0
    CambioGiocatore = 0
    if (NJogador=2 & Jogador=1 & CambioGiocatore=0) then
      Jogador = 2
      Show("Highlight2")
      Hide("Highlight1")
      CambioGiocatore=1
    end
    if (NJogador=2 & Jogador=2 & CambioGiocatore=0) then
      Jogador = 1
      Show("Highlight1")
      Hide("Highlight2")
      CambioGiocatore=1
    end
    ** RunScript("ScriptErro")
    if (Audio=0) then
      WavePlay("Errore", "")
    end
  end
end
DisplayValue("PontosA", "PontosA")
DisplayValue("PontosB", "PontosB")
if (ScoreA=22) then
  MP3Open("<SrcDir>Musica.mp3")
  Show("HotSpot9")

```

```

    MP3Play()
    ScoreA=0
end
if (ScoreB=22) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
end
if (ScoreA=20 & ScoreB=2) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=18 & ScoreB=4) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=16 & ScoreB=6) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=14 & ScoreB=8) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=12 & ScoreB=10) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=10 & ScoreB=12) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=8 & ScoreB=14) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
if (ScoreA=6 & ScoreB=16) then
    Show("HotSpot9")
    MP3Open("<SrcDir>Musica.mp3")
    MP3Play()
    ScoreB=0
    ScoreA=0
end
end

```

```

if (ScoreA=4 & ScoreB=18) then
  Show("HotSpot9")
  MP3Open("<SrcDir>Musica.mp3")
  MP3Play()
  ScoreB=0
  ScoreA=0
end
if (ScoreA=2 & ScoreB=20) then
  Show("HotSpot9")
  MP3Open("<SrcDir>Musica.mp3")
  MP3Play()
  ScoreB=0
  ScoreA=0
end
if (ScoreA=11 & ScoreB=11) then
  Show("HotSpot9")
  WavePlay("Bravo", "")
  Show("Bonus")
  Pause("3000")
  Hide("Bonus")
  Show("GroupTampa")
  ScoreB=0
  ScoreA=0
  Jogador=1
  NJogador=2
  PontosA=0
  TotalPontosA=0
  PontosB=0
  TotalPontosB=0
  RefJogador=1
  if (Jogador=1) then
    Hide("GroupB")
  end
  if (Jogador=2) then
    Show("GroupB")
  end
  if (NJogador=1) then
    LoadText("PrimeiroJogador", "JogadorUN$")
  end
  if (NJogador=2) then
    Show("GroupB")
    LoadText("PrimeiroJogador", "JogadorA$")
    LoadText("SegundoJogador", "JogadorB$")
  end
  Hide("Highlight2")
  RunScript("Script")
end

```

□ **Script Rx (matrice di 44 caselle in GroupTampa)**

```

** T = Tentativi
** V = Valore del campo
** N = Nome dell'oggetto
TA=TA+1
VA$=fx$
NA$='R1'
TastoPremuto = x
RunScript("ScriptTestar")
if (TA>2) then
  TA=0
end

```

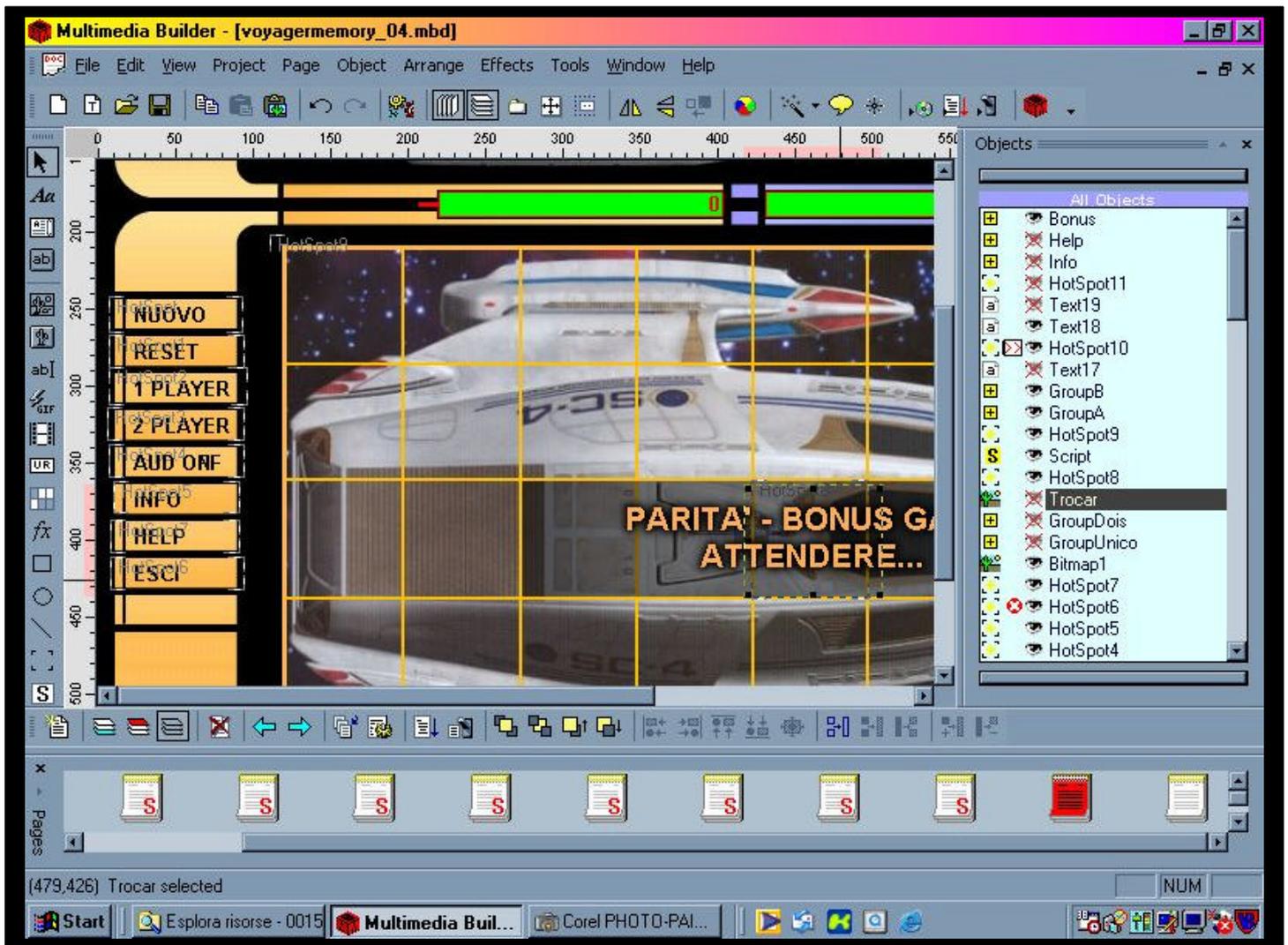


Fig. 11.8: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

❑ **Script Script**

```

ElementoLista = 0
MP3ListLoad ("<Embedded>\Array.num")
MP3ListRND()
for ElementoLista=1 to 44
  LoadText ("NumeElemento$", "<List>ElementoLista")
  NumeroCasella[ElementoLista]=VAL(NumeElemento$)
  NumeImmagine$ = 'T' + CHAR(ElementoLista)
  TipoImmagine$ = '<Embedded>A' + NumeElemento$ + '.jpg'
  ReplaceImage ("NumeImmagine$", "TipoImmagine$")
next ElementoLista
if (NJogador = 2) then
  Show("Highlight1")
end
    
```

❑ **Script TastoNuovo**

```

Show ("GroupTampa")
Show ("HotSpot9")
Hide ("Help")
Hide ("Info")
if (Audio=0) then
  WavePlay ("Beep", "")
end
    
```

### ❑ Script TastoReset

```
Show("GroupTampa")
Hide("Help")
Hide("Info")
RunScript("Script")
if (Audio=0) then
  WavePlay("Beep", "")
end
```

### ❑ Script Tasto 1 Giocatore

```
Hide("HotSpot9")
Hide("Help")
Hide("Info")
RunScript("ScriptBTNUM")
if (Audio=0) then
  WavePlay("Beep", "")
end
```

### ❑ Script Tasto 2 Giocatori

```
Hide("Help")
Hide("Info")
Hide("HotSpot9")
RunScript("ScriptBTNDois")
if (Audio=0) then
  WavePlay("Beep", "")
End
```

### ❑ Script HotSpot11

```
MP3Open("<SrcDir>\Musica.mp3")
MP3Play()
Show("HotSpot4")
Show("Text5")
Hide("Text18")
Hide("HotSpot11")
Hide("Help")
Hide("Info")
Audio = 1
```

## 12. LADER – LO SCIACALLO

- ❑ Versione 2.1 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2001 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti di riproduzione e memorizzazione elettronica sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 946 Kb

### 12.1 REGOLE DEL GIOCO

Lo sciacallo è un gioco di strategia per due giocatori ispirato al famosissimo "Forza 4". L'obiettivo del giocatore è di allineare sul tabellone 4 pedine dello stesso colore in orizzontale, verticale o diagonale. Periodicamente però, una delle postazioni viene liberata casualmente, rimuovendo l'eventuale pedina posizionata dal giocatore.

Questo stratagemma rende il gioco più dinamico e ricco di suspense, costringendo i giocatori a pianificare un'adeguata strategia di gioco. Se nessuno dei due giocatori riesce a raggiungere l'obiettivo, vince il primo che raggiunge le 50 mosse. Per mossa si intende l'allineamento orizzontale, verticale od obliquo di 3 pedine dello stesso colore. Il display posizionato sopra il tasto start indica il turno di gioco, cioè quale giocatore deve fare la propria mossa.

Premere il tasto "START" per iniziare una nuova partita, il tasto "AU OFF" per disabilitare l'audio ("AU ON" per abilitarlo), "INFO" per avere informazioni sulla versione del gioco e "EXIT" per uscire e tornare a Windows.

### 12.2 SUGGERIMENTI E TRUCCHI

Non esiste alcun trucco specifico per vincere in questo gioco.

### 12.3 LA REALIZZAZIONE (THE MAKING OF)

#### ❑ LA STORIA

L'idea iniziale da cui è partorito questo gioco prevedeva una realizzazione sia hardware che software, cioè doveva essere un vero e proprio gioco elettronico tascabile. I costi e le difficoltà d'implementazione hanno poi costretto al solo sviluppo software.

La prima versione, del 1999, è chiamata "Thief For Joke" (cioè "Ladro Per Scherzo"... ) e la grafica è piuttosto mediocre, tanto che la sua diffusione tramite internet rimane insignificante (vedi Fig. 12.2).

Era stato previsto, poi, di sviluppare un'intelligenza artificiale per permettere al giocatore sfide in solitario contro il "computer". I costi di programmazione, confrontati con lo scarso successo di pubblico, hanno costretto a numerose revisioni al progetto originale, abbandonando anche questa idea.

In seguito nasce nel 2000 Lader (dal dialetto bresciano "Ladro") ancora ad evidenziare il "furto" di pedine dalle postazioni sul tabellone di gioco. Viene realizzata una nuova ed accattivante veste grafica. A causa del nome poco gradito, nel 2001 viene rilasciata l'ultima versione, la 2.1, chiamata "Lo Sciacallo" ma, a causa dell'impossibilità di giocare da soli, questo videogame riscuote poco successo. Infatti, dal 1999 il videogame è stato prelevato dal sito web [www.evweb.it](http://www.evweb.it) poco più di 300 volte.



Fig. 12.1: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 12.9).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 12.2 un'immagine della prima versione del videogame. In fig. 12.3 l'interfaccia principale del videogioco. In fig. 12.4 come appaiono le informazioni su autore e release del gioco. In fig. 12.5: Schermata di game over in caso di vittoria. Infine, in fig. 12.6 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



Fig. 12.2: un'immagine della prima versione del videogame, realizzata nel 1999.



Fig. 12.3: schermata principale del gioco.



Fig. 12.4: informazioni su autore e release del gioco.



Fig. 12.5: schermata di game over in caso di vittoria.

## 12.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
PageTimer("1000", "Page 1")
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
MatrixSet("Matrix[0,0]", "1")
MatrixSet("Matrix1[0,0]", "3")
MatrixSet("Matrix3[0,0]", "3")
Testo_MosseG1$='00'
Testo_MosseG2$='00'
LoadText("MosseG1", "Testo_MosseG1$")
LoadText("MosseG2", "Testo_MosseG2$")
```

### □ Script Matrix2 – Up

```
MatrixSet("Matrix2[MXCOL, MXROW]", "1")
MatrixSet("MatrixA01[0,0]", "1")
MatrixSet("MatrixA02[0,0]", "1")
MatrixSet("MatrixA03[0,0]", "1")
MatrixSet("MatrixA04[0,0]", "1")
MatrixSet("MatrixA05[0,0]", "1")
MatrixSet("MatrixA06[0,0]", "1")
MatrixSet("MatrixA07[0,0]", "1")
MatrixSet("MatrixA08[0,0]", "1")
MatrixSet("MatrixA09[0,0]", "1")
MatrixSet("MatrixA10[0,0]", "1")
MatrixSet("MatrixA11[0,0]", "1")
```

```

MatrixSet("MatrixA12[0,0]", "1")
MatrixSet("MatrixA13[0,0]", "1")
MatrixSet("MatrixA14[0,0]", "1")
MatrixSet("MatrixA15[0,0]", "1")
MatrixSet("MatrixA16[0,0]", "1")
Hide("G1Vinto")
Hide("G2Vinto")
MosseG1=0
MosseG2=0
AuOn=1
ControlloMosse=0
Pedina=0
Testo_MosseG1$='00'
Testo_MosseG2$='00'
Show("Giocatore1")
Hide("Giocatore2")
Hide("HotSpot2")
LoadText("MosseG1", "Testo_MosseG1$")
LoadText("MosseG2", "Testo_MosseG2$")
Player1=1
Player2=0
bot1=0
.
.
botal6=0
botb1=0
.
.
botb16=0
CMos1 = 1
.
.
CMos48 = 1
Hide("info")

```

❑ **Script Matrix2 – Down**

```
MatrixSet("Matrix2[MXCOL, MXROW]", "2")
```

❑ **Script Script**

```

if ( Player1=1 & bot1=1) then
  Show("Matrix01")
  MatrixSet("MatrixA01[0,0]", "3")
  MatrixSet("Matrix01[1,1]", "1")
  botal=1
end
if ( Player1=1 & bot2=1) then
  Show("Matrix02")
  MatrixSet("MatrixA02[0,0]", "3")
  MatrixSet("Matrix02[1,1]", "1")
  bota2=1
end
if ( Player1=1 & bot3=1) then
  Show("Matrix03")
  MatrixSet("MatrixA03[0,0]", "3")
  MatrixSet("Matrix03[1,1]", "1")
  bota3=1
end
if ( Player1=1 & bot4=1) then
  Show("Matrix04")
  MatrixSet("MatrixA04[0,0]", "3")
  MatrixSet("Matrix04[1,1]", "1")
  bota4=1
end
if ( Player1=1 & bot5=1) then

```

```

Show("Matrix05")
MatrixSet("MatrixA05[0,0]", "3")
MatrixSet("Matrix05[1,1]", "1")
bota5=1
end
if ( Player1=1 & bot6=1) then
  Show("Matrix06")
  MatrixSet("MatrixA06[0,0]", "3")
  MatrixSet("Matrix06[1,1]", "1")
  bota6=1
end
if ( Player1=1 & bot7=1) then
  Show("Matrix07")
  MatrixSet("MatrixA07[0,0]", "3")
  MatrixSet("Matrix07[1,1]", "1")
  bota7=1
end
if ( Player1=1 & bot8=1) then
  Show("Matrix08")
  MatrixSet("MatrixA08[0,0]", "3")
  MatrixSet("Matrix08[1,1]", "1")
  bota8=1
end
if ( Player1=1 & bot9=1) then
  Show("Matrix09")
  MatrixSet("MatrixA09[0,0]", "3")
  MatrixSet("Matrix09[1,1]", "1")
  bota9=1
end
if ( Player1=1 & bot10=1) then
  Show("Matrix10")
  MatrixSet("MatrixA10[0,0]", "3")
  MatrixSet("Matrix10[1,1]", "1")
  bota10=1
end
if ( Player1=1 & bot11=1) then
  Show("Matrix11")
  MatrixSet("MatrixA11[0,0]", "3")
  MatrixSet("Matrix11[1,1]", "1")
  bota11=1
end
if ( Player1=1 & bot12=1) then
  Show("Matrix12")
  MatrixSet("MatrixA12[0,0]", "3")
  MatrixSet("Matrix12[1,1]", "1")
  bota12=1
end
if ( Player1=1 & bot13=1) then
  Show("Matrix13")
  MatrixSet("MatrixA13[0,0]", "3")
  MatrixSet("Matrix13[1,1]", "1")
  bota13=1
end
if ( Player1=1 & bot14=1) then
  Show("Matrix14")
  MatrixSet("MatrixA14[0,0]", "3")
  MatrixSet("Matrix14[1,1]", "1")
  bota14=1
end
if ( Player1=1 & bot15=1) then
  Show("Matrix15")
  MatrixSet("MatrixA15[0,0]", "3")
  MatrixSet("Matrix15[1,1]", "1")
  bota15=1
end
end

```

```

if ( Player1=1 & bot16=1) then
  Show("Matrix16")
  MatrixSet("MatrixA16[0,0]","3")
  MatrixSet("Matrix16[1,1]","1")
  bot16=1
end
if (MosseG1<10) then
  Testo_MosseG1$ = '0' + CHAR(MosseG1)
end
if (MosseG1>10) then
  Testo_MosseG1$ = CHAR(MosseG1)
end
Player1=0
Player2=1
bot1=0
.
.
bot16=0
LoadText("MosseG1","Testo_MosseG1$")
LoadText("MosseG2","Testo_MosseG2$")
Hide("Giocatore1")
Show("Giocatore2")
c=RND(15)+1
if (Pedina=c) then
  c=RND(15)+1
  if (Pedina=c) then
    c=RND(15)+1
  end
end
if (c=1) then
  Show("MatrixB1")
end
if (c=2) then
  Show("MatrixB2")
end
if (c=3) then
  Show("MatrixB3")
end
if (c=4) then
  Show("MatrixB4")
end
if (c=5) then
  Show("MatrixB5")
end
if (c=6) then
  Show("MatrixB6")
end
if (c=7) then
  Show("MatrixB7")
end
if (c=8) then
  Show("MatrixB8")
end
if (c=9) then
  Show("MatrixB9")
end
if (c=10) then
  Show("MatrixB10")
end
if (c=11) then
  Show("MatrixB11")
end
if (c=12) then
  Show("MatrixB12")
end
end

```

```

if (c=13) then
  Show("MatrixB13")
end
if (c=14) then
  Show("MatrixB14")
end
if (c=15) then
  Show("MatrixB15")
end
if (c=16) then
  Show("MatrixB16")
end
RunScript("Script3")

```

### □ Script Script3

```

Show("Analisi")
if ( botal=1 & bota2=1 & bota3=1 & bota4=1) then
  Show("GlVinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( bota5=1 & bota6=1 & bota7=1 & bota8=1) then
  Show("GlVinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( bota9=1 & botal0=1 & botal1=1 & botal2=1) then
  Show("GlVinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botal3=1 & botal4=1 & botal5=1 & botal6=1) then
  Show("GlVinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botal=1 & bota5=1 & bota9=1 & botal3=1) then
  Show("GlVinto")
  c=0
  Hide("Analisi")

```

```

Hide("Giocatore1")
Hide("Giocatore2")
Hide("GroupMatriciB")
if (AuOn=1) then
  MidiPlay("GameOver", "[LOOP]")
end
end
if ( bota2=1 & bota6=1 & bota10=1 & bota14=1) then
  Show("G1Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( bota3=1 & bota7=1 & bota11=1 & bota15=1) then
  Show("G1Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( bota4=1 & bota8=1 & bota12=1 & bota16=1) then
  Show("G1Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( bota1=1 & bota6=1 & bota11=1 & bota16=1) then
  Show("G1Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( bota13=1 & bota10=1 & bota7=1 & bota4=1) then
  Show("G1Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( botb1=1 & botb2=1 & botb=1 & botb4=1) then
  Show("G2Vinto")
  c=0

```

```

Hide("Analisi")
Hide("Giocatore1")
Hide("Giocatore2")
Hide("GroupMatriciB")
if (AuOn=1) then
  MidiPlay("GameOver", "[LOOP]")
end
end
if ( botb5=1 & botb6=1 & botb7=1 & botb8=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botb9=1 & botb10=1 & botb11=1 & botb12=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botb13=1 & botb14=1 & botb15=1 & botb16=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botb1=1 & botb5=1 & botb9=1 & botb13=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botb2=1 & botb6=1 & botb10=1 & botb14=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
if ( botb3=1 & botb7=1 & botb11=1 & botb15=1) then
  Show("G2Vinto")

```

```

c=0
Hide("Analisi")
Hide("Giocatore1")
Hide("Giocatore2")
Hide("GroupMatriciB")
if (AuOn=1) then
  MidiPlay("GameOver", "[LOOP]")
end
end
if ( botb4=1 & botb8=1 & botb12=1 & botb16=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( botb1=1 & botb6=1 & botb11=1 & botb16=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("GroupMatriciB")
  Hide("Giocatore2")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if ( botb13=1 & botb10=1 & botb7=1 & botb4=1) then
  Show("G2Vinto")
  c=0
  Hide("Analisi")
  Hide("Giocatore1")
  Hide("Giocatore2")
  Hide("GroupMatriciB")
  if (AuOn=1) then
    MidiPlay("GameOver", "[LOOP]")
  end
end
end
if (bota1 = 1 & bota2 = 1 & bota3 = 1 & CMos1 = 1) then
  MosseG1 = MosseG1 + 1
  CMos1 = 0
end
if (bota1 = 1 & bota6 = 1 & bota11 = 1 & CMos2 = 1) then
  MosseG1 = MosseG1 + 1
  CMos2 = 0
end
if (bota1 = 1 & bota5 = 1 & bota9 = 1 & CMos3 = 1) then
  MosseG1 = MosseG1 + 1
  CMos3 = 0
end
if (bota2 = 1 & bota6 = 1 & bota10 = 1 & CMos4 = 1) then
  MosseG1 = MosseG1 + 1
  CMos4 = 0
end
if (bota2 = 1 & bota7 = 1 & bota12 = 1 & CMos5 = 1) then
  MosseG1 = MosseG1 + 1
  CMos5 = 0
end
if (bota3 = 1 & bota6 = 1 & bota9 = 1 & CMos6 = 1) then
  MosseG1 = MosseG1 + 1
  CMos6 = 0
end

```

```

end
if (bota3 = 1 & bota7 = 1 & bota11 = 1 & CMos7 = 1) then
  MosseG1 = MosseG1 + 1
  CMos7 = 0
end
if (bota2 = 1 & bota3 = 1 & bota4 = 1 & CMos8 = 1) then
  MosseG1 = MosseG1 + 1
  CMos8 = 0
end
if (bota4 = 1 & bota7 = 1 & bota10 = 1 & CMos9 = 1) then
  MosseG1 = MosseG1 + 1
  CMos9 = 0
end
if (bota4 = 1 & bota8 = 1 & bota12 = 1 & CMos10 = 1) then
  MosseG1 = MosseG1 + 1
  CMos10 = 0
end
if (bota5 = 1 & bota6 = 1 & bota7 = 1 & CMos11 = 1) then
  MosseG1 = MosseG1 + 1
  CMos11 = 0
end
if (bota5 = 1 & bota10 = 1 & bota15 = 1 & CMos12 = 1) then
  MosseG1 = MosseG1 + 1
  CMos12 = 0
end
if (bota5 = 1 & bota9 = 1 & bota13 = 1 & CMos13 = 1) then
  MosseG1 = MosseG1 + 1
  CMos13 = 0
end
if (bota6 = 1 & bota10 = 1 & bota14 = 1 & CMos14 = 1) then
  MosseG1 = MosseG1 + 1
  CMos14 = 0
end
if (bota6 = 1 & bota11 = 1 & bota16 = 1 & CMos15 = 1) then
  MosseG1 = MosseG1 + 1
  CMos15 = 0
end
if (bota6 = 1 & bota7 = 1 & bota8 = 1 & CMos16 = 1) then
  MosseG1 = MosseG1 + 1
  CMos16 = 0
end
if (bota7 = 1 & bota10 = 1 & bota13 = 1 & CMos17 = 1) then
  MosseG1 = MosseG1 + 1
  CMos17 = 0
end
if (bota7 = 1 & bota11 = 1 & bota15 = 1 & CMos18 = 1) then
  MosseG1 = MosseG1 + 1
  CMos18 = 0
end
if (bota8 = 1 & bota12 = 1 & bota16 = 1 & CMos19 = 1) then
  MosseG1 = MosseG1 + 1
  CMos19 = 0
end
if (bota8 = 1 & bota11 = 1 & bota14 = 1 & CMos20 = 1) then
  MosseG1 = MosseG1 + 1
  CMos20 = 0
end
if (bota9 = 1 & bota10 = 1 & bota11 = 1 & CMos21 = 1) then
  MosseG1 = MosseG1 + 1
  CMos21 = 0
end
if (bota10 = 1 & bota11 = 1 & bota12 = 1 & CMos22 = 1) then
  MosseG1 = MosseG1 + 1
  CMos22 = 0
end
end

```

```

if (bota13 = 1 & bota14 = 1 & bota15 = 1 & CMos23 = 1) then
  MosseG1 = MosseG1 + 1
  CMos23 = 0
end
if (bota14 = 1 & bota15 = 1 & bota16 = 1 & CMos24 = 1) then
  MosseG1 = MosseG1 + 1
  CMos24 = 0
end
if (botb1 = 1 & botb2 = 1 & botb3 = 1 & CMos25 = 1) then
  MosseG2 = MosseG2 + 1
  CMos24 = 0
end
if (botb1 = 1 & botb6 = 1 & botb11 = 1 & CMos26 = 1) then
  MosseG2 = MosseG2 + 1
  CMos26 = 0
end
if (botb1 = 1 & botb5 = 1 & botb9 = 1 & CMos27 = 1) then
  MosseG2 = MosseG2 + 1
  CMos27 = 0
end
if (botb2 = 1 & botb6 = 1 & botb10 = 1 & CMos28 = 1) then
  MosseG2 = MosseG2 + 1
  CMos28 = 0
end
if (botb2 = 1 & botb7 = 1 & botb12 = 1 & CMos29 = 1) then
  MosseG2 = MosseG2 + 1
  CMos29 = 0
end
if (botb3 = 1 & botb6 = 1 & botb9 = 1 & CMos30 = 1) then
  MosseG2 = MosseG2 + 1
  CMos30 = 0
end
if (botb3 = 1 & botb7 = 1 & botb11 = 1 & CMos31 = 1) then
  MosseG2 = MosseG2 + 1
  CMos31 = 0
end
if (botb2 = 1 & botb3 = 1 & botb4 = 1 & CMos32 = 1) then
  MosseG2 = MosseG2 + 1
  CMos32 = 0
end
if (botb4 = 1 & botb7 = 1 & botb10 = 1 & CMos33 = 1) then
  MosseG2 = MosseG2 + 1
  CMos33 = 0
end
if (botb4 = 1 & botb8 = 1 & botb12 = 1 & CMos34 = 1) then
  MosseG2 = MosseG2 + 1
  CMos34 = 0
end
if (botb5 = 1 & botb6 = 1 & botb7 = 1 & CMos35 = 1) then
  MosseG2 = MosseG2 + 1
  CMos35 = 0
end
if (botb5 = 1 & botb10 = 1 & botb15 = 1 & CMos36 = 1) then
  MosseG2 = MosseG2 + 1
  CMos36 = 0
end
if (botb5 = 1 & botb9 = 1 & botb13 = 1 & CMos37 = 1) then
  MosseG2 = MosseG2 + 1
  CMos37 = 0
end
if (botb6 = 1 & botb10 = 1 & botb14 = 1 & CMos38 = 1) then
  MosseG2 = MosseG2 + 1
  CMos38 = 0
end
if (botb6 = 1 & botb11 = 1 & botb16 = 1 & CMos39 = 1) then

```

```

MosseG2 = MosseG2 + 1
CMos39 = 0
end
if (botb6 = 1 & botb7 = 1 & botb8 = 1 & CMos40 = 1) then
    MosseG2 = MosseG2 + 1
    CMos40 = 0
end
if (botb7 = 1 & botb10 = 1 & botb13 = 1 & CMos41 = 1) then
    MosseG2 = MosseG2 + 1
    CMos41 = 0
end
if (botb7 = 1 & botb11 = 1 & botb15 = 1 & CMos42 = 1) then
    MosseG2 = MosseG2 + 1
    CMos42 = 0
end
if (botb8 = 1 & botb12 = 1 & botb16 = 1 & CMos43 = 1) then
    MosseG2 = MosseG2 + 1
    CMos43 = 0
end
if (botb8 = 1 & botb11 = 1 & botb14 = 1 & CMos44 = 1) then
    MosseG2 = MosseG2 + 1
    CMos44 = 0
end
if (botb9 = 1 & botb10 = 1 & botb11 = 1 & CMos45 = 1) then
    MosseG2 = MosseG2 + 1
    CMos45 = 0
end
if (botb10 = 1 & botb11 = 1 & botb12 = 1 & CMos46 = 1) then
    MosseG2 = MosseG2 + 1
    CMos46 = 0
end
if (botb13 = 1 & botb14 = 1 & botb15 = 1 & CMos47 = 1) then
    MosseG2 = MosseG2 + 1
    CMos47 = 0
end
if (botb14 = 1 & botb15 = 1 & botb16 = 1 & CMos48 = 1) then
    MosseG2 = MosseG2 + 1
    CMos48 = 0
end
if (MosseG1 < 10) then
    Testo_MosseG1$ = '0' + CHAR(MosseG1)
    LoadText("MosseG1", "Testo_MosseG1$")
end
if (MosseG1 >= 10) then
    Testo_MosseG1$ = CHAR(MosseG1)
    LoadText("MosseG1", "Testo_MosseG1$")
end
if (MosseG2 < 10) then
    Testo_MosseG2$ = '0' + CHAR(MosseG2)
    LoadText("MosseG2", "Testo_MosseG2$")
end
if (MosseG2 >= 10) then
    Testo_MosseG2$ = CHAR(MosseG2)
    LoadText("MosseG2", "Testo_MosseG2$")
end
if (MosseG1=50) then
    Show("G1Vinto")
end
if (MosseG2=50) then
    Show("G2Vinto")
end
ControlloMosse=0
ScriptTimer("Script1", "1000")

```

## □ Script Script2

```

if ( Player2=1 & bot1=1) then
  Show("Matrix01")
  MatrixSet("MatrixA01[0,0]", "3")
  MatrixSet("Matrix01[1,1]", "2")
  botb1=1
end
if ( Player2=1 & bot2=1) then
  Show("Matrix02")
  MatrixSet("MatrixA02[0,0]", "3")
  MatrixSet("Matrix02[1,1]", "2")
  botb2=1
end
if ( Player2=1 & bot3=1) then
  Show("Matrix03")
  MatrixSet("MatrixA03[0,0]", "3")
  MatrixSet("Matrix03[1,1]", "2")
  botb3=1
end
if ( Player2=1 & bot4=1) then
  Show("Matrix04")
  MatrixSet("MatrixA04[0,0]", "3")
  MatrixSet("Matrix04[1,1]", "2")
  botb4=1
end
if ( Player2=1 & bot5=1) then
  Show("Matrix05")
  MatrixSet("MatrixA05[0,0]", "3")
  MatrixSet("Matrix05[1,1]", "2")
  botb5=1
end
if ( Player2=1 & bot6=1) then
  Show("Matrix06")
  MatrixSet("MatrixA06[0,0]", "3")
  MatrixSet("Matrix06[1,1]", "2")
  botb6=1
end
if ( Player2=1 & bot7=1) then
  Show("Matrix07")
  MatrixSet("MatrixA07[0,0]", "3")
  MatrixSet("Matrix07[1,1]", "2")
  botb7=1
end
if ( Player2=1 & bot8=1) then
  Show("Matrix08")
  MatrixSet("MatrixA08[0,0]", "3")
  MatrixSet("Matrix08[1,1]", "2")
  botb8=1
end
if ( Player2=1 & bot9=1) then
  Show("Matrix09")
  MatrixSet("MatrixA09[0,0]", "3")
  MatrixSet("Matrix09[1,1]", "2")
  botb9=1
end
if ( Player2=1 & bot10=1) then
  Show("Matrix10")
  MatrixSet("MatrixA10[0,0]", "3")
  MatrixSet("Matrix10[1,1]", "2")
  botb10=1
end
if ( Player2=1 & bot11=1) then
  Show("Matrix11")
  MatrixSet("MatrixA11[0,0]", "3")

```

```

    MatrixSet("Matrix11[1,1]", "2")
    botb11=1
end
if ( Player2=1 & bot12=1) then
    Show("Matrix12")
    MatrixSet("MatrixA12[0,0]", "3")
    MatrixSet("Matrix12[1,1]", "2")
    botb12=1
end
if ( Player2=1 & bot13=1) then
    Show("Matrix13")
    MatrixSet("MatrixA13[0,0]", "3")
    MatrixSet("Matrix13[1,1]", "2")
    botb13=1
end
if ( Player2=1 & bot14=1) then
    Show("Matrix14")
    MatrixSet("MatrixA14[0,0]", "3")
    MatrixSet("Matrix14[1,1]", "2")
    botb14=1
end
if ( Player2=1 & bot15=1) then
    Show("Matrix15")
    MatrixSet("MatrixA15[0,0]", "3")
    MatrixSet("Matrix15[1,1]", "2")
    botb15=1
end
if ( Player2=1 & bot16=1) then
    Show("Matrix16")
    MatrixSet("MatrixA16[0,0]", "3")
    MatrixSet("Matrix16[1,1]", "2")
    botb16=1
end
if (MosseG2<10) then
    Testo_MosseG2$ = '0' + CHAR(MosseG2)
end
if (MosseG2>10) then
    Testo_MosseG2$ = CHAR(MosseG2)
end
Player1=1
Player2=0
bot1=0
.
.
bot16=0
LoadText("MosseG1", "Testo_MosseG1$")
LoadText("MosseG2", "Testo_MosseG2$")
Hide("Giocatore2")
Show("Giocatore1")
c=RND(15)+1
if (Pedina=c) then
    c=RND(15)+1
    if (Pedina=c) then
        c=RND(15)+1
    end
end
if (c=1) then
    Show("MatrixB1")
end
if (c=2) then
    Show("MatrixB2")
end
if (c=3) then
    Show("MatrixB3")
end
end

```

```

if (c=4) then
  Show("MatrixB4")
end
if (c=5) then
  Show("MatrixB5")
end
if (c=6) then
  Show("MatrixB6")
end
if (c=7) then
  Show("MatrixB7")
end
if (c=8) then
  Show("MatrixB8")
end
if (c=9) then
  Show("MatrixB9")
end
if (c=10) then
  Show("MatrixB10")
end
if (c=11) then
  Show("MatrixB11")
end
if (c=12) then
  Show("MatrixB12")
end
if (c=13) then
  Show("MatrixB13")
end
if (c=14) then
  Show("MatrixB14")
end
if (c=15) then
  Show("MatrixB15")
end
if (c=16) then
  Show("MatrixB16")
end
RunScript("Script3")

```

#### □ Script Script1

```

LoadText("MosseG1", "Testo_MosseG1$")
LoadText("MosseG2", "Testo_MosseG2$")
bot1=0
.
.
bot16=0
if ( c=1) then
  Hide("MatrixB1")
  bot1=0
  Hide("Matrix01")
  Show("MatrixA01")
  MatrixSet("MatrixA01[1,1]", "1")
  bot1=0
  CMos1 = 1
  CMos2 = 1
  CMos3 = 1
  CMos26 = 1
  CMos27 = 1
  CMos28 = 1
end
if ( c=2) then
  Hide("MatrixB2")

```

```

bota2=0
Hide("Matrix02")
Show("MatrixA02")
MatrixSet("MatrixA02[1,1]","1")
botb2=0
CMos1 = 1
CMos4 = 1
CMos5 = 1
CMos8 = 1
CMos24 = 1
CMos28 = 1
CMos32 = 1
CMos29 = 1
end
if ( c=3) then
Hide("MatrixB3")
bota3=0
Hide("Matrix03")
MatrixSet("MatrixA03[1,1]","1")
Show("MatrixA03")
botb3=0
CMos1 = 1
CMos6 = 1
CMos7 = 1
CMos8 = 1
CMos24 = 1
CMos30 = 1
CMos31 = 1
CMos32 = 1
end
if ( c=4) then
Hide("MatrixB4")
bota4=0
Hide("Matrix04")
Show("MatrixA04")
MatrixSet("MatrixA04[1,1]","1")
botb4=0
CMos8 = 1
CMos9 = 1
CMos10 = 1
CMos32 = 1
CMos33 = 1
CMos34 = 1
end
if ( c=5) then
Hide("MatrixB5")
bota5=0
Hide("Matrix05")
Show("MatrixA05")
MatrixSet("MatrixA05[1,1]","1")
botb5=0
CMos3 = 1
CMos11 = 1
CMos12 = 1
CMos13 = 1
CMos27 = 1
CMos35 = 1
CMos36 = 1
CMos37 = 1
end
if ( c=6) then
Hide("MatrixB6")
bota6=0
Hide("Matrix06")
Show("MatrixA06")

```

```

MatrixSet( "MatrixA06[1,1]", "1" )
botb6=0
CMos2 = 1
CMos4 = 1
CMos6 = 1
CMos11 = 1
CMos14 = 1
CMos15 = 1
CMos16 = 1
CMos26 = 1
CMos28 = 1
CMos30 = 1
CMos35 = 1
CMos38 = 1
CMos39 = 1
CMos40 = 1
end
if ( c=7) then
  Hide( "MatrixB7" )
  bota7=0
  Hide( "Matrix07" )
  Show( "MatrixA07" )
  MatrixSet( "MatrixA07[1,1]", "1" )
  botb7=0
  CMos5 = 1
  CMos7 = 1
  CMos9 = 1
  CMos11 = 1
  CMos16 = 1
  CMos17 = 1
  CMos18 = 1
  CMos29 = 1
  CMos31 = 1
  CMos33 = 1
  CMos35 = 1
  CMos40 = 1
  CMos41 = 1
  CMos42 = 1
end
if ( c=8) then
  Hide( "MatrixB8" )
  bota8=0
  Hide( "Matrix08" )
  Show( "MatrixA08" )
  MatrixSet( "MatrixA08[1,1]", "1" )
  botb8=0
  CMos10 = 1
  CMos16 = 1
  CMos19 = 1
  CMos20 = 1
  CMos34 = 1
  CMos40 = 1
  CMos43 = 1
  CMos44 = 1
end
if ( c=9) then
  Hide( "MatrixB9" )
  bota9=0
  Hide( "Matrix09" )
  Show( "MatrixA09" )
  MatrixSet( "MatrixA09[1,1]", "1" )
  botb9=0
  CMos3 = 1
  CMos6 = 1
  CMos13 = 1

```

```

CMos21 = 1
CMos27 = 1
CMos30 = 1
CMos37 = 1
CMos45 = 1
end
if ( c=10) then
  Hide("MatrixB10")
  bot10=0
  Hide("Matrix10")
  Show("MatrixA10")
  MatrixSet("MatrixA10[1,1]","1")
  botb10=0
  CMos4 = 1
  CMos9 = 1
  CMos12 = 1
  CMos14 = 1
  CMos17 = 1
  CMos21 = 1
  CMos22 = 1
  CMos28 = 1
  CMos33 = 1
  CMos36 = 1
  CMos38 = 1
  CMos41 = 1
  CMos45 = 1
  CMos46 = 1
end
if ( c=11) then
  Hide("MatrixB11")
  bot11=0
  Hide("Matrix11")
  Show("MatrixA11")
  MatrixSet("MatrixA11[1,1]","1")
  botb11=0
  CMos2 = 1
  CMos7 = 1
  CMos15 = 1
  CMos18 = 1
  CMos20 = 1
  CMos21 = 1
  CMos22 = 1
  CMos26 = 1
  CMos31 = 1
  CMos39 = 1
  CMos42 = 1
  CMos44 = 1
  CMos45 = 1
  CMos46 = 1
end
if ( c=12) then
  Hide("MatrixB12")
  bot12=0
  Hide("Matrix12")
  Show("MatrixA12")
  MatrixSet("MatrixA12[1,1]","1")
  botb12=0
  CMos5 = 1
  CMos10 = 1
  CMos19 = 1
  CMos22 = 1
  CMos29 = 1
  CMos34 = 1
  CMos43 = 1
  CMos46 = 1

```

```

end
if ( c=13) then
  Hide("MatrixB13")
  botal3=0
  Hide("Matrix13")
  Show("MatrixA13")
  MatrixSet("MatrixA13[1,1]","1")
  botb13=0
  CMos13 = 1
  CMos17 = 1
  CMos23 = 1
  CMos37 = 1
  CMos41 = 1
  CMos47 = 1
end
if ( c=14) then
  Hide("MatrixB14")
  botal4=0
  Hide("Matrix14")
  Show("MatrixA14")
  MatrixSet("MatrixA14[1,1]","1")
  botb14=0
  CMos14 = 1
  CMos20 = 1
  CMos23 = 1
  CMos24 = 1
  CMos38 = 1
  CMos44 = 1
  CMos47 = 1
  CMos48 = 1
end
if ( c=15) then
  Hide("MatrixB15")
  botal5=0
  Hide("Matrix15")
  Show("MatrixA15")
  MatrixSet("MatrixA15[1,1]","1")
  botb15=0
  CMos12 = 1
  CMos18 = 1
  CMos23 = 1
  CMos24 = 1
  CMos36 = 1
  CMos42 = 1
  CMos47 = 1
  CMos48 = 1
end
if ( c=16) then
  Hide("MatrixB16")
  botal6=0
  Hide("Matrix16")
  Show("MatrixA16")
  MatrixSet("MatrixA16[1,1]","1")
  botb16=0
  CMos15 = 1
  CMos19 = 1
  CMos24 = 1
  CMos39 = 1
  CMos43 = 1
  CMos48 = 1
end
Hide("Analisi")

```

## □ Script MatrixPedinepassive – Down

```

if (bota5<>1 & botb5<>1) then
  MatrixSet("MatrixA05[MXCOL,MXROW]", "2")
  WaveStop()
  if (AuOn=1) then
    WavePlay("tone", "")
  end
end
end

```

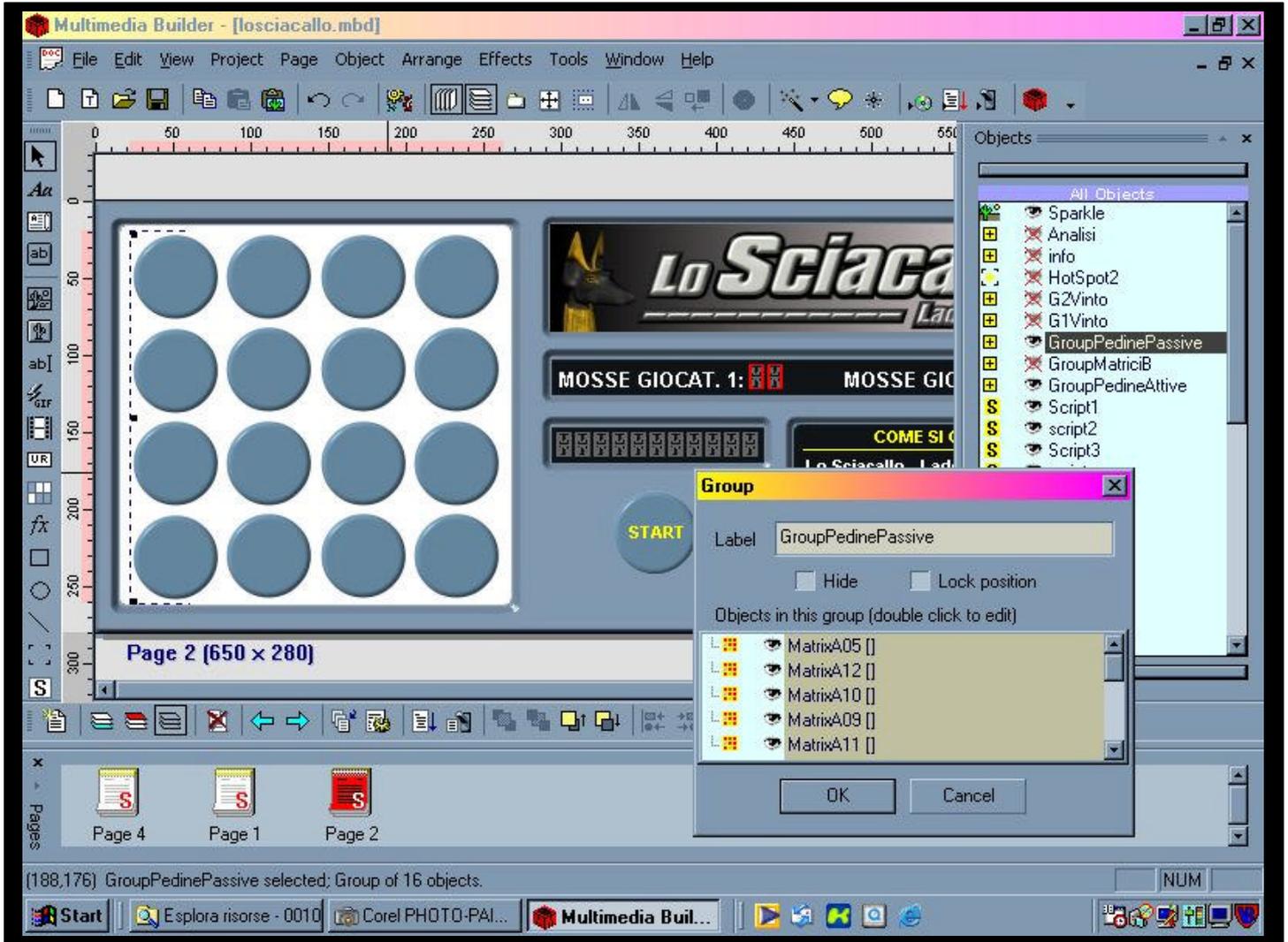


Fig. 12.6: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

❑ **Script MatrixPedinepassive - Up**

```

if (Player1=1 & Player2=0 & bota5<>1 & botb5<>1) then
  MatrixSet("MatrixA05[MXCOL,MXROW]", "1")
  ScriptTimer("script", "50")
  Pedina=5
  bot5=1
end
end

```

❑ **Tasto AuON**

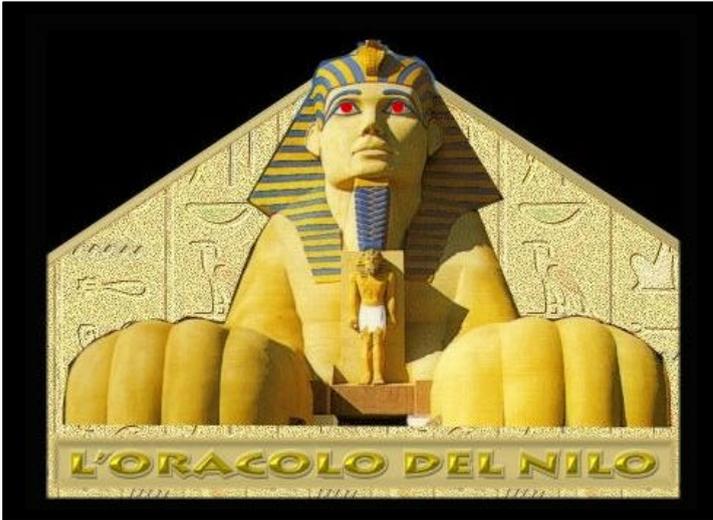
```

MidiPlay("Cantina", "[LOOP]")
Hide("Auon")
Show("Auoff")
AuOn=1

```

### 13. L'ORACOLO DEL NILO

- ❑ Versione 1.0 - Freeware.
- ❑ EVSoftware - Copyright © 2001 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



#### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,20 Mb

#### 13.1 REGOLE DEL GIOCO

L'Oracolo del Nilo è un gioco di reazione ambientato nel mondo dell'antico Egitto. Per giocare è sufficiente premere sul tasto "START". In questo modo l'Oracolo seleziona una sequenza di tre geroglifici che il giocatore deve individuare tra quelle proposte, premendo sul tasto "STOP".

All'inizio della partita vengono assegnati 10 tentativi che corrispondono a 10 risposte errate. Il display posto tra le sequenze di geroglifici segnala le varie fasi di gioco: osservazione, risposta, esito, ... mentre quello a destra indica il punteggio e i tentativi rimasti. Ogni 1000 punti i tentativi vengono incrementati di uno. E' possibile, inoltre, selezionare la velocità di rotazione dei geroglifici (in millisecondi) attraverso gli appositi tasti "<" / ">". Il tasto "ESC" consente di chiudere il programma e tornare a Windows.

#### 13.2 SUGGERIMENTI E TRUCCHI

Il videogame risulta in apparenza di difficoltà elementare ma non è così. Per i più inesperti conviene selezionare una velocità di rotazione dei geroglifici relativamente bassa, al fine di abituare i propri riflessi al cambio repentino di immagini.

#### 13.3 LA REALIZZAZIONE (THE MAKING OF)

##### ❑ LA STORIA

L'Oracolo del Nilo è nato nel 2002, sulla base della "Game Engine" sviluppata nel 1999 per il videogame "Las Vegas". L'idea originale prevedeva l'implementazione di un videogame che stimolasse i riflessi visivi del giocatore. Las Vegas, infatti, aveva delle regole elementari: sullo schermo comparivano 3 dadi da gioco in rotazione, mostrando alternativamente le varie facce (numerata da 1 a 6). Il giocatore doveva semplicemente premere il tasto "START" per lanciare i dadi e premere successivamente il tasto "STOP" quando tutti e tre i dadi mostravano la stessa faccia. Tema musicale: "Scatman World" di Scatman.

La necessità di rendere più divertente il gioco ha portato prima ad una modifica della "Game Engine" e successivamente all'introduzione del tema "Egiziano". Da un lato i geroglifici si sono rivelati un ottimo modo per costringere il giocatore a prestare la massima attenzione e

dall'altro una valida scusa per una grafica originale, ben rappresentata dalla sfinge dagli occhi infuocati (animazione che ha richiesto diverse ore di lavoro).

Curiosamente reperire un'immagine di una sfinge che si rivelasse adatta allo scopo è stata un'impresa più complicata del previsto. Dopo numerose ricerche in rete e vagliato numerose immagini (clipart e gif animate comprese...) si è scelta una fotografia della vista frontale dell'Hotel Luxor di Las Vegas.

Il tema musicale si è adattato di conseguenza alla nuova ambientazione "faraonica", passando dalla musica pop al più melodico e vagamente egiziano della colonna sonora del film "Stargate".

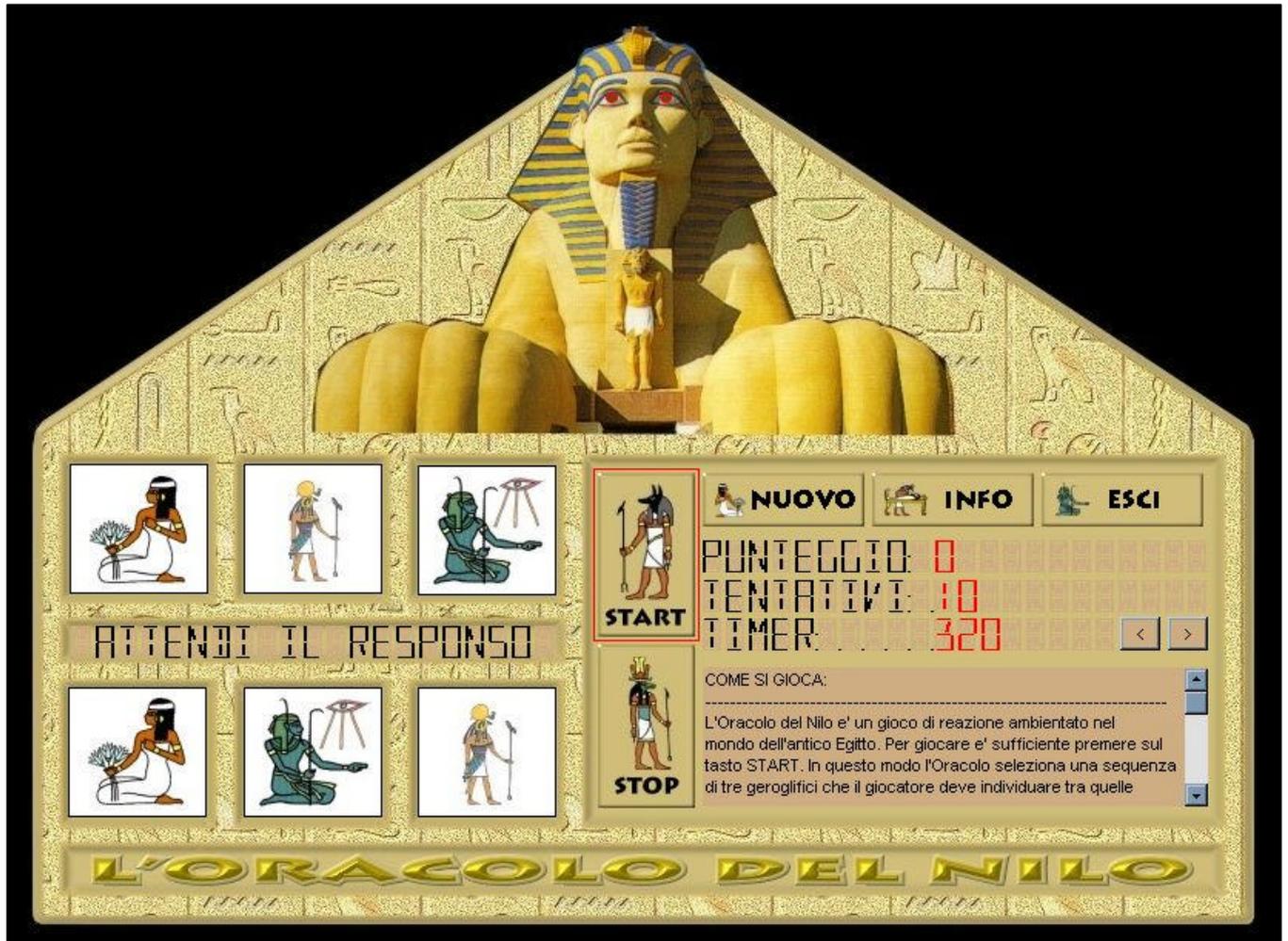


Fig. 13.1: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 13.6).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 13.2 un'immagine della schermata principale del videogame. In fig. 13.3 la finestra con le informazioni su autore e release del gioco. In fig. 13.4 c'è l' Help On Line con le istruzioni per giocare. In fig. 13.5: un'immagine del videogame Las Vegas, da cui ha avuto origine L'Oracolo del Nilo. Infine, in fig. 13.6 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

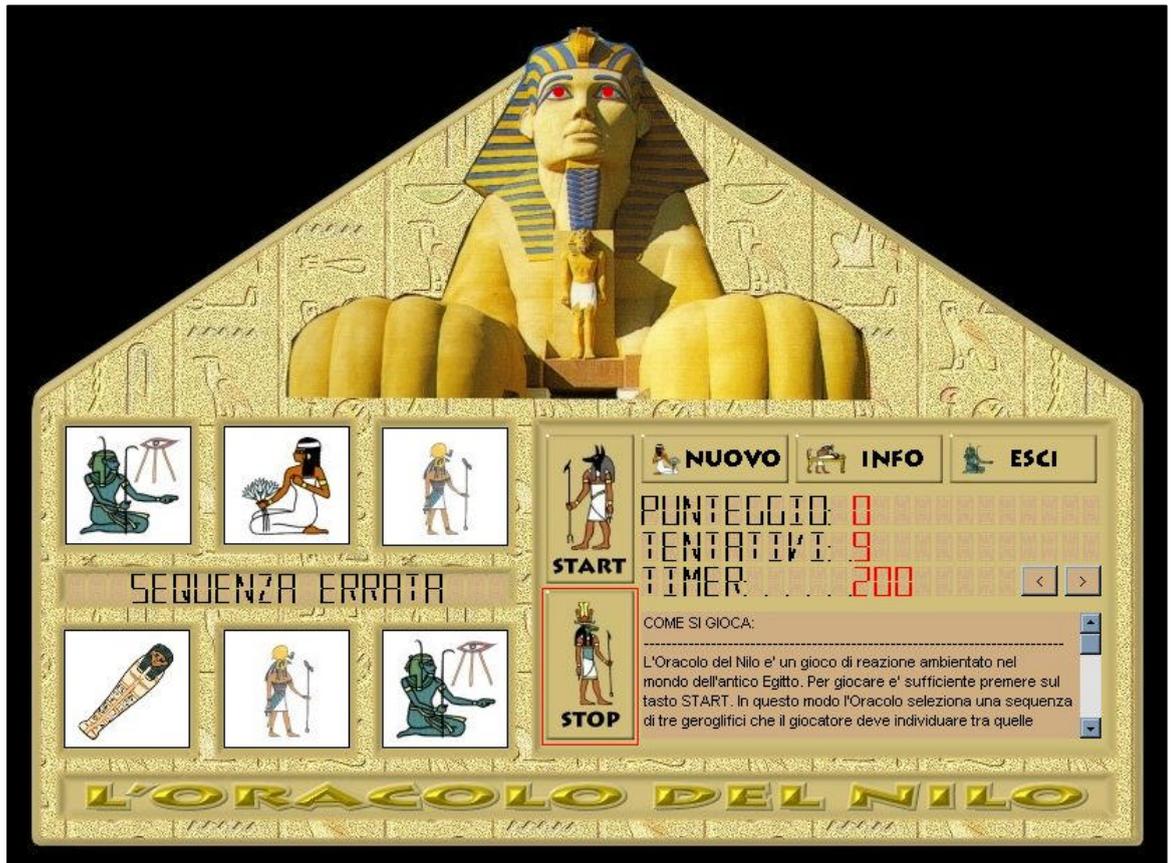


Fig. 13.2: interfaccia principale del videogioco.



Fig. 13.3: informazioni su autore e release del gioco.



Fig. 13.4: Help On Line - Istruzioni per giocare.



Fig. 13.5: schermata del vecchio videogioco Las Vegas.

## 13.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
PageTimer("1000", "Page 2")
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
InstallFont("<Embedded>\Lithogrb.ttf")
nTrials = 10
Speed = 200
NBonus = 1
DisplayValue("Speed", "Speed")
DisplayValue("Trials", "nTrials")
```

### □ Tasto NUOVO

```
nTrials = 10
nScore = 0
Speed = 200
NBonus = 1
DisplayValue("Speed", "Speed")
DisplayValue("Trials", "nTrials")
DisplayValue("Score", "nScore")
Hide("GroupInfo")
Hide("Fine")
WavePlay("Tone", "")
```

### □ Tasto INFO ON

```
Show("HotSpot5")
Show("GroupInfo")
```

### □ Tasto INFO OFF

```
Hide("HotSpot5")
Hide("GroupInfo")
```

### □ Tasto START

```
MidiPlay("stargate", "[LOOP]")
Conta = 0
Stop = 0
Hide("HotSpot")
Hide("HotSpot1")
Hide("GroupInfo")
Hide("Error")
Hide("Fine")
Hide("Correct")
Show("Wait")
RunScript("Script")
```

### □ Tasto STOP

```
MidiStop()
Hide("Find")
Hide("HotSpot1")
Show("HotSpot")
Stop = 1
isScore = FALSE
if (nScore >= 1000*NBonus ) then
  NBonus = Nbonus + 1
  nTrials = nTrials + 1
end
if (a=a1 | b=b1 | c=c1) then
```

```

    isScore = TRUE
end
if (isScore) then
    nScore =nScore+5
    isBigScore = FALSE
    if (a=a1 & b=b1 & c=c1) then
        isBigScore = TRUE
    end
    if (isBigScore) then
        if (DisMes = 0) then
            Speed = Speed - 5
            WavePlay("Bravo", "")
            Show("Correct")
        end
        nScore =nScore *a
    end
    if (isBigScore = FALSE) then
        if (DisMes = 0) then
            WavePlay("Errore", "")
            Show("Error")
        end
    end
    DisplayValue("Score", "nScore")
end
if (isScore = FALSE) then
    if (DisMes = 0) then
        WavePlay("Errore", "")
        Show("Error")
    end
    nTrials = nTrials - 1
end
if (nTrials<1) then
    Show("Fine")
    DisplayValue("Score", "nScore")
    MP3Stop()
    WavePlay("Errore", "")
    Hide("Error")
    nTrials = 10
    nScore = 0
    Speed = 200
    NBonus = 1
    Hide("GroupInfo")
end
Hide("HotSpot1")
DisplayValue("Trials", "nTrials")

```

#### □ Script Script

```

if (Speed = 0) then
    Speed = 500
end
Hide("Dice1A518")
Hide("Dice2P438")
Hide("Dice3V718")
a = RND(5) + 1
b = RND(5) + 1
c = RND(5) + 1
if (a=1) then
    Show("OneA518")
end
if (a=2) then
    Show("TwoA518")
end
if (a=3) then
    Show("ThreeA518")

```

```

end
if (a=4) then
  Show("FourA518")
end
if (a=5) then
  Show("FiveA518")
end
if (a=6) then
  Show("SixA518")
end
if (b=1) then
  Show("OneS672P438")
end
if (b=2) then
  Show("TwoS672P438")
end
if (b=3) then
  Show("ThreeS672P438")
end
if (b=4) then
  Show("FourS672P438")
end
if (b=5) then
  Show("FiveS672P438")
end
if (b=6) then
  Show("SixS672P438")
end
if (c=1) then
  Show("OneD008V718")
end
if (c=2) then
  Show("TwoD008V718")
end
if (c=3) then
  Show("ThreeD008V718")
end
if (c=4) then
  Show("FourD008V718")
end
if (c=5) then
  Show("FiveD008V718")
end
if (c=6) then
  Show("SixD008V718")
end
DisplayValue("Trials", "nTrials")
Conta = Conta + 1
if (Conta = 20) then
  RunScript("Script2")
end
if (Conta <> 20) then
  ScriptTimer("Script", "Speed")
end

```

#### □ Script Script2

```

if (Stop <> 1) then
  Hide("Wait")
  Show("HotSpot1")
  Show("Find")
  Hide("Dicel")
  Hide("Dicela")
  Hide("Dicelb")
  a1 = RND(5) + 1

```

```

b1 = RND(5) + 1
c1 = RND(5) + 1
if (a1=1) then
  Show("One")
end
if (a1=2) then
  Show("Two")
end
if (a1=3) then
  Show("Three")
end
if (a1=4) then
  Show("Four")
end
if (a1=5) then
  Show("Five")
end
if (a1=6) then
  Show("Six")
end
if (b1=1) then
  Show("Onea")
end
if (b1=2) then
  Show("Twoa")
end
if (b1=3) then
  Show("Threea")
end
if (b1=4) then
  Show("Foura")
end
if (b1=5) then
  Show("Fivea")
end
if (b1=6) then
  Show("Sixa")
end
if (c1=1) then
  Show("Oneb")
end
if (c1=2) then
  Show("Twob")
end
if (c1=3) then
  Show("Threeb")
end
if (c1=4) then
  Show("Fourb")
end
if (c1=5) then
  Show("Fiveb")
end
if (c1=6) then
  Show("Sixb")
end
DisplayValue("Trials","nTrials")
ScriptTimer("Script2","Speed")
end
if (Stop = 1) then
  Hide("Find")
end
end

```

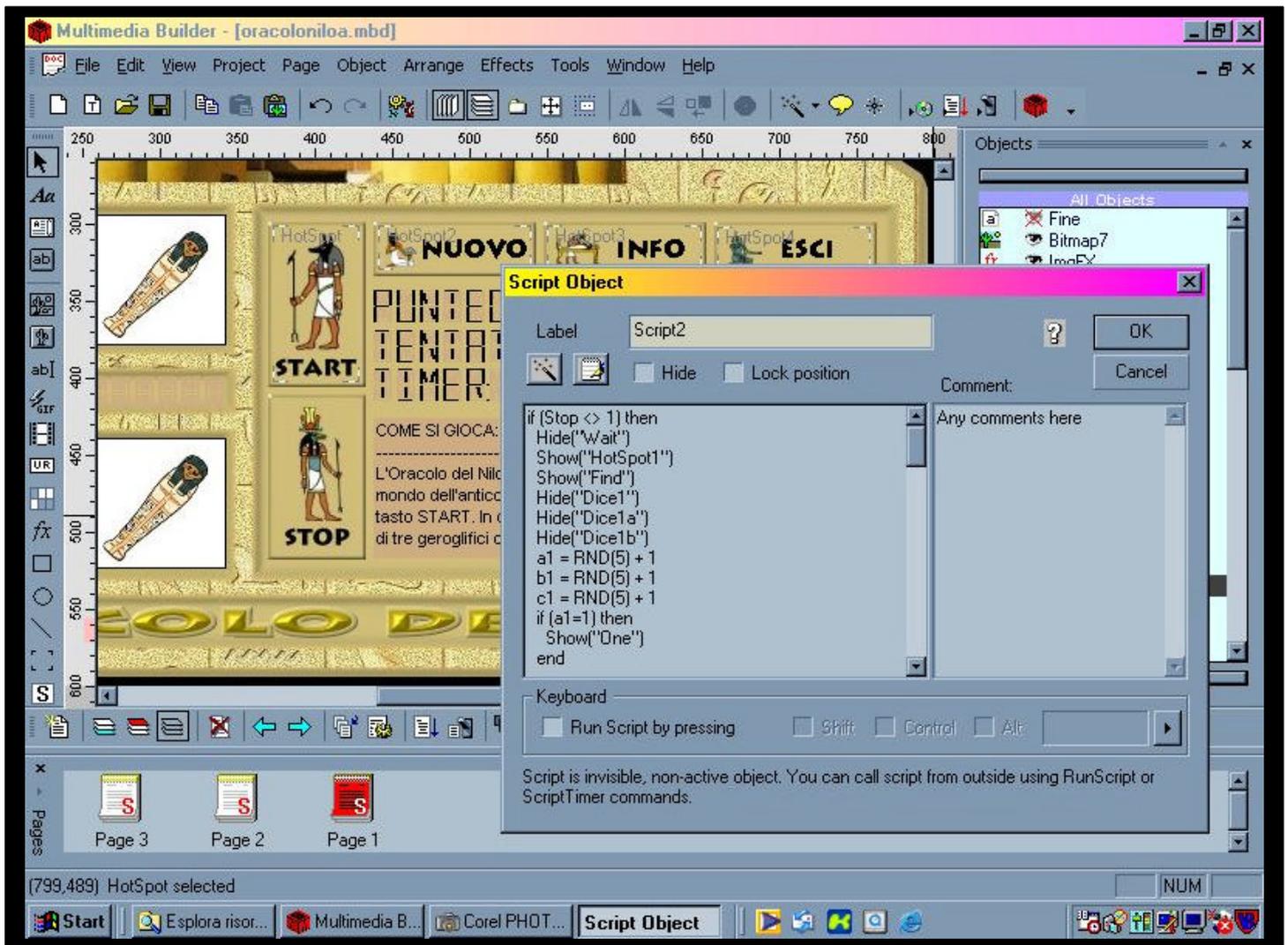


Fig. 13.6: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

## 14. IL CERVELLONE - MASTERMIND

- ❑ Versione 1.0 - Freeware.
- ❑ EVSoftware - Copyright © 2002 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,01 Mb

### 14.1 REGOLE DEL GIOCO

Il Cervellone è un videogioco d'intelligenza per un solo giocatore. L'Obiettivo è di indovinare la sequenza di quattro dischi colorati estratta casualmente dal computer. A disposizione un'urna di 10 dischi di differenti colori, tra i quali il giocatore deve scegliere la giusta sequenza, trascinando uno per volta i dischi situati in basso a destra fin sul pallottoliere, partendo dalla riga in basso indicata dalla freccia. Una volta effettuata la propria giocata, deve essere premuto il tasto bianco/blu con la freccia gialla. Il computer segnala con un semaforo giallo il numero di colori indovinati ma posizionati nelle fessure sbagliate e con un semaforo verde quelli nelle giuste postazioni. Un timer scandisce il tempo massimo disponibile per "pensare" alla propria giocata.

Tutte le opzioni possono essere selezionate dal menu di sinistra. Per effettuare una nuova partita si deve premere il pulsante "NUOVO". Il tasto "INFO" apre la finestra con le informazioni sulla versione, autore e copyright. Il tasto "HELP" mostra all'utente le istruzioni per giocare mentre quello "AUDIO" seleziona/deseleziona la musica di sottofondo, alternando diversi brani. Il volume della musica può essere modificato con il regolatore a barra verticale in basso a sinistra dello schermo. Il tasto "ESCI" chiude il videogioco e riporta al sistema operativo Windows. I due "Message Display" in basso indicano il punteggio e segnalano i secondi rimasti e il numero della partita

### 14.2 SUGGERIMENTI E TRUCCHI

Essendo un gioco di logica non si possono dare particolari suggerimenti. Conviene però prestare attenzione al trascinamento dei dischi colorati. Posizionarsi con il mouse su uno dei dischi contenuti nell'urna. Tenendo premuto il tasto sinistro trascinare il disco sopra la postazione di destinazione. Non rilasciare il pulsante del mouse fino a quando non ci si è posizionati correttamente sulle coordinate desiderate.

### 14.3 LA REALIZZAZIONE (THE MAKING OF)

#### ❑ LA STORIA

Il Cervellone è una rivisitazione del famosissimo videogame "Mastermind", apparso in diverse sembianze sia come videogame che come gioco da tavolo in tutti i paesi del mondo.

Le regole molto semplici ma la necessità di concentrazione, memoria e logica ne fanno un “classico” dei giochi moderni. Questo è anche evidenziato dal numero elevato dei download effettuati dal sito internet [www.evweb.it](http://www.evweb.it), fin dalla sua prima apparizione, nel 2002. Per realizzare la sua “Game Engine” si è utilizzata una DLL (*Dinamic Link Library*) sviluppata da un programmatore anonimo e che consente l’effetto di trascinamento dei dischi colorati da una postazione ad un’altra, gestendo le coordinate di spostamento in tempo reale tramite mouse. Senza questo espediente in videogame non sarebbe mai stato implementato.



Fig. 14.2: interfaccia principale del videogame.

□ **I SOFTWARE UTILIZZATI**

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 14.5).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

□ **BOZZE E SCREENSHOT**

Ecco alcuni screenshot tratti dall’ultima versione del gioco.

In fig. 14.2 un’immagine tratta dell’interfaccia principale del videogame.

In fig. 14.3 ci sono le istruzioni del gioco, mentre in fig. 14.4 la finestra con le informazioni su autore, release, sito internet.

Infine, in fig. 14.5 una schermata dell’ambiente di programmazione Mediachance Multimedia Builder.



Fig. 14.3: al centro la finestra con le istruzioni per giocare.



Fig. 14.4: informazioni su autore, release e sito internet.

## 14.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
if (frompage = 4 | frompage = 3) then
  Return()
end
VolumeUp("50")
AudioON = 1
Hide("Tau")
Show("Tau1")
RunScript("new")
```

### □ Script Tasto NUOVO

```
Goto Page 2
```

### □ Script Tasto HELP

```
Show("Help")
Hide("Info")
Hide("Group3")
```

### □ Script Tasto INFO

```
Hide("Help")
Show("Info")
Hide("Group3")
```

### □ Script Tasto AUDIO

```
Musica = Musica + 1
if (AudioON = 0) then
  AudioON = 1
  Musica = 1
end
Hide("Tau")
if (Musica = 1 & AudioON = 1) then
  MidiPlay("Musical", "LOOP")
  Show("Tau1")
end
if (Musica = 2 & AudioON = 1) then
  MidiPlay("Musica2", "LOOP")
  Show("Tau2")
end
if (Musica = 3 & AudioON = 1) then
  MidiPlay("Musica3", "LOOP")
  Show("Tau3")
end
if (Musica = 4 & AudioON = 1) then
  MidiPlay("Musica4", "LOOP")
  Show("Tau4")
end
if (Musica = 5 & AudioON = 1) then
  MidiPlay("Musica5", "LOOP")
  Show("Tau5")
end
if (Musica = 6 & AudioON = 1) then
  MidiPlay("Musica6", "LOOP")
  Show("Tau6")
end
if (Musica = 7 & AudioON = 1) then
```

```

    MidiPlay("Musica7","LOOP")
    Show("Tau7")
end
if (Musica = 8 & AudioON = 1) then
    MidiPlay("Musica8","LOOP")
    Show("Tau8")
end
if (Musica = 9 & AudioON = 1) then
    MidiPlay("Musica9","LOOP")
    Show("Tau9")
end
if (Musica = 10 & AudioON = 1) then
    MidiPlay("Musica10","LOOP")
    Show("Tau10")
end
if (Musica > 10) then
    AudioON = 0
    MidiStop()
    Show("Tauoff")
end

```

#### ❑ Script Mous0

```
mous1 = 0
```

#### ❑ Script Volume Slide

```

if (mous1 = 1) then
    ym = MOUSEY()
    newy = ym-starty
    if (newy > 515) then
        newy = 515
    end
    if (newy < 475) then
        newy = 475
    end
    MoveObject("slider","x,newy")
    Refresh()
    volume = (newy - 515) * (-2.5)
    VolumeUp("volume")
    ScriptTimer("volume slide","0")
end

```

#### ❑ Script rnd4

```

for check=0 to infinity
    char4 = RND(9) + 1
    if (char4<>char1 & char4<>char2 & char4<>char3) then
        Return()
    end
end
next check

```

#### ❑ Script rnd3

```

for check=0 to infinity
    char3 = RND(9) + 1
    if (char3<>char1 & char3<>char2) then
        RunScript("rnd4")
        Return()
    end
end
next check

```

#### ❑ Script rnd2

```

for check=0 to infinity
    char2 = RND(9) + 1
    if (char2<>char1) then

```

```

    RunScript("rnd3")
    Return()
end
next check

```

#### □ Script rnd1

```

char1=RND(9) + 1
RunScript("rnd2")

```

#### □ Script Check

```

cravi = 0
bikove = 0
if (select1 = char1) then
    bikove = bikove + 1
    punti = punti + 100
end
if (select2 = char2) then
    bikove = bikove + 1
    punti = punti + 100
end
if (select3 = char3) then
    bikove = bikove + 1
    punti = punti + 100
end
if (select4 = char4) then
    bikove = bikove + 1
    punti = punti + 100
end
if (select1 = char1 | select1 = char4 | select1 = char3 | select1 = char2) then
    cravi = cravi + 1
    punti = punti + 50
end
if (select2 = char2 | select2 = char4 | select2 = char3 | select2 = char1) then
    cravi = cravi + 1
    punti = punti + 50
end
if (select3 = char3 | select3 = char4 | select3 = char1 | select3 = char2) then
    cravi = cravi + 1
    punti = punti + 50
end
if (select4 = char4 | select4 = char1 | select4 = char3 | select4 = char2) then
    cravi = cravi + 1
    punti = punti + 50
end
select1 = 0
select2 = 0
select3 = 0
select4 = 0
line2 = line - 1
linea$ = CHAR(line2)
craval$ = 'craval-' + linea$
crava2$ = 'crava2-' + linea$
crava3$ = 'crava3-' + linea$
crava4$ = 'crava4-' + linea$
if (cravi = 1) then
    Show("craval$")
end
if (cravi = 2) then
    Show("craval$")
    Show("crava2$")
end
if (cravi = 3) then
    Show("craval$")
    Show("crava2$")
end

```

```

    Show("crava3$")
end
if (cravi = 4) then
    Show("crava1$")
    Show("crava2$")
    Show("crava3$")
    Show("crava4$")
end
bik1$ = 'bik1-' + linea$
bik2$ = 'bik2-' + linea$
bik3$ = 'bik3-' + linea$
bik4$ = 'bik4-' + linea$
if (bikove = 1) then
    Show("bik1$")
end
if (bikove = 2) then
    Show("bik1$")
    Show("bik2$")
end
if (bikove = 3) then
    Show("bik1$")
    Show("bik2$")
    Show("bik3$")
end
if (bikove = 4) then
    Show("bik1$")
    Show("bik2$")
    Show("bik3$")
    Show("bik4$")
    RunScript("you win")
end
select1 = 0
select2 = 0
select3 = 0
select4 = 0
DisplayValue("Punteggio", "punti")

```

#### □ Script Scroll

```

if (mous = 1) then
    xm = MOUSEX()
    ym = MOUSEY()
    newx = xm-startx
    newy = ym-starty
    MoveObject("object$", "newx,newy")
    Refresh()
    ScriptTimer("scroll", "0")
end

```

#### □ Script DetectPosition

```

Refresh()
if (object$ = 'red') then
    thecolor$ = '255,0,0'
    select = 1
end
if (object$ = 'blue') then
    thecolor$ = '0,0,255'
    select = 2
end
if (object$ = 'green') then
    thecolor$ = '0,255,0'
    select = 3
end
if (object$ = 'yellow') then
    thecolor$ = '255,255,0'

```

```

    select = 4
end
if (object$ = 'orange') then
    thecolor$ = '255,173,91'
    select = 5
end
if (object$ = 'black') then
    thecolor$ = '0,0,0'
    select = 6
end
if (object$ = 'siv') then
    thecolor$ = '192,192,192'
    select = 7
end
if (object$ = 'white') then
    thecolor$ = '255,255,255'
    select = 8
end
if (object$ = 'rozov') then
    thecolor$ = '255,160,255'
    select = 9
end
if (object$ = 'lilav') then
    thecolor$ = '192,0,255'
    select = 10
end
xa = MOUSEX()
ya = MOUSEY()
colory$ = '-26'
if (ya<150 | ya>440) then
    Return()
end
if (ya>150 & ya<170 & line = 10) then
    colory$ = '260'
end
if (ya>180 & ya<200 & line = 9) then
    colory$ = '230'
end
if (ya>210 & ya<230 & line = 8) then
    colory$ = '202'
end
if (ya>240 & ya<260 & line = 7) then
    colory$ = '172'
end
if (ya>270 & ya<290 & line = 6) then
    colory$ = '144'
end
if (ya>300 & ya<320 & line = 5) then
    colory$ = '114'
end
if (ya>330 & ya<350 & line = 4) then
    colory$ = '86'
end
if (ya>360 & ya<380 & line = 3) then
    colory$ = '58'
end
if (ya>390 & ya<410 & line = 2) then
    colory$ = '28'
end
if (ya>420 & ya<440 & line = 1) then
    colory$ = '0'
end
if (colory$ = '-26') then
    Return()
end
end

```

```

if (xa<620 | xa>740) then
  Return()
end
if (xa>625 & xa<650) then
  colorx$ = '0'
  select1 = select
  if (select = select2 | select = select3 | select = select4) then
    Return()
  end
end
if (xa>651 & xa<688) then
  colorx$ = '28'
  select2 = select
  if (select = select1 | select = select3 | select = select4) then
    Return()
  end
end
if (xa>689 & xa<715) then
  colorx$ = '58'
  select3 = select
  if (select = select1 | select = select2 | select = select4) then
    Return()
  end
end
if (xa>716 & xa<745) then
  colorx$ = '90'
  select4 = select
  if (select = select1 | select = select2 | select = select3) then
    Return()
  end
end
RunScript("colorize")
Return()

```

#### □ Script Colorize

```

command$ = 'colour ' + thecolor$
PluginSet("dg", "command$")
command$ = 'filledbox ' + colorx$ + ',' + colory$ + ',26,26'
PluginSet("dg", "command$")

```

#### □ Script You Win

```

Hide("ImgFX")
Hide("Text1")
if (char1 = 1) then
  original1$ = '255,0,0'
end
if (char1 = 2) then
  original1$ = '0,0,255'
end
if (char1 = 3) then
  original1$ = '0,255,0'
end
if (char1 = 4) then
  original1$ = '255,255,0'
end
if (char1 = 5) then
  original1$ = '255,173,91'
end
if (char1 = 6) then
  original1$ = '0,0,0'
end
if (char1 = 7) then
  original1$ = '192,192,192'
end

```

```
if (char1 = 8) then
  original1$ = '255,255,255'
end
if (char1 = 9) then
  original1$ = '255,160,255'
end
if (char1 = 10) then
  original1$ = '192,0,255'
end
if (char2 = 1) then
  original2$ = '255,0,0'
end
if (char2 = 2) then
  original2$ = '0,0,255'
end
if (char2 = 3) then
  original2$ = '0,255,0'
end
if (char2 = 4) then
  original2$ = '255,255,0'
end
if (char2 = 5) then
  original2$ = '255,173,91'
end
if (char2 = 6) then
  original2$ = '0,0,0'
end
if (char2 = 7) then
  original2$ = '192,192,192'
end
if (char2 = 8) then
  original2$ = '255,255,255'
end
if (char2 = 9) then
  original2$ = '255,160,255'
end
if (char2 = 10) then
  original2$ = '192,0,255'
end
if (char3 = 1) then
  original3$ = '255,0,0'
end
if (char3 = 2) then
  original3$ = '0,0,255'
end
if (char3 = 3) then
  original3$ = '0,255,0'
end
if (char3 = 4) then
  original3$ = '255,255,0'
end
if (char3 = 5) then
  original3$ = '255,173,91'
end
if (char3 = 6) then
  original3$ = '0,0,0'
end
if (char3 = 7) then
  original3$ = '192,192,192'
end
if (char3 = 8) then
  original3$ = '255,255,255'
end
if (char3 = 9) then
  original3$ = '255,160,255'
```

```

end
if (char3 = 10) then
    original3$ = '192,0,255'
end
if (char4 = 1) then
    original4$ = '255,0,0'
end
if (char4 = 2) then
    original4$ = '0,0,255'
end
if (char4 = 3) then
    original4$ = '0,255,0'
end
if (char4 = 4) then
    original4$ = '255,255,0'
end
if (char4 = 5) then
    original4$ = '255,173,91'
end
if (char4 = 6) then
    original4$ = '0,0,0'
end
if (char4 = 7) then
    original4$ = '192,192,192'
end
if (char4 = 8) then
    original4$ = '255,255,255'
end
if (char4 = 9) then
    original4$ = '255,160,255'
end
if (char4 = 10) then
    original4$ = '192,0,255'
end
command$ = 'colour ' + original1$
PluginSet("dg2", "command$")
command$ = 'filledbox 0,0,26,26'
PluginSet("dg2", "command$")
command$ = 'colour ' + original2$
PluginSet("dg2", "command$")
command$ = 'filledbox 29,0,26,26'
PluginSet("dg2", "command$")
command$ = 'colour ' + original3$
PluginSet("dg2", "command$")
command$ = 'filledbox 59,0,26,26'
PluginSet("dg2", "command$")
command$ = 'colour ' + original4$
PluginSet("dg2", "command$")
command$ = 'filledbox 90,0,26,26'
PluginSet("dg2", "command$")
Refresh()
Refresh()
RunScript("HaiVinto")

```

## ❑ Script You Lose

```

Hide("ImgFX")
Hide("Text1")
if (char1 = 1) then
    original1$ = '255,0,0'
end
if (char1 = 2) then
    original1$ = '0,0,255'
end
if (char1 = 3) then

```

```
    original1$ = '0,255,0'  
end  
if (char1 = 4) then  
    original1$ = '255,255,0'  
end  
if (char1 = 5) then  
    original1$ = '255,173,91'  
end  
if (char1 = 6) then  
    original1$ = '0,0,0'  
end  
if (char1 = 7) then  
    original1$ = '192,192,192'  
end  
if (char1 = 8) then  
    original1$ = '255,255,255'  
end  
if (char1 = 9) then  
    original1$ = '255,160,255'  
end  
if (char1 = 10) then  
    original1$ = '192,0,255'  
end  
if (char2 = 1) then  
    original2$ = '255,0,0'  
end  
if (char2 = 2) then  
    original2$ = '0,0,255'  
end  
if (char2 = 3) then  
    original2$ = '0,255,0'  
end  
if (char2 = 4) then  
    original2$ = '255,255,0'  
end  
if (char2 = 5) then  
    original2$ = '255,173,91'  
end  
if (char2 = 6) then  
    original2$ = '0,0,0'  
end  
if (char2 = 7) then  
    original2$ = '192,192,192'  
end  
if (char2 = 8) then  
    original2$ = '255,255,255'  
end  
if (char2 = 9) then  
    original2$ = '255,160,255'  
end  
if (char2 = 10) then  
    original2$ = '192,0,255'  
end  
if (char3 = 1) then  
    original3$ = '255,0,0'  
end  
if (char3 = 2) then  
    original3$ = '0,0,255'  
end  
if (char3 = 3) then  
    original3$ = '0,255,0'  
end  
if (char3 = 4) then  
    original3$ = '255,255,0'  
end  
end
```

```

if (char3 = 5) then
    original3$ = '255,173,91'
end
if (char3 = 6) then
    original3$ = '0,0,0'
end
if (char3 = 7) then
    original3$ = '192,192,192'
end
if (char3 = 8) then
    original3$ = '255,255,255'
end
if (char3 = 9) then
    original3$ = '255,160,255'
end
if (char3 = 10) then
    original3$ = '192,0,255'
end
if (char4 = 1) then
    original4$ = '255,0,0'
end
if (char4 = 2) then
    original4$ = '0,0,255'
end
if (char4 = 3) then
    original4$ = '0,255,0'
end
if (char4 = 4) then
    original4$ = '255,255,0'
end
if (char4 = 5) then
    original4$ = '255,173,91'
end
if (char4 = 6) then
    original4$ = '0,0,0'
end
if (char4 = 7) then
    original4$ = '192,192,192'
end
if (char4 = 8) then
    original4$ = '255,255,255'
end
if (char4 = 9) then
    original4$ = '255,160,255'
end
if (char4 = 10) then
    original4$ = '192,0,255'
end
command$ = 'colour ' + original1$
PluginSet("dg2", "command$")
command$ = 'filledbox 0,0,26,26'
PluginSet("dg2", "command$")
command$ = 'colour ' + original2$
PluginSet("dg2", "command$")
command$ = 'filledbox 29,0,26,26'
PluginSet("dg2", "command$")
command$ = 'colour ' + original3$
PluginSet("dg2", "command$")
command$ = 'filledbox 59,0,26,26'
PluginSet("dg2", "command$")
command$ = 'colour ' + original4$
PluginSet("dg2", "command$")
command$ = 'filledbox 90,0,26,26'
PluginSet("dg2", "command$")
Refresh()

```

```
RunScript("HaiPerso")
```

## □ Script New 2

```
MoveObject("run", "587,416")
select1 = 0
select2 = 0
select3 = 0
select4 = 0
line = 1
Musica = Musica + 1
Hide("Tau")
HaiVinto = 0
Show("Group2")
Show("Group3")
Hide("BloccaCampo")
DisplayValue("Punteggio", "punti")
DisplayValue("Game", "game")
DisplayValue("Secondi", "secondi")
Show("Rectangle4")
if (Musica = 1 & AudioON = 1) then
    MidiPlay("Musical", "LOOP")
    Show("Tau1")
end
if (Musica = 2 & AudioON = 1) then
    MidiPlay("Musica2", "LOOP")
    Show("Tau2")
end
if (Musica = 3 & AudioON = 1) then
    MidiPlay("Musica3", "LOOP")
    Show("Tau3")
end
if (Musica = 4 & AudioON = 1) then
    MidiPlay("Musica4", "LOOP")
    Show("Tau4")
end
if (Musica = 5 & AudioON = 1) then
    MidiPlay("Musica5", "LOOP")
    Show("Tau5")
end
if (Musica = 6 & AudioON = 1) then
    MidiPlay("Musica6", "LOOP")
    Show("Tau6")
end
if (Musica = 7 & AudioON = 1) then
    MidiPlay("Musica7", "LOOP")
    Show("Tau7")
end
if (Musica = 8 & AudioON = 1) then
    MidiPlay("Musica8", "LOOP")
    Show("Tau8")
end
if (Musica = 9 & AudioON = 1) then
    MidiPlay("Musica9", "LOOP")
    Show("Tau9")
end
if (Musica = 10 & AudioON = 1) then
    MidiPlay("Musical0", "LOOP")
    Show("Tau10")
end
Hide("cravabik1")
Hide("cravabik2")
Hide("cravabik3")
Hide("cravabik4")
Hide("cravabik5")
```

```

Hide("cravabik6")
Hide("cravabik7")
Hide("cravabik8")
Hide("cravabik9")
Hide("cravabik10")
command$ = 'colour 114,125,145'
PluginSet("dg", "command$")
PluginSet("dg2", "command$")
command$ = 'cls'
PluginSet("dg", "command$")
PluginSet("dg2", "command$")
RunScript("rnd1")

```

## □ Script New

```

MatrixSet("Matrix3[0,0]", "1")
MoveObject("run", "587,416")
select1 = 0
select2 = 0
select3 = 0
select4 = 0
line = 1
colonna = 0
punti = 0
tempo = 300000
game = 1
secondi = tempo/1000
Musica = RND(9) + 1
Hide("Tau")
HaiVinto = 0
Show("Group2")
Show("Group3")
Hide("BloccaCampo")
MatrixSet("Matrix[0,0]", "1")
DisplayValue("Punteggio", "punti")
DisplayValue("Game", "game")
DisplayValue("Secondi", "secondi")
Show("Rectangle4")
if (Musica = 1 & AudioON = 1) then
    MidiPlay("Musical", "LOOP")
    Show("Tau1")
end
if (Musica = 2 & AudioON = 1) then
    MidiPlay("Musica2", "LOOP")
    Show("Tau2")
end
if (Musica = 3 & AudioON = 1) then
    MidiPlay("Musica3", "LOOP")
    Show("Tau3")
end
if (Musica = 4 & AudioON = 1) then
    MidiPlay("Musica4", "LOOP")
    Show("Tau4")
end
if (Musica = 5 & AudioON = 1) then
    MidiPlay("Musica5", "LOOP")
    Show("Tau5")
end
if (Musica = 6 & AudioON = 1) then
    MidiPlay("Musica6", "LOOP")
    Show("Tau6")
end
if (Musica = 7 & AudioON = 1) then
    MidiPlay("Musica7", "LOOP")
    Show("Tau7")
end

```

```

end
if (Musica = 8 & AudioON = 1) then
    MidiPlay("Musica8", "LOOP")
    Show("Tau8")
end
if (Musica = 9 & AudioON = 1) then
    MidiPlay("Musica9", "LOOP")
    Show("Tau9")
end
if (Musica = 10 & AudioON = 1) then
    MidiPlay("Musica10", "LOOP")
    Show("Tau10")
end
Hide("cravabik1")
Hide("cravabik2")
Hide("cravabik3")
Hide("cravabik4")
Hide("cravabik5")
Hide("cravabik6")
Hide("cravabik7")
Hide("cravabik8")
Hide("cravabik9")
Hide("cravabik10")
command$ = 'colour 114,125,145'
PluginSet("dg", "command$")
PluginSet("dg2", "command$")
command$ = 'cls'
PluginSet("dg", "command$")
PluginSet("dg2", "command$")
RunScript("rnd1")

```

#### □ Script HaiVinto

```

Show("Winner")
Hide("Text15")
Hide("Text16")
Show("Text18")
Show("Text17")
Hide("Rectangle4")
game = game + 1
Musica = Musica + 1
HaiVinto = 1
tempo = tempo - 1000
if (tempo <= 0) then
    tempo = 1000
end
if (punti > 10000 & game > 10 & game < 20) then
    tempo = (tempo * 1000)
end
Hide("Group2")
ScriptTimer("new2", "10000")

```

#### □ Script HaiPerso

```

Show("Winner")
Hide("Text18")
Hide("Text17")
Show("Text16")
Show("Text15")
Hide("Rectangle4")
Hide("Group2")
Show("BloccaCampo")

```

❑ **Script Timer**

```

colonna = colonna + 1
MatrixSet ("Matrix[39-colonna,1]", "2")
sec = tempo / 38
secondi = secondi - 8
DisplayValue ("Secondi", "secondi")
if (colonna<>38 & line<=10) then
    ScriptTimer ("Timer", "sec")
end
if (colonna=38 & HaiVinto=0) then
    RunScript ("you lose")
    Show ("BloccaCampo")
end
end
    
```

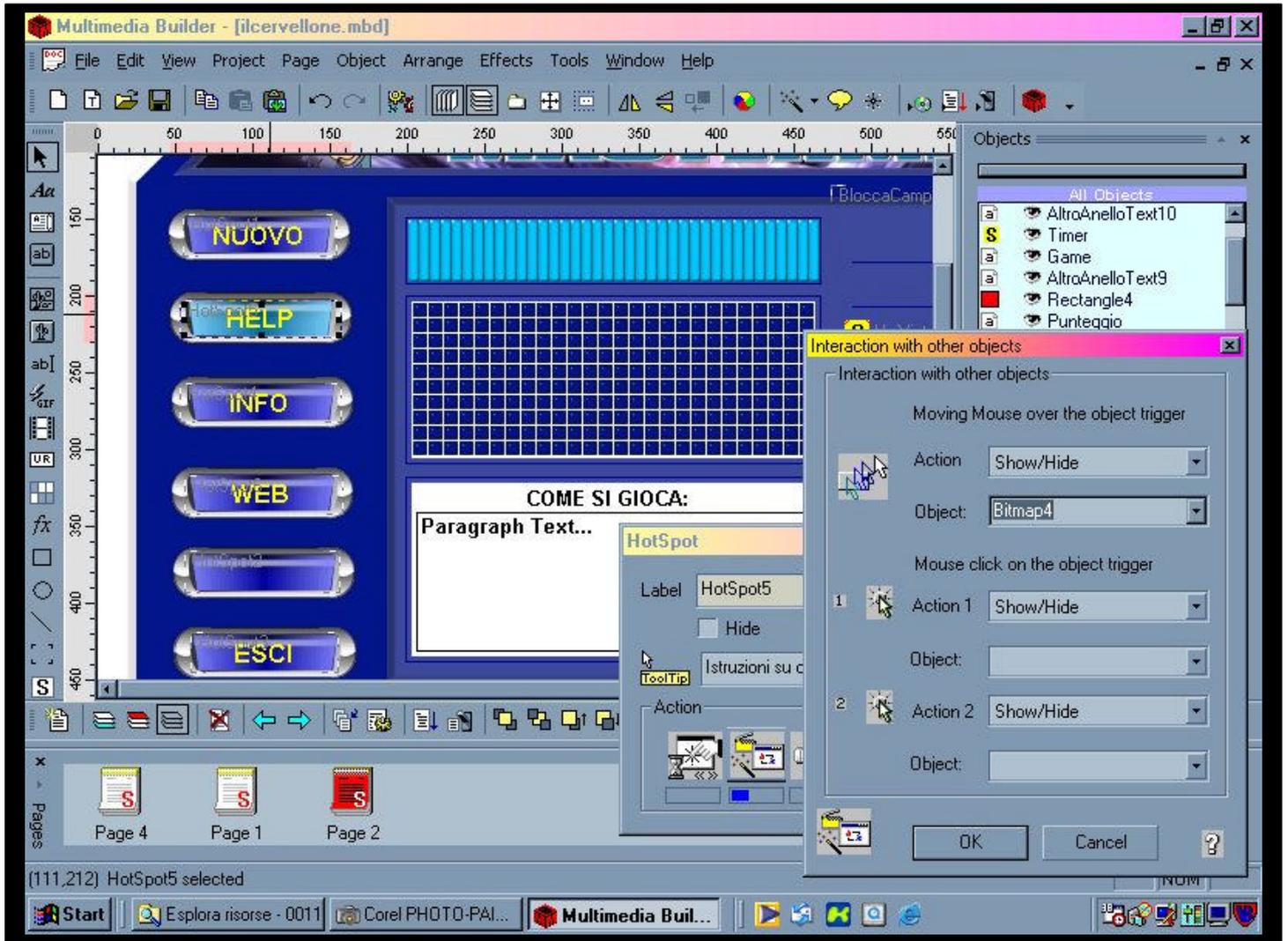


Fig. 14.5: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

## 15. THE DRAGON SIMON

- ❑ Versione 1.0 - Freeware.
- ❑ EVSoftware - Copyright © 2002 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.
- ❑ Le immagini del drago Elliott sono tratte dal film omonimo: "Elliott, il drago invisibile" e di proprietà esclusiva della Walt Disney Productions.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,81 Mb

### 15.1 REGOLE DEL GIOCO

The Dragon Simon rappresenta una rivisitazione di uno dei classici dei videogame elettronici degli anni '80: Simon. Studiato per un solo giocatore, lo scopo del gioco consiste nel ricordare e riprodurre una sequenza di suoni e colori proposta dal computer.

In questa versione "cartoon" viene richiesto di osservare e memorizzare le fasce colorate della pancia del Drago. In seguito, si deve riprodurre la sequenza nell'ordine corretto. Errori e vittorie vengono mostrati sul display laterale. Sono disponibili 3 livelli di difficoltà, selezionabili da menu premendo il tasto corrispondente (1, 2 o 3). Per iniziare una nuova partita basta premere il tasto "NUOVO". Dopo, per cominciare a giocare, memorizzando la sequenza proposta, bisogna premere il tasto "START". A differenza del "SIMON" classico, in questa versione ad ogni step viene mostrato solo l'elemento successivo della sequenza. Questo comporta che tocca al giocatore ricordarsi tutti gli elementi precedenti della sequenza. Le informazioni su autore, release, sito web, copyright, nonché le istruzioni di gioco sono mostrate in una finestra accessibile premendo il tasto "INFO-HELP". Per terminare il gioco e tornare al sistema operativo occorre premere il tasto "ESCI".

### 15.2 SUGGERIMENTI E TRUCCHI

Se non si ricorda la corretta sequenza da eseguire, conviene sfruttare il tasto "REPLAY". Ad ogni pressione, infatti, viene riproposta l'intera sequenza generata dal computer e che il giocatore deve ricordare e ripetere.

### 15.3 LA REALIZZAZIONE (THE MAKING OF)

#### ❑ LA STORIA

Nel 2002 viene l'idea di riproporre un classico degli "scacciapensieri" elettronici degli anni '80 ma in formato software. Per renderlo più appetibile si prende la decisione di realizzare una grafica molto accattivante e colorata. In prima istanza si sceglie il tema: il drago. Scegliere il drago "giusto" ha comportato una ricerca di diverse ore, scrutando i meandri della rete... alla ricerca di immagini, a seconda delle possibilità in formato cartoon, di draghi dall'aspetto "simpatico". Per caso ci s'imbatte in un sito di film della "Walt Disney" dove capeggia Elliott,

un drago a cartoni animati protagonista di un famosissimo lungometraggio Disney che mescolava attori in carne e ossa con cartoni animati. Il personaggio ben si è prestato all'idea; le fasce in cui è suddiviso il suo addome potevano essere colorate diversamente e fungere da tasti. E così è stato. Scelta la fotografia giusta è stata modificata con un software di grafica digitale per dare un effetto tridimensionale all'animale. In seguito è stata aggiunta la spiaggia ed il mare. Un elemento grafico estremamente complesso da implementare è stato l'effetto cielo con le nuvole generate dal computer. Tale effetto tendeva a rallentare la "Game Engine" a tal punto che, con un pò di riluttanza, si è deciso di disattivarlo in fase di gioco. Rimane visibile, quindi, solo tra una partita e l'altra e durante la consultazione della finestra di "HELP".



Fig. 15.1: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 15.4).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco.

In fig. 15.2 un'immagine tratta dell'interfaccia principale del videogame.

In fig. 15.3 la finestra con le informazioni su autore, release, sito internet e le istruzioni del gioco.

Infine, in fig. 15.4 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



Fig. 15.3: Al centro la finestra con le istruzioni per giocare.

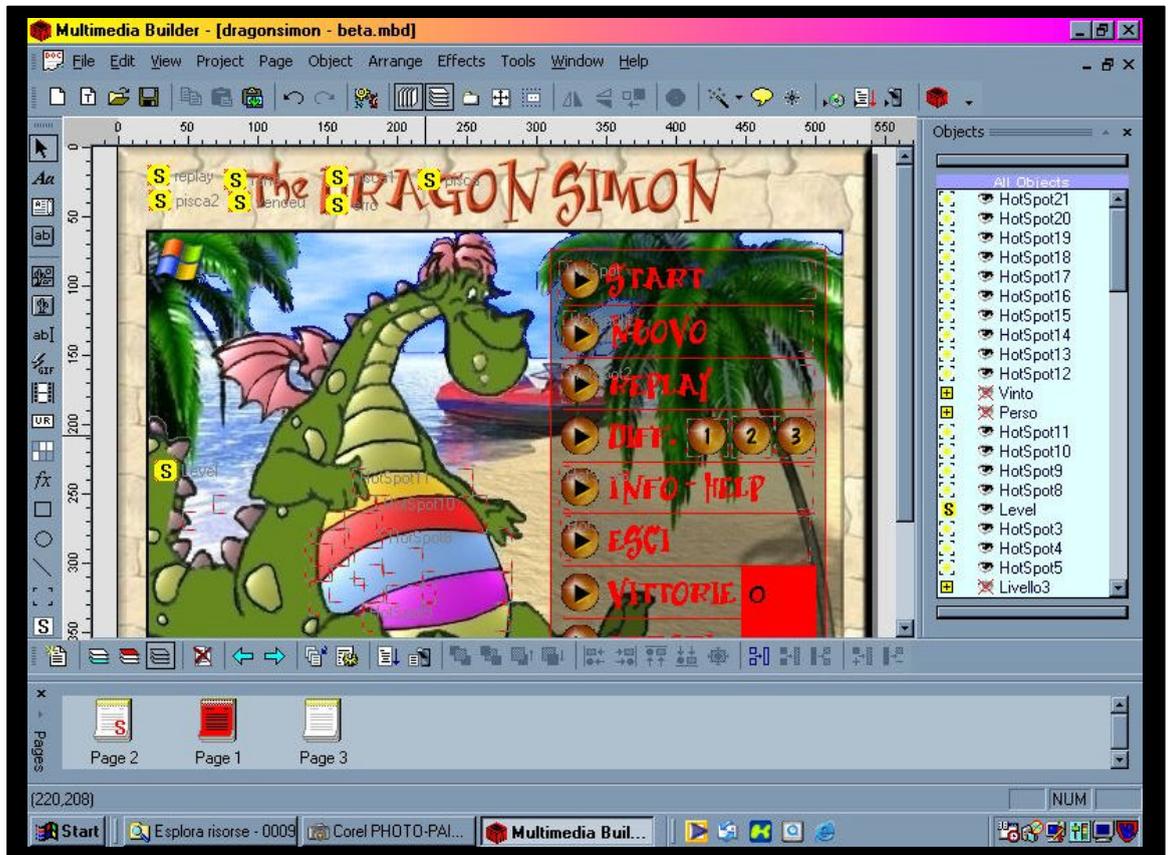


Fig. 15.4: l'ambiente di programmazione Multimedia Builder.

## 15.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
InstallFont("<Embedded>\junist.ttf")
```

### □ Script Pisca

```
if (r = 1) then
  Hide("b3a")
end
if (r = 2) then
  Hide("b1a")
end
if (r = 3) then
  Hide("b4a")
end
if (r = 4) then
  Hide("b2a")
end
```

### □ Script Erro

```
DisplayValue("cont1", "erros")
inicio = 0
Hide("b4a")
Hide("b3a")
Hide("b2a")
Hide("b1a")
Show("ImgFX")
Show("Perso")
if (stop = 1) then
  Return()
end
```

### □ Script Pisca1

```
if (y$ = '1') then
  Hide("b3a")
end
if (y$ = '2') then
  Show("b1")
  Hide("b1a")
end
if (y$ = '3') then
  Show("b4")
  Hide("b4a")
end
if (y$ = '4') then
  Show("b2")
  Hide("b2a")
end
```

### □ Script Venceu

```
Show("ImgFX")
Show("Vinto")
if (stop = 1) then
  Return()
end
```

## □ Script Rand

```

Hide ("ImgFX")
r = RND(3) + 1
x = x + 1
y = 0
cp$ = ''
if (part = x - 1) then
  RunScript("venceu")
  cont = cont + 1
  DisplayValue("cont", "cont")
  Return()
end
if (r = 1) then
  WavePlay("beep3", "")
  Show("b3a")
end
if (r = 2) then
  WavePlay("beep1", "")
  Show("bla")
end
if (r = 3) then
  WavePlay("beep4", "")
  Show("b4a")
end
if (r = 4) then
  WavePlay("beep2", "")
  Show("b2a")
end
if (x = 1) then
  aa = r
  aa$ = CHAR(r)
end
if (x = 2) then
  aa1 = r
  aa1$ = CHAR(r)
end
if (x = 3) then
  aa2 = r
  aa2$ = CHAR(r)
end
if (x = 4) then
  aa3 = r
  aa3$ = CHAR(r)
end
if (x = 5) then
  aa4 = r
  aa4$ = CHAR(r)
end
if (x = 6) then
  aa5 = r
  aa5$ = CHAR(r)
end
if (x = 7) then
  aa6 = r
  aa6$ = CHAR(r)
end
if (x = 8) then
  aa7 = r
  aa7$ = CHAR(r)
end
if (x = 9) then
  aa8 = r
  aa8$ = CHAR(r)
end
end

```

```
if (x = 10) then
  aa9 = r
  aa9$ = CHAR(r)
end
if (x = 11) then
  aa10 = r
  aa10$ = CHAR(r)
end
if (x = 12) then
  aa11 = r
  aa11$ = CHAR(r)
end
if (x = 13) then
  aa12 = r
  aa12$ = CHAR(r)
end
if (x = 14) then
  aa13 = r
  aa13$ = CHAR(r)
end
if (x = 15) then
  aa14 = r
  aa14$ = CHAR(r)
end
if (x = 16) then
  aa15 = r
  aa15$ = CHAR(r)
end
if (x = 17) then
  aa16 = r
  aa16$ = CHAR(r)
end
if (x = 18) then
  aa17 = r
  aa17$ = CHAR(r)
end
if (x = 19) then
  aa18 = r
  aa18$ = CHAR(r)
end
if (x = 20) then
  aa19 = r
  aa19$ = CHAR(r)
end
if (x = 21) then
  aa20 = r
  aa20$ = CHAR(r)
end
if (x = 22) then
  aa21 = r
  aa21$ = CHAR(r)
end
if (x = 23) then
  aa22 = r
  aa22$ = CHAR(r)
end
if (x = 24) then
  aa23 = r
  aa23$ = CHAR(r)
end
if (x = 25) then
  aa24 = r
  aa24$ = CHAR(r)
end
if (x = 26) then
```

```

aa25 = r
aa25$ = CHAR(r)
end
if (x = 27) then
aa26 = r
aa26$ = CHAR(r)
end
if (x = 28) then
aa27 = r
aa27$ = CHAR(r)
end
if (x = 29) then
aa28 = r
aa28$ = CHAR(r)
end
if (x = 30) then
aa29 = r
aa29$ = CHAR(r)
end
x$ = x$ + CHAR(r)
ScriptTimer("pisca", "250")

```

## □ Script Replay

```

Pause("200")
Hide("ImgFX")
if (aa$ <> '') then
  if (aa = 1) then
    Show("b3a")
  end
  if (aa = 2) then
    Show("b1a")
  end
  if (aa = 3) then
    Show("b4a")
  end
  if (aa = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa1$ <> '') then
  if (aa1 = 1) then
    Show("b3a")
  end
  if (aa1 = 2) then
    Show("b1a")
  end
  if (aa1 = 3) then
    Show("b4a")
  end
  if (aa1 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa2$ <> '') then
  if (aa2 = 1) then
    Show("b3a")
  end
  if (aa2 = 2) then

```

```

    Show( "b1a" )
end
if (aa2 = 3) then
    Show( "b4a" )
end
if (aa2 = 4) then
    Show( "b2a" )
end
Refresh()
ScriptTimer( "pisca2", "450" )
Pause( "550" )
end
if (aa3$ <> '' ) then
    if (aa3 = 1) then
        Show( "b3a" )
    end
    if (aa3 = 2) then
        Show( "b1a" )
    end
    if (aa3 = 3) then
        Show( "b4a" )
    end
    if (aa3 = 4) then
        Show( "b2a" )
    end
    Refresh()
    ScriptTimer( "pisca2", "450" )
    Pause( "550" )
end
if (aa4$ <> '' ) then
    if (aa4 = 1) then
        Show( "b3a" )
    end
    if (aa4 = 2) then
        Show( "b1a" )
    end
    if (aa4 = 3) then
        Show( "b4a" )
    end
    if (aa4 = 4) then
        Show( "b2a" )
    end
    Refresh()
    ScriptTimer( "pisca2", "450" )
    Pause( "550" )
end
if (aa5$ <> '' ) then
    if (aa5 = 1) then
        Show( "b3a" )
    end
    if (aa5 = 2) then
        Show( "b1a" )
    end
    if (aa5 = 3) then
        Show( "b4a" )
    end
    if (aa5 = 4) then
        Show( "b2a" )
    end
    Refresh()
    ScriptTimer( "pisca2", "450" )
    Pause( "550" )
end
if (aa6$ <> '' ) then
    if (aa6 = 1) then

```

```

    Show( "b3a" )
end
if (aa6 = 2) then
    Show( "b1a" )
end
if (aa6 = 3) then
    Show( "b4a" )
end
if (aa6 = 4) then
    Show( "b2a" )
end
Refresh( )
ScriptTimer( "pisca2", "450" )
Pause( "550" )
end
if (aa7$ <> '') then
    if (aa7 = 1) then
        Show( "b3a" )
    end
    if (aa7 = 2) then
        Show( "b1a" )
    end
    if (aa7 = 3) then
        Show( "b4a" )
    end
    if (aa7 = 4) then
        Show( "b2a" )
    end
    Refresh( )
    ScriptTimer( "pisca2", "450" )
    Pause( "550" )
end
if (aa8$ <> '') then
    if (aa8 = 1) then
        Show( "b3a" )
    end
    if (aa8 = 2) then
        Show( "b1a" )
    end
    if (aa8 = 3) then
        Show( "b4a" )
    end
    if (aa8 = 4) then
        Show( "b2a" )
    end
    Refresh( )
    ScriptTimer( "pisca2", "450" )
    Pause( "550" )
end
if (aa9$ <> '') then
    if (aa9 = 1) then
        Show( "b3a" )
    end
    if (aa9 = 2) then
        Show( "b1a" )
    end
    if (aa9 = 3) then
        Show( "b4a" )
    end
    if (aa9 = 4) then
        Show( "b2a" )
    end
    Refresh( )
    ScriptTimer( "pisca2", "450" )
    Pause( "550" )
end

```

```

end
if (aa10$ <> '') then
  if (aa10 = 1) then
    Show("b3a")
  end
  if (aa10 = 2) then
    Show("b1a")
  end
  if (aa10 = 3) then
    Show("b4a")
  end
  if (aa10 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa11$ <> '') then
  if (aa11 = 1) then
    Show("b3a")
  end
  if (aa11 = 2) then
    Show("b1a")
  end
  if (aa11 = 3) then
    Show("b4a")
  end
  if (aa11 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa12$ <> '') then
  if (aa10 = 1) then
    Show("b3a")
  end
  if (aa12 = 2) then
    Show("b1a")
  end
  if (aa12 = 3) then
    Show("b4a")
  end
  if (aa12 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa13$ <> '') then
  if (aa13 = 1) then
    Show("b3a")
  end
  if (aa13 = 2) then
    Show("b1a")
  end
  if (aa13 = 3) then
    Show("b4a")
  end
  if (aa13 = 4) then
    Show("b2a")
  end
end

```

```

Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa14$ <> '') then
  if (aa14 = 1) then
    Show("b3a")
  end
  if (aa14 = 2) then
    Show("b1a")
  end
  if (aa14 = 3) then
    Show("b4a")
  end
  if (aa14 = 4) then
    Show("b2a")
  end
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa15$ <> '') then
  if (aa15 = 1) then
    Show("b3a")
  end
  if (aa15 = 2) then
    Show("b1a")
  end
  if (aa15 = 3) then
    Show("b4a")
  end
  if (aa15 = 4) then
    Show("b2a")
  end
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa16$ <> '') then
  if (aa16 = 1) then
    Show("b3a")
  end
  if (aa16 = 2) then
    Show("b1a")
  end
  if (aa16 = 3) then
    Show("b4a")
  end
  if (aa16 = 4) then
    Show("b2a")
  end
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa17$ <> '') then
  if (aa17 = 1) then
    Show("b3a")
  end
  if (aa17 = 2) then
    Show("b1a")
  end
  if (aa17 = 3) then
    Show("b4a")
  end
end

```

```

if (aa17 = 4) then
  Show("b2a")
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa18$ <> '') then
  if (aa18 = 1) then
    Show("b3a")
  end
  if (aa18 = 2) then
    Show("bla")
  end
  if (aa18 = 3) then
    Show("b4a")
  end
  if (aa18 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa19$ <> '') then
  if (aa19 = 1) then
    Show("b3a")
  end
  if (aa19 = 2) then
    Show("bla")
  end
  if (aa19 = 3) then
    Show("b4a")
  end
  if (aa19 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa20$ <> '') then
  if (aa20 = 1) then
    Show("b3a")
  end
  if (aa20 = 2) then
    Show("bla")
  end
  if (aa20 = 3) then
    Show("b4a")
  end
  if (aa20 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa21$ <> '') then
  if (aa21 = 1) then
    Show("b3a")
  end
  if (aa21 = 2) then
    Show("bla")
  end
end

```

```

if (aa21 = 3) then
  Show("b4a")
end
if (aa21 = 4) then
  Show("b2a")
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa22$ <> '') then
  if (aa22 = 1) then
    Show("b3a")
  end
  if (aa22 = 2) then
    Show("b1a")
  end
  if (aa22 = 3) then
    Show("b4a")
  end
  if (aa22 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa23$ <> '') then
  if (aa23 = 1) then
    Show("b3a")
  end
  if (aa23 = 2) then
    Show("b1a")
  end
  if (aa23 = 3) then
    Show("b4a")
  end
  if (aa23 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa24$ <> '') then
  if (aa24 = 1) then
    Show("b3a")
  end
  if (aa24 = 2) then
    Show("b1a")
  end
  if (aa24 = 3) then
    Show("b4a")
  end
  if (aa24 = 4) then
    Show("b2a")
  end
  Refresh()
  ScriptTimer("pisca2", "450")
  Pause("550")
end
if (aa25$ <> '') then
  if (aa25 = 1) then
    Show("b3a")
  end
end

```

```

if (aa25 = 2) then
    Show("b1a")
end
if (aa25 = 3) then
    Show("b4a")
end
if (aa25 = 4) then
    Show("b2a")
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa26$ <> '') then
    if (aa26 = 1) then
        Show("b3a")
    end
    if (aa26 = 2) then
        Show("b1a")
    end
    if (aa26 = 3) then
        Show("b4a")
    end
    if (aa26 = 4) then
        Show("b2a")
    end
    Refresh()
    ScriptTimer("pisca2", "450")
    Pause("550")
end
if (aa27$ <> '') then
    if (aa27 = 1) then
        Show("b3a")
    end
    if (aa27 = 2) then
        Show("b1a")
    end
    if (aa27 = 3) then
        Show("b4a")
    end
    if (aa27 = 4) then
        Show("b2a")
    end
    Refresh()
    ScriptTimer("pisca2", "450")
    Pause("550")
end
if (aa28$ <> '') then
    if (aa28 = 1) then
        Show("b3a")
    end
    if (aa28 = 2) then
        Show("b1a")
    end
    if (aa28 = 3) then
        Show("b4a")
    end
    if (aa28 = 4) then
        Show("b2a")
    end
    Refresh()
    ScriptTimer("pisca2", "450")
    Pause("550")
end
if (aa29$ <> '') then

```

```

if (aa29 = 1) then
    Show("b3a")
end
if (aa29 = 2) then
    Show("b1a")
end
if (aa29 = 3) then
    Show("b4a")
end
if (aa29 = 4) then
    Show("b2a")
end
Refresh()
ScriptTimer("pisca2", "450")
Pause("550")
end
if (aa30$ <> '') then
    if (aa30 = 1) then
        Show("b3a")
    end
    if (aa30 = 2) then
        Show("b1a")
    end
    if (aa30 = 3) then
        Show("b4a")
    end
    if (aa30 = 4) then
        Show("b2a")
    end
    Refresh()
    ScriptTimer("pisca2", "450")
    Pause("550")
end
end

```

□ **Script Pisca2**

```

Hide("b3a")
Hide("b1a")
Hide("b4a")
Hide("b2a")

```

□ **Script Tasto Start**

```

y = 0
if (stop = 0) then
    Return()
end
stop = 0
r = 0
x = 0
x$ = ''
cp$ = ''
aa1$ = ''
aa2$ = ''
aa3$ = ''
aa4$ = ''
aa5$ = ''
aa6$ = ''
aa7$ = ''
aa8$ = ''
aa9$ = ''
aa10$ = ''
aa11$ = ''
aa12$ = ''
aa13$ = ''
aa14$ = ''

```

```

aa15$ = ''
aa16$ = ''
aa17$ = ''
aa18$ = ''
aa19$ = ''
aa20$ = ''
aa21$ = ''
aa22$ = ''
aa23$ = ''
aa24$ = ''
aa25$ = ''
aa26$ = ''
aa27$ = ''
aa28$ = ''
aa29$ = ''
aa30$ = ''

```

```
Hide("bla")
```

```
Hide("b2a")
```

```
Hide("b3a")
```

```
Hide("b4a")
```

```
aa1 = 0
```

```
aa2 = 0
```

```
aa3 = 0
```

```
aa4 = 0
```

```
aa5 = 0
```

```
aa6 = 0
```

```
aa7 = 0
```

```
aa8 = 0
```

```
aa9 = 0
```

```
aa10 = 0
```

```
aa11 = 0
```

```
aa12 = 0
```

```
aa13 = 0
```

```
aa14 = 0
```

```
aa15 = 0
```

```
aa16 = 0
```

```
aa17 = 0
```

```
aa18 = 0
```

```
aa19 = 0
```

```
aa20 = 0
```

```
aa21 = 0
```

```
aa22 = 0
```

```
aa23 = 0
```

```
aa24 = 0
```

```
aa25 = 0
```

```
aa26 = 0
```

```
aa27 = 0
```

```
aa28 = 0
```

```
aa29 = 0
```

```
aa30 = 0
```

```
RunScript("rand")
```

#### □ Script Tasto Nuovo

```
stop = 1
```

```
y = 0
```

```
if (part = 0) then
```

```
    part = 10
```

```
end
```

```
RunScript("Level")
```

```
Hide("Perso")
```

```
Hide("Vinto")
```

#### □ Script Tasto Replay

```
RunScript("replay")
```

### □ Script Tasto Level

```
part = 10  
RunScript("Level")
```

### □ Script Level

```
if (part = 10) then  
  Show("Bitmap8")  
  Hide("Livello2")  
  Hide("Livello3")  
end  
if (part = 20) then  
  Show("Livello2")  
  Hide("Livello1")  
  Hide("Livello3")  
end  
if (part = 30) then  
  Show("Livello3")  
  Hide("Livello2")  
  Hide("Livello1")  
end
```

## 16. GUARDIE E LADRI

- ❑ Versione 1.0 - Freeware.
- ❑ EVSoftware - Copyright © 2003 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,32 Mb

### 16.1 REGOLE DEL GIOCO

Guardie e Ladri è un gioco di abilità per un solo giocatore. Quest'ultimo, nei panni di un ladro incallito, deve rubare tutti i sacchi di monete d'oro che vengono disseminati sul tabellone di gioco nel corso della partita. Ad ostacolare la sua missione c'è una squadra di agenti di polizia, incaricati di arrestarlo.

Per iniziare una nuova partita basta premere il tasto "NUOVO". Viene così resettato il tabellone di gioco, posizionando casualmente il ladro, i poliziotti ed i primi sacchi di monete. Alla prima pressione di uno dei tasti "UP", "DOWN", "LEFT" o "RIGHT" il gioco ha inizio, mettendo in movimento i protagonisti. Il tabellone di gioco è da considerarsi "continuo". Questo comporta che se il ladro esce alla destra dello schermo, rientra alla sua sinistra, e viceversa. Sul display in alto a destra vengono mostrati il punteggio raggiunto ed il numero di "arresti" concessi. Inizialmente vengono accordate solo 5 possibilità di arresto.

Tutte le opzioni possono essere selezionate dal menu di sinistra. Il tasto "INFO" apre la finestra con le informazioni sulla versione, autore e copyright. Il tasto "HELP" mostra all'utente le istruzioni per giocare mentre i tasti "SI"/"NO" consentono di attivare/disattivare la musica di sottofondo (gli altri effetti audio rimangono comunque attivi). Il piccolo display posto al di sopra di tali tasti consente di monitorare la situazione in essere. I tasti "-"/"+" permettono, invece, di diminuire/aumentare la velocità del ladro e dei poliziotti. Infine, il tasto "ESCI" chiude il programma e riporta a Windows.

Velocità e numero di poliziotti e sacchi di monete variano a seconda del livello di gioco raggiunto.

### 16.2 SUGGERIMENTI E TRUCCHI

Non ci sono particolari strategie per giocare a questo videogame. Unico trucco che si può consigliare è giocare di sponda, sfruttando la continuità dal tabellone. Non attraversare mai un gruppo di poliziotti quando sembra confluire in un'unica zona, anche se ci sono spiragli.

## 16.3 LA REALIZZAZIONE (THE MAKING OF)

### □ LA STORIA

“Guardie e Ladri” nasce nel marzo del 2003 sfruttando l’idea base della “Game Engine” realizzata da *Sthauckland* per il videogioco “Honeypot”. Il videogame, non è mai stato reso disponibile per la distribuzione gratuita tramite Internet.



Fig. 16.1: interfaccia principale del videogame.

### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 16.6).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall’ultima versione del gioco. In fig. 16.2 un’immagine della schermata principale del videogame. In fig. 16.3 la finestra con le informazioni su autore e release del gioco. In fig. 16.4 c’è l’ Help On Line con le istruzioni per giocare. In fig. 16.5: schermata di game over.

Infine, in fig. 16.6 una schermata dell’ambiente di programmazione Mediachance Multimedia Builder.



Fig. 16.2: interfaccia principale del videogioco.

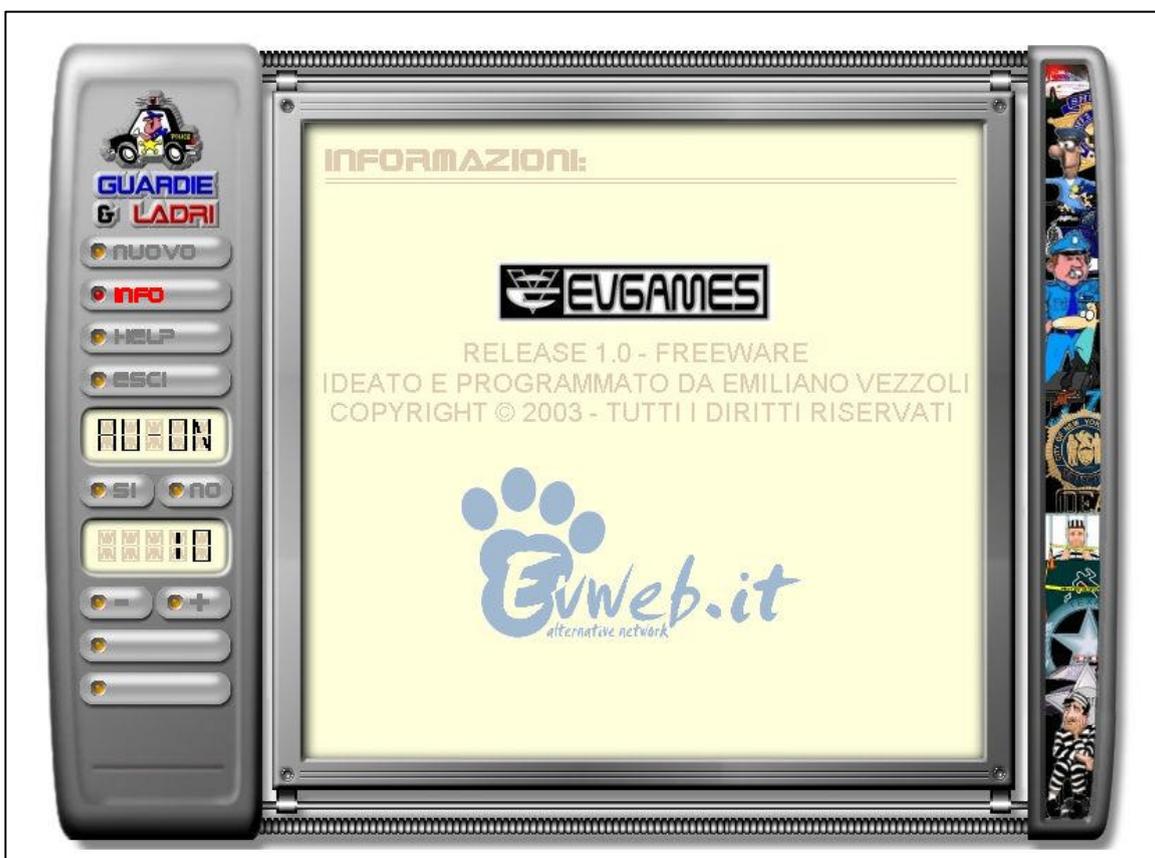


Fig. 16.3: informazioni su autore e release del gioco.



Fig. 16.4: Help On Line - Istruzioni per giocare.



Fig. 16.5: schermata di Game Over.

## 16.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

PageTimer ("1000", "Page 3")
InstallFont (<Embedded>\Led_real.ttf")
InstallFont (<Embedded>\Arial.ttf")
InstallFont (<Embedded>\BoomBox.ttf")
NuovaPartita = 1
Level = 1
if (NuovaPartita = 0) then
    life$=CHAR(lf)
    if (life$='0') then
        NextPage()
    end
    Show("KO")
    Pause("2000")
end
Hide("KO")
Speed = 10
MatrixSet("Matrix[0,0]", "0")
MatrixSet("Matrix[7,7]", "1")
x=7
y=7
for t=1 to 5
    a=RND(14)+1
    b=RND(14)+1
    MatrixSet("Matrix[a,b]", "3")
next t
NumeroPoliziotti = 5 * Level
for r= 1 to NumeroPoliziotti
    ex[r]=RND(14)+1
    ey[r]=RND(14)+1
    ex=ex[r]
    ey=ey[r]
    MatrixSet("Matrix[ex,ey]", "2")
next r
if (jj=0) then
    score=0
    lf=5
end
jj=1
Show("Matrix")
NuovaPartita = 0
Speed$=CHAR(Speed)
Level$=CHAR(Level)
life$=CHAR(lf)
LoadText("Speed", "Speed$")
LoadText("Text25", "life$")
LoadText("Level", "Level$")
MidiPlay("LEMON1TR", "loop")

```

### □ Tasto NUOVO

```

if (jj=0) then
    score=0
    lf=5
end
jj=1
RunScript("Start")
NuovaPartita = 1
if (Audio=0) then

```

```

    WavePlay( "Beep" , " " )
end
Level = 1

```

#### □ Tasto INFO ON

```

Show( "Info2" )
Hide( "Come" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
end
Show( "HotSpot10" )

```

#### □ Tasto INFO OFF

```

Hide( "Info2" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
end
Hide( "HotSpot10" )

```

#### □ Tasto HELP ON

```

Show( "Come" )
Hide( "Info2" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
end
Show( "HotSpot11" )

```

#### □ Tasto HELP OFF

```

Hide( "Come" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
end
Hide( "HotSpot11" )

```

#### □ Tasto ATTIVA AUDIO

```

MidiPlay( "LEMON1TR" , "loop" )
Show( "AudioON" )
Hide( "AudioOFF" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
end

```

#### □ Tasto DISATTIVA AUDIO

```

MidiStop( )
Hide( "AudioON" )
Show( "AudioOFF" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
end

```

#### □ Tasto SPEED LESS

```

Speed = Speed - 1
Speed$=CHAR( Speed )
LoadText( "Speed" , "Speed$" )
if (Audio=0) then
    WavePlay( "Beep" , " " )
End

```

❑ **Tasto SPEED MORE**

```
Speed = Speed + 1
Speed$=CHAR (Speed)
LoadText ( "Speed" , "Speed$" )
```

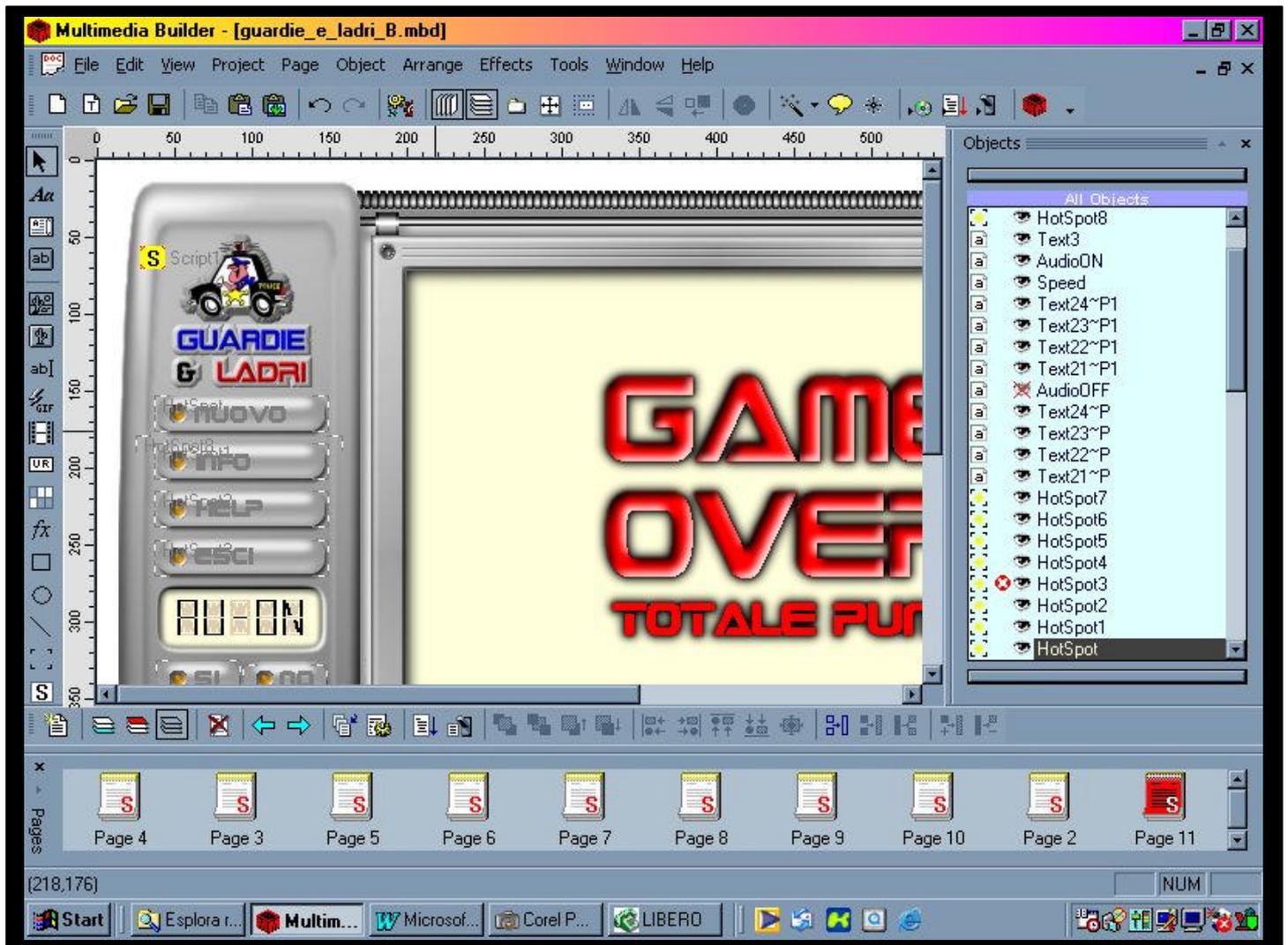


Fig. 16.6: una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

❑ **Script Script**

```
dir=1
RunScript ( "go" )
```

❑ **Script Script1**

```
dir=2
RunScript ( "go" )
```

❑ **Script Script2**

```
dir=3
RunScript ( "go" )
```

❑ **Script Script3**

```
dir=4
RunScript ( "go" )
```

❑ **Script GO**

```

life$=CHAR(lf)
if (life$='0') then
  NextPage()
end
LoadText("Text25","life$")
MatrixSet("Matrix[x,y]","0")
if (dir=1) then
  y=y-1
end
if (dir=2) then
  y=y+1
end
if (dir=3) then
  x=x-1
end
if (dir=4) then
  x=x+1
end
if (x>15) then
  x=1
end
if (x<1) then
  x=15
end
if (y>13) then
  y=1
end
if (y<1) then
  y=13
end
MatrixGet("Matrix[x,y]","img")
if (img=3) then
  WavePlay("mmenu","")
  score=score+10
  if (score/Level = 1000) then
    lf = lf + 1
    Level = Level + 1
    score = score + 200
    WavePlay("Win","")
    Speed = Speed + 20
    Level$ = CHAR(Level)
    LoadText("Level","Level$")
    LoadText("Speed","Speed$")
  end
  sc$=CHAR(score)
  LoadText("Text","sc$")
  LoadText("Text3","sc$")
  a=RND(14)+1
  b=RND(12)+1
  MatrixSet("Matrix[a,b]","3")
  a=RND(14)+1
  b=RND(12)+1
  MatrixSet("Matrix[a,b]","3")
end
if (img=2) then
  RunScript("lifel")
end
MatrixSet("Matrix[x,y]","1")
ScriptTimer("enemy","Speed")

```

#### □ Script ENIMY

```

for r = 1 to 5
  ex=ex[r]
  ey=ey[r]

```

```

    MatrixSet("Matrix[ex,ey]", "0")
next r
for t = 1 to 5
    j=(t*2)
    if (ex[t]<x) then
        ex[t]=ex[t]+(1/j)
    end
    if (ex[t]>x) then
        ex[t]=ex[t]-(1/j)
    end
    if (ey[t]<y) then
        ey[t]=ey[t]+(1/j)
    end
    if (ey[t]>y) then
        ey[t]=ey[t]-(1/j)
    end
    ex=ex[t]
    ey=ey[t]
    MatrixSet("Matrix[ex,ey]", "2")
    if (x=ex[t] ) then
        if (y=ey[t]) then
            RunScript("life1")
        end
    end
end
next t
ScriptTimer("go", "150")

```

□ **Script LIFE1**

```

lf=lf-1
if (lf<1) then
    Page("11")
end
Page("2")

```

□ **Script START**

```

MatrixSet("Matrix[0,0]", "0")
MatrixSet("Matrix[7,7]", "1")
x=7
y=7
for t=1 to 5
    a=RND(14)+1
    b=RND(12)+1
    MatrixSet("Matrix[a,b]", "3")
next t
NumeroPoliziotti = 5 * Level
for r= 1 to NumeroPoliziotti
    ex[r]=RND(14)+1
    ey[r]=RND(12)+1
    ex=ex[r]
    ey=ey[r]
    MatrixSet("Matrix[ex,ey]", "2")
next r
LoadText("Text", "sc$")
NuovaPartita = 0

```

## 17. IL BRUCO – SNAKY III

- ❑ Versione 3.0 - Freeware.
- ❑ EVSoftware - Copyright © 1999-2003 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win95/98/XP/NT
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 2,98 Mb

### 17.1 REGOLE DEL GIOCO

Il Bruco è un gioco di abilità per un solo giocatore. Si tratta di una rivisitazione di uno dei classici dei videogame: Snake.

Il giocatore veste i panni di un bruco e deve cibarsi del maggior numero di mele disperse per il prato. Ad ogni mela mangiata corrispondono un incremento della lunghezza del bruco e 1 punto da sommare al punteggio totale. La difficoltà consiste nell'evitare, aumentando le dimensioni del bruco, di mordersi la coda oppure di schiantarsi contro i bordi del tabellone di gioco. Per iniziare una nuova partita basta premere il tasto "NUOVO". Viene così resettato il tabellone di gioco, posizionando casualmente il bruco e le prime mele. Alla prima pressione dei uno dei tasti "UP", "DOWN", "LEFT" o "RIGHT" il gioco ha inizio, mettendo in movimento il bruco. Sul display in alto a destra vengono mostrati il punteggio raggiunto ed il livello.

Tutte le opzioni possono essere selezionate dal menu di sinistra. Il tasto "INFO" apre la finestra con le informazioni sulla versione, autore e copyright. Il tasto "HELP" mostra all'utente le istruzioni per giocare mentre i tasti "SI"/"NO" consentono di attivare/disattivare la musica di sottofondo (gli altri effetti audio rimangono comunque attivi). Il piccolo display posto al di sopra di tali tasti consente di monitorare la situazione in essere. I tasti "-"/"+" permettono, invece, di diminuire/aumentare la velocità del bruco. Infine, il tasto "ESCI" chiude il programma e riporta a Windows.

### 17.2 SUGGERIMENTI E TRUCCHI

Non ci sono particolari strategie per giocare a questo videogame. Unico trucco che si può consigliare è di prestare attenzione al bordo (evidenziato in verde) del campo da gioco, evitando, con l'aumentare delle dimensioni del bruco, di mordersi la coda oppure di schiantarsi contro i bordi del tabellone stesso.

### 17.3 LA REALIZZAZIONE (THE MAKING OF)

#### □ LA STORIA

Giunto alla sua terza versione nel 2003, con un design completamente rinnovato ed ispirato al tema "Metal" di "Guardie e ladri", questo videogame nasce nel 1999 e nonostante la sua semplicità e gli innumerevoli cloni presenti sui siti internet mondiali, riscuote un discreto successo (oltre 400 download).

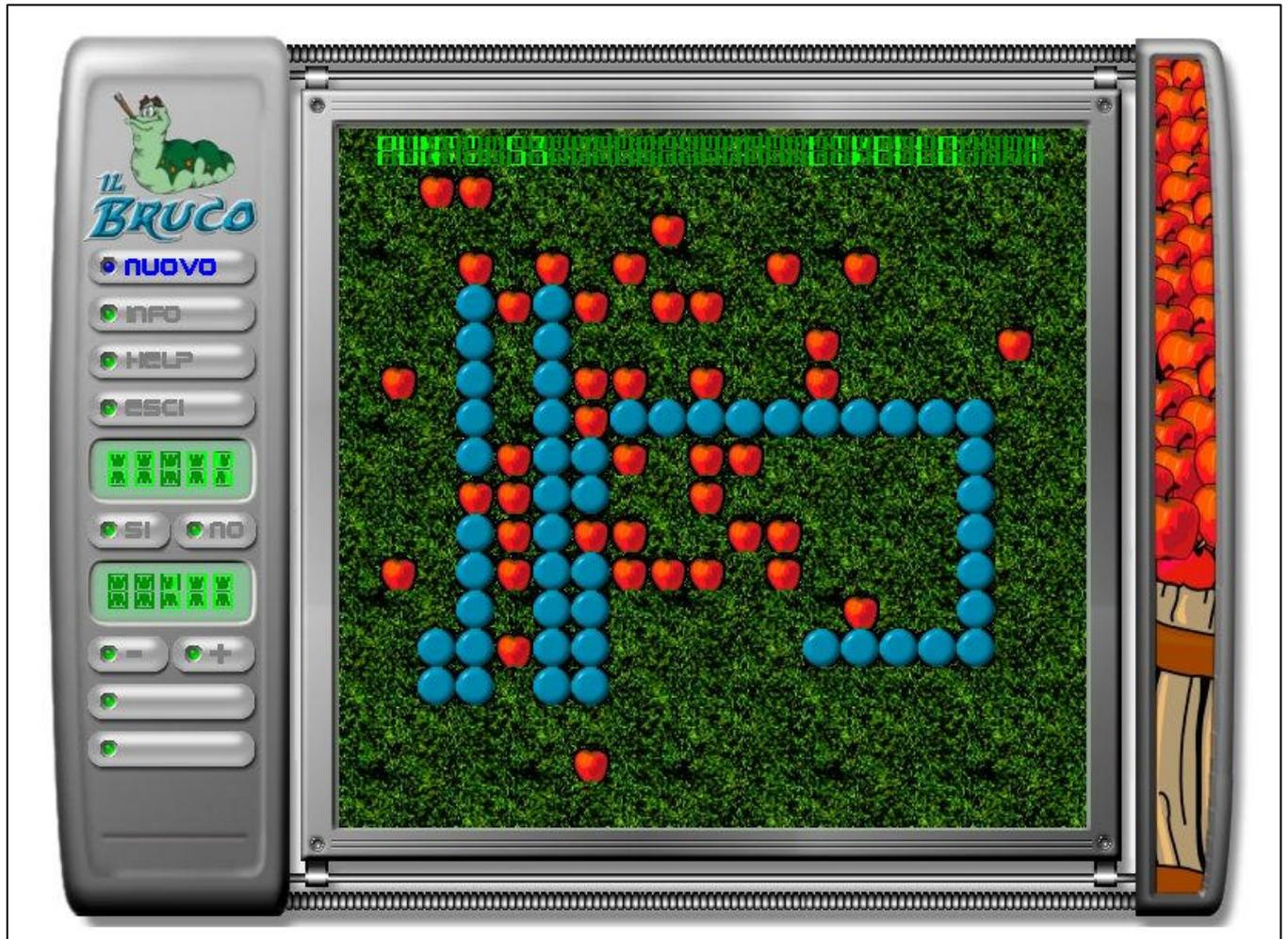


Fig. 17.1: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 17.6).
- *Grafica e Immagini*: Corel Photo Paint, Jasc Paint Shop Pro, Ulead Gif Animator, Ulead Photo Impact, Sierra Print Artist.
- *Audio e musica*: Xing Encoder, Mp3 GoWave Dj, Mp3 Compressor, GoldWave Studio.

#### □ BOZZE E SCREENSHOT

Ecco alcuni screenshot tratti dall'ultima versione del gioco. In fig. 17.2 un'immagine della schermata principale del videogame con la classifica dei migliori giocatori. In fig. 17.3 la finestra con le informazioni su autore e release del gioco. In fig. 17.4 c'è l' Help On Line con le istruzioni per giocare. In fig. 17.5: schermata di inserimento nominativo giocatore. Infine, in fig. 17.6 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.



Fig. 17.2: classifica dei tre migliori giocatori.



Fig. 17.3: Informazioni su autore e release del gioco.



Fig. 17.4: Help On Line - Istruzioni per giocare.



Fig. 17.5: inserimento nome del giocatore tra i migliori tre.

## 17.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Arial.ttf")
InstallFont("<Embedded>\BoomBox.ttf")
DisplayValue("Counter","Size")
DisplayValue("Counter1","Size")
MatrixSet("Matrix[0,0]","0")
for i=0 to 17
  col = RND(16)+1
  row=RND(16)+1
  MatrixSet("Matrix[col,row]","2")
next i
MatrixSet("Matrix[6,7]","1")
MatrixSet("Matrix[6,8]","1")
MatrixSet("Matrix[6,9]","1")
SnakeX[0] =6
SnakeX[1] =6
SnakeX[2] =6
SnakeY[0] =7
SnakeY[1] =8
SnakeY[2] =9
Size = 3
direct = 0
Speed = 300
MidiPlay("Music","loop")
Speed$=CHAR(Speed)
life$=CHAR(lf)
Level = 1
Level$ = CHAR(Level)
LoadText("Speed","Speed$")
LoadText("Level","Level$")
ScriptTimer("Script","500")

```

### □ Tasto NUOVO

```

if (jj=0) then
  score=0
  lf=5
end
jj=1
RunScript("Start")
NuovaPartita = 1
if (Audio=0) then
  WavePlay("Beep","")
end
Level = 1

```

### □ Tasto INFO ON

```

Show("Info2")
Hide("Come")
if (Audio=0) then
  WavePlay("Beep","")
end
Show("HotSpot10")

```

#### □ Tasto INFO OFF

```
Hide("Info2")
if (Audio=0) then
    WavePlay("Beep", "")
end
Hide("HotSpot10")
```

#### □ Tasto HELP ON

```
Show("Come")
Hide("Info2")
if (Audio=0) then
    WavePlay("Beep", "")
end
Show("HotSpot11")
```

#### □ Tasto HELP OFF

```
Hide("Come")
if (Audio=0) then
    WavePlay("Beep", "")
end
Hide("HotSpot11")
```

#### □ Tasto ATTIVA AUDIO

```
MidiPlay("Music", "loop")
Show("AudioON")
Hide("AudioOFF")
if (Audio=0) then
    WavePlay("Beep", "")
End
```

#### □ Tasto DISATTIVA AUDIO

```
MidiStop()
Hide("AudioON")
Show("AudioOFF")
if (Audio=0) then
    WavePlay("Beep", "")
end
```

#### □ Tasto SPEED MORE

```
Speed = Speed - 10
if (Speed < 0) then
    Speed = 100
end
Speed$=CHAR(Speed)
LoadText("Speed", "Speed$")
if (Audio=0) then
    WavePlay("Beep", "")
end
```

#### □ Tasto SPEED LESS

```
Speed = Speed + 10
Speed$=CHAR(Speed)
LoadText("Speed", "Speed$")
if (Audio=0) then
    WavePlay("Beep", "")
end
```

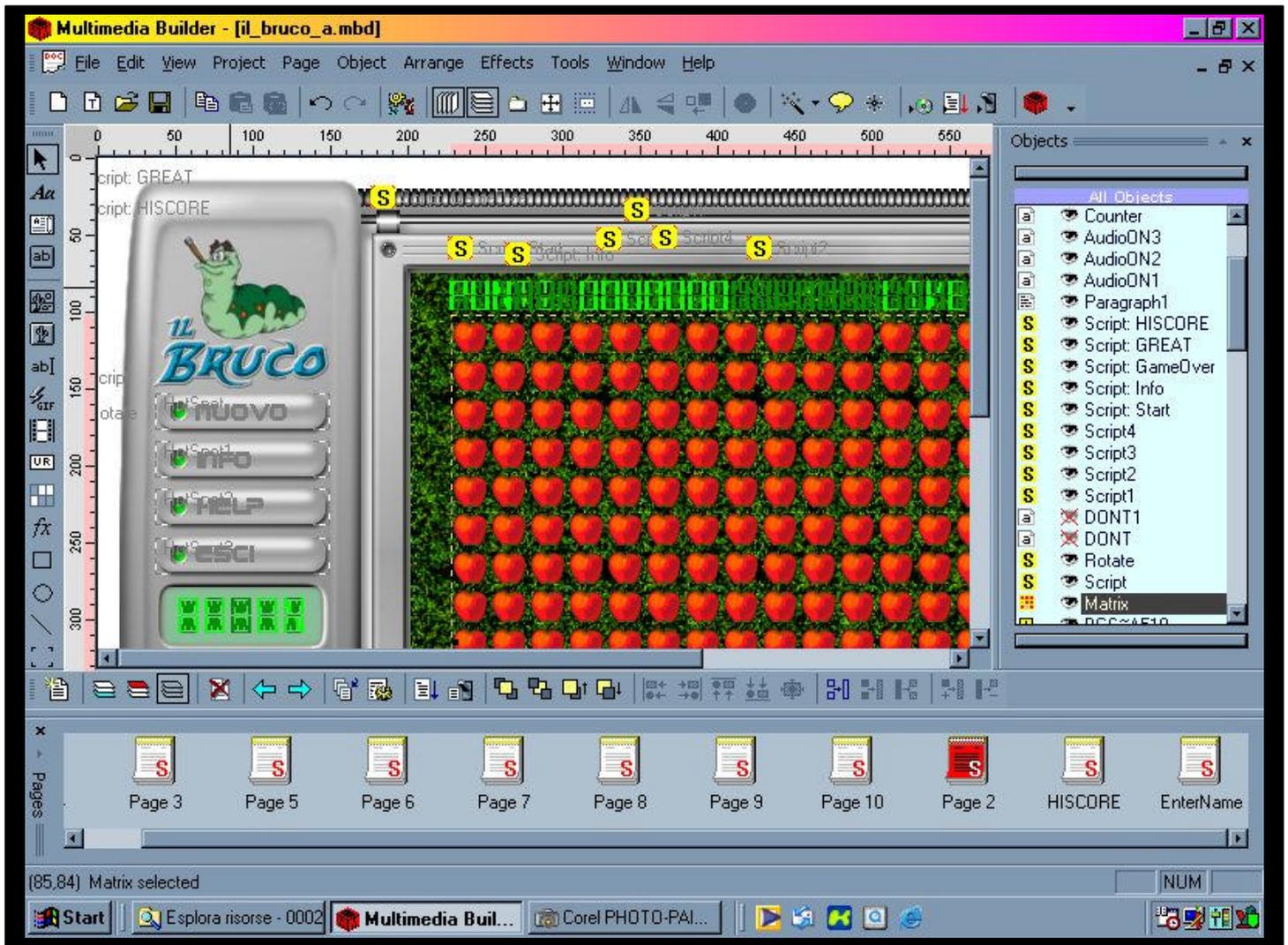


Fig. 17.6: Una schermata dell'ambiente di programmazione Mediachance Multimedia Builder.

#### ❑ Script Script

```

if (Size/Level = 1000) then
    Level = Level + 1
    Size = Size + 200 + Level
    Speed = Speed - 20
    Level$ = CHAR(Level)
    Speed$ = CHAR(Speed)
    LoadText ("Level", "Level$")
    LoadText ("Speed", "Speed$")
end
if (direct=0) then
    ScriptTimer ("Script", "500")
    Return()
end
if (direct=1) then
    RunScript ("Rotate")
    SnakeY[0]=SnakeY[0]-1
end
if (direct=2) then
    RunScript ("Rotate")
    SnakeY[0]=SnakeY[0]+1
end
if (direct=3) then
    RunScript ("Rotate")

```

```

    SnakeX[0]=SnakeX[0]-1
end
if (direct=4) then
    RunScript("Rotate")
    SnakeX[0]=SnakeX[0]+1
end
col = SnakeX[0]
row = SnakeY[0]
colE = SnakeX[Size]
rowE = SnakeY[Size]
MatrixGet("Matrix[col,row","idx")
if (idx=2) then
    Size = Size+1
    WavePlay("tone","")
    DisplayValue("Counter","Size")
    xx = RND(11)+1
    yy=RND(11)+1
    MatrixSet("Matrix[xx,yy]","2")
    xx = RND(11)+1
    yy=RND(11)+1
    MatrixSet("Matrix[xx,yy]","2")
end
if (idx=1) then
    WavePlay("ah","")
    Hide("Counter")
    Hide("Counter1")
    Show("KO1")
    ScriptTimer("Script: GameOver","2000")
    Return()
end
if (idx=-1) then
    WavePlay("ah1","")
    Hide("Counter")
    Hide("Counter1")
    Show("KO2")
    ScriptTimer("Script: GameOver","2000")
    Return()
end
MatrixSet("Matrix[col,row]","1")
MatrixSet("Matrix[colE,rowE]","0")
ScriptTimer("Script","Speed")

```

□ **Script Rotate**

```

for i=Size to 1
    SnakeX[i] = SnakeX[i-1]
    SnakeY[i] = SnakeY[i-1]
next i

```

□ **Script X**

```

Direct = X

```

□ **Script: Start**

```

Size=0
WavePlay("beep","")
Page("MainScreen")

```

□ **Script: GameOver**

```

SCORE=Size-3
if (SCORE>SCORE3) then
    ScriptTimer("Script: GREAT","1000")
    Return()
end
ScriptTimer("Script: HISCORE","1000")

```

❑ **Script: GREAT**

```
Page("EnterName")
```

❑ **Script: HIGHSCORE**

```
Page("HISCORE")
```

❑ **Main Script Page Highscore**

```
Show("Master Page::HotSpot: Info")
DisplayValue("Name1", "TopName1$")
DisplayValue("Score1", "SCORE1")
DisplayValue("Name2", "TopName2$")
DisplayValue("Score2", "SCORE2")
DisplayValue("Name3", "TopName3$")
DisplayValue("Score3", "SCORE3")
PageTimer("20000", "Page 2")
LoadText("Speed", "Speed$")
```

❑ **Main Script Page EnterName**

```
OK=0
types=0
maxchar=4
Status=0
Name$=' '
EditText$=' '
LoadText("NameText", "Name$")
WavePlay("Power", "")
PageTimer("30000", "HISCORE")
LoadText("Speed", "Speed$")
```

❑ **Script KeyCheck**

```
if (OK=1) then
    Page("HISCORE")
end
ScriptTimer("KeyCheck", "500")
```

❑ **Script ScriptOK**

```
WavePlay("Beep", "")
if (SCORE>SCORE1) then
    SCORE3=SCORE2
    TopName3$=TopName2$
    SCORE2=SCORE1
    TopName2$=TopName1$
    SCORE1=SCORE
    TopName1$=Name$
    RunScript("Updated")
    Return()
end
if (SCORE>SCORE2) then
    SCORE2=SCORE
    TopName2$=Name$
    RunScript("Updated")
    Return()
end
if (SCORE>SCORE3) then
    SCORE3=SCORE
    TopName3$=Name$
    RunScript("Updated")
    Return()
end
```

**□ Script Updated**

```
SaveVariable( "TopPlayer1" , "TopName1$" )
SaveVariable( "TopPlayer2" , "TopName2$" )
SaveVariable( "TopPlayer3" , "TopName3$" )
SaveVariable( "Score1" , "SCORE1" )
SaveVariable( "Score2" , "SCORE2" )
SaveVariable( "Score3" , "SCORE3" )
Page( "HISCORE" )
```

**□ Script ScriptA**

```
types=types+1
if ( types<maxchar ) then
  Name$=Name$+'A'
  LoadText( "NameText" , "Name$" )
end
```

**□ Script ScriptCLEAR**

```
Name$=' '
types=0
LoadText( "NameText" , "Name$" )
```

## 18. MICROTETRIS

- ❑ Versione 1.0 - Shareware.
- ❑ EVSoftware - Copyright © 2003 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.
- ❑ Il gioco originale Tetris è un prodotto commerciale depositato della società Elorg



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win98/2000/NT/XP
- ❑ Processore: Pentium II 200Mhz o superiore
- ❑ Memoria RAM: 32Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 800 Kb

### 18.1 REGOLE DEL GIOCO

MicroTetris è un divertente videogame di abilità e azione. Attraverso l'area di gioco una serie di mattoncini colorati cade dall'alto verso il basso, incastrandosi con gli altri precedentemente depositati sul fondo. Quando questi formano una riga completa di blocchi, quest'ultima scompare facendo scorrere verso il basso gli altri mattoni.

Dato che questo è l'unico modo per rimuovere i blocchi, il giocatore deve formare delle righe orizzontali di pezzi il più velocemente possibile.

E' consentito modificare l'orientamento dei pezzi solo mentre questi sono in fase di caduta. Possono essere ruotati e traslati orizzontalmente, nonché forzati a scendere più velocemente verso il fondo dell'area di gioco. Occorre utilizzare i tasti "UP", "DOWN", "LEFT" e "RIGHT" della tastiera.

Lo scopo del gioco consiste nel prolungare la partita per più tempo possibile, continuando ad accumulare punti. Se l'intera area di gioco risulta occupata dai mattoncini, il giocatore ha perso.

Per cominciare una nuova partita è sufficiente premere sul pulsante del menu a tendina denominato "MENU" e selezionare "NEW GAME". Per le informazioni su autore e release occorre selezionare "ABOUT" mentre per le istruzioni di gioco "HELP".

Per modificare il livello di gioco manualmente (in automatico si sale di livello con l'aumentare dei punti) basta servirsi degli appositi tasti "<" e ">". All'incremento del livello corrisponde un aumento della velocità di caduta dei mattoncini.

### 18.2 SUGGERIMENTI E TRUCCHI

L'area di gioco risulta particolarmente piccola. Se non si presta attenzione fin dall'inizio alla partita si rischia di occuparla velocemente perdendo il controllo del gioco. Conviene cercare di posizionare i pezzi in modo strategico, evitando di lasciare inutili gap "buchi" difficilmente colmabili durante il corso della partita.

In questa versione del videogame non è prevista un'anteprima del mattoncino che viene fatto cadere nello step successivo a quello in corso. Questo comporta che non si può pianificare in alcun modo la disposizione dei pezzi.

### 18.3 LA REALIZZAZIONE (THE MAKING OF)

#### □ LA STORIA

MicroTetris è il primo di una serie di videogame a basso costo (in termini di “*ore uomo*”) di cui fanno parte anche “*SpiderBuster*” e “*MiniFlippy*”. L’idea è stata quella di realizzare un’interfaccia generica componibile con il design di “Windows XP”, per poterla poi riutilizzare in maniera versatile e flessibile per diversi videogame. Design semplice, funzionale ma accattivante al tempo stesso, nonché delle “*Game Engine*” altrettanto ridotte. A differenza degli altri giochi della collezione “Gameisland”, si è scelto di utilizzare la lingua inglese per i menu, informazioni e istruzioni di gioco, al fine di garantirne una migliore diffusione internazionale.

MicroTetris rappresenta pertanto la versione basic del famoso videogame “Nintendo”. E’ stato realizzato in poco più di 5 ore di lavoro nell’Agosto del 2003. Non è stato reso disponibile per il download gratuito in internet, ma è stato inserito nella versione 7.0 di “Gameisland”.



Fig. 18.1: interfaccia principale del videogame.

#### □ I SOFTWARE UTILIZZATI

Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder (vedi fig. 18.6).
- *Grafica e Immagini*: Corel Photo Paint, Ulead Photo Impact.
- *Audio e musica*: da archivi midi e wav disponibili gratuitamente in internet.

#### □ BOZZE E SCREENSHOT

Nelle pagine seguenti sono stati riportati alcuni screenshot del videogame.

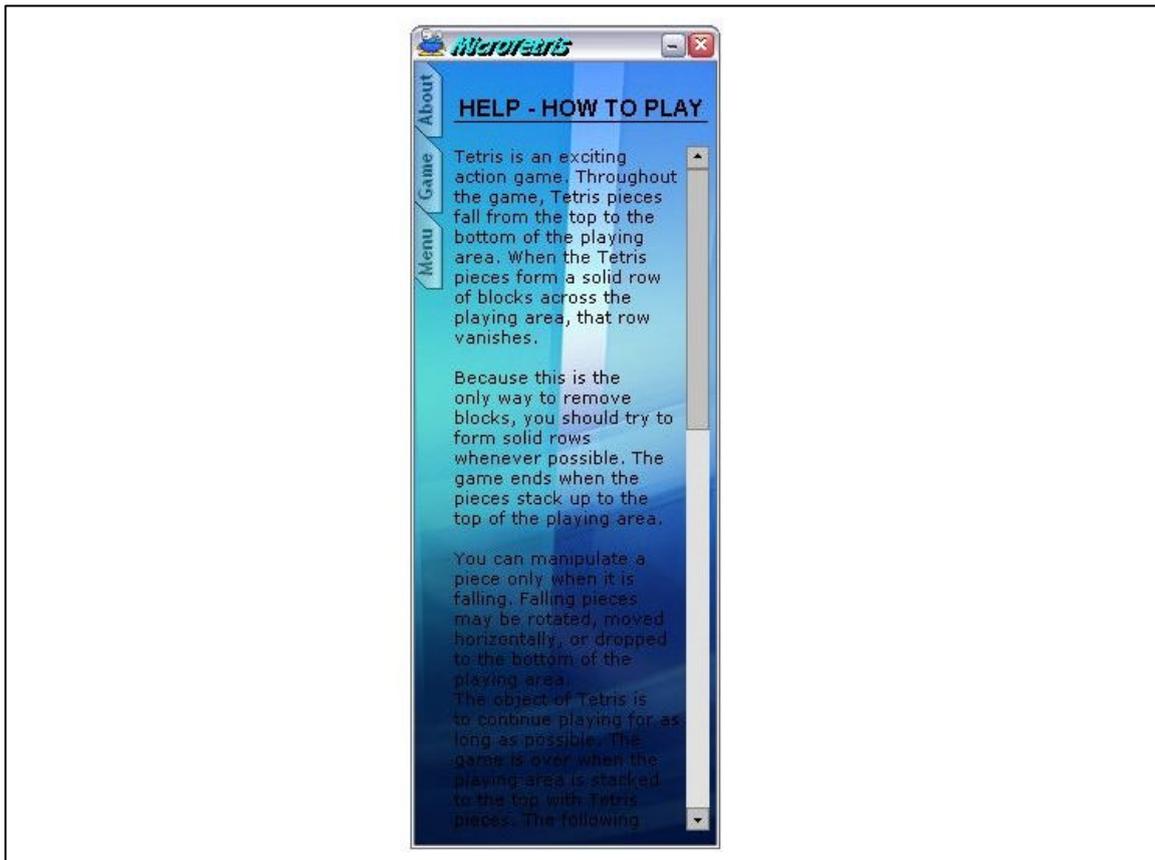


Fig. 18.2: Help On Line - Istruzioni per giocare.



Fig. 18.3: Informazioni su autore e release del gioco.



Fig. 18.4: la schermata di Gema Over.

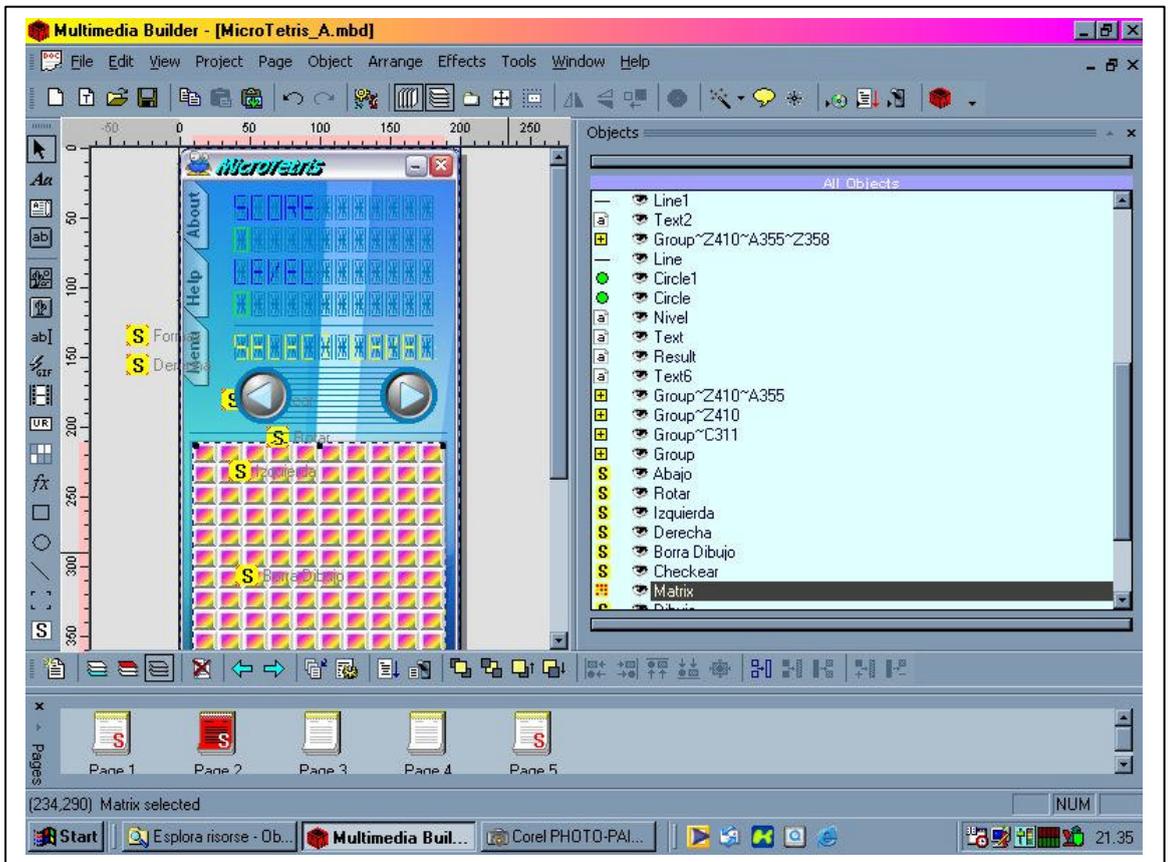


Fig. 18.5: l'ambiente di programmazione Mediachance Multimedia Builder.

## 18.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```
InstallFont("<Embedded>\Arial.ttf")
InstallFont("<Embedded>\Crille.ttf")
InstallFont("<Embedded>\Led_real.ttf")
Pause("100")
livello = 1
punteggio = 0
DisplayValue("Livello","livello")
DisplayValue("Result","punteggio")
RunScript("Game")
```

### □ Script Game

```
for gil=1 to 21
  tes=0
  for er=1 to 10
    MatrixGet("Matrix[er,gil]","k")
    tes=tes+k
  next er
  if (tes=10) then
    punteggio=punteggio+tes
    WavePlay("crollo","")
    for i=1 to gil-1
      y=-i+gil
      for er=1 to 10
        MatrixGet("Matrix[er,y]","k")
        MatrixSet("Matrix[er,y+1]","k")
      next er
    next i
    tes=0
    DisplayValue("Result","punteggio")
    juan=punteggio/200
    nabo=INT(juan)
    if (juan=nabo) then
      livello=livello+1
      if (livello=16) then
        livello =15
      end
    end
  end
end
next gil
forma=1
col=5
lin=1
block=RND(7)+1
if (block=1) then
  A[1]=1
  A[2]=1
  A[3]=1
  A[4]=1
  A[5]=0
  A[6]=0
  A[7]=0
  A[8]=0
end
if (block=2) then
  A[1]=1
  A[2]=1
  A[3]=1
```

```
A[4]=0
A[5]=0
A[6]=0
A[7]=1
A[8]=0
end
if (block=3) then
  A[1]=0
  A[2]=1
  A[3]=1
  A[4]=1
  A[5]=0
  A[6]=1
  A[7]=0
  A[8]=0
end
if (block=4) then
  A[1]=1
  A[2]=1
  A[3]=0
  A[4]=0
  A[5]=0
  A[6]=1
  A[7]=1
  A[8]=0
end
if (block=5) then
  A[1]=0
  A[2]=1
  A[3]=1
  A[4]=0
  A[5]=1
  A[6]=1
  A[7]=0
  A[8]=0
end
if (block=6) then
  A[1]=1
  A[2]=1
  A[3]=1
  A[4]=0
  A[5]=0
  A[6]=1
  A[7]=0
  A[8]=0
end
if (block=7) then
  A[1]=0
  A[2]=1
  A[3]=1
  A[4]=0
  A[5]=0
  A[6]=1
  A[7]=1
  A[8]=0
end
nofill=0
RunScript("Controllo")
if (nofill=0) then
  RunScript("Schermo")
  RunScript("Sotto")
end
```

### □ Script Schermo

```

if (forma=1) then
  for i=1 to 4
    if (A[i]<>0) then
      MatrixSet ("Matrix[i+col-1,lin]", "A[i]")
    end
  next i
  for i=5 to 8
    if (A[i]<>0) then
      MatrixSet ("Matrix[i-5+col,lin+1]", "A[i]")
    end
  next i
end
if (forma=3) then
  for i=5 to 8
    w=-i+9
    if (A[i]<>0) then
      MatrixSet ("Matrix[w-1+col,lin]", "A[i]")
    end
  next i
  for i=1 to 4
    w=-i+5
    if (A[i]<>0) then
      MatrixSet ("Matrix[w+col-1,lin+1]", "A[i]")
    end
  next i
end
if (forma=4) then
  for i=1 to 4
    w=-i+5
    if (A[i]<>0) then
      MatrixSet ("Matrix[col,w-1+lin]", "A[i]")
    end
  next i
  for i=5 to 8
    w=-i+9
    if (A[i]<>0) then
      MatrixSet ("Matrix[col+1,w-1+lin", "A[i]")
    end
  next i
end
if (forma=2) then
  for i=5 to 8
    if (A[i]<>0) then
      MatrixSet ("Matrix[col,i-5+lin]", "A[i]")
    end
  next i
  for i=1 to 4
    if (A[i]<>0) then
      MatrixSet ("Matrix[col+1,i-1+lin]", "A[i]")
    end
  next i
end

```

### □ Script Controllo

```

if (forma=1) then
  for i=1 to 4
    if (A[i]<>0) then
      MatrixGet ("Matrix[i+col-1,lin]", "b")
      if (b>0) then
        nofill=nofill+1
      end
    end
  next i
end

```

```

next i
for i=5 to 8
  if (A[i]<>0) then
    MatrixGet("Matrix[i-5+col,lin+1]", "b")
    if (b<>0) then
      nofill=nofill+1
    end
  end
end
next i
end
if (forma=3) then
  for i=5 to 8
    w=-i+9
    if (A[i]<>0) then
      MatrixGet("Matrix[w-1+col,lin]", "b")
      if (b<>0) then
        nofill=nofill+1
      end
    end
  end
next i
for i=1 to 4
  w=-i+5
  if (A[i]<>0) then
    MatrixGet("Matrix[w+col-1,lin+1]", "b")
    if (b<>0) then
      nofill=nofill+1
    end
  end
end
next i
end
if (forma=4) then
  for i=1 to 4
    w=-i+5
    if (A[i]<>0) then
      MatrixGet("Matrix[col,w-1+lin]", "b")
      if (b<>0) then
        nofill=nofill+1
      end
    end
  end
next i
for i=5 to 8
  w=-i+9
  if (A[i]<>0) then
    MatrixGet("Matrix[col+1,w-1+lin]", "b")
    if (b<>0) then
      nofill=nofill+1
    end
  end
end
next i
end
if (forma=2) then
  for i=5 to 8
    if (A[i]<>0) then
      MatrixGet("Matrix[col,i-5+lin]", "b")
      if (b<>0) then
        nofill=nofill+1
      end
    end
  end
next i
for i=1 to 4
  if (A[i]<>0) then
    MatrixGet("Matrix[col+1,i-1+lin]", "b")
    if (b<>0) then
      nofill=nofill+1
    end
  end
end

```

```

    end
  next i
end

```

### □ Script Zeta Schermo

```

if (forma=1) then
  for i=1 to 4
    if (A[i]<>0) then
      MatrixSet("Matrix[i+col-1,lin]", "0")
    end
  next i
  for i=5 to 8
    if (A[i]<>0) then
      MatrixSet("Matrix[i-5+col,lin+1]", "0")
    end
  next i
end
if (forma=3) then
  for i=5 to 8
    w=-i+9
    if (A[i]<>0) then
      MatrixSet("Matrix[w-1+col,lin]", "0")
    end
  next i
  for i=1 to 4
    w=-i+5
    if (A[i]<>0) then
      MatrixSet("Matrix[w+col-1,lin+1]", "0")
    end
  next i
end
if (forma=4) then
  for i=1 to 4
    w=-i+5
    if (A[i]<>0) then
      MatrixSet("Matrix[col,w-1+lin]", "0")
    end
  next i
  for i=5 to 8
    w=-i+9
    if (A[i]<>0) then
      MatrixSet("Matrix[col+1,w-1+lin", "0")
    end
  next i
end
if (forma=2) then
  for i=5 to 8
    if (A[i]<>0) then
      MatrixSet("Matrix[col,i-5+lin]", "0")
    end
  next i
  for i=1 to 4
    if (A[i]<>0) then
      MatrixSet("Matrix[col+1,i-1+lin]", "0")
    end
  next i
end
end

```

### □ Script Derecha

```

RunScript("Zeta Schermo")
nofill=0
col=col+1
RunScript("Controllo")
if (nofill=0) then

```

```

    RunScript("Schermo")
end
if (nofill<>0) then
    col=col-1
    RunScript("Schermo")
end

```

#### □ Script Sinistra

```

RunScript("Zeta Schermo")
nofill=0
col=col-1
RunScript("Controllo")
if (nofill=0) then
    RunScript("Schermo")
end
if (nofill<>0) then
    col=col+1
    RunScript("Schermo")
end

```

#### □ Script Rotar

```

RunScript("Zeta Schermo")
nofill=0
forma=forma+1
if (forma=5) then
    forma=1
end
RunScript("Controllo")
if (nofill=0) then
    RunScript("Schermo")
end
if (nofill<>0) then
    forma=forma-1
    if (forma=0) then
        forma=4
    end
end
end

```

#### □ Script Sotto

```

if (livello=0) then
    tempo=550
end
if (livello=1) then
    tempo=500
end
if (livello=2) then
    tempo=450
end
if (livello=3) then
    tempo=400
end
if (livello=4) then
    tempo=350
end
if (livello=5) then
    tempo=300
end
if (livello=6) then
    tempo=250
end
if (livello=7) then
    tempo=200
end
end

```

```

if (livello=8) then
    tempo=170
end
if (livello=9) then
    tempo=150
end
if (livello=10) then
    tempo=130
end
if (livello=11) then
    tempo=110
end
if (livello=12) then
    tempo=100
end
if (livello=13) then
    tempo=90
end
if (livello=14) then
    tempo=70
end
if (livello=15) then
    tempo=50
end
DisplayValue ("Livello", "livello")
ScriptTimer ("Sotto", "tempo")
RunScript ("Zeta Schermo")
nofill=0
lin=lin+1
RunScript ("Controllo")
if (nofill=0) then
    RunScript ("Schermo")
end
if (nofill<>0) then
    lin=lin-1:
    RunScript ("Schermo")
    if (lin=1) then
        MatrixSet ("Matrix[0,0]", "0")
        Page ("Page 5")
    end
    RunScript ("Game")
end
end

```

## 19. SPIDERBUSTER

- ❑ Versione 1.0 - Shareware.
- ❑ EVSoftware - Copyright © 2003 - Ideato e programmato da Emiliano Vezzoli.
- ❑ Tutti i diritti sono riservati.



### CONFIGURAZIONE DI SISTEMA RICHIESTA:

- ❑ Sistema operativo richiesto: Win98/2000/NT/XP
- ❑ Processore: Pentium III 300Mhz o superiore
- ❑ Memoria RAM: 64Mb o superiore
- ❑ Scheda Audio: 16 bit o superiore
- ❑ Risoluzione video consigliata: 800x600 Pixel
- ❑ Spazio occupato su HD: 1,9 Mb

### 19.1 REGOLE DEL GIOCO

Il giocatore veste i panni di un cacciatore e deve percorrere l'itinerario proposto sul tabellone, catturando i 6 ragni situati nelle corrispettive stazioni. Il raggiungimento di una stazione fa illuminare ed attivare il tratto di percorso utilizzato per raggiungerla, guadagnando 100 punti.

Per raggiungere una stazione e catturare il ragno è necessario servirsi del puntatore mouse, trascinandolo lentamente lungo la pista irregolare descritta sul tabellone di gioco. Ogni volta che si esce col mouse dalla pista si perdono 50 punti e tutto il percorso viene disattivato, costringendo il giocatore a ricominciare dalla prima stazione. Per vincere si devono attivare tutte le stazioni, catturando i 6 ragni.

Per rendere la partita più frenetica e avvincente è stato introdotto un sistema a timer: il giocatore ha infatti 3 minuti di tempo per completare il gioco. Il tempo a disposizione viene mostrato a display e graficamente da una barra verticale di colore verde.

Per cominciare una nuova partita è sufficiente premere sul pulsante del menu a tendina denominato "MENU" e selezionare "NEW GAME". Per le informazioni su autore e release occorre selezionare "ABOUT" mentre per le istruzioni di gioco "HELP".

### 19.2 SUGGERIMENTI E TRUCCHI

Trattandosi di un gioco a tempo è indispensabile non farsi prendere dal panico, vedendo l'inesorabile spegnimento dei led del display verde, secondo dopo secondo. Il giocatore deve mantenere la calma, non prestare attenzione al tempo che passa ma, al contrario, mantenere ben ferma la mano nella fase di trascinamento del mouse.

E' utile, dopo aver raggiunto una delle 6 stazioni, far riposare i muscoli della mano, inevitabilmente tesi.

E' ovvio che per poter giocare a SpiderBuster è indispensabile possedere un puntatore mouse in perfetta efficienza. Anomalie o malfunzionamenti del dispositivo, infatti, rendono impossibile le manovre delicate di trascinamento richieste per poter giocare.

### 19.3 LA REALIZZAZIONE (THE MAKING OF)

#### □ LA STORIA

SpiderBuster è il secondo di una serie di videogame a basso costo (in termini di “*ore uomo*”) di cui fanno parte anche “*MicroTetris*” e “*MiniFlippy*”. L’idea è stata quella di realizzare un’interfaccia generica componibile con il design di “Windows XP”, per poterla poi riutilizzare in maniera versatile e flessibile per diversi videogame. Design semplice, funzionale ma accattivante al tempo stesso, nonché delle “*Game Engine*” altrettanto ridotte. A differenza degli altri giochi della collezione “Gameisland”, si è scelto di utilizzare la lingua inglese per i menu, informazioni e istruzioni di gioco, al fine di garantirne una migliore diffusione internazionale.

SpiderBuster vede la luce nell’Agosto del 2003, con il solo obiettivo di implementare un semplice “meccanismo” di percorsi attivi/disattivi gestiti tramite il puntatore mouse. Non si tratta, quindi, di un vero e proprio videogioco. Non esiste una “*Game Engine*” di tipologia classica, perché non necessaria. Rappresenta, invece, un buon esempio di come realizzare un prodotto multimediale che obbliga l’utente a servirsi del mouse per interagire con il software.

Il tema dei ragni è stato scelto quasi casualmente. In principio si era pensato di utilizzare un cartone animato famoso, come “*Goldrake*” o “*Mazinga*”. Il seguito il tema “*Ghost*” (fantasmi) era apparso più accattivante ma “*Ghostbuster*” era un nome troppo sfruttato e così al posto dei fantasmi sono comparsi i ragni. Non è stato reso disponibile per il download gratuito in internet, ma è stato inserito nella versione 7.0 di “*Gameisland*”.

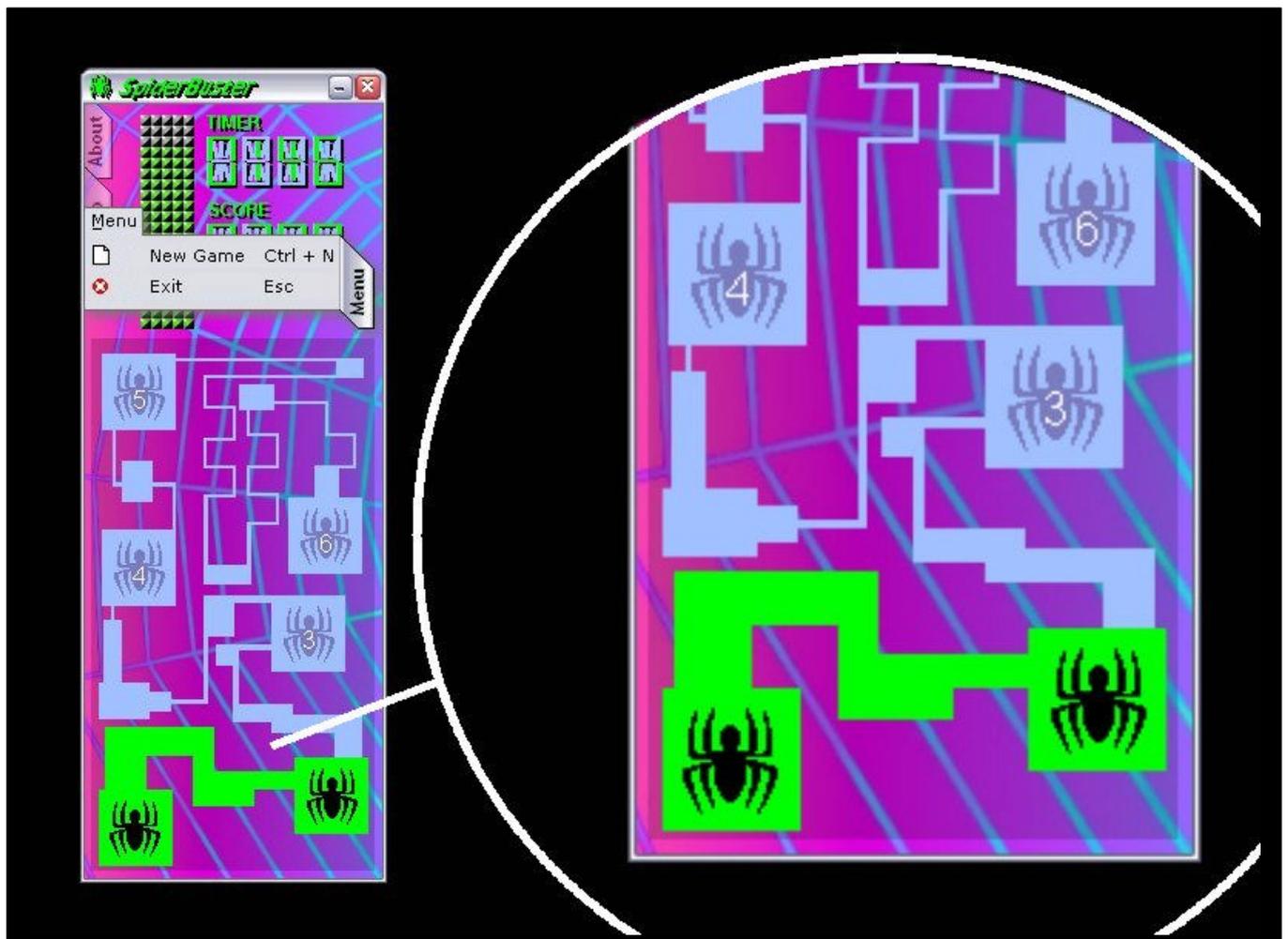


Fig. 19.1: interfaccia principale del videogame.

## □ I SOFTWARE UTILIZZATI

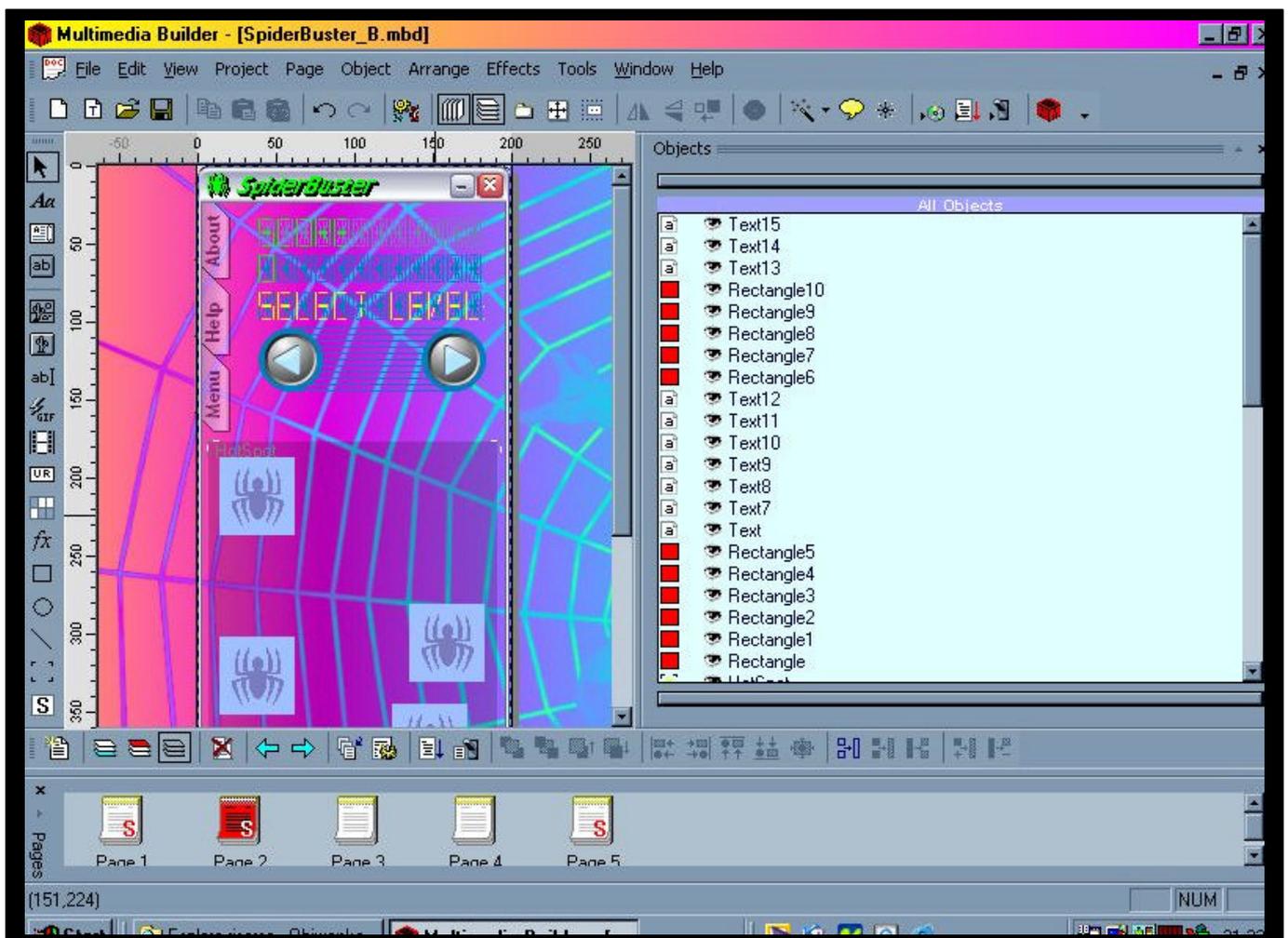
Di seguito sono riportati i software utilizzati per la realizzazione del videogioco:

- *Game Engine*: Mediachance Multimedia Builder 4.9 (vedi fig. 19.2).
- *Grafica e Immagini*: Corel Photo Paint 8, Ulead Photo Impact 4.2, Paint Shop Pro 6.0.
- *Audio e musica*: da archivi midi e wav disponibili gratuitamente in internet.

## □ BOZZE E SCREENSHOT

Nelle pagine seguenti sono stati riportati alcuni screenshot del videogame. In fig. 19.1 l'interfaccia principale del videogame con il percorso di gioco. In fig. 19.2 una schermata dell'ambiente di programmazione Mediachance Multimedia Builder 4.9.

Nelle figg. 19.3 e 19.4 ci sono, rispettivamente, le finestre con le istruzioni di gioco e le informazioni su autore, release e copyright. Nelle figg. 19.5 e 19.6 le schermate di game over in caso di sconfitta e di vittoria del giocatore.



**Fig. 19.2:** l'ambiente di programmazione Mediachance Multimedia Builder.

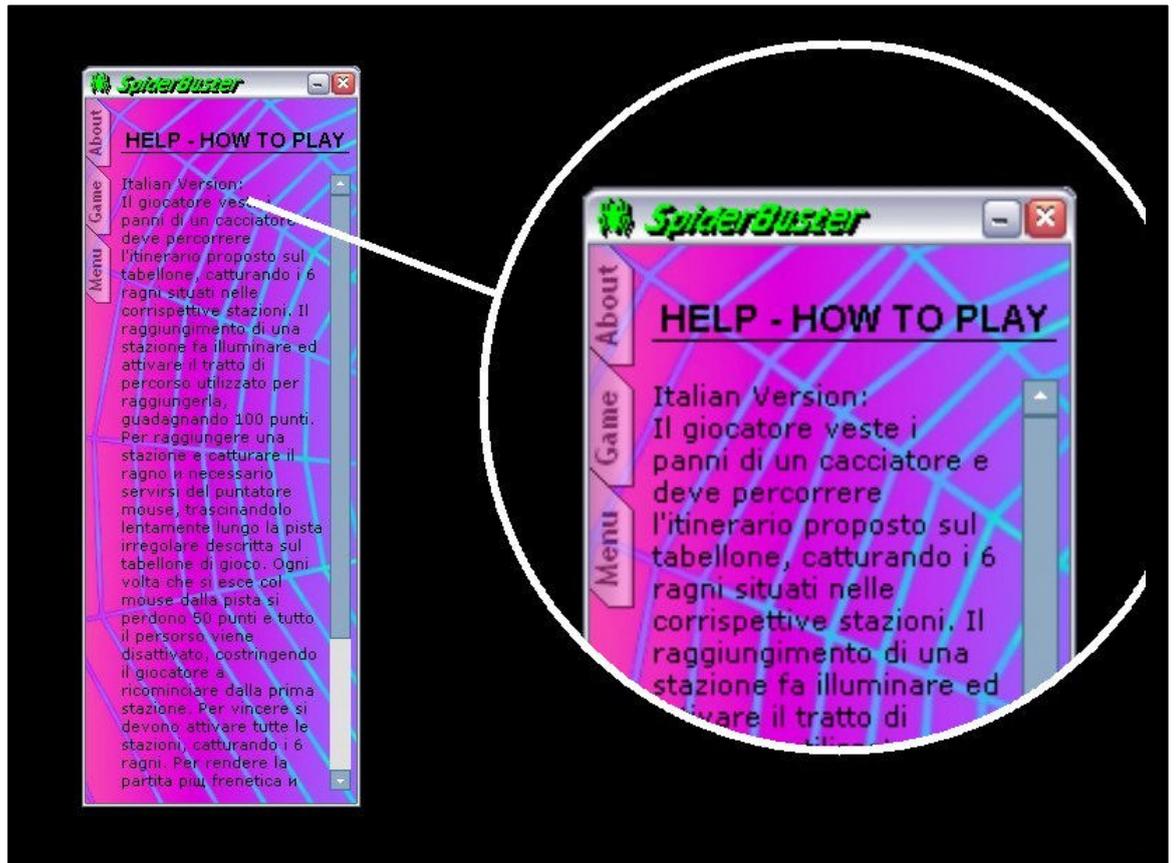


Fig. 19.3: Help On Line - Istruzioni per giocare.



Fig. 19.4: informazioni su autore e release del gioco.



Fig. 19.5: la schermata di Game Over.



Fig. 19.6: la schermata di Game Over, in caso di vittoria.

## 19.4 IL CODICE

Di seguito vengono riportati i codici sorgente degli script associati agli oggetti principali del videogame.

### □ Main Page

```

InstallFont("<Embedded>\Arial.ttf")
InstallFont("<Embedded>\Crille.ttf")
InstallFont("<Embedded>\Led_real.ttf")
InstallFont("<Embedded>\Webdings.ttf")
MatrixSet("MatrixTimer[0,0]", "1")
Punteggio = 0
Stadio = 0
Tempo = 180
Timer = Tempo*1000
Sec = Tempo
riga = 0
ContaParziale = 0
if (Sec<10 & Sec>=0) then
    Secondi$ = '0000' + CHAR(Sec)
end
if (Sec<100 & Sec>=10) then
    Secondi$ = '000' + CHAR(Sec)
end
if (Sec<1000 & Sec>=100) then
    Secondi$ = '00' + CHAR(Sec)
end
if (Sec<10000 & Sec>=1000) then
    Secondi$ = '0' + CHAR(Sec)
end
if (Punteggio<10 & Punteggio>=0) then
    Punteggio$ = '0000' + CHAR(Punteggio)
end
if (Punteggio<100 & Punteggio>=10) then
    Punteggio$ = '000' + CHAR(Punteggio)
end
if (Punteggio<1000 & Punteggio>=100) then
    Punteggio$ = '00' + CHAR(Punteggio)
end
if (Punteggio<10000 & Punteggio>=1000) then
    Punteggio$ = '0' + CHAR(Punteggio)
end
LoadText("TextTimer", "Secondi$")
LoadText("Punti", "Punteggio$")
ScriptTimer("Timer", "2000")

```

### □ HotSpot AttivaStazione X

```

if (Stadio = 0) then
    Stadio = 1
    Show("Zone_X_Active")
    Punteggio = Punteggio + 100
    DisplayValue("Punti", "Punteggio")
    WavePlay("vinto", "")
end

```

### □ HotSpot Disattiva Zone

```

if (Stadio <> 0) then
  Stadio = 0
  Hide("Zone_Attive")
  WavePlay("perso", "")
  Punteggio = Punteggio - 50
  DisplayValue("Punti", "Punteggio")
end

```

### □ Script Timer

```

Timer = Timer - 1000
Sec = Sec - 1
ContaParziale = ContaParziale + 1000
if (ContaParziale=9000) then
  ContaParziale = 0
  riga = riga + 1
  MatrixSet("MatrixTimer[1,riga]", "2")
  MatrixSet("MatrixTimer[2,riga]", "2")
  MatrixSet("MatrixTimer[3,riga]", "2")
  MatrixSet("MatrixTimer[4,riga]", "2")
  MatrixSet("MatrixTimer[5,riga]", "2")
end
if (Sec<10 & Sec>=0) then
  Secondi$ = '0000' + CHAR(Sec)
end
if (Sec<100 & Sec>=10) then
  Secondi$ = '000' + CHAR(Sec)
end
if (Sec<1000 & Sec>=100) then
  Secondi$ = '00' + CHAR(Sec)
end
if (Sec<10000 & Sec>=1000) then
  Secondi$ = '0' + CHAR(Sec)
end
if (Punteggio<10 & Punteggio>=0) then
  Punteggio$ = '0000' + CHAR(Punteggio)
end
if (Punteggio<100 & Punteggio>=10) then
  Punteggio$ = '000' + CHAR(Punteggio)
end
if (Punteggio<1000 & Punteggio>=100) then
  Punteggio$ = '00' + CHAR(Punteggio)
end
if (Punteggio<10000 & Punteggio>=1000) then
  Punteggio$ = '0' + CHAR(Punteggio)
end
LoadText("TextTimer", "Secondi$")
LoadText("Punti", "Punteggio$")
if ( Sec>0 ) then
  ScriptTimer("Timer", "1000")
end
if ( Sec<=0 ) then
  Page("Page 5")
end

```

### □ Script Tasto NewGame

```

MoveTo("Group menu", "-175, 84, 20, EASYFROM")
Show("menu")
Hide("Group menu")
Hide("GameOver")
Page("Page 2")

```

**□ Script HaiVinto**

`Page( "Page 6" )`

**□ Script Tasto ChiudiMenu**

`MoveTo( "Group menu", "-175, 84, 20, EASYFROM" )`

`Show( "menu" )`

`Hide( "Group menu" )`