

Una favola per la rete

Digerito l'HTML ed acquisita un po' di confidenza con JavaScript, ci siamo avventurati nella realizzazione di un prodotto ipermediale per la rete.

Nello specifico, come dice il titolo, si tratta di una favola (Biancaneve e l'Aritmetica!) ristrutturata in maniera che le scene principali si colleghino in qualche maniera con alcuni momenti didattici.

Piu' in generale, piaccia o meno il soggetto, è comunque il raggiungimento dello scopo prefissoci -realizzare un prodotto ipermediale per il Web!- che ci pare giusto evidenziare. Invitandovi pertanto a fare "un giro" su C&Vweb n. 10 (dove troverete già in linea la versione demo della favola) in questo articolo cercheremo di analizzare la serie di risvolti tecnici che hanno caratterizzato la fase realizzativa del prodotto.

di Bruno Rosati

Biancaneve e l'Aritmetica: è questo il titolo imposto alla fatica estiva che il sottoscritto, grazie pure al contributo del Webmaster Marco Tau che ringrazio pubblicamente per la preziosa supervisione, è riuscito a compiere.

Nel prodotto, pensato per la rete, ma ovviamente giocabile anche in locale, ci sono molte cose interessanti da evidenziare. A partire dall'uso massiccio di vari costrutti JavaScript (counter, procedure di caricamento casuale, controlli su "click-event", ecc.) per arrivare alle "trovate" poste alla base di ogni pagina di gioco. Lo scopo di tutto ciò è quello di arrivare a dimostrare quanto è possibile realizzare d'ipermediale per il Web.

Vediamo insieme su queste pagine (e su quelle telematiche di C&Vweb n. 10) cosa siamo stati capaci di realizzare.

La struttura principale

L'idea di base era quella di prendere la favola, dividerla per scene (scritte, recitate vocalmente come estratte, previo liberatoria, da qualche film) ed inserire, tra una scena e l'altra,

dei moduli di gioco.

I moduli di gioco a loro volta, potevano essere strettamente attinenti ai contenuti della scena (com'è nei moduli "A domanda ... rispondi!", l'Anagramma e il Paroliere... con i quali, saltando su C&Vweb n. 10, potrete interagire) oppure esulare completamente. In quest'ultimo caso, idea nell'idea, si è pensato di realizzare un gioco aritmetico. Una sorta di espediente didattico con il quale "interrogare" il concorrente per verificare le sue capacità di calcolo (le quattro operazioni servite in tutte le salse!) ed il livello di deduzione logica applicata alla risoluzione di alcuni semplici problemi (tipo:

Al mercato la mamma ha comprato...).

Letta, vista o sentita ogni singola scena (nel demo ne abbiamo inserite cinque) parte la pagina di gioco. Per ciascun gioco (ne abbiamo messi a punto ben sei) si è cercato di realizzare almeno tre varianti, in modo che la giocabilità della favola interattiva risultasse più varia e prolungata nel tempo. In ogni gioco bisogna dare risposte (digitandole direttamente oppure cliccando su di una serie di bottoni). Le risposte giuste devono

dare dei punti, quelle sbagliate delle penalità (tentativi).

Da questa rapida elencazione dei criteri stabiliti, deriva immediatamente l'intera lista delle cose che si è dovuto fare, sia a livello di codice che a livello di creatività multimediale.

Dal punto di vista del codice ciò che andava realizzato, era una serie di controlli in grado di gestire:

- un totalizzatore di punti (risposte giuste) e di tentativi (risposte sbagliate);
- il caricamento casuale delle varianti previste (tre) per ciascun modulo di gioco ideato;
- la verifica delle risposte date (via mouse o digitate in campi prefissati).

Questo, ovviamente, in generale. Nel particolare sono state realizzate almeno una dozzina tra funzioni, variabili ed array per le specificità portate dall'ideazione di ogni singolo gioco.

A sua volta, dal punto di vista multimediale, lo sviluppo della favola ha richiesto la creazione di:

- una serie di GIF animate (alcune puramente estetiche, altre in funzione del gioco e quindi con controlli più particolarizzati sulle temporizzazioni delle sin-

gole immagini che le compongono);
 - dei brani MIDI (tagliati nella giusta durata richiesta oppure mandati in loop. In quest'ultimo caso facendo attenzione ad istruire la page ai rispettivi TAG usati da Explorer e Navigator;
 - delle acquisizioni audio (in WAV-True-Speech) ed altre video (dei single-frame da convertire poi in GIF);
 - una serie di titoli, logo ed altri clip di varia natura.
 Cosa si è utilizzato per realizzare tutto ciò?

Per quanto riguarda il codice è parso subito naturale fare ricorso al JavaScript, con il quale (tra variabili, funzioni, gestori di eventi mouse/tastiera, ecc.) si è integrata la scarsa dinamicità dell'HTML. Per quel che concerne il supporto multimediale, oltre all'irrinunciabile Paint Shop Pro, si è ricorsi ai servizi del Microsoft GIF Animator per quanto riguarda il montaggio delle GIF animate; al Microsoft Music Producer per elaborare i brani MIDI ed al semplice, ma prezioso Sound Recorder di Win95 (l'unico, che io sappia, che consente di acquisire ed editare il Wav-True-Speech).
 Codice JavaScript da un lato ed appli-

Figura 1 - L'home page di "Biancaneve e l'Aritmetica" presente all'indirizzo telematico di C&Vweb n. 10.

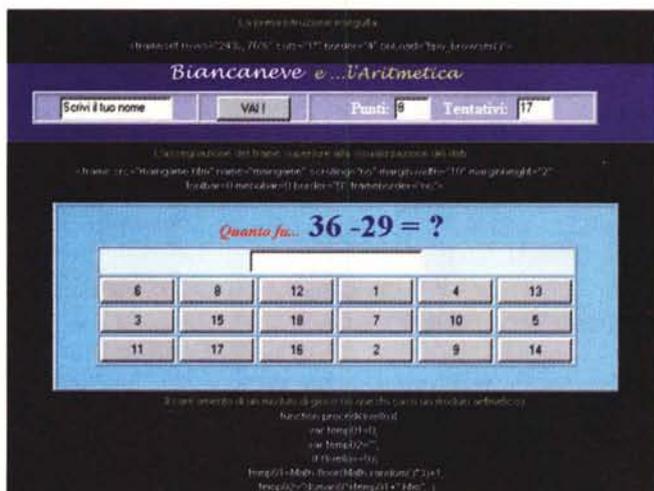


Figura 2 - La struttura a cornici della pagina di gioco. Nel frame superiore i dati variabili (nome, punti, tentativi); in quello inferiore le varie situazioni di gioco.

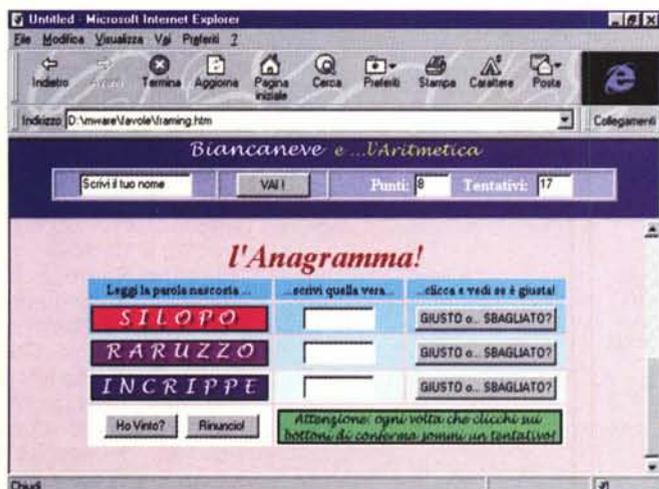


Figura 3 - Il modulo di gioco dell'Anagramma.

cativi multimediali dall'altro, vediamo allora un po' più nei particolari cosa e come si è dovuto realizzare. Lo spazio, come al solito tiranno, non ci consente che rapidi excursus.

L'organizzazione delle pagine

L'impostazione data ai criteri interattivi della favola ha subito richiesto d'organizzare le pagine su due distinti frame. In tal modo ci siamo ritrovati a poter disporre di una prima cornice costantemente dedicata alla zona dei dati (i campi di testo relativi all'eventuale inserimento del nome, il contatempo e l'aggiornamento di punti e tentativi) ed una seconda cornice, più ampia della prima, a disposizione delle scene o dei giochi. Tale impostazione è schematizzata in figura 2.

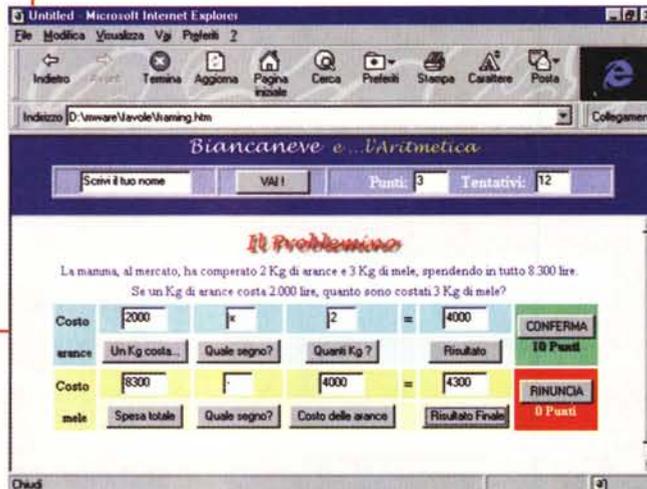
Nel particolare, la struttura così derivata viene gestita attraverso la page principale (identificata come framing.htm) nella quale sono settati tutti i parametri che governeranno le pagine. Per avere un'idea sul tipo di controlli applicati alla struttura delle pagine è sufficiente osservare la tabella 1. In questa è possibile rilevare la presenza di tutti i TAG indispensabili per stabilire il settaggio del numero di righe/colonne e per l'assegnazione delle cornici.

Nello specifico la cornice superiore, quella pari al 24% e denominata "main-game", è stata assegnata al caricamento stabile della pagina principale, la



Figura 4 - La struttura schematizzata posta "dietro" la tabella del gioco de L'Anagramma. Per avere un immediato riscontro delle funzionalità di ogni controllo è bene osservare contemporaneamente anche la tabella n. 2.

Figura 5 - Terza variabile della parte "didattica" della favola interattiva è questo momento aritmetico incentrato sulla risoluzione di alcuni "problemcini". Come appare evidente, oltre ai dati ed ai campi dove digitare le proprie risposte, c'è una serie di bottoni di controllo. Cliccarci sopra costa sempre un tentativo, ma non abbandona mai il "piccolo giocatore"!



"maingame.htm", appunto dedicata alla visualizzazione della serie dei campi di controllo.

Quella inferiore, d'ampiezza pari al 76%, è stata a sua volta assegnata alla visualizzazione di tutte le altre pagine (di

scena e di gioco) che compongono la favola.

Per assegnare tali posizioni s'è fatto ricorso all'usuale TAG:

<FRAME SRC="nome_del_file" name="nome_della_cornice"> comprensivo dei controlli su margini, possibilità di scrolling e presenza o meno delle barre del menu e di stato.

Sempre comprese tra i TAG <SCRIPT>... </SCRIPT>, nell'head dello stesso file principale "framing.htm", sono presenti le istruzioni JavaScript relative alla definizione delle variabili globali (che controlleranno il numero dei punti e dei tentativi), la funzione per il caricamento random delle pagine dei giochi e la funzione che individua il tipo di browser utilizzato dall'utente.

Nel caso specifico di quest'ultima (vedi sempre la tabella n. 1) appare evidente che la nostra funzione si limita alla sola distinzione tra il Netscape Communicator 4.x e il Microsoft Internet Explorer 4.0x per Windows 95. Va comunque evidenziato subito il fatto che "Biancaneve e l'Aritmetica" funziona perfettamente con tutte le versioni di Navigator (a partire dalla 3.xx) e di Explorer (a partire dalla versione 3.01).

Fatta questa doverosa introduzione, scendiamo un po' più in profondità andando finalmente a vedere il funzionamento generale delle funzioni di controllo del gioco.

La prima domanda che ci poniamo è su come funziona il controllo per l'assegnazione dei punti. Ovvero com'è operata la distinzione tra risposte giuste e risposte sbagliate?

Osserviamo di nuovo la tabella n. 1 (relativa al file principale framing.htm).

Proprio come prima serie di istruzioni JavaScript abbiamo definito delle semplici variabili.

Tra queste evidenziamo ovviamente quelle che in tale parte di articolo ci interessano:

```
var punti=0,tentativi=0;
```

A tali variabili, a seconda di come è strutturato ogni singolo gioco, vi fanno continuo riferimento le funzioni (es: "function Nome_Funzione(!)", le condizioni (if...) ed i gestori di eventi (OnClick, OnMouseOver, ecc.) che lo JavaScript pone come struttura di controllo ad ogni specifica situazione di ogni spe-

TABELLA 1

(Script estratti dal file principale)

```
//definizione variabili globali
var punti=0,tentativi=0;

//funzione che cambia le "domande"
function procedi(livello){
  var temp01=0;
  var temp02="";
  //livello 0 (dal bottone VAI!):doman01-03
  if (livello==0){
    temp01=Math.floor(Math.random()*3)+1;
    temp02="doman0"+temp01+".htm";
    return temp02; }

<frameset rows="24%, 76%" cols="1*" border="4" onLoad="tipo_browser()">
<frame src="maingame.htm" name="maingame" scrolling="no" toolbar=0 menubar=0 border="0"
frameborder="no">
<frame src="doman038.htm" name="domande" scrolling="auto" border="0" frameborder="no">

(La versione completa del frame principale FRAMING.htm è scaricabile da C&Vweb n.10)
```

Figura 6 - Un modello (da valutare ancora se preferire a quello attuale) relativo al modulo di gioco "A domanda... rispondi!". In questo caso la domanda comporterebbe il "checking" di più risposte. Forse verrà preferito al precedente.



cifico gioco.

Esemplifichiamo andando ad osservare sia la forma estetica (Figura - 3) che gli "estratti" dal codice operativo (tabella n. 2) del gioco L'Anagramma.

Com'è possibile dedurre sia osservando la schermata che il codice stesso, tale gioco basa essenzialmente sulla soluzione di tre distinti anagrammi. Tali soluzioni vanno digitate nei relativi campi di testo posti subito alla destra degli anagrammi (stilizzati come GIF da PaintShop Pro). Accanto a ciascun campo è poi posizionato il relativo bottone di controllo "Giusto o... Sbagliato?". Nella dinamica del gioco, una volta che si è digitata la soluzione di un anagramma, si può cliccare sul relativo bottone di controllo per vedere se la risposta è giusta oppure errata.

Premendo ad esempio il bottone di con-

trollo relativo al primo anagramma (nella tabella n. 2 corrisponde all'istruzione "d") l'evento, cioè il click, lancerà (onClick="Risposta1()") il controllo di quanto scritto nel campo. Il controllo è assegnato alla funzione denominata "Risposta1()" (che, sempre nella tabella n. 2, corrisponde all'insieme delle istruzioni "a").

Tale funzione verifica se, nel campo di testo costituito dall'istruzione "c", il valore digitato (cioè il "value") corrisponde alla parola esatta.

A tale compito, in questo caso, vi assolve una semplice condizione IF, che istruita dov'è posizionato il valore da controllare (la posizione viene specificata segnalando i parametri document, nome del FORM e nome dell'INPUT TYPE) va a vedere se la parola è uguale (==) a quella corretta ("pisolo").



Figura 7 - Abbiamo acquisito l'AVI in "full-frame" di una scena del film. Ora ne estrarremo i fotogrammi più significativi e li faremo diventare GIF animate da incastonare, via PaintShop Pro, all'interno di un riquadro statico...

Se queste effettivamente corrispondono verrà incrementato (top.tentativi++;) ed assegnato (top.maingame.document.score.tentati-

TABELLA 2

(Script estratti dalle funzioni di controllo dell'ANAGRAMMA)

a) funzione che verifica se la Prima Risposta (pisolo) è giusta o sbagliata

```
function Rispostal () {
  if (document.Terzogioco.anag01.value=="pisolo"){
    window.alert("OK, risposta esatta!");
    top.tentativi++;
    top.maingame.document.score.tentativi.value=top.tentativi;
  }
  else {
    window.alert("No, la risposta è sbagliata!");
    top.tentativi+2;
    top.maingame.document.score.tentativi.value=top.tentativi;
  }
}
```

b) funzione di controllo sull'esattezza dei tre anagrammi (pisolo, azzurro, principe)

```
function Valid_Anagl() {
  if (document.Terzogioco.anag01.value=="pisolo"){
    if (document.Terzogioco.anag02.value=="azzurro"){
      if (document.Terzogioco.anag03.value=="principe"){
        window.alert("OK, risposta esatta: prendi 3 punti!");
        top.punti=top.punti+3;
        top.maingame.document.score.punti.value=top.punti;
        top.domande.location.replace(top.procedi(7));
      }
      else {
        window.alert("Oh no! Risposta sbagliata: hai sprecato 3 tentativi!");
        top.tentativi=top.tentativi+3;
        top.maingame.document.score.tentativi.value=top.tentativi;
      }
    }
  }
}
```

c) Primo Campo di scrittura delle parole (relativo a "pisolo")

```
<INPUT TYPE="text" NAME="anag01" VALUE="" SIZE="10" >
```

d) Bottone di controllo sul singolo campo ("GIUSTO o... SBAGLIATO?")

```
<INPUT TYPE="button" NAME="giusto_sbagliato1" VALUE="GIUSTO o... SBAGLIATO?"; onClick="Rispostal()" >
```

e) Bottone di controllo su tutti i campi ("Ho Vinto?")

```
<INPUT TYPE="button" NAME="risposta" VALUE="Ho Vinto?"; onClick="Valid_Anagl();" >
```

f) Bottone di "salvataggio" per uscire dal gioco

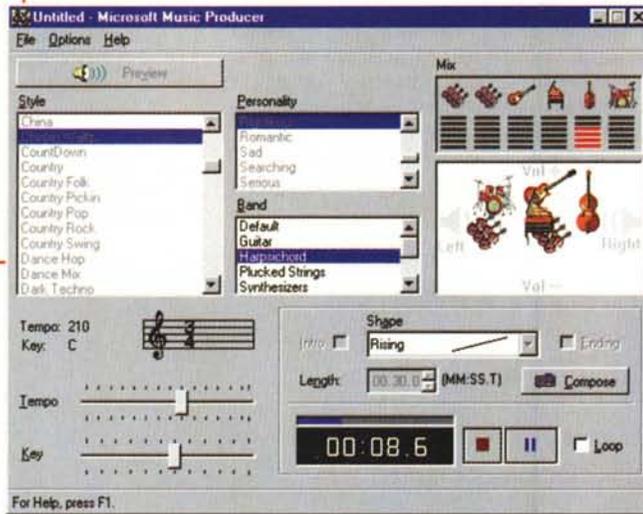
```
<INPUT TYPE="button" NAME="uscita" VALUE="Rinuncio!"; onClick="top.domande.location.replace(top.procedi(7))" >
```

(La versione completa del frame principale FRAMING.htm è scaricabile da C&Vweb n.10)



Figura 8 - ... sul quale sono già posizionati il numero della scena ed il tipo di azione a cui si riferisce!

Figura 9 - Un momento d'estro di Music Producer. Riposizionando gli strumenti, diminuendo oppure aumentando il tempo e provando vari stili, stiamo per realizzare un brano da dedicare alla presentazione delle singole scene che caratterizzano la favola in questione!



vi.value=top.tentativi;) il numero dei tentativi aggiornati. Ciò avverrà nel frame superiore (dov'è posizionato, nel file "maingame.htm, il value del campo denominato tentativi e relativo al form score). Oltre a ciò verrà anche visualizzato (window.alert) un messaggio di conferma. Se al contrario (else) la condizione IF non sussiste, invece di incrementare di un valore il numero dei tentativi, proprio per dare una distinzione tra giusto/sbagliato, aumenterà di due (top.tentativi+2;). Oltre a ciò verrà visualizzato un messaggio negativo. Sempre nello specifico del gioco dell'Anagramma, al bottone "Ho Vinto" è associato a sua volta un controllo "onClick" che lancia la più complessa funzione (function Valid_Anag1() {} per mezzo della quale vengono controllati, sempre via IF, tutti e tre i campi di scrittura. Se tali campi risultano corretti (==) viene visualizzato un messaggio di conferma ed assegnati 3 punti (top.punti =

top.punti+3;) invocando la giusta posizione del campo "punti" allo stesso modo di come, più sopra, il controllo aveva intercettato il campo dei tentativi. Se il controllo dà esito negativo al contrario (else) verrà visualizzato un messaggio negativo e quindi assegnati tre tentativi sbagliati. Solo nel caso che l'IF viene soddisfatta, con un'ultima istruzione (inserita prima della graffa che separa dall'ELSE) viene infine chiamata la funzione per il caricamento di una delle tre pagine che costituiscono il gioco successivo. La funzione invocata si chiama "Procedi" ed è strutturata (vedi di nuovo la tabella n. 1) da due variabili temporali (temp01, inizializzata a zero, e temp02) e gestite da una serie di IF che verificano il valore assegnato al parametro (livello) della funzione "procedi" e che, di conseguenza a questo valore, vanno a consegnare

le due variabili. Ad esempio, se (IF) il livello invocato è pari a zero, verrà istruita un'istruzione che valorizza la variabile temp01 come segue:

```
temp01 = Math.floor(Math.random()*3)+1;
Se, al contrario, il livello invocato è pari a 1, l'istruzione valorizzerà la variabile sommandovi un valore tre volte (cioè 3 numeri di pagine) superiore:
temp01 = Math.floor(Math.random()*3)+4;
```

In ciascuna di queste due circostanze esemplificative verranno ricostruite le URL delle pagine assegnate alla variabile temp02.

Nel primo caso (livello zero) l'operazione generale (temp02="doman0"+temp01+".htm";) darà come risultato doman001.htm (oppure doman002/doman003).

Nel secondo caso (livello 1) darà invece come risultato: doman004.htm (oppure doman005/doman006).

Andando a denominare con tale successione cronologica tutte le pagine che si realizzano ed assegnando un valore crescente per ogni livello della funzione "procedi", si riesce a realizzare una struttura a ricerca casuale ogni tre, specifiche pagine.

Ovviamente, aumentando il numero delle pagine alternative (per dare cioè più "vivibilità" al gioco in questione) bisognerà aumentare anche il fattore "X" da inserire nell'operazione:

Math.floor(Math.random()*X). Benché ancora in fase di "l'imaggio", tali istruzioni costituiscono una base interattiva già sufficientemente collaudata che, per la sua flessibilità è riadattabile alle situazioni più varie.

Va comunque precisato il fatto che tali strutture logiche, nel corso delle ultime revisioni del codice della favola, potrebbero subire alcune modifiche. In realtà quanto stiamo scrivendo avviene intorno alla seconda metà del mese di agosto (i tempi di compilazione degli articoli!) e non coincide ancora con il prodotto finito. Malgrado ciò, le variazioni di codice non saranno più strutturali, ma solo marginali.

Passiamo ora a vedere il lato multimediale della faccenda (anche qui troverete ad attendervi delle applicazioni JavaScript sicuramente interessanti.

Le GIF animate

Quelle che andavano realizzate erano sia GIF di pura estetica (logo, titoli stilizzati, clip d'abbellimento, ecc.) che più

complesse animazioni, d'armonizzare poi al contesto di una scena o di un gioco. Banali le prime, preferiamo sfruttare l'intero spazio di questo capitolo per descrivere più dettagliatamente le seconde.

Diversamente da quanto teorizzato in sede di progetto (e da quanto prospettato nell'articolo "propedeutico" apparso sul numero di settembre) le GIF animate non le abbiamo più utilizzate per cercare di fondere la parte grafica con i riferimenti testuali delle singole scene.

Tale tecnica, al momento pratico, s'è dimostrata talmente complessa e lunga nella sua fase di applicazione da richiedere uno slittamento della pubblicazione. Ci abbiamo rinunciato e preferito imboccare un'altra strada.

In realtà, la soluzione adottata per la presentazione di ogni singola scena (com'è possibile verificare su C&Vweb n. 10) è organizzata in maniera più razionale e per certi versi più interessante. Di certo è più completa.

Osservando difatti la figura 10 possiamo vedere una struttura tabellare all'interno della quale, sono presenti:

- una GIF animata (composta da un identificativo statico relativo al numero della scena attuale, l'effettivo quadro animato, e di nuovo un identificativo statico per il titolo della scena stessa);
- una semplice GIF, usata per l'ancoraggio alla tabella del gioco collegato alla scena specifica;
- il testo in HTML della specifica scena (e quindi via via "copiabile" fino a ricostruire l'intera favola);
- il player multimediale per poter ascoltare la ripresa audio (estratta da un film concessoci, con un colpo di fortuna, in licenza!). Tale riproduzione avviene in automatico grazie all'utilizzo di una normale istruzione "embedded" del tipo:

```
<EMBED SRC="audio/audscn01.wav" NAME="snd" WIDTH="143" HEIGHT="33" MASTERSOUND>
```

Credo che appaia subito evidente (soprattutto a chi ci naviga dentro!) quanto sia più completo tale assetto che non quello preventivato. Invece di animare in un'unica GIF i frame del film e il testo della favola, la soluzione adottata permette al giocatore di disporre di tutto il tempo che gli necessita per leggere la favola, di godersi il loop dei frame e, se vuole, d'ascoltare la colonna sonora del film grazie al player incorporato nella tabella.

Dal punto di vista del codice, la realizzazione del modello base prima e di tutte le specifiche scene poi, è quanto di più semplice possa esistere nell'ambito dell'HTML.

Figura 10 - Il gioco nella fase di presentazione di una scena. Da notare, oltre al blocco delle GIF ed al player "embedded" nella cella dedicata, la presenza della GIF "Inizio del gioco". Quest'ultima è ancorata alla prima parole ("A domanda...") del titolo assegnato alla tabella del gioco che segue. Un semplice click sulla GIF in questione...



Figura 11 - ... ed ecco posizionarsi in schermo la tabella del gioco.

Dimensionata la tabella (da 500 pixel e composta di quattro celle disposte su due righe) e sistemati testo e GIF, le uniche parti di codice più "complesse" risultano essere quelle per il link in "embedding" con il file audio da riprodurre e quella per l'ancoraggio dalla tabella in questione alla tabella del gioco annesso alla scena.

Una complessità comunque relativa se si è un poco infarinati sui TAG e se soprattutto si dispone di un editor HTML di tipo WYSIWYG. Personalmente ho usato il Composer di Netscape Communicator 4.01 e, per quanto riguarda il settaggio dell'Anchor (Figura - 12) tutto è avvenuto in pieno automatismo richiamando dei semplici menu.

Per chi non lo sapesse, gli Anchor sono dei comandi che vengono utilizzati per andare da un punto ad un altro specifico punto detto "target", presente sia più avanti nella pagina

che su altre pagine.

Nel caso della scena in questione la GIF statica denominata "Inizio del gioco" è stata ancorata, sulla stessa pagina della scena, alla parola "domanda" presente nel titolo assegnato al gioco "A domanda... rispondi!".

Tale titolo è posto in testa alla tabella contenente lo schema del gioco.

L'Anchor che n'è derivato, a soddisfazione dei puristi, è il seguente:

```
< A H R E F = "# d o m a n d a" > < I M G SRC="botgame.GIF" BORDER=0 HEIGHT=23 WIDTH=160></A>
```

ed ha come riferimento il TAG: collegato a sua volta alla riga d'intestazione della tabella del gioco che è la seguente:

```
<FONT SIZE=+3>A domanda ...</I></FONT>
```

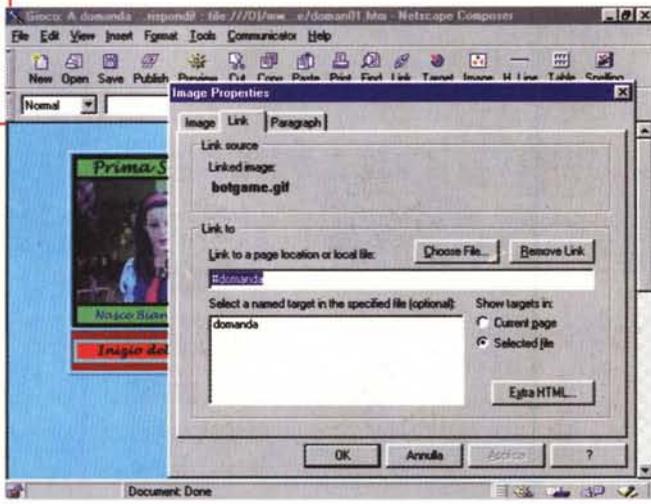
Con tali istruzioni il codice HTML, una volta cliccato sulla GIF "Inizio del gioco" (Figura - 10) ci ha permesso di spostare la visione della pagina (così come mostrato in Figura - 11) direttamente sulla tabella del gioco.

Un altro, interessante uso delle GIF ani-



Figura 12 - La struttura della tabella di presentazione alle scene qui schematizzata con tutti i riferimenti ai controlli inseriti sia in HTML che in JavaScript.

Figura 13 - Netscape Composer al lavoro sul pannello delle proprietà di un'immagine per fissare un "anchor".



mate, è quello che abbiamo organizzato per l'ultimo gioco della favola (Figura - 14). Tale gioco, denominato "Il Trovarisposte", si basa sulla cosiddetta tecnica del bottone rotante. Un bottone cioè, sul quale appaiono sequenzialmente le varie risposte tra cui scegliere (in realtà si tratta di link che, se cliccati, possono portare ciascuno verso una specifica pagina o URL). Tra queste risposte, che si visualizzano una dopo l'altra ogni "x" secondi, solo una è ovviamente quella che soddisfa la domanda e che fa procedere nel gioco. Le altre, un po' com'è nel "gioco dell'oca", portano indietro verso le pagine delle scene e dei giochi precedenti. Da quello che abbiamo potuto verificare, in un quadro dinamico così articolato (dove anche la domanda a cui rispondere è esposta in modo abbastanza subdolo), oltre che per abbellire la scena, la

GIF animata di "turno" poteva benissimo rappresentare un momento di aiuto. Quello che abbiamo allora pensato di realizzare è una specie di lavagna scolastica dove un gessetto invisibile disegna ciclicamente numeri, segni aritmetici e spunti di vario genere, in modo da costituire un continuo suggerimento ed, al tempo stesso, un'altra fonte di apprendimento. Lo storyboard di tale sviluppo animato (Figura - 15) è in grado di rendere più evidente tale soluzione.

Audio: dal VCR è meglio! ... filmati che diventano GIF animate e creazioni MIDI!

Lo strano titolo di questo capitolo (Au-

dio: dal VCR è meglio!) fissa in una frase quanto ho potuto verificare in sede di acquisizione ed editing delle parti della colonna sonora che abbiamo avuto la fortuna di utilizzare. Partito con l'acquisire in Wave-PCM per poi convertire o in Wave-TrueSpeech oppure in MPEG-mp2 (quest'ultimo ora leggibile per mezzo dell'ActiveMovie anche dal Netscape Communicator 4.x) mi sono presto trovato nella situazione di dover operare una scelta: -scegliere la conversione in Wav-TrueSpeech, codec rapidissimo al caricamento (un vero streamer, ovvero già in riproduzione via via che il file viene scaricato), ma anche di qualità notevolmente inferiore al formato mp2? -oppure scegliere il formato mp2, di maggiore qualità, ma anche di maggiori tempi di attesa per il caricamento? Dovendo scegliere... non ho scelto! Fatto un'altra serie di prove, riacquisendo dall'output audio del VCR direttamente in TrueSpeech, ho potuto constatare che tale formato (grazie ai tagli imposti dal filtro posto all'output del VCR) quando acquisisce direttamente sforna una qualità di sintesi decisamente superiore ad un file equivalente derivato da una conversione. La scoperta dell'acqua calda! Comunque sia i file in Wav-TrueSpeech che parlano dalle pagine della favola sono miei testimoni. Navigando su C&Vweb n.10 lo potrete constatare per vostro conto. Per quanto poi concerne le sequenze video, sempre prese dagli spezzoni del film concessoci di utilizzare, queste, dopo averle acquisite in Avi-full frame, le abbiamo desequenziate in GIF, estrapolando quindi i fotogrammi più significativi (un primo piano, uno stacco di scena, un movimento diverso, un'espressione che cambia, ecc.) per incastonarli via via in un quadro preesistente, nel quale sono già inseriti i riferimenti statici relativi al numero della scena e del titolo relativo all'azione che, per la specifica animazione, si svolgerà in esso. Tale lavoro s'è svolto utilizzando, in fase di acquisizione la mia fedele, VideoBlaster RT-300 in congiunzione con il VidCap32 e l'inoscidabile VidEdit. In ambiente di cattura ho provveduto ad acquisire con un iniziale quadro video pari a 240x180, con un frame rate pari a 15 frame al secondo e con metodo di compressione escluso (full-frame). In sede di editing, il quadro video, ripulito dalle linee periferiche eventualmente disturbate (effetto shifting), è stato ri-

dotto a soli 160x120 dot per poi procedere all'estrazione da questo dei fotogrammi più significativi. Quest'ultimi, una volta salvati come .dib, l'ho infine caricati in ambiente PaintShop Pro 4.12 per convertirli definitivamente in GIF (256 colori "ditherati"). A questo punto è cominciata l'opera finale, ovvero l'inserimento di ogni GIF-frame all'interno del quadro preesistente. Ogni file così ottenuto (composto dalla denominazione della scena, dal singolo frame significativo e dal commento all'azione) l'ho quindi salvato con una denominazione cronologica.

A completamento dell'opera le GIF preparate dal Paintshop Pro sono state caricate nel GIF Animator di Microsoft e settate ciascuna per durata in schermo, fino a costituire il file animato finale.

Dall'audio e dal video un salto, infine, nel mondo dell'elaborazione musicale per la realizzazione della colonna sonora originale. Al riguardo ci siamo saggiamente avvalsi dell'apporto (determinante!) del Microsoft Music Producer.

Avremmo potuto utilizzare dei brani MIDI già belli e confezionati e probabilmente più d'atmosfera. Ma avremmo sfruttato il lavoro di altri (tali Chopin, Schumann, Haydn, ecc.!).

Standoci l'opportunità abbiamo preferito utilizzare il Music Producer e ricavare dei tessuti musicali in tutto e per tutto originali e tutt'altro che scontati!

Tutti i file musicali che accompagnano il caricamento delle pagine (per le scene uno specifico genere musicale, per i giochi un altro genere ancora) sono stati realizzati dal Music Producer. Ridisposizione stereofonica degli strumenti, scelta del tempo, della chiave, dello stile e della personalità del brano, nonché gli eventuali tagli ad hoc della durata. Il risultato finale è in circa 30 brani MIDI diversi per un totale di appena 150-200 Kbyte di musiche originali. Se volete, vi passo i file-progetto...

Conclusioni

OK, lo confesso: realizzare "una favola per la rete" mi ha entusiasmato.

In realtà s'è trattato di una vera e propria sfida che ho lanciato a me stesso, al Web ed all'HTML! Quelle che ritengo di aver ottenuto sono tutte risposte positive. Il Web può essere conquistato anche dall'ipermedialità, purché questo sia il frutto di ideazioni che sappiano conciliarsi con i limiti del solito "doppio telefonico" che condiziona la vita di ogni navigatore.

Va subito detto che con il solo HTML

Figura 14 - Modulo di gioco "Il trovarisposte". La domanda è un "piccolo enigma"... la risposta giusta si alterna ogni due secondi (su di un bottone rotante) ad una serie di risposte sbagliate. Se si sbaglia risposta si viene portati indietro nel gioco. Come se ne esce fuori?

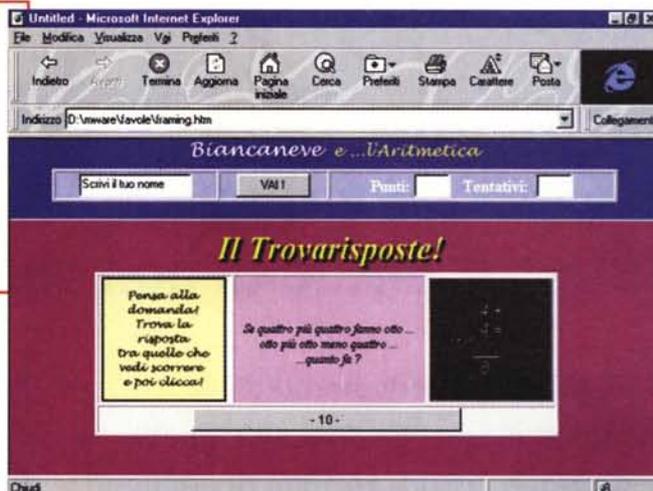


Figura 15 - ... magari con l'aiuto di una GIF animata che, in loop, prova a dare consigli e suggerimenti!

saremmo riusciti a tirar fuori cose modeste. Corroborato invece col supporto del JavaScript, anche il semplice linguaggio della rete, s'è come rivitalizzato. E' stato possibile discernere tra le risposte, assegnare punti, far apparire messaggi, aiuti e dare consigli. Insomma, introdurre una buona dose d'interattività.

Facendo tutto ciò non abbiamo certo segnato una tappa fondamentale nella storia del Web ma, forse un po' immodestamente, già applicato ciò che fra non molto con l'avvento dell'Explorer 4.0 (quale estensione verso Windows 98 dell'attuale Windows 95!) sarà possibile, anzi quasi obbligatorio fare. Ovvero, scrivere il codice dei prodotti ipermediali di ogni produzione di Desktop Presentation, in HTML. Meglio ancora, in Dynamic-HTML.

Tra non molto difatti tutto ciò che è presentazione ipermediale verrà realizzato

abbandonando i famigerati runtime o gli (in)eseguibili da 100 Mbyte. Diventerà tutto Dynamic-HTML e potrà indifferentemente girare su Internet oppure in locale, sul PC oppure sul Mac o altri Sistemi Operativi.

Vi sembra poco?

Noi, la nostra brava esercitazione l'abbiamo fatta e su C&Vweb n. 10 ne rischiamo giudizi e lazzi. Al riguardo una precisazione: se siete ancora imbarcati su vecchie versioni di Navigator (la 2.00) e di Explorer (la 3.00) è ora di dismetterle. Non solo perché non rispondono pienamente ai controlli JavaScript da noi utilizzati, ma anche perché è ora che facciate l'upgrade!

In definitiva... navigate sulla favola gente, e se eventualmente v'interessa parlarla off-line, mandate un messaggio al mio solito recapito su MC-link e vedremo cosa si può fare.

MS