Guida HTML5: il Canvas - Parte 3

Benvenuto alla terza parte di questa guida dedicata al Canvas. Ora che conosciamo tutti i meccanismi di base che regolano questo strumento, possiamo cominciare a creare qualcosa di più complesso e decisamente più interessante come un rudimentale programma di disegno.

Prepariamo l'interfaccia

Come prima cosa dobbiamo associare delle azioni a ogni possibile evento del mouse, ovvero: *onMouseDown* (quando il pulsante del mouse è premuto sull'elemento), *onMouseUp* (quando il pulsante del mouse viene rilasciato), *onMouseOut* (quando il puntatore lascia l'elemento) e *onMouseMove* (quando il puntatore si muove all'interno dell'elemento).

Per iniziare a utilizzare il programma, l'utente dovrà selezionare uno dei colori differenti dalla barra degli strumenti e definirne lo spessore. Andremo a realizzare questi elementi attraverso dei semplici tag

clickabili.

Come puoi vedere l'attributo *onclick* del tag

richiama una funzione JavaScript denominata *modificaColore*. Questa funzione dovrà accettare due parametri: il colore e il riferimento all'icona cliccata.

Ripetiamo lo stesso procedimento adattandolo per settare lo spessore attraverso il metodo *lineWidth*:

Definiamo le azioni

Come prima cosa abbiamo bisogno di definire e settare una variabile globale, che denomineremo *disegnoAttivo*, da poter interpellare a ogni azione per conoscere l'attuale stato:

Dopodiché possiamo creare la funzione *inizioDisegno* che all'inizio di quest'articolo abbiamo associato all'evento *onmousedown*:

Proseguiamo con la funzione *disegno*:

E terminiamo con la funzione *interrompoDisegno* nella quale settiamo la variabile *disegnoAttivo* su false:

Salviamo il risultato

Ora che tutte le azioni utili al disegno vero e proprio sono state definite, possiamo creare le ulteriori due necessarie al salvataggio e alla pulizia.

Cominciamo col trattare la più semplice delle due, la funzione *pulisci*:

Come puoi vedere non è altro che un semplice richiamo al metodo *clearRect*. La funzione salva, invece, attraverso il metodo *toDataUrl* converte l'immagine in una sequenza di caratteri formattati come fosse un URL:

Ovviamente non dimentichiamo di aggiungere i comandi per richiamare l'azione appena definita nell'HTML principale del nostro programma:

Con questo metodo, tuttavia, per salvare l'immagine sarà necessario che l'utente, attraverso il tasto destro del mouse, faccia clic fisicamente su "Salva immagine con nome...". Certo, questa non è la soluzione ottimale ma è la più semplice da attuare in un contesto come questa guida che ha come unico argomento l'HTML5.

Conclusione

Nel corso di questo articolo abbiamo imparato come creare un basilare programma di disegno utilizzando pienamente alcuni metodi messi a disposizione da Canvas. Nella prossima lezione impareremo le ultime nozioni sul Canvas, ovvero come ritagliare immagini, utilizzare pattern e sfumature, gestire le ombre.

Ti ricordo che, come consuetudine, puoi scaricare il codice completo di questa versione da questo link!

GUIDA HTML5: GLI ARTICOLI
1) Guida HTML5:Introduzione
2) Guida HTML5: la prima pagina

3) Guida HTML5: la struttura
4) Guida HTML5: Immagini e outlines
5) Guida HTML5: nuovi elementi semantici
6) Guida HTML5: i form – Parte 1
7) Guida HTML5: i form – Parte 2
8) Guida HTML5: i form – Parte 3
9) Guida HTML5: i form – Parte 4
10) Guida HTML5:i tag audio e video – parte 1
11) Guida HTML5:i tag audio e video – parte 2
12) Guida HTML5: I player video
13) HTML5: Il Canvas – Parte 1
14) Guida HTML5: Il Canvas - Parte 2
15) HTML5: Il Canvas - Parte 3
16) HTML5: Il Canvas - Parte 4
17) HTML5: Web storage