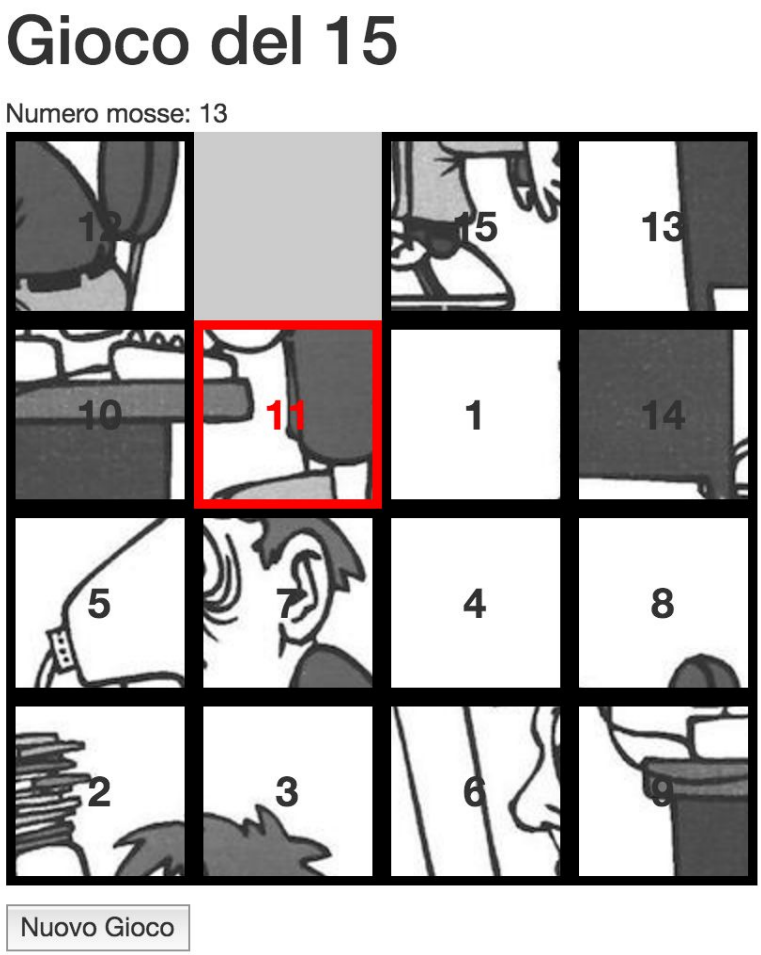
Homework 4 - Gioco del 15

Introduzione

Il gioco del 15 è un gioco classico che consiste in una griglia 4x4 di quadrati numerati da 1 a 15 più uno mancante. Lo scopo del gioco è quello di ordinare i quadrati mouvendoli nella posizione libera per spostarli nella griglia.



Lo scopo dell'esercizio è quello di scrivere i seguenti file:

- Il file **style.css** che definisce lo stile per la pagina.
- Il file **game.js** che contiene tutto il javascript del gioco.
- I file **bg1.jpg**, **bg2.jpg**, ...

Dovete prendere da internet almeno due immagini (**bg1.jpg**, **bg2.jpg**, ...) per il gioco che vanno consegnate con gli altri file, e che permetteranno di disegnare il puzzle. Tutte le immagini devono avere una dimensione 400px X 400px.

Codice di partenza: http://ppl.eln.uniroma2.it/pw/hw4/hw4.zip

Viene fornito il file **index.html** che crea la pagina. Il file HTML da usare va scaricato e NON deve essere modificato manualmente. Per cambiare la pagina dovete scrivere il codice JS opportuno e creare gli stili nel file CSS.

IMPORTANTE: Il file index.html non deve essere modificato manualmente!!!!

Aspetto

Nella pagina sono disegnate le quindici tessere che rappresentano il puzzle. Il puzzle nel complesso occupa 400x400 pixel. Ogni tessera di puzzle occupa i 100x100 pixel di cui 5px su tutti e quattro i lati sono occupati dal bordo nero. Questo lascia pixel 90x90 di superficie all'interno di ogni piastrella. Il file HTML fornito non contiene i quindici div per rappresentare i pezzi del puzzle, e non si deve modificare il file HTML; usando JavaScript creare i pezzi del puzzle e aggiungerli alla pagina. Inizialmente la pagina dovrebbe apparire con il puzzle con i pezzi nell'ordine corretto, con 1 in alto a sinistra, 4 in alto a destra, 13 in basso a sinistra, la casella vuota in basso destra, etc. e con il banner che mostra di aver vinto come in figura (si veda dopo per il banner).

Ogni tessera visualizza un numero da 1 a 15, in un font 22pt bold. Ogni tessera mostra parte dell'immagine di background. La porzione di immagine visualizzata da ogni div è legata al numero di tessera. La tessera "1" mostra la porzione di 100x100 in alto a sinistra dell'immagine. La piastrella "2" mostra il successivo pezzo 100x100px dell'immagine componendo alla fine l'immagine completa quando si imposta come l'immagine di sfondo di ogni singolo pezzo. Si usi la proprietà background-position per ottenere l'effetto voluto.

Sopra al puzzle è presente un div per visualizzare numero delle mosse. Sotto al puzzle è presente un div che contiene il bottone nuova partita.

Svolgimento del Gioco

Quando si clicca su una tessera, se quella si trova accanto alla cella vuota, questa viene spostata nello spazio vuoto. Se il tassello non è vicino alla cella vuota, non si verifica alcuna azione. Allo stesso modo, se si clicca sulla cella vuota o altrove sulla pagina non si verifica alcuna azione.

Quando il mouse passa sopra una tessera che può essere spostata (ossia vicina a quella vuota), il colore del suo bordo e il testo dovrebbe diventare rosso. Inoltre, il cursore del mouse deve trasformarsi in "mano". Quando il cursore non è più sulla cella, sia l'aspetto della cella che quello del cursore devono tornare normali.

Algoritmo Shuffle

Quando si preme il tasto nuovo gioco, le tessere del puzzle sono riorganizzate in modo casuale per fornire all'utente un nuovo puzzle da risolvere.
Le piastrelle devono essere posizionate in uno stato risolvibile. Alcuni stati puzzle non sono risolvibili; per esempio, il puzzle non può essere risolto se si scambiano solo le tessere 14 e 15. Quindi l'algoritmo per mischiare non può semplicemente generare delle posizioni casuali per le 15 tessere. Si realizzi quindi una funzione simuli il movimento casuale delle tessere per un numero elevato di volte (ed esempio 1000 spostamenti) partendo della posizione finale delle tessere. Lo spostamento deve essere solo logico e non mostrato all'utente a cui verrà presentato il risultato finale.

Nel file game.js va definita la funzione di shuffle che opererà l'operazione di mescolamento delle tessere:

```
function shuffle(){
    var positions = [[1,5,9,13],[2,6,10,14],[3,7,11,15],[4,8,12,0]];
    …
    return positions;
}
```

Notifica di fine gioco

Aggiungere codice per rilevare quando l'utente ha risolto il puzzle. Quando l'utente si sposta tutti i pezzi del puzzle al loro posto corretto, la pagina dovrebbe visualizzare un messaggio di congratulazioni per l'utente (come nella figura precedente).
Il messaggio dovrebbe apparire come parte della pagina e non come alert o come finestra pop-up. Ad esempio, è possibile aggiungere un div con del testo e mostrarlo usando il css.

Per semplificare la correzione, tale messaggio va mostrato come la pagina viene caricata e va levato quando viene premuto il button per la nuova partita.

Backgound random

Ogni volta che viene cliccato il tasto nuovo gioco va cambiata l'immagine di background. Memorizzare le immagini nella stessa cartella del file JS. Le immagini devo tutte essere di dimensione 400px X 400px. Si consiglia di scaricarle da internet e modificarle usando dei programmi per l'elaborazione delle immagini.

Strategia di sviluppo

Si suggerisce di usare la seguente strategia

- Scrivere la funzione che esegue lo shuffle e testarne il funzionamento.
- Scrivere una funzione che esegue il rendering dei 15 elementi senza alcun background dietro di loro, verificando che le posizioni risultino corrette.
- Impostare il background delle varie tessere e posizionarlo opportunamente.
- Scrivere la funzione che esegue gestisce il click della tessera.
- Scrivere il codice per determinare se una tessera può muoversi o no (ossia se è vicina ad una vuota).
- Creare il codice che gestisce il cambiamento delle tessere e del mouse.
- Implementare la fine partita

Valutazione

Il codice HTML e CSS deve passare i validatori del W3C ed essere stilisticamente corretto.
Il codice JS dovrebbe utilizzare gli stili e le classi CSS piuttosto che impostare manualmente tutte le proprietà. Ad esempio, invece di impostare il .style di un oggetto DOM, definire una classe nel file css ed assegnare il className all'oggetto. Tuttavia le proprietà di stile relative alle posizioni delle tessere ed allo sfondo non conviene metterle nel file CSS ma conviene variarle dal codice JS.
Registrare in JavaScript i listener per i diversi eventi e non nel codice HTML.
Non si deve usare nessun framework o librerie JavaScript come jQuery.
Il file .js deve essere eseguito in modalità rigorosa mettendo "use strict"; in cima. Controllare la correttezza del codice con un validatore JS (http://www.jslint.com/).

Suggerimenti

- Utilizzare il posizionamento assoluto per impostare le posizioni x / y di ogni tessera.
- Non fare in modo esplicito un div per rappresentare la casella vuota del puzzle.
- Non memorizzare le piazze del puzzle in un array 2-D. Questo potrebbe sembrare una buona struttura a causa della 4x4 aspetto della griglia, ma è cattivo stile e sarà difficile tenerlo aggiornato come i quadrati si muovono.

Altri spunti di codice (non valuatati)

- Testare le animazioni css3 per muovere le tessere, invece spostarle in modo rigido.
- Generalizzazione del gioco a griglie di dimensione diversa (3x3 o 6x6).
- Gestione del drag della cella
- Gestione di schermi piccoli