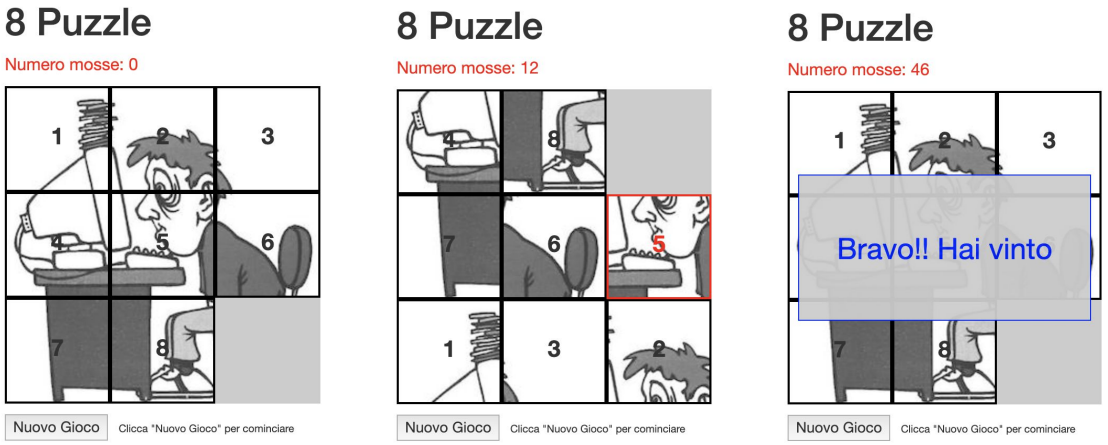
8 Puzzle

Introduzione

8 Puzzle è un gioco che consiste in una griglia 3x3 di quadrati numerati da 1 a 8 più uno mancante. Lo scopo del gioco è quello di ordinare i quadrati usando la cella libera. Nella figura seguente sono mostrate delle fasi di gioco:



Lo scopo dell'esercizio è quello di scrivere o creare 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 300px X 300px.

Codice di partenza: http://pw.netgroup.uniroma2.it/hw2/8puzzle.zip
Video descrittivo: desc-8puzzle.mp4

Viene fornito il file **index.html** che crea la pagina, i file style.css e game.js vuoti, ed un background di test (bg1.jpg).

IMPORTANTE: Il file index.html non deve essere modificato manualmente!!!! Per cambiare la pagina dovete scrivere il codice JS opportuno e creare gli stili nel file CSS.

Aspetto

Nella pagina sono disegnate le otto tessere che rappresentano il puzzle. Il puzzle nel complesso occupa 300x300 pixel. Ogni tessera di puzzle occupa i 100x100 pixel di cui 2px su tutti e quattro i lati sono occupati dal bordo nero. Il file HTML fornito non contiene i div per rappresentare i

pezzi del puzzle, e non si deve modificare il file HTML; usando JavaScript creare degli elementi per i diversi i pezzi del puzzle e aggiungerli alla pagina.
Ogni tessera visualizza un numero da 1 a 8, in un font 22px 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 100x100px 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 il colore del suo bordo e il testo diventano 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 puzzle non sono risolvibili; per esempio, il puzzle non può essere risolto se si scambiano solo le tessere 7 e 8. Quindi l'algoritmo per mischiare non può semplicemente generare delle posizioni casuali per le tessere. Per capire determinare la risolvibilità si può scrivere un algoritmo basato sulle osservazioni seguenti:.
Invece di rappresentare il puzzle con una matrice, lo rappresentiamo, mediante un vettore contenente i numeri da 1 a 8 e dove 0 indica la cella vuota. Il vettore che rappresenta lo stato stato finale è: **[1, 2, 3, 4, 5, 6, 7, 8, 0]**.
Diamo due definizioni:

- **inversione**: un'inversione è una coppia di tessere che non sono nell'ordine corretto;
- **polarità** - il numero totale di inversioni in un vettore.

Il puzzle è risolvibile se la polarità è pari!!

Esempio di calcolo della polarità:
[1, 3, 4, 7, 0, 2, 5, 8, 6]
1 -> 0 inversioni
3 -> 1 inversione (solo il 2)
4 -> 1 inversione (solo il 2)

7 -> 3 inversioni (2, 5 e 6)
2, 5, 8 -> 0 inversioni
Polarità = 4 => il Puzzle è risolvibile

Per altre info:
https://datawookie.netlify.app/blog/2019/04/sliding-puzzle-solvable/

Notifica di fine gioco

Aggiungere una funzione per rilevare quando l'utente ha risolto il puzzle. Quando l'utente sposta tutti i pezzi del puzzle al loro posto corretto, la pagina visualizza un messaggio di congratulazioni per l'utente (si veda la figura o il video).
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.

Background 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 devono tutte essere di dimensione 300px X 300px. 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 crea le 8 celle, solo con i numeri da 1 a 8 senza alcun background dietro di loro, verificando che le posizioni risultino corrette.
- Scrivere la funzione che sposta le 8 celle a seconda di un vettore di posizioni (es. posizioni = [1, 3, 4, 7, 0, 2, 5, 8, 6];).
- Scrivere la funzione che esegue lo shuffle del vettore.
- Scrivere la funzione che gestisce il click di una tessera determinando se una tessera può muoversi o no (ossia se è vicina ad una vuota). La funzione cambia eventualmente il vettore delle posizioni e riesegue la funzione di spostamente delle celle.
- Implementare la fine partita
- Impostare il background delle varie tessere e posizionarlo opportunamente
- Gestire i cambio random del background

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 tipo 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 posizioni del puzzle in un array 2-D. Questo potrebbe sembrare una buona struttura a causa della 3x3 aspetto della griglia, ma sarà difficile tenerlo aggiornato come i quadrati si muovono.