Relazione Progetto Tecnologie Web
A.A. 2009-2010

A Cura di:

Daniele Lovato 578396

Simone Daminato 574545

Francesco Nwokeka 582813

Alessandro Capogna 574320

2Steps2Hell Website – Comunità di gioco Online

La seguente relazione illustra in modo esauriente e dettagliato le fasi di analisi e progettazione del sito 2steps2hell, sito di una comunità di gioco online.

Per prima cosa il sito è stato progettato per essere uno strumento dove reperire news, informazioni e contatti riguardanti un team di giocatori di un noto gioco open source: UrbanTerror.

Il team in questione è il clan “2Steps2Hell” (two-steps-to-hell).

I visitatori del sito saranno principalmente i membri di questo gruppo, ma anche esterni che vogliono reperire informazioni riguardo al gruppo in questione. Sarà possibile reperire news sulle faccende del clan, risultati partite, iscrizioni e date di vari tornei. C'è anche una sezione con la spiegazione del gioco.

Inoltre saranno disponibili i contatti dei vari membri in caso di bisogno di ulteriori informazioni non reperibili dal sito.

Il reperimento di informazioni appena descritto è stato reso semplice da un menù navigazione basilare che permette l’accesso alle varie pagine del sito che riportano uno solo degli argomenti elencati. Perciò gli utenti non dovranno percorrere tutto il sito, ma cercare solo la pagina interessata con un semplice click sul menù.

Gli utenti possono aver accesso al sito da qualunque browser (Firefox, Chromium, Explorer, Opera) e da qualsiasi dispositivo (pc a qualunque risoluzione, dispostivi portatili). Inoltre, difficilmente il sito sarà visitato da utenti con problemi di daltonismo, ma per renderlo comunque accessibile anche a questa categoria di utenti il sito è stato realizzato anche in versione “daltonizzata”.

Passiamo alla descrizione di ogni singola pagina del sito:

~ Home

Rappresenta la home del sito, in cui vengono visualizzate le informazioni generali sul clan per quanto riguardano gli allenamenti, contatti e informazioni per coloro che vogliono entrare a far parte del gruppo di gioco.

~ Il Gioco

Pagina che descrive il gioco Urban terror con tutte le informazioni più importanti riguardanti la storia, sviluppatori,armi del gioco e modalità di gioco , insomma tutto quello che serve per conoscere di che tipo di gioco si tratta.

~ Clan

La pagina che descrive i membri del clan, dai veterani alle reclute, con foto e descrizione

relativa alle capacità del membro e il ruolo che ricopre nella squadra.

~ News

pagina xml usata per le notizie e collegata con una pagina xsl in cui vengono visualizzate le news riguardanti novità sul torneo, risultati delle partite, prossime partite, novità riguardanti nuove release del gioco e molto altro.

~ Gestione Notizie

la pagina utilizzata per la gestione delle news da visualizzare sulla home. Da qui è possibile modificare e/o cancellare una news. È una semplice pagina di inserimento di dati tramite form ed è accessibile solo dagli amministratori del sito. Infatti ad un click sul menù navigazione verrà richiesto nome utente e password per potervi accedere.

Layout e Css

Il layout di ogni pagina del nostro sito è composto da un header, un menù navigazione e un contenuto che formano due colonne affiancate, e poi un piepagina.

L'header è abbastanza complesso, questo perché abbiamo voluto creare un header con un forte impatto visivo e che fosse elastico al tempo stesso. E' composto da un contenitore principale div con id="header" all'interno del quale stanno tutti gli altri contenitori. Questo div principale contiene la prima delle molte immagini di sfondo, ossia l'immagine che in rapporto alla profondità verrà sovrastata da tutte le altre. All'interno dell'header si trova un contenitore span logo1, che contiene il logo principale dell'header che è il logo di urban terror. Allo stesso livello di logo 1, c'è un altro span con id="logo2" che contiene la pistola visualizzata sulla destra della pagina. Header, logo1 e logo 2 formano la rappresentazione principale dell'header, e la distinzione di 3 contenitori diversi con 3 immagini di sfondo rispetto ad 1 solo contenitore con un'unica immagine è stata necessaria per far si che nel ridimensionamento della pagina l'immagine non venisse tagliata, ma anzi che ogni contenitore si spostasse componendo un'immagine chiara anche per dimensioni ridotte.All'interno dello span logo1 si trovano altri 2 span titoloimg e scrittatitolo. Questi 2 contenitori contengono rispettivamente un'immagine del titolo e la scritta del titolo. L'immagine è visualizzata con o senza stile all'interno dell'header, mentre la scritta deve essere visualizzata solamente nell'anteprima di stampa in sostituzione all'immagine del titolo.

Sotto l'header sta un contenitore div con id="posizione", ed è un div che contiene il testo che indica la nostra posizione nel sito. Nonostante il sito sia piccolo e quindi sia veramente difficile perdersi, abbiamo scelto di visualizzare comunque questa barra per chiarezza. All'interno di posizione c'è un altro div con id="linkDaltonici", contenitore che presenta un link, collegato ad uno script javascript che serve per cambiare foglio di stile, in base alla visualizzazione che si vuole avere(daltonica o no).

Sotto al contenitore posizione, nella parte sinistra della pagina si trova il menù navigazione. Il menù navigazione è composto da un contenitore principale "nav" che al suo interno contiene 5 span di classe button. Dato che i button erano tutti uguali abbiamo pensato di accomunarli sotto un'unica classe di contenitori. Ogni span button contiene un link alla pagina specificata, esclusa la pagina che si sta visualizzando in quel preciso momento che contiene solo testo non linkato.

Affiancato al menù navigazione sta un altro contenitore chiamato content, che presenta il contenuto principale della pagina. All'interno del content sono stati utilizzati altri div, delle liste ordinate ul, li, i paragrafi p, e i fieldstet per delimitare l'area del div.

Infine sotto a tutti gli altri div si trova il div piepagina, che visualizza le icone del W3C che indica che il sito ha superato il test di validità rispetto alla specifica XML Strict e CSS, e poi del testo che indica l'ultima data di modifica del sito.

Project.css

Questo è il layout principale assegnato alla pagina web, impostato come media="screen". Da qualunque pc si stia visualizzando il sito web, questo sarà il layout principale caricato. Prima di analizzare nel dettaglio questo file css, va fatto notare che il layout ha anche un id="layout"; questo id servirà successivamente al javascript. Nel dettaglio, all'interno del file css ad ogni div viene settato uno stile appropriato.

- All’header viene settata la propria immagine di sfondo e la larghezza(100%), e un’altezza standard di 13em che impedisce che l’header diventi tanto grande da impedire la visualizzazione del content senza ricorrere allo scroll. Lo stesso viene fatto per gli span logo1 e logo2, trasformati in elementi di blocco, e posizionati uno a sinistra e uno a destra all’interno del loro contenitore. E’ importante notare che senza i posizionamenti con float: left e float:right i due contenitori non sarebbero riusciti a coesistere sul medesimo piano, e uno dei due sarebbe inevitabilmente scivolato al di sotto dell’altro. Ultimo particolare è che il contenitore span scrittatitolo viene settato invisibile tramite display:none, infatti non è questo il layout in cui deve venir visualizzato.

- Al div posizione viene impostata un’immagine di sfondo, un’altezza, un’identazione e il testo viene visualizzato in grasseto. Il div contenuto in esso, ossia linkDaltonici viene posizionato sulla destra con una larghezza del 65% in modo che il testo dell’uno e dell’altro div non si sovrapponga qualora la pagina venga ridimensionata. Oltre ad un’identazione da destra, viene cambiato il colore del link attivo in un giallo intenso, in modo che il contrasto con lo sfondo sia forte e che non vi sia difficoltà di lettura.

- Il div nav viene posizionato sulla sinistra con una larghezza fissa di 10em, onde evitare che ridimensionando la pagina i bottoni si sformino. Gli oggetti di classe button presentano le seguenti caratteristiche: sono tutti span trasformati in elementi di blocco, con un’altezza fissa a 3em mentre larghezza del 100% del contenitore; Il testo è identato di circa 10px da sinistra, e viene impostata un’immagine di sfondo per rendere l’idea del bottone. Abbiamo implementato che al passaggio su un button, l’immagine di sfondo cambiasse, mantenendo per il resto le stesse impostazioni di un bottone non selezionato.

- La classe key è stata creata appositamente per evidenziare in un link l’accesskey corrispondente. L’accesskey sarà sottolineata, in grassetto, e di colore nero. Oltre a key esiste una variante keyCNav che serve per visualizzare le accesskey del menù scorciatoia del content di colore bianco.

- Il div content è posizionato affiancato a destra al menù navigazione,con una larghezza in percentuale, in modo che per quanto possibile si adatti ad una risoluzione minore. Purtroppo la presenza del menù nav con larghezza fissa rende la risoluzione minima della pagina piuttosto limitata. Oltre a questo, il content presenta un pudding di 1em che lo incornicia e testo di colore bianco in contrasto con lo sfondo.All’interno di content troviamo ContentNav, ossia un contenitore che contiene i vari link scorciatoia per raggiungere le varie voci della pagina. La larghezza di ContentNav combacia con quella di content e il testo visualizzato sempre nel centro del div.All’interno di content inoltre ad ogni div è stato impostato un pudding di 3em di modo che i div siano distanziati l’uno dall’altro.

- Infine il div piepagina è impostato con un pudding minino di 0.2em, allineamento del testo a sinistra e il colore del testo sempre di bianco. La proprietà clear è impostata su both in modo che il div stia al di sotto di tutti gli altri.

Ecco una visualizzazione della pagina standard, in larghezza standard e minima:

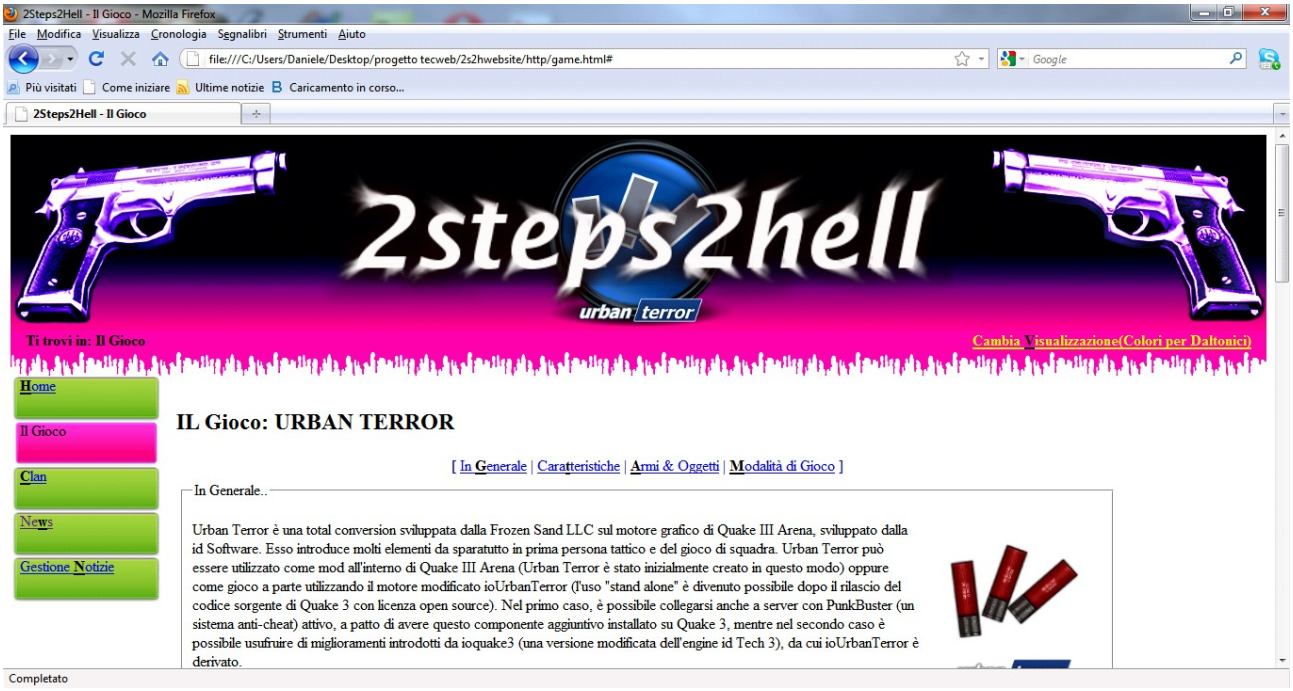
Fullscreen:

Minima:

daltonized.css

Questo file css contiene il layout per una visualizzazione in modalità daltonica. Come layout in sé non è molto diverso da project.css se non per il fatto che su header, logo1, logo2, posizione e button le immagini caricate sono immagine con una tonalità di colorazione diversa, appositamente regolata grazie al sito vischeck per una buona visualizzazione anche per utenti con difetti visivi. Lo sfondo è trasformato da nero a bianco, e il testo nero, così che il contrasto sia più forte e quindi il testo sia più facilmente leggibile. Degli altri accorgimenti è importante sottolineare che le keyCNav, classe per visualizzare le accesskey del menù scorciatoia nel content sono colorate di nero contrariamente al colore bianco che hanno sul foglio di stile principale.

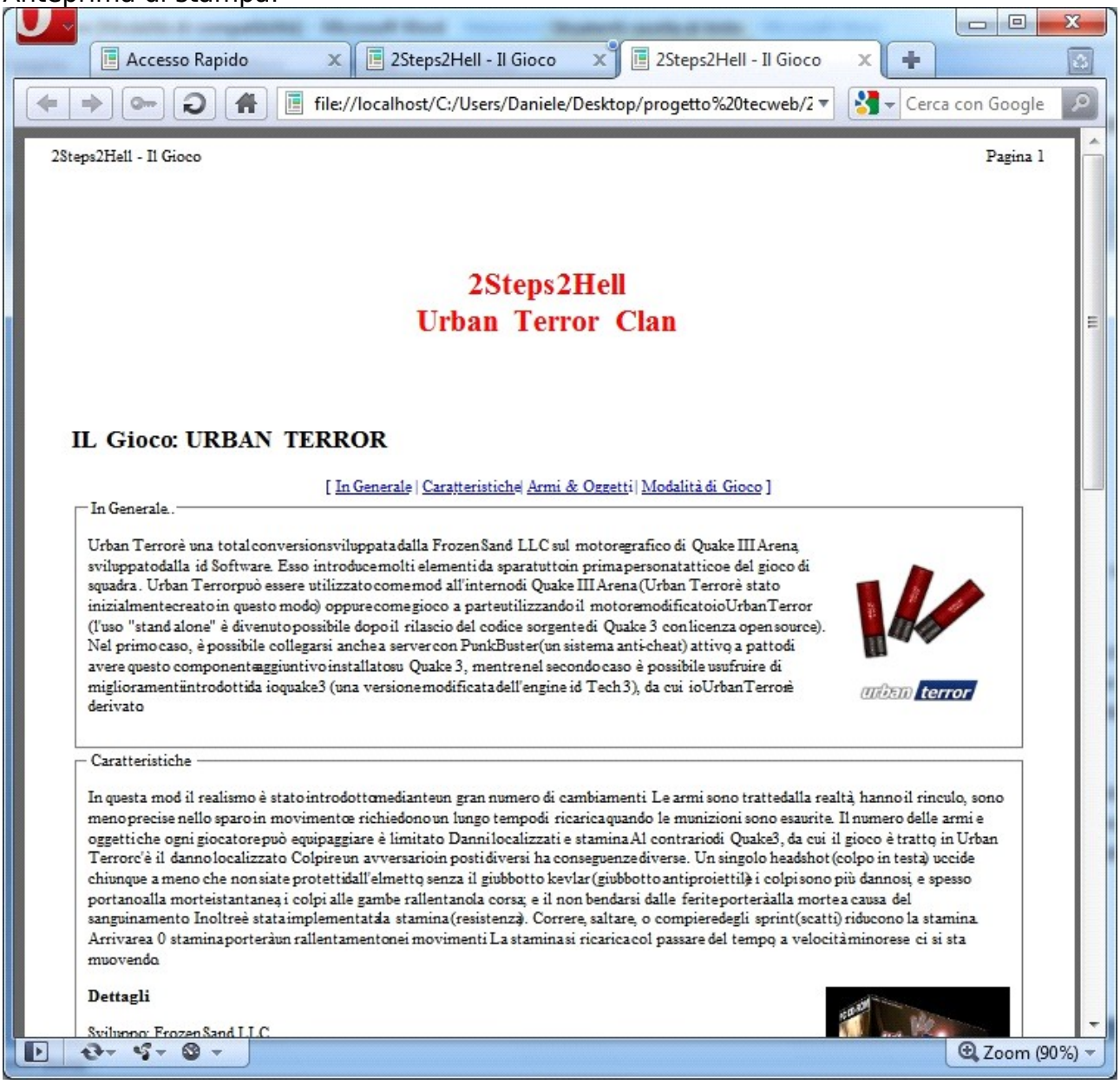
Esempio di pagina daltonizzata:



print.css

Questo file css è il layout per a stampa della pagina web, impostato come media=”print”. Abbiamo pensato che per la stampa non occorressero immagini di layout, perciò abbiamo deciso di rimuovere qualunque immagine lasciando solamente il testo. Oltre alla rimozione dell’immagine, vengono rimossi il div posizione, il div nav e il div titoloimg, mentre è reso visibile il div scritta titolo, che contiene la scritta di colore rosso(che richiama quindi un po’ il tema della pagina) che rimpiazza l’immagine del titolo proprio per la stampa. Lo sfondo di ogni div della pagina è bianco, mentre il testo e ogni altro elemento escluse le immagini di contenuto sono in scala di grigi.

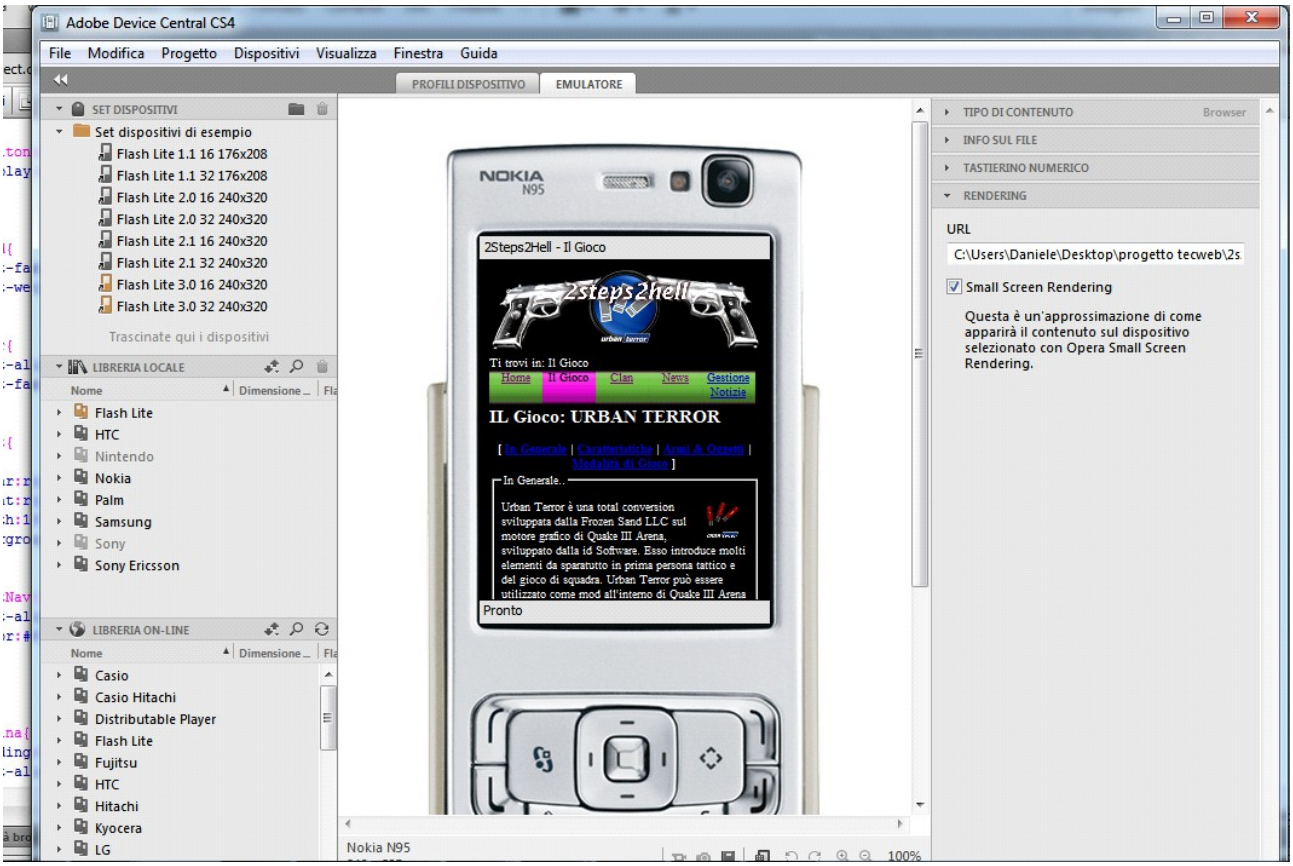
Anteprima di stampa:



handheld.css

Rappresenta il layout della pagina web visualizzabile dai dispositivi portatili. In base alla nostra analisi, un dispositivo portatile con uno schermo piuttosto piccolo avrebbe limitato molto la visualizzazione del contenuto del sito. Perciò abbiamo pensato di proporre lo stesso header del layout normale, ma con delle dimensioni ridotte. Inoltre il menù navigazione posto a lato del contenuto avrebbe ridotto ulteriormente la visualizzazione del content, perciò abbiamo pensato di spostare i bottoni del menù navigazione subito sotto la barra posizione. Non abbiamo aggiunto alcuna scorciatoia per saltare la navigazione, dato che il contenuto è comunque visualizzabile nonostante l'header e il menù nav, e per i contenuti più sostanziosi all'inizio di ogni pagina c'è un ContentNav che ci aiuta a raggiungere il contenuto di nostro interesse anche in fondo alla pagina.Per il resto il layout non differisce da quello principale, se non che è stato rimosso il link per il cambio di visualizzazione, per ovvi motivi di compatibilità, e per rendere il sito visibile a una persona con problemi visivi abbiamo rimosso alcune immagini di sfondo e utilizzato per i bottoni colori neutri che non creassero problemi visivi ne a persone con problemi visivi, ne a persone comuni. Le accesskey non sono più visibili, dato che sono inutilizzabili su un cellulare.

Esempio di visualizzazione su cellulare:



Accorgimenti sull’accessibilità

Questa sezione è una breve descrizione degli accorgimenti che abbiamo fatto per rendere il sito accessibile ad un pubblico vasto.

- Ogni link è visualizzato dal colore che lo contraddistingue in qualunque sito web, ossia blu sottolineato. Solo in un caso abbiamo modificato il blu in giallo, per rendere il contrasto con lo sfondo più evidente.

- Ogni link è identificato da un’accesskey, realizzata con la classe di contenitori span key che ritrae la accesskey indicata di colore nero, in grassetto e sottolineata. In altro contesto l’accesskey è bianca al posto che nera, per contrasto con lo sfondo. Così la presenza dell’accesskey è indicata in modo evidente e allo stesso tempo non richiede un avvertimento testuale della loro presenza.

- Oltre alle accesskey sono presenti i tabindex, che danno un’ordine agli elementi selezionati col premere del tasto tab. Abbiamo pensato che in quelle pagine in cui vi fossero dei piccoli menù scorciatoia(usati per raggiungere le varie parti in cui è diviso il testo) fosse più importante per un utente scorrere prima le varie argomentazioni della pagina, e poi il menù navigazione. Quindi abbiamo organizzato i tabindex in modo che scorrano prima i link “interni” al content, e poi i link “esterni”.

Lo stesso vale se all'interno del content della pagina sono presenti delle form. Tramite tabindex saranno selezionate, quindi, per prime le form e poi i link del menù di navigazione.

Nonostante questo però abbiamo preferito riservare la prima posizione del tabindex al link per il cambio di visualizzazione, qualora un utente con disabilità incapace di visualizzare correttamente la pagina debba prima cambiarne la visualizzazione e poi visualizzarne il contenuto.

- Ogni immagine presenta un attributo alt, che ha lo scopo qualora l’immagine non venisse caricata, di visualizzare una breve riga di testo fornendo informazioni riguardo ad essa. L’assenza delle immagini non comporta stravolgimento del layout, che resta comunque ordinato.

- Sono stati inseriti i metatag necessari per la corretta visualizzazione della pagina in una buona posizione nei risultati di ricerca di un motore di ricerca. Questi metatag sono: description, breve descrizione della pagina in questione; keywords, ossia le parole chiave con cui è più semplice che il motore di ricerca identifichi la nostra pagina come target di ricerca; author, gli autori della pagina; language, linguaggio adottato all’interno della pagina.

- Per quanto riguarda il testo, ogni parola in lingua diversa dall’italiano(inglese) è stata appropriatamente indicata con lo span xml:lang=”en” che permette agli screen reader di riconoscere la parola e di “leggerla” correttamente in lingua inglese.

- Non sono state utilizzate tabelle, che sono difficilmente accessibili, non essendo necessaria la loro presenza per il nostro sito.

Javascript

Il javascript è stato inserito nella pagina per implementare il cambio di visualizzazione in modo semplice.All'interno del file provaScript.js si trovano 5 funzioni; di queste setCookie, del Cookie e getCookie sono quelle che ci sono state introdotte a lezione, e hanno lo scopo rispettivamente di creare, cancellare o leggere un cookie. Verranno utilizzati nelle altre funzioni. Le nostre creazioni sono leggiCookie e daltonizza.

- LeggiCookie() viene lanciato al caricamento della pagina e verifica la presenza del cookie che indica se la pagine deve essere visualizzata con colori particolari o no. In caso affermativo, viene ricercato e selezionato l'elemento con id="layout", che non è altro che il foglio di stile principale che viene caricato per la pagina, e all'attributo href viene sostituito un nuovo indirizzo dove è presente il foglio di stile adatto, ossia daltonized.css.

- Daltonizza() viene caricato al click sul link per la visualizzazione daltonica. Il suo compito è verificare la presenza o meno del cookie per la visualizzazione speciale, e in caso affermativo controllare il valore del cookie e caricare il foglio di stile corrispondente. Se il cookie contiene il valore "dalt" significa che l'utente ha cliccato sul link e vuole tornare alla visualizzazione standard,perciò verrà cancellato il vecchio cookie, ne verrà settato uno nuovo con valore "nonDalt" e verrrà caricato all'elemento layout il foglio di stile project.css; se invece il cookie contiene il valore "nonDalt" oppure il cookie non era neppure stato settato, allora viene creato il cookie con valore "dalt" e viene caricato all'elemento layout il foglio di stile daltonized.css.

  Esiste anche una sorta di copia di provaScript, che è prova Script-cgi.js che è lo script che invoca la pagina di gestione notizie che si trova in una directory diversa e quindi ha bisogno di indirizzi diversi per andare a prendere i file css e sostituirli nel documento.

XSL & XML

Per la visualizzazione delle news abbiamo utilizzato l'xsl. Il file in questione è “newsReader.xsl”.

Questo file è “linkato” con il file-database di notizie “newsDB.xml”. All'interno del file xsl troviamo la pagina web da visualizzare con una sezione dove vengono organizzati, in ordine decrescente (ovvero dalla più recente a quella più vecchia), le notizie inserite.

Ecco il pezzo di codice in questione:

```
<!-- output notizie -->
        <xsl:for-each select="news/article">
        <xsl:sort order="descending" select="date"/>
          <fieldset>
            <legend class="newsLgd" >
              <xsl:value-of select="title"/>
            </legend>

            <p class="newsPar" >
              <xsl:value-of select="body"/><br/>
              Autore: <xsl:value-of select="author"/><br/>
              In Data: <xsl:value-of select="date"/><br/>
            </p>
          </fieldset>
          <br/>  <!-- linea vuota per dividere le notizie -->
        </xsl:for-each>
```

Come contenitore per le notizie abbiamo utilizzato un file XML (con relativo XML Schema), strutturato come segue:

```
<news>
        <article id=”1”>
                <title></title>
                <date></date>
                <body></body>
                <author></author>
        </article>
        <article id=”2”>
                …
        …
</news>
```

I tag body contengono un campo CDATA, in modo da permettere l'inserimento di tag xhtml, per poter usare visualizzazioni avanzate (nel qual caso servirebbero script un poco più avanzati, noi li abbiamo fatto basilari, quindi ora come ora i tag xhtml non sono controllati, e non funzionano con l'xslt, vengono visualizzati invece che interpretati, mentre nella lista generata dallo script perl funzionano).

I tag della struttura non hanno bisogno di commenti, in ogni caso contengono, nell'ordine: il titolo, la data dell'inserimento o ultima modifica, il contenuto e l'autore.

PERL

Tutta l'amministrazione delle news è basata su script in perl (cgi).

I moduli utilizzati sono CGI, XML::LibXML e XML::Twig.

Abbiamo sei script, ognuno per uno scopo specifico, che utilizzano i file head.html e foot.html per impostare la pagina html (intestazone, menu' ecc sono comuni a tutti gli script e fissati, cambia solo il contenuto).

Vediamo singolarmente la loro funzione:

-listaNews.cgi:

> Lo script principale, aperto subito dopo aver inserito nome utente e password, presenta l'intera lista delle news ordinate dalla più vecchia alla più recente, preceduta dal link per inserirne una nuova. All'interno di ogni news sono poi presenti i link per modificare o eliminare la news.

-deleteNews.cgi:

> Elimina una news dal file XML, prendendo in input con sistema GET l'id della news da eliminare.

-editFormNews.cgi:

> Riceve come input dalla lista l'id della news da modificare, e presenta all'utente il form per la modifica.

-editNews.cgi:

> Riceve in input i dati modificati da editFormNews.cgi, cerca la news nel file XML e ne modifica i dati.

-newFormNews.cgi:

> Presenta all'utente il form per l'inserimento di una nuova news.

-addNews.cgi:

> Riceve in input i dati da newFormNews.cgi, e crea una nuova news all'interno del file XML.

Gli script di inserimento e modifica inoltre, prima di salvare le modifiche, fanno un test per controllare se l'XML rimane ancor valido rispetto all'XML Schema, e ne sistemano l'output indentandolo correttamente.

Sicurezza (accesso)

Per quanto riguarda la sicurezza (accesso con password), tutti gli script sono protetti da password impostata tramite una coppia di file, .htaccess e .htpasswd, all'interno della cartella cgi-bin.

Le credenziali di accesso impostate di default sono:

-user: zamy

-password: elettroni

Perchè funzioni corettamente bisogna essere sicuri che nel file .htaccess sia impostato correttamente il percorso del file .htpasswd (per esempio, /var/www/cgi-bin/.htpasswd).

All'interno del file .htpasswd vanno inseriti gli utenti, in formato utente:password , dove la password è codificata (a questo proposito, si trovano tanti script di codifica utili online).