

Progettazione e Produzione Multimediale: progettazione e sviluppo di un'applicazione web che permetta il gioco del puzzle con opere d'arte

Andrea Rossini

14 gennaio 2023

Sommario

Per questo progetto relativo all'esame di Progettazione e Produzione Multimediale si vuole realizzare un gioco su grande schermo, ad interazione naturale, che consenta a 2 giocatori affiancati nello sfidarsi nel ricomporre, il più velocemente possibile, l'immagine di alcune opere d'arte presentate in sequenza e scomposte in quadrati "mischiati" (proprio come un puzzle); il vincitore visualizzerà sullo schermo le informazioni dell'opera ricostruita correttamente (titolo e breve descrizione).

1 Introduzione

L'obiettivo dell'esercizio è quello di realizzare la suddetta applicazione rispettando i seguenti requisiti:

- Utilizzo di javascript e php,
- Creazione di un database con backend configurabile,
- L'applicazione deve essere responsive (adattabile in ambienti mobile e desktop)

Il gioco procede nei seguenti passi:

1. I due giocatori inseriscono i propri nomi in una pagina di login (sezione 2.2);
2. Una volta effettuato l'accesso, si viene indirizzati alla pagina principale (sezione 2.3) dove, inizialmente, si vedrà un'anteprima di come verranno mostrate le immagini da ricomporre (figura 1). I giocatori hanno anche la possibilità di leggere le regole del gioco (premendo l'apposito bottone);
3. Quando si decide di iniziare la partita, il primo giocatore è abilitato a ricomporre l'immagine che verrà presentatagli: dovrà farlo nel più breve tempo possibile (nel frattempo, il secondo giocatore vede scomparire il suo riquadro, al fine di non poter anticipatamente studiare il suo puzzle ed essere poi avvantaggiato);
4. Una volta che il giocatore 1 termina, appare il riquadro del giocatore 2, il quale, a sua volta, dovrà ricomporre l'immagine mostrata;
5. Quando anche il giocatore 2 finisce, il gioco confronta i tempi impiegati da entrambi, e colui che "ha fatto prima" vince la partita, vedendo comparire un riquadro con il titolo ed una breve descrizione dell'opera corrispondente, come mostrato in figura 2;

1.1 Documentazione del codice

Il programma sviluppato contiene al suo interno i seguenti file:

- "index.html", che rappresenta la schermata iniziale del gioco appena questo viene aperto; ha lo scopo di far inserire ai 2 giocatori i loro nomi prima dell'effettivo avvio della sfida.
- "mainPage.php", ossia, come dice il nome, la pagina principale del gioco nella quale i 2 giocatori si sfidano.



Figura 1: Schermata principale del gioco dopo aver effettuato il login.

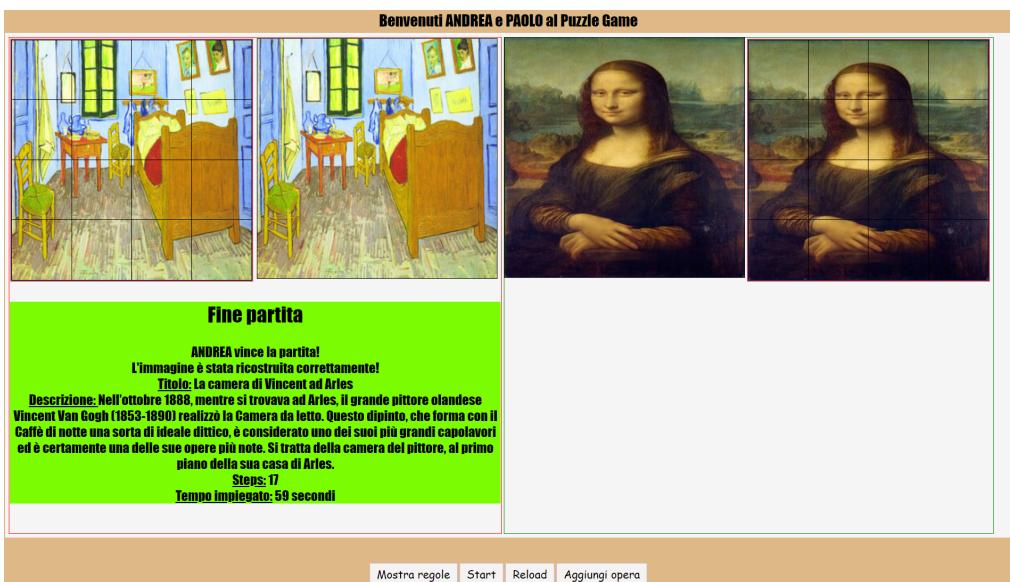


Figura 2: Schermata principale alla fine di una partita: in questo caso, ANDREA ha impiegato meno tempo di PAOLO a terminare il puzzle, quindi vede comparire le informazioni relative alla propria opera.

- "puzzle-game.js", contenente di fatto tutto il codice necessario all'esecuzione della partita che si sta svolgendo sulla "mainPage.php".
- "startStyle.css", che contiene un piccolo foglio di stile per la pagina iniziale di login.
- "puzzle-game.css", ossia il foglio di stile della pagina principale del gioco.
- vari file "mediaScreen-.css", che contengono le media queries per diverse dimensioni dello schermo.
- "addImage.php", una pagina che permette agli utenti di aggiungere manualmente le opere che preferiscono.
- "insert.php", il file php che si occupa di caricare l'immagine tramite FTP e di inviare i dati inseriti in *addImage* al database online.
- la cartella "images", contenente le immagini delle opere che verranno mostrate all'interno del gioco.

2 Struttura

2.1 Considerazioni preliminari

Specifiche del calcolatore: Il codice è stato sviluppato e testato interamente su un ACER Swift SF314-52, con processore Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz 2.71 GHz e 8 GB di RAM; il Sistema Operativo è Windows 10 Home con architettura a 64 bit.

Ambiente di sviluppo: Il codice è stato scritto interamente con l'IDE PhpStorm 2022.2.3, con versione di PHP 8.1.

Browser utilizzato: Google Chrome.

2.2 Pagina iniziale di login

Questa semplice pagina ha il solo scopo di permettere ai giocatori di inserire i loro nomi; è formata dall'immagine dell'opera "Bonaparte valica il gran San Bernardo", sotto la quale viene posta, con tag <h2>, una frase di benvenuto. Sotto a questa viene un <h3> che chiede di inserire i nomi negli spazi sottostanti: questi, tramite un <form> con l'attributo *method* settato a "post", verranno quindi inviati alla pagina principale (*action* = mainPage.php). La struttura per inserire le informazioni richieste è ottenuta tramite una <table> con 2 righe: ogni <tr> è composta da un <td> contenente la <label> con l'id dell'elemento e un <input> con *type* = text per digitare il nome (N.B: è stato marcato come *required*, dal momento che non è possibile far partire il gioco se entrambi i giocatori non hanno eseguito questa azione).

Ancora sotto vi è un <button> con *value* = "Nuova partita" e *type* = submit in modo da inviare tutti i dati del form al suo gestore; quando viene premuto, si passa alla pagina principale del gioco.

2.3 Pagina principale

Dopo aver inserito i nomi e premuto sul tasto "Nuova partita", si viene indirizzati sulla schermata principale, nella quale i 2 giocatori vedranno un esempio di come verranno presentate loro le immagini da ricostruire, insieme ai pulsanti che permettono di visualizzare le regole, far partire il gioco, ricaricare l'immagine e aggiungere manualmente un'immagine a piacimento. La sua struttura è la seguente:

- Un <h3> che, tramite PHP, pesca i nomi dei 2 utenti e li stampa in caratteri maiuscoli (tramite la funzione *strtoupper()*) all'interno di un messaggio di benvenuto;
- Un <p> che, in caso di larghezza dello schermo troppo piccola, avvisa di ruotare quest'ultimo per poter usufruire dell'applicazione;
- il <div> *mainPanel*, che contiene lo spazio di gioco degli sfidanti; questo viene suddiviso nei due <div> *player1* e *player2*;

- Entrambi gli spazi per i giocatori sopra citati contengono: un `` (*fillable* e *fillableSecondPlayer*, rispettivamente) rappresentante la griglia del puzzle, vuota inizialmente, dove andranno inseriti i pezzi dell'immagine; un `<div>` contenente l'immagine dell'opera da ricostruire (*originalImageBox* e *originalImageBoxSecondPlayer*, rispettivamente); un altro `` che racchiude la griglia con i pezzi del puzzle "mischianti" (*sortable* e *sortableSecondPlayer*, rispettivamente) e, infine, uno `` (*showEndGame*, lo stesso per entrambi) che mostrerà al vincitore le informazioni sull'opera corrispondente;
- Una serie di `<div>` che rappresentano il tempo impiegato e gli step eseguiti da ciascun giocatore (*currentTimeBox*, *currentTimeBoxSecondPlayer*, *numStepBox* e *numStepBoxSecondPlayer*);
- Il `<div> buttons`, il quale contiene i 4 pulsanti per le già citate funzioni: *rules* mostra un popup con le regole del gioco (chiamando la funzione *showRules()* definita nello script javascript in fondo alla pagina); *startGame* chiama la funzione *start()* per dare inizio al gioco, cambiando le immagini e facendo partire cronometro e contatore dei passi; *changeThePhoto* chiama l'omonima funzione per cambiare l'immagine, qualora un giocatore non fosse soddisfatto di quella assegnatagli, e di conseguenza far ripartire da 0 il cronometro (lasciando però inalterato il numero di passi); *addImage* permette di passare ad una pagina dove poter inserire manualmente nel database l'immagine di un'opera a scelta.

Quando i giocatori decidono di iniziare la partita, si preme il pulsante Start: a questo punto il riquadro del giocatore 2 scompare e giocatore 1 inizia il suo turno. Nell'istante in cui il secondo giocatore termina il suo puzzle, viene visualizzata l'informazione dell'opera del vincitore (viene rimossa la proprietà *display: none* del `<div>` *showEndGame*) insieme al titolo, i passi effettuati ed il tempo impiegato (tutte informazioni che vengono recuperate dal database online *my_artpuzzle*).

3 Implementazione

Di seguito, un'illustrazione di come sono state implementate le principali componenti dell'applicazione.

3.1 Stile delle pagine

Sia nella pagina di login che in quella principale è stato scelto di utilizzare come tipo di font il carattere **Copperplate Gothic Bold fantasy** (visibile [qui](#)) e come colore di sfondo **burlywood** (#DEB887). Come unità di misura dei font-size viene usata la viewport width (**vw**): questo perchè tale unità risulta molto utile al fine di rendere l'applicazione responsive, dal momento che permette al testo di adattarsi alla larghezza della pagina visualizzata in un particolare browser.

3.2 Costruzione delle griglie

Per costruire le 4 griglie previste nel gioco, si seguono i seguenti passi:

1. Si calcola il parametro $percentage = 100/(gridSize - 1) = 33.3333$ (con gridSize = 4, essendo la griglia 4x4): esso servirà per posizionare correttamente le celle nelle griglie;
2. Per ogni $i = 0, \dots, 15$ si calcolano 4 parametri: **posX** = $percentage * (i\%gridSize)$, in modo da posizionare le celle 4 per volta in orizzontale (0%, 33.333%, 66.667% e 99.999%), **posY** = $percentage * Math.floor(i/gridSize)$, così da posizionare 4 celle per riga (infatti, il valore rimarrà lo stesso per ogni "blocco" di 4 celle, consentendo a queste di essere posizionate nella corretta riga), **posXHorizontalShift** e **posYVerticalShift**, calcolati come i precedenti, ma "shiftati" in orizzontale e verticale;
3. Le 4 griglie hanno quindi le seguenti coordinate: *fillable* = (posX, posY) (in alto a sinistra), *sortable* = (posX, posYVerticalShift) (stessa ascissa di fillable ma ordinata diversa, in basso a sinistra), *fillableSecondPlayer* = (posXHorizontalShift, posY) (stessa ordinata di fillable e ascissa diversa, in alto a destra) e *sortableSecondPlayer* = (posXHorizontalShift, posY-VerticalShift) (in basso a destra, stessa ascissa di fillableSecondPlayer e stessa ordinata di sortable);

- Alle due sortable vengono poi assegnate come immagine di background le 2 opere estratte a caso dal database, quindi si esegue la funzione `shuffle()` su di esse (che altro non fa che mischiare gli elementi della lista che gli viene passata, quindi "mescola" le varie celle con i pezzetti di immagine).

3.3 Gestione spostamento caselle

Per quanto riguarda lo spostamento delle celle, è stato fatto uso degli eventi Drag & Drop (questo per poter verificare il corretto funzionamento su PC, per la compatibilità con i dispositivi touch è stato integrato lo script `dragDropTouch.js`, che converte i suddetti eventi in eventi touch).

Quindi, al verificarsi dell'evento **ondragstart** (che può avvenire sia su fillable che su sortable, dal momento che è possibile spostare una casella anche se già inserita nella griglia di destinazione in caso di errore) si salva l'id della casella selezionata, mentre in **ondragover** si disabilita il comportamento di default che impedirebbe il trascinamento di elementi tramite `preventDefault()`; quando si verifica **ondrop** su fillable, si tiene traccia di:

- **source**, ossia l'id della cella di origine;
- **destination**, ossia l'id della cella di destinazione;
- **p**, ossia il parentNode della destinazione (serve per verificare che la cella di destinazione sia valida, essendo "figlia" della griglia fillable: in pratica si vuole effettuare un ulteriore controllo che la casella dove stiamo rilasciando l'elemento trascinato faccia effettivamente parte della corretta griglia di destinazione).

Se i 3 precedenti parametri sono validi, si aggiunge la sorgente tra i figli della cella destinazione e si crea l'array `valuesId` composto da tutti i figli di sortable, ossia i pezzi dell'immagine, e si verifica che siano o meno ordinati correttamente tramite la funzione `isImageSorted()`.

Quando entrambi i giocatori hanno correttamente riordinato le immagini assegnate loro (funzione `manageSecondPlayer()`), si guarda chi ha impiegato meno tempo e, a seconda del vincitore, viene fatto comparire l'endGame con le informazioni relative all'opera, insieme alla disabilitazione temporanea del trascinamento delle celle (che verrà riabilitato all'avvio di una nuova partita).

3.4 Caricamento manuale di immagini nel database

Al fine di permettere agli utenti di avere la possibilità di aggiungere manualmente al gioco le immagini delle opere che preferiscono, è stata prevista una pagina (raggiungibile premendo il tasto **"Aggiungi opera"** in `mainPage.php`) dove inserire i campi necessari alla corretta visualizzazione dell'opera (quindi titolo, autore, data, descrizione, il file dell'immagine ed il path): selezionando, a questo punto, il tasto **"Carica"**, prima i dati inseriti vengono inviati ed inseriti nel database tramite la query SQL `INSERT INTO images (title, author, date_of_realization, description, located_at_path) VALUES (...)` (figura 3) e successivamente viene eseguito il caricamento dell'immagine tramite FTP.

(N.B: purtroppo, anche avendo la certezza che il codice scritto funziona, è stato scoperto che Altervista (servizio di hosting scelto) non permette di abilitare le funzionalità di trasferimento FTP, ed avendo realizzato tale funzionalità molto a ridosso della data di esame, non è stato possibile trasferire l'applicazione su un altro servizio di hosting che supportasse FTP in tempi brevi (e soprattutto senza pagare..)).

3.5 Supporto schermi minori

Per rendere l'applicazione utilizzabile anche dai dispositivi con schermi più piccoli (come smartphone e tablet), oltre al già citato script `dragDropTouch.js` e all'utilizzo della `viewport width` come unità di misura dei `font-size`, vengono utilizzate una serie di media queries che riposizionano e ridimensionano i vari elementi in base alla larghezza dello schermo del dispositivo utilizzato; una volta raggiunta una larghezza massima troppo piccola ($\leq 360px$) per permettere un'esperienza d'uso soddisfacente per gli utenti, entrerà in azione una media query che farà comparire un'esortazione a ruotare lo schermo del dispositivo per poter giocare.

4 Messa online dell'applicazione

Una volta scritto il codice, è stato scelto Altervista come servizio di hosting online. Il sito web creato "riceve" il codice sorgente grazie a FileZilla, al quale forniamo la cartella contenente il

```

<?php

$ftp_server = "ftp.artpuzzle.altervista.org";
$ftp_username = "artpuzzle";
$ftp_password = "YHWH5KCr2FCY";

// Restituisce il filename temporaneo del file in cui viene memorizzato sul server
// il file caricato
$file = $_FILES['file']['tmp_name'];
// Restituisce il nome originale del file da caricare
$remote_file = $_FILES['file']['name'];
// connection_id è l'handler
$connection_id = ftp_connect($ftp_server);

$login_result = ftp_login($connection_id, $ftp_username, $ftp_password);

if ((!$connection_id) || (!$login_result)) {
    die("Connessione FTP fallita...");
}

if (ftp_put($connection_id, $remote_file, $file, mode: FTP_BINARY)) {
    echo "file caricato con successo";
} else {
    echo "caricamento del file fallito";
}

// Chiudi connessione FTP
ftp_close($connection_id);

$conn = mysql_connect( hostname: "localhost", username: "artpuzzle", password: "", database: "my_artpuzzle");

// Check connection
if($conn === false){
    die("ERROR: Could not connect.
        . mysql_connect_error());
}

// Taking all 5 values from the form data(input)
$title = $_REQUEST['title'];
$author = $_REQUEST['author'];
$date_of_realization = $_REQUEST['date_of_realization'];
$description = $_REQUEST['description'];
$located_at_path = $_REQUEST['located_at_path'];

// Inserimento query
$sql = "INSERT INTO images (title, author, date_of_realization, description, located_at_path) VALUES ('$title',
    '$author', '$date_of_realization', '$description', '$located_at_path')";

if(mysql_query($conn, $sql)){
    echo "<h3>I dati sono stati inseriti nel database.
        . Per vedere i cambiamenti aprire phpMyAdmin</h3>";
    echo nl2br( string: "\n$title\n $author\n "
        . "$date_of_realization\n $description\n $located_at_path");
} else{
    echo "ERROR: Hush! Sorry $sql.
        . mysqli_error($conn);
}
}

```

Figura 3: Codice PHP per caricamento immagine tramite FTP e invio dati inseriti al database tramite query SQL.

	<input type="button" value="←"/>	<input type="button" value="→"/>	<input type="button" value="id"/>	<input type="button" value="title"/>	<input type="button" value="author"/>	<input type="button" value="date_of_realization"/>	<input type="button" value="description"/>	<input type="button" value="located_at_path"/>
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	1 Statua della Libertà	Auguste Bartholdi	1866-10-28	La Statua della Libertà (Statue of Liberty) è un m...	images/statue-of-libertyResized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	2 Vlandante sul mare	Caspar David Friedrich	1818-10-07	Uno dei temi del romanticismo è quello del viaggio...	images/Vlandante-sul-mare-resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	3 David	Michelangelo Buonarroti	1501-06-18	David è una famosa scultura creata da Michelangelo...	images/David-Resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	4 Bonaparte valica il Gran San Bernardo	Jacques-Louis David	1800-09-12	Questa frase si riferisce all'evento in cui Napo...	images/napoleone-Resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	5 L'urlo	Edvard Munch	1893-03-02	"L'urlo" è un famoso dipinto dell'artista norveg...	images/Urlo-resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	6 Il Bacio (di Hayez)	Francesco Hayez	1859-07-06	"Il bacio" è un famoso dipinto dell'artista itali...	images/bacio-hayez-resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	7 Il Bacio (di Klimt)	Gustav Klimt	1907-08-07	"Il bacio" è un famoso dipinto dell'artista austri...	images/il-bacio-di-klimt-resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	8 Notte stellata	Vincent Van Gogh	1889-01-10	"La notte stellata" è un famoso dipinto dell'arti...	images/notte-stellata-resized.jpg
<input type="checkbox"/>	<input type="button" value="Modifica"/>	<input type="button" value="Copia"/>	<input type="button" value="Elimina"/>	9 La Ragazza con l'orecchino di perla	Jan Vermeer	1665-05-08	"Ragazza con l'orecchino di perla" è un famoso di...	images/ragazza-orecchino-resized.jpg

Figura 4: Campi che compongono la caratterizzazione delle immagini nella cartella *images* su PhpMyAdmin.

progetto (dopo aver inserito host e password).

Quindi, per la creazione del database, le immagini delle opere vengono caricate da locale a remoto sempre tramite FileZilla e, in seguito, si crea (con l'utilizzo di SQL in PhpMyAdmin) il database my_artpuzzle con al suo interno la cartella *images* contenente 6 campi (figura 4):

- **id**: un intero (auto_increment, in modo da incrementarlo ogni volta che viene aggiunto un nuovo elemento alla cartella) che rappresenta il numero identificativo dell'immagine rappresentata;
- **title**: un *varchar*(50) che caratterizza il titolo dell'opera;
- **author**: *varchar*(30) che rappresenta l'autore dell'opera;
- **date_of_realization**: una *date* indicante la data di completamento dell'opera;
- **description**: *varchar*(1000) per la breve descrizione dell'opera;
- **located_at_path**: *varchar*(50) per il percorso dove trovare l'immagine

Al fine di recuperare dal database online le immagini delle opere e le informazioni relative (come il titolo, la data di realizzazione e la descrizione), al caricamento di *mainPage.php* viene eseguito il codice PHP mostrato in figura 5.

L'applicazione può essere provata all'indirizzo <https://artpuzzle.altervista.org>.

```

<?php

$connection = mysqli_connect( 'localhost', 'username: 'artpuzzle', 'password: '' , database: 'my_artpuzzle'); // Establishing Connection with Server
if (mysqli_connect_errno()) {
    echo "Failed to connect to MySQL: " . mysqli_connect_error();
}

$path="";
$title="";
$query = mysqli_query($connection, query: 'select * from images');
$row_number = mysqli_num_rows($query);
$rows = array();

echo "<script> const images = [];</script>";

while ($row = mysqli_fetch_array($query)) {
    $path = $row['located_at_path'];
    $title = $row['title'];
    $description = $row['description'];

    echo "<script>
        images.push({'src': '$path', 'title': '$title', 'description': '$description'});
        </script>";
}

?>

<script>

```

Figura 5: Codice PHP che permette il collegamento al database online e il reperimento delle immagini e delle relative informazioni.