

Design Report

Introduction

The project of this coursework is an interactive user client web programming consisting of CSS, HTML5, javascript (JS), and media materials. This client mainly contains two parts: video player and web game with 2 HTML5 files. The video player page realizes the operation function of the video file through JS and the function of play all by constructing a continuous playlist. In addition, another web page is a card matching game that can earn scores by looking for the same cards. Therefore, this report would introduce the project of this coursework from the perspective of style design and features implementation.

Stylesheet Design

The stylesheet code of this project is written in the CSS file. In this part, this report will introduce the client interface design from theme, consistency, dynamic creation, and some details.

This client adds hyperlinks in the header section for redirection to improve the connection between the two web pages.

The video resources used by the video player this time are all movie trailers, so the page design of the video player is also based on the movie playback website. In order to provide a better sense of immersion[4], the background color of the video player is dark gray and black, and the video content part is filled with black, just like many famous streaming video websites[1]. Furthermore, in order to facilitate video switching and keep the integrity of the page, the playlist part is placed on the right side of the video play window. The screenshot of the video player is shown in Figure 1.

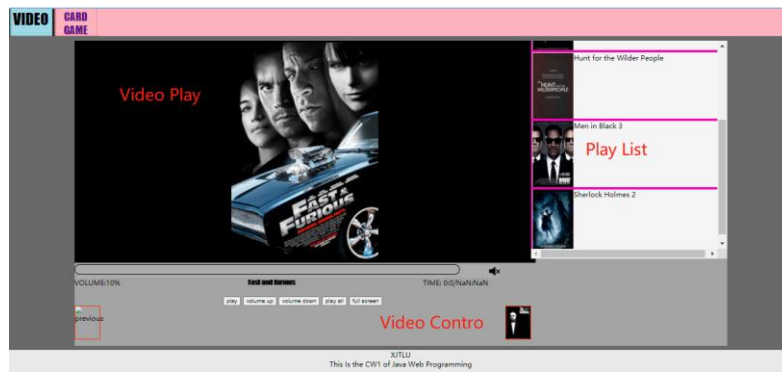


Figure 1. The video play screenshot

On the other hand, in order to create a relaxed atmosphere in the game interface, light green color is used instead. The screenshot of this card matching game is shown in Figure 2.

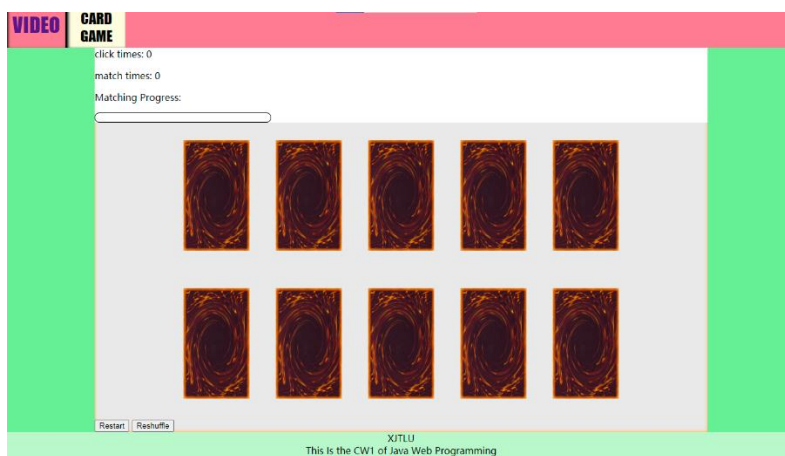


Figure 2. the screenshot of game

At the same time, in order to maintain the consistency of the two web pages when jumping, the header and footer use the same page design. The content part of this client has set the margins of the left and right parts to narrow the area that users need to pay attention to and bring a better experience.

Meanwhile, the video list part is dynamically created to expand the function of the video player. In other words, the playlist is generated by obtaining the number and name of the video through loop creating elements by JS. This means the playlist can grow dynamically as more videos are added.

Feature Design

In this section, the report would introduce the key features of this project and how to achieve them.

Video Player

a) Play All

The play all button realizes the function of creating an automatic sequence playlist for the video player, that is, the next one is automatically played after one video is played. Considering that what is being created is a playlist, so when the last song is played, it will not be played in a loop again. In addition, when the video is switched, the pictures and functions of the previous and the next part will also be switched. When the last video is played, click the next function, and the web page will alert: "This is the last one!".

b) Control Module

The control module implements some fundamental video control functions (such as mute, play, pause, and so on) through JS. Generally, this module realizes video control feature by modifying the attributes of the video object. The realization of Seekbar will introduce the realization process in detail.

c) Seekbar

The seekbar feature could display and adjust the video play progress in real-time. The video object can get the current time and total duration of the video being played, and the percentage of the two can get the position of the progress bar. Write this function as updateBar function and join the video object time update EventListener to realize the function of the seekbar reflecting the progress of the video playback. Based on the above feature, the click monitoring of the seekbar can obtain the percentage of the progress div. According to the formula: $\text{current time} = \text{playing percentage} * \text{total video duration}$, the function of modifying the current time to adjust the video progress is realized.

d) Playlist

As the previous selection mentioned, the playlist of this player is created dynamically.

Therefore, the video switching function is realized by adding a function modifying the video src every loop. The video playback function area is shown in Figure 3.

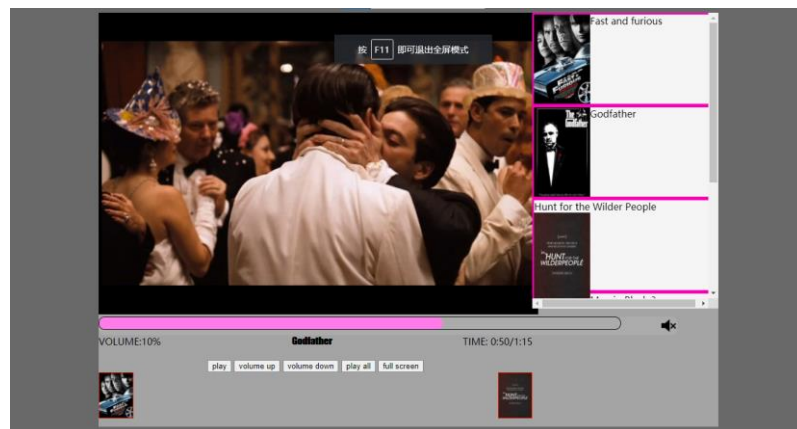


Figure 3.The player control module

Card Matching Game

The game interface of the client provides an interactive game of card matching. Users need click on two images to play this game. If the images opened on the two sites are the same, the state will be maintained. If they are not the same, the two images will return to the flip state, and the number of clicks and the number of successful matches will be recorded at the same time. When all the images are matched, the website would remind users that the game is won, and the game is shuffled to start the game again. The screenshots of the game are shown in Figure 4.

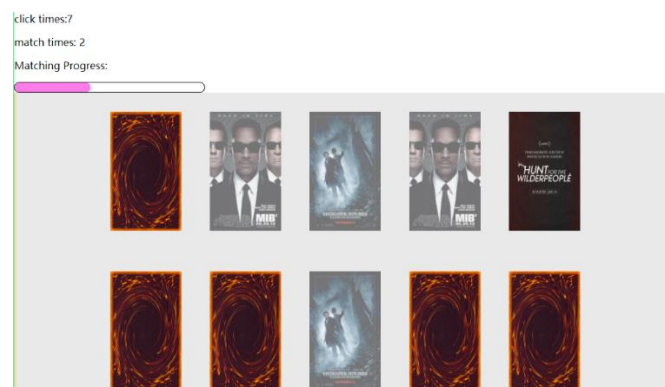


Figure 4. The screenshot of the game progress

At the same time, in order to improve the accessibility of the game, this project adds two functions: restart and reshuffle. The reshuffle function is the same as when the game is won,

and the order of the images is shuffled while restarting the game. But the restart function would only restart the game but not change randomly the order of the images, that is, the game state would be restored to the state before the click. The algorithm of this game is shown in Figure 5.

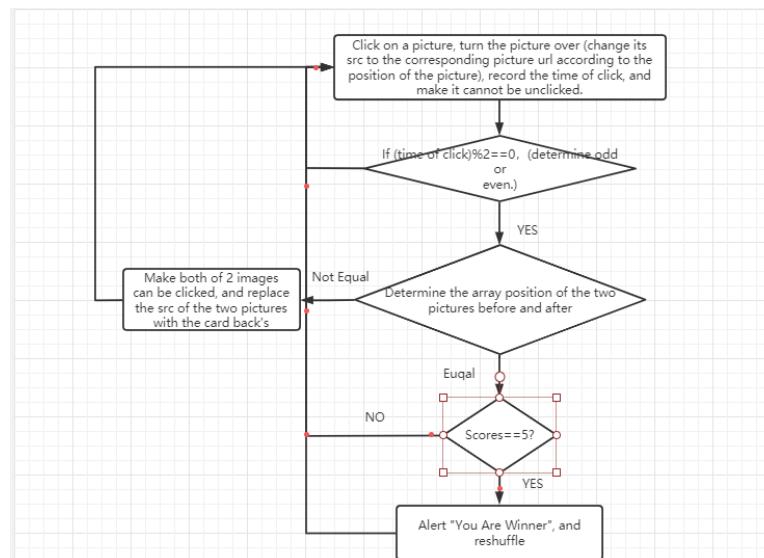


Figure 5. Flow chart of the card matching game

Reference

- [1] www.netflix.com
- [2] <https://www.youtube.com/watch?v=OkxsTkuLQP0>
- [3] <https://www.yugioh-card.com/>
- [4] Z. Hao, Z. Limiao and H. Hua, "A Web Design Mode for Browsers to CSS Compatibility Issues," 2012 Fourth International Conference on Multimedia Information Networking and Security, Nanjing, China, 2012, pp. 160-163, doi: 10.1109/MINES.2012.53.