Department of Computer Science
City University of Hong Kong
CS2204 Fundamentals of Internet Application Development
Course Work No. 3 (CW3)

Learning Outcomes:

- Able to write Javascript and execute them in Web pages;
- Able to apply Javascript to manipulate elements of Web pages, access data and process forms
- Able to use external Javascript to store data

1. Overview

You will use the Web pages you have prepared in CW2 and enhance their interactivities. Javascript will be added and you could also change the HTML/CSS to facilitate the functions of Javascript. Detail requirements are listed in the following sections.

2. External Javascript

An external Javascript http://courses.cs.cityu.edu.hk/cs2204/cw3/cinemas.js must be used in the *Now Showing* and *Ticket pages*. This script provides a function *getCinemas()* to return information about cinemas and show date time. You also need to define another external Javascript *movies.js* in which a function *getMovies()* can be called to return the movie information you have used in CW1 and CW2. Specifications of the functions *getMovies()* and *getCinemas()* are in the Appendix. During testing, you may use a local copy of *cinemas.js* and edit it to give more test data. The above link must be used in your final submission and different data may be used to test your pages.

3. Movies Page

- prepare your external Javascript *movies.js*
- using the function *getMovies()*, generate the HTML for the *Now Showing* and *Up Coming* lists of movies. The CSS styles of the page should be maintained as in CW2
- you cannot assume and hard-coded the number of movies in your script
- set up and play the video using information for the 1st movie in the *Now Showing* list
- at the end of video playing, wait for 2 seconds and switch to the next movie in the *Now Showing* list; continue with the *Up Coming* list or go back to the 1st movie if necessary
- when a movie thumbnail (image) is clicked, the current showing video will be stopped and the video of the clicked movie will be shown; at the end of it, continue showing the next one in the list after waiting 2 seconds

4. Now Showing Page

- using the function *getCinemas()*, generate the *Select Cinema* drop down box and corresponding movies and show date time available for the first cinema
- when another cinema is selected, movies and their shows will be updated
- the movie name and show index (see appendix I) will be sent to *Ticket* page when the Buy Ticket button is clicked
- layout and styles done in CW2 must be preserved

5. Ticket Page

- base on the movie name and show index sent from the *Now Showing* page, build the headings (or you can change them to input boxes) showing the cinema and movie names, the show date time and house number
- assuming all cinemas have the same seat plan (defined by you in CW2), set up event handlers for all the seats (tds) which when clicked will create a booked seat in the *Seat Booked* box
- a booked seat will have its background color changed and it cannot be booked again (i.e. no effect on further clicking)
- when the Confirm button is clicked relevant information will be sent to the *Print* page

6. Print page

- extract all information sent from the *Ticket* page and build all tickets for printing
- functionalities of the Print page in CW2 must be preserved

7. Assessment

You will be assessed by how much and how well you can apply what have been learnt from the course, some considerations are:-

- the requirements are met
- two different methods of setting up event handler should be used (i.e. both un-obtrusive and obtrusive), <u>state clearly in your updated design page where you used them</u>
- the Javascript should be clear, tight without redundant code and appropriate techniques are used;
- no Javascript libraries nor frameworks such as jQuery, YUI, Bootstrap or React etc. should be used;
- sensible use of external, embedded or inline scripts
- show a clear separation of structure, presentation and behavior
- use appropriate HTML tags, classes, ids or CSS to help your Javascript but at the same time do not over use id and class names
- your pages should work in more than one browser
- arrange your web site directories properly, e.g. HMTL, style sheets, images, and scripts etc.

8. Due Date and submission

- the due date has been announced in Canvas, the deadline time is 11:55 pm
- submit a zip file of your web site with appropriate folders set up so that it could be used directly by unzip; do not include videos

9. Miscellaneous information

- all techniques to achieve the requirements can be found in the lecture slides or examples in Canvas
- some common techniques include but not limited to: loops; use Array to store information; get hold of elements by id/tag/CSS selectors; loops to scan through a certain type of tag; change elements' properties (especially styles or innerHTML); use

document.write( ) method; setTimeout/setInterval; create HTML, show or hide elements dynamically and properly set up event handlers, etc.

- there may be more than one way to write Javascript for a problem
- it is important to first think of a way to solve the problem before writing your codes

~ End ~

Appendix

Sample *movies.js*. Should provide at least 3 movies in both the *Now Showing* and *Up Coming* lists.

```
function getMovies() {
  return [
    {
      id:1,
      type:"now",
      thumbnail:"casablanca.png",
      src:"casablanca.mp4",
      name:"Cassblanca",
      cast:"Humphrey Bogart, Ingrid Bergmen",
      director:"Michael Curtiz",
      duration: 130
    },
    {
      id:2,
      type:"now",
      thumbnail:"wildlife.png",
      src:"wildlife.mp4",
      name:"Big Buck",
      cast:"Big Bunny, Little Fox",
      director:"Forest King",
      duration: 100
    },
    {
      id:3,
      type:"upcoming",
      thumbnail:"nature.png",
      src:"nature.mp4",
      name:"Nature",
      cast:"Parrot, birds",
      director:"Mother Nature",
      duration: 100
    }
  ]
}
```

http://courses.cs.cityu.edu.hk/cs2204/cw3/cinemas.js

```
function getCinemas() {
  return [
{
   branchName:"City Cinema - Festival Walk",
   movies:[
      {   id:1,
         shows:[
         {
            index:1,
            datetime:"8 Nov, Sun - 9:00am",
            house:3
         },
         {
            index:2,
            datetime:"10 Nov, Tue - 11:00am",
            house:3
         }
         ]
      },
      {   id:2,
         shows:[
         {
            index:3,
            datetime:"9 Nov, Mon - 2:00pm",
            house:1
         }
         ]
      }
   ]
},
{
   branchName:"City Cinema - Mong Kok",
   movies:[
      {
         id:2,
         shows:[
            {
               index:4,
               datetime:"11 Nov, Wed - 9:00am",
               house:4
            },
            {
               index:5,
               datetime:"11 Nov, Wed - 11:00am",
               house:2
            }
         ]
      }
   ]
},
{
   branchName:"City Cinema - Central",
   movies:[
            {
               id:1,
               shows:[
                  {
                     index:6,
                     datetime:"11 Nov, Wed - 9:00am",
                     house:4
                  }
               ]
            }
         ]
      }
   ]
}
```

Sample Javascript to access the information in the cinemas and movies arrays (available in Canvas CW3 folder as *testdata.html*)

```
   <script src="cinemas.js"></script>
   <script src="movies.js"></script>
   <script>
      function getMovieName(movieArray, id) {
         for (let i=0; i<movieArray.length; i++) {
            if (movieArray[i].id == id) return movieArray[i].name;
         }
         return undefined;
      }
      //movies is an array of objects each containing info about a movie
      let movies=getMovies();
      //cinemas is an array of objects of cinema
      //  each cinema object contains branch name and an array of objects of movies now showing
      //  each movie object contains the movie id and an array of objects of shows
      //  each show object contains the show index, date time and house no.
      let cinemas=getCinemas();
      for (let i=0; i<cinemas.length; i++) {
         console.log(cinemas[i].branchName+"\n");
         for (let j=0; j<cinemas[i].movies.length; j++) {
            console.log("  movie id "+cinemas[i].movies[j].id+"\n");
            console.log("  movie name " + getMovieName(movies, cinemas[i].movies[j].id));
            for (let k=0; k<cinemas[i].movies[j].shows.length; k++) {
               console.log("    index "+cinemas[i].movies[j].shows[k].index+"\n");
               console.log("    datetime "+cinemas[i].movies[j].shows[k].datetime+"\n");
               console.log("    house "+cinemas[i].movies[j].shows[k].house+"\n\n");
            }
         }
      }
   </script>
```