COMP3322A Modern Technologies on World Wide Web

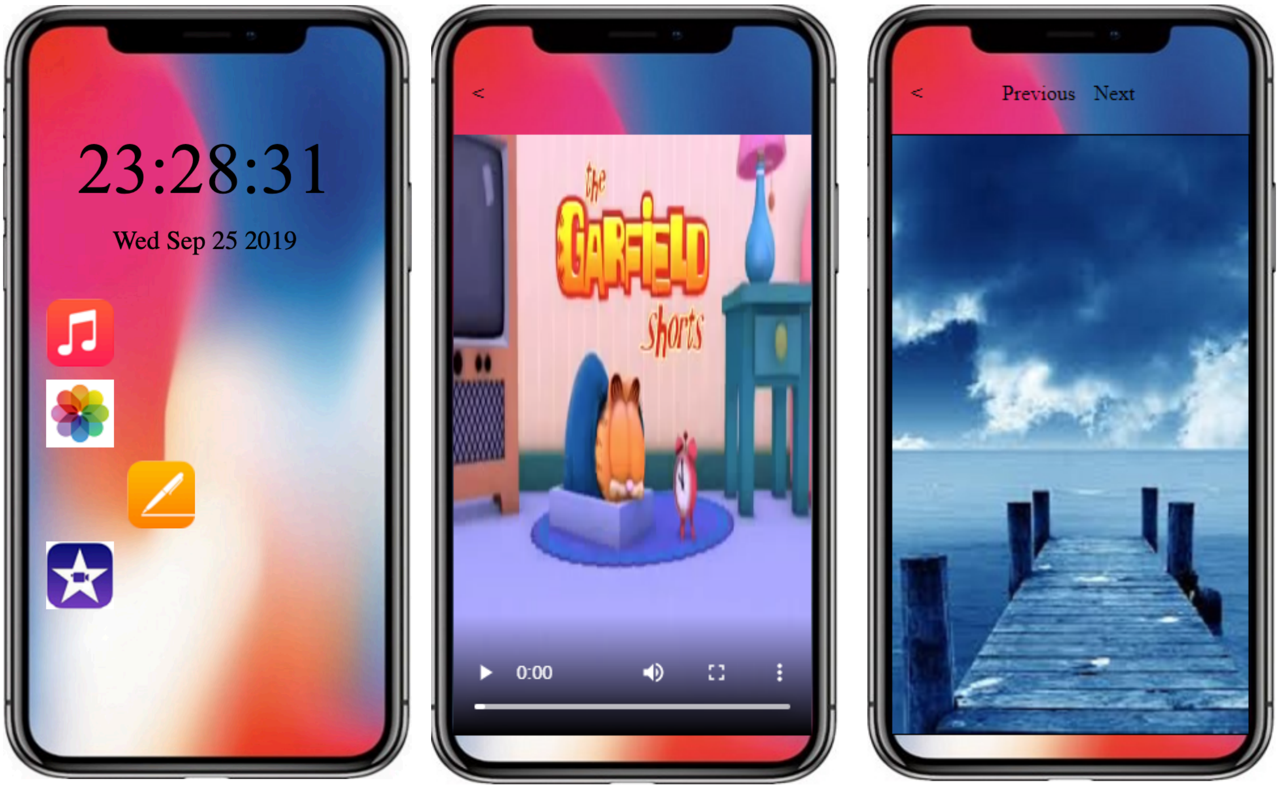
Lab3: HTML5

Overview

In this lab, we will implement a simple iPhone simulator page using a number of new elements and DOM APIs that HTML5 provides.

Fig. 1 shows what the completed page will look like:

- Upon page load, current time, day, and 4 icons are displayed on the simulator screen (Fig. 1(a)). The displayed time is updated every second. The icons can be dragged around and dropped to different locations in the center area on the screen.
- After clicking the video icon (the purple one with a star inside), the page will be as shown in Fig. 1(b): a video is displayed with control bar at the bottom; clicking "<" at the top leads back to the page view shown in Fig. 1(a).
- After clicking the photo icon (the flower shaped one), the page will be as shown in Fig. 1(c): a photo is displayed; clicking "Previous" or "Next" at the top leads to display of another different photo, respectively; clicking "<" at the top leads back to the page view shown in Fig. 1(a).



(a) home screen (b) after clicking video icon (c) after clicking photo icon

Fig. 1 Screenshots of the HTML5 page

Lab Exercise

Download "**lab3_materials.zip**" from HKU Moodle, and extract it to a folder named "**lab3**". There are 11 files provided in the folder: **index.html**, **style.css**, 8 images, and 1 mp4 video file. You are going to implement all your code in **index.html**. You do not need to change any other files provided.

Part 1: Implement the home screen and app icon drag-and-drop

[Step1 (Date and Time Display)]: In **index.html**, two elements whose ids are "time" and "day" have been provided; we also provide two functions drawTime() and drawDay() which are run upon page load. The drawDay() function has been implemented too. Please learn from drawDay() function's implementation to complete the drawTime() function. **Hint:** you may find JavaScript Date object's toTimeString method (https://www.w3schools.com/jsref/jsref_totimestring.asp) and String object's split method (https://www.w3schools.com/jsref/jsref_split.asp) useful.

We have also given the following line of code in **index.html**, which calls drawTime function once every second to update the displayed time (see https://www.w3schools.com/js/js_timing.asp):

```
setInterval(drawTime, 1000);
```

[Step2 (Drag-and-Drop of App Icons)]: In **index.html**, we provide a <div> element of id "iconblocks", inside which we have 16 child <div> elements, representing blocks in a 4X4 grid area in the middle of the iPhone home screen. We have four <img> elements in four grid blocks, displaying four app icons (pages.png, video.png, music.png, photos.png), respectively.

To allow drag of any app icon and drop it onto any grid block which currently does not have an app icon inside, please make each app icon <img> element draggable, and register "drag(event)" as the event handler for its ondragstart event; for each <div> element that corresponds to each of the 16 grid blocks, register allowDrop(event) as the handler for ondragover event, and drop(event) as the handler for ondrop event. Complete the event handler functions allowDrop(ev), drag(ev), and drop(ev).

In drop(ev), in order to achieve the functionality of dropping the dragged element only when the grid block is empty, use ev.currentTarget to identify the current <div> element, instead of ev.target as given in the example on page 35 of lecture slides 7_HTML5_COMP3322A_f2019.pdf (where the element to receive the drop is originally empty). (You can give it a try to investigate the difference between using ev.target and ev.currentTarget in drop(ev), and refer to https://developer.mozilla.org/en-US/docs/Web/API/Event/currentTarget.) You should tell if the current <div> element has a child or not (use children.length), and append the dragged <img> element there only if it has no child.

Part 2: Implement the video screen

[Step3 (Video Screen)]: In **Index.html**, we have registered a handler function openVideo() for the onclick event on the video app icon. In function openVideo(), we first make the time, day and icon blocks disappear with the following given code:

```
document.getElementById('time').style.display = "none";
document.getElementById('day').style.display = "none";
document.getElementById('iconblocks').style.display = "none";
```

Then we create a <nav> element corresponding to the navigation bar on the top of the screenshot in Fig. 1(b), with the following given code:

```
var navbar = document.createElement("nav");
navbar.setAttribute("id","navId");
var div1 = document.createElement("div");
div1.setAttribute("id","returnDiv");
div1.innerHTML = "<";
div1.setAttribute("onclick","return2HomeScreen()");
navbar.appendChild(div1);
var body =  document.getElementsByTagName("body")[0];
body.appendChild(navbar);
```

The navigation bar on this page view includes only a clickable <div> containing a "<", clicking which the page view will go back to the home screen as shown in Fig. 1(a) (we will implement the return2HomeScreen() function in Part IV).

Next, in function openVideo(), you should implement the JavaScript code that creates a <video> element, with id "videoId", attribute "controls", and a child <source> element pointing to Garfield.mp4 of type video/mp4. Append the <video> element into <body>, such that your page view will be as the one shown in Fig. 1(b).

Part 3: Implement the photos screen

[Step4 (Photos Screen)]: In **Index.html**, we have registered a handler function openPhotos() for the onclick event on the photos app icon. Implement the openPhotos() function as follows:

- Learning from the code in openVideo(), make the time, day and icon blocks disappear;
- Learning from the code in openVideo(), create a <nav> element containing not only a "<" division, but also a <div> of content "Previous" and a <div> of content "Next", respectively, such that you will implement the top navigation bar as shown in Fig. 1(c). Register previousPhoto() as the handler for onclick event of the

"Previous" element and nextPhoto() as the handler for onclick event of the "Next" element.

- Then, create a <canvas> element, with id "photo_canvas", width "272" and height "457". Draw the photo corresponding to photo_list[global_photo_index] on the canvas, occupying the entire canvas. Append the <canvas> element into <body>, such that your page view will be as the one shown in Fig. 1(c).

[Step5 (Previous Photo)]: Implement function "previousPhoto()", which reduces the global variable global_photo_index by 1 if it is not zero (otherwise, retain its value of 0), and then draw the photo corresponding to photo_list[global_photo_index] on the canvas.

[Step6 (Next Photo)]: Implement function "nextPhoto()", which increments the global variable global_photo_index by 1 if it is not the last index in array photo_list[] (otherwise, retain its current value), and then draw the photo corresponding to photo_list[global_photo_index] on the canvas.

Part 4: Return to the home screen

[Step7 (Return to Home Screen)]: "<" on the upper left corner of the video screen in Fig. 1(b) and the photos screen in Fig. 1(c) should have been made clickable with the event handler "return2Homescreen()". In return2Homescreen() function, we have provided the following code to check if the <nav> element exists and if so, remove the element (note that this is different from just making the element invisible or not displayed on the screen):

```
var ele = document.getElementById('navId');
if(ele){
	ele.parentElement.removeChild(ele);
}
```

Learn from the above to implement the code to check if the <video> element and the <canvas> element exist, and if so, remove them respectively. Then, make the time, day and app icons reappear on the screen. **Hint:** you can change the display style of those elements to "block".

Submission:

Please finish this lab exercise before 23:59 Wednesday Oct. 2, 2019.

Please compress all the files provided in a .zip file and submit it on Moodle:
(1) Login Moodle.
(2) Find "Labs" in the left column and click "Lab 3".
(3) Click "Add submission", browse your .zip file and save it. Done.
(4) You will receive an automatic confirmation email, if the submission was successful.
(5) You can "Edit submission" to your already submitted file, but ONLY before the deadline.