Πανεπιστήμιο Κρήτης
HY359 – Διαδικτυακός Προγραμματισμός (Web Programming)
Χειμερινό Εξάμηνο 2016
Διδάσκων: Παναγιώτης Παπαδάκος
Υπεύθυνος Βοηθός: Κωνσταντίνος Σγόντζος

2η Σειρά Ασκήσεων

Διάρκεια: 18/10 – 31/10
Αξία: 8% του τελικού σας βαθμού
Θεματική ενότητα : **JavaScript - APIs**

Άσκηση 1. Tiled Image Viewer (TIV) [60%]

Αποτελεί μέρος του συνόλου των ασκήσεων που θα συννενοθούν στο τέλος.

Σκοπός αυτής της εργασίας είναι να εμπλουτίσετε/βελτιώσετε τον TIV client που φτιάξατε στην άσκηση 1.

Α (30%) - Υλοποίηση TIV API

Συγκεκριμένα θα πρέπει να δημιουργήσετε ένα JavaScript αρχείο tiv.js σε ένα directory js στο οποίο θα πρέπει να υλοποιήσετε ότι σας ζητείται στα παρακάτω ερωτήματα. Το αρχείο αυτό θέλουμε να μας παρέχει ένα JavaScript API που θα μας δίνει ότι λειτουργικότητα χρειαζόμαστε για τη λειτουργία του TIV (δείτε τις διαφάνειες περί JavaScript Functions πως μπορούμε να το καταφέρουμε αυτό).

Θέλουμε να φτιάξουμε μία function TIVAM (όπου AM αντικαθιστάτε με το AM σας) η οποία και θα προσφέρει το API του TIV. Συγκεκριμένα θα πρέπει να προσφέρει τη παρακάτω λειτουργικότητα:

- **loadImages() -** κρατάει εσωτερικά στο TIVAM τα files όλων των εικόνων που υπάρχουν στο συγκεκριμένο directory που έχει επιλεχθεί στο input με id='imagesCollection' (δείτε παρακάτω), π.χ σε ένα πίνακα loadedImages. Ενημερώνει δηλαδή το API με τα αντικείμενα των φορτωμένων εικόνων. Αν το imagesCollection δεν είναι ορισμένο ο loadedImages είναι κενός.

- **getLoadedImages() -** επιστρέφει τον πίνακα loadedImages

- **showLoadedImages(elem)** - ζωγραφίζει μέσα στο elem που περνάτε σαν argument όλες τις εικόνες που έχουν γίνει load, χρησιμοποιώντας τα tiles που φτιάξατε από τη προηγούμενη άσκηση. Αν δεν έχει γίνει καμία εικόνα load τότε ζωγραφίζει με μεγάλα γράμματα οτι δεν έχει γίνει load κανένα collection.

Πλέον μπορούμε να εμφανίζουμε όλες τις εικόνες που περιλαμβάνονται μέσα σε κάποιο directory. Οπότε τοποθετήστε ένα κουμπί στην κεντρική οθόνη του TIV ώστε μέσω αυτού να μπορείτε να φορτώσετε αλλά και να ζωγραφίσετε όλες τις εικόνες που περιέχει το directory που σας δίνει σαν είσοδο ο χρήστης (υπόδειξη: δείτε τον κώδικα που δίνεται παρακάτω - χρήση <input id="images" type="file" webkitdirectory mozdirectory directory/>).

```
<!-- Βοηθητικός κώδικας για το πως να διαβάζετε ένα directory και να βλέπετε ποια αρχεία έχει!
     Χρησιμοποιεί HTML5 και το API για το filesystem! --->
<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
<title>Read files from directory</title>
<script>
function readAndShowFiles() {
  var files = document.getElementById("images").files;
  for (var i = 0; i < files.length; i++) {
    var file = files[i];
    // Have to check that this is an image though
    // using  the file.name TODO
    var reader = new FileReader();
    // Closure to capture the file information.
```

```
    reader.onload = (function(file) {
      return function(e) {
        // Render thumbnail.
        var span = document.createElement('span');
        span.innerHTML = ['<img  src="', e.target.result,
          '" title="', escape(file.name), '">'].join('');
         document.getElementById('list').insertBefore(span, null);
      };
    })(file);
    // Read in the image file as a data URL.
    reader.readAsDataURL(file);
  }
}
</script>
</head>

<body>
<form name="uploadForm">
<p>
<input id="images" type="file" webkitdirectory mozdirectory directory name="myFiles"
onchange="readAndShowFiles();" multiple/>
<span id="list"></span>
</form>
</body>
</html>
```

B (30%) Extend TIV API - Image Detailed Info – EXIF Data – Maps

Σε αυτό το υποερώτημα της άσκησης θέλουμε να μπορούμε πατώντας πάνω σε μια εικόνα να ανανεώνεται η οθόνη, έτσι ώστε να μας δείχνει την ίδια εικόνα σε μεγαλύτερη ανάλυση μαζί με το URL της. Επιπλέον, επειδή οι φωτογραφίες πολύ συχνά περιέχουν extra μεταδεδομένα, π.χ. πότε τραβήχτηκαν, με τη μηχανή, χαρακτηριστικά της λήψης καθώς και γεωγραφικά δεδομένα, θα θέλαμε δεξιά της εικόνας να δείξουμε τις exif πληροφορίες που έχει διαθέσιμες η συγκεκριμένη εικόνα και αν υπάρχει γεωγραφική πληροφορία ο χάρτης εμφάνισής της.

- **showImage(index, elem)** εμφανίζει την εικόνα με index index στον loadedImages στο elem που περνάμε σαν όρισμα (π.χ. τον container όλων των tiles)

- **showImageDetailedExifInfo(index, elem)** γράφει μέσα στο elem που του περνάμε σαν παράμετρο όλες τις EXIF πληροφορίες που έχει η εικόνα με index index στον loadedImages (μπορείτε να χρησιμοποιήσετε τη βιβλιοθήκη https://github.com/exif-js/exif-js)

- **showImageDetailedExifWithMap(index, elem)** αντίστοιχα με την **showImageDetailedExifInfo** μόνο που επιπλέον τοποθεσία που τραβήχτηκε πάνω στον χάρτη η εικόνα με χρήση κάποιου marker (https://developers.google.com/maps/documentation/javascript/), αν υπάρχουν γεωγραφικά δεδομένα.

Σημειώσεις:

- Προτείνεται η χρήση του browser chrome για αυτή την άσκηση, αφού οι περισσότεροι browsers (συμπεριλαμβανομένου του firefox) δεν υποστηρίζουν το filesystem API. http://caniuse.com/#feat=filesystem
- Ο validator της HTML μπορεί να πετάξει λάθη για τα attributes του input tag που χρησιμοποιείτε, τα οποιά όμως μπορείτε να αγνοήσετε (δεν θα επηρεάσουν τη βαθμολογία σας). Δυστυχώς η υποστήριξη των εργαλείων για HTML5 δεν είναι ακόμα ώριμη.

Άσκηση 2. Multi Filter Function [25%]

Στον παρακάτω κώδικα δηλώνουμε μία global function με το όνομα **makeMultiFilter** η οποία δέχεται σαν παράμετρο ένα array **originalArray** και επιστρέφει μία function (ας την ονομάσουμε **arrayFilterer**) η οποία μπορεί να χρησιμοποιηθεί για να κάνουμε filter τα elements αυτού του array. Η **arrayFilterer** εσωτερικά κάνει track ένα array που ονομαζεται **currentArray** και το οποίο αρχικά ορίζεται να είναι ίδιο με **originalArray**. Η arrayFilterer παίρνει 2 παραμέτρους:

- **pred** – Μία function η οποία παίρνει ένα στοιχείο του **currentArray** και επιστρέφει boolean. Αυτή η function καλείται για κάθε στοιχείο του **currentArray** και το **currentArray** γίνεται update βάσει του αποτελέσματος της filter function. Αν η pred δεν είναι function τότε η επιστρεφόμενη τιμή της arrayFilterer θα είναι η τιμή του currentArray χωρίς να έχει γίνει κάποιο filtering.

- **callback** – Μία function που θα κληθεί όταν έχει γίνει το filtering πάνω στα στοιχεία του currentArray, το οποίο παίρνει σαν όρισμα. Το **this** μέσα στη callback θα πρέπει να κάνει reference στο **originalArray**. Αν η callback δεν είναι κάποια function, αγνοείται. Η return τιμή της callback δεν χρησιμοποιείται.

Η arrayFilterer επιστρέφει τον εαυτό της εκτός και αν το filter argument δεν είναι οριστεί οπότε και επιστρέφει το **currentArray**. Θα πρέπει να είναι δυνατό να έχουμε πολλαπλές arrayFilterer functions την ίδια στιγμή.

Συμπληρώστε τις γραμμές που λείπουν στις γραμμές 1-16 με τον κατάλληλο κώδικα ωστε ο κώδικας που ακολουθεί να τρέχει σωστά. Δίνονται βοηθητικά σχόλια. Για κάθε γραμμή που συμπληρώνετε θα γίνεται και ένα commit όπου θα εξηγείτε το λόγο που τη συμπληρώσατε με αυτό τον τρόπο. Προσοχή, θα χρειαστεί να γίνει ένα κοινό commit για τις γραμμές 7,8 και άλλο ένα για τις γραμμές 12,13.

Σημειώσεις:

Προτείνεται η χρήση του online REPL για JavaScript repl.it (https://repl.it)

```
1 function makeMultiFilter(array) {
2       // What we track
3        // TO COMPLETE!
4        // TO COMPLETE!
5       return (function arrayFilterer(pred, callback) {
6               // If filter not a function return current Array
7               // TO COMPLETE TOGETHER!
8               // TO COMPLETE TOGETHER!
9               // Filter out things
10              // TO COMPLETE!
11              // If callback is a function, execute callback
12              // TO COMPLETE TOGETHER!
13              // TO COMPLETE TOGETHER!
14              // TO COMPLETE! We have to return something!
15      })
16 }
17 // !!! THE CODE THAT FOLLOWS USES THE makeMultiFilter function
18 // arrayFilterer can be used to repeatedly filter this input array
19 var arrayFilterer = makeMultiFilter([1,2,3]);
20
21 // call arrayFilterer to filter out all the numbers not equal to 2
22 arrayFilterer(function (elem) {
23  return elem !== 2; // check if element is not equal to 2
24 }, function (currentArray) {
25  console.log(this); // prints [1,2 3]
26   console.log(currentArray); // prints [1, 3]
27 });
28
29 // call arrayFilterer filter out all the elements not equal to 3. No callback used
30 arrayFilterer(function (elem) {
31  return elem !== 3; // check if element is not equal to 3
32 });
33
34 // call arrayFilterer with no filter should return the current array
35 var currentArray = arrayFilterer();
36 console.log('currentArray', currentArray);   // prints [1] since we filtered out 2 and 3
37
38 // Since arrayFilterer returns itself filters calls can be chained like:
39 function filterTwos(elem) { return elem !== 2; }
40 function filterThrees(elem) { return elem !== 3; }
41 var currentArray = makeMultiFilter([1,2,3])(filterTwos)(filterThrees)();
42 console.log('currentArray', currentArray);   // prints [1] since we filtered out 2 and 3
43
44 // Multiple active filters at the same time
45 var arrayFilterer1 = makeMultiFilter([1,2,3]);
46 var arrayFilterer2 = makeMultiFilter([4,5,6]);
47 console.log(arrayFilterer1(filterTwos)());       // prints [1,3]
48 console.log(arrayFilterer2(filterThrees)());     // prints [4,5,6]
49 // Προσοχη αν τρέξετε console.log(array) πρέπει να πάρετε undefined
50 // Το ίδιο αν τρέξετε console.log(originalArray) ή console.log(currentArray)
```

Το υπόλοιπο 15% του βαθμού θα κατανεμηθεί βάσει των παρακάτω 3 κριτηρίων:

- **jslint, code quality – 5%:** θα κρίνεται από το αν η σελίδα σας δεν εμφανίζει λάθη/warnings στον jslint, καθώς και στη γενική ποιότητα του κώδικά σας
- **ελκυστικότητα εμφάνισης σελίδων (στυλιστική συνέπεια) – 5%**
- **git – 5%:** θα κρίνεται από τη σωστή χρήση του git (π.χ. να υπάρχουν αρκετά commits που να περιγράφουν με σαφήνεια πως κάνατε την άσκηση, με κατανοητή περιγραφή, καθαρό ιστορικό, κτλ.)

Σημειώσεις:

Μη ξεχνάτε τη χρήση του *"use strict";* για την *JavaScript*
Μπορείτε να χρησιμοποιήσετε το *jslint* http://www.jslint.com/ για να βελτιώσετε την ποιότητα του *js* κωδικά σας.

Τρόπος Παράδοσης

Οι ασκήσεις θα παραδίδονται μόνο μέσω git, σύμφωνα με τις οδηγίες που σας έχουν δοθεί. Συγκεκριμένα στο repository σας στο bitbucket το οποίο θα πρέπει να έχει γίνει ήδη share στο hy359, στο folder a2 θα πρέπει να υπάρχουν δύο subfolders, ***tiv και multiFilter***, όπου θα περιέχεται ο κώδικας για κάθε άσκηση. **Θα πρέπει να φροντίσετε ότι όλα όσα έχετε κάνει έχουν γίνει σωστά commit και βρίσκοντα online στο bitbucket.**

Προγραμματίστε καλά το χρόνο σας και αποφύγετε να ασχοληθείτε με την εργασία τελευταία στιγμή.

Στις 00:00 της 1/11 θα γίνει αυτόματο pull από όλα τα repositories που έχουν γίνει share στο hy359 και βάσει αυτών θα βαθμολογηθείτε. Εκπρόθεσμες ασκήσεις **δεν θα γίνονται δεκτές** (μόνο σε ειδικές περιπτώσεις σε συμφωνία με τον διδάσκοντα).

Αντιγραφή

Σε περίπτωση αντιγραφής θα μηδενίζονται άμεσα οι εργασίες όλων των εμπλεκόμενων.

Καλή εργασία