

Πανεπιστήμιο Κρήτης

ΗΥ359 – Διαδικτυακός Προγραμματισμός (Web Programming)

Χειμερινό Εξάμηνο 2024/2025

Διδάσκων: **Μιχαήλ Μουνταντωνάκης**

Υπεύθυνος Βοηθός: **Γ. Κοκιάδης** (Συντονιστής) και οι υπόλοιποι βοηθοί

## 2η Σειρά Ασκήσεων (Ατομική)

Διάρκεια: Δευτέρα 21/10 – Κυριακή 10/11

Αξία: 10% του τελικού σας βαθμού

(Συνολο ασκήσεων: 110 μονάδες)

Θεματική ενότητα: Εξοικείωση με JavaScript – APIs - Ajax - Maps

### Άσκηση 1. Βελτίωση Φόρμας (25%)

*Αποτελεί μέρος του συνόλου του project*

Εμπλουτισμός/βελτίωση της φόρμας εγγραφής που φτιάξατε στην άσκηση 1.

#### a) (4 μονάδες) Password Mismatch - DOM manipulation

- Να ελεγχθεί ότι τα 2 passwords που έχουν εισαχθεί από τον χρήστη είναι τα ίδια.
- Σε περίπτωση λάθους θα πρέπει να τυπωθεί κατάλληλο μήνυμα δίπλα από τα πεδία (όχι με χρήση alert).
- Επίσης θα πρέπει ο χρήστης να μπορεί πατώντας ένα κουμπί να βλέπει το password του (και πατώντας το ξανά να το κρύβει).

#### b) (9 μονάδες) Ασφάλεια Password

Θα πρέπει να ελεγχθεί με απλούς κανόνες ότι το password είναι ασφαλές. Συγκεκριμένα οι κανόνες πρέπει να είναι οι εξής:

- Να απαγορεύονται στο password οι ακολουθίες “fire”, “fotia”, “ethelontis”, “volunteer”, είτε δίνονται με μικρά είτε με κεφαλαία, είτε με συνδυασμό (μέσω javascript) **(2 μονάδες)**
- Αν τουλάχιστον το 50% του password και πάνω είναι αριθμοί να του εμφανίζει μήνυμα δίπλα από τα πεδία (όχι με χρήση alert) ότι είναι weak password.
  - πχ 12345aB\$ (5 αριθμοί στα 8 γράμματα) **(2 μονάδες)**
- Αν τουλάχιστον το 50% του password και πάνω είναι ο ίδιος χαρακτήρας να του εμφανίζει μήνυμα δίπλα από τα πεδία (όχι με χρήση alert) ότι είναι weak password.
  - πχ a1a\$αχα (4 χαρακτήρες α στους 8) **(2 μονάδες)**
- Αν το password περιέχει τουλάχιστον 1 σύμβολο, 1 κεφαλαίο γράμμα, 1 αριθμό και 1 ένα μικρό γράμμα να του εμφανίζει strong password. **(2 μονάδες)**

- πχ. wadP0@21 (τουλάχιστον 1 μικρό, τουλάχιστον 1 μεγάλο, και σύμβολα @ και αριθμό 3 ).
- Σε όλες τις άλλες περιπτώσεις να του εμφανίζει medium password (**1 μονάδα**)
  - πχ. aacdet3@ (δεν είναι ούτε weak ούτε strong).
- **Τέλος να μην επιτρέπει το submit αν το password είναι weak**
  
- Μπορείτε να έχετε ένα κουμπί δίπλα στο password που να γράφει see password strength. Μόλις το πατάτε, μπορεί να γράφει σε ένα μήνυμα δίπλα στο password (όχι με χρήση alert) αν το password είναι weak, medium ή strong. Εναλλακτικά μπορεί να γίνει και με τη χρήση του event onchange.
- Hint: Για να διαβάσετε όλους τους χαρακτήρες μιας λέξης:
 

```
var text="abcdef";
for(var i=0; i<text.length;i++){
  console.log(text.charAt(i));
}
```

c) (4 μονάδες) **Έξτρα φόρμα για Εθελοντή Πυροσβέστη:** Αν ο χρήστης είναι Εθελοντής Πυροσβέστης

- Να εμφανίζονται τα εξής έξτρα πεδία
  - **Κατηγορία (1 μονάδα)** (volunteer\_type)
    - Μάχιμοι Εθελοντές πυροσβέστες (simple)
    - Οδηγοί-Χειριστές Πυροσβεστικών οχημάτων (driver)
  - **Ύψος (1 μονάδα)** (height)
    - Από 1.45-2.20
  - **Βάρος (1 μονάδα)** (weight)
    - Από 40-160 kg
  - **Να αλλάζει το μήνυμα στους όρους χρήσης, συγκεκριμένα να επεκτείνετε ως εξής (1 μονάδα):** “Δηλώνω υπεύθυνα ότι ανήκω στο ενεργό δυναμικό των εθελοντών πυροσβεστών.”

d) Έξτρα έλεγχοι (6 μονάδες)

- Αν αλλάζει ξανά την επιλογή σε Απλό Χρήστη, να εξαφανίζονται οι επιλογές και να αλλάζει το μήνυμα στους όρους (**3 μονάδες**)
- Αν η ηλικία του εθελοντή πυροσβέστη είναι μικρότερη του 18 ή μεγαλύτερη του 55, τότε να μην τον αφήνει να κάνει register (**3 μονάδες**)

e) Τελικό αποτέλεσμα (2 μονάδες)

- Να τυπώνει την φόρμα σε json μέσα στη σελίδα στο τέλος όταν πατηθεί το submit και έχουν περάσει σωστά οι έλεγχοι, πχ κάτω από τη φόρμα σε κάποιο div, έτσι ώστε να είναι έτοιμο για την επόμενη σειρά ασκήσεων.

## Άσκηση 2 Χρήση Χαρτών Open Street Maps [20%]

Αποτελεί μέρος του συνόλου του project

Εμπλουτισμός/βελτίωση της φόρμας εγγραφής που φτιάξατε στην άσκηση 1 με χάρτες.  
**SOS Δείτε το σχετικό tutorial και τον κώδικα για το RAPID API και τα OSM maps.**

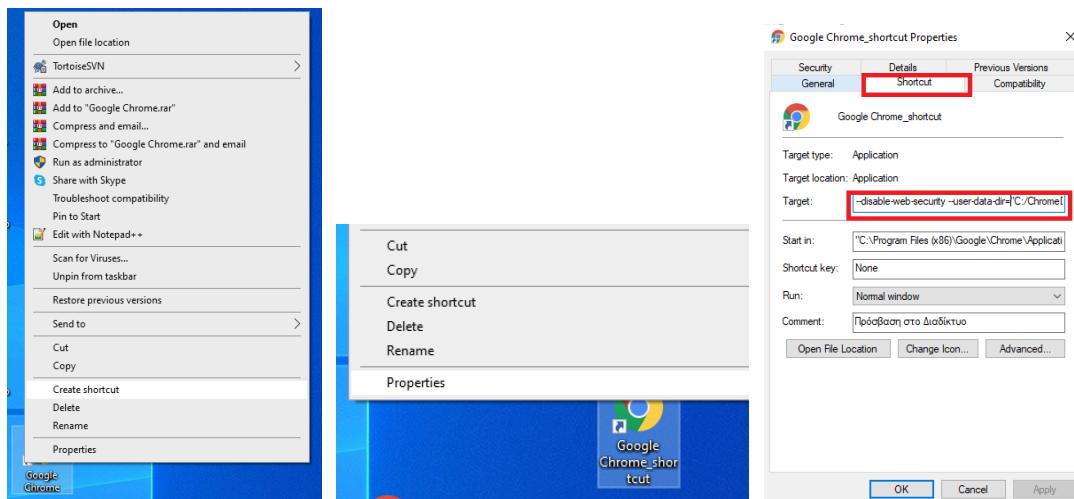
### ΠΡΟΣΟΧΗ!!!

Σε περίπτωση που ο browser παραπονιέται για προβλήματα με τον CORS (Cross-Origin Resource Sharing) μηχανισμό μπορείτε να κάνετε τα εξής:

- Να ανοίξετε την εφαρμογή του browser που χρησιμοποιείτε με απενεργοποιημένα το security μηχανισμό (π.χ. για τον chrome --disable-web-security)
- Θα πρέπει να βάλετε τη σελίδα σας σε κάποιο web server που θα τρέχει στο μηχάνημά σας (π.χ. Apache).
- Να εγκαταστήσετε κάποιο plugin το οποίο κάνει override τον CORS μηχανισμό (δεν συστήνεται)

### Οδηγίες για απενεργοποίηση CORS

- Δημιουργήστε ένα shortcut του chrome στην επιφάνεια εργασίας
- Δεξί κλικ στο shortcut και κλικ στα Properties.
- Πηγαίντε στο tag Shortcut και επεξεργαστείτε το Target property.
- Βάλτε του την τιμή **--disable-web-security --user-data-dir="C:/ChromeDevSession"** μετά το τιμή που έχει πχ "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"



- Ανοίξτε το chrome από αυτό το shortcut

## a) (10 μονάδες) Χρήση OpenStreetMaps (OSM) API - Επαλήθευση διεύθυνσης

Συνενώνοντας τις τιμές των πεδίων “Χώρα”, “Δήμος” και “Διεύθυνση” από τη φόρμα, θα πρέπει να επαληθεύσετε μέσω του **nominatim OSM geocode service** αν η συγκεκριμένη περιοχή υπάρχει ή όχι (πχ μέσω ενός κουμπιού).

<https://rapidapi.com/GeocodeSupport/api/forward-reverse-geocoding>

- Θα πρέπει να κάνετε το κατάλληλο request σε αυτό το service και να πάρετε την απάντησή του σε JSON μορφή (δείτε το παράδειγμα στο μάθημα). Θεωρήστε ότι η τοποθεσία υπάρχει εφόσον το API επιστρέφει μία μη κενή λίστα αποτελεσμάτων στην JSON απάντηση που επιστρέφει και αν δίνει πάνω από μία τοποθεσίες επιλέξτε την πρώτη σαν πιο σχετική.
- Σε περίπτωση που η απάντηση του service είναι κενή (άρα δηλαδή δεν γνωρίζει το OSM τη συγκεκριμένη τοποθεσία), θα πρέπει να τυπώνεται κατάλληλο μήνυμα λάθους στη σελίδα, σε κάποιο DOM element (όχι χρήση alert).
- Όλες οι απαντήσεις από τα παραπάνω services είναι σε JSON το οποίο μπορείτε να κάνετε parse ώστε να δημιουργήσετε ένα JavaScript αντικείμενο από την πλευρά της Javascript μέσω της εντολής `JSON.parse(json_str)`.
- **Να κρατήσετε τα lat και lon, γιατί θα τα χρειαστείτε στο επόμενο βήμα, αλλά και για να τα αποθηκεύεται στη βάση.**
- Αν η περιοχή δε βρίσκεται στην Κρήτη, να εμφανίζεται μήνυμα ότι η υπηρεσία είναι διαθέσιμη μόνο στην Κρήτη αυτή τη στιγμή.
  - Hint: Τσεκάρετε το `display_name` που επιστρέφετε από το json αν έχει μέσα την Κρήτη.

## b) (10%) Χρήση OSM maps - Εμφάνιση χάρτη

Δεδομένης μιας σωστής τοποθεσίας (Χώρα, Δήμος και Διεύθυνση) θα πρέπει να δίνεται η επιλογή στο χρήστη μέσω κάποιου κουμπιού,

- να βλέπει σε χάρτη την τοποθεσία την οποία δήλωσε με ένα marker.
- Σε περίπτωση που ο χρήστης αλλάξει την διεύθυνση, θα πρέπει να χαθούν από το DOM τα elements που σχετίζονται με τον container (π.χ. `div`) του χάρτη και να γίνει ξανά η επαλήθευση της τοποθεσίας για την εμφάνιση του κουμπιού.
- Σε περίπτωση που το geocode έχει επιστρέψει περισσότερες από μία σωστές τοποθεσίες, μπορείτε να χρησιμοποιήσετε την πρώτη. Κεντράρετε το χάρτη στη συγκεκριμένη θέση. Να λάβετε υπόψη ότι το OSM από default χρησιμοποιεί Spherical Mercator Coordinates αντί των κοινά χρησιμοποιούμενων WGS84 GPS συντεταγμένων. Οπότε θα πρέπει να γίνει μια μετατροπή από EPSG:4326 (WGS84) σε EPSG:900913 (Spherical Mercator) μέσω της `transform` που δίνεται από τη `OpenLayers.LonLat` (δείτε <http://dev.openlayers.org/docs/files/OpenLayers/BaseTypes/LonLat-js.html>).

- **Δείτε φυσικά και το έτοιμο παράδειγμα που είδαμε στο μάθημα.**

### Χρήση OSM maps - geocoding services

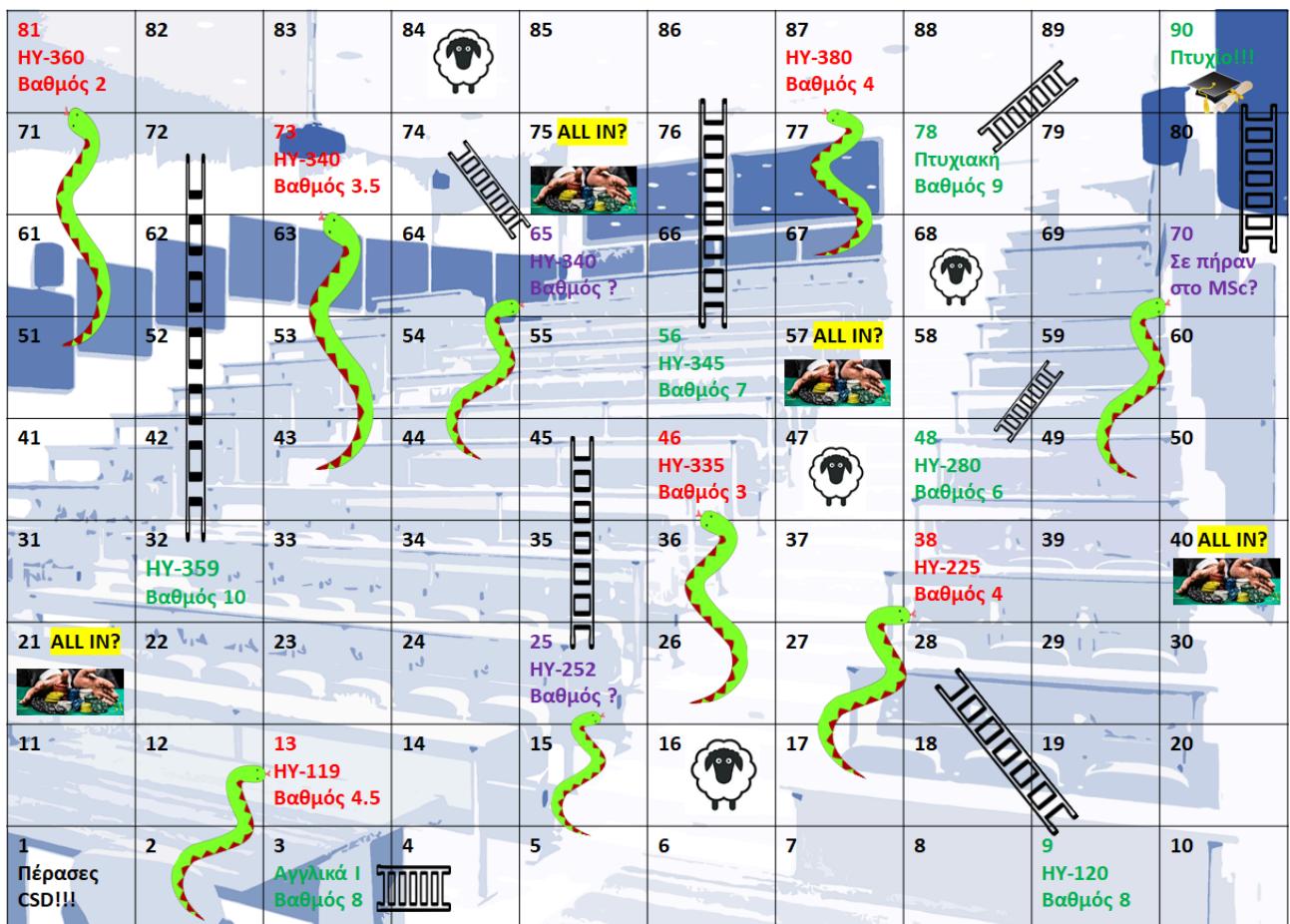
Καλείστε να χρησιμοποιήσετε το API των OSM maps όσο και το geocoding service. Δείτε κυρίως τα παραδείγματα στο elearn και τους παρακάτω βοηθητικούς συνδέσμους:

- [https://www.w3schools.com/html/html5\\_geolocation.asp](https://www.w3schools.com/html/html5_geolocation.asp)
- <https://wiki.openstreetmap.org/wiki/Nominatim>
- <https://rapidapi.com/GeocodeSupport/api/forward-reverse-geocoding>
- [https://wiki.openstreetmap.org/wiki/OpenLayers\\_Marker\\_Example](https://wiki.openstreetmap.org/wiki/OpenLayers_Marker_Example)
- [https://wiki.openstreetmap.org/wiki/Openlayers\\_POI\\_layer\\_example](https://wiki.openstreetmap.org/wiki/Openlayers_POI_layer_example)

**Σημείωση:** Σημαντικό είναι να καταλάβετε τι ακριβώς κάνει ο κώδικας και να αποφύγετε απλά το έτοιμο copy-paste (θυμηθείτε τα αντικείμενα που είδαμε στο μάθημα).

- Προσπαθήστε οι συναρτήσεις που θα φτιάξετε να είναι γενικές έτσι ώστε π.χ. αν στο μέλλον χρειαστεί να ζωγραφίσετε το map σε κάποιο άλλο div να μπορείτε να το κάνετε εύκολα δίνοντας το κατάλληλο όρισμα στην αντίστοιχη συνάρτηση.
- Αντίστοιχα αν χρειαστεί να εισάγετε νέους markers να μπορεί να γίνει μέσω της κατάλληλης κλήσης μιας συνάρτησης, η οποία θα μπορεί να δέχεται το map πάνω στο οποίο θέλουμε να ζωγραφίσουμε τους markers και το αντίστοιχο location.
- Για τα requests που θα χρειαστεί να κάνετε στα OSM services θα χρησιμοποιήσετε ajax requests (δείτε και τον κώδικα που έχει διθεί στο μάθημα και το σχετικές διαφάνειες). Επίσης αφιερώστε λίγο χρόνο για τη σχεδίαση της σελίδας. Σκεφτείτε ότι ό,τι χτίζεται τώρα θα χρησιμοποιηθεί και στη συνέχεια της εργασία σας στο μάθημα.

## Άσκηση 3. Υλοποίηση Παιχνιδιού Φιδάκι και Σκάλες (CSD Edition 2024-25) [55%+10% bonus]



**Φτιάξτε μέσω HTML και Javascript το παιχνίδι Φιδάκι και Σκάλες.**

Για να εξοικειωθείτε με τη Javascript, θα φτιάξετε το κλασικό παιχνίδι **Φιδάκι και Σκάλες (CSD Edition 2024)**, χρησιμοποιώντας μόνο **HTML, CSS και Javascript**. Στο παιχνίδι αυτό υπάρχουν 90 τετραγωνάκια, και ένα ταμπλό 9x10 (9 γραμμές και 10 στήλες). Οταν ο παίκτης περνάει ένα μάθημα στο csd, τότε ανεβαίνει με τη σκάλα στο αντίστοιχο τετράγωνο. Όταν κόβεται στο μάθημα, συναντάει ένα φίδι και γυρνάει στη θέση που βρίσκεται η ουρά του φιδιού. Όμως το παιχνίδι έχει και κάποιες έξτρα λειτουργίες.

Κάθε παίκτης έχει στην διάθεσή του 1 πιόνι, το οποίο μετακινείται μόνο με ένα και μοναδικό ζάρι. Ο σκοπός του παιχνιδιού είναι να φτάσει κάποιος στον τερματισμό. Υποθέτουμε ότι οι 2 παίκτες μοιράζονται την ίδια οθόνη και το ίδιο ποντίκι.

## Βοηθητικό Υλικό για να ξεκινήσετε

### Σας δίνεται κώδικας (και σχετικό presentation) ο οποίος

- αρχικοποιεί και δείχνει το γραφικό του ταμπλό
- αρχικοποιεί τις θέσεις σε ένα object, για να ξέρετε τι είναι κάθε θέση
  - απλή, με σκάλα, με φιδάκι, και με τα δύο, με all in και με πρόβατο
- έχει ένα παράδειγμα για το πως θα φαίνονται τα πιόνια όταν κινούνται, ουσιαστικά αλλάζουν οι φωτογραφίες.

### α) Σχεδίαση Αρχικής HTML σελίδας (5%)

Μπορείτε να σχεδιάσετε όπως θέλετε τη σελίδα σας. Για διευκόλυνση, σας δίνεται έτοιμος κώδικας που φτιάχνει τα τετραγωνάκια. Επίσης σας συνιστώ όταν κινούνται τα πιόνια, απλά να αλλάζετε τις εικόνες από τα αντίστοιχα τετράγωνα που επηρεάζονται. Σας δίνονται οι εικόνες για όλα αυτά, αλλά και σχετικός κώδικας.

Στην αρχική σελίδα πρέπει να έχετε εκτός από αυτά που δίνονται, ένα info box (μπορεί να είναι ένα άλλο div) που ο σκοπός του είναι να αναγράφει ποιος παίκτης παίζει, να δείχνει το αποτέλεσμα του ζαριού, κλπ. Επίσης χρειάζεται ένα κουμπί για το ζάρι, που θα δίνει ένα τυχαίο αριθμό από το 1 έως το 6.

### β) Συναρτήσεις Javascript (50%)

Θα πρέπει να έχετε τις κατάλληλες συναρτήσεις που να ελέγχουν

- **Αρχικοποίηση νέου παιχνιδιού (3 μονάδες) → newGame()**
  - Δίνει σειρά σε κάποιο παίκτη, αρχικοποιεί το infobox, μεταβλητές κλπ
- **Κίνηση Παίκτη (21 μονάδες)**
  - Θα έχετε έναν event handler που θα ενεργοποιείται μόλις πατηθεί το ζάρι να μετακινείται το πιόνι στη σωστή θέση με βάση τον αριθμό που έφερε → play()
    - Απλή κίνηση σε κουτάκι που δεν έχει σκάλα ή φίδι (4 μονάδες)
    - Κίνηση σε κουτάκι με σκάλα (5 μονάδες)
    - Κίνηση σε κουτάκι με φίδι (5 μονάδες)
    - Κίνηση που έχει και φίδι και σκάλα (7 μονάδες)
      - Σε αυτήν την περίπτωση ο παίκτης δίνει το μάθημα, και ένας τυχαίος βαθμός εμφανίζεται από 0-10. Αν φέρει από 0-4 τότε ακολουθεί τη διαδρομή του φιδιού, αν φέρει 5-10 τότε ακολουθεί τη σκάλα.
- **Συνάρτηση για το ποιος παίκτης παίζει (1 μονάδα) → getPlayerTurn()**
- **Αλλαγή σειράς → changePlayerTurn()**
  - Αν ο παίκτης φέρει 6 ξαναπαίζει (2 μονάδες)
  - Άλλιως αλλαγή σειράς (1 μονάδες)
- **Έλεγχος αν ο παίκτης κέρδισε → hasPlayerWon()**

- Για να κερδίσει ο παίκτης πρέπει να φέρει ακριβώς τη ζαριά που χρειάζεται για να φτάσει στο κουτάκι 90. Άλλιώς πηγαίνει προς τα πίσω. (**3 μονάδες**)
- Μόλις κερδίσει κάποιος, επιστροφή μηνύματος στους χρήστες και να τερματίζει το παιχνίδι (**2 μονάδες**)
- **Σωστή ανανέωση σελίδας HTML με κάθε κίνηση** (**5 μονάδες**) → `updateGUI()`
  - Να ανανεώνει το ταμπλό, ζάρι, infoBox κλπ
- **Έξτρα Κανόνες**
  - All In (**7 βαθμοί**)
    - Αν ο παίκτης πέσει σε κουτάκι με all in, τότε έχει τη δυνατότητα να πάει στο κουτάκι 90 και να κερδίσει είτε στο κουτάκι 1 στην αρχή. Η πιθανότητα να κερδίσει στην κλήρωση βγαίνει από τον αριθμό που βρίσκεται
      - πχ στο κουτάκι 21, έχει 21% πιθανότητα να πάει στο τέλος και να κερδίσει, και 79% να πάει στην αρχή.
  - Sheep (**5 βαθμοί**)
    - Αν ο παίκτης πέσει σε κουτάκι με το πρόβατο, τότε μεταφέρεται στο κουτάκι που βρίσκεται και ο αντίπαλος του, είτε τον συμφέρει είτε όχι.

**Μπορείτε να βάλετε όσες παραμέτρους θέλετε, να κάνετε όποιους ελέγχους επιθυμείτε και να ορίσετε νέες συναρτήσεις. Αρκεί να μην χρησιμοποιείται αυτούσιο κώδικα από το internet.**

**Θεωρείστε ότι τα πιόνια ξεκινάνε από έξω, όχι από τη θέση 1.**

- **πχ την πρώτη φορά που θα παίξει ο παίκτης, αν τύχει 3, θα πάει στη θέση 3.**

**Event Handling και ανανέωση HTML σελίδας:** Θα πρέπει να κάνετε σωστά το event handling, έτσι ώστε να γίνονται οι απαραίτητες ενέργειες και η ανανέωση της σελίδας. Μπορείτε να χρησιμοποιήσετε φυσικά και jquery.

### γ) Bonus (10%)

- **Effects - Sounds (5 μονάδες):** Να φαίνεται και η κίνηση του πιονιού σε κάθε κουτάκι (πχ, με τη χρήση timeouts κλπ), και να υπάρχουν ήχοι (πχ για το ζάρι).
- **Παιχνίδι με υπολογιστή (5 μονάδες):** Μπορείτε να υλοποιήσετε ένα παιχνίδι (**σε διαφορετικό αρχείο**) όπου να παίζει και ο υπολογιστής ενάντια στον παίκτη.

### Hints

- **SOS. Δείτε και το παράδειγμα με τα games που δείξαμε στο μάθημα.**
- Το input μπορείτε να το δίνετε μέσω κουμπιών
  - Μπορείτε το ζάρι να είναι ένα κουμπί, όταν η κίνηση είναι επιτρεπτή αλλάζει με τη σωστή φωτογραφία/χρώμα (Εικόνες έχουν αναρτηθεί στο elearn)

- Οι θέσεις που έχουν φιδάκι ή σκάλα και πού οδηγούν, αλλά και οι ειδικές θέσεις υπάρχουν μέσα στον κώδικα, και επίσης γίνονται print ,μέσω της console.log().

Σημειώσεις:

*Mη ξεχνάτε τη χρήση του “use strict”; για την JavaScript. Το parse ενός JSON string γίνεται με χρήση της JSON.parse(str) που επιστρέφει το javascript object που αντιστοιχεί στο str, δεδομένου ότι το str είναι μια σωστή αναπαράσταση JSON . Μπορείτε να χρησιμοποιήσετε κάποιον online linter όπως ο o jshint <http://jshint.com/> για να βελτιώσετε την ποιότητα του js κώδικά σας.*

### Τρόπος Παράδοσης

Η Παράδοση θα γίνεται μέσω elearn. Θα πρέπει να παραδώσετε ένα zip που να περιέχει ένα φάκελο A2\_AM, όπου AM ο αριθμός μητρώου σας. Μπορείτε να έχετε ένα φάκελο για τις ασκήσεις 1 και 2, και ένας άλλο φάκελο για την άσκηση 3. Εκπρόθεσμες ασκήσεις δεν θα γίνονται δεκτές.

### Αντιγραφή!

Σε περίπτωση αντιγραφής θα μηδενίζονται άμεσα οι εργασίες όλων των εμπλεκόμενων. Άλλωστε κατά την εξέταση θα σας γίνουν πολλές ερωτήσεις που θα δείξουν κατά πόσο κατανοείτε τον κώδικα που έχετε παραδώσει. Οπότε ασχοληθείτε με την άσκηση και παραδώστε ότι καλύτερο μπορείτε και ότι δεν καταλαβαίνετε ρωτήστε στο elearn.

Καλή εργασία