

## Programmation Web

### TP n° 5 : TP noté

NOM :

PRENOM :

Numéro d'étudiant :

Téléchargez sur Moodle l'archive correspondante à votre sujet. A la fin, veuillez rendre une archive (limité à 50 MB) avec votre TP sur Moodle.

## 1 Node/Express

Normalement vous avez déjà installé pour vos TPs précédent `express` et `ejs` (il fallait faire `npm install express` et `npm install ejs`) en complément à `node`. Les fichiers qui vous serviront de trame pour cet exercice sont `serveur3.js`, `public/welcome.html`, et `views/ok.ejs`, `views/banni.ejs`, `views/debug.ejs`, `views/question.ejs`

Le principe de cet exercice est de créer un serveur qui maintient une liste de personne qu'on a identifiées comme n'étant pas des robots. Pour cela on leur pose 3 questions simples. S'ils n'arrivent pas à répondre ils sont bannis, sinon leur route sera acceptée pour toujours.

1. Complétez la première règle `debug` en l'associant au patron `debug.ejs` pour que la route correspondante affiche le contenu courant du tableau `userList` sous la forme :

- Prof : ok
- Visiteur0 : banni
- Visiteur1 : 0

2. Pour la règle correspondant à `/:name` tout va dépendre du statut de la personne dans `userList`.

1. Si cette personne a déjà validé ses tests alors elle a été marquée comme `ok` et il faut la rediriger vers `ok.ejs` et l'accueillir.

2. Si elle a échoué alors elle a été marquée `banni` et une tentative d'utiliser cette route la conduira à `banni.ejs`. Cette page propose cependant une solution : être oublié et effacé des registres. Complétez le formulaire et introduisez une route `desinscription`. Indications : Pour transmettre une valeur via le formulaire sans la demander explicitement vous pourrez utiliser `input type="hidden"`. Lors de la désinscription, vous afficherez dans la console qui a été oublié. Vous pourrez utiliser `delete` pour supprimer un élément d'un objet associatif. Puis redirigez l'utilisateur vers la route par défaut, en utilisant `redirect()`

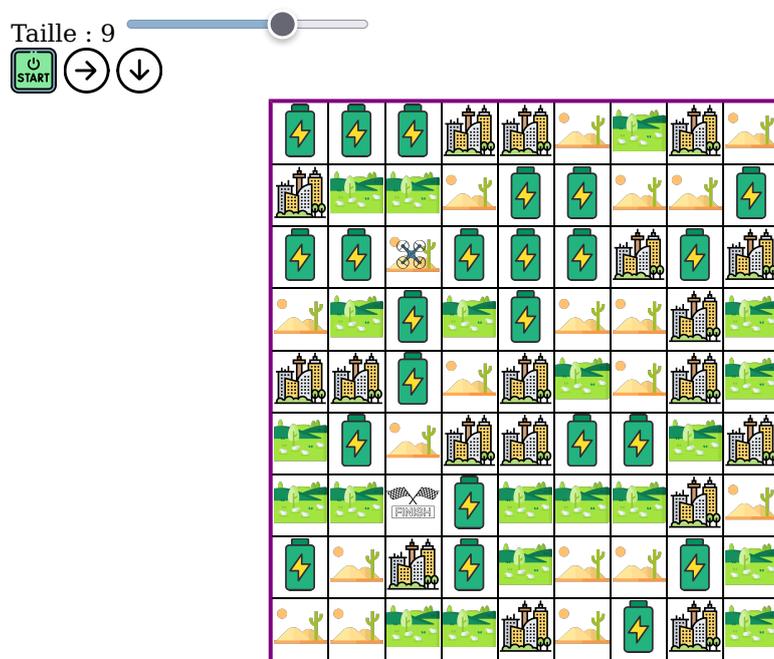
3. Il reste le cas d'une personne connue ou inconnue qui n'a pas complété ses tests. La valeur numérique associée dans `userList` indique le nombre de succès qu'elle a eu. Vous pouvez donc commencer par traiter le cas d'une personne inconnue en vous assurant déjà qu'elle soit connue dans cette liste.

Le test va consister à trouver le maximum entre deux chiffres choisis aléatoirement. Le serveur va préparer la question en choisissant ces valeurs, et il stockera dans `expectedAnswer` la réponse attendue de l'utilisateur. Le serveur l'amènera à passer son test via `question.ejs`. Vous y trouverez un formulaire prérempli. Il vous faut poser la question sous une forme non triviale : affichez **dans un désordre complet** à l'intérieur de la `div "question"` un nombre `nbBlue` de carrés bleus et un nombre `nbRed` de carrés rouges. Ces valeurs vous auront été transmises par le serveur. Indication : `<div class="carre blue"> </div>`

3. Terminez par le traitement de la réponse.

## 2 CSS + JQuery

Dans cette partie, vous allez devoir modifier les trois fichiers `drone.html`, `drone.js` et `drone.css` pour obtenir petit à petit un résultat comme celui là (on ne voit pas très bien, mais un drone est en (2,2) )



[css simple]

4. Changez (dans le `.css`) la taille des images présentes dans le `div` "boutons" afin qu'elles aient une taille comparable à de petites icônes.
5. Centrez la "zoneJeu" afin qu'elle occupe la moitié de la largeur de la page, et qu'elle soit centrée horizontalement.
6. Entourez le "terrain" d'une bordure violette de largeur 3
7. modifiez les attribut `border` et `border-collapse` du tableau et/ou de ses éléments afin que les séparations entre cases apparaissent sous la forme d'un trait simple.

[jquery]

8. Dans la fonction `reinit()` cachez le "drone"
9. Dans `drone.js` écrivez la fonction `placeImages` : elle devra alimenter la table "terrain" en construisant  $N$  lignes de  $N$  colonnes contenant une image aléatoire prise dans `tab`. Ces images seront de classe "photo" et leurs id respectives seront numérotées 0, 1, 2, etc.. si on les regarde dans le sens habituel de lecture : de gauche à droite puis de haut en bas.
10. Remplacez l'une de ces images (choisie aléatoirement) par `arrivee.png`

[gestion du slider - vue en tp2b]

11. Lorsque la valeur du slider change vraiment, on souhaite tout réinitialiser :
  - cacher le drone
  - redessiner une nouvelle table "terrain" de la nouvelle dimension
  - afficher sur la console que la dimension a changé (on veut les deux valeurs : l'ancienne et la nouvelle)
  - afficher aussi cette valeur dans le `html` (voir dessin, à coté de "Taille")
 Ecrivez cela.

**[gestion du drone]**

12. Le "drone" (actuellement caché) est un élément que nous allons ensuite déplacer très précisément au dessus du "terrain", c'est à dire que les `position` de ces deux `div` et aussi celle de la "zoneJeu" devront être ajustées parmi `static`, `relative` et `absolute` (en l'état l'ordre d'écriture fait que le "drone" arrive dans le flux après le "terrain" et est affiché en dessous ce qui n'est pas satisfaisant). Trouvez la bonne combinaison `css` pour les positionnements.
13. lorsqu'on appuiera sur le bouton "start" le drone apparaîtra : les variables `Xdrone`, `Ydrone` seront utilisées pour le repérer dans  $[0, N]^2$  où  $(0, 0)$  représente sa case de départ en haut à gauche de la table "terrain". Pour que son placement soit élégant, vous le ferez occuper une largeur exprimée en un pourcentage de "zoneJeu" :  $\frac{100}{2*N} \%$  et vous le placerez en le décalant en haut et à gauche de  $\frac{100}{4*N} \%$
14. L'appuis sur l'icône `fleche.png` fera évoluer le drone d'une case vers la droite, sans le faire sortir de l'écran. Pour cela vous invoquerez sur le drone la méthode d'objet `x.animate( , )` où le second argument est un entier dont l'unité est interprétée en milliseconde, et le premier argument est un objet de la forme `{attribut : nouvelleValeur}` : l'élément `x` du DOM verra alors linéairement son attribut changer pour prendre la nouvelle valeur. Il vous faudra calculer cette nouvelle valeur (un pourcentage) en fonction de `Xdrone` et de `N`.
15. (**hors barème**) Si vous avez terminé, poursuivez en introduisant quelques règles supplémentaires pour faire de tout cela un petit jeu.