IUT – Département informatique
2ème Année – Formation initiale

1 février 2016

Jean-Michel Ilié, Camille Kurtz

TP
question aléatoire et chargement dynamique en jquery

Rappel :
La variable globale `tabObject` est un tableau javascript qui stocke pour chaque question un objet (ou tableau associatif) contenant 3 propriétés :

(1) le texte de la question,
(2) un tableau de choix possibles,
(3) l'indice de la bonne réponse parmi les choix.

Question 1 : Définition d'une interface riche
Faire une recherche sur jquery-demos à partir de google, tester et visualiser le code source correspondant à `droppable` et `draggable`. Lors d'un cliquer-glisser une div déplacée va devenir la div-fille (au sens, imbrication du terme) d'une autre div utilisée comme réceptacle. Graphiquement la div déplacée est plus petite que la div réceptacle, pour bien représenter l'inclusion de la div déplacée

S'inspirer de la démo de jquery pour que l'utilisateur sélectionne sa réponse graphiquement par un cliquer-glisser. Initialement le texte de la question s'affiche dans la fenêtre et les choix apparaissent dans les div-choix. Dans un temps limité, l'utilisateur fait un `cliquer-glisser` à partir d'une `div-choix` jusqu'à la `div-reponse` pour que s'affiche le résultat de la réponse dans la `div-réponse`.
Pour une question donnée, des `div` seront définis dynamiquement, à raison d'une par choix possible de réponse, les `div-choix`, et à raison d'une supplémentaire pour la réponse , la `div-réponse`.

- Les div-choix sont repérés par l'attribut `class= "draggable"`.
- La div-réponse a comme identifiant `id= "droppable"`.
- Ces div contiennent chacune un paragraphe `<p>` pour l'affichage d'un texte.

Il est possible d'ajouter des styles en exploitant les classes de style proposées avec jquery-ui, par exemple avec la fonction `$.addClass()` de jquery. Le tableau interne de l'objet `$` doit alors contenir les références aux div sur lesquelles on désire ajoutée une classe.

Les gestions événementielles des `div-choix` et de la `div-reponse` sont à déclarer pour être pris en compte au moment du chargement des objets dans la page.

- l'utilisation de la fonction `$.draggable()` en référence aux `div-choix` suffit à les rendre cliquables et déplaçables. Le tableau interne de l'objet $ devra contenir donc les références à ces `div`.
  Cette fonction peut avoir un objet en paramètre, soit `$.draggable({`*`objet`*`})`, dont les propriétés caractérisent le comportement de l'objet déplacé, lors de son déplacement. La propriété `revert:"invalid"` de cet objet précise que la `div-choix` déplacée ne revienne pas à sa place si elle n'est pas finalement placée dans la `div-réponse`.

```
$( .draggable" ).draggable({
```

```
            revert: "invalid";
      });
```

- l'utilisation de la fonction `$.droppable()` en référence à la `div-réponse` suffit à la rendre réactive à la fin d'un clic dans son espace graphique, c'est le fameux `drop`. Pour être effectif, le tableau interne de l'objet `$` devra contenir la référence à cette `div`.
  Le paramètre optionnel de cette fonction est un objet dont la propriété `drop` permet de définir sous la forme d'une fonction, le comportement de l'objet receveur, la `div-reponse`, lors de la réception.

```
$( "#droppable" ).droppable({
      drop: function( event, ui ) {….}
});
```

  Pour écrire le code de la fonction `$.droppable()`, on exploite les 2 points suivants :
  - Au moment du `drop`, la référence `this` réfère alors la `div-reponse`.
  - De plus, la référence à la div déplacée est donnée par l'objet jquery `ui.draggable`.

  Il est ainsi possible d'accéder à l'identifiant de la div déplacée et de le comparer à l'indice de la bonne réponse. Une fois le résultat connu, le message `repOK` ou `repKO` peut être affiché, dans le paragraphe de la `div-reponse`.

Question 2 : Lecture des données du jeu sur le serveur web

Au lieu de spécifier le tableau de questions statiquement dans le code javascript, le tableau sera transmis à partir d'un script PHP au moment de chargement de la page HTML en mémoire, via une requête AJAX.
Le tableau pourra être défini statiquement dans le script PHP ou mieux être construit dynamiquement à partir d'une base de données.
Le code PHP peut être développé suivant le paradigme MVC.

Une méthode Ajax en javascript-jquery est une fonction javascript demandant dynamiquement l'exécution d'un script (ou d'un exécutable) sur un serveur web. Classiquement, la transmission `HTTP` suit la méthode `GET` ou `POST`. A l'instar des liens ou des formulaires, on peut préciser des paramètres en entrée. La réponse du script (à savoir ces echo, et autres fonctions produisant des données sur la sortie standard du script) est transmise par le serveur web puis traitée au niveau du navigateur par une fonction que l'on doit implémenté et précisé en paramètre de la requète. L'écriture de cette fonction précise que faire du contenu transmis, qui peut donc prendre différentes formes : un texte html à placer, une donnée javascript. Avant leur transmission les données sont linéarisées au format json, sachant que le format json est reconnu syntaxiquement en javascript.

- la linéarisation en php est faite par la fonction `json_encode($D)` , où $D est la donnée à linéarisée, par exemple un tableau PHP. On obtient une chaîne formatée à transmettre vers le navigateur, avec un `echo()` par exemple.
- La fonction inverse est `json_decode($D)` . Elle pourra servir à récupérer une donnée json transmise en paramètre d'une requête http.
- Les paramètres d'une requète Ajax en Jquery sont donnés sous la forme d'un objet json,
  `$.ajax({`*objet*`})` voir un exemple plus loin, en annexe.
- si la donnée (texte) transmise est a priori correcte, il suffit de faire évaluer la chaîne correspondante comme une expression javascript et de l'affecter à une variable
  `var tabObject = eval('(' + texteJSON + ')');` où est le texte JSON à transformer en données javascript.

- A noter que si l'on précise type:"json" comme propriété de l'objet passé en paramètre d'une requête ajax, alors l'évaluation est fait directement si bien que l'on récupère directement une donnée javascript.

Exemple d'utilisation de la fonction $.ajax

```
Voici un exemple de la fonction $.ajax :
$.ajax({
    url: 'url_adressant_un_script_php',
    type: 'POST',       /*ou GET*/
    data: {/* paramètres ou données transmis au script php*/
        user: 'jmi', /*c'est un exemple*/
        age: 19
    },
   error: function() {
     // En cas d'erreur, on le signale en écrivant cette fonction
          },
    success: function(data) {
     // data est la donnée transmise par le script
       // et on écrit cette fonction pour l'exploiter
         }
     dataType : 'json',  /*data sera une donnée javascript
                         issue d'une réception au format json*/
                         /*Autres valeurs pour dataType: 'html', 'xml'.*/

  });
```