TP 07 — Objets, XML et Morpion

fabien.givors@unice.fr

2014-2015

- Le présent TP a pour objet de vous faire réaliser un mini-jeu de morpion, jouable en réseau.
- Ce faisant, vous allez commencer à concevoir en utilisant le paradigme « Objet »
- Finalement, vous allez utiliser la bibliothèque DOM (DOMDocument, DOMElement, etc.) afin de lire et écrire des données au format XML.

1 Structures de données et affichage

Rappel des règles du jeu de morpion :

- Le jeu de morpion consiste en une grille 3×3 comportant chacune soit un cercle, soit une croix, soit rien.
- Au début de la partie, la grille est vide.
- Un joueur est choisi pour commencer.
- Tour à tour, jusqu'à ce qu'une combinaison gagnante apparaisse, les joueurs ajoutent un cercle ou une croix sur un emplacement libre de la grille.
- Lorsqu'une combinaison gagnante apparaît ou que la grille est pleine, la partie se termine.

Exercice 1. Dans un fichier Morpion.php, créer une classe Morpion. Cette classe devra contenir :

1. Un champ privé `$grille` qui stockera les éléments de la grille.
2. Un champ privé `$joueur` qui stockera le numéro du joueur ayant commencé la partie.
3. Un constructeur prenant comme paramètre un numéro de joueur (1 ou 2) et initialise un jeu de Morpion dans lequel le joueur en question commence la partie.
4. Une méthode publique `nbTours()` qui compte le nombre de tours écoulés.
5. Une méthode publique `aQuiLeTour()` qui renvoie 1 si c'est au joueur 1 de jouer, et 2 sinon.
6. Une méthode publique `partieGagnee()` qui renvoie 1 si le joueur 1 a gagné, 2 si le joueur 2 a gagné, 0 sinon.
7. Une méthode publique `cocheCase($joueur,$c,$l)` qui, si c'est bien à `$joueur` de jouer, et si la partie n'est pas gagnée, coche la case de la colonne `$c` et de la ligne `$l` (les numérotations commençant à 0.)

Exercice 2. On veut maintenant pouvoir insérer ce jeu dans une page. Pour ce faire, nous allons définir une méthode publique `afficheGrille()`.

1. Définir une méthode publique `afficheGrille()` qui affiche la partie en cours, sous forme de tableau[1]
2. Compléter la méthode pour que le tableau soit inclus dans un formulaire, et que les cases vides contiennent un bouton de type radio dont la valeur représente les coordonnées de la case sous forme « x,y ».
3. Compléter la méthode pour qu'elle prenne en paramètre une variable `$joueur`, et rajoute un champ *hidden* au formulaire, dont la valeur est le numéro du joueur.
4. Compléter la méthode pour qu'elle affiche « C'est à vous de jouer » si c'est le cas, ou « C'est à votre adversaire de jouer » sinon.

[1] On pourra utiliser des o et des x pour représenter les cercle et les croix, ou, pour les adeptes de CSS, des cases bleues et des cases rouges, des loutres et des ornithorynques, *whatever*.

2 Jouer

Exercice 3. Dans un premier temps, nous allons créer une page PHP permettant de jouer au jeu depuis le même navigateur.

- Créer une page `jeu-ms.php`. Cette page commence par restaurer ou initialiser la session. S'il n'y a pas de partie en cours, ou si la partie en cours est gagnée, alors une nouvelle partie est initialisée. Le joueur commençant la nouvelle partie est le joueur 2 si la partie précédente était commencée par le joueur 1, le joueur 1 sinon.
- N'oubliez pas de définir un `session_name` unique (votre login par exemple).
- La page `jeu-ms.php` prend maintenant en paramètre via `GET` un numéro de joueur, et affiche la grille pour le joueur en question. Si le numéro de joueur n'est pas valide, ou inexistant, la page doit afficher deux liens, un qui permet de jouer en tant que joueur 1, l'autre qui permet de jouer en tant que joueur 2.
- Compléter la page pour que celle-ci reçoive les données via POST, effectue les actions demandées par le joueur, puis redirige (avec header) vers la page courante.

3 XML et sauvegarde de partie

Nous allons maintenant tenter de modifier notre code pour que le jeu puisse être joué depuis plusieurs ordinateurs différents. Problème : jusqu'à présent, les parties étaient stockées dans la session. Ce ne pourra plus être le cas maintenant. Nous allons donc stocker les parties dans des fichiers XML.

Pour cela, nous allons devoir imaginer une structure de fichier XML permettant de représenter une partie. Je vous propose la structure suivante, mais vous pouvez l'adapter à votre guise :

```
<morpion>
  <joueur>1</joueur>
  <grille>
    <l>
       <c>0</c>
       <c>0</c>
       <c>2</c>
    </l>
    <l>
       <c>1</c>
       <c>1</c>
       <c>2</c>
    </l>
    <l>
       <c>0</c>
       <c>0</c>
       <c>0</c>
    </l>
  </grille>
</morpion>
```

Exercice 4. Export de la partie Dans la classe Morpion…

1. Créez une méthode privée `exportXML()` qui construise et retourne un objet DOM XML représentant la partie en cours suivant le schéma donné ci-dessus.
2. Créez une méthode publique `sauvePartie()` qui enregistre la partie en cours dans le fichier `partie.xml`.

Exercice 5. Import de la partie Dans la classe Morpion…

1. Créez une méthode publique `chargePartie()` qui charge un DOM en depuis le fichier `partie.xml` et la transmet à la méthode `importDOM($dom)`.
2. Créez une méthode privée `importDOM($dom)` qui initialise `$grille` et `$joueur` en fonction d'un DOM `$dom`.

4 Jouer (bis)

Exercice 6. Dans un fichier `jeu-mj.php` reprenant le fichier `jeu-ms.php`, adaptez le code pour que le jeu ne soit plus stocké dans les sessions et qu'il puisse être joué depuis des navigateurs différents.