



# Rattrapage de Java EE

## 2010 – 2011

### Conditions d'examen :

- Tous les documents sont autorisés sauf Internet.
- Le devoir se fait sur machine et se rendra sur une clef USB fournie par les examinateurs.

**Note Importante :** Vous devez implémenter le site web suivant en **JEE**. Le fichier que vous devez rendre est un **WAR** contenant **l'ensemble de votre site AINSI QUE LES SOURCES Java**. Tout autre type de rendu (zip du projet Eclipse, zip des fichiers, ...) entraînera automatiquement une note nulle.

## Pendu

Le but de cet exercice est de réaliser un site permettant à un utilisateur de jouer au jeu du pendu. On veut que les traitements se fassent du côté du serveur, donc à chaque fois que le joueur voudra jouer, une requête sera envoyée au serveur qui renverra la page modifiée.

Le principe du jeu du pendu est le suivant :

- Un mot est généré, contenant **n** lettres, et est affiché : 1 tiret par lettre cachée.
- L'utilisateur doit ensuite entrer lettre par lettre, pour deviner le mot caché
- S'il entre une lettre qui appartient au mot, cette lettre n'est plus cachée dans l'affichage du mot.
- Lorsque toutes les lettres ont été entrées, il gagne et peut rejouer.

-----

Lettres Jouées :

Quelle lettre voulez-vous jouer?

e t h i q u e

Lettres Jouées :

a z e t h i q u

Quelle lettre voulez-vous jouer?

**GAGNE !!** Voulez-vous rejouer?

-----

Lettres Jouées :

a z

Quelle lettre voulez-vous jouer?

**Erreur : Vous ne pouvez pas jouer cette lettre, choisissez entre a et z**

\_ e \_ \_ u

Lettres Jouées :

a z e q r t u c

Quelle lettre voulez-vous jouer?

**Erreur : Lettre incorrecte**

## Remarques :

- Pour réaliser cet exercice, nous respecterons les normes MVC du JEE.
- L'utilisation des JSTL et des CSS n'est pas obligatoire mais vous procurera un bonus s'ils sont correctement utilisés.
- Le code doit être relativement clair (évités le plus possible de mettre de gros blocs de code Java dans vos JSP => utilisez des méthodes statiques).

Question 1 : Commencez par créer le modèle (**JavaBean**) de votre pendu. Vous pouvez créer le modèle qui vous convient. Pour vous orienter vers une solution viable : nous pouvons supposer que notre jeu de pendu contient 2 beans : un bean contenant le dictionnaire de mots, et un autre contenant le mot en cours de déchiffrement, la liste des lettres déjà jouées par l'utilisateur et un tableau de booléens (de la taille du mot) qui associe à chaque lettre du mot, *true* si la lettre correspondante du mot à été trouvée (et *false* sinon).

Question 2 : Créez maintenant la vue de votre site Web. Si le dictionnaire n'a pas encore été généré, vous afficherez un message de bienvenue et les beans seront créés (attention on veut du MVC, passez par une servlet). Sinon affichez le jeu en cours. Vous pourrez vous inspirer de la vue précédente pour générer votre application.

Question 3 : Créez le (ou les) contrôleur de votre site Web, qui récupère les différents paramètres et crée/modifie les beans du modèle. Lors de la création des beans, vous devez générer le dictionnaire. La manière la plus simple de faire est de créer un dictionnaire à la main, en Java. Vous pouvez éventuellement commencer ainsi, cependant vous n'aurez tous les points que si vous faites une lecture d'un fichier *dico.txt*, stocké sur votre serveur, qui contiendra les mots de votre dictionnaire (une dizaine de mots suffisent). Stockez les beans dans le(ou les) scopes appropriés. Le contrôleur doit être appelé dans le formulaire par l'URL suivante : (url du site)/Proposition.

Question 4 : Le contrôleur doit gérer d'éventuelles erreurs de saisies du formulaire : entrée incorrecte, pas une lettre, plus d'une lettre, ...

Question 5 : Ajoutez un compteur pour chaque lettre incorrecte. Si 7 lettres incorrectes ont été jouées, le joueur perd.

## Annexe Java / CSS

- **Java :** Transformer un String en tableau de char.

```
String mot = "toto";
char[] mot = mot.toCharArray();
```
- **Java :** Passer un String en majuscule / minuscule

```
String mot = "toto";
String maj = mot.toUpperCase();
String min = mot.toLowerCase();
```
- **CSS :** Pour changer la couleur du texte :

```
color : yellow;
```