Programmation Orientée Objet
L2S4 2007–2008
hoenen@icps.u-strasbg.fr

Université Louis Pasteur
UFR de math-info
11 mars 2008

TP2

Le but de ce TP est de réaliser un jeu simple du pendu en mode console en manipulant chaînes de caractères et tableaux. N'oubliez pas de garder un oeil sur l'API java, notamment sur la classe `String` et ses méthodes.

```
 -----
 |/ |
 |  o
 | /|\
 | / \
_|_
S I _ G E S
```

FIG. 1 – Affichage du pendu en mode console.

Haut et Court

Ici on va construire la classe qui contient toutes les informations nécessaires au jeu : le mot à trouver, le nombre de fautes autorisées, le nombre de fautes commises. tous les caractères seront traités en majuscule.

1. Ajoutez un tableau à 2 dimensions (un tableau de tableaux) de taille 6×6 pour décrire le dessin du pendu. Ce tableau ne doit pas être modifiable et doit être un attribut de classe et non d'instance.
2. Ajoutez un deuxième tableau de même taille qui contient pour chaque élément du tableau de dessin, le nombre d'erreur à partir duquel cet élément du dessin est affiché. Écrivez une fonction qui affiche le dessin du pendu en fonction du nombre d'erreurs commises.
3. Le dessin est composé de 20 caractères. On veut pouvoir donner le nombre de fautes autorisées f ($1 \leq f \leq 20$) en argument de la ligne de commande au lancement du programme. Comment faire pour que le dessin du pendu se mette correctement à jour à chaque faute commise ?
4. Trouvez un moyen rapide pour se souvenir des lettres déjà trouvées, autrement dit pour décrire le mot à trous. Ajoutez une méthode qui affiche le mot à trou (`_ I _ G _ _`).
5. Ajoutez une méthode qui, étant donné un caractère, renvoit un booléen indiquant si oui ou non le caractère appartient au mot recherché. Cette méthode doit modifier l'état du mot à trous.
6. Ajoutez une méthode qui teste l'état du jeu et détermine si on a trouvé toutes les lettres.

On va maintenant s'intéresser au `main` et à la gestion du joueur. Vous pourrez utiliser `InputStreamReader` pour considérer l'entrée standard comme un flux de caractère.

7. Ajoutez une méthode `main` qui va construire une instance de la classe pendu et l'initialise à l'aide du nombre d'erreur autorisées et du mot à trouver passés en argument de la ligne de commande. Pensez à convertir les caractères en majuscules.
8. Testez l'affichage du pendu étape par étape pour différents nombres d'erreur autorisées.
9. Ajoutez la boucle principale du jeu : on demande à l'utilisateur de rentrer un caractère jusqu'à ce que la solution soit trouvée ou que le nombre de fautes autorisées soit atteint (n'oubliez pas d'afficher le nouveau pendu à chaque faute).

De l'héritage ?

On veut pouvoir différencier le coeur du jeu (ses règles) et l'aspect intéraction avec le joueur.

1. Transformez votre classe `pendu` en une hiérarchie de classes : la super de niveau supérieur doit contenir le coeur du jeu et celle au niveau inférieur doit contenir le main et les méthodes nécessaires à l'intéraction (affichage / leture) avec l'utilisateur en mode console.

2. Comment être sur que les différentes sous classes implantent toutes les méthodes nécessaires au bon fonctionnement du jeu ? Faites les modifications nécessaires pour vous en assurer.
3. Essayez d'écrire une nouvelle sous classe un peu plus *sexy*, en utilisant par exemple des boites de dialogue pour interagir avec l'utilisateur.