

## Plateformes de Développement d'Applications Distribuées

## Travaux Dirigés N° 01

## Préliminaires : installation de J2SE et de NetBeans

L'installation de l'environnement de développement **NetBeans 6.7.1 + J2SDK 1.6** sur les machines de l'ENIS se fera par les étudiants sur les postes Ubuntu. Il s'agit simplement de récupérer une archive précompilée préparée pour les TDs. Cette archive peut aussi être utilisée sur tout PC exécutant le système d'exploitation Ubuntu (8.04, 8.10 ou 9.04).

## Récupération et extraction de l'archive

Après avoir récupéré l'archive `ENIS_J2EE_Package_R4I.tgz` (l'adresse sera communiquée au cours de la séance de TD), il faudra l'extraire. Ensuite, dans un terminal, saisir les commandes suivantes :

```
~% cd /opt
~% tar xzvvf /chemin/vers/ENIS_J2EE_Package_R4I.tgz
```

À l'issue des deux commandes ci-dessus, on disposera d'un répertoire `/opt/J2EE` contenant l'installation de **NetBeans** ainsi que des outils de développement et d'exécution **JAVA 6** de SUN. On disposera aussi d'un fichier nommé `env.sh` qui permet de placer correctement les variables d'environnement nécessaires pour la compilation, le déploiement et l'exécution des applications réparties. Ce fichier doit être "**chargé**" **chaque fois** qu'un nouveau terminal (shell) est ouvert. Le chargement du fichier `env.sh` s'effectue, **dans le nouveau terminal**, à l'aide des commandes suivantes :

```
~% source /opt/J2EE/env.sh
```

Pour lancer **NetBeans**, il suffit de saisir la commande `netbeans&` dans un terminal.

## Exercice : Tours de Hanoi

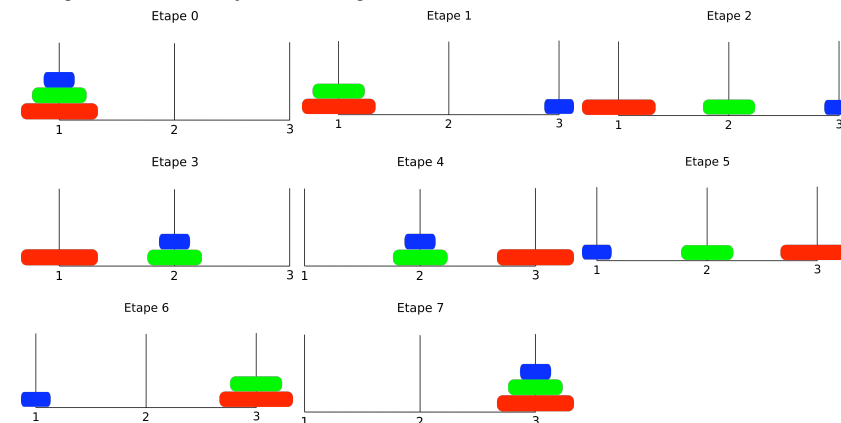
"Les Tours de Hanoi" est un jeu classique qui consiste à déplacer un ensemble de disques de tailles différentes à partir d'une position de départ vers une position d'arrivée en utilisant une position intermédiaire. Le nombre de disques (formant une tour) est appelé l'ordre du jeu.

Les disques sont placés sur la position de départ; celui de plus grand diamètre étant le plus bas et chaque disque est de diamètre plus petit que ceux de tous les disques situés en dessous de lui.

Deux règles doivent être vérifiées lors de la résolution du jeu :

1. On ne déplace qu'un seul disque à la fois.
2. Un disque ne peut se retrouver au dessus d'un disque de plus petit diamètre.

Exemple de résolution du jeu avec 3 disques :



L'objectif de ces travaux dirigés est de construire une application web distribuée qui permet l'illustration de la résolution du problème des "tours de Hanoi". Cet application utilisera un servlet.

**Un exemple d'exécution de l'application est donné à la fin de ce document.**

Question 1 : Création d'un nouveau projet et ajout du fichier `index.html`

Créer un nouveau projet d'application Web en utilisant **NetBeans**. La création d'un nouveau projet s'effectue en utilisant la commande "New Project" du menu "Fichier".

## Conseils :

- Nom du projet : `HanoiAvecServlet`
- Serveur d'application : `Glassfish 2.1` (et **non pas** `Glassfish 3`)

On remarquera que **NetBeans** génère un grand nombre de fichiers situés dans une arborescence bien organisée. Parmi les fichiers et répertoires les plus importants, on peut citer :

- **Web Pages** (répertoire `web`) : ce répertoire rassemble toute la partie web de l'application (pages HTML, JSP, images...).
  - `index.jsp` : fichier de bienvenue principal de l'application généré par **NetBeans**
  - `WEB-INF/web.xml` : descripteur de déploiement de l'application web. On remarquera qu'il précise `index.jsp` comme *fichier de bienvenue* principal
- **Source Packages** (répertoire `src`) : ce répertoire rassemble les fichiers sources Java de l'application
- `Build.xml` (visible uniquement sous l'onglet *Files*) : fichier utilisé par l'outil `ant` pour compiler, déployer et exécuter l'application.

La première tâche de notre application consiste à remplacer le fichier `index.jsp` par un simple fichier HTML (`index.html`), puisque, dans ce TD, on n'utilisera pas la technologie JSP. Il ne faut pas oublier de modifier le descripteur de déploiement (`web.xml`) pour qu'il pointe vers le nouveau fichier bienvenue.

**Conseils :**

- Pour supprimer un fichier, cliquer dessus avec le bouton droit et choisir “Delete”
- Pour ajouter un nouveau fichier HTML, cliquer sur le nom du projet avec le bouton droit, choisir “New → HTML” et enfin, choisir **index** comme nom du nouveau fichier.

Remplacer le contenu de la section **<body>** du fichier HTML par le contenu suivant :

```
<h1>Tours de Hanoi</h1>
Cette page permet d'illustrer la résolution automatique du problème des
tours de Hanoi en utilisant un Servlet.
<p></p>

<form action="HanoiServlet" method="post">
  Nombre de disques : <input type="text" name="ntours" size="25">
  <p></p>
  <input type="submit" value="Résoudre">
  <input type="reset" value="Effacer">
</form>
```

**Question 2 : Déploiement et exécution de l'application**

On va maintenant déployer et exécuter l'application répartie (incomplète) en suivant les étapes suivantes :

1. Cliquer sur le nom du projet avec le bouton droit et choisir “Deploy”. Cette commande cause le démarrage du serveur Glassfish 2.1 si ce dernier n'est pas déjà démarré. Ensuite cette commande cause le déploiement de l'application HanoiAvecServlet sur le serveur.
2. Cliquer sur le nom du projet avec le bouton droit et choisir “Run”. Cette commande cause l'ouverture d'un navigateur web et l'affichage de la page : <http://localhost:8080/HanoiAvecServlet/>. Si le navigateur web ne s'ouvre pas, on pourra l'ouvrir manuellement et afficher la page dont l'adresse est donnée ci-dessus.

Bien entendu, notre application ne marche pas encore correctement puisque le servlet qui est censé résoudre le problème des tours de Hanoi n'existe pas encore. Ce sera l'objectif de la question suivante.

**Question 3 : Ajout du servlet hanoi.HanoiServlet**

Ajouter une nouvelle classe de servlet appelée `hanoi.HanoiServlet`. Pour ce faire, cliquer sur le nom du projet avec le bouton droit et choisir “New → Servlet”

**Conseils :**

- Nom de la classe : `HanoiServlet`
- Paquetage : `hanoi`

On remarquera que le serveur génère une classe assez complète contenant les méthodes suivantes :

- `processRequest` : méthode à compléter qui contiendra le travail du servlet

- `doGet` : traitement par la méthode GET
- `doPost` : traitement par la méthode POST

**Question :** Quelle est méthode qui sera exécutée par le servlet lors du traitement d'une requête cliente? Justifier.

S'inspirer de la version incomplète fournie avec ce TD pour compléter la classe `HanoiServlet` (l'adresse de cette version sera fournie lors de la séance).

**Question 4 : Déploiement et exécution de l'application**

Déployer, exécuter la nouvelle version de l'application et vérifier l'exactitude de ses résultats.

**Annexe : Exemple d'exécution**