

IN56 – Examen final  
Partie 1  
Documents de cours et TDs autorisés

Un jeu d'échecs est constitué d'un damier de 8x8 cases (alternées noir et blanc) et de deux ensembles de 16 pièces (Noires et Blanches).

Les types de pièces sont au nombre de 6 : Roi (R), Dame (D), Tour (T), Cavalier (C), Fou (F) et Pion (P).

Il y a 16 pièces par couleur : 1 Roi, 1 Dame, 2 Tours, 2 Cavaliers, 2 Fous, 8 Pions.

1/ (2 + 2 pts) Un état de la partie d'échecs est représenté par un fichier au format XML, de la forme 'jeu.xml' figurant en annexe 1. L'image représente la configuration graphique de la partie décrite.

1.1/ Décrire la DTD du fichier 'jeu.xml'

1.2/ Décrire le schéma XML du fichier 'jeu.xml', le plus précis possible

2/ (3 + 3 + 2 pts) Par ailleurs, la configuration des pièces de ce jeu est décrite au format XML, par le fichier 'partie.xml' figurant en annexe 2.

2.1/ Transformation XML ==> XML

Décrire la transformation XSL qui fait passer les données décrites dans 'jeu.xml' dans le format donné par 'partie.xml' ; penser d'abord à traiter les pièces 'blanc' puis les pièces 'noir'.

2.2/ Transformation XML ==> HTML

A partir du fichier 'jeu.xml', décrire la transformation XSL qui dessine la configuration du jeu dans une table 9x9 (une ligne et une colonne sont utilisés pour les numéros en plus de 64 cases du jeu).

Dans une case contenant une pièce, on inscrira un code (chaîne de caractères) constitué du nom de la pièce, suivi du caractère '\_' puis du code de couleur ; ex pour la dame blanche : 'D\_B' ; pour simplifier, on ne colore pas les cases ;

Note : pour éviter une boucle complexe, on peut créer une règle pour chaque ligne et l'appeler séquentiellement 8 fois.

Le résultat de la transformation doit donner le rendu HTML ci-dessous :

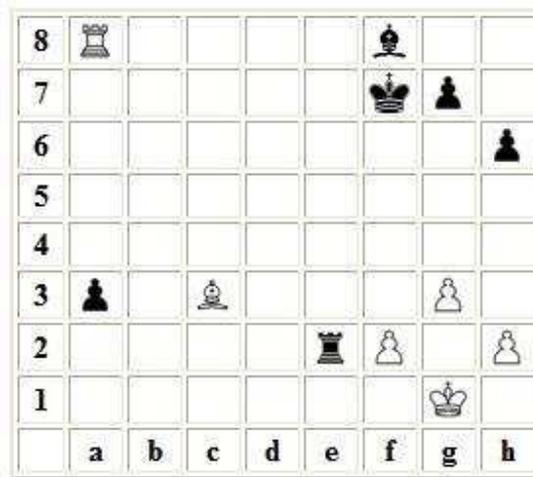
|          |          |          |          |          |          |          |          |          |
|----------|----------|----------|----------|----------|----------|----------|----------|----------|
| <b>8</b> | T_B      |          |          |          |          | F_N      |          |          |
| <b>7</b> |          |          |          |          |          | R_N      | P_N      |          |
| <b>6</b> |          |          |          |          |          |          |          | P_N      |
| <b>5</b> |          |          |          |          |          |          |          |          |
| <b>4</b> |          |          |          |          |          |          |          |          |
| <b>3</b> | P_N      |          | F_B      |          |          |          | P_B      |          |
| <b>2</b> |          |          |          |          | T_N      | P_B      |          | P_B      |
| <b>1</b> |          |          |          |          |          |          | R_B      |          |
|          | <b>a</b> | <b>b</b> | <b>c</b> | <b>d</b> | <b>e</b> | <b>f</b> | <b>g</b> | <b>h</b> |

### 2.3/ Variante avec images

Pour la transformation précédente, on souhaite placer une image dans chaque case non vide, dont le contenu figure dans un fichier au format PNG. Le nom de ce fichier sera composé du même code que dans la question précédente, suivi de l'extension .png ; ex du Fou noir : F\_N.png

Décrire uniquement la règle traitant une case non vide (qui crée la cellule du tableau et insère l'image PNG) ;  
rappel de la balise : `<img src= « URI » >`

Voici la configuration HTML résultant de cette version avec images :



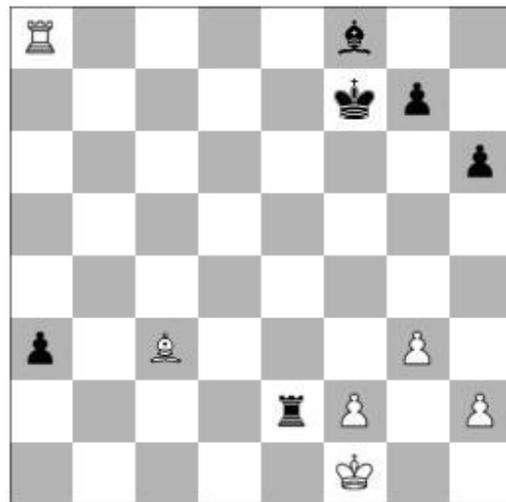
### 3/ Configuration graphique SVG (2 pts)

Décrire le début du contenu du fichier graphique SVG représentant la figure du jeu en damier (image en annexe 2); on ne fera pas figurer les numéros de lignes et colonnes ; décrire la première ligne seulement.

- taille des cases : 32 x 32 pixels
- origine de l'image : 0, 0 dans une fenêtre SVG
- couleurs de cases : blanc : #FFFFFF ; gris : #A0A0A0

L'image de la pièce contenue dans une case non vide est le fichier PNG décrit à la question précédente.

Le rendu complet de la partie en SVG donne le résultat graphique ci-joint



Annexe 1 : fichier 'jeu.xml'

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
```

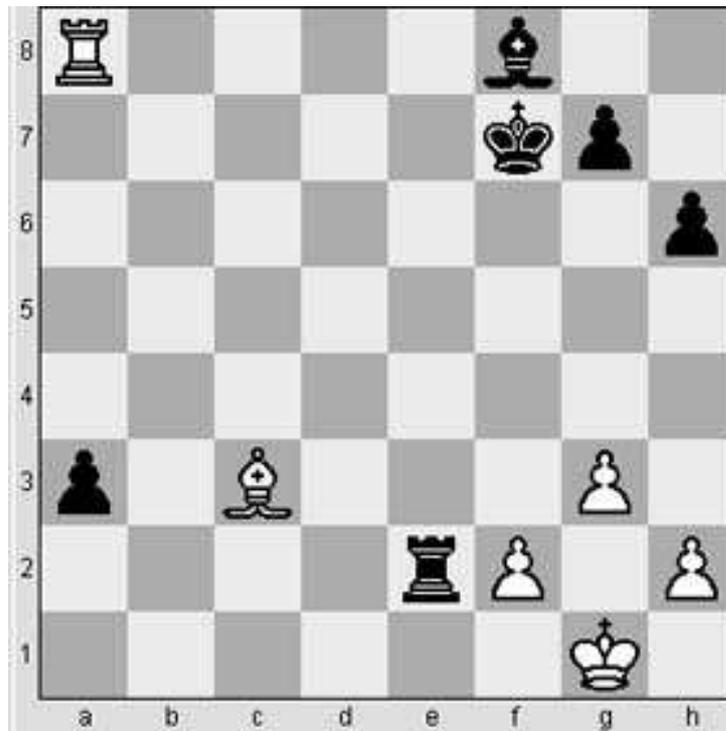
```
= <jeu>
  <case col="a" lin="1" />
  <case col="a" lin="2" />
  = <case col="a" lin="3">
    <piece nom="P"
      couleur="N" />
  </case>
  <case col="a" lin="4" />
  <case col="a" lin="5" />
  <case col="a" lin="6" />
  <case col="a" lin="7" />
  = <case col="a" lin="8">
    <piece nom="T"
      couleur="B" />
  </case>
  <case col="b" lin="1" />
  <case col="b" lin="2" />
  <case col="b" lin="3" />
  <case col="b" lin="4" />
  <case col="b" lin="5" />
  <case col="b" lin="6" />
  <case col="b" lin="7" />
  <case col="b" lin="8" />
  <case col="c" lin="1" />
  <case col="c" lin="2" />
  = <case col="c" lin="3">
    <piece nom="F"
      couleur="B" />
  </case>
  <case col="c" lin="4" />
  <case col="c" lin="5" />
  <case col="c" lin="6" />
  <case col="c" lin="7" />
  <case col="c" lin="8" />
  <case col="d" lin="1" />
  <case col="d" lin="2" />
  <case col="d" lin="3" />
  <case col="d" lin="4" />
  <case col="d" lin="5" />
  <case col="d" lin="6" />
  <case col="d" lin="7" />
  <case col="d" lin="8" />
  <case col="e" lin="1" />
  = <case col="e" lin="2">
    <piece nom="T"
      couleur="N" />
  </case>
  <case col="e" lin="3" />
  <case col="e" lin="4" />
  <case col="e" lin="5" />
```

```

  <case col="e" lin="6" />
  <case col="e" lin="7" />
  <case col="e" lin="8" />
  <case col="f" lin="1" />
  = <case col="f" lin="2">
    <piece nom="P"
      couleur="B" />
  </case>
  <case col="f" lin="3" />
  <case col="f" lin="4" />
  <case col="f" lin="5" />
  <case col="f" lin="6" />
  = <case col="f" lin="7">
    <piece nom="R"
      couleur="N" />
  </case>
  = <case col="f" lin="8">
    <piece nom="F"
      couleur="N" />
  </case>
  = <case col="g" lin="1">
    <piece nom="R"
      couleur="B" />
  </case>
  <case col="g" lin="2" />
  = <case col="g" lin="3">
    <piece nom="P"
      couleur="B" />
  </case>
  <case col="g" lin="4" />
  <case col="g" lin="5" />
  <case col="g" lin="6" />
  = <case col="g" lin="7">
    <piece nom="P"
      couleur="N" />
  </case>
  <case col="g" lin="8" />
  <case col="h" lin="1" />
  = <case col="h" lin="2">
    <piece nom="P"
      couleur="B" />
  </case>
  <case col="h" lin="3" />
  <case col="h" lin="4" />
  <case col="h" lin="5" />
  = <case col="h" lin="6">
    <piece nom="P"
      couleur="N" />
  </case>
  <case col="h" lin="7" />
  <case col="h" lin="8" />
</jeu>
```

Annexe 2 : fichier 'partie.xml'

```
<?xml version="1.0" encoding="ISO-8859-1" ?>  
= <partie>  
  = <blancs>  
    <T col="a" lin="8" />  
    <F col="c" lin="3" />  
    <P col="f" lin="2" />  
    <R col="g" lin="1" />  
    <P col="g" lin="3" />  
    <P col="h" lin="2" />  
  </blancs>  
  = <noirs>  
    <P col="a" lin="3" />  
    <T col="e" lin="2" />  
    <R col="f" lin="7" />  
    <F col="f" lin="8" />  
    <P col="g" lin="7" />  
    <P col="h" lin="6" />  
  </noirs>  
</partie>
```



IN56 – Examen final  
Partie 2  
Documents de TPs autorisés

Rédiger sur une copie séparée

6 points

Un étudiant d'IN56 souhaite réaliser une application web en JSP prenant en charge un vote électronique. Chaque personne enregistrée ne doit pouvoir voter qu'une seule fois (ou voter blanc) pour un candidat. La liste des candidats (nom et prénom) est supposée connue et non modifiable. Un administrateur doit permettre d'ouvrir et de clôturer les votes. Le décompte des votes est consultable après clôture.

Aidez cet étudiant en lui fournissant, conformément au MVC :

- Le diagramme de contexte de son application.
- Le diagramme de navigation (enchaînements de pages).
- Les beans qu'il devra employer: attributs, portée, pages d'appel. Fournissez une description de chaque bean de sorte que l'étudiant puisse rédiger son code source (il n'est pas obligatoire d'indiquer du pseudo-code).

Le langage de modélisation UML est recommandé.