M1 - MIASHS IC Année 2021/2022

Projet « Puissance 4 »

UE Représentation des Données et des Connaissances

Etudiant : Mathis Ruffieux

Sommaire

I) Le schéma XSD

Le schéma XSD permet de représenter les configurations du jeu Puissance 4, en vérifiant la validité des fichiers XML.

Pour cela, on vérifie que chaque balise se situe à l'endroit attendu. La balise racine se nomme *puissance4* et contient une balise *terrain* qui représente le terrain du jeu. Dans cette balise, il y a des balises *ligne* au nombre de 6 représentant les lignes du terrain et dans chaque ligne doivent se trouver des balises *colonne* au nombre de 7.

Pour dénombrer ces lignes et ces colonnes, nous avons donc besoin d'un id sur chacune d'entre elles. Il faut donc que cet attribut soit présent, de type *idType*, représenté par un int compris entre 1 et 7.

Ainsi, chaque balise *colonne* représente une case du jeu par la triangulation de sa position en lignes et en colonnes.
Chacune de ces cases doit contenir une chaîne de caractère représentant le nom d'un joueur (*joueur1*; *joueur2*) ou un élément vide (*vide*), représentés par le type *joueurType*.

- Il n'est pas accepté par le schéma XSD de laisser une balise *colonne* non remplie, si il n'y a pas de jeton à cette position il faut écrire en toute lettres "vide".
- Par ailleurs, il n'est pas accepté d'écrire autre chose que "joueur1" ou "joueur2" pour représenter les joueurs, car le "joueur1" joue toujours le premier coup.

II) Documents XML

Dans le répertoire XML se trouve les fichiers XML utilisés pour la génération de fichiers HTML avec la transformation XSLT.
Avant de transformer ces fichiers, il est nécessaire de vérifier leur validité avec le fichier XSD se trouvant dans le répertoire parent.

Pour représenter tous les types de victoires possibles, nous avons différents fichiers représentants chacun une configuration de jeu :

- La victoire du joueur 1 en diagonale (de haut - gauche à bas - droite)
- La victoire du joueur 1 en ligne
- Un exemple de triche du joueur 1 en jouant trop de jetons par rapport au joueur 2
- La victoire du joueur 2 en colonne
- La victoire du joueur 2 en diagonale (de bas - gauche à haut - droite)

- Un exemple de triche du joueur 2 en jouant un jeton volant (et par la même occasion en jouant plus de jetons que le joueur 1)
- Un terrain rempli sans victoire (match nul)

Nous avons aussi un répertoire *non_valides* avec des exemples de fichiers xml qui ne sont pas validés par notre schéma XSD.

Les exemples qui provoquent une erreur sont :
- des attributs id qui ne sont pas compris entre 1 et 7
- des balises *ligne* et *colonne* sans attribut (alors qu'il devrait y avoir un attribut id sur chacune d'entre elles)
- des noms de joueurs erronés, par exemple "joueur3" n'est pas valide
- des balises *colonne* sans contenu alors qu'il est nécessaire d'y écrire soit le nom d'un joueur, soit le mot "vide"
- la balise parente *terrain* manquante

III) Transformation XSLT

Ici nous avons 2 fichiers XSLT avec la même implémentation xsl.
- Le fichier transform.xslt permet de générer des fichiers html sans mise en forme CSS
- Le fichier transform_style.xslt permet de générer des fichiers html avec une mise en forme CSS et plus de vecteurs SVG.

(La première transformation sert uniquement si l'affichage pose des problèmes.)
L'affichage est adapté au navigateur Firefox.

1) Le fichier XSLT

Le fichier match avec la balise *terrain* du fichier XML.
Ensuite, on parcourt les lignes et les colonnes en créant un tableau avec des balises *tr* et *td*.
Dans cette boucle, le fichier lit le contenu de chaque balise *colonne* :
- si il est écrit 'vide', un cercle contenant la couleur du fond est affiché en svg
- si il est écrit 'joueur1' un cercle de couleur rouge (crimsom)
- si il est écrit 'joueur2' un cercle de couleur jaune (gold)

L'interprétation s'affiche en dessous du terrain.
D'abord nous voulons tester si un joueur est victorieux. Pour cela, nous allons parcourir l'ensemble des balises lignes et colonnes, en stockant leur attribut id respectifs dans une variable *IdLigne* et *IdColonne*.

Les différents types de victoires sont testés :

- en ligne
- en colonne
- en diagonale (de haut - gauche à bas - droite)
- en diagonale (de bas - gauche à haut - droite)

Par exemple pour une victoire en ligne, il faut que l'élément "." (qui représente la chaîne de caractère "joueur1", "joueur2" ou "vide") soit égal à ses trois éléments voisins en ligne.
Si ces quatres occurrences sont égales entre elles, un texte est affiché en fonction du nom du joueur victorieux. (Mais rien n'est affiché si l'élément "." correspond à "vide")

Différents types de triches sont aussi testés :

- Si le joueur 2 à joué plus de jetons que le joueur 1
- Si le joueur 1 à joué deux ou plus de deux jetons que le joueur 2 (car le joueur 1 peut avoir un coup d'avance)
- Si le jeton d'un joueur est volant

Pour cela, on compte le nombre de jetons de chaque joueur que l'on stocke dans des variables *nbJoueur1* et *nbJoueur2*.
Il est donc possible de comparer ces égalités sur ces deux variables.

Pour tester si un jeton est volant on vérifie dans la boucle qui parcourt les lignes et les colonnes si la case à la position en dessous de la position du jeton correspond à "vide", auquel cas un message de triche est affiché.

Par ailleurs, nous voulons aussi tester si une partie est nulle, c'est-à-dire que le terrain est rempli et qu'il n'y a pas de victoire.
Pour cela, on compte le nombre d'occurrences du mot "vide" enregistrées dans une variable *nbVide*. Pour tester que le terrain est rempli, il suffit de vérifier que cette variable vaut 0.
La partie est donc nulle si le terrain est rempli et qu'aucun message de victoire n'est affiché. (Cf. fichier *terrain_plein.html*)

2) Vecteurs SVG et mise en forme CSS

Dans le fichier transform_style.xslt, différents types de vecteurs SVG sont appelés. Nous pouvons trouver dans le répertoire "svg" deux blobs desquels sont extraits leur code svg, ainsi que deux fichiers circles_black.svg et layer1.svg générés à l'aide du site app.haikei.app.

Le fichier style.css se trouve lui dans le répertoire “css”.

Nos fichiers html générés par le fichier transform_style.xslt sont divisés en trois sections :

- une section “top”
- une section “middle”
- une section “bottom”

Dans la section “top” se trouve uniquement des balises de texte “Projet de Puissance 4” et “Mathis Ruffieux - 2021”, superposés à la section middle.

Le premier vecteur SVG appelé est une vague se situant dans la section “middle”. Elle est affichée en haut de cette section grâce au CSS pour servir de fond à l’écriture de la section “top”.

Le second vecteur SVG est un blob lui aussi dans la section “middle”. Il peut s’animer grâce à une fonction javascript du package kute.js, en passant d’un vecteur à une autre. Le code pour générer les deux sprites de ce blob se trouve dans les fichiers blob1.svg et blob2.svg du répertoire “svg”.
Le blob sert à représenter la forme de notre terrain, par dessus lequel sont affichés les différents jetons des joueurs ainsi que les cases vides ayant pour couleur la couleur d'arrière-plan de la page.

S'ensuit la partie interprétation, se trouvant dans la section “bottom”.
Cette section est séparée de la section “middle” avec le code SVG du fichier layer1.svg.

Nous avons un dernier vecteur SVG pour illustrer le fond de l’interprétation, ce sont trois cercles non remplis ayant une épaisseur de la même couleur que la vague séparant les sections “top” et “middle”.