

Projet 6 : XML

Utilisation du XML pour les résultats

Enoncé

Implémenter la vue complète du village, la demande de construction et l'affichage des ressources du joueur.

Ressources

Les ressources suivantes sont disponibles sur le site http://cours.endor.be/2eme_ajax .

- common.js : librairie javascript de base
- img/* : différentes images pour le projet
- map.txt : carte à afficher
- db.sql, db.sqlite : source SQL de la DB et fichier SQLite associé (attention, DB différente de l'exercice précédent)

Ces fichiers de ressources ne doivent pas être modifiés (à l'exception de db.sqlite qui sera modifiée quand on travaillera sur la DB).

Cahier des tâches

JavaScript

Lorsque l'on demande la vue du village, le RPC retournera un fichier XML indiquant l'état de construction des différents bâtiments (absent, construit, en cours). Le Javascript devra récupérer ces informations et les afficher.

Pour les villages du joueur, à côté des bâtiments absents, un bouton permettra d'en demander la construction. Lorsque l'on cliquera sur un ce des boutons, un RPC sera appelé pour en demander la construction et une fois ce dernier exécuté, on rafraichira la vue du village. Dans le cas où un bâtiment peut être présent une seule fois, un seul niveau peut être construit à la fois

Une fonction permettra de demander au serveur les ressources du joueur (nourriture, bois, minerai). Cette fonction mettra à jour un affichage qui sera présent sur toutes les vues (connectées). On appellera cette fonction au login ainsi qu'à chaque rafraichissement de la vue du village.

PHP

Le RPC qui affiche la vue du village sera mis à jour afin de retourner dans un objet XML les informations suivantes : village (à 1 si un village est construit), propriétaire, self (à 1 si le village appartient au joueur, 0 sinon), champ1, champ2, foret1, foret2, mine1,mine2,caserne,forge, archerie, habitation1, habitation2, habitation3. Si le village appartient à un autre joueur, on retournera simplement l'information présent ou absent pour un bâtiment, dans le cas où le bâtiment serait en construction, on le signalerait simplement comme absent.

Le RPC qui demande la construction d'un bâtiment vérifiera d'abord que le bâtiment n'existe pas et n'est pas en cours de construction. Une fois cette vérification faite, on vérifiera que le joueur

dispose d'assez de ressources. Si tout est correct, on déduira les ressources nécessaires à la construction et on marquera le bâtiment comme en cours de construction.

Le RPC qui retourne les ressources du joueur se contentera de retourner les trois valeurs pour le joueur connecté.

Données de jeu

Pour la construction des bâtiments, il sera nécessaire de dépenser une certaine quantité de ressources. Ces quantités sont reprises dans la table ci-dessous.

Bâtiment	Temps de construction	Nourriture	Bois	Minerai
Champ	4	100		
Forêt	4	50	50	
Mine	4		50	50
Caserne	6	200	200	50
Forge	8	200	150	200
Archerie	8	200	300	50
Habitation	6	100	100	

Pour indiquer qu'un bâtiment est en cours de construction, on utilisera une valeur négative qui indiquera le temps restant pour la construction.

Grille d'évaluation

Domaine	Critère	Indicateur	++ + +/- -	Commentaires
Forme	Respect des consignes	Le travail est remis dans les temps (avant le cours suivant)		
		Les contraintes sont respectées		Les consignes de l'énoncé sont respectées
Respect des standards	Code lisible	Le code est correctement indenté		
		Les noms des variables sont explicites et respectent les règles de nommage		
	Standard html et CSS récents respectés ?	La page n'utilise pas les balises ou de propriétés CSS dépréciées ou incorrectes		
Correction	Erreurs	Le programme tourne sans erreurs		Pas de messages d'erreurs ou d'erreurs visibles lors de l'utilisation
	Résultat	Les fonctionnalités implémentées effectuent bien les tâches demandées		Le résultat est bien celui attendu (pas de résultat incorrect)
	Complet	Toutes les fonctionnalités demandées sont présentes		
	Portabilité	Le programme peut tourner sans problème sur tous les navigateurs récents		
		Les chemins vers les ressources sont exprimés de façon relative et correspondent à la norme web		
Qualité de la programmation	Organisation	Les codes Javascript et les feuilles de styles sont dans des fichiers séparés		
		Une arborescence logique est utilisée		
	Code propre	Pas de répétition inutile de blocs de codes, de conditions, ...		
		Pas de variables ou paramètres inutiles		
		Les extensions des éléments statiques correspondent à leur contenu. Les extensions standards sont utilisées		
	Elégance	Les algorithmes utilisés sont efficaces et bien choisis		
Design	Affichages corrects	Les affichages sont corrects, disposés de façon logique		
		Absence d'artefacts de rendu (espacements, bordures,... dont la présence n'est pas voulue et nuit à l'affichage)		
Documentation	RPC Documentés	Les RPC AJAX sont documentés (noms, entrées, sorties, utilités, effets de bord, ...)		Prévoir un fichier TXT qui décrit les RPC : nom du fichier PHP, utilité (nom long), paramètres en entrée, valeurs en sortie, détails (ce que le RPC fait, pas comment il le fait)

