

AJAX - Enoncé 6

D.Moreaux

25 novembre 2024

Résumé

Créer les éditeurs d'actions et de status en partageant une bonne partie du code entre les deux volets (la seule différence est la valeur de `tbl`)

On évitera de faire un refresh à chaque opération en modifiant la version côté navigateur directement lors de chaque opération.

1 La structure d'un code

Les codes seront sauvés dans un tableau sérialisé. Ce tableau contiendra deux branches : les conditions et les instructions.

Chaque branche contiendra un ensemble d'entrées. Pour les conditions, l'ordre a peu d'importance (elles doivent toutes être validées) par contre, pour les instructions, l'ordre est important.

Chaque entrée contiendra un numéro d'instruction ou de condition et de 0 à 2 paramètres qui peuvent être une pièce, un objet, un message ou une valeur numérique (dans tous les cas, ce sera un entier).

On essaiera de réutiliser le même code dans les différents cas de figure. Pour cela, on disposera de deux tableaux contenant les détails (numéro, noms, paramètres) des conditions et des instructions.

1.1 Les conditions

N	Cond	par1	par2	description
0	TRUE			Condition par défaut, toujours vraie
1	AT	room		La position est la pièce indiquée
2	NOTAT	room		La position n'est pas la pièce indiquée

3	ATGT	room		La position est une pièce après la pièce indiquée
4	ATLT	room		La position est une pièce avant la pièce indiquée
5	PRESENT	objet		L'objet est porté, transporté ou dans la pièce courante
6	ABSENT	objet		L'objet n'est ni porté, ni transporté, ni dans la pièce courante
7	WORN	objet		L'objet est porté
8	NOTWORN	objet		L'objet n'est pas porté
9	CARRIED	objet		l'objet est transporté (mais pas porté)
10	NOTCARR	objet		L'objet n'est pas transporté
11	CHANCE	1-100		Cette condition a un certain pourcentage de chance d'être validée
12	ZERO	flag ¹		Le flag est nul
13	NOTZERO	flag		Le flag est non nul
14	EQ	flag	valeur	Le flag est égal à la valeur
15	GT	flag	valeur	Le flag est supérieur à la valeur
16	LT	flag	valeur	Le flag est inférieur à la valeur

Les informations pourront être stockées au niveau du Javascript en utilisant un tableau à 3 entrées : le nom de la condition, le type du premier paramètre et le type du second paramètre.

1.2 Les instructions

N	Instr	par1	par2	description
0	NOP			No opération (ne fait rien)
1	INVENTORY			Liste l'équipement de l'utilisateur
2	DESCRIBE			Décrit la pièce courante
3	QUIT			Demande si on désire quitter
4	END			Demande si on désire recommencer une partie
5	DONE			Arrête le traitement de la table en cours

1. Le numéro de flag est un entier supérieur ou égal à 0. Certaines valeurs ont une signification spéciale

6	OK			Affiche OK puis DONE
7	ANYKEY			Attend l'appui d'une touche
8	SAVE			Sauve la partie
9	LOAD			Charge une partie
10	TURNS			Affiche le nombre de tours
11	SCORE			Affiche le score
12	CLS			Efface l'écran
13	DROPALL			Dépose tous les objets portés/transportés
14	AUTOGET			GET de l'objet dans word2
15	AUTODROP			DROP de l'objet dans word2
16	AUTOWEAR			WEAR de l'objet dans word2
17	AUTOREM			REMOVE de l'objet dans word2
18	GOTO	room		Se déplace dans room
19	REMOVE	objet		Objet de worn à carried
20	GET	objet		Objet de pièce courante à carried
21	WEAR	objet		Objet de carried à worn
22	DROP	objet		Objet de carried à pièce courante
23	CREATE	objet		Objet mis à la pièce courante
24	DESTROY	objet		Objet mis en not created
25	PLACE	objet	room	L'objet est mis dans la pièce indiquée
26	SWAP	objet1	objet2	Les positions des objets 1 et 2 sont échangées
27	PAUSE	durée		Pause de durée millisecondes
28	MESSAGE	msg		Affiche le message
29	SET	flag ²		Met le flag à 255
30	CLEAR	flag		Met le flag à 0
31	PLUS	flag	valeur	Ajoute un nombre à un flag
32	MINUS	flag	valeur	Retire un nombre à un flag
33	LET	flag	valeur	Donne une valeur à un flag

Cette liste d'action pourra être étendue plus tard (par exemple, si on décide d'implémenter des sons, des images, des changements de

2. Par défaut, on dispose d'une trentaine de flags mais il suffit d'utiliser un numéro de flag pour le créer

couleur, . . .). Il faudra donc s'assurer que le code puisse être facilement adapté (tout en restant dans la configuration action + 0/1/2 paramètres)

2 Le code

Au niveau PHP, on définira les fonctions suivantes :

- Lister les actions. Cette fonction retournera également les listes de pièces, objets, messages et verbes
- Effacer une action.
- Sauver une action. Si l'id est absent (ou a une valeur spécifique), crée une nouvelle entrée sinon, remplace l'entrée existante.

Lors du listing, on triera les entrées en fonction des mots utilisés puis de l'ordre de création. Les id des actions ne seront jamais montrées à l'utilisateur. Cela ne pose dès lors pas de problème d'effacer une action en laissant son id inutilisé.

Au niveau Javascript, la fonction qui affiche les actions affichera les actions les unes après les autres avec des icônes éditer et effacer et proposera un bouton nouveau.

Un seul écran gèrera la création et l'édition (dans le premier cas, on part d'un formulaire vide, dans le second cas, l'écran sera pré-rempli).

Il permettra de choisir deux mot (on ajoutera une entrée *vide* pour le cas où le mot n'aura pas d'importance) et affichera les listes de conditions et d'actions.

Les conditions et actions seront affichées sous la forme d'une liste déroulante *statique* qui contiendra les opérations possibles et de deux zones où pourront apparaître soit un champ texte pour entrer une valeur numérique, soit une liste déroulante des pièces, les objets, les messages ou les mots. Les zones inutiles seront masquées.

Un bouton sauver permettra d'envoyer la totalité de l'action au serveur.

Pour les listes de conditions et d'instructions, prévoir une icône pour effacer une ligne et un bouton pour ajouter une ligne. Les changements seront gérés par le Javascript jusqu'au moment où on cliquera sur enregistrer.