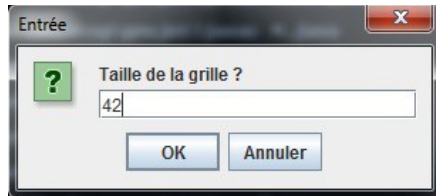


Projet Java : Placement de mots



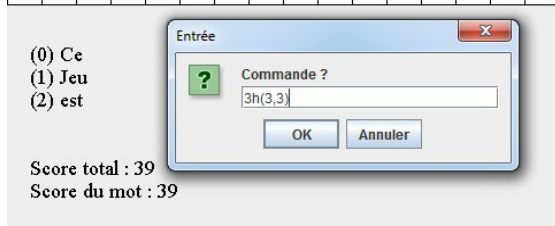
Au début le programme demande la taille du jeu il affiche une grille de la taille demandée avec des nombres aléatoires à l'intérieur.

4	8	8	8	9	4	3	9	4	9	1	2	0	4	2	0
0	6	2	0	5	6	9	5	7	6	4	8	1	4	4	2
3	2	1	2	1	6	8	2	7	1	1	8	6	4	9	5
2	3	5	m	a	g	n	i	f	i	q	u	e	2	0	0
9	9	5	9	0	1	7	6	6	8	3	0	4	8	1	1
4	7	9	2	4	9	7	1	5	8	5	8	1	6	4	7
5	4	2	9	3	0	4	0	4	1	4	2	4	1	3	9
1	5	8	9	1	1	3	3	4	4	6	9	7	9	8	4
9	5	0	3	5	2	4	2	2	4	4	5	5	3	2	5
5	5	2	0	0	7	5	2	3	4	9	9	0	9	8	1
1	0	0	0	6	2	8	0	0	9	3	1	5	8	4	9
8	0	8	5	8	1	5	5	9	0	7	6	1	9	7	3
4	0	6	5	7	0	7	5	2	7	1	8	6	7	4	3
0	4	5	5	8	3	5	0	3	5	6	1	6	8	5	2
6	8	5	8	6	8	3	1	4	7	2	4	4	2	8	1
1	8	7	3	6	3	5	5	9	7	5	5	8	3	4	0

Puis il affiche une liste de mots écrits dans un fichier texte en ne gardant que les mots plaçables et un nombre maximal de 30.

Il demande à l'utilisateur de rentrer une commande du type « 0v(1,0) » pour que le joueur choisisse où il veut placer un mot. Puis il vérifie la conformité de la commande avant d'interpréter la commande.

Il vérifie que le mot ne soit pas déjà placé, qu'il ne croise pas un autre mot et qu'il rentre dans la grille.



Si le mot peut être placé, il supprime alors le mot de la liste et ajoute les points du mot au score.

Le programme s'assure alors qu'il reste au moins un tour au joueur avant de demander à nouveau une commande au joueur qui peut décider de quitter le jeu facilement.

Choix de programmation

```
JOptionPane.showMessageDialog(null, "Entier attendu", "alerte", JOptionPane.ERROR_MESSAGE);
```

Nous avons choisi d'utiliser les messages d'erreurs Swing car nous utilisons des JPanel. Nous avons utilisés pour écrire.

```
SwingUtilities.updateComponentTreeUI(f);
```

Nous avons été confrontés au problème de la méthode repaint() qui n'actualisait pas la liste.

```
cases[latitude][longitude] = mot.substring(i, i + 1);
```

Pour rentrer les mots dans la grille, nous avons changé le type de notre tableau à deux dimensions en chaîne de caractère. Ainsi, on pouvait afficher la grille sans faire de distinction.

```
Pattern pattern = Pattern.compile("([0-9]+)([hHvV])\\((([0-9]+),([0-9]+)\\)");  
Matcher matcher = pattern.matcher(commande);
```

Nous avons utilisé cette RegExp pour récupérer les 4 paramètres séparément ;