

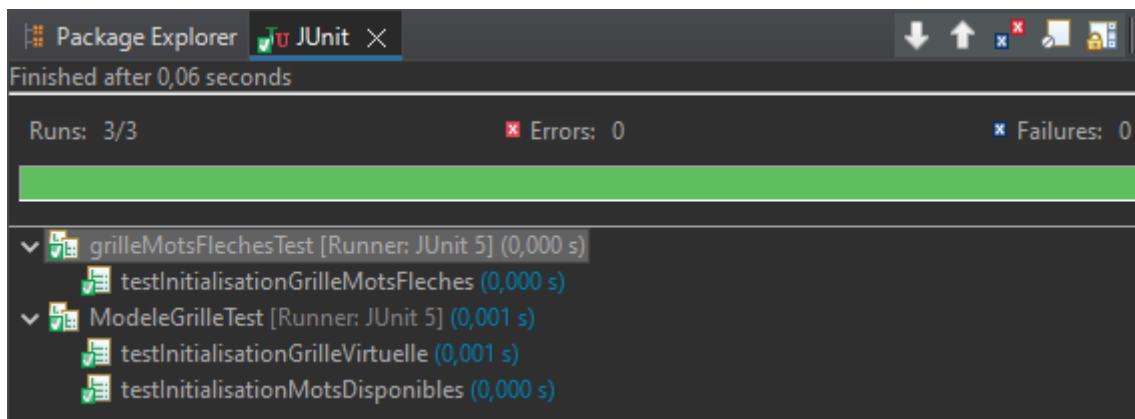


Résultat Tests Unitaires

Projet JAVA | Mot Fléchés

Yohan Davion

Récapitulatif de l'exécution des tests unitaires



1. `testInitialisationGrilleMotsFleches()` :

- Dans cette méthode de test je vérifie si la grille est correctement initialisée avec les dimensions spécifiées (5 lignes et 5 colonnes sont les dimensions donnée pour les tests Unitaires) et si les mots disponibles ont été correctement chargés depuis le fichier.
- Pour cela, nous utilisons des assertions (`assertEquals`) pour comparer les valeurs attendues avec celles réellement retournées par la classe `GrilleMotsFleches`, dans le cas où la valeur reçue est différente de celle attendue alors une `AssertException` est levée.

2. `testInitialisationMotsDisponibles()` :

- Dans cette méthode je vérifie que la liste des mots disponibles est correctement initialisée lorsque nous créons une instance de `ModeleGrille`.
- Dans la méthode `setUp()`, nous créons une instance de `ModeleGrille` avec des dimensions spécifiées (5 lignes et 5 colonnes).
- Ensuite, dans la méthode de test `testInitialisationMotsDisponibles()`, j'appelle la méthode `getMotsDisponibles()` de l'objet `modele` pour obtenir la liste des mots disponibles.
- L'assertion `assertEquals(0, motsDisponibles.size())` vérifie que la taille de la liste des mots disponibles est égale à zéro. Cela signifie que la liste des mots disponibles est vide, ce qui est le comportement attendu lors de la création d'une nouvelle grille de mots fléchés.

Classe de Test « ModeleGrilleTest » :

TEST OK

```
1 package model;
2
3 import org.junit.Before;
4
5 public class ModeleGrilleTest {
6     private ModeleGrille modele;
7
8     @Before
9     public void setUp() {
10         int n = 5;
11         int m = 5;
12         modele = new ModeleGrille(n, m);
13     }
14
15     @Test
16     public void testInitialisationMotsDisponibles() {
17         List<String> motsDisponibles = modele.getMotsDisponibles();
18         assertEquals(0, motsDisponibles.size());
19     }
20
21     @Test
22     public void testInitialisationGrilleVirtuelle() {
23         char[][] grilleVirtuelle = modele.getGrilleVirtuelle();
24         assertEquals(5, grilleVirtuelle.length);
25         assertEquals(5, grilleVirtuelle[0].length);
26     }
27 }
28
29
30
31
32
33
34
35
```

Classe de Test « grilleMotsFlechesTest » :

TEST OK

```
1 package util;
2
3 import org.junit.Before;
4
5
6
7
8 public class grilleMotsFlechesTest {
9
10     private GrilleMotsFleches grilleMotsFleches;
11
12     @Before
13     public void setUp() {
14         int n = 5;
15         int m = 5;
16         grilleMotsFleches = new GrilleMotsFleches(n, m, "liste_français.txt");
17     }
18
19     @Test
20     public void testInitialisationGrilleMotsFleches() {
21         assertEquals(5, grilleMotsFleches.getNbLignes());
22         assertEquals(5, grilleMotsFleches.getNbColonnes());
23
24         assertNotNull(grilleMotsFleches.getMotsDisponibles());
25
26         char[][] grille = grilleMotsFleches.getGrille();
27         for (int i = 0; i < grilleMotsFleches.getNbLignes(); i++) {
28             for (int j = 0; j < grilleMotsFleches.getNbColonnes(); j++) {
29                 assertEquals(' ', grille[i][j]);
30             }
31         }
32     }
33 }
34
```