Fiche annexe sur les fichiers de tests.

Chaque test doit être écrit dans un **fichier séparé**.
On rangera ces fichiers dans une **arborescence spécifique**:

```
test/test_unitaire_ok/test_unitaire_ok_01.txt
                      test_unitaire_ok_01_sortie.txt
                     (test_unitaire_ok_01_entree.txt)
                      test_unitaire_ok_02.txt
                      test_unitaire_ok_02_sortie.txt
                     (test_unitaire_ok_02_entree.txt)
                      test_unitaire_ok_03.txt
                      test_unitaire_ok_03_sortie.txt
                     (test_unitaire_ok_03_entree.txt)
                      ...
    /test_unitaire_ko/test_unitaire_ko_01.txt
                      test_unitaire_ko_01_sortie.txt
                     (test_unitaire_ok_01_entree.txt)
                      test_unitaire_ko_02.txt
                      test_unitaire_ko_02_sortie.txt
                     (test_unitaire_ok_02_entree.txt)
                      test_unitaire_ko_03.txt
                      test_unitaire_ko_03_sortie.txt
                     (test_unitaire_ok_03_entree.txt)
                      ...
    /test_integration/test_integration_01.txt
                      (test_integration_01_entree.txt)
                      test_integration_01_sortie.txt
                      test_integration_02.txt
                      (test_integration_02_entree.txt)
                      test_integration_02_sortie.txt
                      test_integration_03.txt
                      (test_integration_03_entree.txt)
                      test_integration_03_sortie.txt
```

Tests OK:
Les fichiers de **`test_unitaires_ok`** sont des tests de **déplacement valides** qui doivent aboutir à un déplacement d'une pièce.

Ils possèdent la structure donnée en exemple.

```
#test_unitaire_ok_01
#
#NOM1: ZZZZ
#PRENOM1: ZZZZ
#NOM2: ZZZZ
#PRENOM2: ZZZZ
#GROUPE: ZZ
```

```
#
# Ce test permet de verifier qu'un pion puisse se deplacer en ligne droite d'une
case lorsqu'il n'y a pas d'obstacles.
#
i
d 1 1 -> 1 2
fin
#Comportement attendu:
# Pion initialement place en (1,1) est deplace en (1,2)
```

Le test précise en commentaire:

- le nom des auteurs,
- le but du test,
- les lignes de contrôles effectivement envoyées,
- le comportement attendu.

Tests KO:

Les tests du répertoire **`test_unitaire_ko`** doivent tester des déplacements ou des entrées textuelles invalides.
Testez bien chaque **cas particulier** que vous devrez gérer au niveau du code.

Lorsque vous coderez les fonctionnalités, vous devrez retrouver au moins un fichier de test correspondant à chaque cas particulier du code.
Vous pouvez vous inspirer des fichiers d'exemples fournis.

Pour aider à coder vos fonctionnalités, vous commenterez également le **type de message d'erreur attendu**, ainsi qu'à quel **niveau** cette erreur devra être détectée (parseur, API, déplacement propre à une pièce, etc.)

Tests d’intégrations:

Les tests d'intégrations vérifient des **combinaisons** de déplacements valides plus complexes afin de permettre de vérifier que le système global s'articule correctement.

La différence avec le `test_unitaire_ok` provient du fait que l'on ne vise pas à tester une unique fonctionnalité mais le **comportement global**.

Vérification automatique:

Tous les fichiers de tests dont le noms est [NOM].txt ayant une correspondance de sortie attendue appelée [NOM]_sortie.txt peuvent être lancés automatiquement et la sortie est comparée à celle attendue.
Le script se lance depuis le repertoire *script_test_automatique/* avec la ligne de commande:

```
$ python run_test.py
```

Dans le principe, ce script réalise les opérations suivantes dans le cas du premier test:

```
#lancement du jeu d'echec et demande du deplacement
../bin/projet_3eti < ../test/test_unitaire_ok/test_unitaire_ok_01.txt

#comparaison entre sortie obtenue et attendue
diff -i -E -b -w -B ../bin/echec_etat.txt
../test/test_unitaire_ok/test_unitaire_ok_01_sortie.txt
```

Ce script réalise cependant ce tests pour tout les fichiers en parcourant la hiérarchie des répertoire.

Note: L'organisation en 3 répertoires de bases: test_ok, test_ko, et test_integration **doit être respectée**.
Par contre, il peut y avoir des sous répertoires dans ceux-ci (par exemple un sous-repertoire par pièce, etc.), de plus les noms des des fichiers peuvent être différents.
ex. test_deplacement_pion_blanc_01.txt sera accepté.