



AVIGNON
UNIVERSITÉ

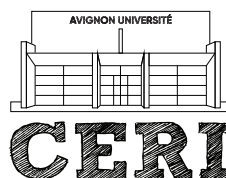
Rapport de test RocketPokemonFactory

20 mai 2022

Master Informatique
UE TECHNIQUES DE TEST

Responsables
Antoine Caubrière

UFR
SCIENCES
TECHNOLOGIES
SANTÉ



CENTRE
D'ENSEIGNEMENT
ET DE RECHERCHE
EN INFORMATIQUE
ceri.univ-avignon.fr

Sommaire

Titre	1
Sommaire	2
1 Tests échoué	3
1.1 PokemonFactoryTest	3
testShouldReturnNullWhenPokedexException	3
testShouldReturnPokemonWithCorrectStats	3
1.2 PokedexTest	3
testShouldReturnNullWhenPokedexException	3
testShouldReturnPokemonWithCorrectStats	4
2 Perspectives d'amélioration	4

1 Tests échoué

Lors du test de l'implémentation de la Team Rocket, plusieurs tests ont échoué. Voici les détails de ce qui n'a pas fonctionné ;

1.1 PokemonFactoryTest

- **testShouldReturnNullWhenPokedexException** : expected null, but was:<fr.uni-vavignon.pokedex.api.Pokemon@2a18f23c>

Ce test échoue car la Team Rocket a défini une partie de ses metadata directement dans RocketPokemonFactory. Un nom d'espèce de pokemon est défini à tort pour l'indice -1.

- **testShouldReturnPokemonWithCorrectStats** : expected:<[Bulbizarre]> but was:<[MISSINGNO]>

Ce test avait été mal conçu et a dû être corrigé.

En mettant en perspective notre test avec la solution de la Team Rocket, nous avons réalisé que notre test était fortement lié à notre propre implémentation de MetadataProvider. En effet, il n'est pas raisonnable de penser que les metadata doivent respecter le même ordre que celui que nous avons choisi puisque cette contrainte n'est spécifiée nulle part. (A moins de vouloir respecter le jeu original mais ce n'est pas le cas).

Pour corriger cette maladresse, notre test fera désormais appel au Pokemon-MetadataProvider pour récupérer les métadonnées correctes.

De plus, la vérification des stats ne prenait pas en compte la limite basse et des metadata inférieure à celle de l'espèce auraient pu ne pas causer d'exception une fois l'erreur ci-dessus corrigée.

Le test a été mis à jour sur la base de ces observations en testant avec les mocks.

En testant à nouveau l'implémentation, le test échoue à nouveau mais pour la bonne raison : le pokemon retourné présente des informations ne correspondant pas à celle de la métadonnée à l'index spécifié.

Le problème vient du fait que la Team Rocket ne se sert pas du PokemonMetadataProvider.

En remplaçant "MISSINGNO" par "Bulbizarre" dans RocketPokemonFactory, le test échoue également car la Team Rocket utilise un générateur aléatoire foireux.

1.2 PokedexTest

- **testShouldReturnNullWhenPokedexException** : expected null, but was:<fr.uni-vavignon.pokedex.api.Pokemon@5af97850>

Ce test échoue pour la même raison que précédemment.

- **testShouldReturnPokemonWithCorrectStats** : : expected:<[Bulbizarre]> but was:<[MISSINGNO]>

Ce test échoue pour la même raison que précédemment.

2 Perspectives d'amélioration

On pourrait envisager de tester si le remplacement du metadataProvider et des metadonnées a bien un impact. C'est à dire vérifier qu'il ne suffit pas de déclarer ses propre métadonnées dans son PokemonFactory.

Par ailleurs, on pourrait vérifier que le temps d'exécution ne dépasse pas une certaine limite au vu du nombre important d'itérations inutiles dans le générateur aléatoire de la TeamRocket.