

Rapport de Techniques de Test

Voici un bref rapport présentant mes remarques sur l'implémentation de IPokemonFactory par la Team Rocket.

Déjà, le principal problème que j'ai eu avec leur code est que les Pokémon qu'ils utilisent pour tester ne sont pas les mêmes que moi. Cela a créé des problèmes puisque je vérifiais notamment que le premier Pokémon était bien un Bulbizarre. Ainsi, avec leurs valeurs, mon test ne passait plus.

Ensuite, ils utilisent la fonction generateRandomStat() qui génère des nombres aléatoires entre 0 et 100. En plus d'être affreusement mal optimisée car elle utilise une boucle for avec un i allant de 0 à un million, il se trouve que cette méthode m'empêche complètement de tester la valeur des caractéristiques de mes Pokémon. Comment tester que l'attaque d'un Pokémon soit égale à une valeur quand cette valeur est aléatoire ? Pour résoudre ce problème, j'ai donc dû changer mes tests, et vérifier un intervalle à la place d'une valeur : je vérifie que l'attaque, la défense et le stamina soient compris entre 0 et 100, et l'IV entre 0 et 1. Ainsi, cela me permet de vérifier mes tests.

Pour finir, la Team Rocket ont oublié de tester le cas où l'index du Pokémon est inférieur à 0 ! J'ai donc dû rajouter un test pour couvrir ce cas-là.

Lors du push en ayant intégré les tests de la Team Rocket, je suis passé de 98% de coverage à 94%. Mais, en rajoutant le test manquant, je suis repassé à 98%.

J'en conclus donc que l'implémentation de la Team Rocket n'est pas spécialement meilleure que la mienne, au contraire. Comme elle génère des valeurs aléatoires, elle rend la phase de test plus compliquée et lourde à mettre en place, en plus de créer des potentielles sources d'erreurs.