

# Rapport de Test et Revue de Code

Projet - TP6 : Giovanni's Touch

December 18, 2024

## Introduction

Ce rapport présente l'analyse de l'implémentation de la classe `RocketPokemonFactory`, fournie par la Team Rocket, selon les critères de qualité du code, des tests automatisés, et des bonnes pratiques. L'objectif est de l'intégrer dans notre projet, d'exécuter notre suite de tests existante, d'identifier les défauts et de proposer des améliorations.

## Résultats des Tests

Après l'intégration de `RocketPokemonFactory`, la couverture de tests a chuté de **90%** à **70%**. Cette diminution est due à plusieurs défauts dans l'implémentation :

### Défauts Identifiés

- **Utilisation d'un `UnmodifiableMap` incomplet** : Le Map `index2name` est initialisé avec seulement trois entrées, ce qui limite la gestion correcte des indices inconnus.
- **Boucle inefficace dans `generateRandomStat()`** : Cette méthode effectue **1 000 000** itérations, ralentissant le programme de manière significative.
- **Gestion incorrecte des indices négatifs** : Pour les indices négatifs, des valeurs arbitraires (`attack`, `defense`, et `stamina` fixés à 1000) sont utilisées, ce qui manque de logique métier.
- **Cas des clés absentes dans `index2name`** : Lorsque l'indice est absent, la valeur par défaut `"MISSINGNO"` est retournée, ce qui n'est pas optimal.
- **Calcul incorrect dans `generateRandomStat()`** : La division par 10 000 produit des résultats incohérents.

# Revue de Code

## Points Positifs

- Respect de la signature de l'interface `IPokemonFactory`.
- Utilisation de `UnmodifiableMap`, garantissant l'immutabilité des données.

## Points Négatifs et Suggestions

- **Initialisation du Map:** Étendre la carte pour inclure tous les indices ou gérer dynamiquement les entrées inconnues.
- **Efficacité de `generateRandomStat()`:** Remplacer par un appel direct à `Random.nextInt()` pour améliorer la performance.
- **Indices négatifs :** Lever une exception ou fournir une logique métier justifiée pour ces cas.
- **Clés absentes :** Lever une exception ou retourner une valeur plus explicite que `"MISSINGNO"`.

## Conclusion

L'implémentation de la Team Rocket présente plusieurs défauts qui impactent la performance et la logique métier. Les recommandations suivantes sont proposées :

- Réduire la complexité de `generateRandomStat()`.
- Étendre ou gérer dynamiquement le Map `index2name`.
- Documenter ou revoir la logique pour les indices négatifs.

Une fois corrigée, cette implémentation pourra être intégrée sans impact négatif sur notre projet.