

Jaremczuk Jonathan

M1 Dev Web

Projet Banc de test

Plan et cas de tests

Liens Github

Repo

https://github.com/OriOn09/test_unitaires

Exécution des tests

https://github.com/OriOn09/test_unitaires/actions/runs/10045550538/job/27763038715

Introduction

Ce document décrit le plan de mise en œuvre des tests pour le projet. Il inclut la description des objectifs de test, les cas de test, les critères de réussite et les détails d'exécution des tests. Les tests sont basés sur le fichier tests.py, qui contient des fonctions pour vérifier les fonctionnalités du script principal.

Objectifs des Tests

1. **Vérifier la validité des pratiques** : Assurer que les pratiques sont correctement évaluées en fonction du niveau et des critères définis.
2. **Calculer correctement les streaks et les vies** : S'assurer que les streaks de pratique et les vies sont calculés correctement selon les règles spécifiées.
3. **Gérer les cas particuliers** : Tester la gestion des jours manqués et la régénération des vies

Environnement de Test

- Configurations Matérielles : Un ordinateur capable d'exécuter des scripts Python.
- Configurations Logicielles :
 - Python 3.10
 - Bibliothèques : pandas, colorama

Cas de Test

Les cas de test sont basés sur les fonctions définies dans le fichier tests.py. Les tests sont organisés pour vérifier différents aspects du code.

Critères de Réussite

Un test est considéré comme réussi si :

- La sortie correspond au résultat attendu.
- Les tests ne génèrent pas d'erreurs non prévues.
- Les fonctions principales (validation de la pratique, calcul des streaks et des vies) se comportent comme prévu dans tous les cas testés.

Exécution des Tests

Les tests sont exécutés via la fonction run_tests() définie dans le fichier tests.py. Chaque test est vérifié par la fonction check, qui affiche un message de succès ou d'échec en fonction des résultats.

Test 1 : Vérifier la pratique valide

Objectif : Vérifier que les pratiques sont considérées comme valides lorsqu'il y a au moins un niveau 2 ou deux niveaux 1.

- **Entrée :**
 - Groupe avec niveaux [1, 1, 2, 2]
- **Résultat attendu :** True

Test 2 : Vérifier la pratique invalide

Objectif : Vérifier que les pratiques sont considérées comme invalides lorsque le nombre de niveaux 1 est insuffisant.

- **Entrée :**
 - Groupe avec niveaux [1, 1]
- **Résultat attendu :** True
- **Entrée :**
 - Groupe avec niveaux [1]
- **Résultat attendu :** False

Test 3 : Vérifier la régénération des vies après cinq jours de pratique

Objectif : Vérifier que les vies sont correctement régénérées après cinq jours de pratique continue.

- **Entrée :**
 - DataFrame avec des pratiques sur plusieurs jours
- **Résultat attendu :** Lives == 2

Test 4 : Vérifier le nombre maximum de vies

Objectif : Vérifier que le nombre de vies ne dépasse pas le maximum après une longue série de jours de pratique.

- **Entrée :**
 - DataFrame avec une série prolongée de pratiques
- **Résultat attendu :** Lives == 2

Test 5 : Vérifier que les streaks continuent avec des pratiques valides

Objectif : Assurer que les streaks augmentent correctement avec des pratiques valides.

- **Entrée :**
 - DataFrame avec des pratiques valides
- **Résultat attendu :** Streak == 1 au début et Streak == 1 à la fin

Test 6 : Vérifier que plusieurs pratiques valides le même jour sont comptées comme une seule pratique

Objectif : Assurer que les pratiques du même jour sont comptées comme une seule.

- **Entrée :**
 - DataFrame avec plusieurs pratiques le même jour
- **Résultat attendu :** Streak == 1

Test 7 : Vérifier les vies après des jours manqués

Objectif : Vérifier que les vies sont correctement réinitialisées après des jours manqués.

- **Entrée :**
 - DataFrame avec des jours manqués
- **Résultat attendu :** Lives == 2

Test 8 : Vérifier les streaks après des jours manqués

Objectif : Vérifier que les streaks sont réinitialisés correctement après des jours manqués.

- **Entrée :**
 - DataFrame avec des jours manqués
- **Résultat attendu :** Streak == 1

Test 9 : Vérifier l'incrémentation des streaks avec des pratiques valides

Objectif : Vérifier que les streaks s'incrémentent correctement avec des pratiques valides consécutives.

- **Entrée :**
 - DataFrame avec des pratiques valides supplémentaires
- **Résultat attendu :** Streak == 2

Test 10 : Vérifier la pratique valide avec des niveaux mixtes

Objectif : Vérifier que les pratiques avec des niveaux mixtes sont considérées comme valides.

- **Entrée :**
 - Groupe avec niveaux [1, 2]
- **Résultat attendu :** True

Test 11 : Vérifier la pratique invalide avec un nombre insuffisant d'exercices de niveau 1

Objectif : Assurer que les pratiques sont invalides lorsqu'il n'y a pas suffisamment d'exercices de niveau 1.

- **Entrée :**
 - Groupe avec niveau [1]
- **Résultat attendu :** False

Test 12 : Vérifier la réinitialisation des vies après la perte de toutes les vies

Objectif : Vérifier que les vies sont réinitialisées après avoir perdu toutes les vies.

- **Entrée :**
 - DataFrame avec des jours de pratique entraînant la perte de toutes les vies
- **Résultat attendu :** Lives == 2

Test 13 : Vérifier la régénération de vie et de reset après une fin de série

Objectif : Vérifier que les streaks et les vies sont correctement réinitialisés après une fin de série de pratique.

- **Entrée :**
 - DataFrame avec une série de pratiques suivie d'une fin
- **Résultat attendu :**
 - Lives == 2
 - Streak == 5

Test 14 : Vérifier que les vies ne dépassent pas le maximum après une longue série de pratiques valides

Objectif : Assurer que les vies ne dépassent pas le maximum après une longue série de pratiques valides.

- **Entrée** :
 - DataFrame avec une série prolongée de pratiques
- **Résultat attendu** : Lives == 2

Critères de Réussite

Un test est considéré comme réussi si :

- La sortie correspond au résultat attendu.
- Les tests ne génèrent pas d'erreurs non prévues.
- Les fonctions principales (validation de la pratique, calcul des streaks et des vies) se comportent comme prévu dans tous les cas testés.