Pierre Feuille Ciseaux

Architecture et Stratégie de test Logiciel

Sujet

Vous allez réaliser un projet logiciel, de type preuve de concept, en plusieurs étapes permettant d'illustrer :

- Architecture logicielle en couches découplées (client / serveur, infrastructure / logique métier / persistance).
- Stratégie de test sur chaque couche et sur plusieurs niveaux (tests unitaires, tests fonctionnels, tests d'intégration, tests end to end).

Etapes

1. Domaine,
2. Persistance,
3. Infrastructure.

Chaque étape de réalisation s'accompagnera d'un travail d'implémentation et de tests. Le projet pourra impliquer du refactoring, d'une étape à l'autre.

Consignes

- Travail à réaliser de façon individuelle,
- Code personnel uniquement,
- Code "vanilla",
- Usage modéré de l'IA,
- Travail à rendre via un dépôt Git distant à *alex@shrp.dev*, si dépôt Git privé, ajouter *shrp777* en tant que collaborateur.

- Renseigner le prénom et le nom de l'auteur du projet dans le fichier README.md placé à la racine.
- Critères d'évaluation : qualité du code, découplage, taux de couverture et pertinence des tests, méthodologie, respect des consignes.

Etape 01 : Couche du Domaine

Dans cette première étape vous allez développer la couche du ***Domaine*** du projet, côté back end. A ce stade, le programme sera exécuté au moyen d'une commande lancée dans le terminal et les données seront stockées uniquement dans la RAM. Il n'y a ni persistance de données ni interface utilisateur.

La couche du ***Domaine*** sera par la suite complétée par les autres couches logicielles (infrastructure et persistance).

L'objectif est de disposer d'un logiciel découplé, architecturé de façon à rendre chaque couche interchangeable et testable indépendamment.

Sujet

Consignes

Implémentez un jeu de *"Pierre, Feuille, Ciseaux"* et les tests unitaires associés, en employant le langage de votre choix (de préférence, JavaScript / TypeScript) et le framework de test de votre choix (de préférence, Jest).

Commencez par rédiger des spécifications fonctionnelles légères, sous forme de Use Case ou de User Stories.
Ces spécifications serviront de base à la rédaction des tests.

Vous pouvez consulter ce lien https://fr.wikipedia.org/wiki/Pierre-papier-ciseaux pour déterminer les règles métier et les cas d'utilisation du programme.

Spécifications générales

Le vainqueur d'une partie est le 1er joueur qui remporte 2 manches. La partie se poursuit tant qu'il n'y a pas de vainqueur.

Votre programme permettra de :

- simuler le choix d'un signe par un joueur de façon aléatoire,
- connaître le résultat d'une manche opposant 2 joueurs fictifs,
- simuler une partie opposant 2 joueurs fictifs :
    - et connaître le vainqueur,
    - et consulter l'historique des manches :
        - signe sélectionné par chaque joueur fictif,
        - résultat de la manche,
        - évolution du score.

Le jeu devra être exécutable par vos tests unitaires et par un point d'entrée du programme (ex : fichier *main*).