



Tests unitaires

Doublures de test

“On ne se mock pas !”



@laryakan

A quoi ça sert les doublures de test ?

À simuler un comportement afin de tester unitairement !

Tester unitairement, c'est s'assurer du fonctionnement d'une partie précise de son programme (unité), indépendamment d'autres composants/unités. C'est pourquoi on simule le reste.

Un test doit aller le plus vite possible. Il doit s'exécuter rapidement et être compréhensible dans sa lecture. Il a vocation à être exécuté de manière récurrente en mettant explicitement en relief le moindre dysfonctionnement.

Les différentes doublures de test

Les doublures “passives”

- **Stub**

- Un objet qui permet de tester un comportement relativement à des valeurs constantes, comme des propriétés statiques par exemple

- **Fake**

- Une classe programmée simplement pour disposer des méthodes appelées au sein du test
- Les méthodes retournent toujours les mêmes résultats, il s'agit souvent d'une simplification à l'extrême de la classe simulée

- **Dummy**

- Elle n'existe que pour être instancié, sans interaction supplémentaire
- Souvent la surcharge d'une classe requise avec des méthodes vides

Les différentes doublures de test

Les doublures “actives”

- **Mock**

- Permet à la fois de simuler un namespace et des méthodes, à la demande, dont on programme le comportement dynamiquement. Usage d'une librairie recommandé !
- L'usage d'un Mock constitue souvent une assertion à part entière, en tant que partie du test unitaire

- **Spy**

- Une classe codée manuellement qui va reproduire le comportement de la classe souhaité en enregistrant les paramètres et les résultats (dans une propriété publique par exemple), afin d'être consulté ultérieurement au sein du test

Le sujet d'aujourd'hui

Les doublures “actives”

L'objectif est de s'assurer que les règles qui vont être énumérées sont respectées (pas forcément de manière exhaustive vu la contrainte de temps).

Ces règles sont en commentaire dans le fichier du contrôleur de jeu.

Récupération du dépôt et installation

Via **GitBash** ou un terminal ayant accès au Web :

git clone https://github.com/laryakan/test-doubles-kata.git

Une fois dans le répertoire du dépôt local, saisissez

composer install

Il ne vous reste plus qu'à ouvrir ce répertoire avec votre éditeur favori.

Vous pouvez lancer une partie avec la commande

php start-game.php

et lancer les tests via la commande

./make-test

(qui est un exécutable bash)

Scénario

Dans le cadre de ce Coding Dojo, le thème étant de une partie de "Jeu de cartes à collectionner" (TGC), le contrôleur jouera le rôle d'une partie entre deux joueurs.

Les règles du jeux sont inspirées de celles de Hearthstone (Simplifiées !)

Démarrage de la partie

Conditions de démarrage de la partie :

- La partie se déroule au tour par tour
- 2 joueurs par partie
- Chaque joueur a 30 points de vie
- Chaque joueur a 1 point de mana sur un maximum de 10
- Chaque joueur possède un deck de 30 cartes de dégâts
- Chaque carte coûte aléatoirement de 0 à 8 point(s) de mana
- Chaque joueur reçoit 3 cartes initiales dans sa main

Déroulement de la partie

Les règles en cours de partie sont :

- Chaque joueur reçoit un point de mana par tour
- Les points de mana sont rechargés à chaque tour
- Chaque joueur tire une carte aléatoire de son deck, à chaque tour
- Chaque joueur peut dépenser jusqu'au maximum de ses points de mana disponible en carte
- Chaque carte inflige son montant de point de mana sous forme de point dégâts à l'adversaire, ce qui lui fait perdre un montant de point de vie équivalent
- Si un joueur a son montant de point de vie ≤ 0 , son adversaire gagne
- Si un joueur ne peut pas jouer car il ne dispose pas d'assez de mana pour jouer une carte, son adversaire joue immédiatement

Conditions spéciales

- Si un joueur dont le deck est vide ne peut pas jouer, il perd 1 point de vie et son adversaire prend la main (pas encore implémenté)
- La main d'un joueur ne peut pas dépasser 5 carte, toute les carte supplémentaire sont retirées du jeu (définitivement)
- Les carte qui coûte 0 de mana n'inflige aucun dégât, elle ne servent à rien

MERCI

Bravo ! Vous avez réussi !

N'hésitez pas à vous resservir de ce Coding Dojo pour vous exercer !