

dojo-poker



StuUngar² (équipe F)

Loïc PALAYER - Edgar BIZEL - Emmanuelle ADOTE - Apoorva APPADOO

Sommaire

1. Déroulement
2. Fonctionnalités
3. Démonstration
4. Confiance
5. Tests
6. Responsabilités

Déroulement

- Outils :
 - Maven
 - Github Actions
 - Conventional Commits
 - Branches et PR
- Cycle de refactoring perpétuel
- Ajout de slices quand nécessaire
 - Exemple : Rejouer

Fonctionnalités

Présentes

- Comparer deux mains
- Afficher la main gagnante et la façon dont elle gagne
- Validation d'entrée
- Possibilité de relancer une partie

Améliorables

- Affichage de la carte ayant permis de gagner : “Victoire avec As”
(dans le cas d'une égalité de premier niveau)

Démonstration

Confiance

Parties en confiance :

- Un coverage de 90 % sur le projet
- Beaucoup de tests

Parties moins en confiance:

- Les 10% qui ne sont pas couverts
 - quelques interactions utilisateur (rejouer)

Pit Test Coverage Report

Project Summary

Number of Classes	Line Coverage	Mutation Coverage	Test Strength
11	91% <div><div></div></div> 270/297	92% <div><div></div></div> 156/169	97% <div><div></div></div> 156/161

Breakdown by Package

Name	Number of Classes	Line Coverage	Mutation Coverage	Test Strength
pokerhand	1	0% <div><div></div></div> 0/4	0% <div><div></div></div> 0/1	0% <div><div></div></div> 0/0
pokerhand.core	8	91% <div><div></div></div> 234/257	92% <div><div></div></div> 131/142	97% <div><div></div></div> 131/135
pokerhand.ui.core	1	100% <div><div></div></div> 28/28	94% <div><div></div></div> 15/16	94% <div><div></div></div> 15/16
utils	1	100% <div><div></div></div> 8/8	100% <div><div></div></div> 10/10	100% <div><div></div></div> 10/10

Report generated by [PIT](#) 1.8.0

Qualité

Parties de bonnes qualité

- Abstraction avec la classe Power
- Implémentation des contrôles
- Implémentation d'une partie
- Formatting cohérent avec spotless

Points perfectibles

- Tests peu lisibles
- Encapsulation parfois mal respectée
- Quelques fonctions documentées avec une documentation pas à jour
- SRP non respecté

Responsabilités

Apoorva

Readme, CI, Maven

Loïc

Implémentation initiale (première slice)

Commun

- Détection de différents types de main et modifications de *Power* appropriées
- Refactoring
- Tests

=== Questions ===



StuUngar² (équipe F)

Loïc PALAYER - Edgar BIZEL - Emmanuelle ADOTE - Apoorva APPADOO



Algorithme
de force

L'algorithme de force

- Carte la plus forte
Exemple : 2, 4, R, 7, 9 devient
Power(1, [R, 9, 7, 4, 2])
- Paire
Exemple : 4, 4, 8, 3, 2 devient
Power(2, [4, 8, 3, 2])
- Double Paire
Exemple : 4, 4, 8, 8, 3 devient
Power (3, [8, 4, 3])
- Brekan
Exemple : 7, 7, 7, 3, D devient
Power (4, [7, D, 3])
- Suite
Exemple : 4, 5, 6, 7, 8 devient
Power (5, [8])
- Couleur
Exemple : 4, 6, V, 8, 1 devient
Power (6, [V, 8, 6, 4, 1])
- Full
Exemple : 3, 3, 8, 8, 8 devient
Power (7, [8, 3])
- Carré
Exemple : 9, 9, 9, 9, 2 devient
Power (8, [9, 2])
- Quinte Flush
Exemple : 7, 8, 9, 10, V devient
Power (9, [V])