

# 7 Wonders

## Projet de développement

Alessandro Pepegna

Loïc Filippi

Meryem Boufalah

Anastasiia Kozlova

Chaeyeon Shim

Université Côte d'Azur

28 Octobre 2020



- Alessandro Pepegna - **Serveur, Statistiques, Stratégie**
- Loïc Filippi - **Moteur de jeu, Effets, Score, Ranking**
- Meryem Boufalah - **Commerce, Sorties textuelles**
- Anastasiia Kozlova - **Commerce, Effets, Âges, Sorties textuelles**
- Chaeyeon Shim - **Tests**

## Fonctionnalités réalisées

Implémentation de toutes les règles de la première version du jeu.

Mise en place d'un moteur de jeu complet.

Robots de jeu.

Statistiques de fin de partie → serveur

+

Moteur de jeu

Effets

Sorties textuelles

-

Serveur

Stratégie

Choix des ressources

# Organisation du code

- 3 modules
  - *java-common*
  - *java-game*
  - *java-server*
- On sépare inventaire et joueur
- Un moteur de jeu
- Les éléments du jeu sont à part
- Un launcher

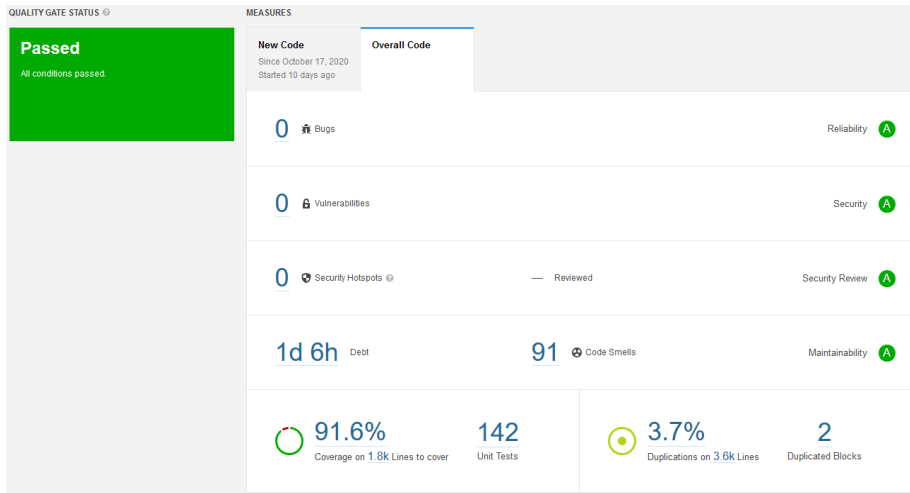
## Strategy

Application du pattern pour la conception des comportements des bots.

Autres patterns potentiels:

- Command → actions des joueurs
- Factory Method → création des cartes

# Analyse par les métriques



- Bonne exploitation de Sonar et de Github

# Points positifs

- Bonne exploitation de Sonar et de Github
- Pattern Strategy bien implementé



- Bonne exploitation de Sonar et de Github
- Pattern Strategy bien implementé
- Utilisation de Cucumber pour les scénarios

# Points positifs

- Bonne exploitation de Sonar et de Github
- Pattern Strategy bien implementé
- Utilisation de Cucumber pour les scénarios
- Responsabilités bien définies

# Points négatifs

- Certaines parties sont peu / pas testées

# Points négatifs

- Certaines parties sont peu / pas testées
- Complexité des tests

# Points négatifs

- Certaines parties sont peu / pas testées
- Complexité des tests
- Le serveur pourrait être plus abouti

# Points négatifs

- Certaines parties sont peu / pas testées
- Complexité des tests
- Le serveur pourrait être plus abouti
- Le module *java-common* est trop grand

# Points négatifs

- Certaines parties sont peu / pas testées
- Complexité des tests
- Le serveur pourrait être plus abouti
- Le module *java-common* est trop grand
- Pas assez de réflexion sur les patterns

## À conserver

- Github flow
- Découpage en US, tâches, sous-tâches
- Répartition des tâches

## À améliorer

- Faire de l'intégration continue sur *develop*
- Découpage des User Stories
- Hébergement de Sonar sur un serveur distant avec scan avec CI
- Lier davantage tests et conception, TDD



# Merci pour votre attention !

