

NOTRE ÉQUIPE AVENTURE



**Jeremy
HIRTH**

Serveur

Statistiques

Tests



**Ralph
EL CHALFOUN**

Structure du jeu

IA

Affichage



**Remi
YACOU**

Mockito

Tests

IA



**Nabil
YACOU**

Logger

Tests

IA

PROJET MASTER

DÉVELOPPER EN JAVA UNE
VERSION NUMERIQUE DU JEU
« SEVEN WONDERS »

BILAN DES FONCTIONNALITÉS

- Bonne couverture des Tests
- Structure Dynamique (Ajout d'éléments facilement possible)

- Manque de tests comportementaux (Cucumber)
- Complexité parfois élevée

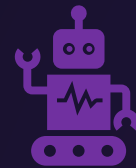


Version numérique du Jeu « Seven Wonders » : tous les éléments et toutes les règles du jeu, serveur et statistiques



Deux possibilités d'exécution du code :

- Déroulement détaillé d'une partie
- Statistiques sur plusieurs parties via le Server

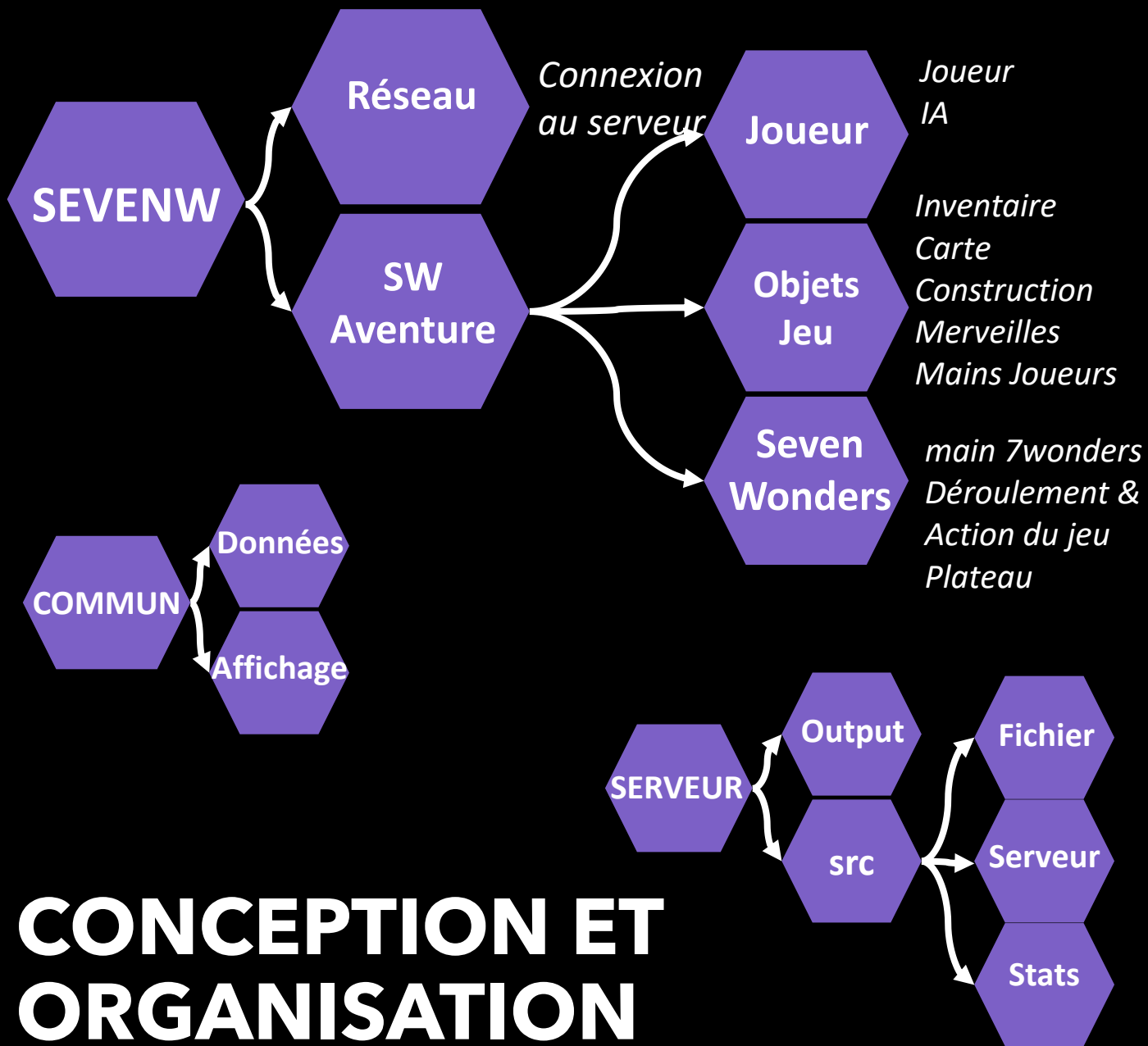


Jouable de 3 à 7 joueurs avec différentes stratégies possibles :

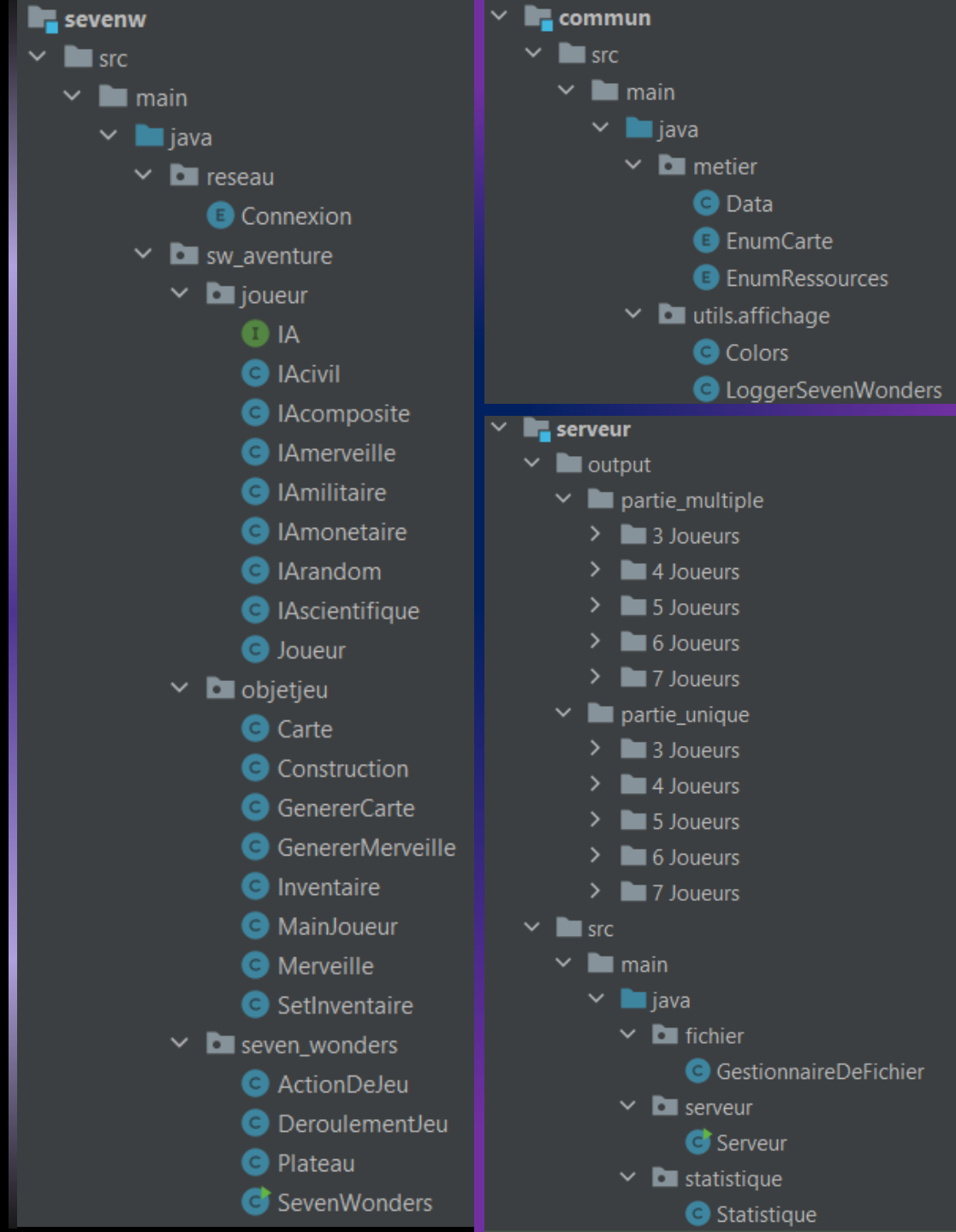
- Stratégie aléatoire
- Stratégies précises
- Stratégie composite



Possibilité d'affichage en couleur d'une partie



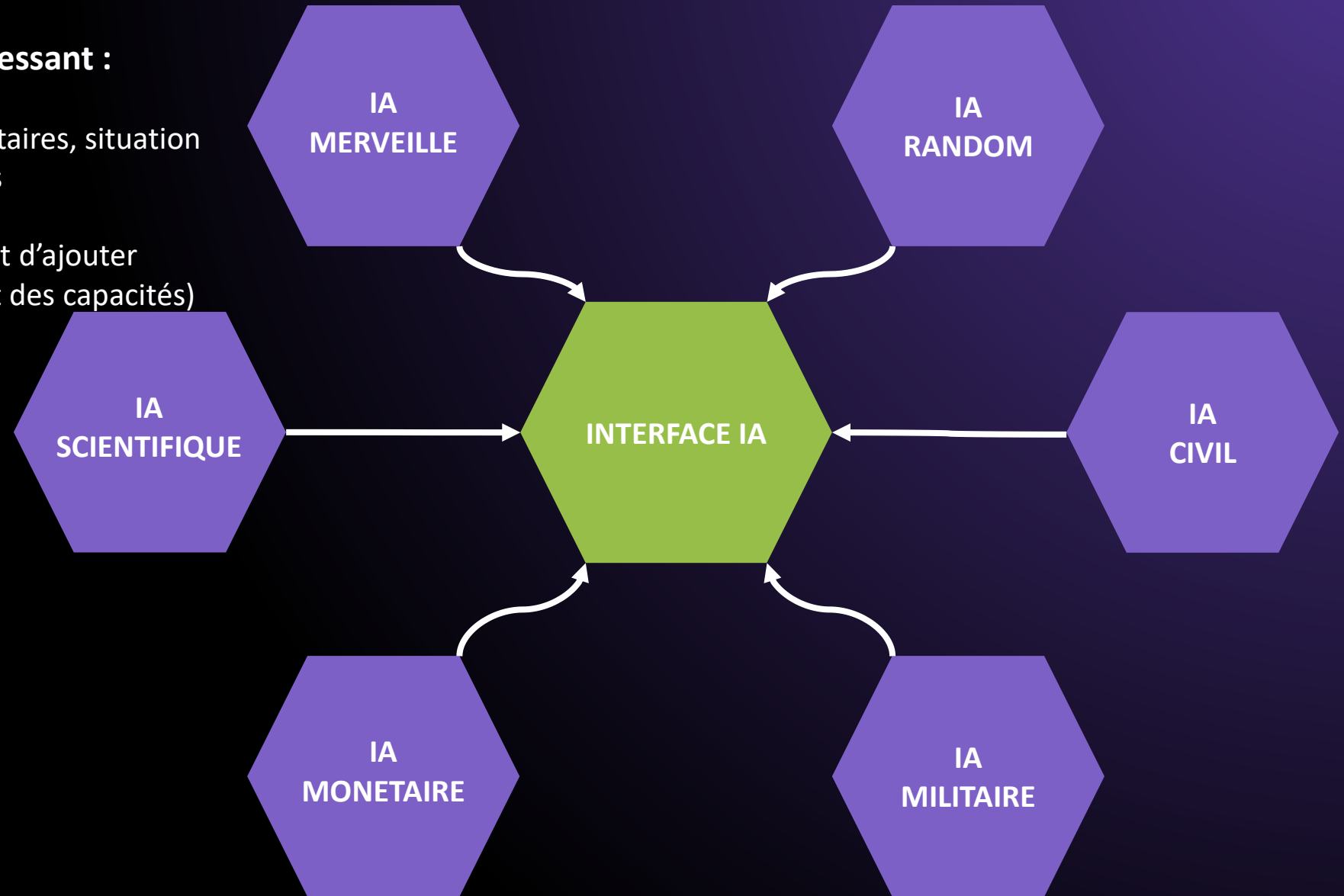
CONCEPTION ET ORGANISATION DU CODE



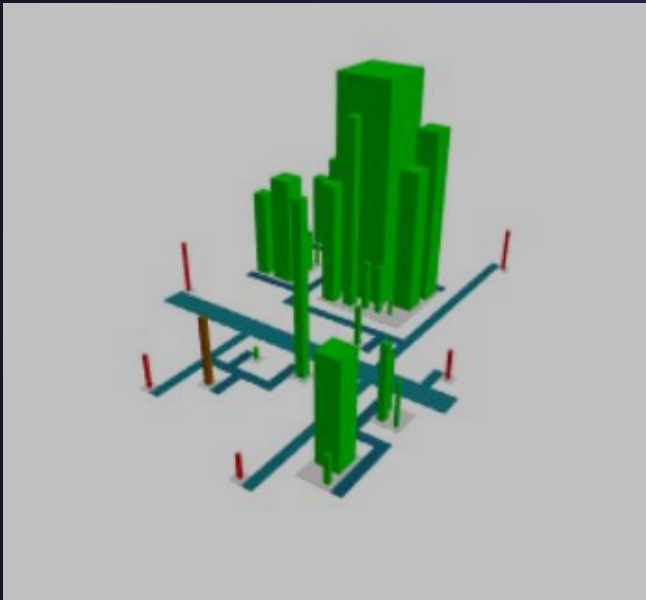
DESIGN PATTERN : Strategy

Autres Design patterns intéressant :

- **Facade** pour Plateau accès aux inventaires, situation de la partie et joueurs
- **Decorator** pour strategy (permet d'ajouter individuellement et dynamiquement des capacités)



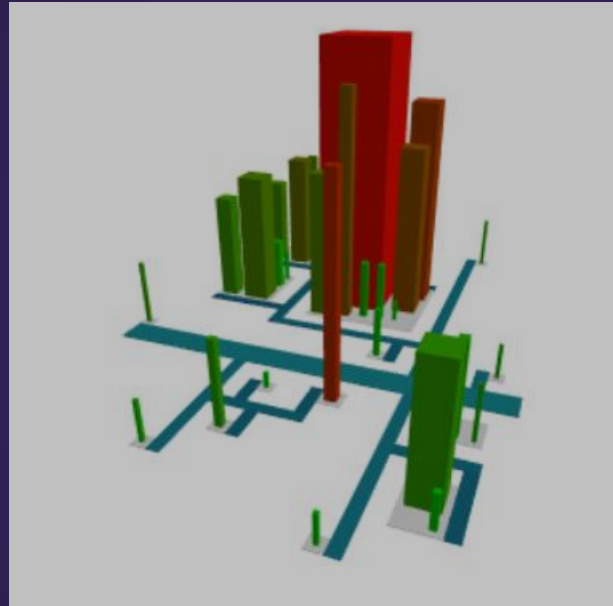
LES METRIQUES



Coverage des Tests

230 tests

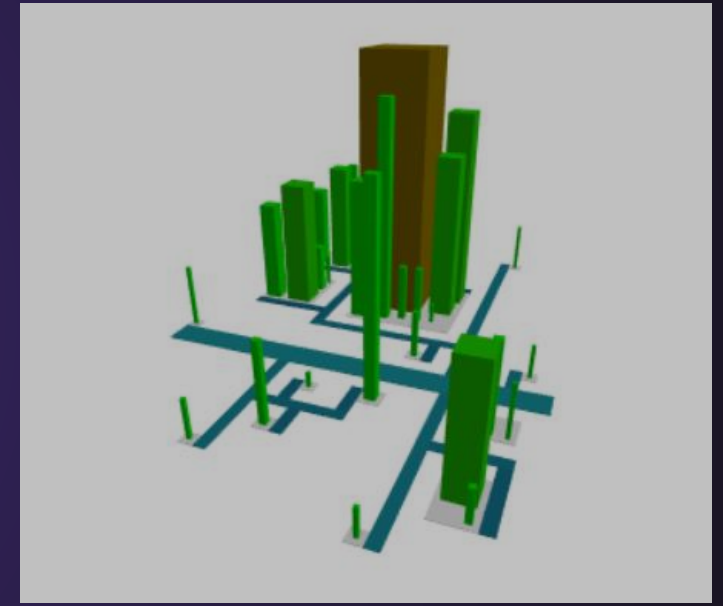
+93%



**Quantité de lignes de
code**

3300 lignes de codes

Construction = 330



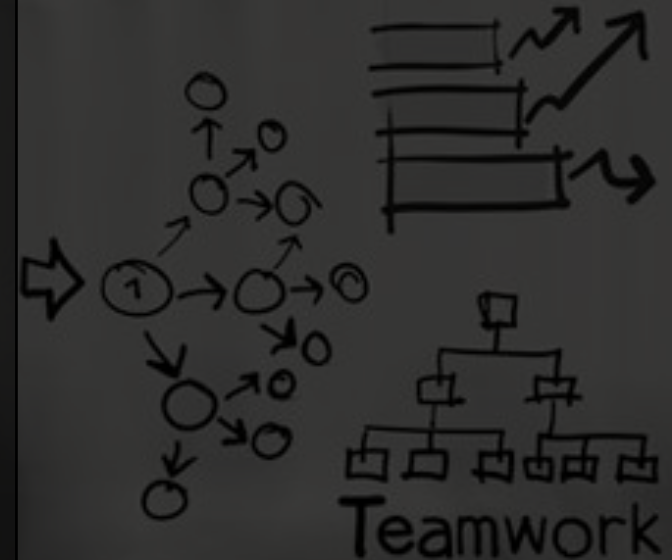
**Complexité
cyclomatique**

**3 méthodes complexes
toutes dans Construction**

POINTS FORTS

POINTS FORTS

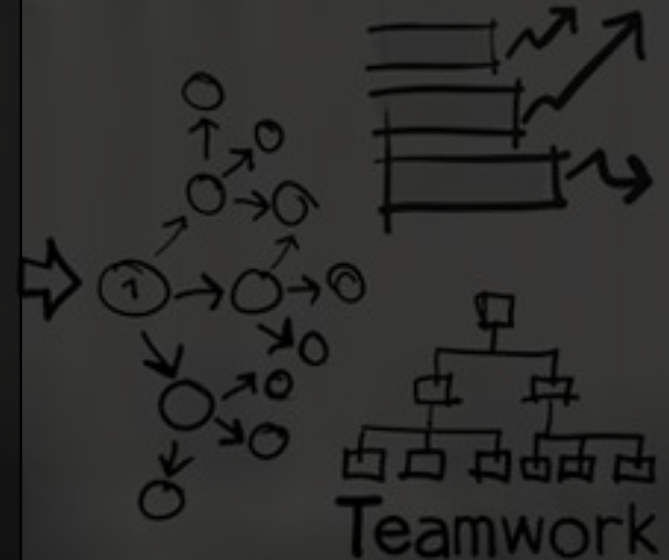
- **Ressemblance fiable du jeu plateau**
 - Tous les éléments et quasi toutes les fonctionnalités du jeu
- **Ajouts/modifications dynamiques**
 - *Interfaces*
 - *Enumérations*
- **Rapport SonarQube satisfait**
 - 5 Codes Smells
 - 0 % de duplication
 - 93 % de coverage
- **Sécurité au niveau du joueur**
 - Il ne peut qu'observer son inventaire et non le modifier
- **Affichage disponible en couleur pour un confort et une lisibilité améliorés**



POINTS FAIBLES

POINTS FAIBLES

- **Classe Construction dense :**
 - *Forte complexité / quantité de lignes de code*
- **Tests unitaires compliqués**
 - *Accumulation de retard des tests à chaque livraison*
 - *Les fonctionnalités ont pris le dessus sur la fiabilité*
- **3 méthodes de complexité élevée**
 - *Beaucoup d'accesseurs entre les classes, elles sont trop dépendantes*
- **Faible diversité des IA**
 - Elles se concentre principalement sur une stratégie par « couleur »



Conserver

SonarQube pour la qualité du code

Le Découpage équilibré

Les Couleurs pour la lisibilité

Le Design Pattern d'interface pour IA

La ponctualité des livraisons

Améliorer - Supprimer

Améliorer la gestion des tests comportementaux

Intégrer au plus vite les Enums pour ne pas repasser sur le code

Privilégier la stabilité à la richesse de fonctionnalité

Trouver d'autres design patterns pour améliorer le code

GESTION DU PROJET, les pratiques

Démonstration...