



Génie Logiciel et Projet de Développement

Juliette Sabatier

Raphael Pietrzak

Dylann Batisse

Margaux Schmied



MASTER
INFORMATIQUE

Implication et expertise

Juliette	Margaux	Dylann	Raphael
Joueurs/Stratégies	Joueurs/Stratégies	Cartes	Cartes
Inventaires	Inventaires	Serveur/Stats	Inventaires
Façade	Dés	GameLoop	Plateau
Tests	Tests/Cucumber	Tests	Tests

Synthèse du projet

▶ Fonctionnalités

- ▶ Le jeu contient 30 cartes utilisables différentes
- ▶ La partie évolue au travers des dés, des cartes et des bonus
- ▶ Chaque joueur a la possibilité d'influencer les autres joueurs et le déroulement du jeu

▶ Confiances

- ▶ Beaucoup de méthodes ne sont testées qu'individuellement
- ▶ Les enchainements de méthodes sont peu testés
- ▶ Les boucles sont peu testées

▶ Qualité de notre code

- ▶ Les responsabilités sont relativement bien réparties
- ▶ Nous pensons avoir bien appliqué des patterns là où il fallait
 - ▶ Sauf pour les Loop qui ont fini par avoir une complexité trop importante
- ▶ Nos méthodes ne sont pas assez documentées
- ▶ Mockito n'est pas assez présent dans nos tests

Organisation du code

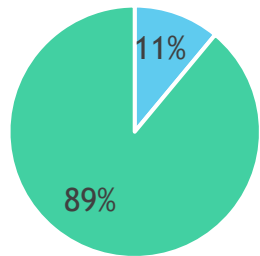
- ▶ Séparation du jeu global et des IA
 - ▶ Les IA ne font que renvoyer leur choix aux questions posées par le jeu
 - ▶ Elles ne peuvent pas modifier les autres classes
- ▶ Tous les éléments nécessaires à faire tourner le jeu sont dans le plateau
 - ▶ Tout est initialisé au même endroit
 - ▶ Le plateau centralise toutes les informations et les fait communiquer entre elles
- ▶ Projet multi-module dans le but de découper chaque partie du projet en plusieurs sous parties
 - ▶ Exemple: séparation du client et du serveur afin de démarrer le serveur indépendamment
- ▶ Une classe permet d'initialiser et de lancer une ou plusieurs parties

Patron de conception

- ▶ Design GRASP SOLID
 - ▶ Répartition des responsabilités de chaque classe
 - ▶ Exemple: séparation en plusieurs blocs de l'inventaire
- ▶ Patterns
 - ▶ GameLoop
 - ▶ Initialise, démarre et fait progresser le jeu
 - ▶ Factory
 - ▶ Facilite la création d'objets multiples
 - ▶ Facade
 - ▶ Permet l'accès à la vitrine d'un objet sans pouvoir le modifier
 - ▶ Strategy
 - ▶ Sélectionne et compose des comportements
 - ▶ NB: la composition sera implémentée durant le TER

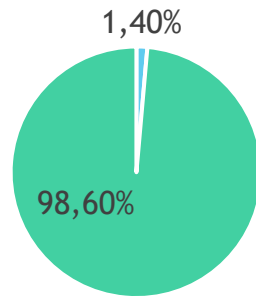
Métrique

Couverture



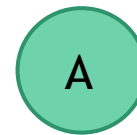
■ Non couvert ■ Couvert

Duplication de code

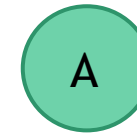


■ Code dupliqué ■ Code non dupliqué

Maintenabilité

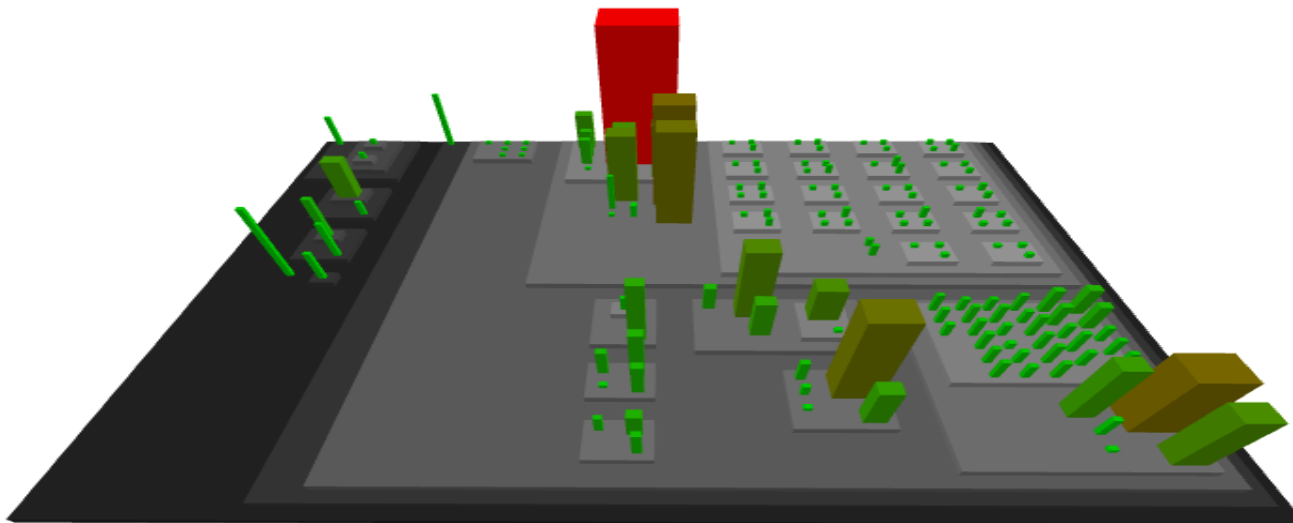


Sécurité



Dette technique

6h07



Pourquoi notre projet est un bon projet?

- ▶ Jouable et complet
- ▶ Fiable grâce aux différents tests (couverture = 89%)
- ▶ Code compréhensible et maintenable
- ▶ Utilisation efficace de design et pattern
- ▶ Bonne cohésion d'équipe
- ▶ Intégration de Git
- ▶ Les livraisons ont été globalement respectées

Pourquoi notre projet est un mauvais projet?

- ▶ Exigences trop élevées de notre part pour les livraisons
- ▶ Pas assez de tests avec cucumber et mockito
- ▶ Répartition non homogène du travail sur la durée
- ▶ Découpage approximatif des tâches
- ▶ Utilisation irrégulière du kanban

Rétrospective

- ▶ À conserver
 - ▶ Cohésion au sein de l'équipe
 - ▶ Git
 - ▶ Patterns
- ▶ À améliorer
 - ▶ Documentation
 - ▶ Cucumber et mockito
 - ▶ Découpage des tâches



MASTER INFORMATIQUE

