

# La Regalata

Groupe TP1 - Rapport final

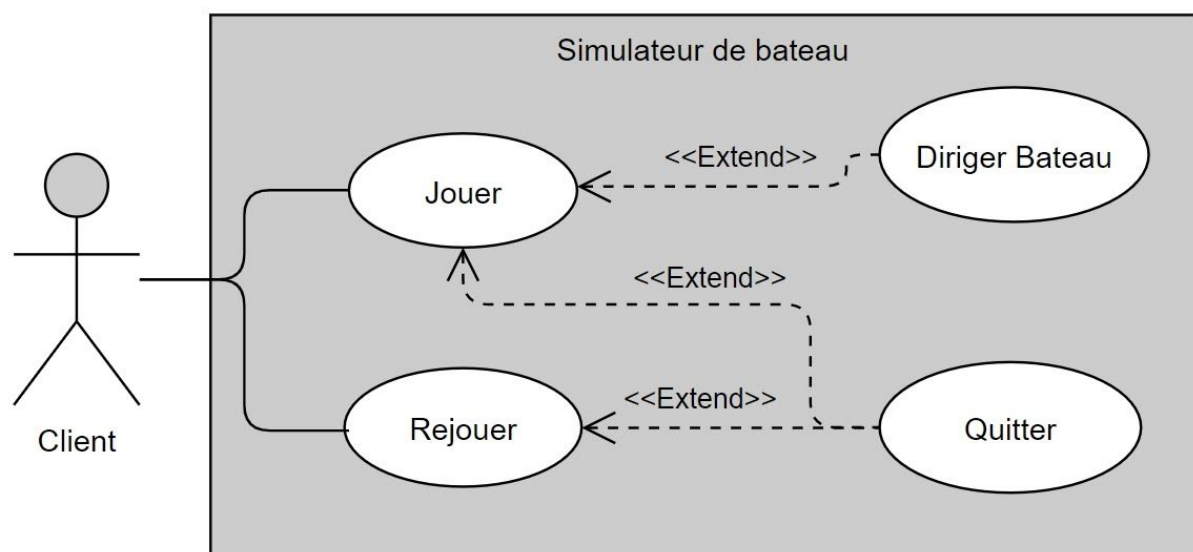
Boris ANODEAU  
Julia ARZUR--DAUNAY  
Gaëtan BARRET  
Camille GUIGNOL  
Geneviève MASIONI  
Nathan MICHEL  
Jérémy POULLAIN  
Guillaume REVEL

## **Analyse des risques :**

Risque prévu	Analyse
Développer une application qui ne correspond pas aux attentes du client.	L'application semble correspondre aux attentes du client. Après quelques retours, nous avons dû effectuer quelques modifications et revoir nos objectifs car ceux-ci étaient trop ambitieux.
Mauvaise intégration des différentes parties à développer.	Cette partie a posé quelques problèmes même si l'intégration a été globalement réussie. En effet, nous avons eu un problème de compatibilité entre différentes parties, mais cela a vite été réglé. Nous avons également eu quelques légères pertes de temps pour fusionner les versions et pour que tout le monde soit à jour, mais celles-ci étaient inévitables, et pour un projet à 8 membres, nous nous sommes plutôt bien débrouillés en communiquant efficacement.
Ne pas rendre les livrables à temps.	Les livrables ont été rendus à temps. Le point avant chaque séance pour situer l'avancement de chacun a été efficace.
Perte de temps due à l'appréhension de nouvelles technologies : JavaFX, JUnit, Gradle, Git.	Finalement, les tâches qui nous ont pris le plus de temps portaient sur le code en lui-même et non sur l'appréhension de nouvelles technologies.

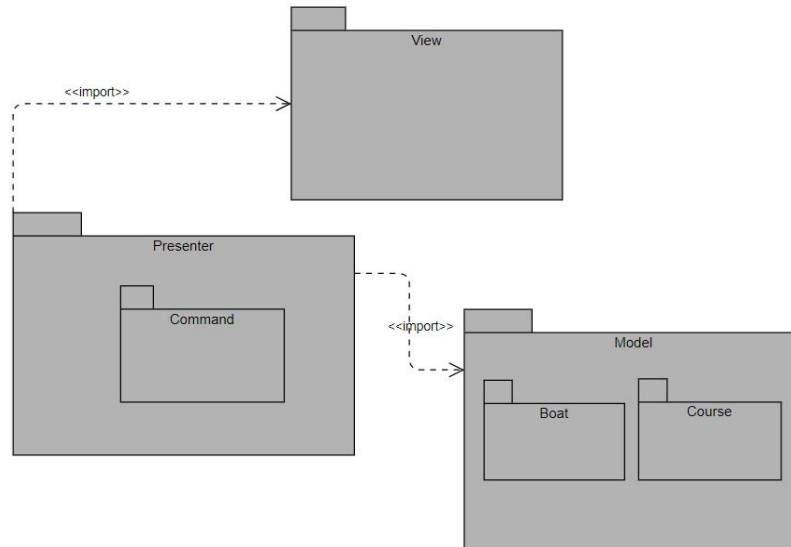
Risque prévu	Analyse
Problèmes relationnels au sein de l'équipe.	Nous n'avons pas eu de problème particulier à ce niveau-là, probablement dû à la bonne communication au sein de l'équipe.
Séances à distance à cause de la situation sanitaire.	Nous n'avons finalement eu que deux séances en présentiel. Néanmoins, nous avons pu communiquer tout au long du projet via Discord. L'habitude de travailler à distance s'est ressentie, et nous n'avons quasiment pas perdu de temps par rapport à des séances en présentiel.
Rendre une application fonctionnelle mais qui ne respecte pas les principes du génie logiciel (les principes SOLID).	L'architecture a été validée au début du projet par tous les membres. Nous avons essayé d'être vigilants sur la propreté de notre code et de celui des autres. Des retours clients suite au MVP 1 nous ont aussi aidé à corriger quelques défauts.
L'effet "démo"	Une branche <i>demo</i> a été mise en place pour la présentation du MVP 1. Pour la démonstration finale, nous avons arrêté de modifier le code une fois la validation effectuée, donc il n'y aura pas d'ajout de dernière minute qui vont perturber la démonstration.

### Diagramme des cas d'utilisation :

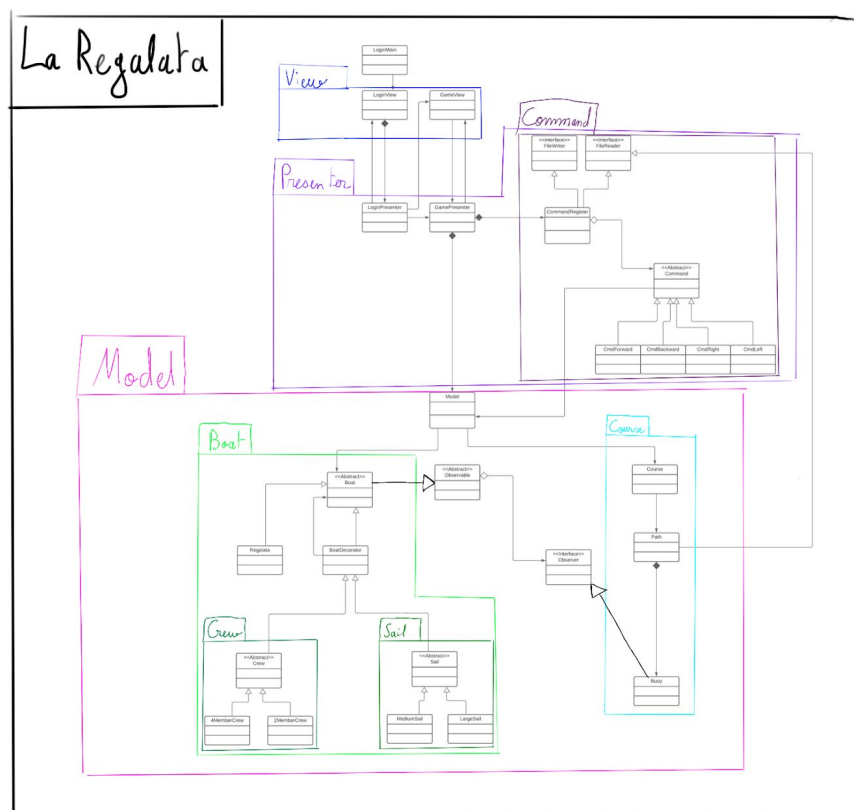


Ce diagramme a dû être révisé : nous n'avons pas implémenté la fonctionnalité permettant de choisir des caractéristiques (taille de voile et membres de l'équipage).

**Diagramme de paquet :**



### Conception UML :



Pour une meilleure lecture, ce diagramme UML est disponible sur le git dans le dossier *files*.

## Liste des fonctionnalités implémentées :

### Fonctionnalités initialement prévues pour le MVP 1 :

- Changer de cap
- Tableau de bord : cap, vitesse de déplacement, force et direction du vent
- Vitesse et direction du vent d'après les conditions météorologiques réelles
- GUI : déplacer le bateau dynamiquement (contrôleur)
- GUI : concevoir le GUI

### Fonctionnalités livrées au MVP 1

### Fonctionnalités finalement reportées au MVP 2

### Fonctionnalités initialement prévues pour le MVP 2 :

- Vitesse et direction du vent d'après les conditions météorologiques réelles
- Calcul du score basé sur le temps de parcours
- Initialisation du jeu par fichier de configuration du plan d'eau et de parcours (prédéfini)
- Choisir la taille (surface) des voiles : grande ou moyenne
- Choisir la taille de l'équipe : 2 ou 4
- Sauvegarde d'une partie pour un bateau seul
- Rejouer automatiquement une partie
- Neutralité du bateau : point de contact avec bouée ou trait de côte.
- Influence des voiles et de l'équipe sur le temps de virement et la vitesse
- Validation des bouées grâce à un signal visuel

### Fonctionnalités du MVP 1 finalement livrées au MVP 2

### Fonctionnalités livrées au MVP 2

### Fonctionnalités finalement abandonnées

### Fonctionnalités non-prévues mais implémentées

Certaines fonctionnalités présentes dans le sujet avaient été abandonnées depuis le rendu du MVP 1, notamment la prise en compte de personnages non-jouables et les pénalités dues au temps de virement.

À l'approche du MVP 2, nous nous étions mis d'accord pour implémenter la fonctionnalité consistant à neutraliser le bateau en cas de contact, mais nous l'avons finalement abandonnée au profit de celle qui permet de sauvegarder et rejouer des parties, car les éléments nécessaires à son implémentation avaient déjà été commencés.

C'est en testant le jeu que nous nous sommes aperçu qu'un signal visuel était nécessaire lors de la validation de chaque bouée, nous avons donc implémenté cette fonctionnalité.

**Liste des tâches correspondant aux fonctionnalités :**

<b>Membre</b>	<b>Rôle</b>	<b>Tâches</b>
Boris ANODEAU	Responsable de version	Gestion des versions Gestion du Git
Julia ARZUR--DAUNAY	Développeur	Classes du package <i>Boat</i> Classe Vector
Gaëtan BARRET	Architecte Développeur	Création de l'architecture Création et utilisation du GUI Classes du Presenter Liens entre Model, View et Presenter Correctifs et aide en général pour le Model
Camille GUIGNOL	Développeur	Classes du package <i>Boat</i> Classe Vector
Geneviève MASIONI	Architecte Développeur	Création de l'architecture Affichage et prise en compte des conditions météorologiques récupérées en JSON
Nathan MICHEL	Développeur	Classe <i>Buoy</i> Écriture du README
Jérémy POULLAIN	Développeur	Classes du package Command Sauvegarde des parties Rejouer les parties Création de la carte avec un fichier texte Refonte de code
Guillaume REVEL	Chef de projet Développeur	Veille à la coordination des tâches Interactions avec le client Rapports et documents Contrôleur bateau Classe <i>Buoy</i>