

# Projet - Bataille navale

## Dossier de conception

---

Julien Romary - Maxime Barbier - Alexis Cesaro - Reynault Sies

## Introduction

Ce document contient les travaux de conception réalisés pour répondre à la spécification du projet de bataille navale de l'UE design pattern. Vous trouverez la liste des différentes fonctionnalités qui ont été recensées, plusieurs diagrammes de classes, plusieurs diagrammes de séquences, ainsi que des prototypages d'interface.

Chaque image fournie est commentée, il y a notamment une justification des choix qui ont été réalisés ainsi que plusieurs précisions lorsque nécessaire.

<b>Introduction</b>	<b>1</b>
<b>Fonctionnalités</b>	<b>3</b>
<b>Diagrammes de classes</b>	<b>4</b>
Diagramme global du modèle de l'application	4
Remarques:	5
Diagramme de classes de la partie Bateaux	6
Remarques:	6
Diagramme de classes des vues	7
Remarques:	7
Diagramme de classes représentant le système de sauvegarde	8
Remarques:	8
<b>Diagrammes de séquences</b>	<b>9</b>
Diagramme de séquences de création de partie	9
Diagramme de séquences de placement d'un bateau	10
Placement d'un bateau déjà existant	11
Diagramme de séquences de fin de placement	12
Diagramme de séquences de fin de tour	13
Remarques:	13
Diagramme de séquences de changement de tactique	15
Remarques:	15
Diagramme de séquences de sauvegarde	16
Remarques:	16
Diagramme de séquences de chargement	17
Remarques:	17
<b>Prototypage de l'interface</b>	<b>18</b>
Menu principal	18
Menus de choix de la tactique et de l'ère	19
Menu de placement des bateaux	20
Menu de jeu	21
Remarques:	22

# Fonctionnalités

## Liste des fonctionnalités

- Jouer une partie

Jouer une partie implique de pouvoir choisir une époque et une tactique de tir pour l'ennemi.

L'utilisateur peut alors placer ses différents bateaux sur sa grille et demander à l'application de démarrer le jeu.

Lorsque le jeu démarre, le joueur joue en premier et demande à l'un de ses bateaux de tirer sur une case de l'ennemi, puis termine son tour. L'ennemi va alors utiliser sa tactique pour choisir une case du joueur à attaquer.

- Sauvegarder / Charger

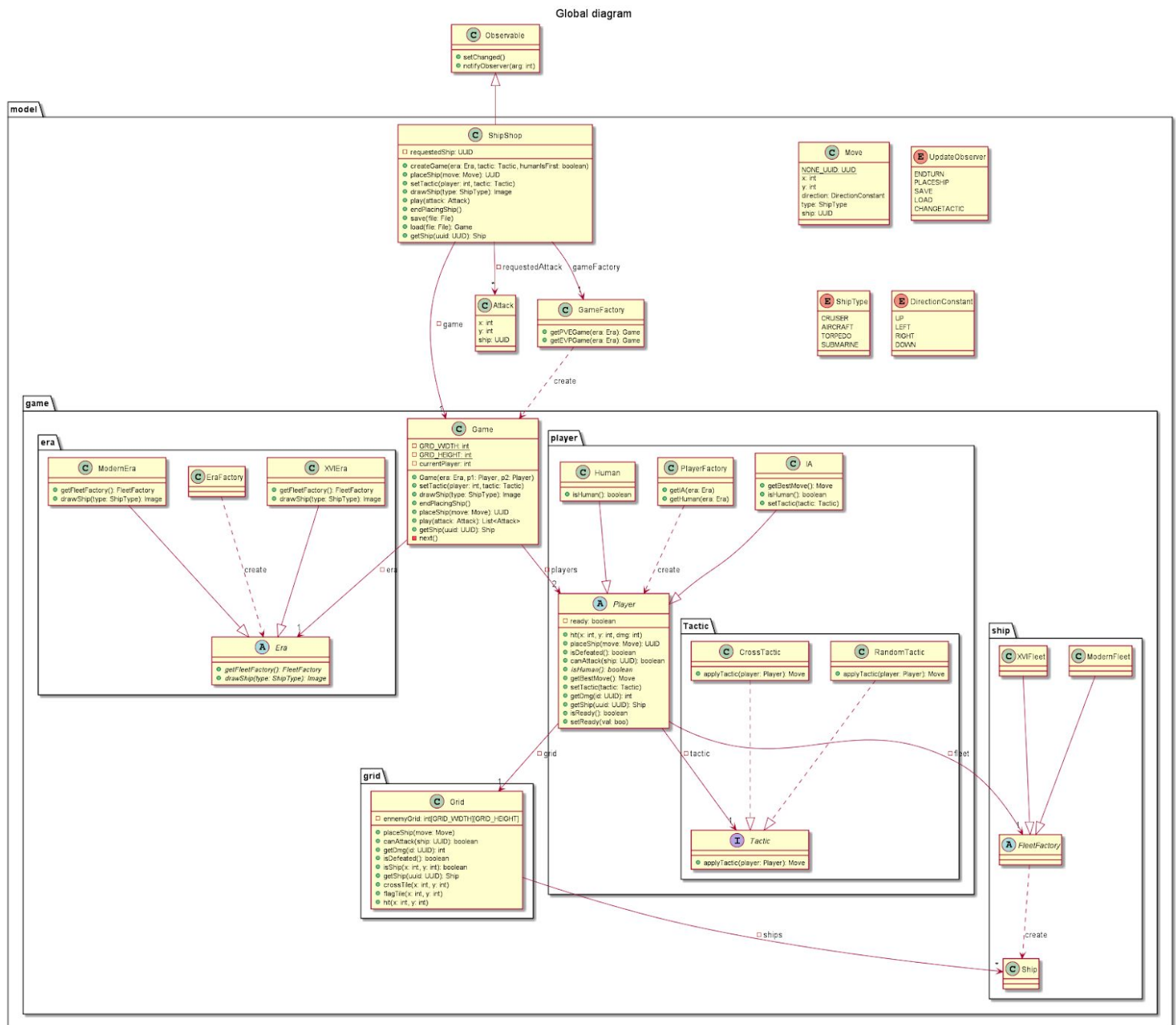
Le joueur peut sauvegarder sa partie en cours et peut la recharger en via le menu principal ou directement en jeu. (ou dans le menu principal)

- Changer la tactique de l'adversaire

Le joueur peut changer la tactique de l'adversaire en pleine partie pour modifier la façon de jouer de l'intelligence artificielle.

# Diagrammes de classes

## Diagramme global du modèle de l'application

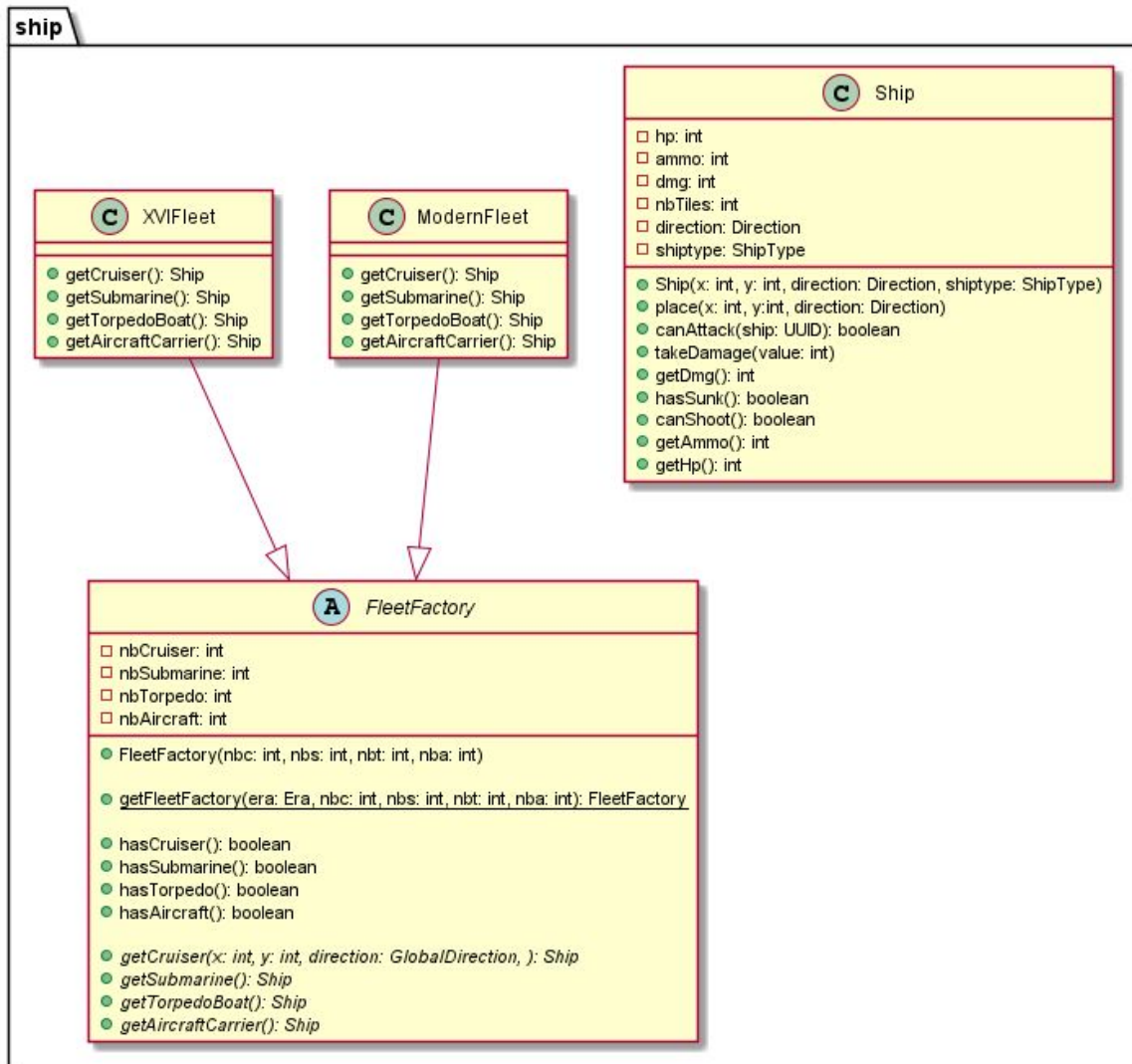


## Remarques:

- Nous avons une classe ShipShop qui est l'observé de notre application, c'est une façade qui sert à faire le lien avec l'interface graphique en filtrant les méthodes permettant de mettre à jour l'interface.
- Une classe Game représente une partie qui contient alors une ère ainsi que deux joueurs.
- L'ère est représentée sous la forme d'une stratégie afin de rendre le changement et l'ajout rapide.
- Cette ère permet de récupérer une flotte de bateaux, cette flotte est une abstract factory de bateaux. Chaque ère retourne une factory qui lui correspond, cela permet de générer les bateaux de manière cohérente et contrôlée.
- La classe Player représente un joueur qui possède une tactique, celle-ci correspond à la tactique de mouvement du joueur IA. Cette stratégie permet de changer facilement le comportement du joueur IA en pleine partie.
- Un joueur connaît sa grille qui est composée de plusieurs bateaux (les bateaux du joueur) et d'un tableau indiquant les informations que connaît le joueur de la grille adverse. (Case touchée, case coulée etc..)
- La grille connaît chaque navire grâce à un id unique (UUID) dans un dictionnaire.

Les choix ont principalement été réalisés pour favoriser l'aspect évolutif et flexible de l'application.

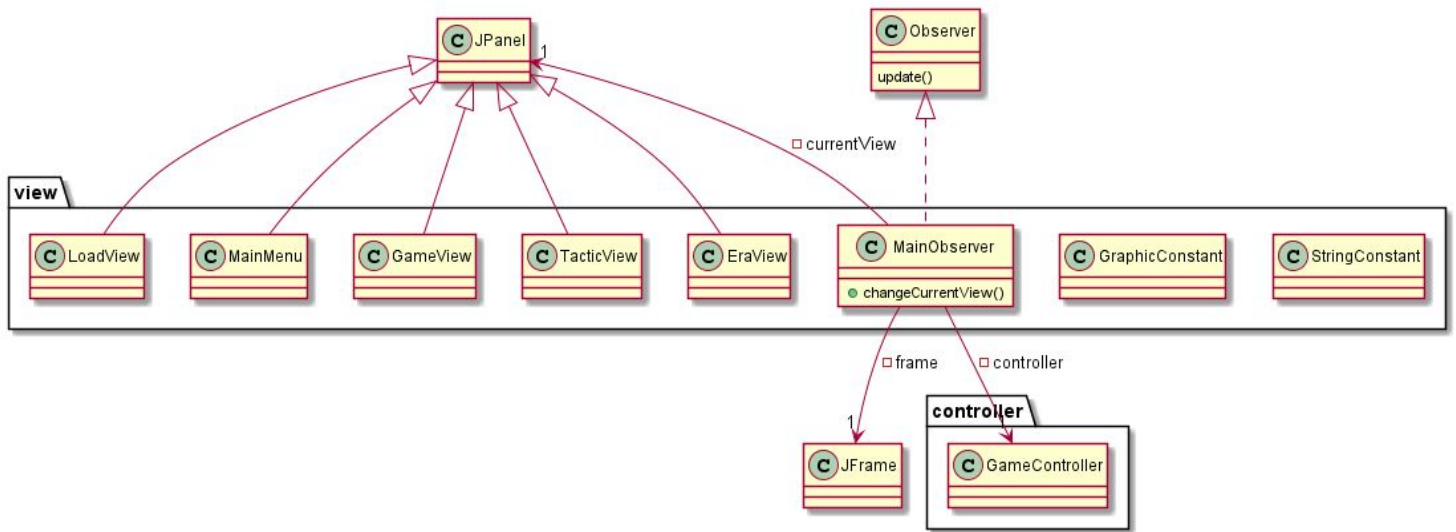
## Diagramme de classes de la partie Bateaux



### Remarques:

- Pour la conception des bateaux, nous avons choisi de créer une classe représentant un bateau avec ses différents attributs et de laisser la création de celui-ci à une abstract factory.
- Cette abstract factory représente une flotte de plusieurs bateaux (avec nombre variable) qui gère la création des navires tout en s'assurant de la cohérence de l'objet. (Pas de création d'un croiseur avec 1 case par exemple)
- La flotte dépend de l'ère de la partie, c'est pourquoi la stratégie qui gère l'ère permet de récupérer la fabrique de bateau. (un exemplaire par joueur)

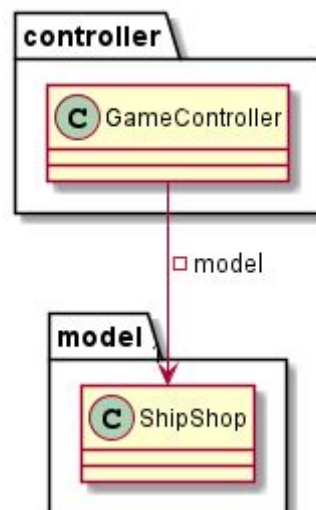
## Diagramme de classes des vues



### Remarques:

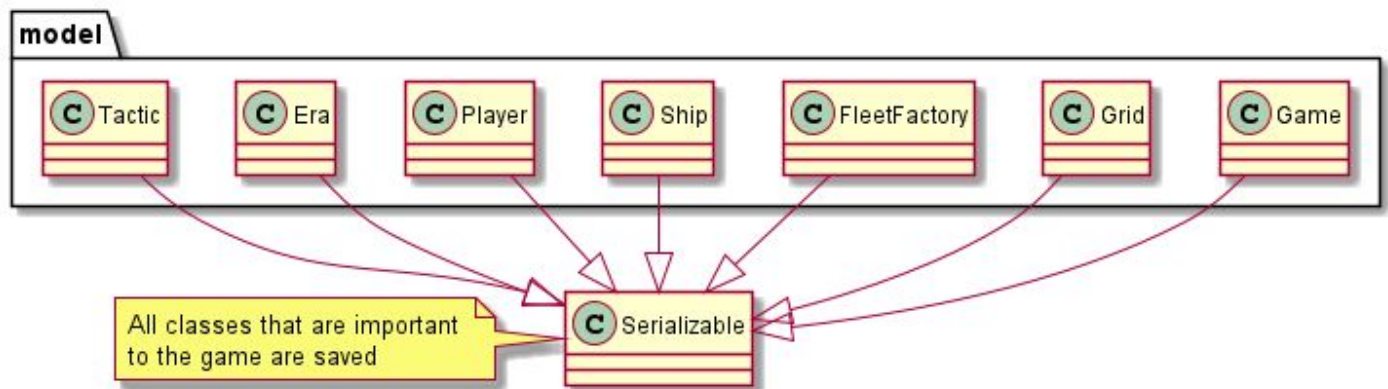
- L'application possède une vue globale qui affichera une interface graphique. (Utilisation d'une stratégie pour changer facilement)
- La vue connaît le contrôleur qui connaît le modèle et vérifiera les données avant de les envoyer.

## Diagramme de classes du contrôleur



## Diagramme de classes représentant le système de sauvegarde

### Classes that extends Serializable



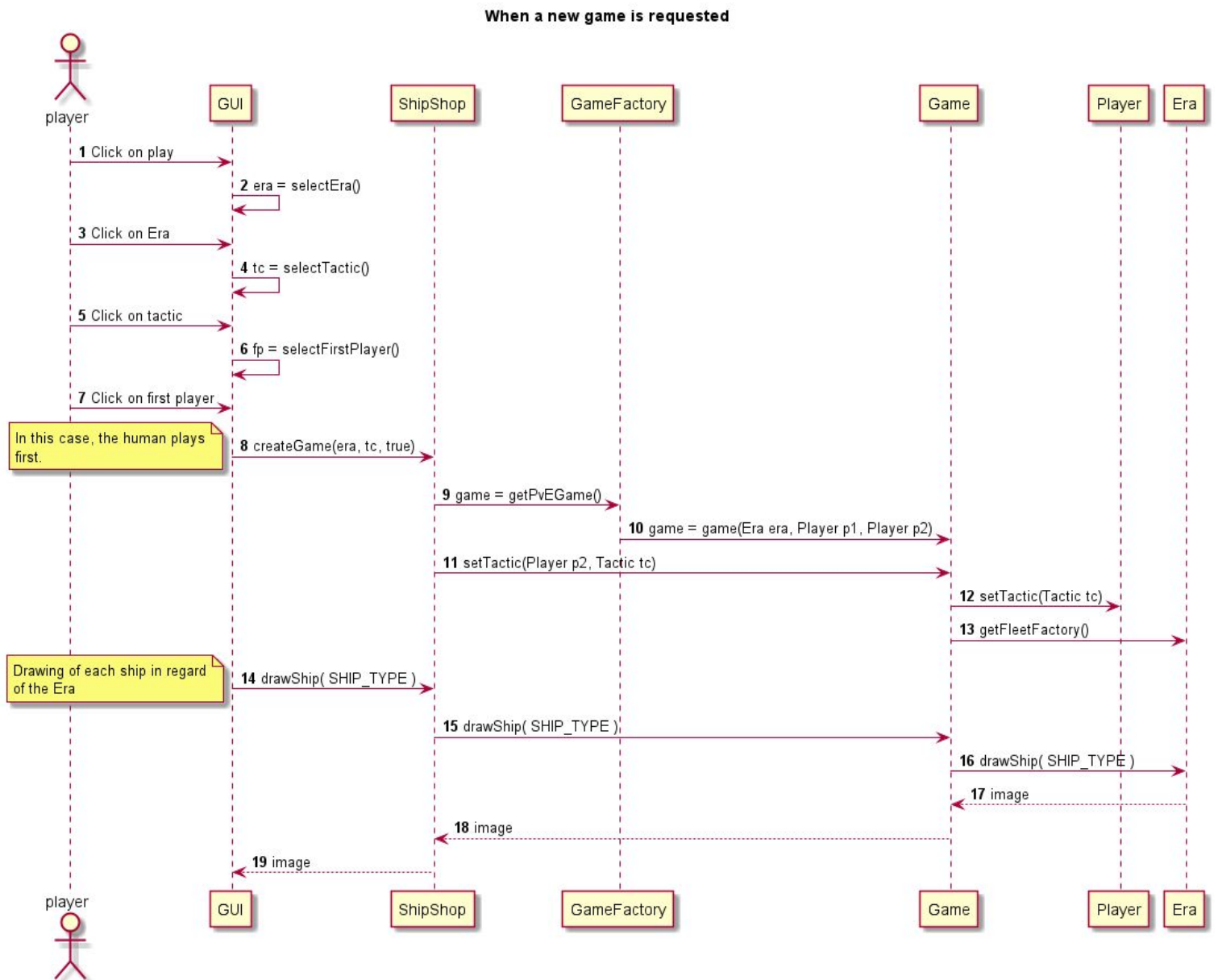
### Remarques:

- Pour gérer la persistance des données, nous avons choisi d'utiliser la classe **Serializable** de java qui permet de facilement sauvegarder divers objets dans un fichier. Pour notre projet, chaque élément du jeu hérite donc de **Serializable**.



# Diagrammes de séquences

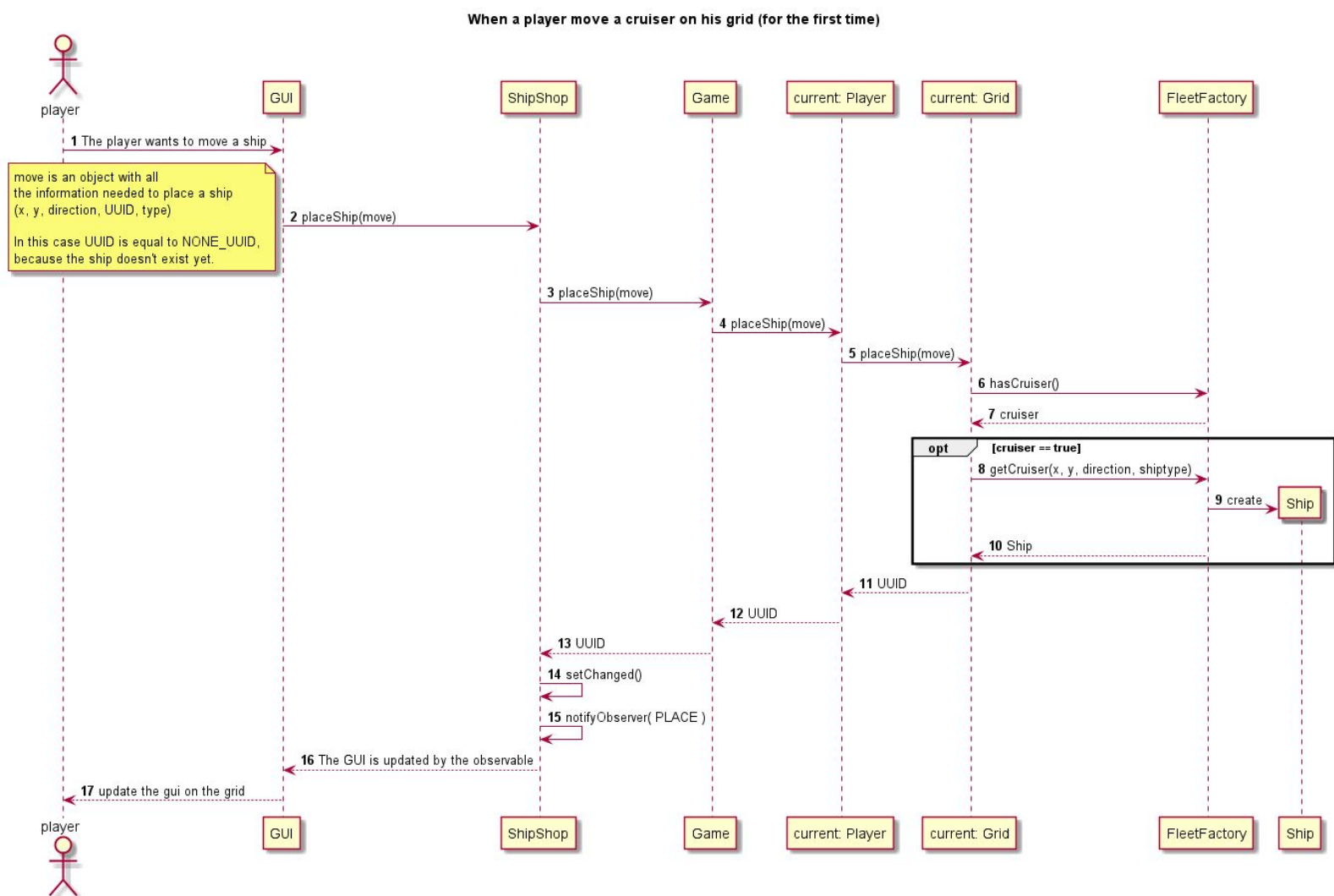
## Diagramme de séquences de création de partie



## Remarques:

- Lors de la création d'une partie, le joueur choisit une ère, une tactique et s'il joue en premier ou non. La partie est créée et le modèle initialise les objets.

## Diagramme de séquences de placement d'un bateau

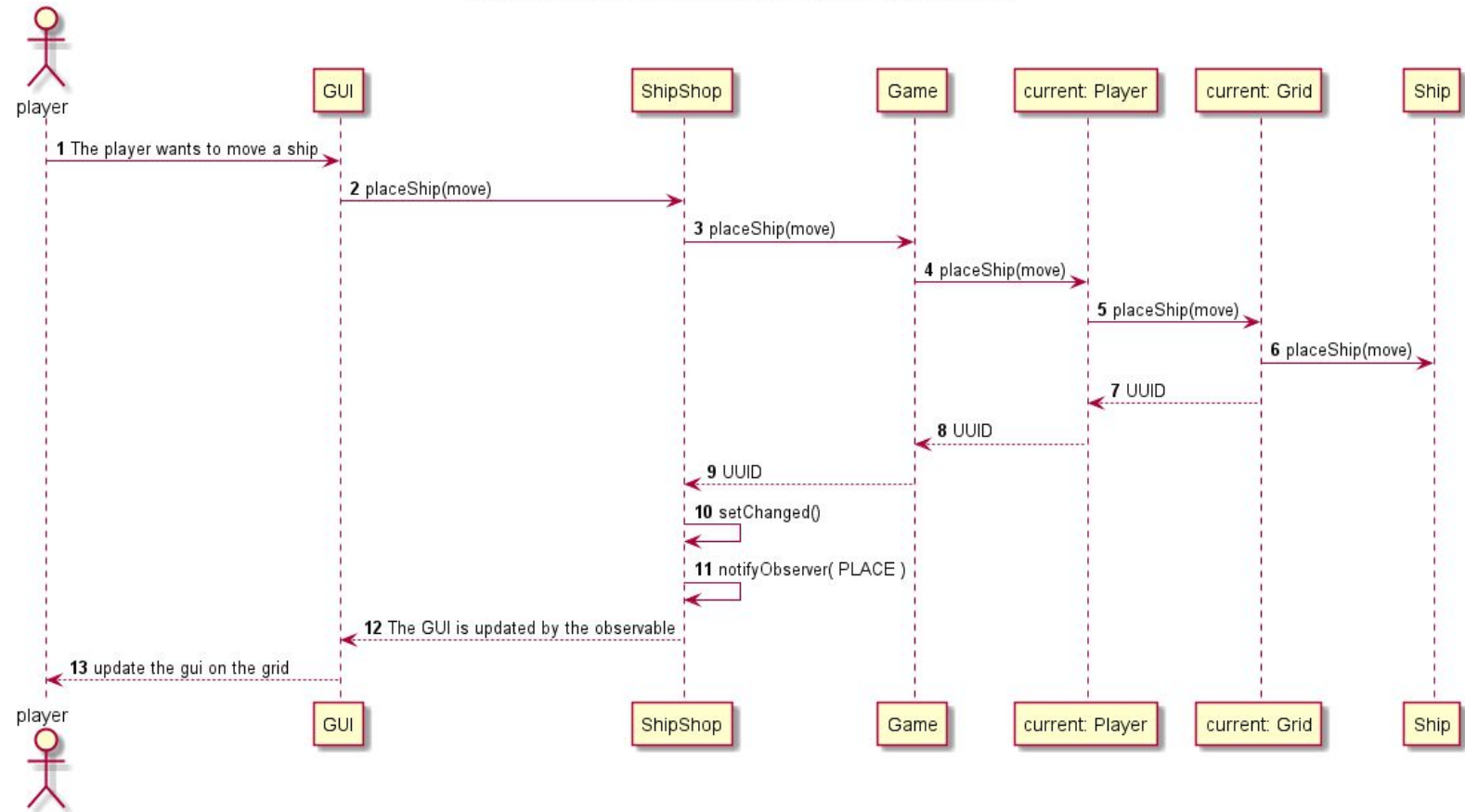


## Remarques:

- La partie commence avec une première étape de placement des bateaux. Lorsque le joueur place un navire sur sa grille, le modèle est mis à jour avec un mouvement composé d'une case x et y, de l'id du bateau et du type. (croiseur, sous-marin etc..)

## Placement d'un bateau déjà existant

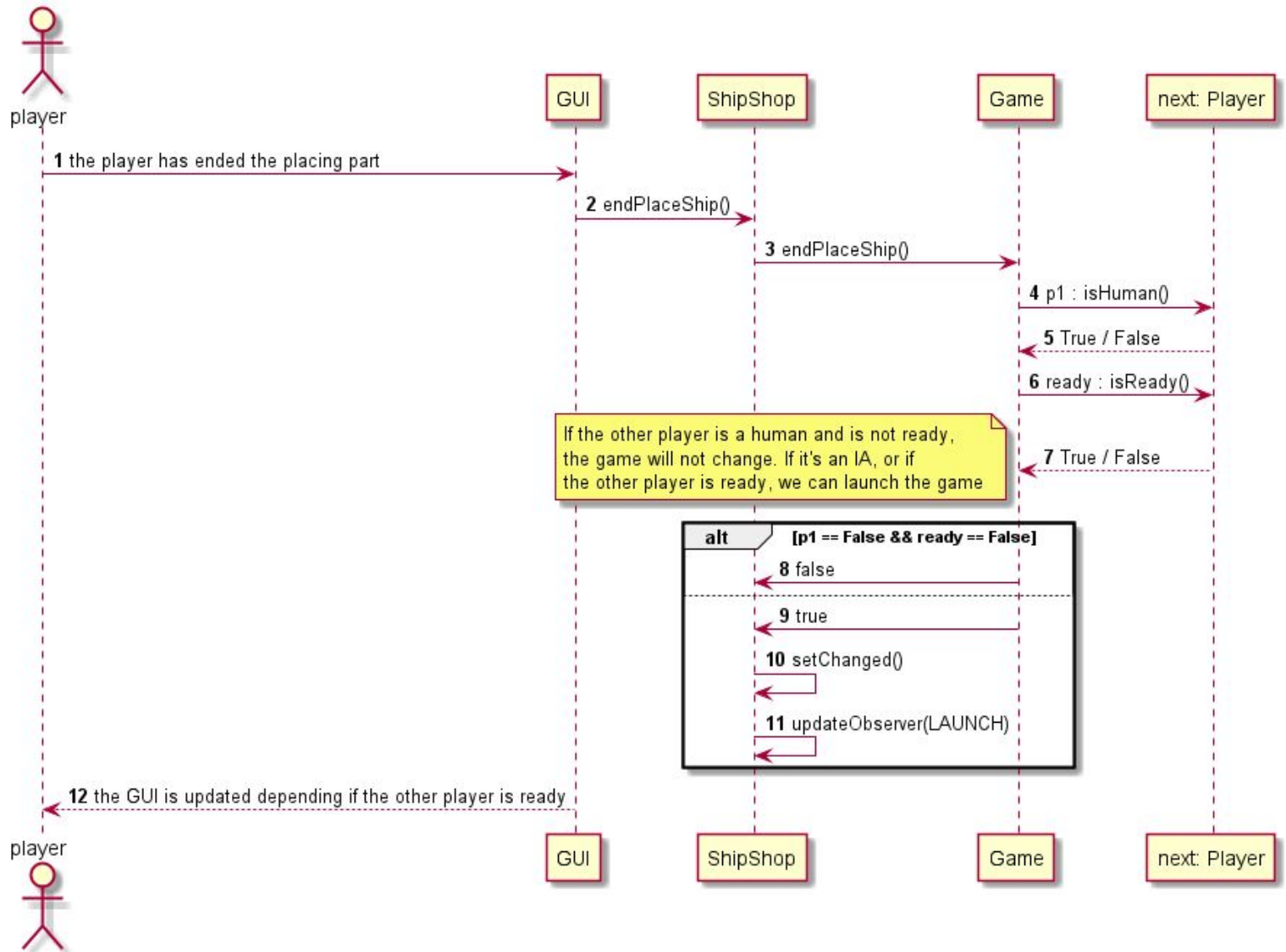
When a player move a cruiser on his grid (for the first time)



### Remarques:

- Si le joueur souhaite déplacer un navire déjà placé dans la grille, le modèle récupère le bateau via le UUID et le modifie.

## Diagramme de séquences de fin de placement



### Remarques:

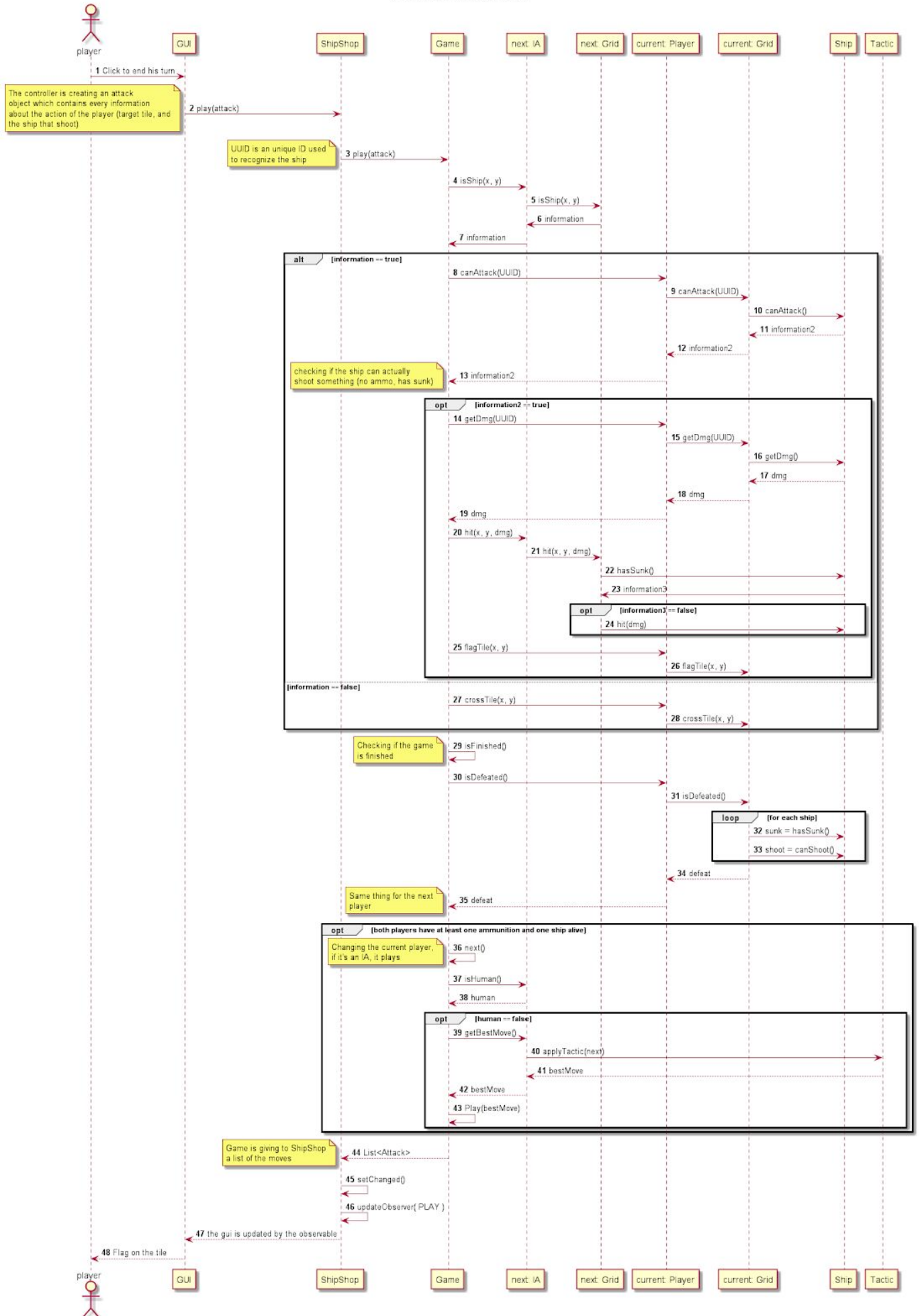
- Si le joueur souhaite mettre fin à l'étape de placement des navires, le modèle vérifie si tous les joueurs sont prêts.

## Diagramme de séquences de fin de tour

### Remarques:

- Ce diagramme de séquence représente le fonctionnement de l'application lorsque le joueur termine son tour, après avoir choisi quel navire attaque quelle case.
- On peut voir qu'après la prise en compte du mouvement du joueur (attaque d'une case de la grille de l'adversaire), on vérifie si le deuxième joueur est une IA, auquel cas, on utilise la stratégie tactique afin de récupérer le mouvement, et on l'applique.

# When a player is ending his turn



## Diagramme de séquences de changement de tactique



### Remarques:

- Ce diagramme de séquence montre ce qui se passe lorsque le joueur demande à l'application de changer de tactique en pleine partie.

## Diagramme de séquences de sauvegarde

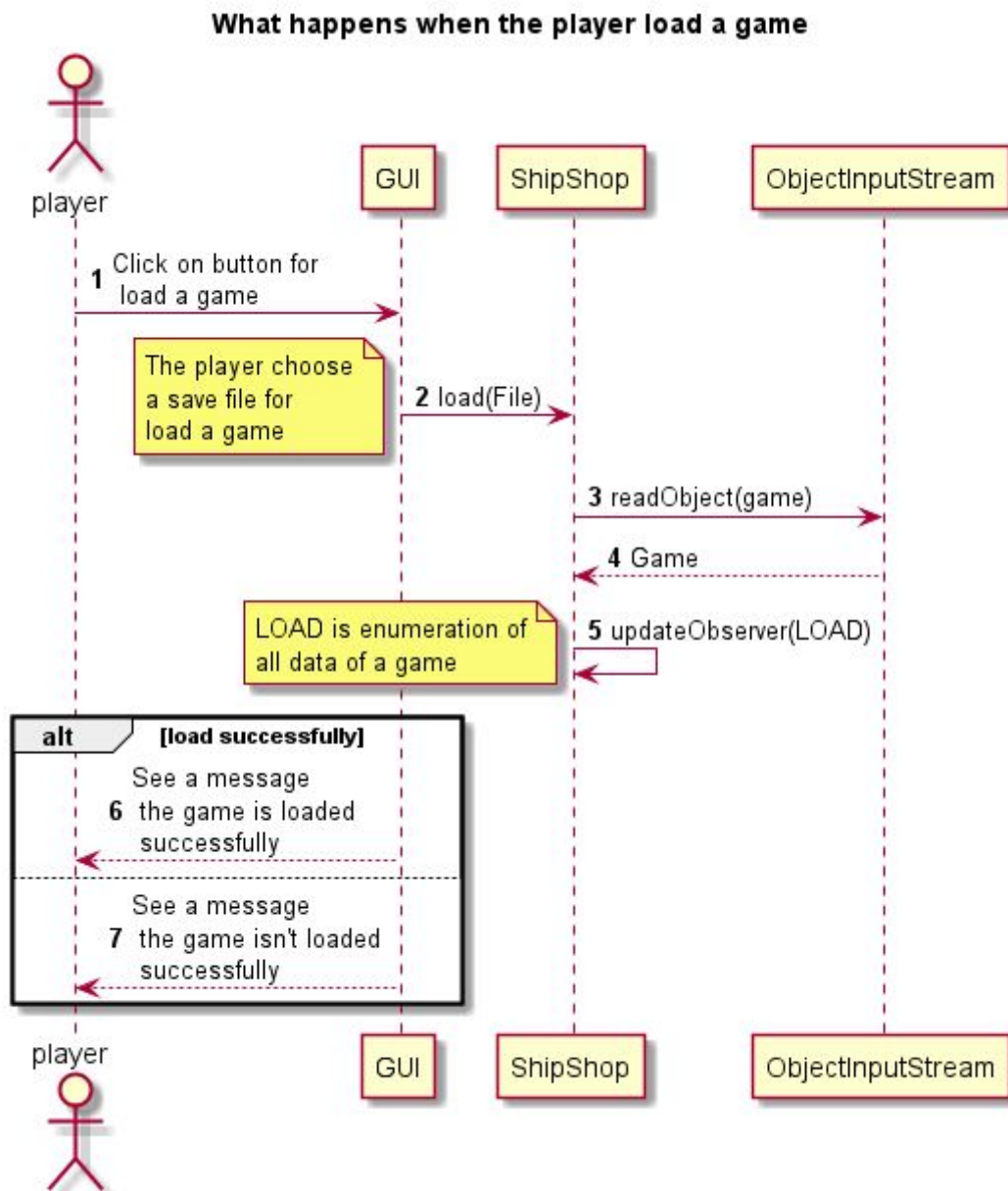


### Remarques:

- Pour sauvegarder une partie en cours, on utilise la classe `Serializable` qui permet de facilement sauvegarder un objet dans un fichier.



## Diagramme de séquences de chargement

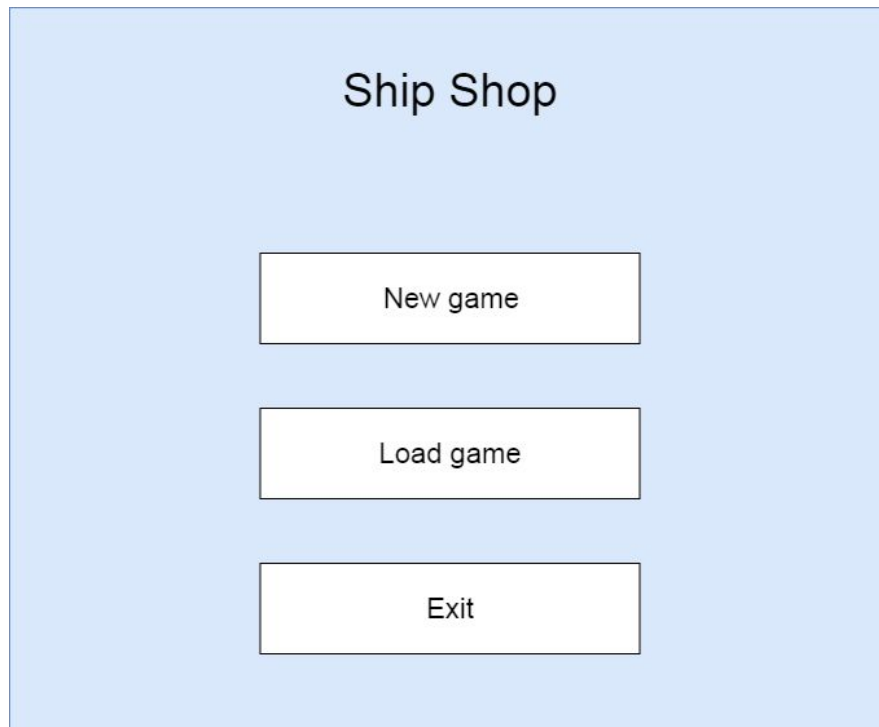


### Remarques:

- Le chargement est également rendu facile par l'utilisation de la classe `Serializable`, ainsi l'interface graphique sera notifiée du chargement et pourra alors se mettre à jour.

# Prototypage de l'interface

## Menu principal



## Menus de choix de la tactique et de l'ère

Choose the era !

Modern



XVI

.....

Choose the  
ennemy tactic !

Random



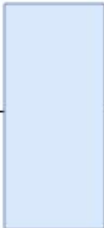



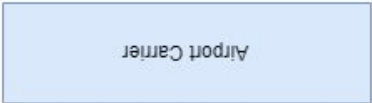
Cross

.....


## Menu de placement des bateaux


				
				
				
				




Airport Carrier



Cruiser



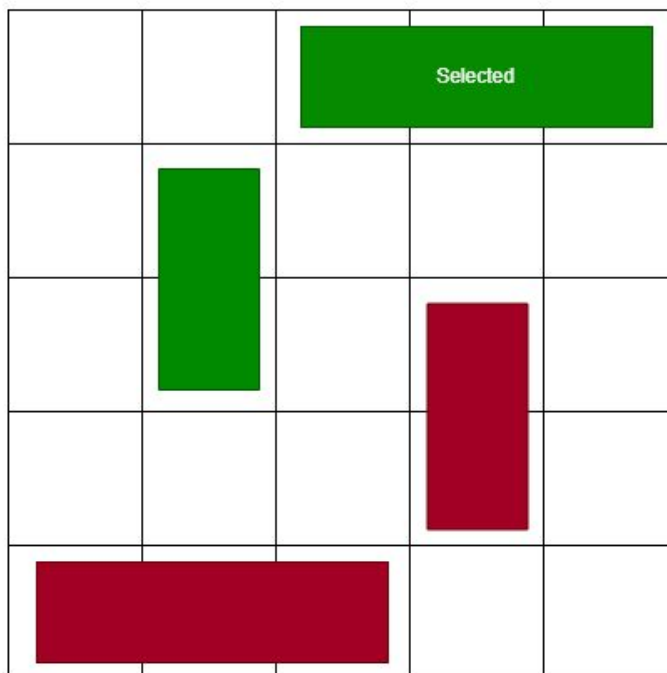
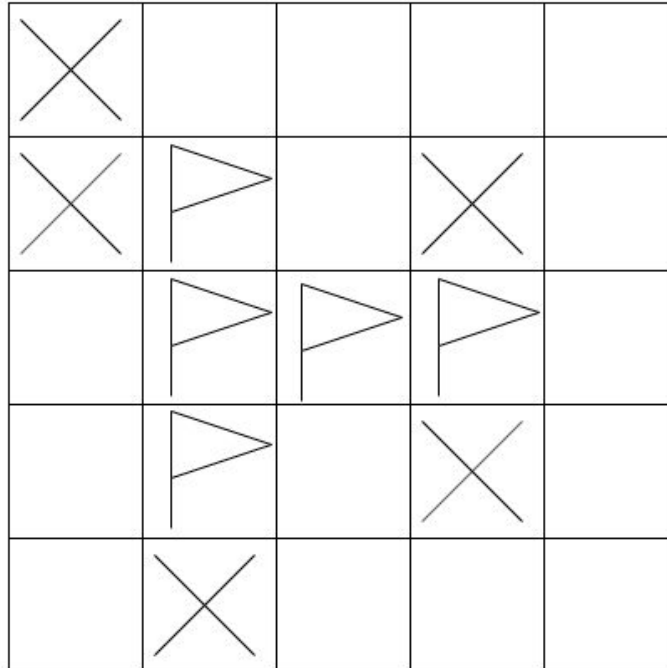
Submarine



Torpedo

Ready

## Menu de jeu



### Selected ship

PV : 10  
Attack : 2  
Ammo : 10

End Turn

## Remarques:

- L'interface de jeu affiche les différents bateaux du joueur dans la grille en dessous, et les différentes tentatives d'attaque dans la grille du dessus.
- On peut voir que les navires affichés sont actualisés lorsqu'il ont coulés pour permettre au joueur de suivre l'avancé de la partie.
- On peut également voir que le joueur peut afficher les informations d'un bateau spécifique en cliquant dessus, il pourra alors choisir la case à attaquer.
- Enfin, il pourra mettre fin à son tour.
- Pour la sauvegarde et le changement de tactique en pleine partie, on peut imaginer deux onglets en haut de l'interface.