

Conception d'architectures distribuées

Conception - Bataille navale

Antoine Courtil

Simon Hajek

Guillaume Petit

SOMMAIRE

Introduction	3
Choix de représentation du modèle	3
Comportement général et changement d'états de l'application	3
Diagrammes de classes	3
Déroulement d'un tour de jeux	6
Sauvegarde de la partie	8
Diagramme d'état de l'application	8
Diagramme de séquence - Comportement général	9
Vues proposées pour l'application	10

Introduction

Ce rapport présente la partie conception du projet bataille navale. Il contient majoritairement des diagrammes UML représentant la structure du programme et quelques exemple de comportement ainsi que des explications sur les choix de conception.

Choix de représentation du modèle

Pour représenter les plateaux, aucun tableau en deux dimensions n'est utilisé. Nous avons choisi d'intégrer les positions dans les différentes classes :

- La position d'un bateau dans la classe *Bateau*
- La position des tirs réussis dans la classe *Bateau*
- La position des tirs échoués dans la classe *Player*

Nous avons choisi cette implémentation afin de ne pas à avoir dupliquer les différentes positions. L'affichage qui découle de cette implémentation ne sera simplement qu'un parcours des différents éléments afin de les ajouter visuellement.

Comportement général et changement d'états de l'application

La boucle principale sera la boucle Evolve de la classe Game qui traitera selon si c'est au joueur ou à l'IA de jouer les cliques et les commandes envoyées. Cette même fonction appellera la sous fonction evolve correspondante à l'état du jeu pour réaliser les traitements nécessaires.

Diagrammes de classes

Diagramme de classe du package game

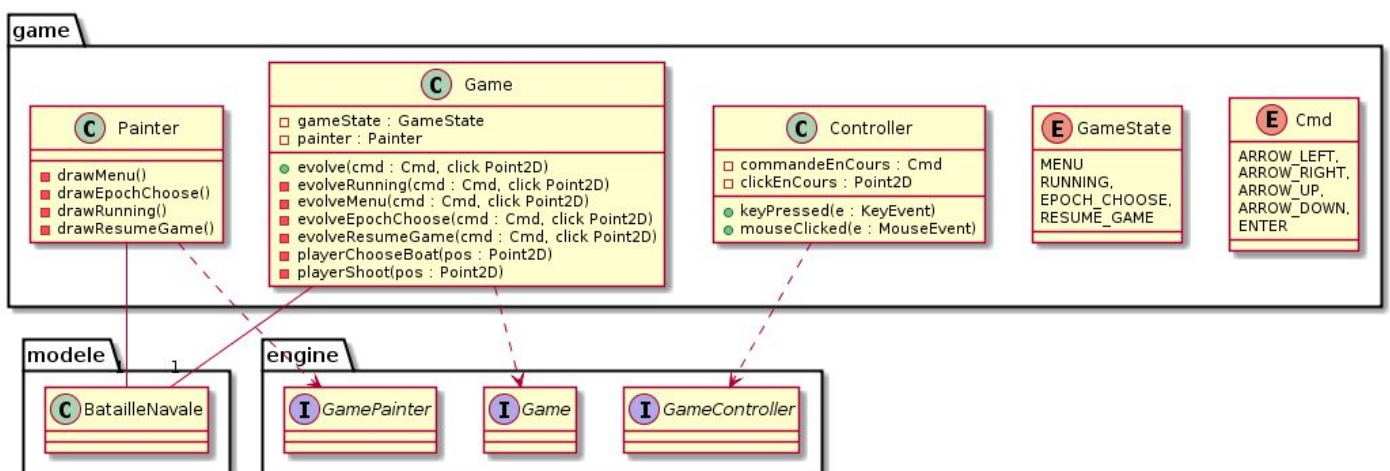


Diagramme de classe du package modele

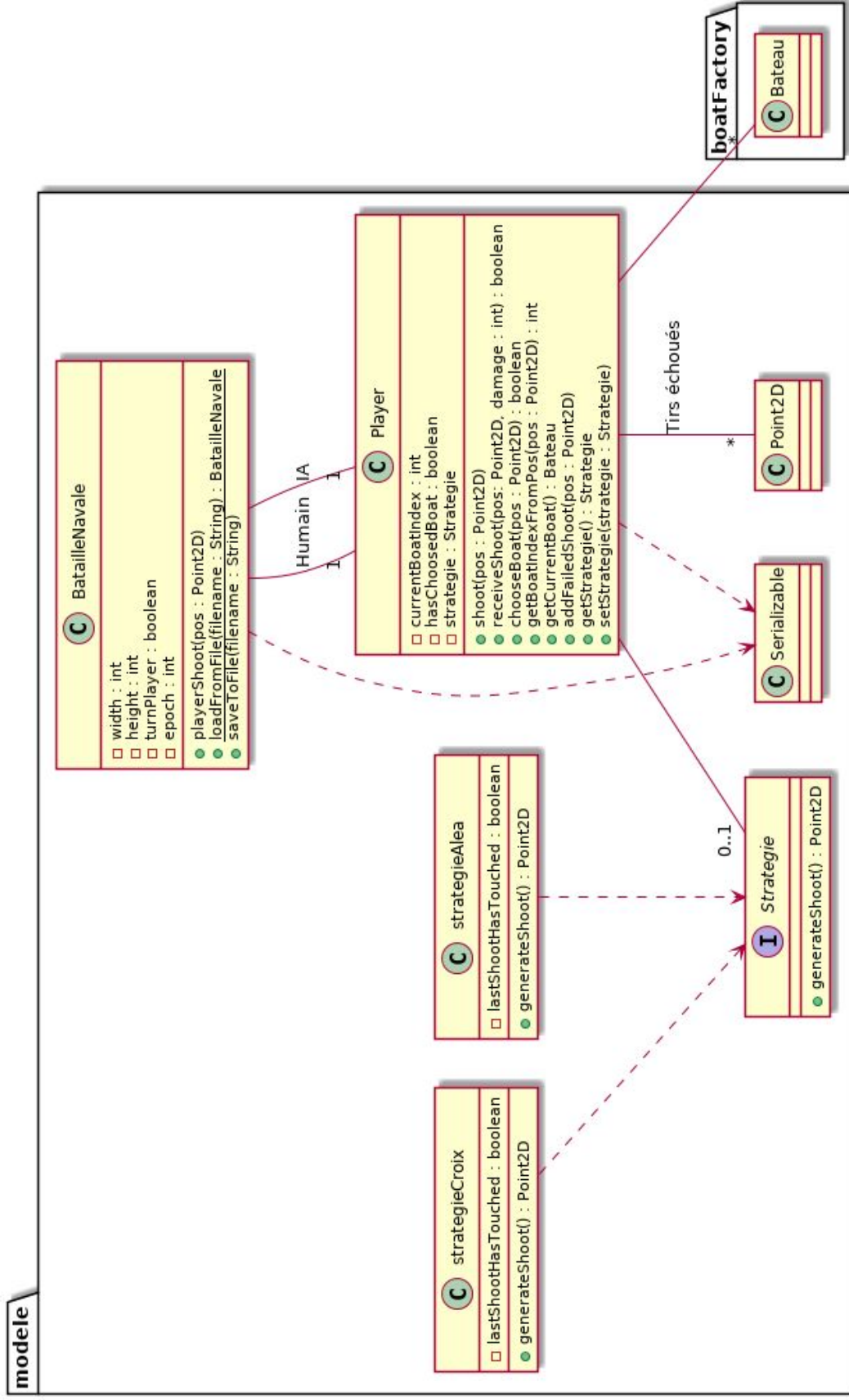


Diagramme de Classe du package Factory

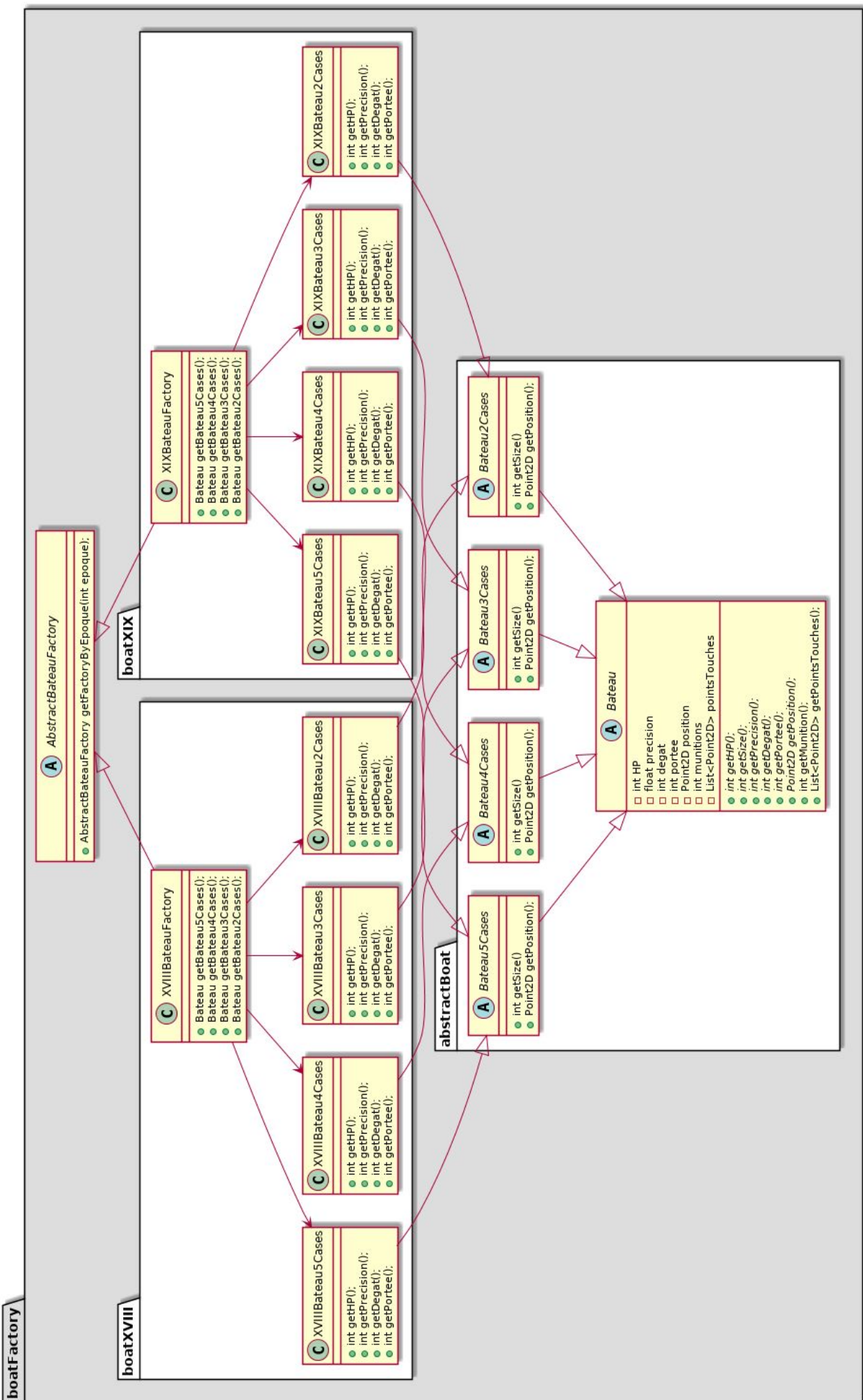
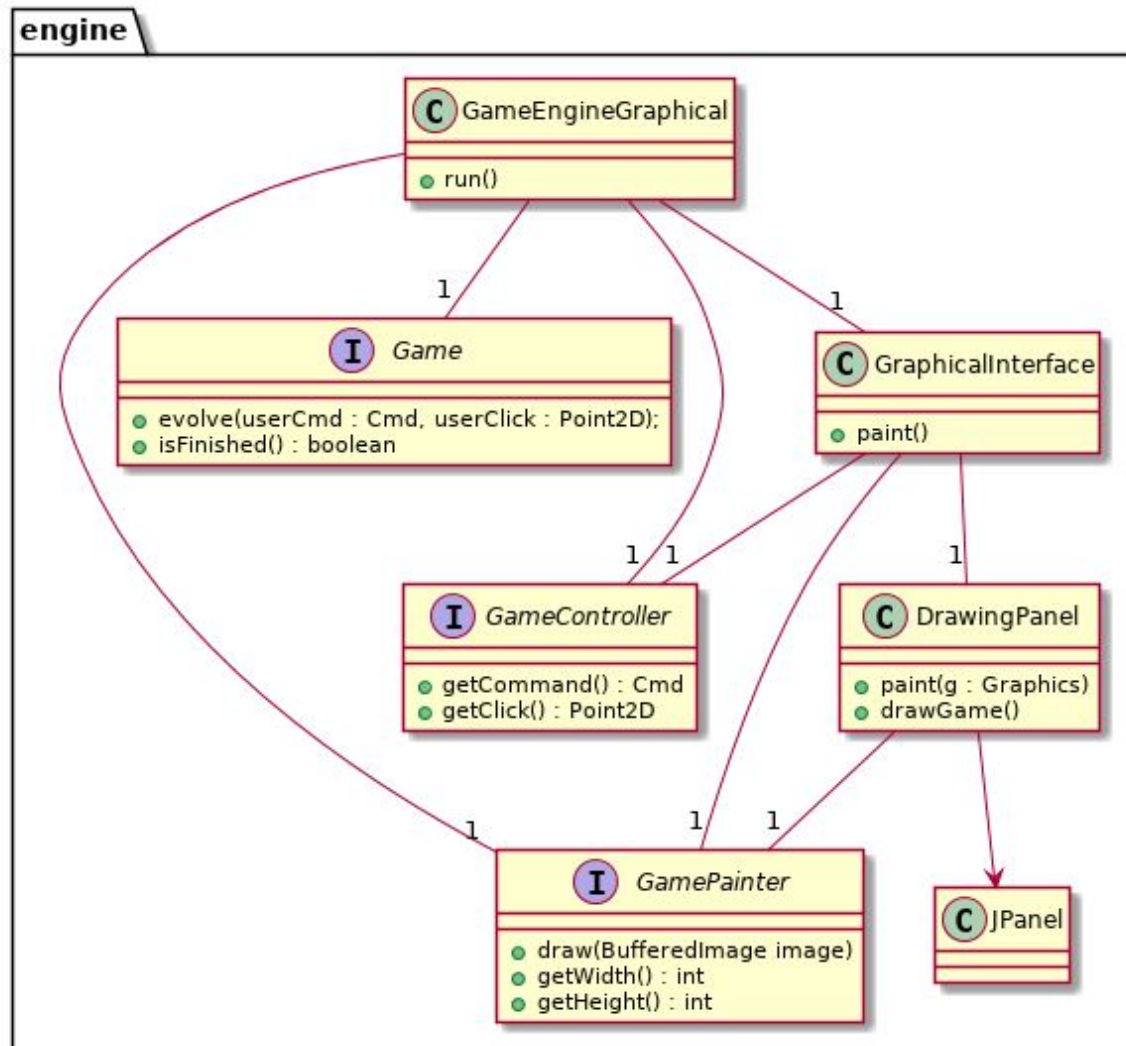


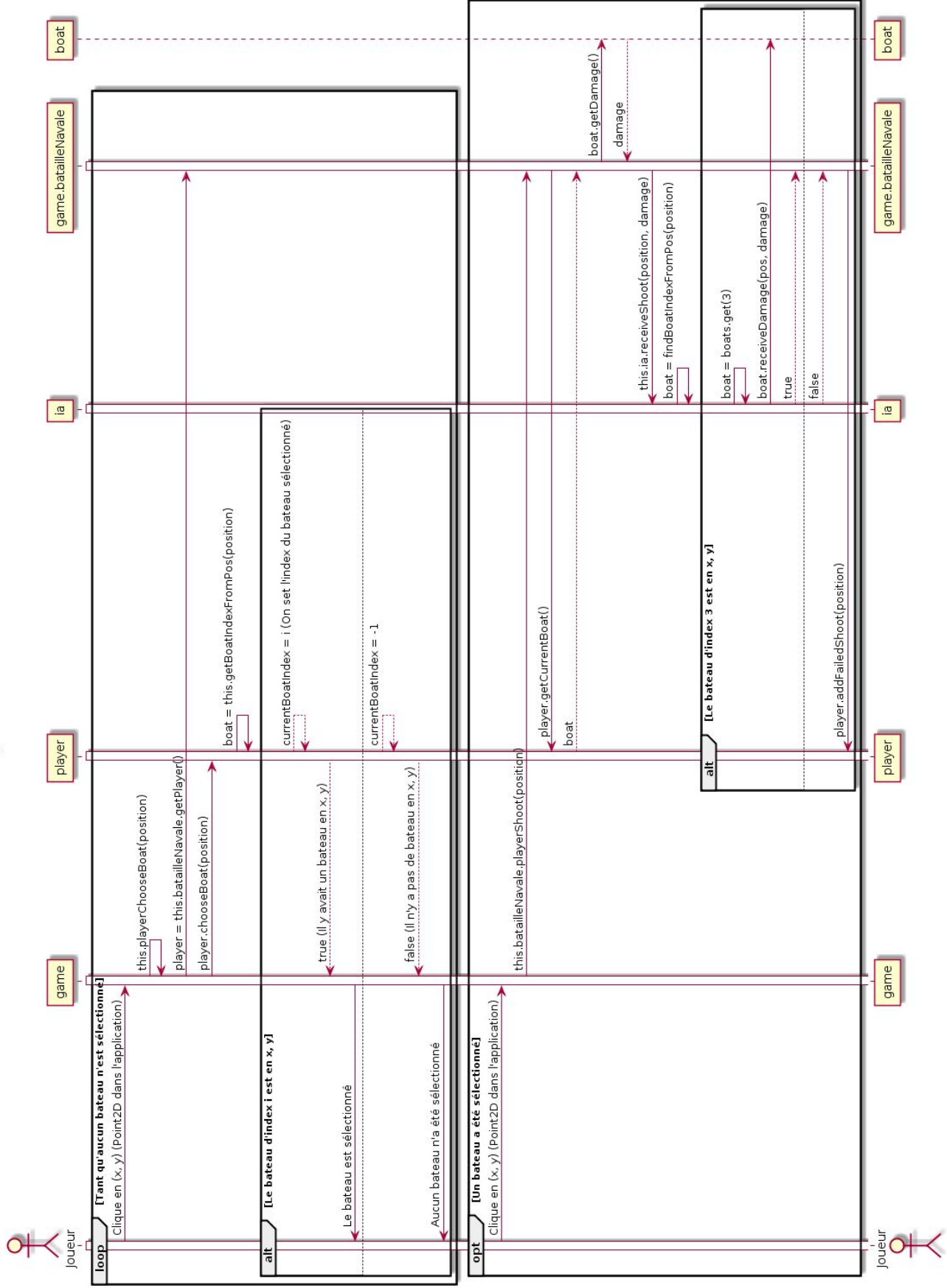
Diagramme de classe du package engine



Déroulement d'un tour de jeu

Pour récupérer la position cliquée, le Contrôleur hérite de *MouseListener* afin de récupérer les positions cliquées sur la fenêtre pour par la suite vérifier la cohérence des cliques.

Le joueur sélectionne un bateau et tir



Sauvegarde de la partie

Pour sauvegarder la partie, nous avons décidé de simplement 'serialiser' la classe *BatailleNavale*. Les classes *BatailleNavale*, *Player* et *Bateau* implémentent donc l'interface *Serializable* et les méthodes *loadFromFile* et *saveToFile* sont utilisées respectivement pour charger et sauvegarder la partie depuis le menu principal.

Diagramme d'état de l'application

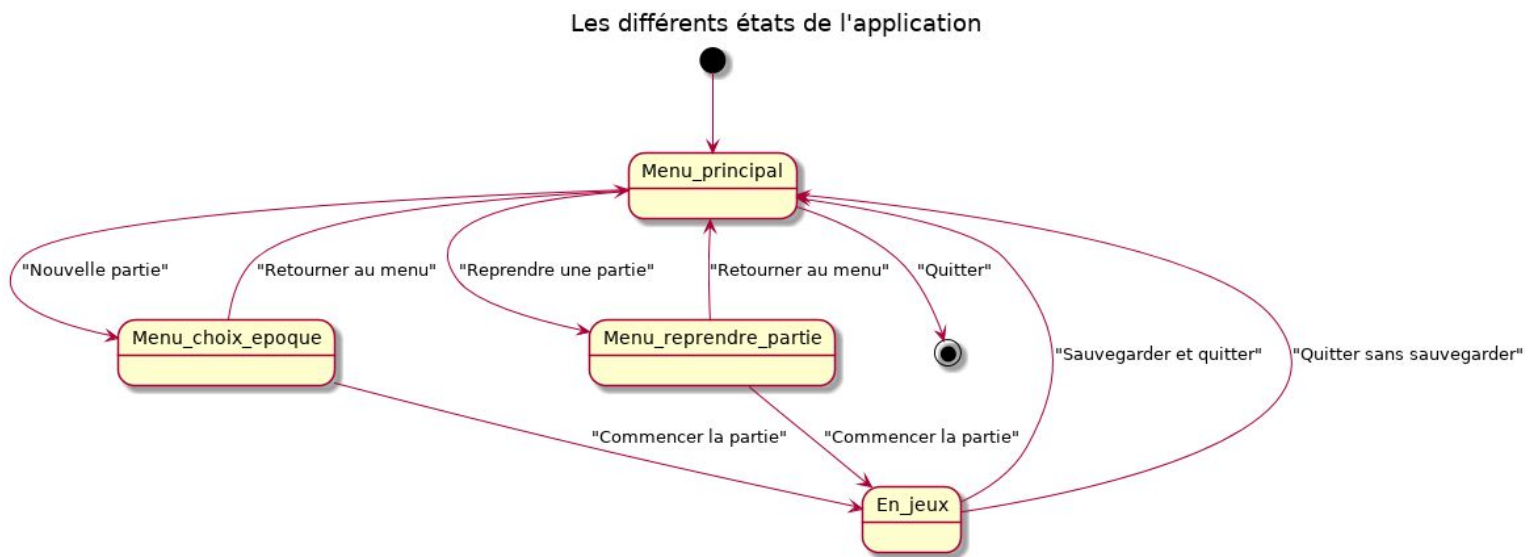
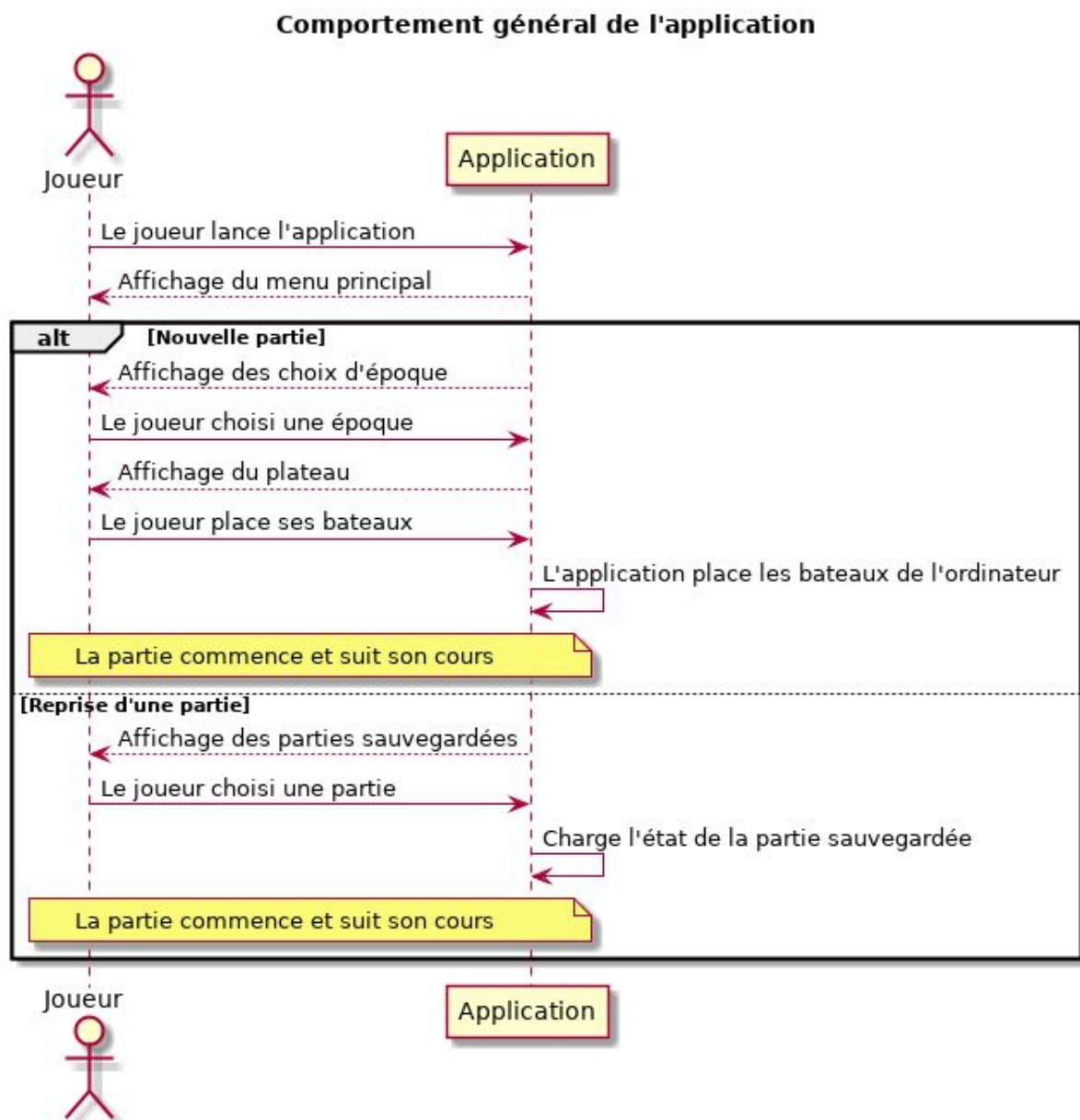


Diagramme de séquence - Comportement général



Vues proposées pour l'application

Menu principal

Nouvelle partie

Reprendre une partie

Quitter

Epoques

XXe siècle

XVIe siècle

Bataille navale

Epoque : XIX

Nombre de tirs ratés : 2

Bateau sélectionné :

Munitions : 3

Vies : 2

Dégats : 3

Save

Quit

Stratégie

Aléatoire

En croix