

Garapati Vivek  
Ramanantsoa Fifaliana  
Yu Ziru

---

# *Rapport de projet*

---

Bataille Navale

L3 MIAGE UEVE  
2015-2016  
Projet informatique

# Sommaire

## **I. Introduction**

## **II. Analyse et conception**

1. Les caractéristiques du jeu
2. Diagrammes UML
  - a) Diagramme de cas d'utilisation
  - b) Diagrammes d'activité
  - c) Diagramme de classe
3. Diagramme de Gantt

## **III. Réalisation et implémentation**

1. Choix des moyens pour la réalisation
2. Mode d'emploi et déroulement d'une partie

## **IV. Principales difficultés**

## **VI. Bilan et Conclusion**

# Introduction

La bataille navale aussi bien appelée « touché-coulé » est un jeu de société dans lequel deux adversaires doivent deviner et bombarder des coordonnées de la flotte adverse jusqu'à ce que tous ces bateaux de l'adversaire soient totalement détruits.

Chacun des membres de notre groupe trouvant intéressant de modéliser le jeu de bataille navale sous forme d'application, nous avons donc choisis de traiter ce sujet.

Afin d'y parvenir, il nous a fallu analyser tout d'abord le sujet à l'aide de diagrammes UML de cas d'utilisation, d'activité et de classes nous organiser en nous répartissant les tâches à accomplir.

Suite à la partie analyse et conception, nous sommes passés à l'implémentation du programme en question. Nous y décrivons le déroulement d'une partie avec les différents choix de modes et de types de jeux ainsi que le fonctionnement de l'IA créée.

Et pour finir, nous citerons les différentes difficultés rencontrées ainsi que les raisons des différentes solutions que l'on a proposées.

## II- Analyse et conception

### 1. Exigences de l'application

Le projet consiste à concevoir et à réaliser une application qui permet de jouer au jeu de Bataille Navale. Il ne s'agit pas ici de la simple bataille Navale classique. En effet, nous devons y rajouter plusieurs fonctionnalités :

Il y a d'abord 3 différents **modes de jeux** :

- (1) Le **mode démo** qui oppose deux intelligences artificielles. Pour cela nous devons prévoir la création d'une IA.
- (2) Le **mode Joueur VS IA**
- (3) Le **mode Joueur VS Joueur**

4 types de bataille navale sont possibles :

- (1) La **bataille navale classique**
- (2) La **mission radar** semblable à la bataille Navale classique sauf que le joueur reçoit l'indication de la cible la plus proche
- (3) **L'opération artillerie** similaire au mode classique. le joueur sélectionne d'abord la coordonnée verticale de la cible et les numéros de la coordonnées horizontale apparaissent une à une.
- (4) **L'alerte Rouge** qui est un mélange des 3 autres modes (attaque en mode artillerie et indication sur la cible la plus proche en cas de tir manqué)

### 2. Diagrammes UML

Afin de mener à bien l'analyse de notre projet et de bien répondre aux exigences de l'application nous avons choisi de modéliser différents aspects du projet avec **UML**.

Les **diagrammes UML** permettant de donner avant tous un point de vue « non informaticien » nous donnent une compréhension plus simple de notre système et sont essentiels dans cette analyse.

#### a) Diagramme de cas d'utilisation

Le diagramme ci-dessous met en évidence les différents services (cas d'utilisations) offerts aux utilisateurs par le système

## DIAGRAMME USE CASE BATAILLE NAVALE



Par rapport à l'application l'utilisateur a le droit de choisir le mode de jeu qu'il souhaite (J VS J, IA VS IA ou J VS IA), le type de partie (Bataille Navale, mode artillerie, mission radar, alerte rouge). C'est lui qui choisit ses navires et qui les place comme il le souhaite. Il peut aussi commencer une nouvelle partie.

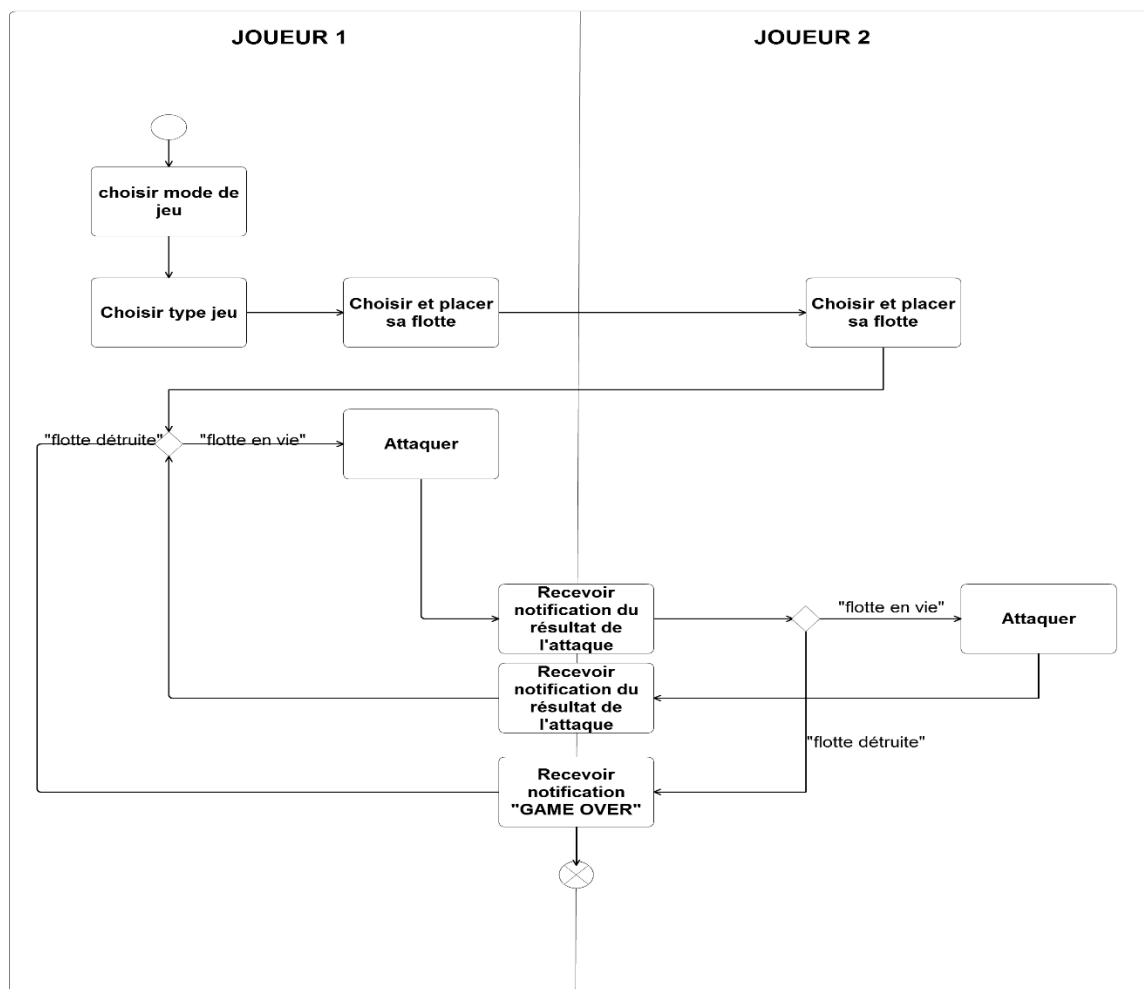
Lors du déroulement d'une partie, c'est alternativement le tour de chaque joueur d'attaquer ou de se faire attaquer. A chaque tir chaque joueur a le droit d'être au courant du résultat (raté, coulé, touché) et aussi de pouvoir visionner l'état de leur flotte respective.

## b) Diagrammes d'activité

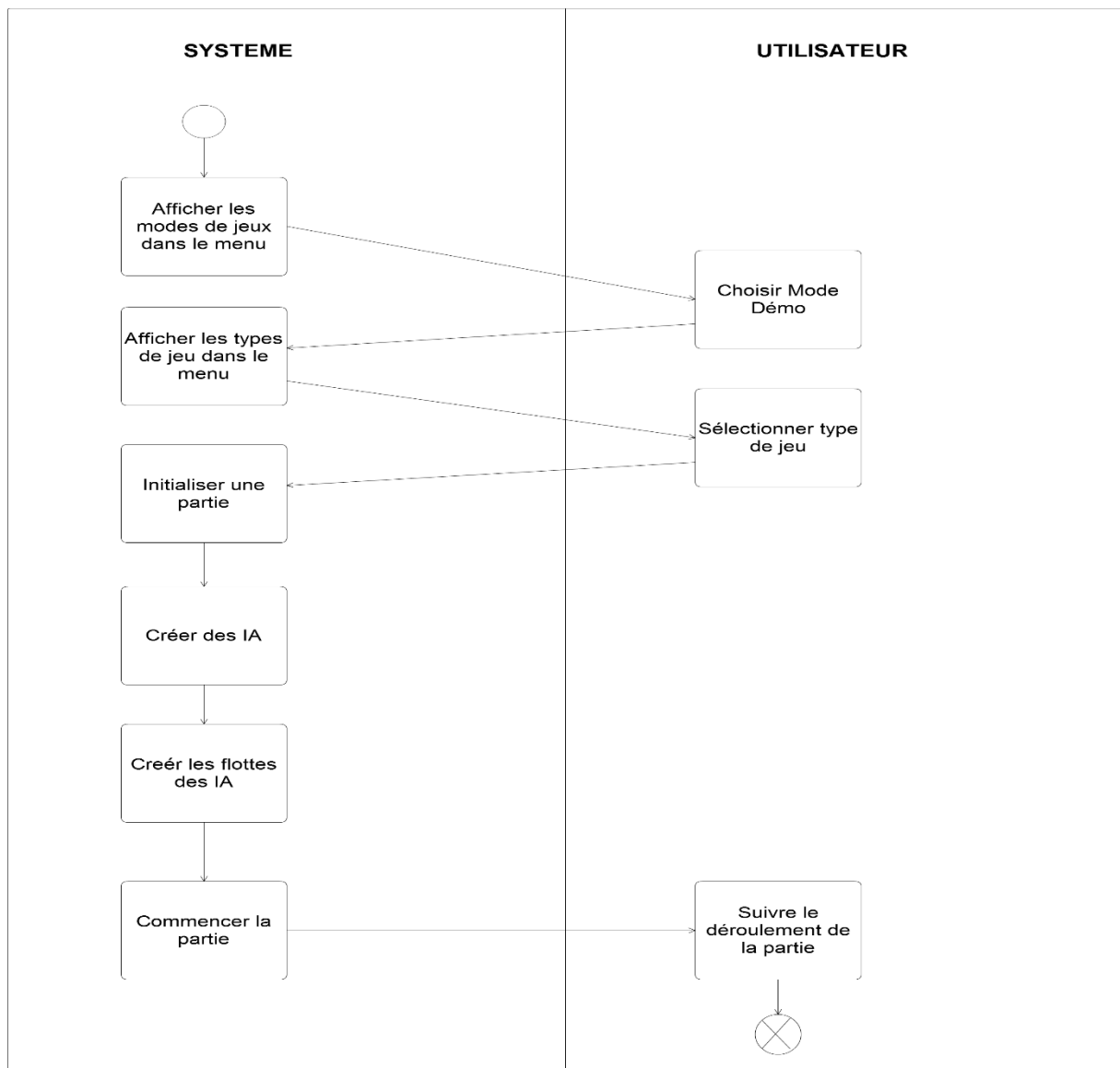
Les diagrammes d'activités ci-dessous détaillent le déroulement des cas d'utilisations et mettent en évidence les interactions entre les acteurs de chacun d'eux.

Ci-dessous un diagramme qui décrit le déroulement d'une partie en général

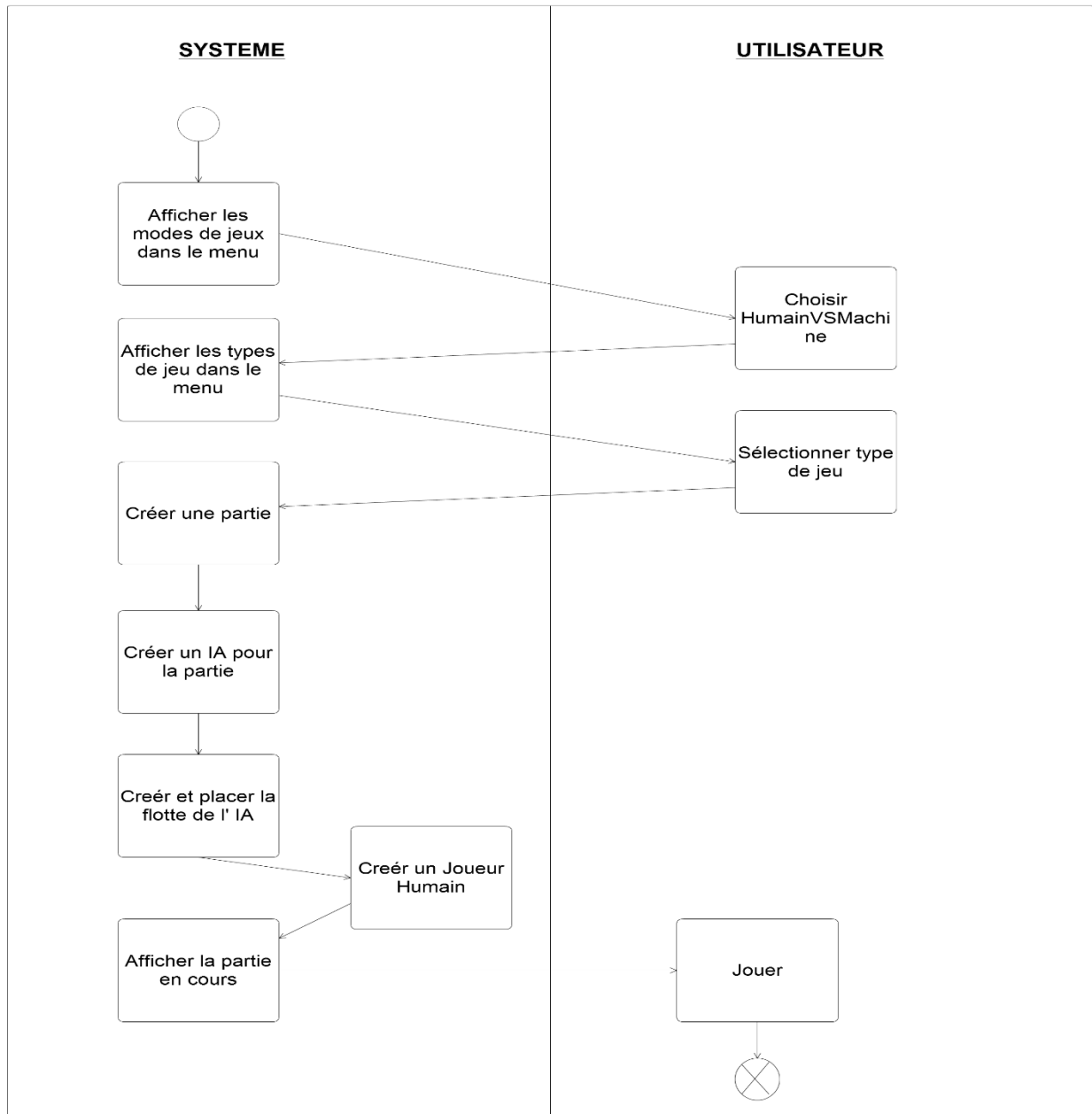
### **DIAGRAMME D'ACTIVITE: CHOIX MODE+TYPE BATAILLE NAVALE**



Nous allons maintenant décrire chaque choix de mode de jeu à l'aide de diagrammes  
DESCRIPTION DE L'USE CASE "REGARDER DEMO"

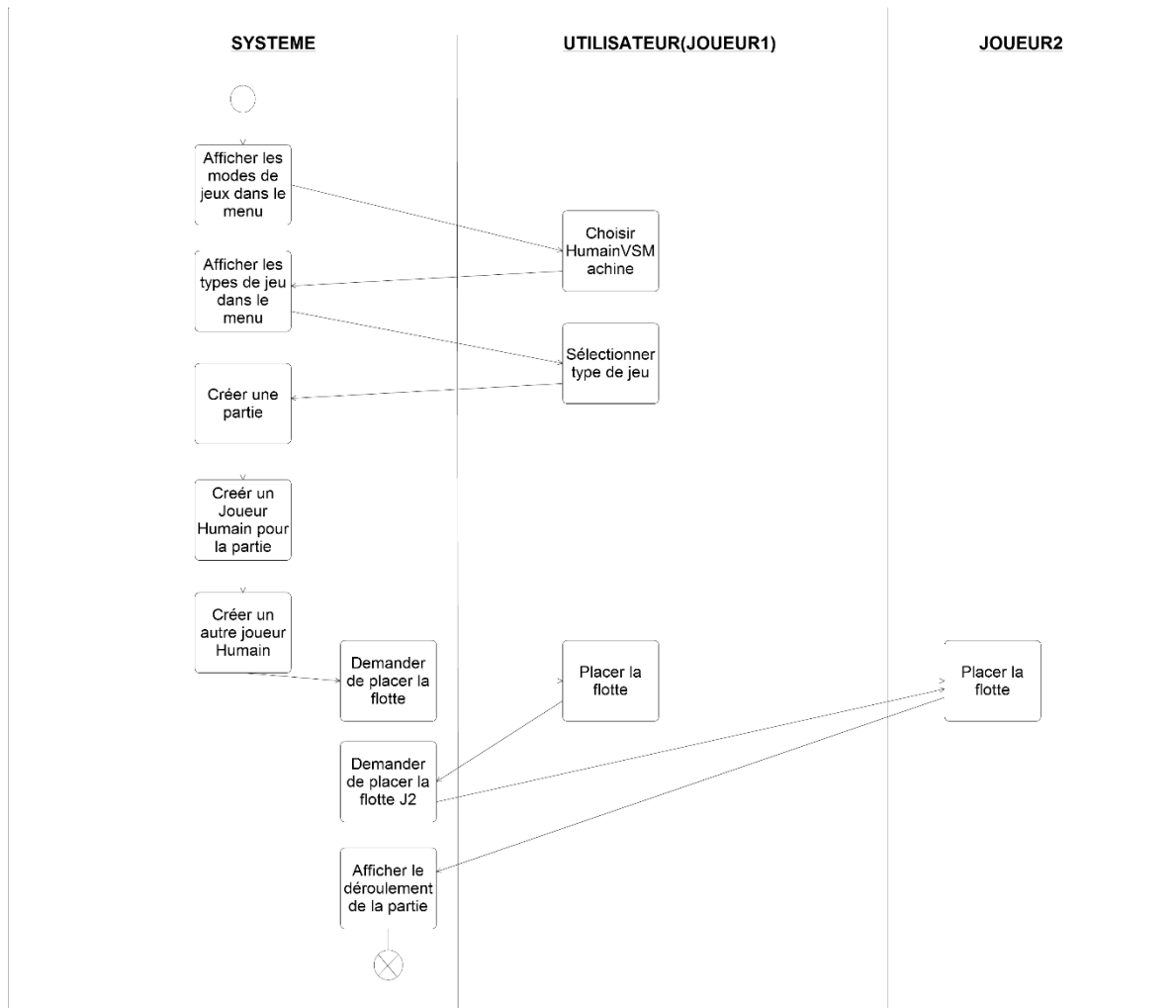


## DESCRIPTION DE L'USE CASE "CHOISIR HUMAIN VS MACHINE"

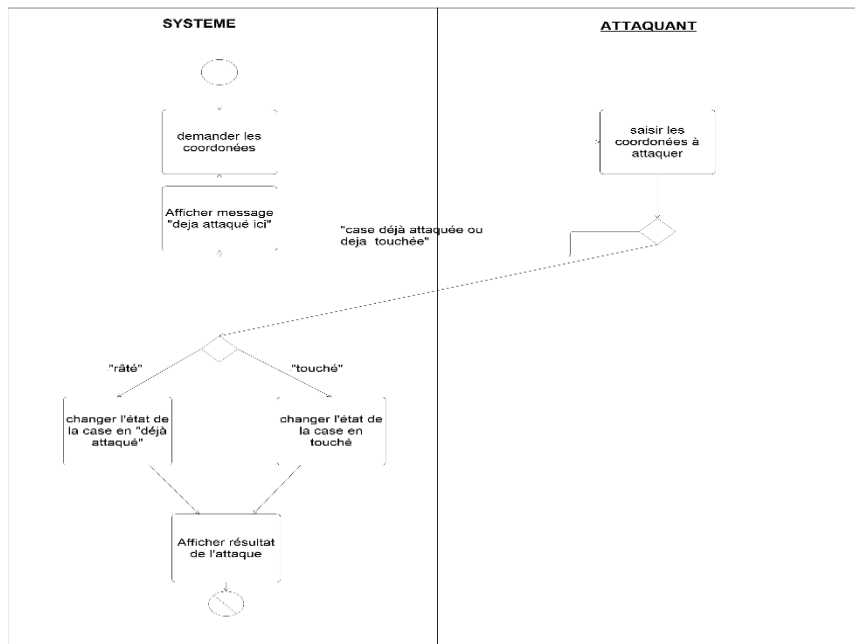




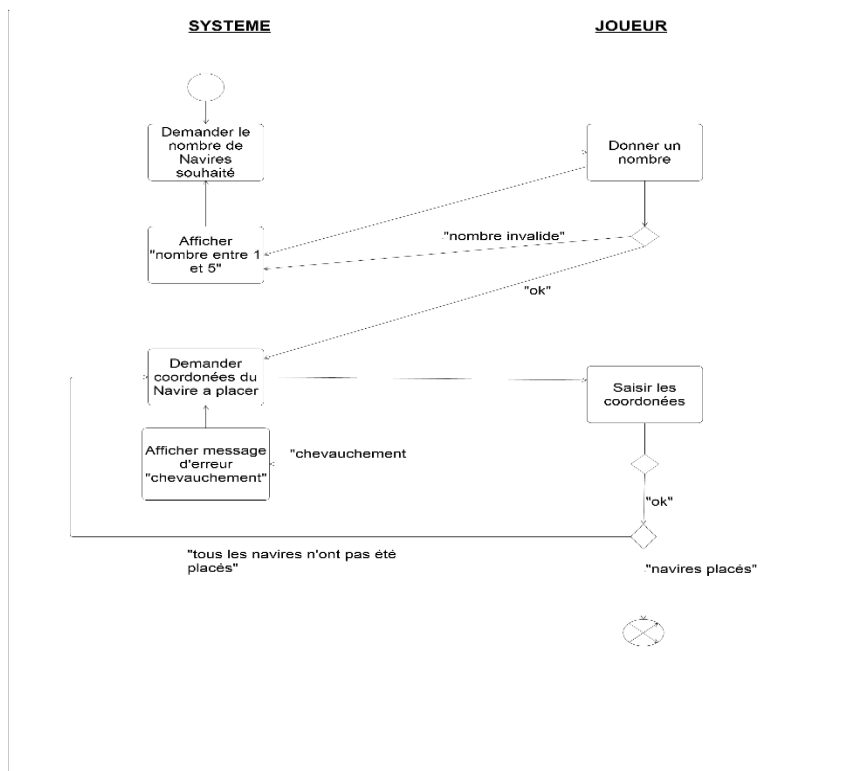
DESCRIPTION DE L'USE CASE "CHOISIR  
HUMAIN VS HUMAIN"



# DESCRIPTION DU CAS D'UTILISATION "ATTAQUER"

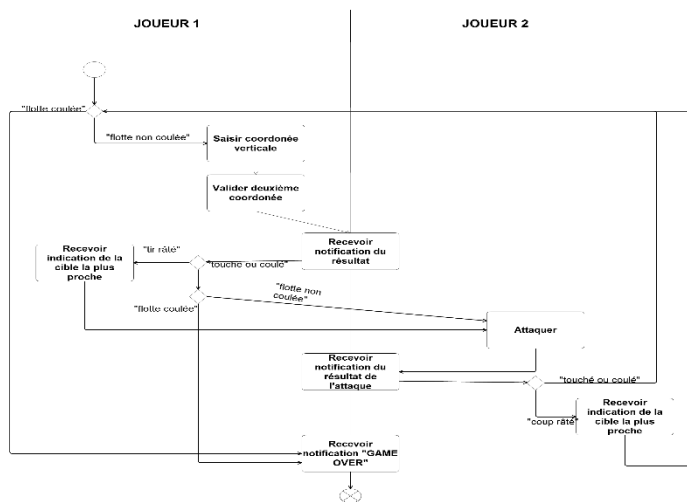


# DESCRIPTION DU CAS D'UTILISATION "CHOISIR ET PLACER NAVIRES"

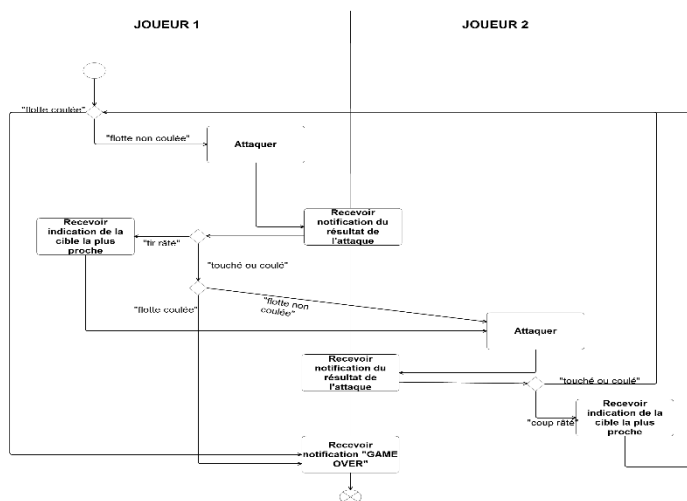


## Description des différents types de jeux différents.

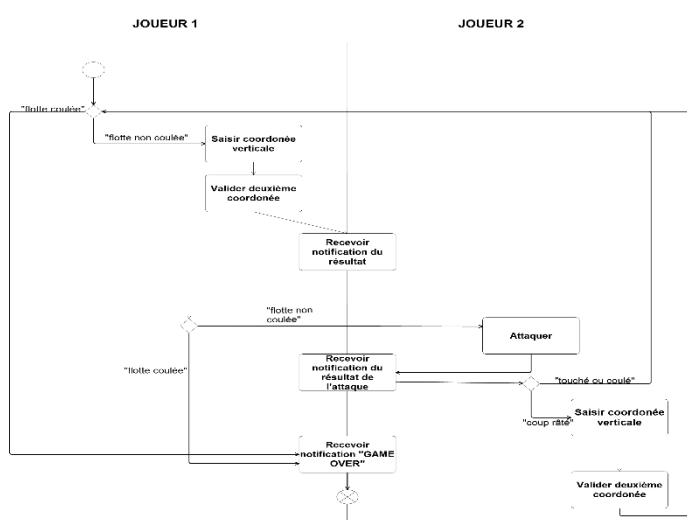
**DIAGRAMME D'UTILISATION ALERTE ROUGE**



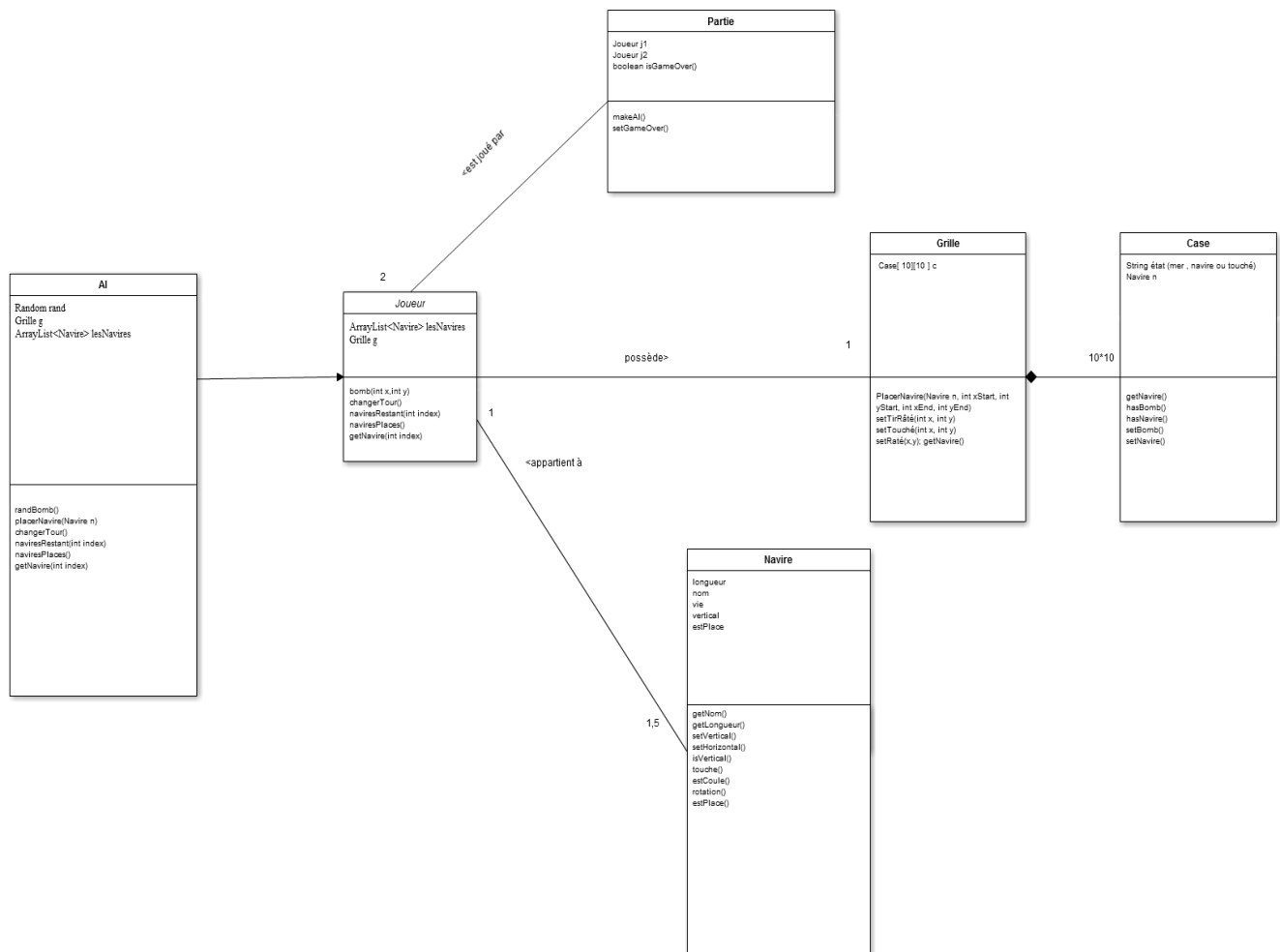
**DIAGRAMME D'UTILISATION TYPE MISSION RADAR**



**DIAGRAMME D'UTILISATION OPERATION ARTILLERIE**

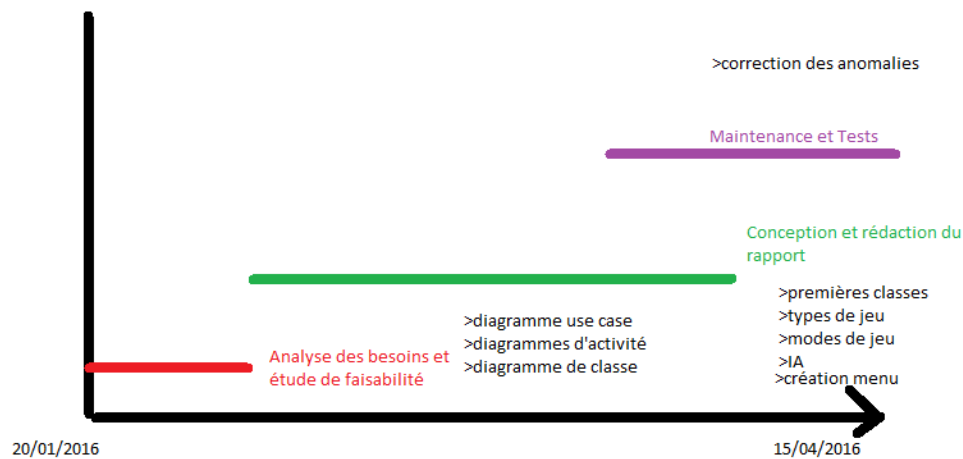


## Diagramme de classes



- Une **partie** est constituée de deux Joueurs.
- Chaque **Joueur** possède une flotte (ensemble de navires) ainsi qu'une grille propre à lui.
- Une **grille** est composée de cases.
- Chaque **case** a un état qui marque si elle est inoccupée, occupée par un Navire ou qu'elle a été bombardée.
- Un **Navire** a une longueur et une vie initialement également à sa longueur. Une fois que la vie tombe à 0 le Navire est considéré coulé.

### 3. Diagramme de Gantt simplifié



## III. Réalisation et implémentation

### 1. Choix des moyens pour la réalisation

Pour la réalisation de ce projet nous avons choisi de coder en **JAVA** car les membres de notre groupe sont des habitués de ce langage. Mais surtout car java est entièrement orienté objet et permet d'avoir une application bien structurée, modulable, maintenable beaucoup plus facilement et efficace. Sans oublier la richesse de l'API java qui nous permet de réaliser plusieurs choses et nous évite de recoder des fonctionnalités basiques.

Pour cela nous avons choisi **Eclipse** pour l'implémentation du code.

### 2 Mode d'emploi

Notre application offre une **interface simple et très facile à utiliser** pour l'utilisateur.

- A. Le menu de sélection de mode de jeu apparaît. SAISIR un nombre entre 1 et 3 dans la console selon le mode souhaité.
- B. Le menu de sélection de type de jeu apparaît. SAISIR un nombre entre 1 et 4 dans la console selon le type souhaité
- C. Les bateaux disponibles s'affichent. Saisir le nombre selon le bateau. Son nom et sa taille s'affichent et saisir sa coordonnée verticale (de A à I)
- D. Saisir la coordonnée horizontale puis choisir la position parmi celles qui sont possibles
- E. Pour attaquer saisir la coordonnée verticale (de A à I) puis la coordonnée horizontale (nombre)

#### Aperçu de la grille

```
Grille J1
- A B C D E F G H I
0 0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0 0
3 0 0 0 0 0 0 0 0
4 0 0 0 0 2 0 0 0
5 0 0 0 0 0 0 0 0
6 0 2 0 0 0 0 0 0
7 0 0 0 0 0 0 0 0
8 0 0 0 0 0 0 0 0 |
9 0 0 0 0 0 0 0 0
```

Les coordonnées horizontales de la grille sont des lettres (de A à I) et les coordonnées verticales sont des nombres allant de 0 à 9. La présence d'un bateau sur une case est marquée par un 1 et les tirs ratés sont marqués par des 2.

Chaque joueur humain peut avoir accès à l'état de sa grille après chaque tir.

## IV-Difficultés rencontrées et bilan

- Au début du projet, nous avons voulu nous baser sur un travail fait l'an dernier pour réaliser le code. L'interface graphique était déjà faite, il nous suffisait de rajouter différentes fonctionnalités. Cependant ce code était rempli de défauts et on n'arrivait pas à adapter nos besoins à celui-ci dû à sa complexité.  
Solution : Nous avons laissé tomber le code et décidé de commencer un autre qui serait mieux organisé et qui marcherait d'abord en mode console. Au moins pour s'assurer d'avoir quelque chose de bien structuré et qui fonctionne avant d'entamer l'interface graphique.
- Au niveau du code, le placement de bateau manuel est assez complexe. Il n'est pas facile de gérer toutes les données et surtout celles acheminées entre les différentes fonctions.
- Il est difficile de définir quand un navire a été coulé. Le problème est à moitié réglé.
- Plus il y a des bateaux IA, plus il y a des anomalies dans l'exécution.

## V-Conclusion

Faire ce projet nous a permis de nous habituer au travail en groupe et aux notions de projet, à organiser notre temps et les tâches selon les capacités de chacun.

Coder une bataille Navale avec autant de fonctionnalités n'était pas une chose facile. En effet, on peut repérer quelques anomalies dans le code et il manque aussi l'interface graphique. Cependant au moins nous avons pu établir un programme de base fonctionnant correctement en mode console qui répond à nos besoins et qui présente une interface simple tout en offrant un mode d'utilisation facile.