

Projet bataille navale

M1 informatique

Diagrammes de conception

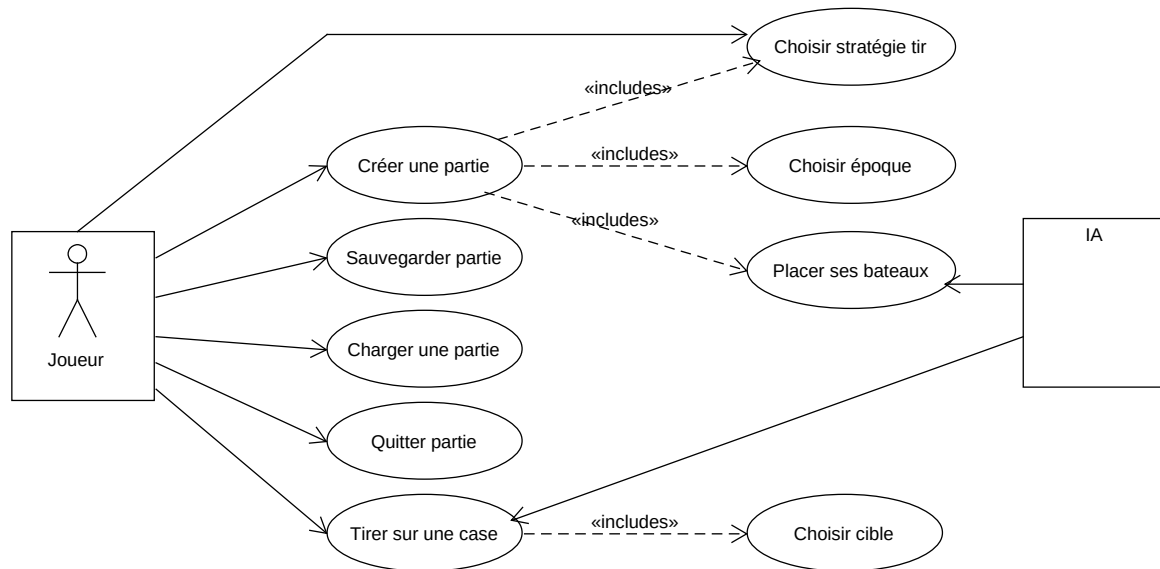
Design Pattern

BELLANGER Clément

LANUEL Charlotte

SCHWAB Lucas

## Diagramme de cas d'utilisation

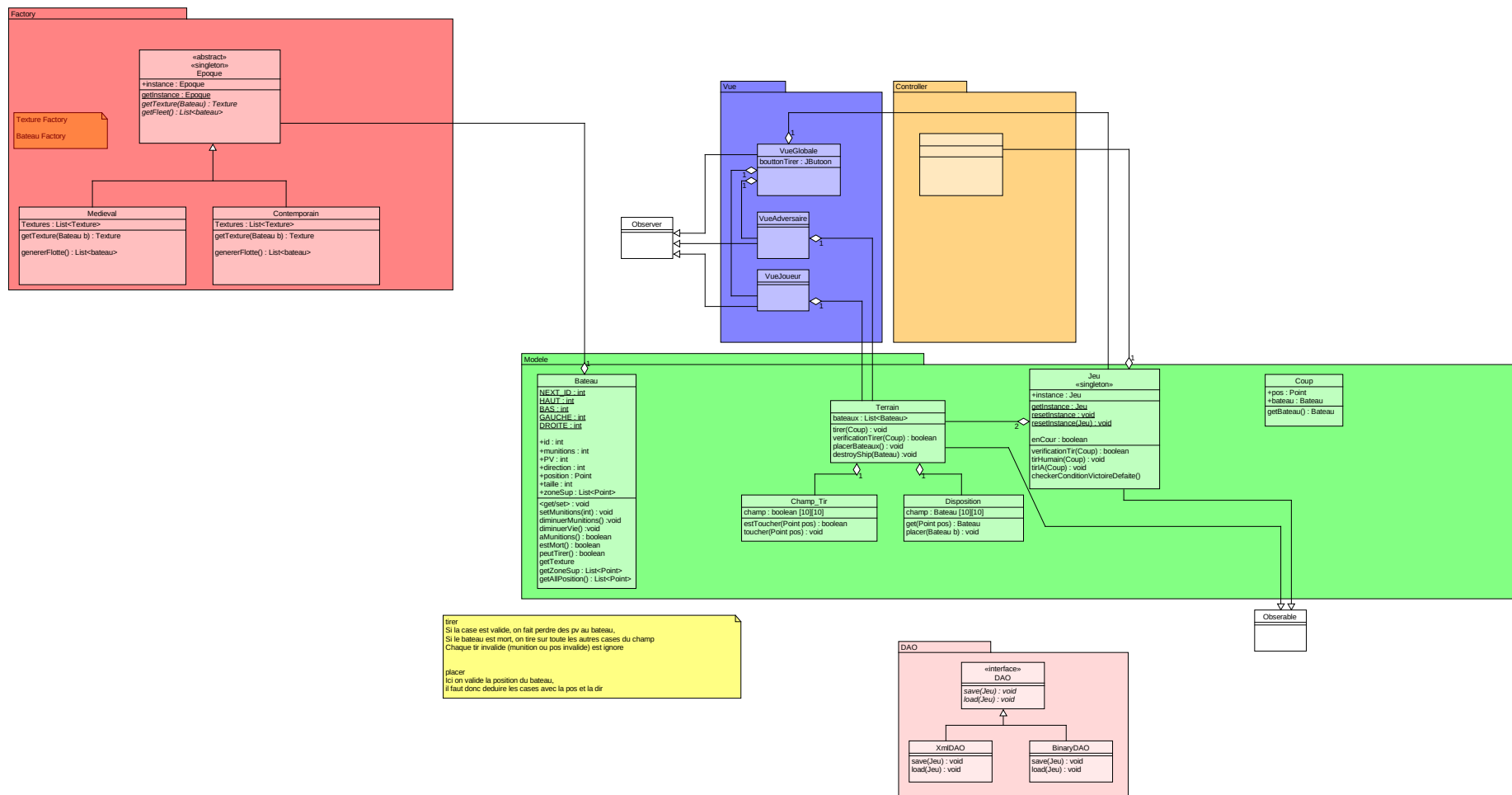


Les cas d'utilisations sont très simples. Un joueur a pratiquement tous les contrôles. Il peut créer une partie et ainsi la paramétrer, jouer (toutes les actions possibles), changer les paramètres en cours de partie, enregistrer une partie et la charger.

La seule chose que l'IA pourra faire automatiquement, une fois la partie créée par le joueur, c'est placer ses bateaux et tirer sur une case en espérant toucher un des bateaux du joueur.

## Diagramme de classes

Le diagramme de classes suivant (page suivante) va montrer la structure que le projet aura. Il n'est cependant pas définitif et pourra être modifié en cours de développement. Il montre néanmoins la base sur laquelle nous allons nous appuyer pour nos développements.



Les principaux design patterns utilisés dans cette application sont :

- Le DAO (Data Access Object qui va permettre de sauvegarder la partie. Nous pensons implémenter deux méthodes de sauvegarde : une en XML et une en binaire.
- Le MVC (Modèle Vue Contrôleur) car il est plus prudent et plus pratique de séparer la Vue, donc la partie que le joueur va voir et interagir avec, les contrôles (est-ce que le bateau peut tirer, est-ce que la case est valide...) , du Modèle (le nombre de cases du bateau, le nombre de missiles disponibles...). Le Controller ne sera impliqué que pour vérifier que les demandes du joueur (et de l'IA pour le choix du bateau et du tir) sont conformes à ce qui peut être fait : on ne pourra pas choisir une case où on a déjà tiré, on ne peut pas tirer avec un bateau mort ou qui n'a plus de munitions.

Les Vues (vue IA et vue Joueur) sont séparées pour simplifier les actions à effectuer et les événements qui en découleront. Chaque vue contient un Terrain (une map qui contient une liste de bateaux avec leurs emplacements). C'est elle qui va permettre de tirer et de « tuer » un bateau.

- Un Observateur-Observable (lié au MVC) qui servira à faire le lien direct entre les classes du modèle : Jeu et Terrain en particulier et leurs vues. A chaque changement des classes du modèle, les vues se mettront à jour.
- Une Factory qui va créer les textures des bateaux en fonction de l'époque choisie par le joueur. La classe Factory est abstraite et sera complétée par les classe Contemporain et Médiéval qui pourront générer une flotte de bateaux avec les textures correspondantes. Générer une flotte signifie créer les bateaux (taille 2-3-4-5) avec la texture correspondante. Au joueur ensuite de les placer en début de partie grâce à la classe Disposition, qui hérite de Terrain.
- Enfin, les classes Jeu et Factory sont des Singletons, puisqu'il ne peut y avoir qu'un seul jeu à la fois et une seule instance de Factory peut connaître toutes les textures à donner aux bateaux.

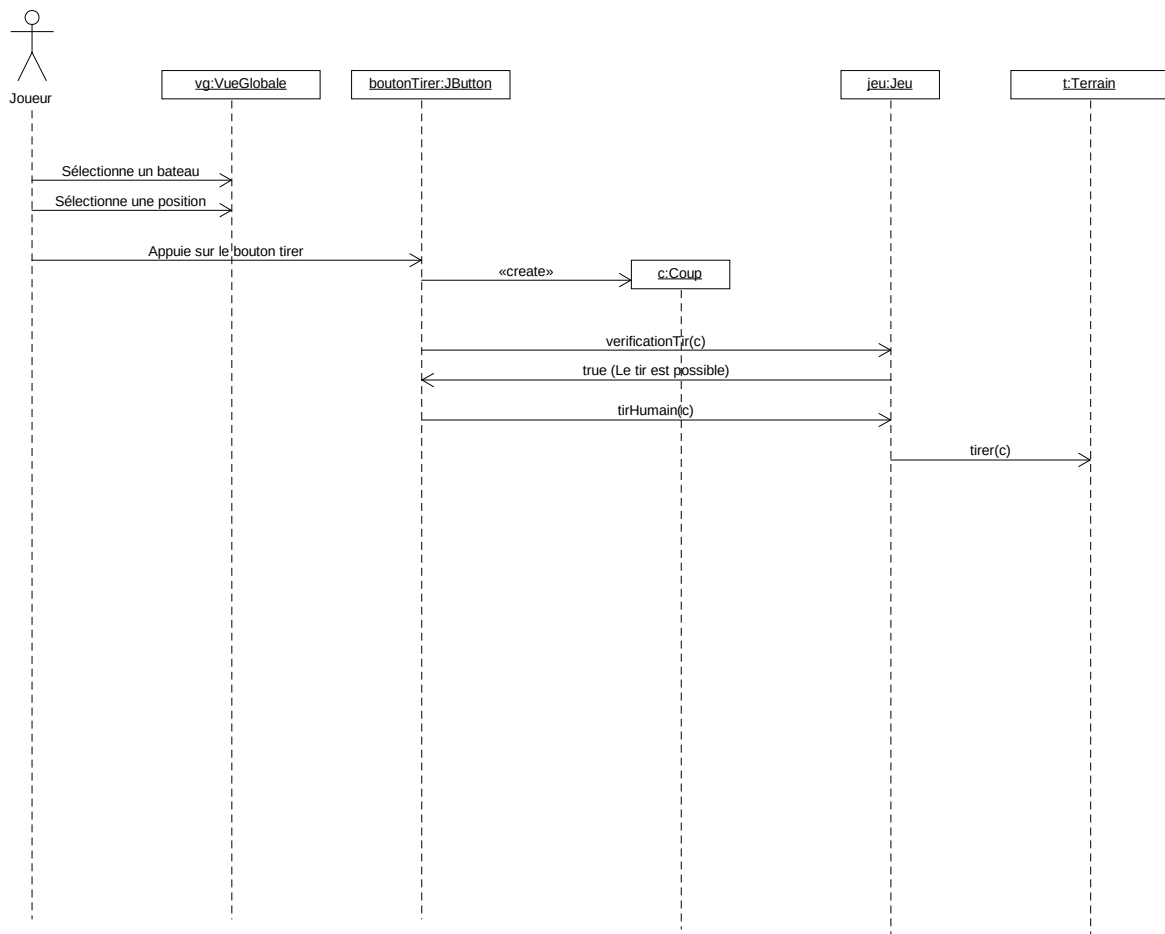
## Diagramme de séquence

Pour les diagrammes de séquence, nous allons d'abord présenter le fonctionnement d'un tir généré par l'appui du bouton tirer par un humain. Ce diagramme présentera un exemple et n'entrera pas dans les détails mais permettra d'avoir une vision claire.

Ce diagramme utilisera la méthode « vérificationTir » de la classe Jeu et « tirer » de la classe Terrain.

Ces deux méthodes seront présentées en détails, chacune ayant son propre diagramme de séquence.

### Le tir effectué par un joueur

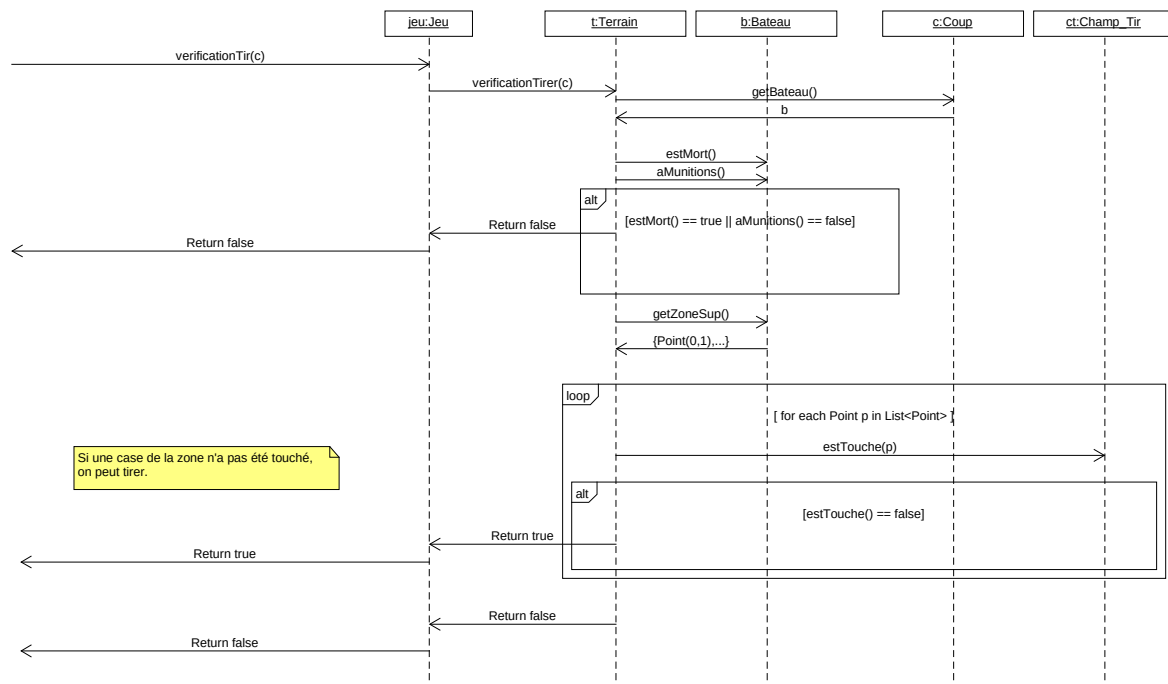


Le diagramme expliquant le tir effectué par un joueur est très simple.

En premier lieu, il faut sélectionner le bateau avec lequel on veut tirer, ensuite sélectionner sur la grille adverse l'emplacement du tir. Une fois fait, le joueur clique sur le bouton Tirer. Ce bouton va déclencher la création d'une instance de Coup qui contiendra le bateau et le tir qu'on veut effectuer (voir son utilité dans le diagramme de séquence de vérification de tir). La première méthode appelée ensuite est `verification(Coup)`. Suivant le retour (vrai ou faux) :

- Si vérification renvoie faux, le joueur doit soit sélectionner un autre bateau (le sien est mort ou n'a plus de munition), soit sélectionner une autre position (celle choisie ayant déjà servie à un tir précédent).
- Si vérification renvoie vrai, alors le Jeu va exécuter les tirs et mettre à jour la grille adverse (au niveau du modèle).

## Vérification tir

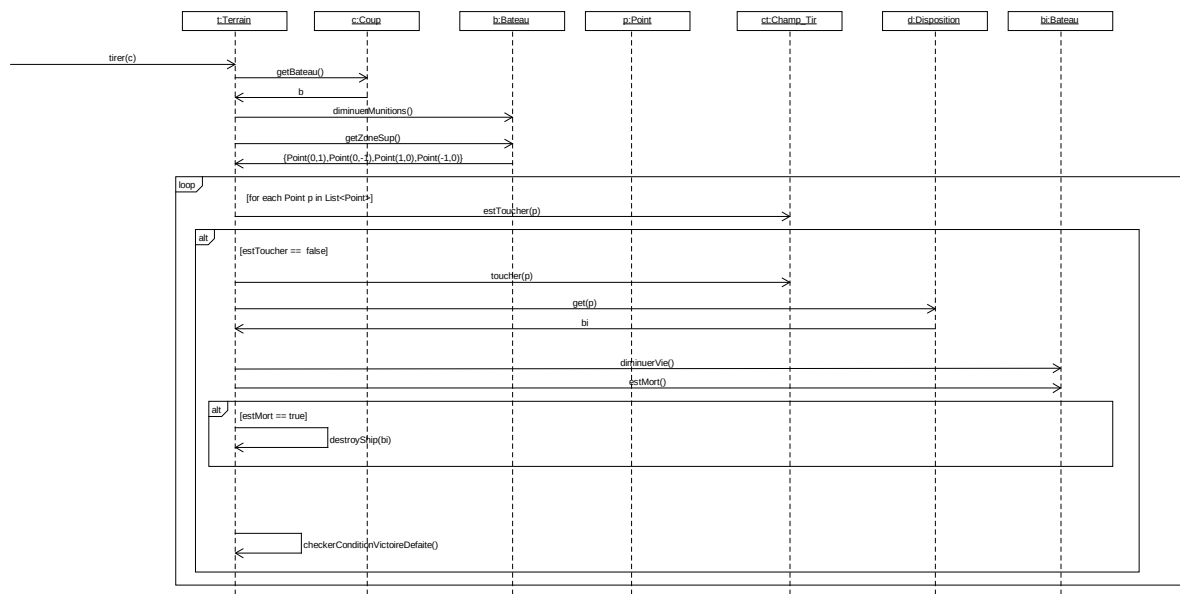


La méthode `vérificationTir` va prendre en paramètre un `Coup` (qui regroupe en fait les données du bateau sélectionné pour tirer et la case sur laquelle on veut tirer). `vérificationTirer` fait partie de la classe `Terrain` et va commencer par appeler la méthode `getBateau`. Le but ici est de vérifier les paramètres du bateau. S'il est mort (si `estMort()` renvoie vrai) alors la méthode `tirer` va renvoyer faux et le tir ne sera pas validé. S'il n'a plus de munitions il ne pourra pas tirer non plus et `tirer` renverra faux. `getZoneSup()` est une méthode que nous avons conçu pour un mode de jeu où les gros bateaux

(5 cases) puissent tirer en zone. Ainsi on ne gèrera pas un tir à la fois mais bien une liste de plusieurs tirs (un même bateau mais plusieurs positions différentes).

Maintenant que les bateaux sont aptes à tirer, il faut vérifier pour la ou les positions choisies qu'elles sont bien valides : si on a tiré à tous les emplacements de la zone, vérificationTirer() renverra faux. Si au moins une seule case de l'emplacement est vide alors vérificationTirer renverra vrai.

## Tirer



Pour tirer, on commence par utiliser le bateau contenu dans coup, on diminue ses munitions (ce qui peut être fait sans problème car on a vérifié le tir avant).

On va chercher la zone de tir du bateau et pour chaque point de celle-ci (et le point central) (les points étant relatifs à la position du point central choisi dans coup), on va toucher les cases du champ de tir. Si celui-ci contient un bateau on va diminuer sa vie et si il est mort, toutes les cases du champ de tir qui contiennent le bateau seront touchés.

On va ensuite vérifier d'abord les conditions de victoire du joueur (si il a coulé tous les bateaux ennemis) puis de défaite (si il a utilisé sa dernière munition).