



utt

---

université de technologie  
Troyes

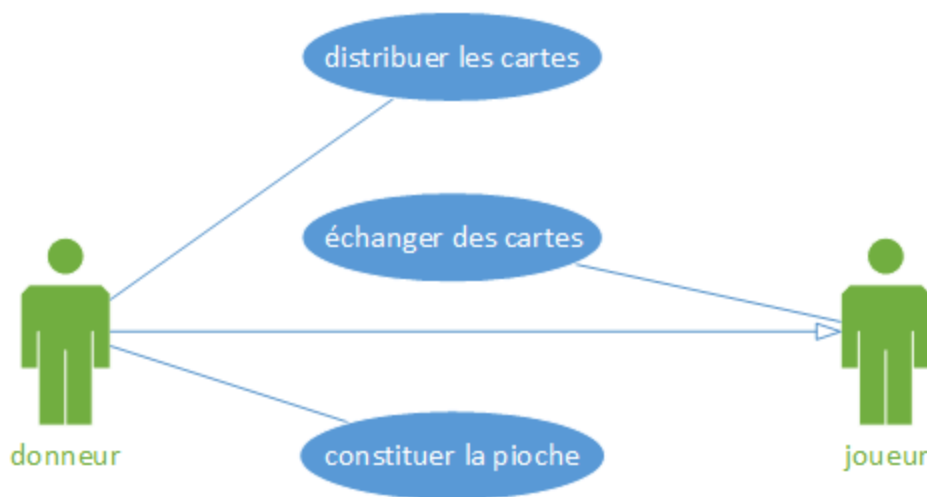
# Rapport LO02

*Livrable n°3 | “La Bataille Norvégienne”*

## Diagrammes de cas d'utilisation

Nous avons choisi de décomposer le jeu en deux parties : la préparation du jeu, et le jeu en lui-même afin de garder des diagrammes clairs et précis. Les diagrammes de cas d'utilisation n'ont pas évolués. L'étude faite lors de la première modélisation UML représente bien le déroulement d'une partie.

### *La préparation*



*Diagramme de la préparation de la Bataille Norvégienne*

---

Le donneur est un joueur désigné, c'est pour cela qu'il y a une notion d'héritage entre les deux protagonistes. Le premier a donc pour rôle de distribuer les cartes aux autres joueurs, quant aux second, il a seulement la possibilité d'échanger ses cartes. Une fois que la distribution est terminée le donneur a aussi la possibilité d'échanger ses cartes puisqu'il est aussi un joueur.

## Le jeu

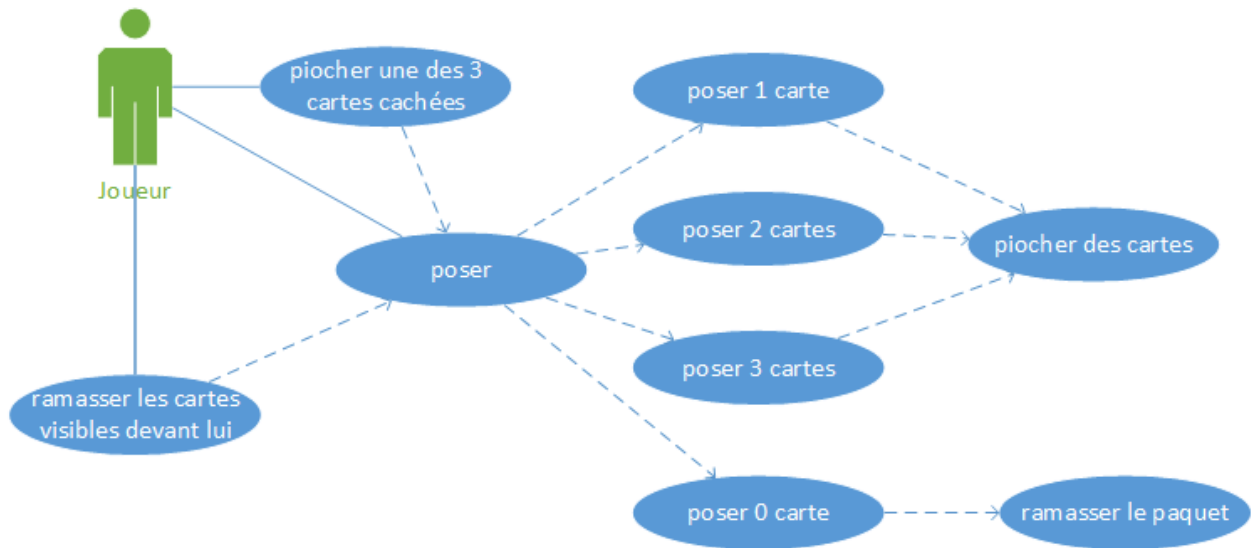


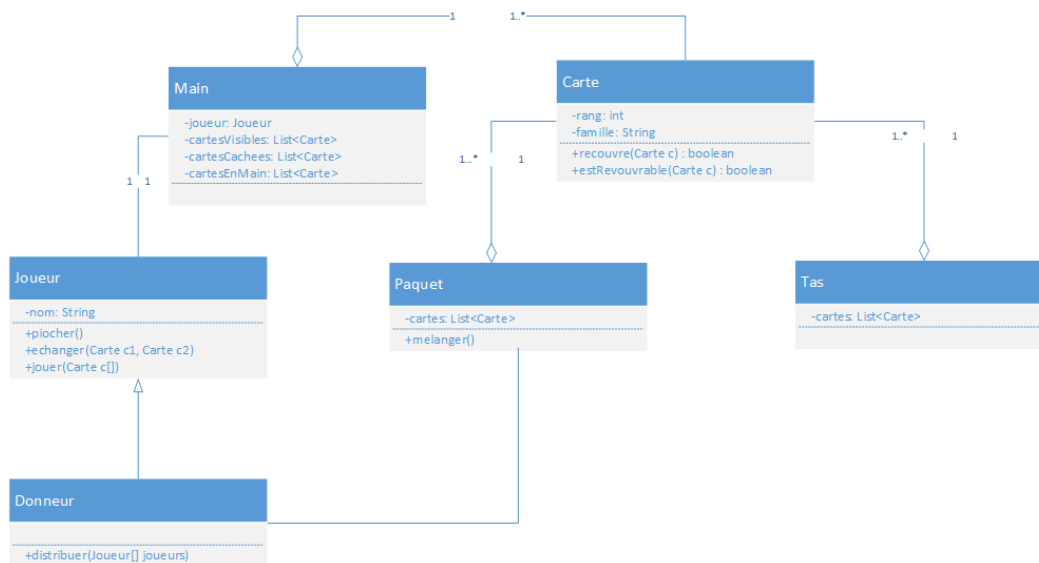
Diagramme d'une manche de la Bataille Norvégienne

Si le joueur a des cartes en main il peut essayer d'en poser entre une et trois, voire aucune si ses cartes ne sont pas supérieures ou égales à la dernière du tas. Poser une à trois carte(s) implique le fait de reprendre des cartes dans la pioche, jusqu'à ce que celle-ci soit épuisée.

Si le joueur a encore des cartes en mains mais qu'il n'a pas la possibilité de poser une seule carte, il ramasse le tas du jeu, qui lui constitue une nouvelle main.

Le joueur sera aussi amené à ramasser une à une, au fur et à mesure, les cartes visibles lui appartenant, et d'ensuite continuer à jouer en essayant de les poser. De même lorsqu'il devra piocher parmi ses trois cartes face cachées qui lui permettront de continuer à poser ses cartes.

## Diagramme des classes



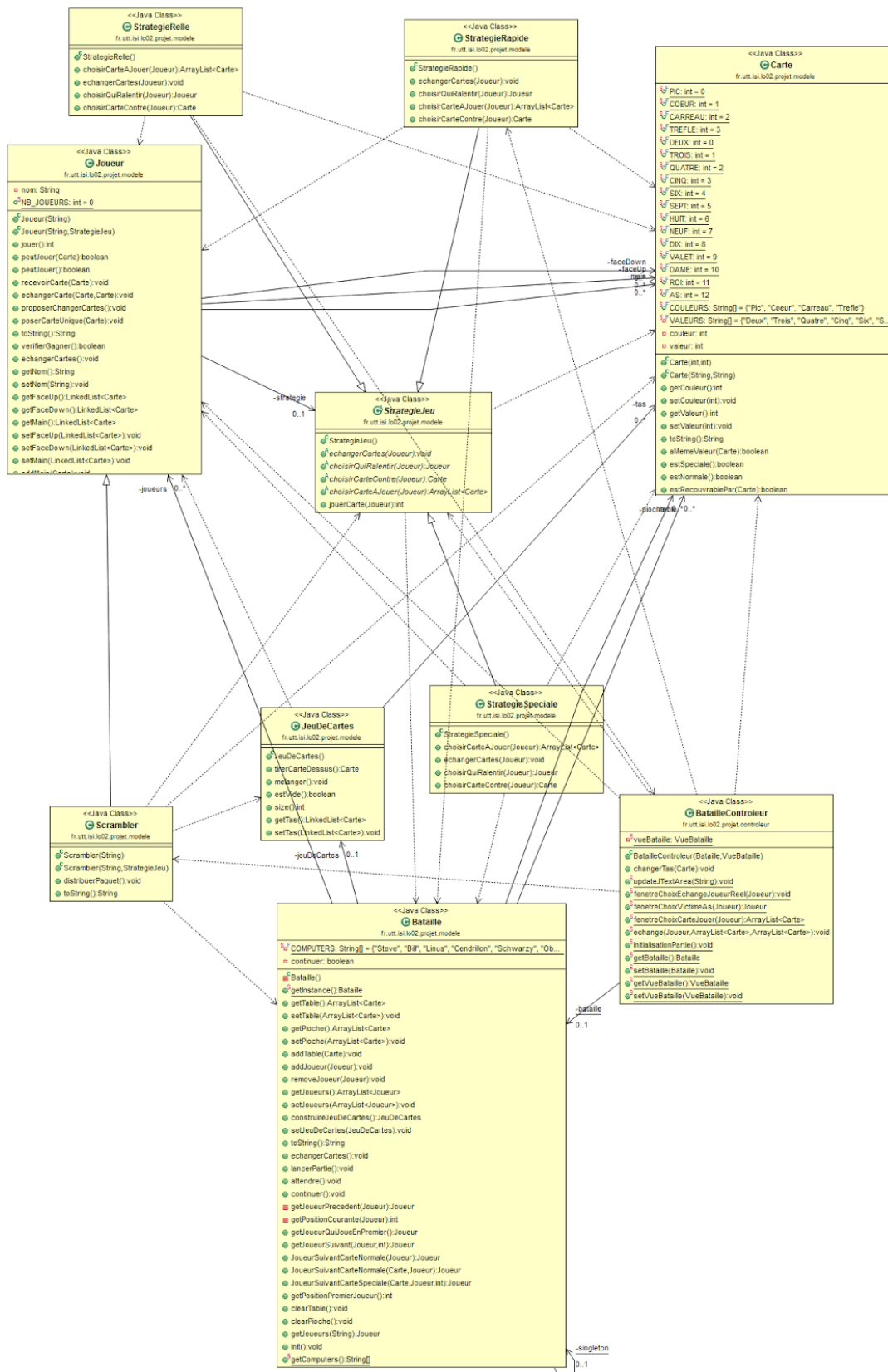
### Diagramme des classes de la Bataille Norvégienne

Ci-dessus un rappel du diagramme des classes initial, réalisé lors de notre phase de réflexion. Ce diagramme nous a beaucoup aidé à la genèse de notre code mais nous nous sommes très vite rendu compte qu'il était incomplet et grandement améliorable.

L'objet "Main" n'était pas justifié, nous avons donc finalement décider de placer les attributs de cette classe directement dans la classe "Joueur".

Les objet "Paquet" et "Tas" n'étaient pas assez pertinent. Etant tous les deux des listes d'objet type Carte, il a été décidé d'en faire des attributs d'un objet appelé Bataille. Cet Objet de type Bataille regroupe tout ce qui est utile au déroulement d'une partie, comme le Tas, la Pioche, la liste des joueurs, et le Jeu de Carte (cette fois-ci, bien qu'étant une simple liste de Carte, nous avons créé un objet afin de pouvoir créer les méthodes dont nous avons besoin) destiné à être distribué entre les joueurs. Bataille suit le modèle Singleton, afin de garantir son unicité.

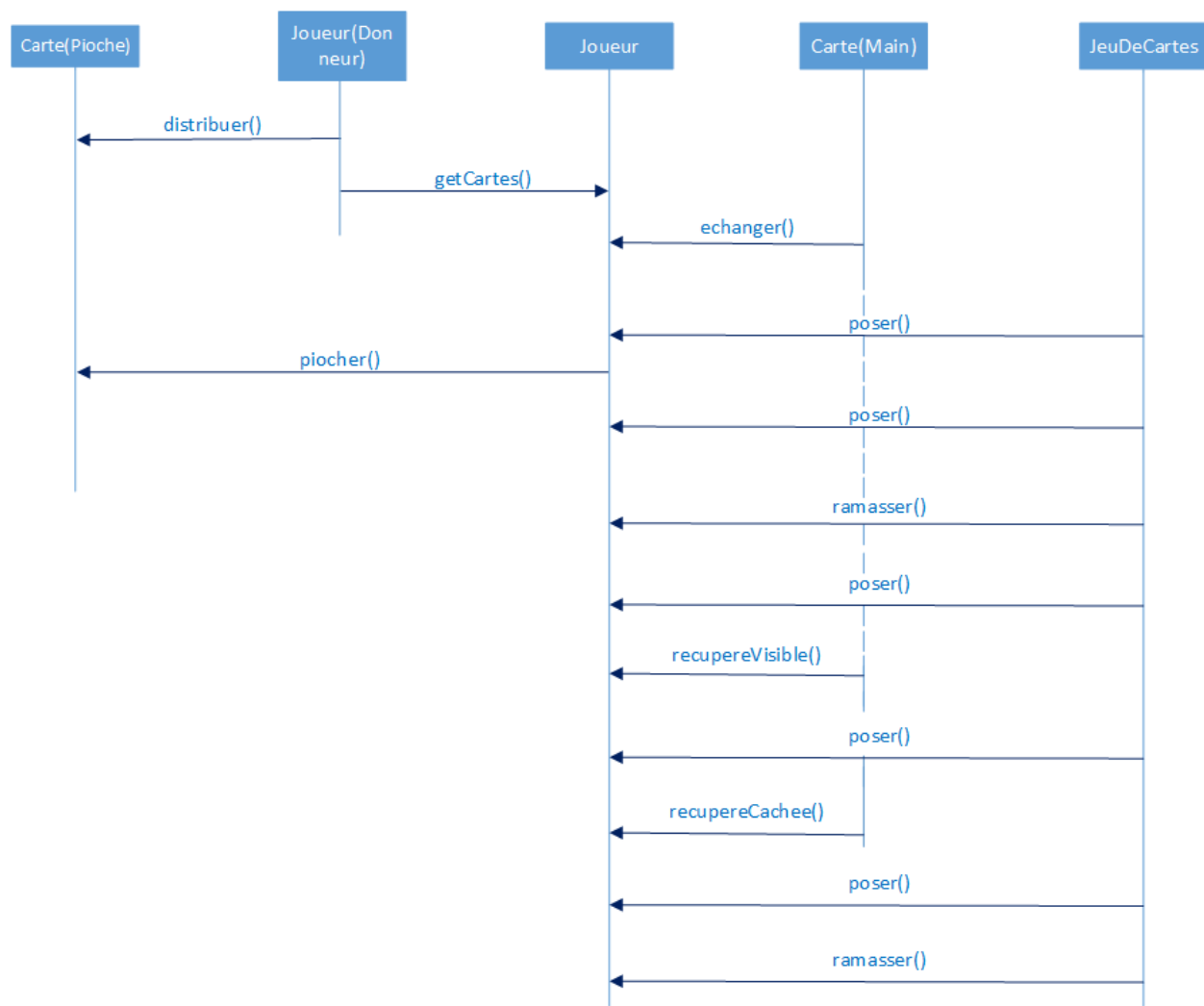
Par la suite, afin de réaliser au mieux une interface graphique, nous avons adopté la structure M(odèle) V(ue) C(ontroleur), ce qui a considérablement augmenté le nombre de classes et de méthodes ainsi que la création de packages. Un diagramme de classe de cette envergure, comme ci-dessous, est assez difficile à lire, mais témoigne d'une bonne compréhension et utilisation de la programmation Orientée Objet.



Nous pouvons effectivement voir que notre analyse préalable est très différente de notre analyse finale. Et pourtant, c'est bien grâce à cette première analyse que nous avons pu amorcer notre code. C'est au cours du développement de l'application que nous nous sommes rendu compte avoir besoin de davantage de méthodes ainsi que d'autres classes.

Le diagramme de classes final étant très volumineux, il est disponible en annexe, en fichier *.png*.

## Diagramme de séquence



## *Diagramme de séquence de la Bataille Norvégienne*

---

Le diagramme de séquence final n'a pas évolué depuis l'analyse initiale. Les seules modifications faites sont des corrections apportées par rapport au compte-rendu de notre chargé de TP.

Le joueur donneur est sélectionné depuis l'interface du jeu. A partir de là l'objet donneur est créé et doit distribuer les cartes, les joueurs récupèrent respectivement neuf cartes chacun (trois faces cachées, trois faces visibles et trois en main). Après la distribution de toutes les cartes l'objet main est créée, les joueurs ont la possibilité après d'échanger les cartes qu'ils ont en main avec celles faces visibles.

La partie commence après que le joueur à gauche du donneur ait posé une ou plusieurs cartes de même valeur, il doit piocher après le même nombre de cartes posées, à leurs tours les joueurs suivants posent une ou plusieurs cartes de même valeurs et de valeurs supérieures à celles posées par le joueur précédant, ils piochent après le même numéro de cartes posées, quand un joueur n'a plus de cartes à jouer il ramasse le tas.

Lorsque la pioche est épuisée les joueurs continuent à jouer jusqu'à ce qu'ils vident leurs mains et récupèrent les cartes faces visibles devant eux, s'ils vident leur main encore ils piochent au hasard une des trois cartes faces cachées jusqu'à ce qu'ils trouvent une carte à jouer, ils récupèrent le tas quand il n'y a plus de cartes à piocher.

Le joueur qui vide sa main (cartes en main, cartes visibles et cartes cachées) a gagné.

## Etat Actuel de l'Application

Toutes les règles de la Bataille Norvégienne ont été implémentées. Le jeu se déroule sans encombre jusqu'à ce qu'un joueur gagne la partie. Deux types de joueurs virtuels existent : le premier va jouer en priorité ses cartes les plus petites, le second va jouer en priorité ses cartes spéciales. Lorsque ces joueurs jouent un As ils choisissent le joueur qui possède le moins de carte dans sa main (sans se choisir eux-même).

Le jeu fonctionne entièrement avec une interface graphique. Les cartes sont représentées de manière graphique d'après des images libres trouvées sur internet. A chaque fois qu'un joueur pose une carte, celle-ci apparaît sur le tapis. Si le tas des cartes jouées est vide (suite à la pose d'un 10 par exemple), ou si la pioche est vide, les cartes sont enlevées du tapis.

Lorsque le joueur réel doit jouer, une fenêtre (de type JDialog) s'ouvre et lui liste toutes ses cartes. Avec sa souris il lui est possible de sélectionner une ou plusieurs cartes. Si elles ne peuvent être jouées, le joueur ne pourra pas valider sa sélection. Si les cartes peuvent être jouées, elles sont posées sur le tapis. S'il reste des cartes dans la pioche, le joueur en pioche. Quand le joueur réel pose un As, une autre fenêtre (de type JDialog) s'ouvre, et il lui est demandé de choisir à qui il souhaite envoyer le tas (via une checkbox à cocher). Un bouton "Continuer" est présent sur l'interface afin de mettre le jeu en 'pause' et suivre chaque mouvement séparément. De plus, une zone de texte est présente afin de garder une trace de toutes les actions réalisées durant la partie.

## Conclusion

L'objectif de ce 3<sup>ième</sup> livrable était de modéliser le jeu « Bataille norvégienne » selon les principes de la méthode d'analyse UML et selon notre code existant, afin de le comparer à notre analyse préalable.

La différence qui existe entre nos diagrammes témoigne d'une évolution de l'application au fur et à mesure du temps. Nous nous sommes rendu compte que l'analyse préalable était indispensable afin de savoir comment amorcer le projet, mais que nous avons vite dû le faire évoluer pour le faire correspondre aux besoins du projet.