

Conception orientée objet et UML

Jeux : SCRABBLE

Réalisé par :

-CHARTIER LUCAS

-EL BOUHMI YASSMINE

-DAROUICH BILAL

-FETTOUHI KENZA

I. Présentation du jeu scrabble :

Le but du jeu du Scrabble est d'avoir le plus de points en créant des mots à l'aide de lettres piochées au hasard. En posant ceux-ci sur le plateau de Scrabble, le joueur obtient des points suivant la valeur des lettres et des cases sur lesquelles figure le mot.

Pour commencer, chaque joueur doit tirer au hasard 7 lettres dans le sac. Avec celles-ci, chacun va alors essayer de créer un ou plusieurs mots.

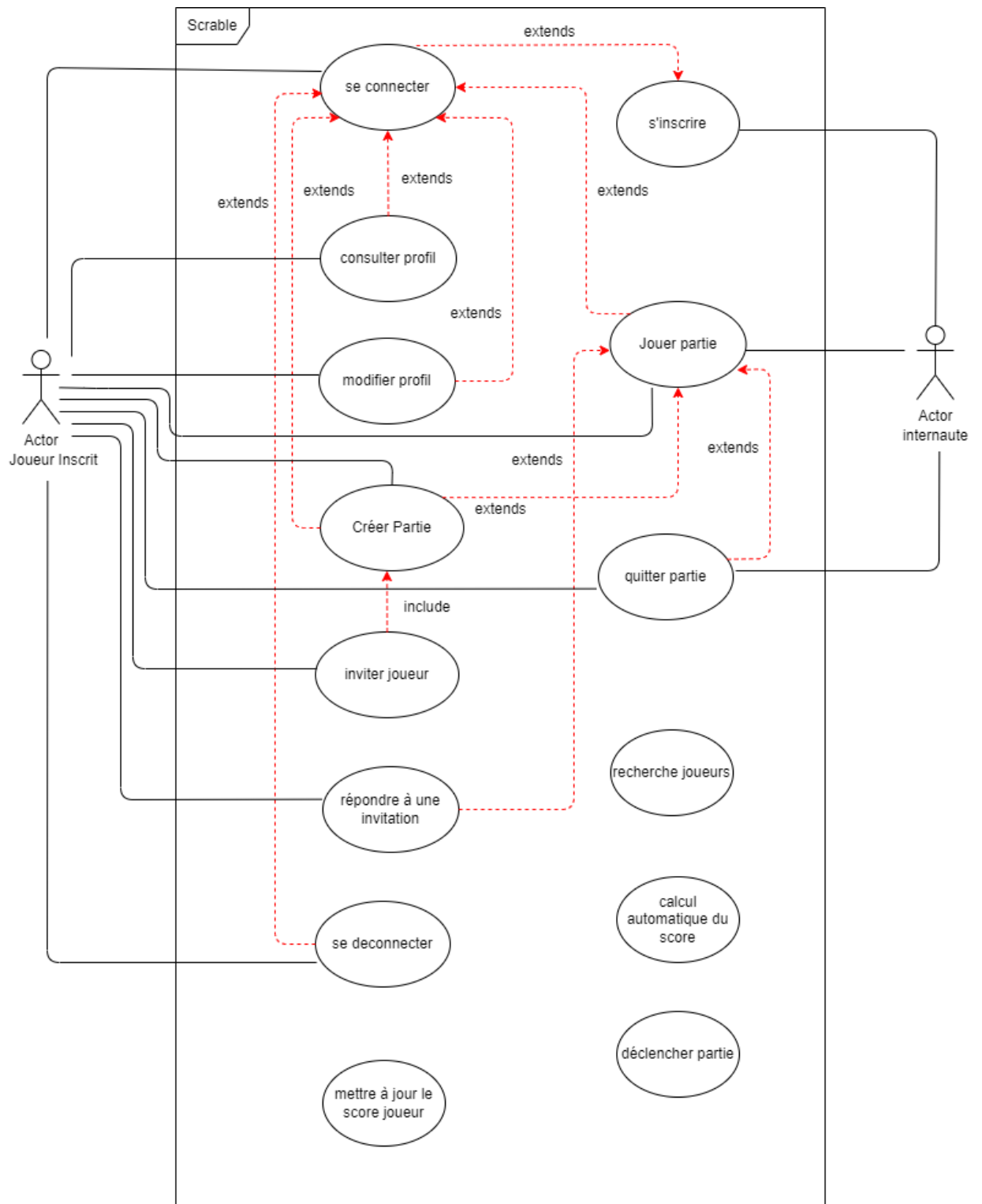
Le premier joueur doit obligatoirement poser le premier mot au centre du plateau, sur l'étoile. Le deuxième joueur doit s'appuyer sur le mot précédent pour placer le sien et ainsi de suite...

Un joueur peut échanger ses lettres par d'autres en piochant dans le sac mais cela lui fera obligatoirement passer son tour.

Un joueur peut choisir aussi l'action de passer son tour sans placer un mot et sans échanger ses lettres. A chaque mot placé dans le plateau, le score obtenu avec le mot s'additionne à celui du joueur. Quand le sac est vide et qu'un des joueurs pose toutes ses lettres, la partie est terminée.

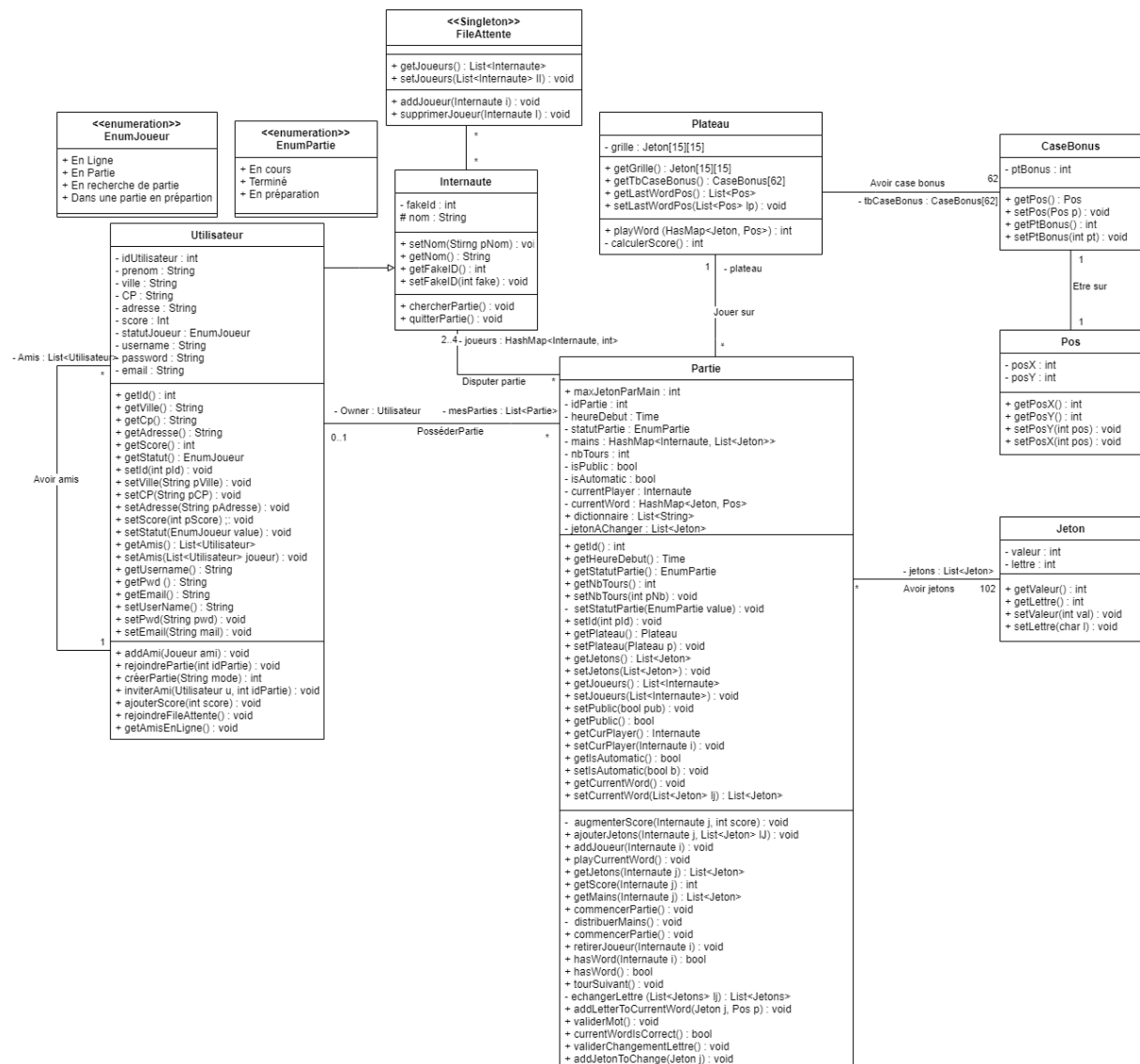
II. Les principaux diagrammes :

1. Diagramme cas d'utilisation :

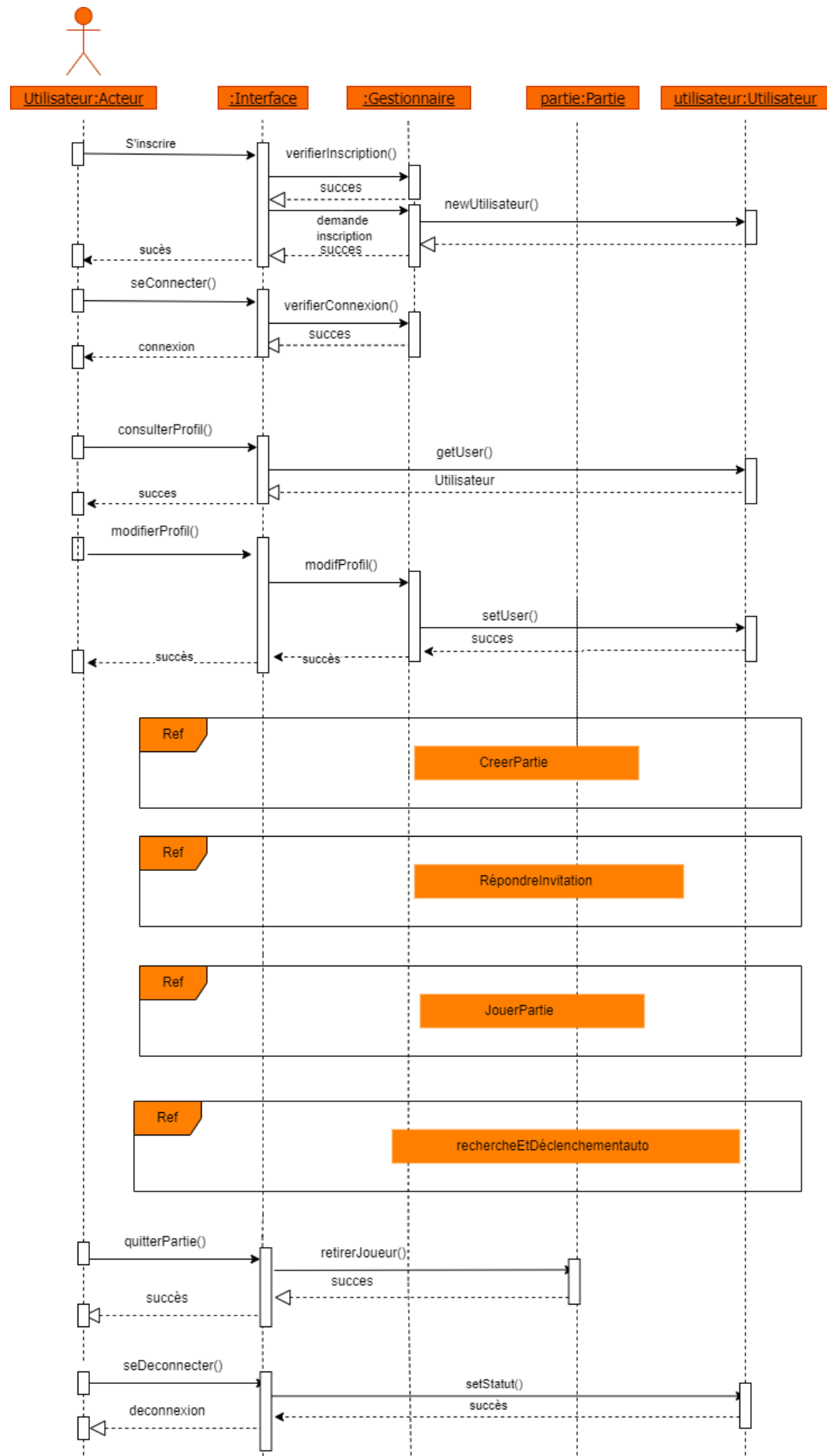


2. Diagramme de classe :

Le diagramme de classe représente toutes les classes de notre système. La file d'attente a été représentée par un singleton, un objet qui ne serait instancié qu'une seule fois durant tout le cycle de vie de l'application, les joueurs cherchant une partie sont ajoutés à une liste dans cet objet, puis en sont retirés lorsqu'ils entrent dans une partie. L'objectif ici était de limiter les appels vers la base de données dans le cas où le nombre de connexions serait important. Le champ dictionnaire est un champ statique. Au démarrage de l'application, la liste est générée depuis un fichier plat qui contient uniquement un mot par ligne et celle-ci est accessible globalement pour toutes les classes et dans toutes les parties. La fonction hasWord() renvoie vrai si au moins un des joueurs de la partie a un mot à poser, celle-ci est surchargée pour vérifier la main d'un joueur avant que celui-ci puisse commencer à poser des mots.



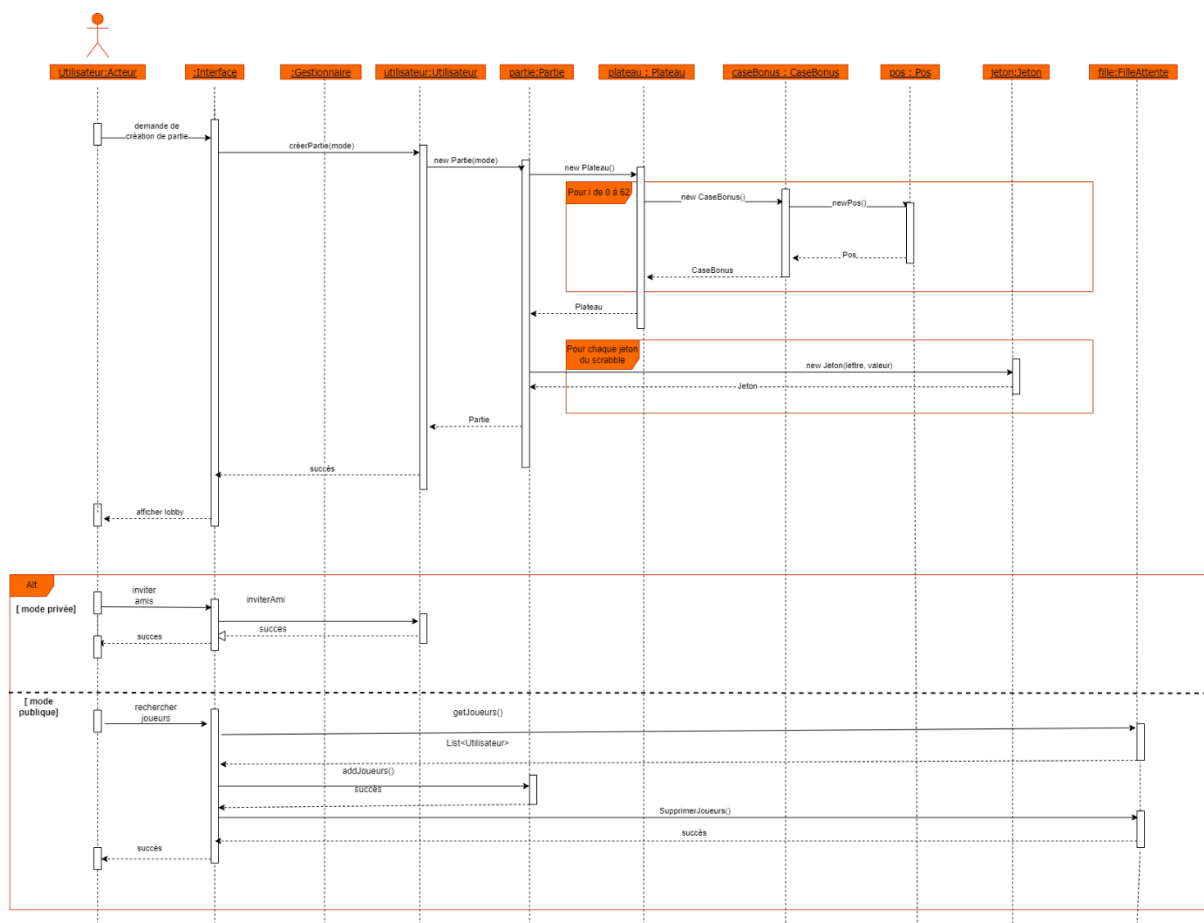
3. Diagramme de séquence général :



3.1 Diagramme de séquence : créer Partie

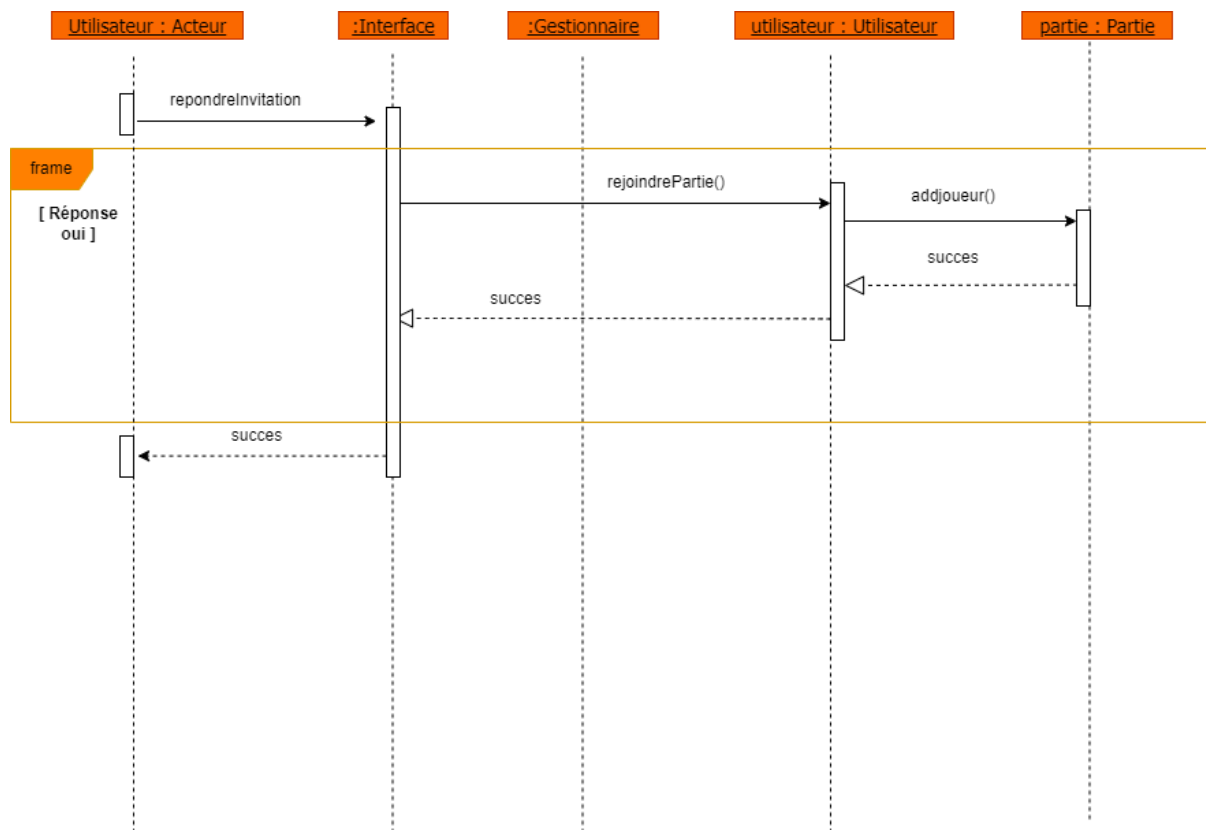
Créer une partie quel que soit son mode (privée ou publique) engendre la création du plateau de jeu, la création des 62 cases bonus avec leurs positions dans le plateau et aussi la création des jetons avec leurs lettres et valeurs.

Si le mode de la partie est privé, l'utilisateur invite ses amis et si le mode est public, 4 joueurs de la liste d'attente sont retirés pour être ajoutés dans la partie de jeu.



3.2 Diagramme de séquence : Répondre Invitation

En acceptant une invitation, l'utilisateur s'ajoute à la liste des joueurs de la partie.



3.3 Diagramme de séquence : jouée Partie

Quand la partie de jeu commence, les jetons se distribuent aux joueurs.

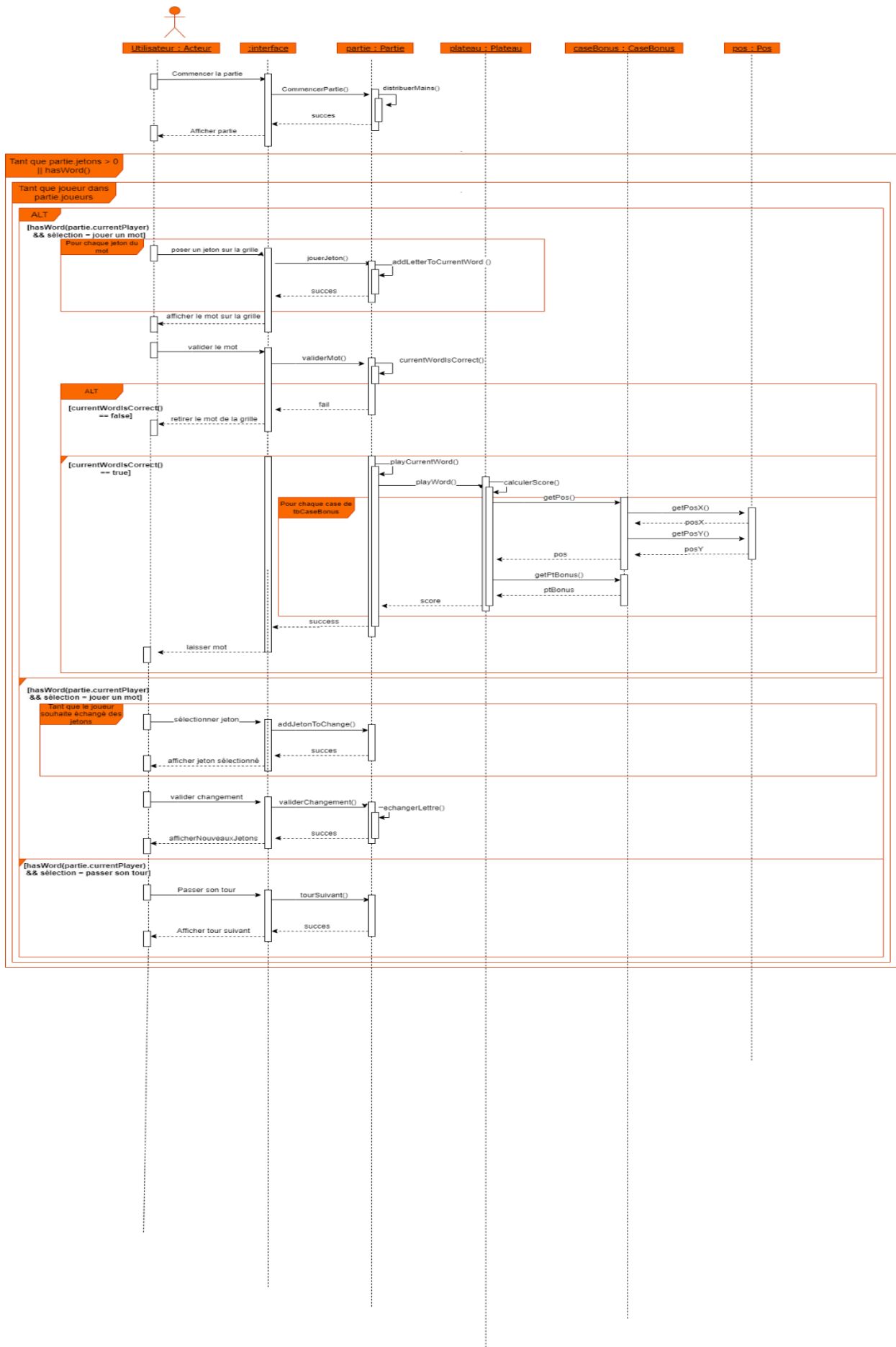
Tant que les conditions d'arrêt du jeu ne sont pas vérifiées (c'est-à-dire tant qu'il existe toujours des jetons dans la pioche ou tant que tous les joueurs ont un mot à jouer) :

Si le joueur en cours a un mot à jouer et choisit de le placer dans le plateau, dans ce cas le joueur place son mot jeton par jeton.

Après avoir fait ceci, une vérification se fait sur le mot placé : S'il n'est pas validé (c'est-à-dire s'il n'est pas correct), le mot se retire de la grille. S'il est validé, le score du mot se calcule en prenant en considération les points bonus des cases bonus s'il y en a pour le mot placé.

Si le joueur en cours choisit d'échanger ses jetons, dans ce cas il sélectionne les jetons qu'il veut et par la suite ils sont tous échangés avec de nouveaux jetons.

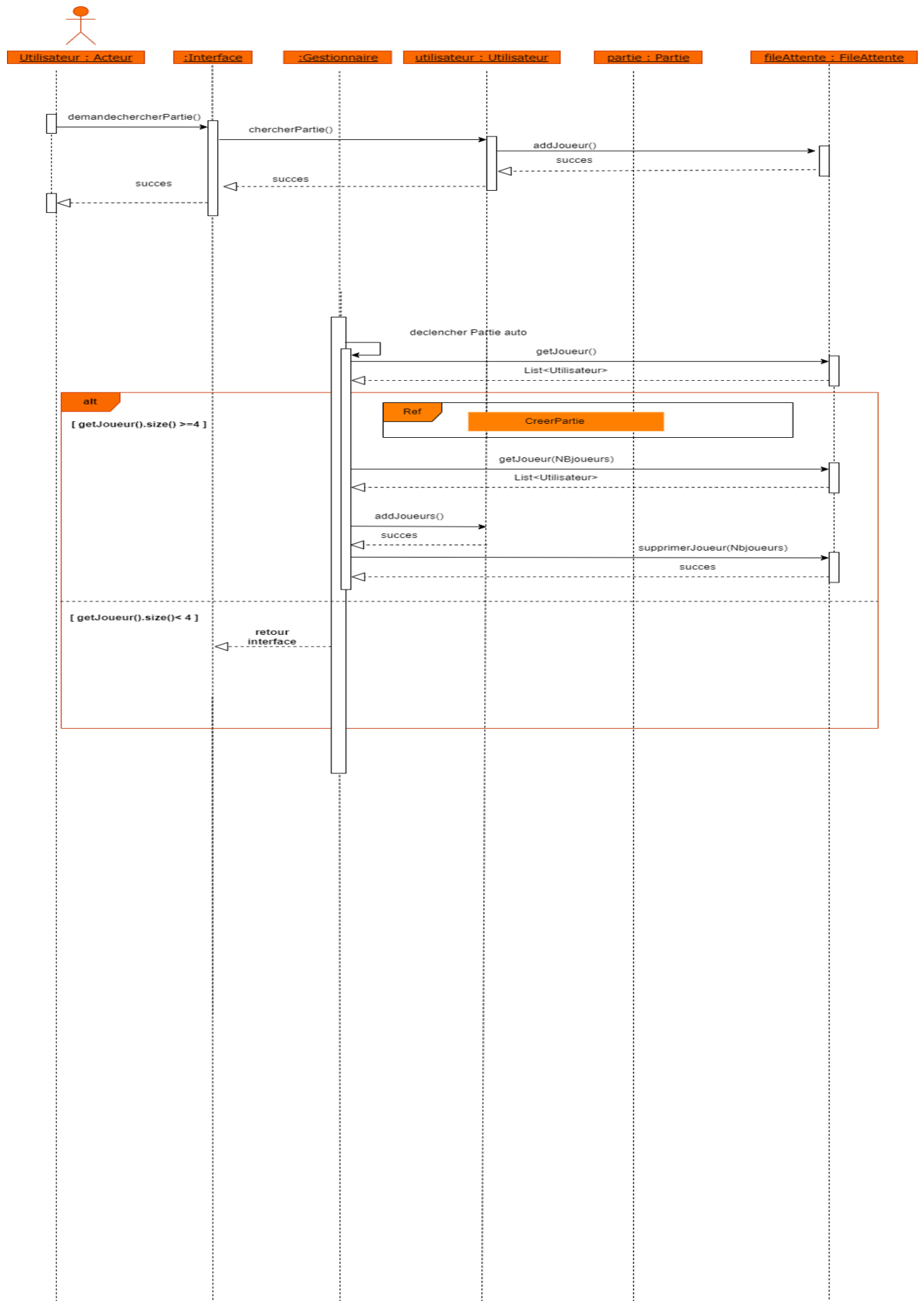
Si le joueur en cours choisit de passer son tour, dans ce cas le tour passe au joueur suivant.



3.4 Diagramme de séquence : Recherche et Déclenchement auto

Si l'utilisateur cherche une partie de jeu, celui-ci s'ajoute dans la file d'attente.

Un déclenchement automatique d'une partie de jeu en mode publique se fait une fois que la file d'attente rassemble 4 joueurs.



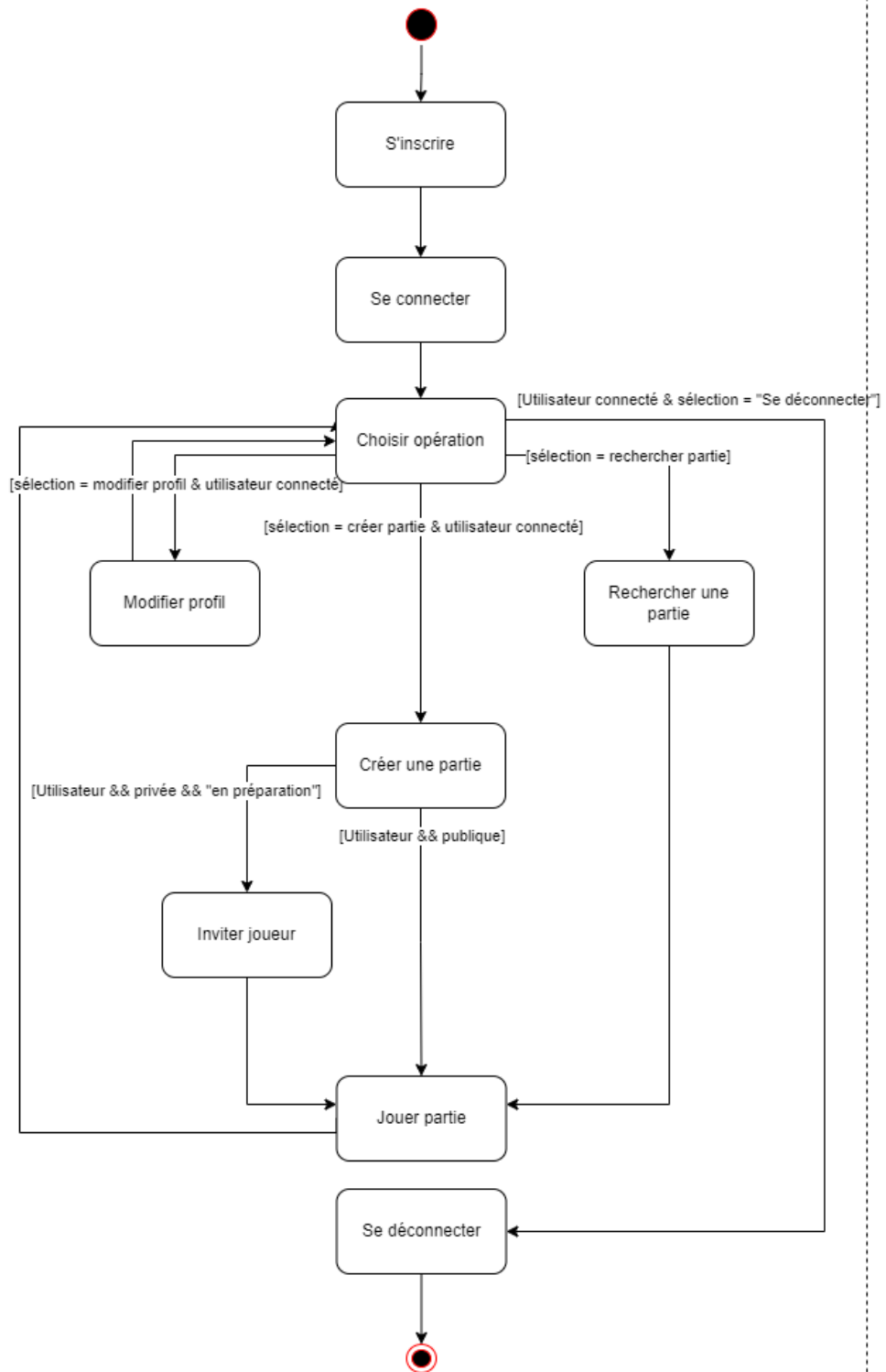
4. Diagramme d'activité général :

Les diagrammes d'activités permettent de mettre l'accent sur les traitements. Ils sont donc particulièrement adaptés à la modélisation du cheminement de flots de contrôle et de flots de données. Ils permettent ainsi de représenter graphiquement le comportement d'une méthode ou le déroulement d'un cas d'utilisation.

Pour le diagramme d'activité global, on a commencé par la fonctionnalité inscription pour les utilisateurs qui n'ont pas de compte. Étant connecté, l'utilisateur a le droit de choisir entre 4 opérations : modifier profil, rechercher une partie, créer une partie et se déconnecter.

Si l'utilisateur choisit de créer une partie, il sélectionne le mode souhaité pour la partie (privé ou public) : si c'est privé, l'utilisateur invite d'autres joueurs, si c'est public, l'utilisateur joue la partie avec d'autres joueurs récupérés automatiquement de la liste d'attente.

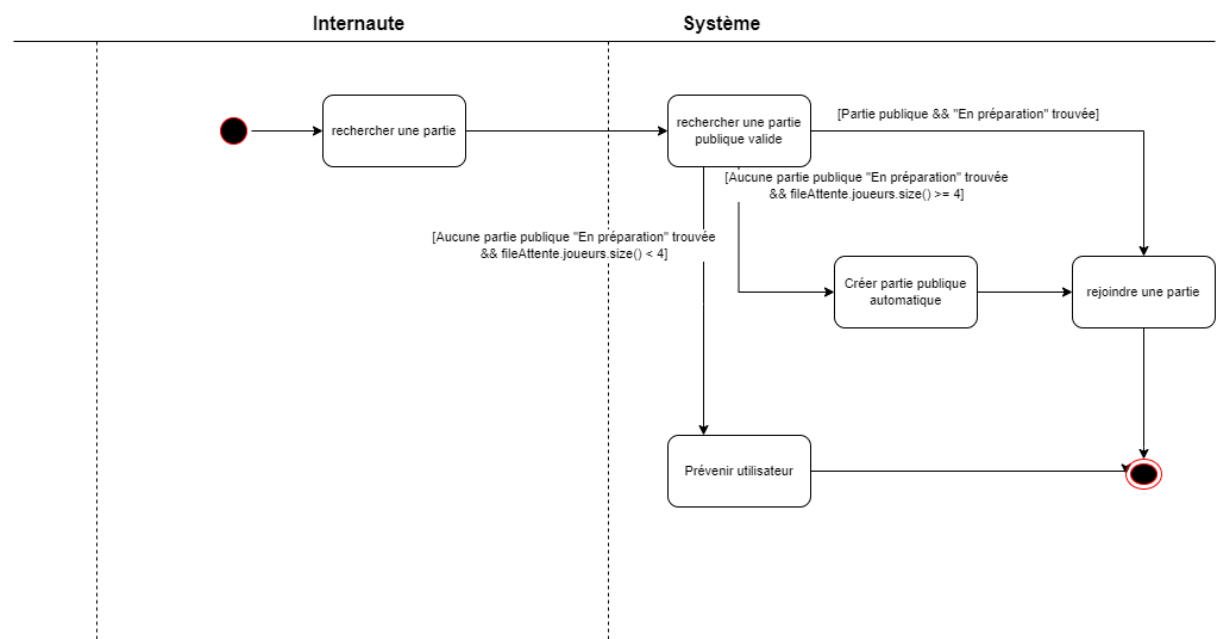
Internaute



4.1 Diagramme d'activité : Recherche partie

Dans ce diagramme, l'utilisateur cherche une partie publique à rejoindre, il est donc posé devant 3 cas :

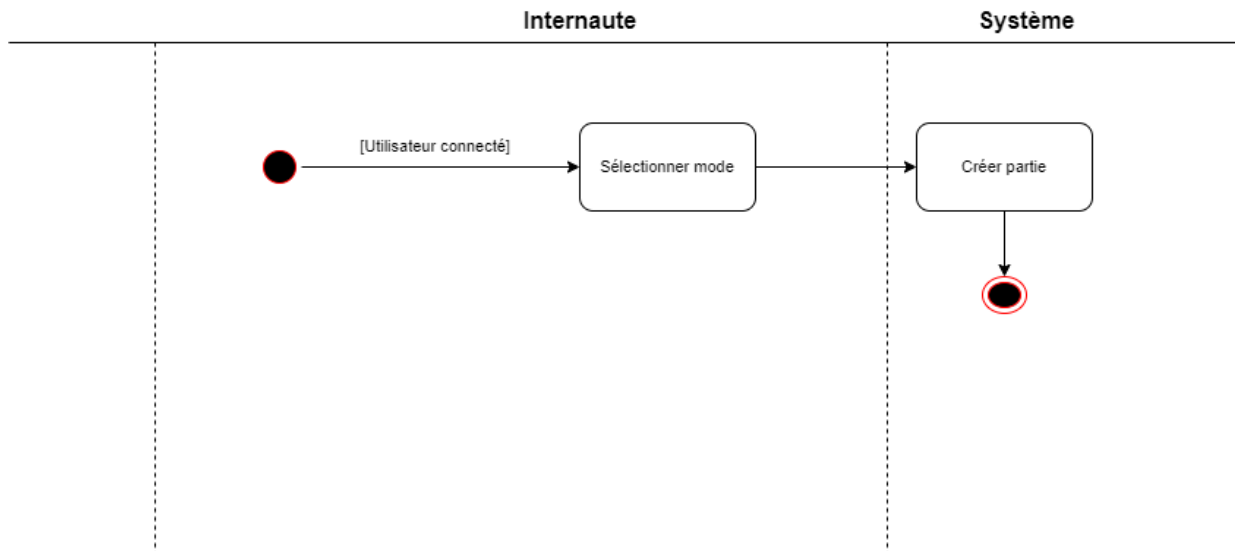
- 1er cas : si l'utilisateur trouve une partie publique en préparation, il peut alors la rejoindre directement.
- 2ème cas : s'il ne trouve aucune partie publique en préparation et le nombre des joueurs en fil d'attente est supérieur ou égal à 4, une partie publique se crée automatiquement, et l'utilisateur peut alors rejoindre la partie.
- 3. 3ème cas : s'il ne trouve aucune partie publique en préparation et que dans la file d'attente on n'a pas encore 4 joueurs, l'utilisateur ne peut pas jouer et il est prévenu.



4.2 Diagramme d'activité : Créer partie

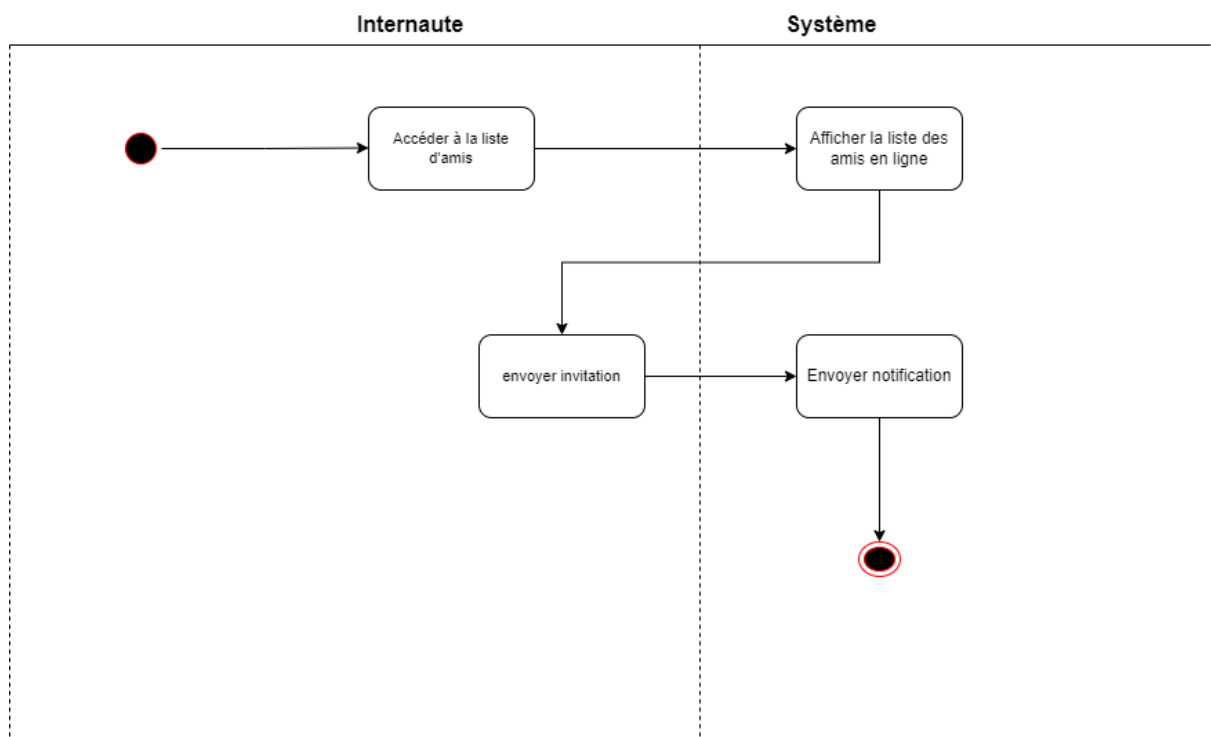
Pour le troisième diagramme d'activité, la fonctionnalité 'créerPartie' ne concerne que les utilisateurs qui sont déjà inscrits et connectés par la suite.

Au début, on vérifie si l'utilisateur est connecté, ensuite ce dernier sélectionne le mode souhaité pour la partie (privé ou public) et à la fin la partie se crée.



4.3 Diagramme d'activité : Inviter joueur

En ce qui concerne le diagramme d'activité : Inviter joueur, en voulant inviter les joueurs dans sa partie privée, le système affiche à l'utilisateur la liste des amis en ligne. L'utilisateur choisit donc qui inviter et une notification s'envoie aux joueurs invités pour qu'ils puissent rejoindre la partie.



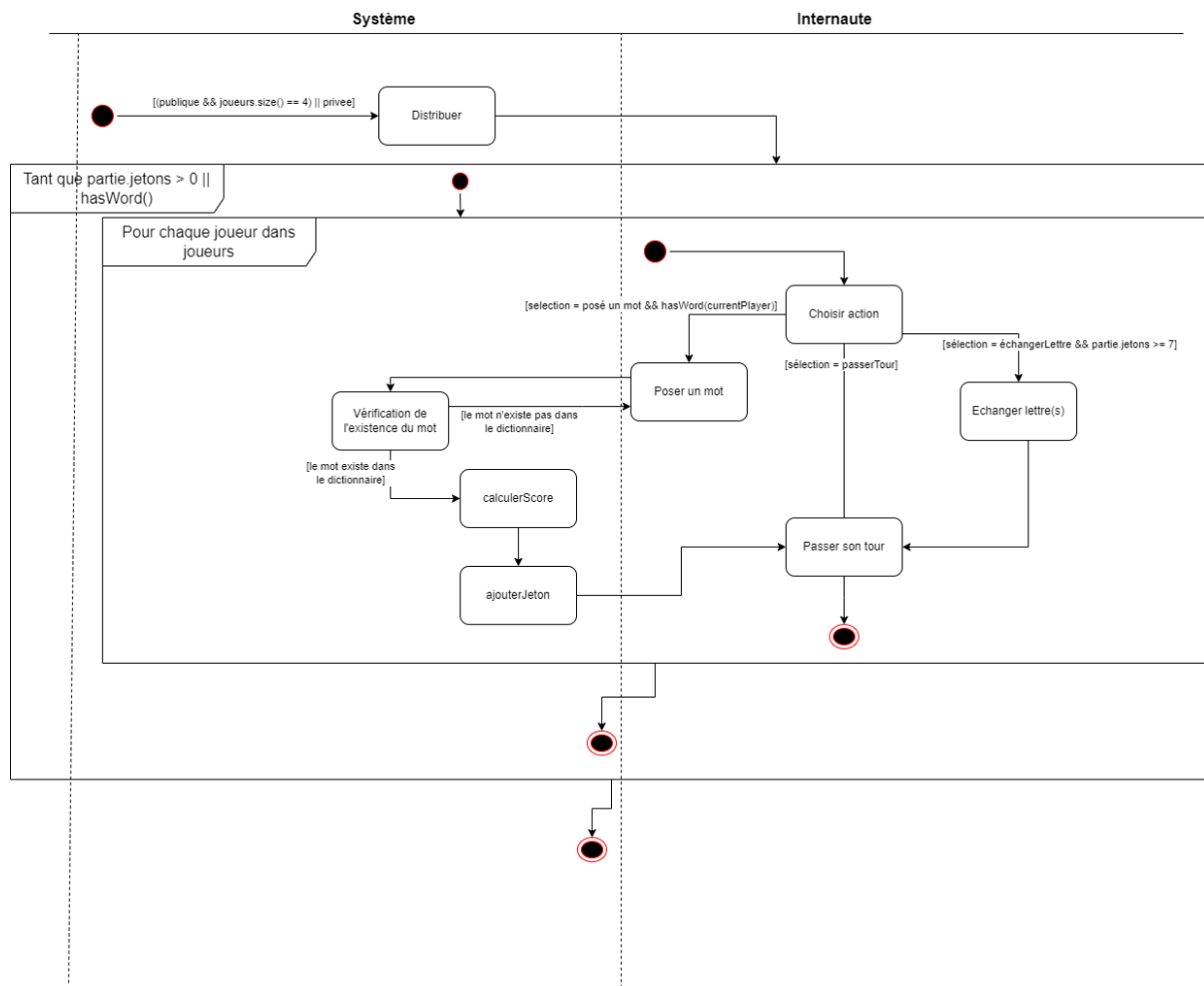
4.4 Diagramme d'activité : Jouer partie :

Pour ce qui est du diagramme d'activité du déroulement du jeu, la partie peut être débutée lorsque les 4 joueurs sont présents pour le cas d'une partie publique. Dans le cas d'une partie privée, celle-ci peut être débutée avant que le salon (lobby) soit complètement rempli.

L'activité "jouer un tour" est alors répétée jusqu'à la fin de la partie et successivement par chacun des participants. C'est ce qui est représenté sur le diagramme. Tant qu'un joueur peut poser un mot, ou que la pioche possède au moins un jeton la partie continue. Chaque tour, un joueur à le choix entre les actions suivantes :

- Passer son tour sans rien faire
- Défausser au moins 1 jetons dans la pioche contre le même nombre de jeton aléatoire
- Poser un mot sur la grille

Dans ce diagramme les "frames" représentent des boucles sur des listes, l'outil étant limité, nous n'avons pas pu représenter plus adéquatement le déroulement de cette fonctionnalité pour le diagramme d'activité.



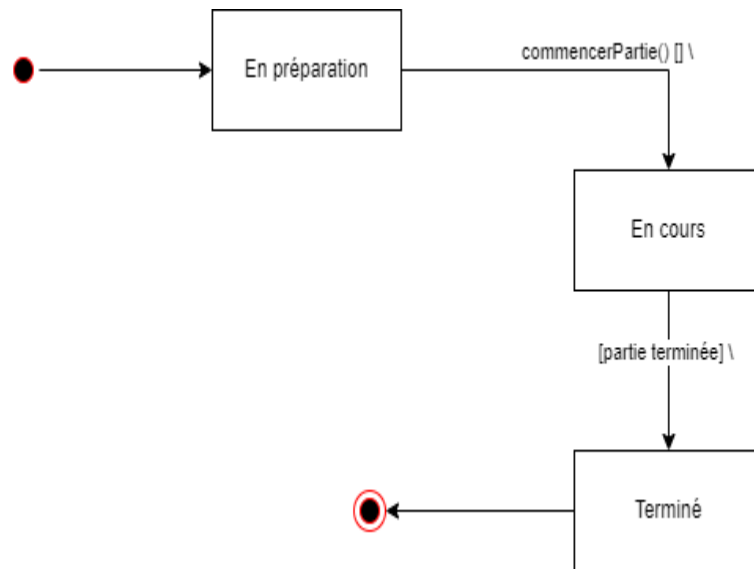
5. Diagramme d'état transition :

Le diagramme d'états-transitions d'UML décrit le comportement interne d'un objet à l'aide d'un automate à états finis.

Dans notre cas, on a choisi de faire deux diagrammes d'état transition : le premier diagramme définit le déroulement interne d'une partie, et le deuxième définit le déroulement du jeu pour un utilisateur.

5.1 Diagramme d'état transition Partie

Ce diagramme décrit la transition de l'état de l'objet Partie, après la création d'une partie son état est "en préparation", lorsque la partie commence son état passe à "en cours", et une fois la partie se termine on passe à l'état "terminé"



5.2 Diagramme d'état transition Joueur :

Ce diagramme décrit la transition de l'état de l'objet Utilisateur, une fois l'utilisateur est connecté son état passe à "en ligne", s'il veut chercher une partie, l'état passe à " en recherche de partie", s'il veut créer une partie dans ce cas-là l'état passe à "dans une partie en préparation", et dans les deux états s'il commence la partie, il passe à l'état "en Partie "

