

## UTBM – UV LO43

### Bases fondamentales de la programmation orientée objet

## Projet – Réalisation du jeu

## Small UTBM world



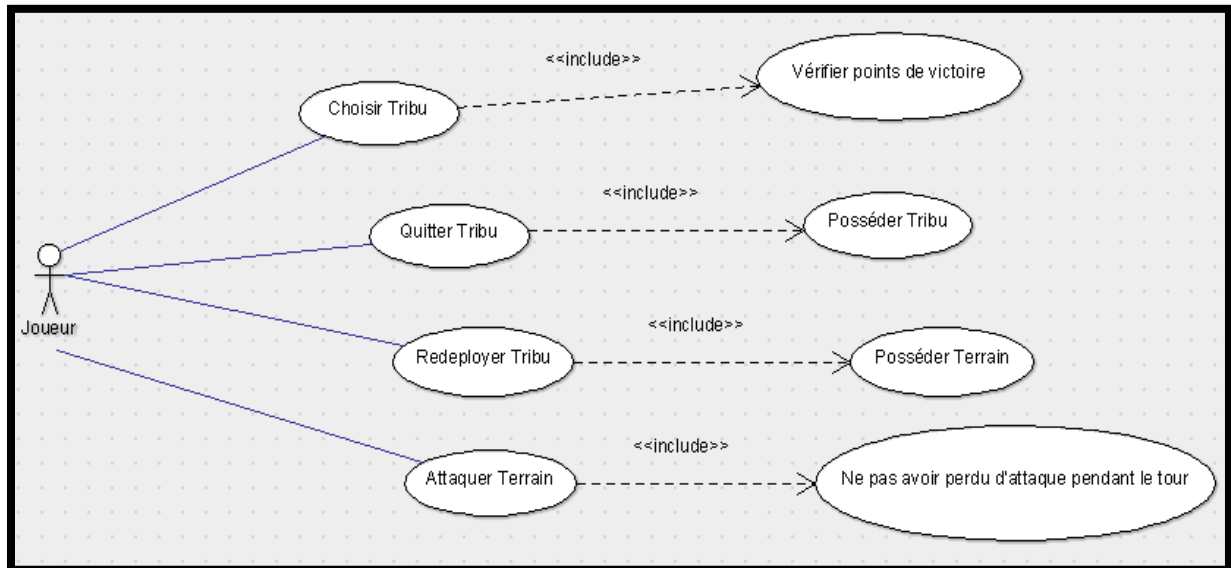
## **Introduction**

Dans le cadre de l'UV LO43, nous devons réaliser un projet de groupe nous permettant d'appliquer la POO que nous étudions durant ce semestre. Pour cela, il nous a été proposé trois projets sur lesquels nous avons choisi SmallWord. SmallWord est un jeu de société qui a été aussi transformé en jeu pc. Ce choix nous a paru judicieux puisqu'il s'agit d'un jeu de conquête de territoires. Cela ressemble à de nombreux jeux qui nous sont déjà connus. Le but de ce projet est donc d'adapter le jeu au monde de l'UTBM. Nous avons redéfini les règles et le plateau de jeu pour qu'il coïncide avec l'ambiance de l'école.

# Partie 1 - Diagrammes UML

Durant les TP, nous nous sommes consacrés en premier lieu à confectionner une UML nous permettant de ne pas partir sur de mauvaises bases.

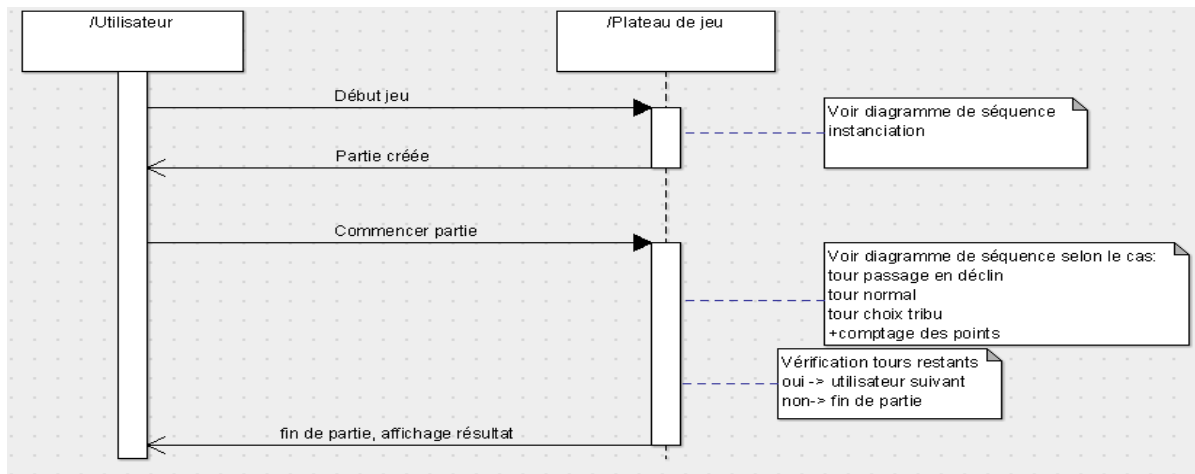
## Diagramme de cas d'utilisation:



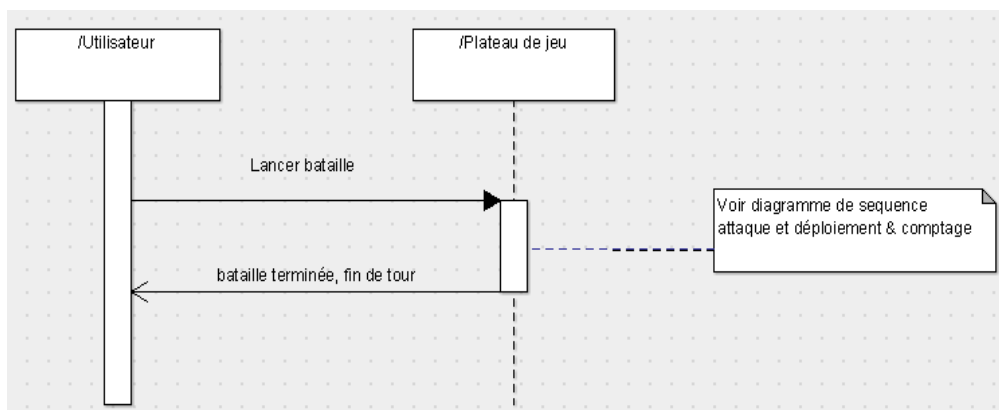
## Diagramme de séquence:

Pour ce diagramme, on a décidé de le séparer en plusieurs parties pour permettre de traiter tous les cas de figure durant l'exécution du programme. Nous avons utilisé le logiciel ArgoUML pour cette partie, le rendu n'est donc pas forcément optimal mais est lisible. On a choisi de faire apparaître Utilisateur qui diffère donc du joueur pour mettre en avant l'utilisateur humain qui contrôle le joueur du plateau de jeu. On a donc:

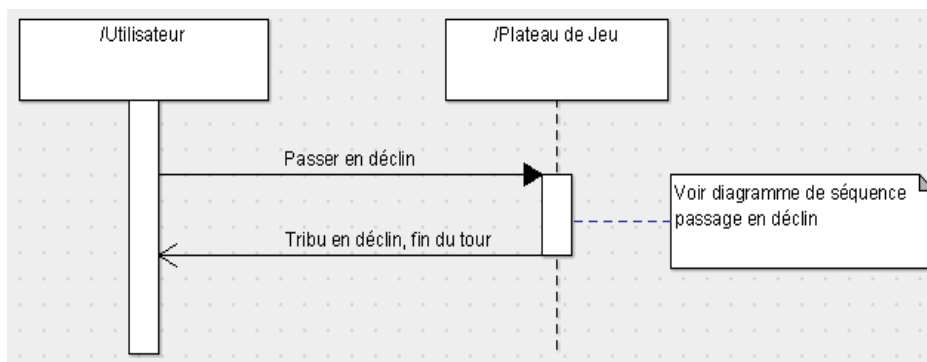
## Partie:



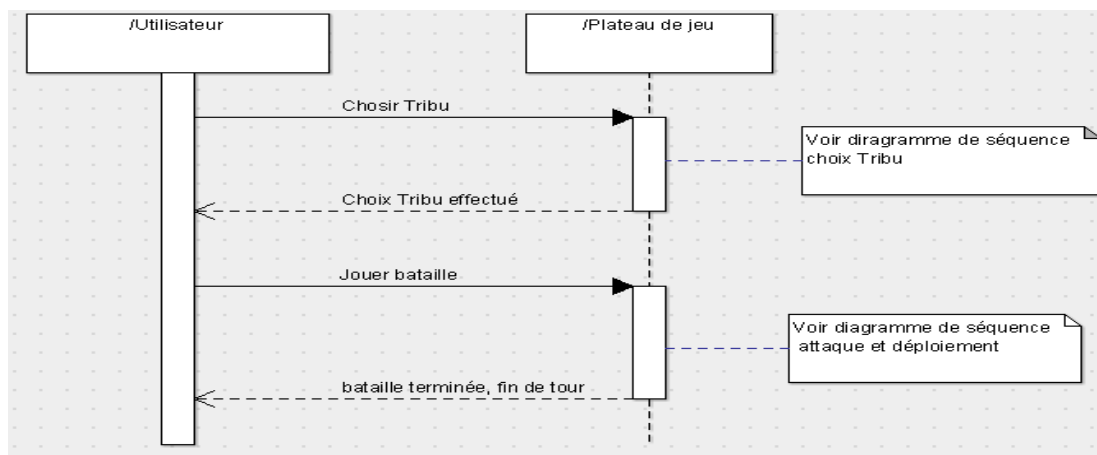
## -Tour normal



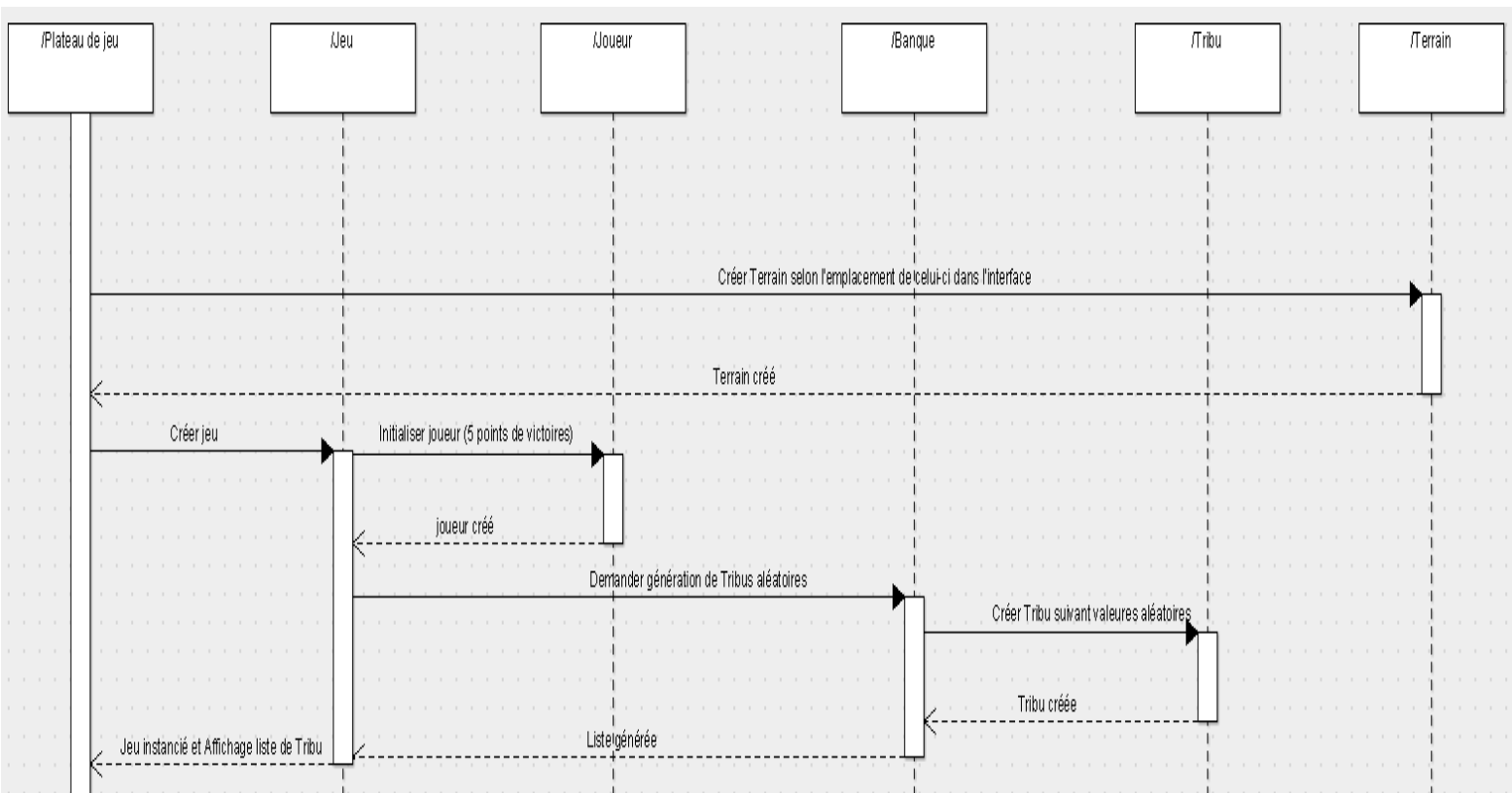
## -Tour du passage en déclin



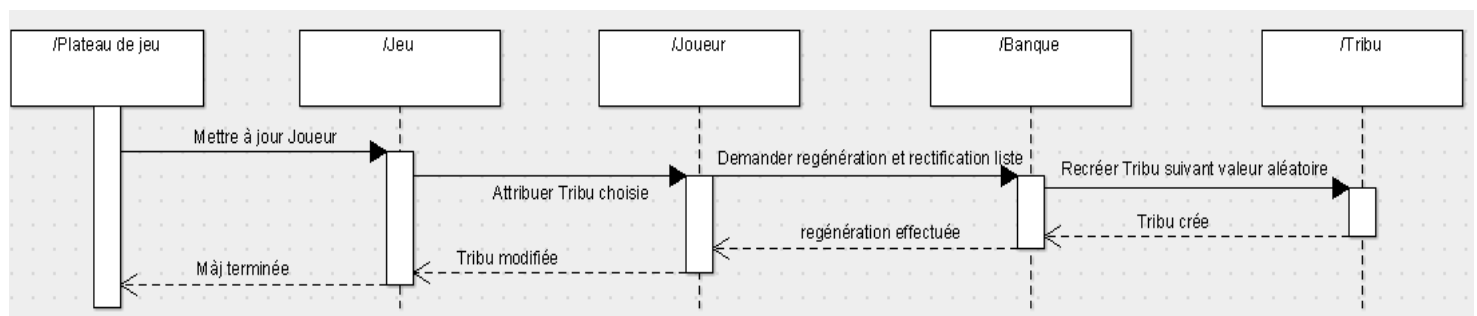
### - Tour de choix de Tribu:



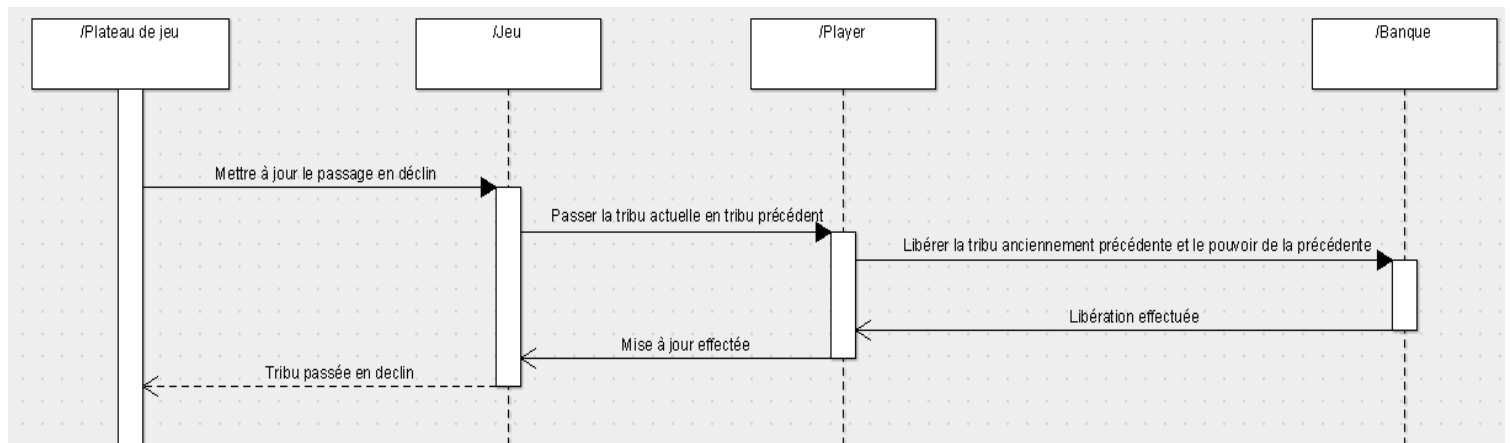
### Instanciation:



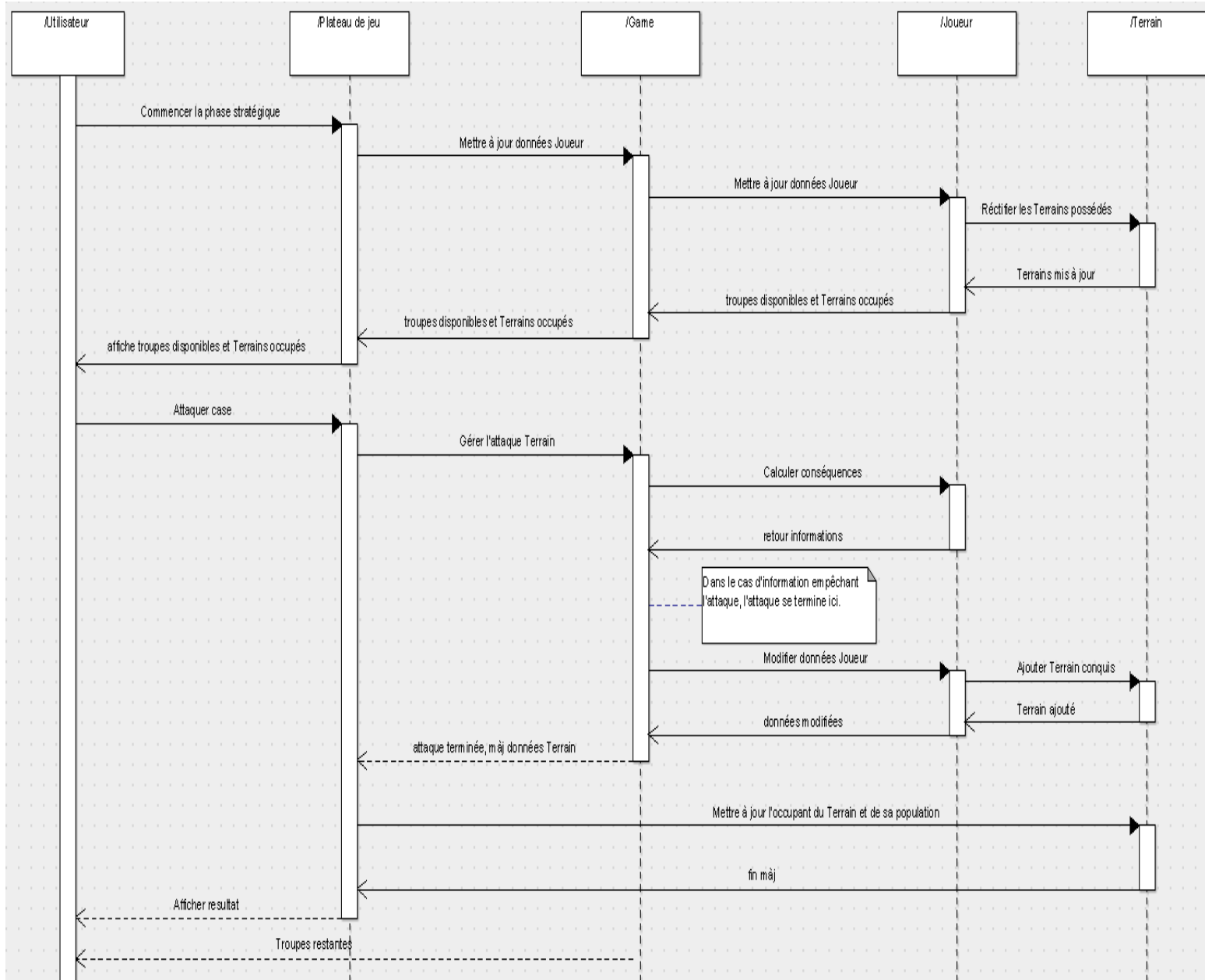
### Choix de la tribu:



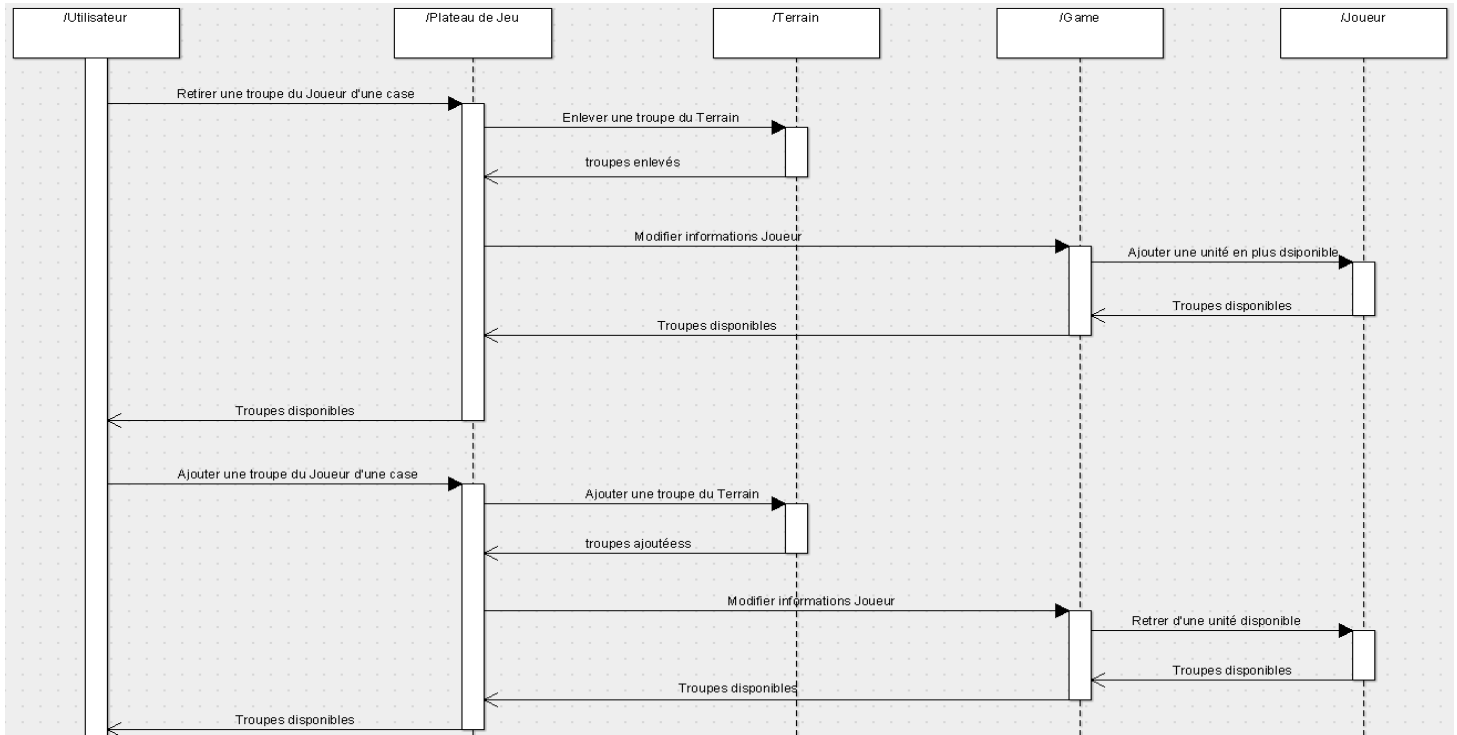
### Passage en déclin:



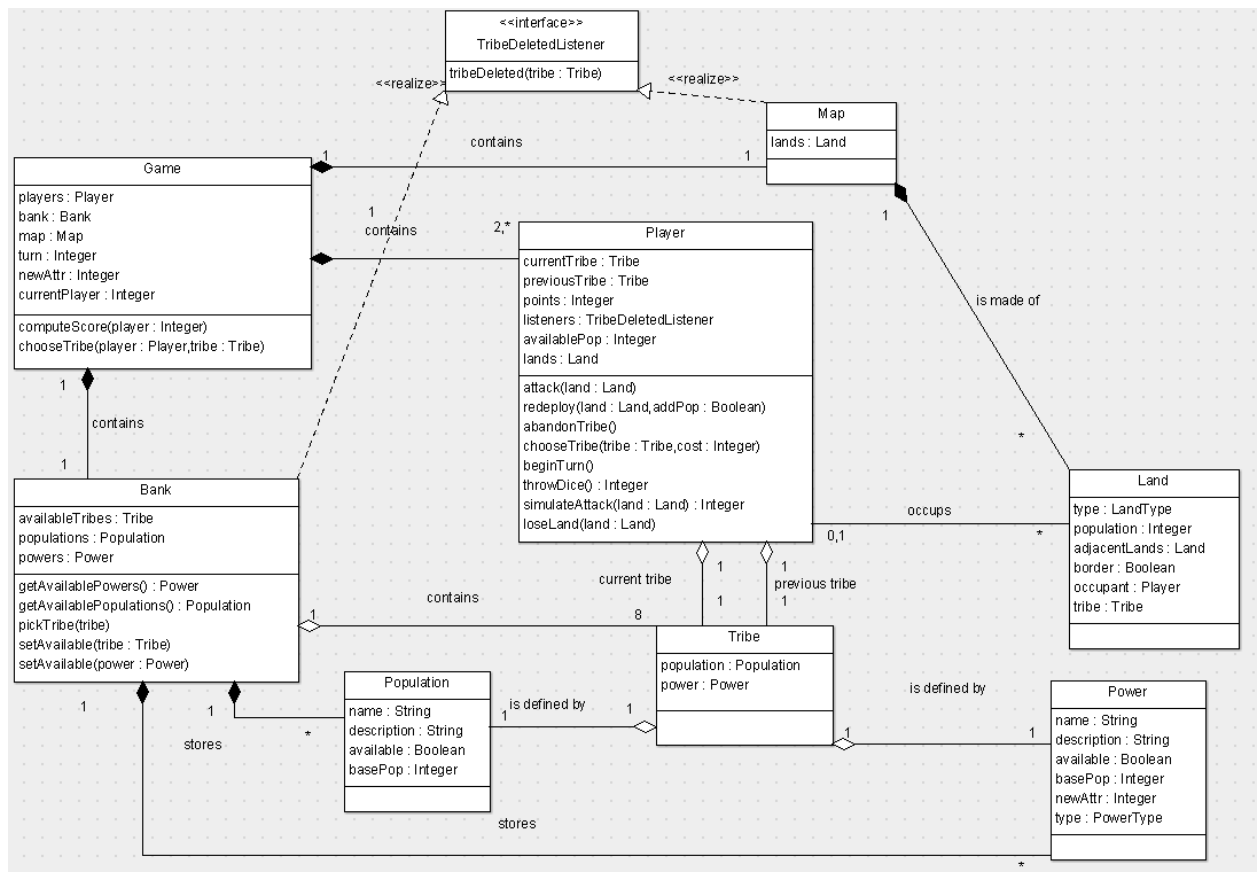
### Attaque:



**Déploiement:**

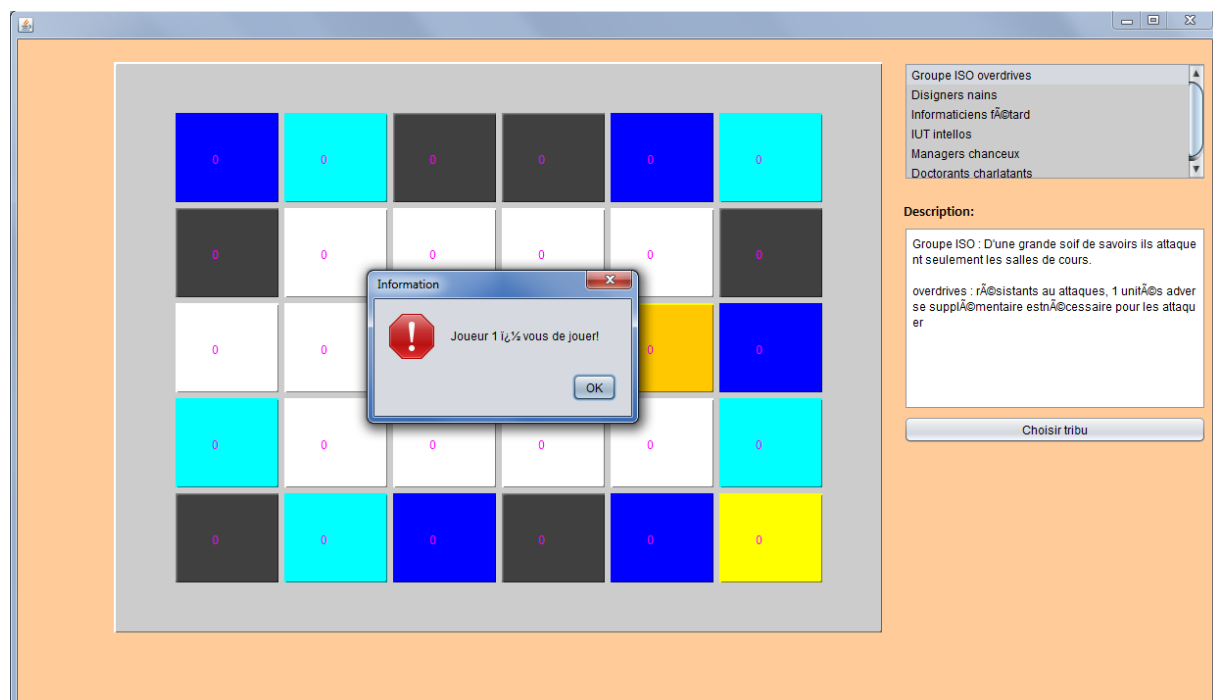


### Diagramme de Classes:



# Partie 2 - Explications de nos choix

## 1 - Interface



Voici l'état actuel de notre interface, elle n'est comme vous pouvez le voir pas terminée. Une seule carte est disponible pour le moment. Nous avons décidé en premier lieu de nous concentrer sur le modèle et de le rendre opérationnel avant de travailler sur l'interface. On peut tout de même jouer mais cela n'est pas facilement compréhensible pour l'instant car l'interface est encore incomplète. Nous avons essayé d'appliquer le modèle MVC comme conseillé en cours, cependant la séparation entre vue et contrôleur nous semble assez floue avec Swing. Nous sommes donc plus proche d'un modèle « Modèle – Interface ».

## 2- Modèle

Ici, nous allons détailler le diagramme de classes.

Nous avons réfléchi pour rendre le modèle le plus simple et logique possible. Nous avons donc une classe principale Game qui possède les classes Bank, Player et Map.

La classe Bank génère aléatoirement une liste de Tribe qui est la combinaison de Power et Population. Cela permet d'avoir accès directement aux choix possibles de tribu (association peuple/pouvoir) pour l'utilisateur. Comme vous pouvez le voir sur le diagramme, on utilise un Listener qui nous permet de faire le lien entre Map et Bank pour gérer l'accessibilité des populations et des pouvoirs quand les tribus disparaissent de la carte.

La classe Player gère presque toute la partie du jeu de conquête et permet de changer de tribu. On a décidé de garder la tribu précédente en attribut pour permettre de compter plus facilement les points à la fin de chaque tour.

On garde aussi la liste des terrains occupés (Land) pour respecter les règles du jeu, l'utilisateur pouvant se tromper on doit bloquer les actions qui ne sont pas autorisées (s'attaquer soi-même, etc..). Enfin il reste l'attribut des troupes disponibles qui est utile pour les raisons précédentes et le redéploiement des troupes.



On respecte donc le cahier des charges du jeu. Nous allons maintenant voir quels sont les changements apportés pour intégrer le monde de l'UTBM.

### **3-UTBM**

Même si le fond reste le même le jeu est d'un tout autre horizon que SmallWorld. En effet, les terrains sur la carte et les peuples disponibles en jeu sont tirés de notre école. On a donc comme peuple:

Professeurs: Leur connaissance du terrain leur permet d'être plus efficaces pour se défendre. Il faut une unité de plus pour les attaquer.

Doctorants: Etant d'une grande capacité d'adaptation ils peuvent attaquer les laboratoires avec une unité de moins.

GI: Leur envie de s'établir sur un ordinateur est tellement grande que leur combativité en est stimulée, ils ont besoin d'une unité en moins pour attaquer les salles GI.

EDIM: Leur charme absolu leur permet de rendre inutile un défenseur adverse lors d'une attaque de terrain.

E: Leur rapidité à juger leurs ennemis leur permet de se replier rapidement, ils ne perdent pas de troupe quand ils perdent un terrain.

IMSI: Leur habileté à s'installer là où on les attend le moins leur assurent un point de victoire en plus pour chaque terrain couloir possédé.

GMC: Ayant besoin des GI pour leurs travaux, ce peuple ne peut pas les attaquer.

TC: Leurs examens pour devenir soldat sont tellement éprouvants qu'une unité est réorientée à chaque tour.

IUT: La chance ne leur est pas permise, ils ne peuvent donc pas compter sur des renforts supplémentaires.

Groupe ISO: D'une grande soif de savoir, ils attaquent seulement les salles de cours.

Anciens: Il faut une unité en plus pour pouvoir les attaquer étant donné leur violence sans égal.

Personnels administratifs: Etant donné qu'ils possèdent un pass très efficace, ils peuvent s'établir n'importe où.

Pour les pouvoirs nous avons retenu:

Intellos: +1 point de victoire bonus si la tribu est installée dans la bibliothèque

Fêtard: +4 points de victoire si le foyer est possédé par la tribu

Bourrés: La tribu a besoin d'une unité en plus pour attaquer mais lance le dé à chaque attaque

Charlatans: A chaque tour, la tribu reçoit une unité en plus.

Chanceux: +7 points lorsque la tribu est choisie.

Courageux: La tribu a besoin d'une unité en moins pour attaquer.

Overdrives: La tribu est résistant aux attaques, une unité adverse supplémentaire est nécessaire pour les attaquer.

Nains: Leurs attaques sournoises leur permettent de ne pas céder de terrain dans les couloirs. Il ne peut donc pas perdre les terrains couloirs quand la tribu le possède.

Pour l'instant le jeu n'est jouable qu'à deux puisqu'il ne contient pas assez de pouvoirs pour être joué avec 3 ou plus de joueurs. Il nous sera cependant très simple techniquement d'ajouter plus de joueurs lorsque nous aurons implémenté de nouveaux peuples et pouvoirs.  
L'équilibre du jeu entre les différents peuples et pouvoirs n'est pas encore final.

## **Conclusion**

Ce projet nous a appris une fois de plus l'importance de la cohésion du groupe. Grâce aux nombreuses heures de TP, nous sommes arrivés à dégager en premier lieu un UML plutôt satisfaisant et ensuite nous nous sommes répartis les tâches pour coder efficacement. La diversité de notre groupe fût enrichissante puisqu'il était composé de personnes avec des cursus bien différents. On a aussi eu beaucoup d'idées par rapport à l'amélioration de notre projet. Malheureusement, elle n'ont pas pu toutes être concrétisées. Nous retiendrons donc comme améliorations possibles: une interface plus dynamique et animée, l'idée d'ajouter la météo comme facteur durant les conquêtes (en plus du dé) ou encore la possibilité d'une interface permettant à l'utilisateur de créer son propre peuple et d'autres pouvoirs dans le jeu.