

UML



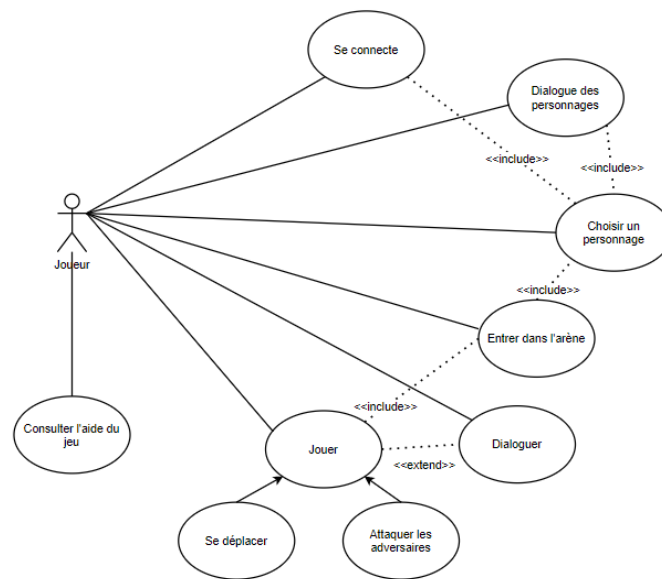
1. Diagramme cas d'utilisation

2 formes :

- Activité d'un acteur (schémas)
- Tableaux

a) Activité d'un acteur (schéma)

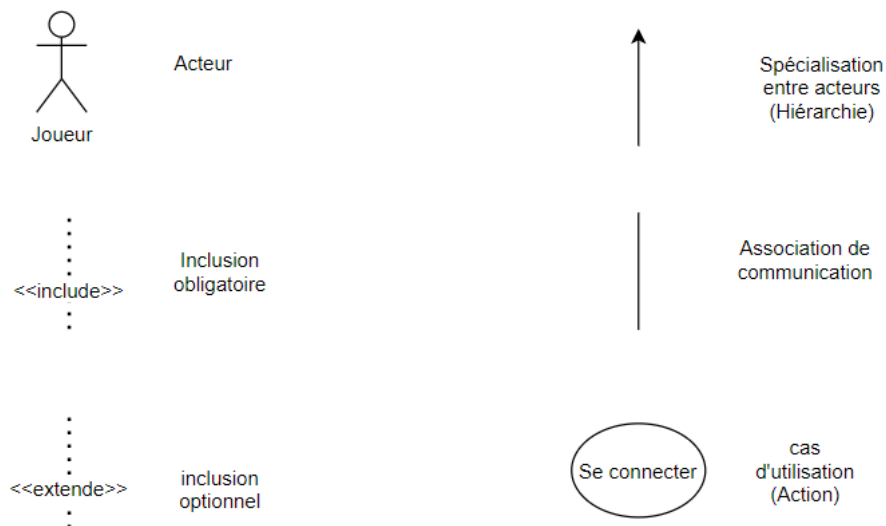
- Un joueur se connecte
- Une fois connecté, il choisit un personnage après avoir consulté le dialogue des personnages.
- Il peut alors entrer dans l'arène et jouer (se déplacer, attaquer les adversaires).
- Il peut aussi, pendant le jeu, dialoguer avec les autres joueurs (chat).
- A tout moment (connecté ou non), le joueur peut consulter l'aide du jeu.



Le diagramme de cas d'utilisation correspond à :

- Représentation (schéma ou tableau) global des activités d'un acteur (vision du système par l'acteur).
- Ensemble des actions possibles entre cet acteur et le système.
- Une base de travail pour construire les interfaces graphiques.

- Syntaxes



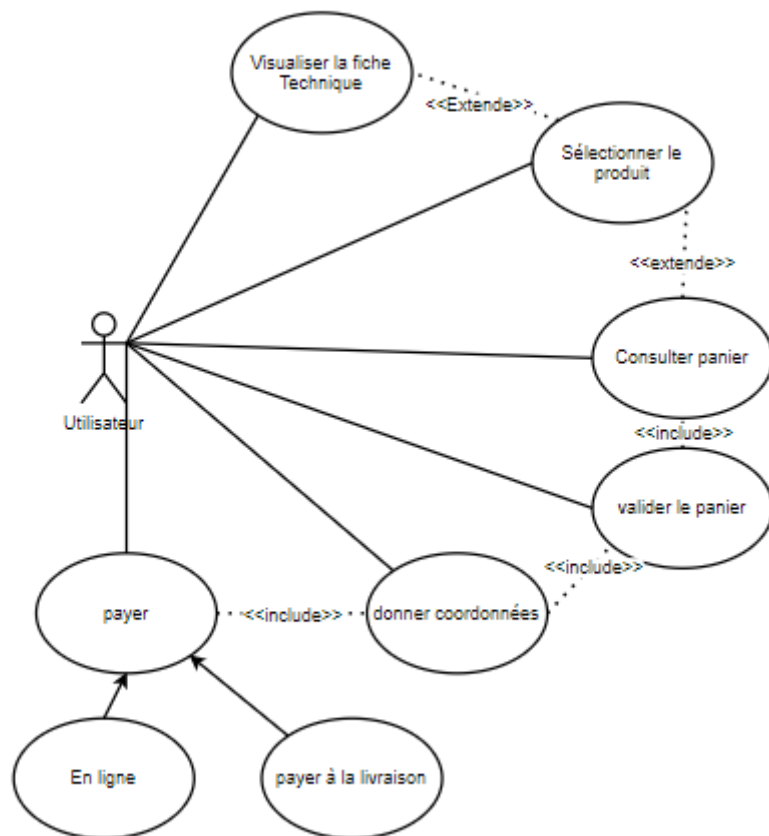
b) Tableau des opérations d'un cas d'utilisation

Le tableau détaille les opérations d'un cas d'utilisation (un cercle).
Il sert de base à la construction des interfaces.

cas d'utilisation	se connecter
Acteur	Joueur
Environnement déclencheur	néant
Intérêt	Accéder au serveur afin d'être authentifié et accéder au jeu
Pré-conditions	serveur actif
Post-conditions	connections établie
Scénario nominal	<ol style="list-style-type: none">1. L'utilisateur saisit l'IP du serveur.2. L'utilisateur demande une connexion au serveur.3. Le serveur répond
Extensions (optionnel)	non
Contraintes	<p>2a → L'adresse IP n'est pas remplie : aller en 1.</p> <p>3a → Le serveur retourne un message d'erreur : aller en 1.</p> <p>3a → Le serveur ne répond pas : aller en 1.</p>

Exercice :

Sur un site commercial, un utilisateur peut sélectionner des produits éventuellement après avoir lu la fiche technique de chaque produit. Il peut ensuite visualiser son panier. S'il décide de valider son panier, il doit donner ses coordonnées et il peut alors soit choisir de payer en liquide, soit de payer à la réception du colis. Dessiner le diagramme de cas d'utilisation correspondant.



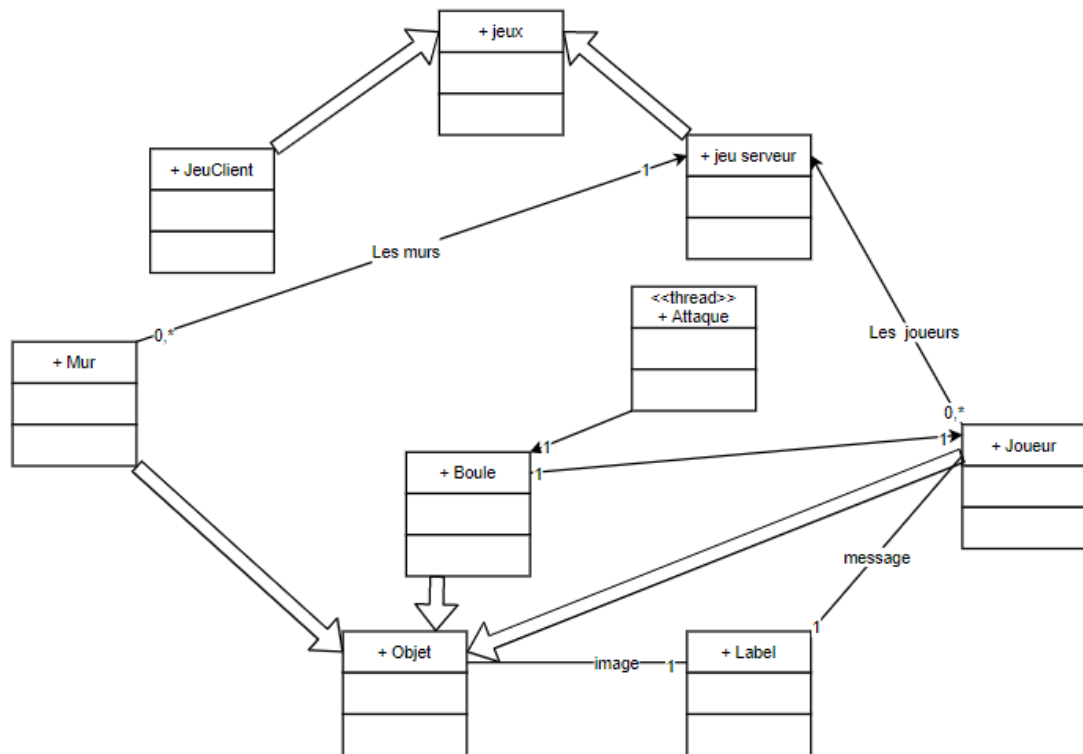
2. Diagramme de classes

Schématisation de classes et leurs liens

Le diagramme représente l'organisation des classes de l'application.

Exemple :

Le but du jeu est de permettre à plusieurs joueurs de se connecter à un serveur et de jouer ensemble. Chaque joueur choisit un personnage, il donne un pseudo et rentre dans l'arène qui contient des murs, il peut tirer des boules pour toucher les adversaires.



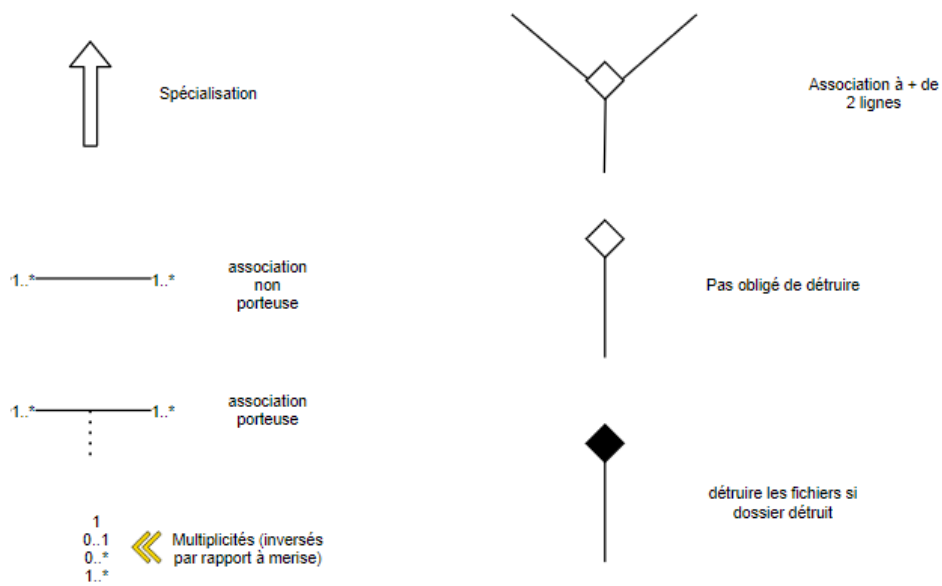
Le diagramme de classes est un schéma :

- représentatif des classes et de leurs liens.
- présentant l'organisation des données et traitements de l'application.

Syntaxe :

+ Joueur	« nom (italique : ne peut pas être instanciée)
-pseudo -vie -positionX -positionY -direction	« propriétés (variables) et portées (pas de propriétés objet)
+avancer +attaquer	« méthodes et portée (généralement pas le constructeur)

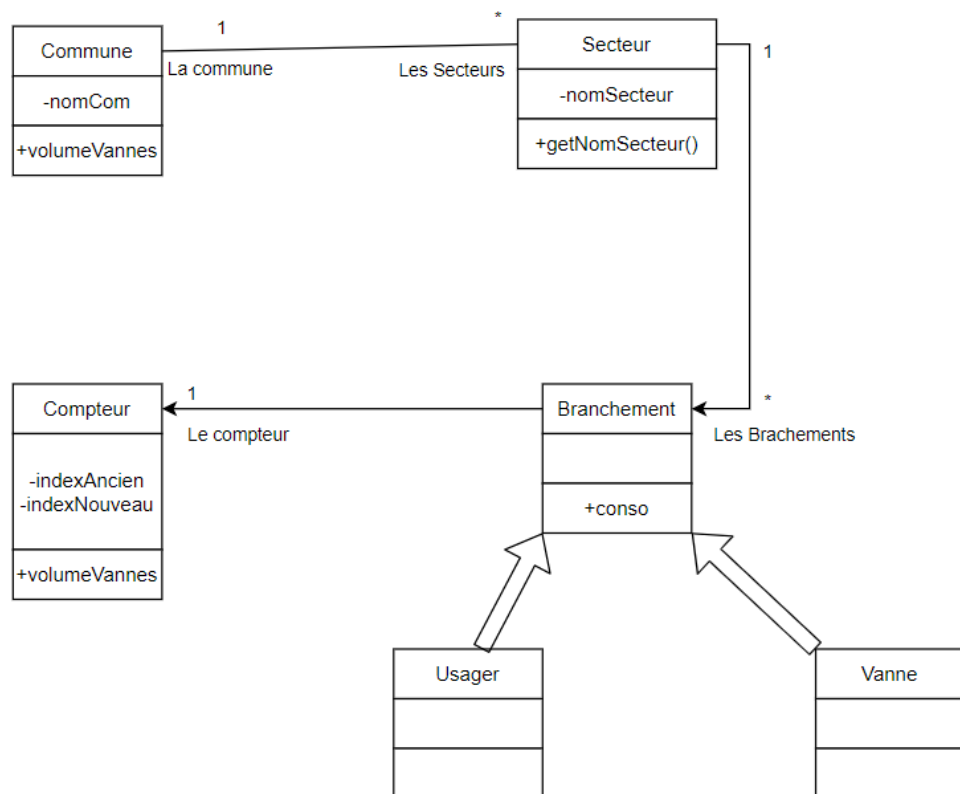
-- : privé + : public # : protégé (que les classes filles peuvent y accéder)



Exercice :

Construire le diagramme de classe correspondant :
<p>→ Classe commune :</p> <ul style="list-style-type: none">◆ Privé<ul style="list-style-type: none">• nomCom = chaîne*• lesSecteurs = collection de Secteur◆ Public<ul style="list-style-type: none">• commune (nomCom : chaîne*)• volumeVannes() : Entier
<p>→ Classe Secteur :</p> <ul style="list-style-type: none">◆ Privé<ul style="list-style-type: none">• monSecteur : chaîne*• laCommune : Commune• lesBranchements : collections de Branchement◆ Public<ul style="list-style-type: none">• secteur(monSecteur : chaîne*, laCommune : commune)• getNomSecteur() : chaîne*
<p>→ Classe Compteur :</p> <ul style="list-style-type: none">◆ Privé<ul style="list-style-type: none">• indexAncien : Entier• IndexNouveau : Entier◆ Public<ul style="list-style-type: none">• Fonction_relevé() : Entier
<p>→ Classe Branchement :</p> <ul style="list-style-type: none">◆ Privé<ul style="list-style-type: none">• leCompteur : Compteur◆ Public<ul style="list-style-type: none">• conso() : Entier
→ <u>Classe Usager hérite de Branchement</u>
→ <u>Classe Vanne hérite de Branchement</u>

chaîne = chaîne de caractère*



[Exercices diagramme de classe supplémentaires](#)