# Analyse et spécification des besoins

L’objectif du système à réaliser est de permettre de jouer à « Tetris » en local ou en réseau. Dans cette partie d’analyse et spécification des besoins nous allons nous intéresser au début au un système complet et professionnel. Nous ferons ensuite quelques hypothèses de simplification pour faciliter la mise en œuvre.

## Diagramme de contexte du système

Le diagramme de contexte suivant fera le focus sur les bords du système notamment les acteurs avec lequel il va interagir, ainsi que ses fonctionnalités principales appelées aussi les cas d’utilisation principaux.

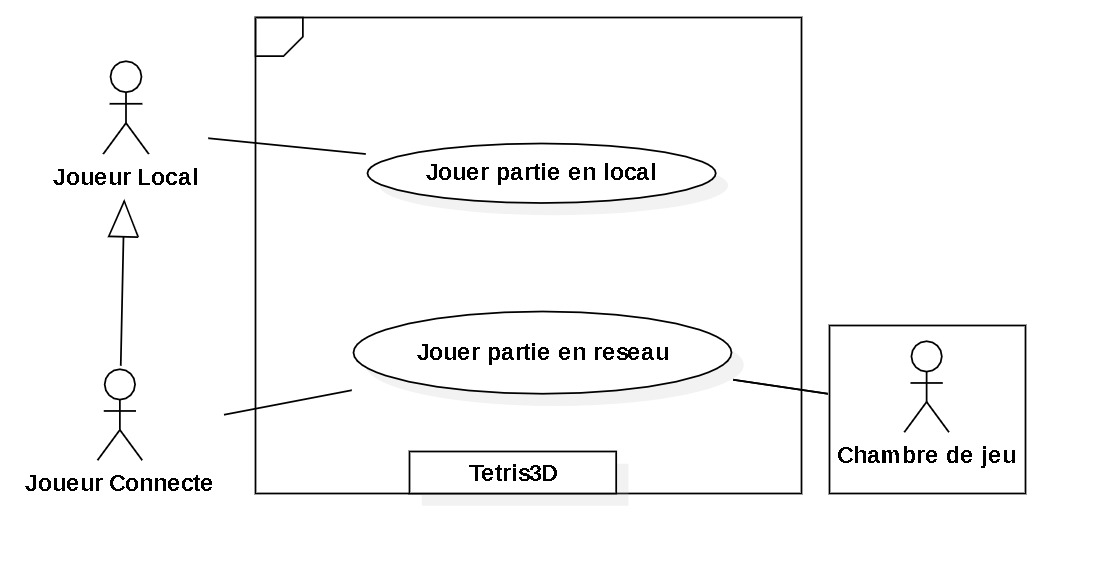


Figure x.x : Diagramme de contexte du système.

Les principaux cas d’utilisation sont :

* Jouer une partie en local dont le bénéficiaire est l’acteur : «Joueur Local » ;
* Jouer une partie en réseau dont l’utilisateur final est l’acteur « Joueur connecte ».

Ici l’un des bords (hors système) avec lequel va interagir le système est l’acteur secondaire « Chambre de jeu » qui fournira les données nécessaires pour jouer en réseau comme les parties en cours et les groupes de jeux…

## 2. Les principaux cas d’utilisation

### 2.1. Cas d’utilisation « Jouer Partie en Local »

Ce cas dans son scénario nominal permet le jeu en local, peut être enrichi par des scénarios alternatifs comme :

* La suspension et la reprise de la partie en cours ;
* La terminaison de la partie par un score plus grand que l’un des trois meilleurs scores anciens déjà réalisé ce qui enclenchera une mise au podium.

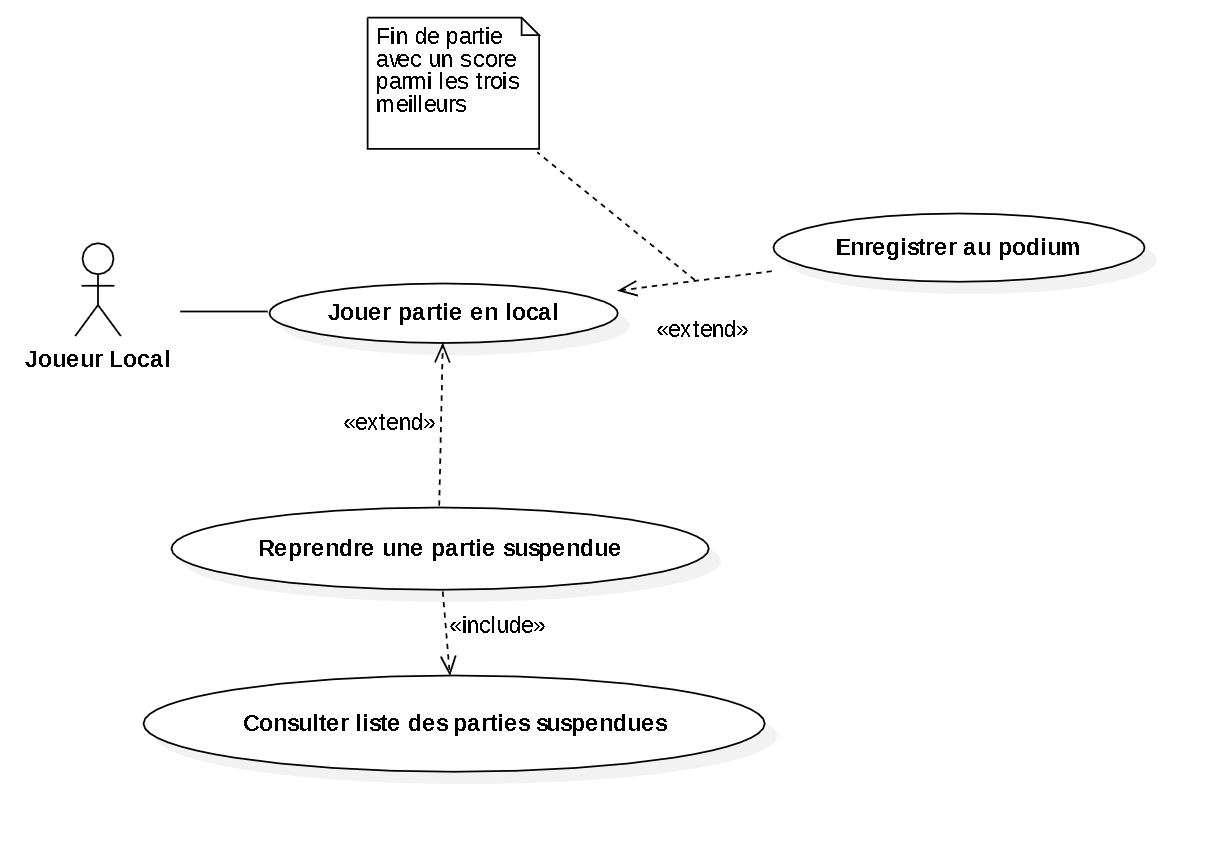


Figure x.x : Cas d’utilisation « Jouer Partie en Local ».

Le scenario nominal du jeu est le plus intéressant des cas d’utilisation à décrire sous forme de diagramme de séquences ou de flux, car d’une part ça représente le corps du business de l’application à mettre en place en mode monoposte. D’autre part ça donne le dialogue complet entre le client et le serveur « Tetris » qui permettra l’implémentation du jeu en réseaux.

1. Le joueur lance une nouvelle partie

2. Le système génère un nouveau tetramino l’envoi au joueur. Une temporisation est lancée.

3. Le joueur envoi sa réponse sous forme d’un caractère :

* KeyDown : pour baisser d’un pas le Tetramino
* KeyLeft : commande la rotation du tetramino
* ESC : Suspendre la partie

4. A la fin de la temporisation, le système tient en considération les commandes de l’utilisateur si elles sont survenues à temps.

4.1. Si le joueur demande la rotation ou la descente d’un pas, le système s’assure qu’il n’y a pas de débordement ni de chevauchement puis elle sera réalisée.

4.2. Une descente vers le bas d’un pas sera réalisée.

4.3. Etude si une ou plusieurs couches sont remplies. Si oui les effacer puis calculer le score et le niveau et les afficher au joueur, aller en 2.

4.4. Si Tetramino bloqué, aller en 2.

4.5. Si la partie n’est pas terminée aller en 3.

4.6. Partie terminée voir si le score réalisé fait partie des 3 meilleurs scores alors fêter cet évènement et ajouter le nom du joueur au podium.

### 2.2. Cas d’utilisation « Jouer Partie en Réseau »

Dans le cas d’utilisation : « Jouer Partie en Réseau », l’acteur principal doit se connecter à la chambre de jeu. Bien évidement s’enregistrer s’il ne possède pas déjà un compte joueur. Il peut ensuite :

* Lancer une nouvelle partie pour jouer tout seul ou attendre ses amis à le rejoindre ;
* Ou d’essayer de joindre une partie en cours ce qui l’amène à :
  + Consulter avant tout la listes des parties en cours ;
  + Puis de choisir une partie après avoir consulté la liste des sessions en cours ;
  + Envoyer une demande de joindre la partie à l’un de ses joueurs ;
* En tant que joueur connecté au réseau en train de jouer une partie en cours, il peut recevoir des demandes de se joindre à lui qu’il pourra accepter ou refuser.

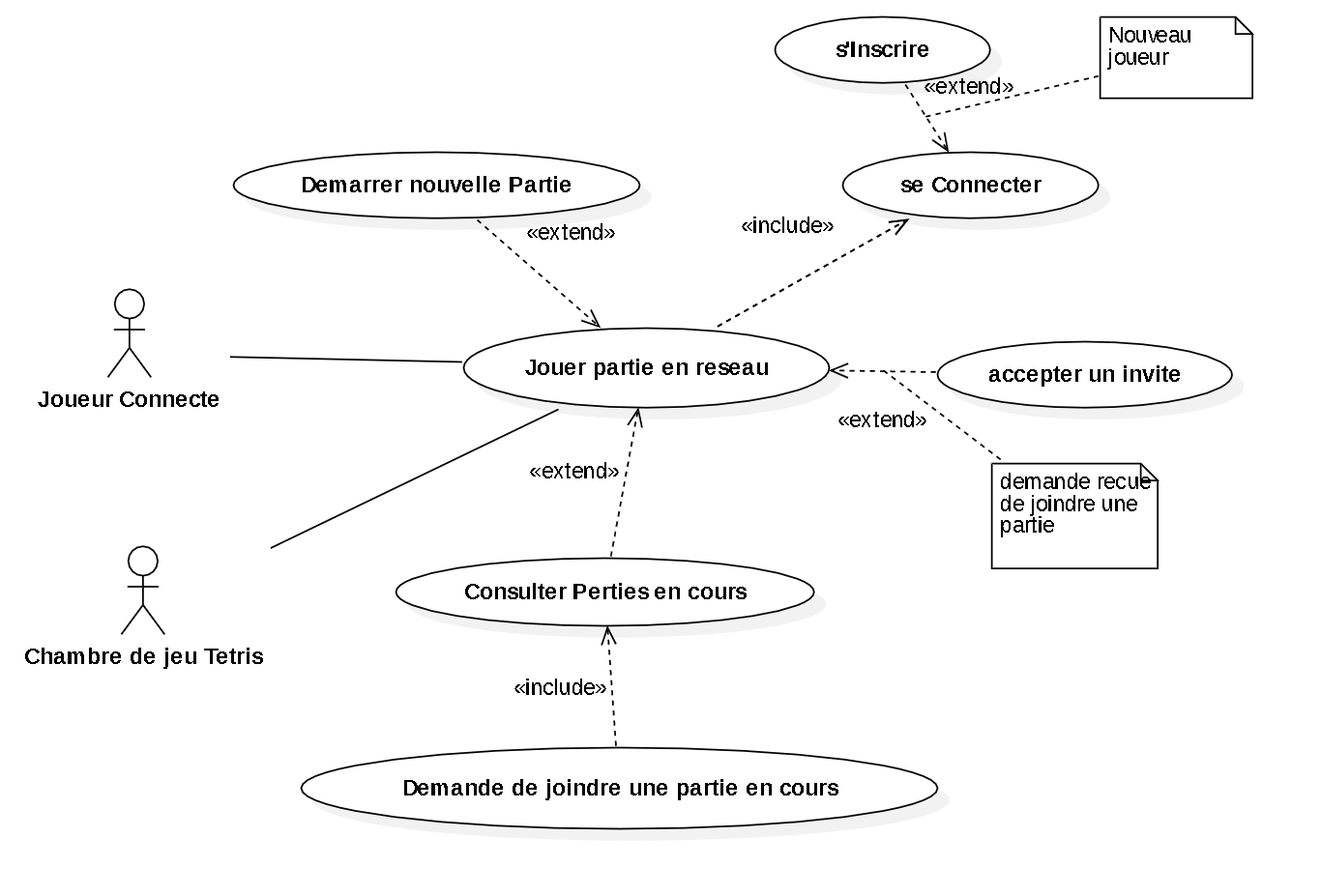


Figure x.x : Cas d’utilisation « Jouer Partie en Réseau ».

### 3. Diagramme de paquetage des cas d’utilisation de tout le système

Fonctionnellement les cas d’utilisation de l’application peuvent être classés en cinq paquetages ou modules :

* Gestion du jeu en monoposte : qu’on déjà décortiqué en détail ;
* La gestion des joueurs avec les sous fonctionnalités (cas) suivantes :
  + S’inscrire auprès du chambre de jeu Tetris ;
  + Se désinscrire ;
  + Se connecter au chambre de jeu ;
  + Se déconnecter.
* La gestion des serveurs des parties en cours en réseau : qui permet :
  1. Recevoir les commandes du client ;
  2. Jouer la partie en cours selon les commandes ;
  3. Envoyer l’avancement de l’état de la partie vers le client. Aller en 1.
* La gestion du client Tetris qui inclus les sous fonctionnalités comme :
  1. Envoyer les commandes du joueur au serveur des parties en cours ;
  2. Recevoir les réponses du serveur ;
  3. Afficher les résultats de la partie en cours. Si ce n’est pas encore la fin de la partie allé en 1.
* La gestion de la chambre de jeux qui est la première entité à commencer le dialogue avec le joueur connecté depuis un poste client :
  + Envoyer la liste de parties en cours ;
  + Permettre au Joueur de demander de joindre une partie encours auprès de l’un de ses joueurs.
  + En cas d’acceptation de l’invitation par l’un des joueurs de la partie, redirection du client au serveur de la partie à joindre ;
  + Mise au podium en cas de réalisation d’un nouveau score exceptionnel.

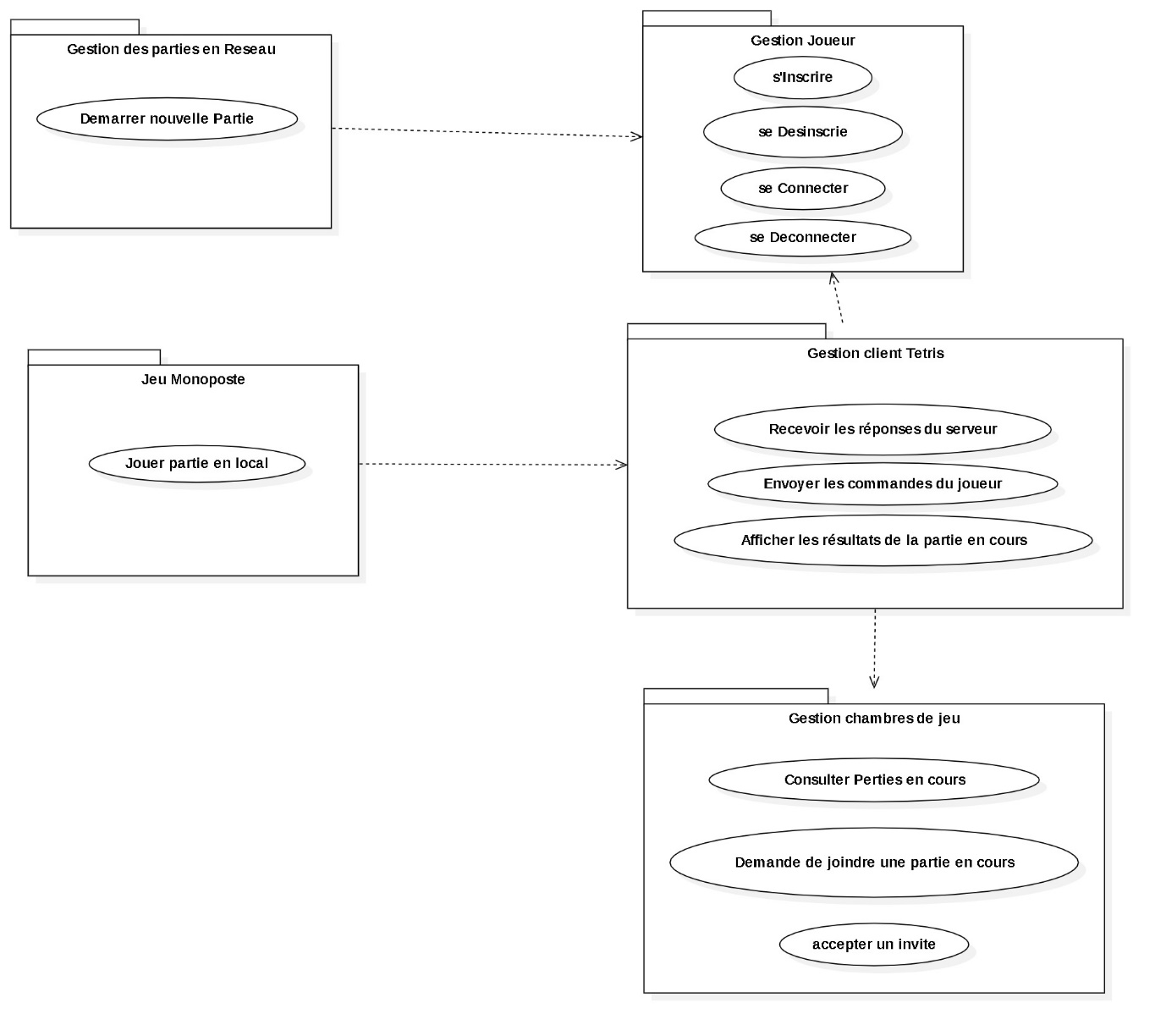


Figure x.x : Diagramme de paquetage des cas d’utilisation de tout le système.

## 4. Dialogue client et serveur Tetris

Pour assurer le fonctionnement du jeu en version réseau il faut standardiser les messages à échanger entre le client et le serveur. Pour se faire on se base sur le diagramme de flux du cas d’utilisation « Jouer partie en local » et on le repartit en taches client et serveur. Ce qui rend plus claire aussi les commandes client et leurs réponses serveur ainsi que les données nécessaires a échanger.

Les deux tableaux suivants résument :

* les commandes client et leurs significations ;
* et les réponses serveur, leurs signification ainsi que les données qui doivent les accompagner.

|  |  |  |
| --- | --- | --- |
| Commandes client | | |
| **Caractère** | **Signification** | **Données** |
| ‘S’ | Démarrer Partie | 1. Type partie :   * 0 : nouvelle partie * 1 : partie en cours   2. Numéro de partie en cours |
| KeyDown | Descendre tetramino un pas vers le bas |  |
| KeyLeft | Rotation tetramino gauche |  |
| ESC | Suspendre le jeu/ Reprendre le jeu |  |

|  |  |  |
| --- | --- | --- |
| Commandes Serveur | | |
| **Caractère** | **Signification** | **Données** |
| ‘O’ | Fin de Partie (GO) | Score, SorePodium |
| ‘T’ | Tetris : quatre couches remplies ; |  |
| 3 | 3 couches remplies ; | Score, Level |
| 2 | 2 couches remplies ; | Score, Level |
| 1 | 1 couche remplies ; | Score, Level |
| B | Tetramino bloqué | Nouveau Tetramino |

## 5. Diagrammes de déploiement

L’architecture technique du système : répartition des composants logiciels ou fonctionnalités du système sur les serveurs ou les nœuds matériels. Les composants logiciels sont déjà détaillés :

* Le client Tetris avec ses fonctionnalités peut être hébergé soit par un PC, soit un téléphone mobile soit par une console de jeu.
* La gestion de chambre de jeu ainsi que celle des joueurs est fait faite sur un système central. Le client tetris se connecte sur cet unique nœud. Il sera redirigé vers le serveur disponible de gestion des parties.
* La gestion de parties se fait sur un ou plusieurs serveurs. Ce qui rend l’architecture extensible. En fait la surcharge d’un serveur sera compense par l’addition d’un nouveau serveur a cette ferme en notifiant la chambre de jeu.

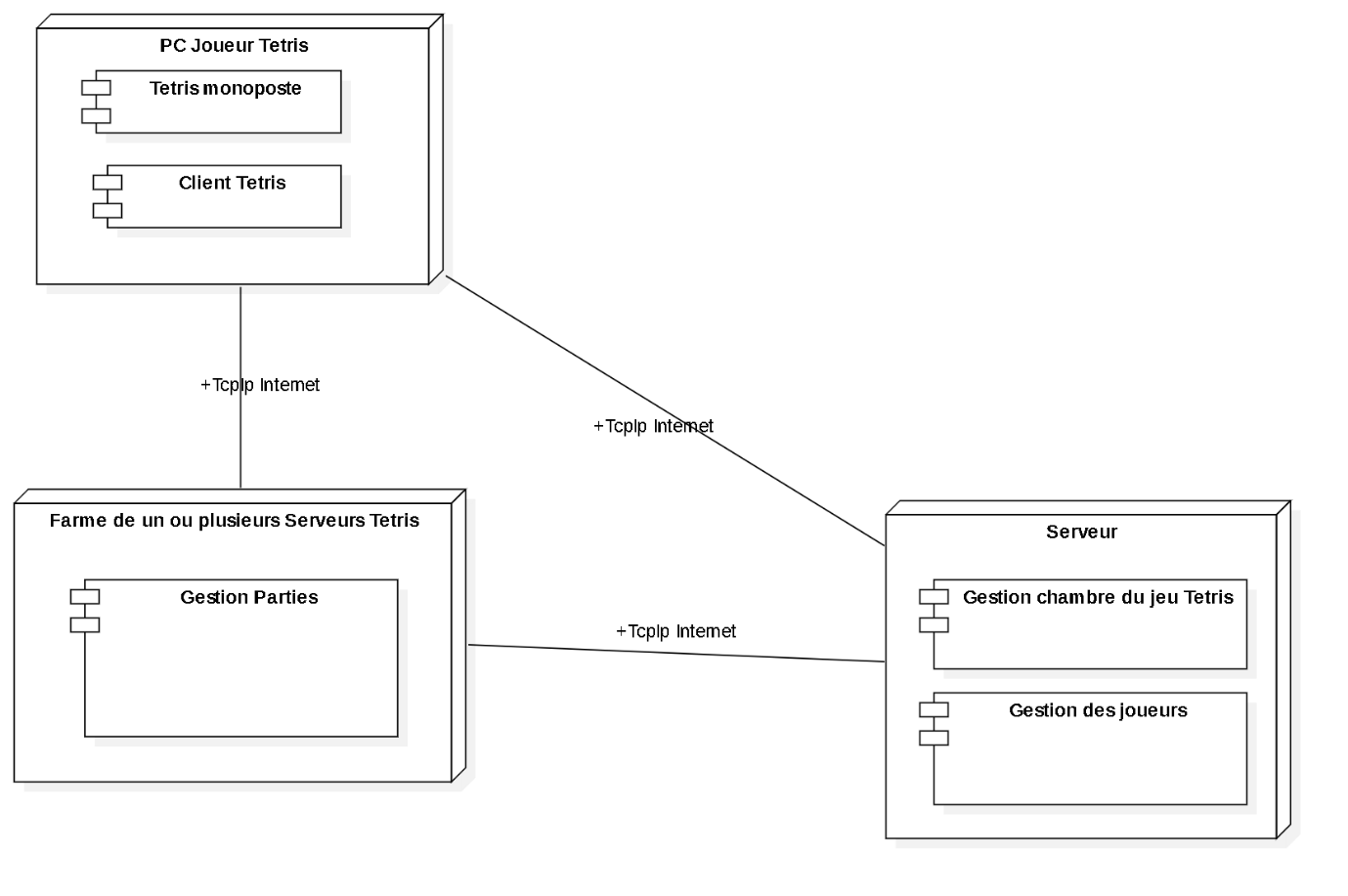


Figure x.x : Diagramme de déploiement et d’architecture du systeme.

## 6. Diagramme de classes du jeu en réseau

Il faut decrire ici le contenu du diagramme.

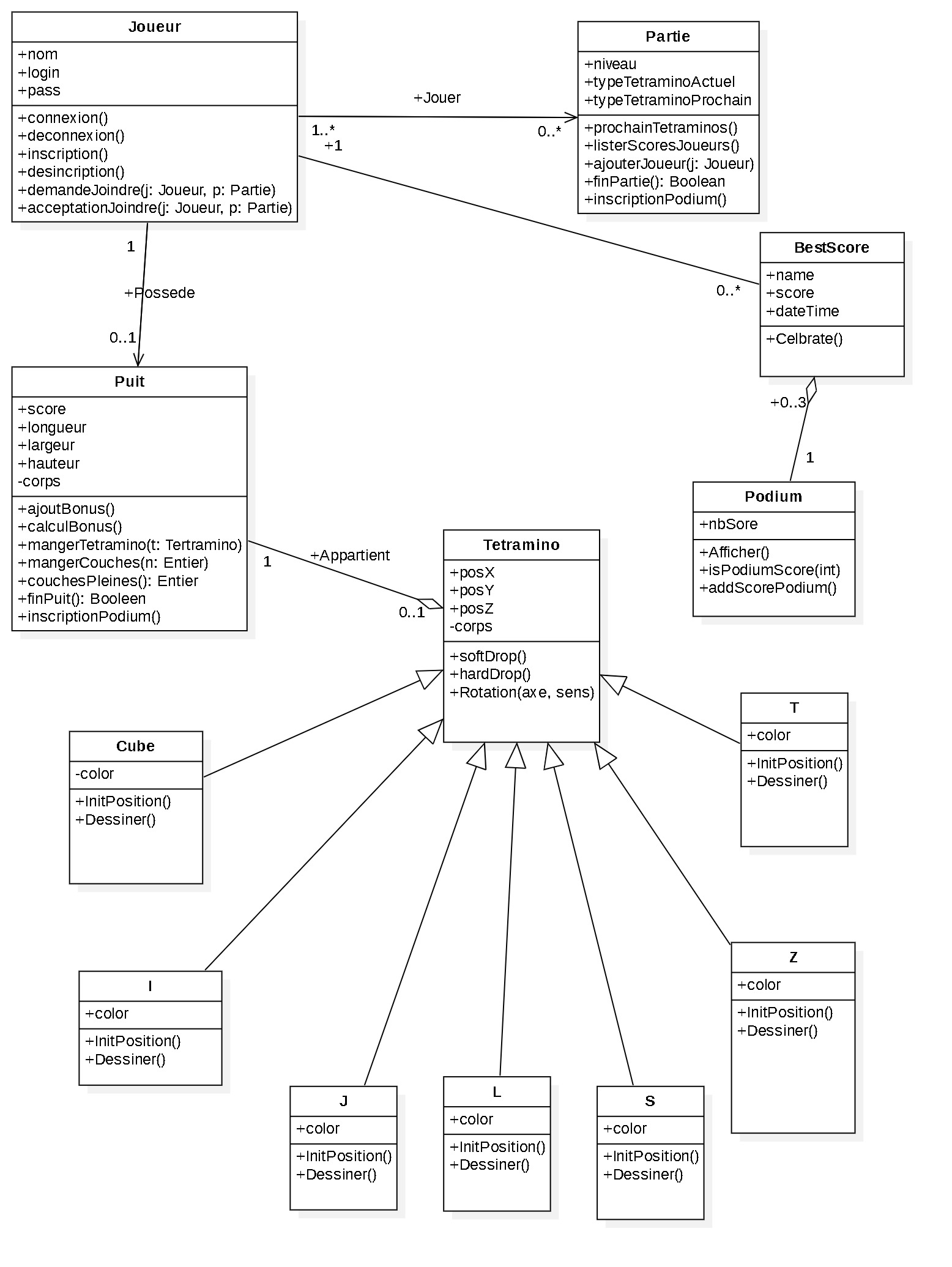


Figure x.x : Diagramme de classes.