

Protocole

Format des requêtes

```
class Request {  
    private String identifier;  
    private Object content;  
}
```

Liste des types de requête

Lobby

- `client.lobby.username, String` Envoie la demande de pseudonyme d'un joueur.
- `server.lobby.username, String` Réponse à la demande de pseudonyme. Le contenu est la chaîne vide si la pseudonyme est accepté, dans le cas contraire il est un message d'erreur.
- `client.lobby.refresh, null` Demande la liste des parties du serveur.
- `server.lobby.refresh, Set<GameInfo>` Renvoie la liste des parties du serveur.
- `client.lobby.quit, null` Quitte la partie que le client avait rejoint.
- `client.lobby.disconnect, null` Déconnexion.
- `client.lobby.create, GameInfo` Crée une partie.
- `server.lobby.create, Boolean` Renvoie `true` si la partie a bien été créée, sinon `false`.
- `client.lobby.message, String` Envoie un message sur le chat principal.
- `client.lobby.join, GameInfo` Rejoindre une partie.
- `server.lobby.join, Boolean` Contient `true` si la partie a bien été rejointe, sinon `false`.

Game

Préparation de la partie

- `client.game.players.refresh, null` Demande une mise à jour de la liste de joueurs.
- `server.game.players.refresh, List<String>` Envoie une mise à jour de la liste de joueurs.
- `server.game.start, List<List<String>>` Indique aux joueurs que la partie commence en communiquant la liste des équipes.

Formation des équipes

- `client.game.teams.refresh, null` Demande une mise à jour de la liste des équipes.
- `server.game.teams.refresh, List<List<String>>` Envoie une mise à jour de la liste des équipes.
- `server.game.teams.leader, null` Indique au client qu'il est chef d'équipe et qu'il doit choisir ses équipiers.
- `client.game.teams.leader, List<String>` Communique au serveur ses équipiers.

Partie

- `client.game.move.request, Movement` Envoie un déplacement.
- `server.game.move.request, Boolean` Accepte ou rejette un déplacement.
- `server.game.move, Movement` Notifie les joueurs d'un déplacement.
- `server.game.next, null` Fait passer au joueur suivant.
- `server.game.end, Integer` Indique que la partie est finie en spécifiant l'identifiant de l'équipe gagnante.
- `server.game.tick, Integer` Indique au client qu'il lui reste un certain nombre de secondes pour jouer.
- `client.game.quit` Indique qu'un joueur a quitté la partie. Il est remplacé par une IA. *(NB : s'il ne reste plus que des IA, la partie s'arrête.)*

Chat

- `client.game.message, String` Envoie un message.
- `server.game.message, String` Fait passer le message sous le format [`<Nom du joueur>`]`<Contenu du message>`.