

## Serveur de jeu, dialogue client/serveur JSON

### *Vue d'ensemble du protocole des services*

#### Codes :

```
public final int START          = 100;
public final int STOP          = 101;

public final int OK            = 200;

public final int MSG           = 301;
public final int PLAYER        = 302;
public final int CREATURE      = 303;
public final int ANIMATION     = 304;
public final int WAVE          = 305;
public final int PLAY          = 306;
public final int TOWER         = 307;
public final int TOWER_DEL     = 306;
public final int TOWER_UP      = 307;
public final int OBJECT        = 350;

public final int NO_MONEY      = 400;
public final int BAD_POS       = 401;
```

#### Liste des types de message :

Etat d'un joueur

Création d'une tour

Lancement d'une  
vague

Amélioration d'une  
tour

Modification de l'état  
d'une partie

Suppression d'une  
tour

Envoi d'un message

Affichage des objets

## Checklist du protocole

Etat	Libellé	Description
<input type="checkbox"/>	Définition des constantes	Mise en place de l'ensemble des constantes listées dans le protocole, plus d'éventuelles modifications à apporter par la suite.
<input type="checkbox"/>	Envoi d'un message	Envoi de messages textes entre le client et serveur, sous forme de ping/pong ou de messages de discussions.
<input type="checkbox"/>	Modification de l'état d'une partie	Envoi des messages de modification de l'état d'une partie par un joueur, en jeu ou hors jeu.
<input type="checkbox"/>	Etat d'un joueur	Envoi des messages de modification de l'état d'un joueur, en jeu ou hors jeu.
<input type="checkbox"/>	Lancement d'une vague	Envoi du signal de lancement d'une vague par un joueur au serveur, avec la réponse associée.
<input type="checkbox"/>	Création d'une tour	Création d'une tour par le client, avec le message de retour associé en cas de réussite ou d'erreur.
<input type="checkbox"/>	Amélioration d'une tour	Amélioration d'une tour par son propriétaire déjà en place, avec comme retour un message associé en cas de réussite ou d'erreur.
<input type="checkbox"/>	Suppression d'une tour	Suppression d'une tour par son propriétaire, avec message de retour associé en cas de réussite ou d'erreur.
<input type="checkbox"/>	Affichage des objets	Ici prendra place la partie la plus complexe. En effet cette partie concernera le retour du serveur à chaque client permettant d'afficher les objets de jeu pour maintenir un affichage cohérent entre chaque client et le serveur.

Attention, pour une description plus complète de chaque type de service, se référer à la page correspondante du présent document.

Affichage d'un objet (Générique) (UDP)

Client

ServeurJeu

```
{
  "TYPE"      : OBJECT
  "Object"    : ID
  "Etat"      : ETAT
  "x"         : X
  "y"         : Y
}
```

Scénario d'utilisation

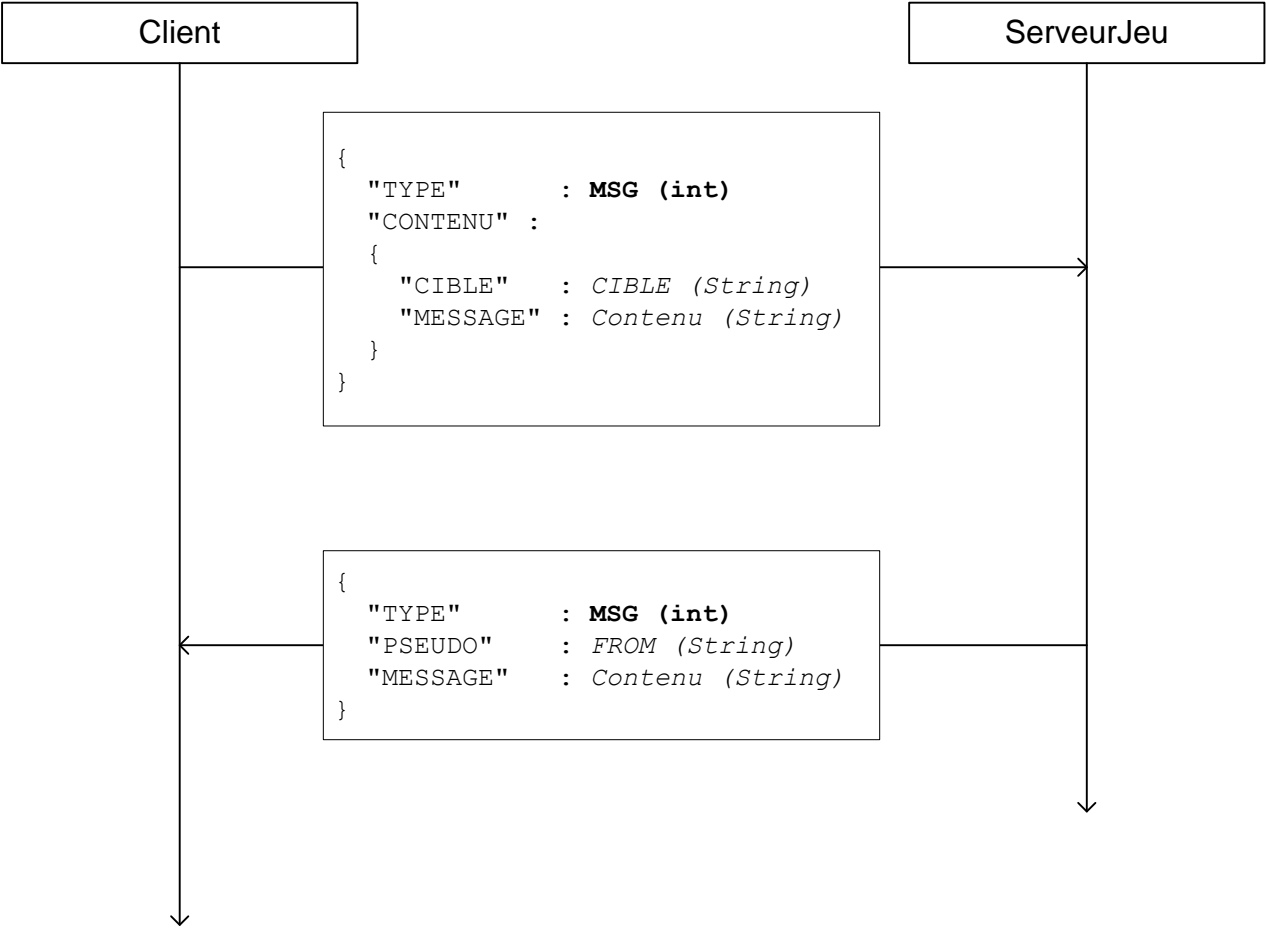
Action Client

2. Réception du paquet et traitement des données.

Action Serveur

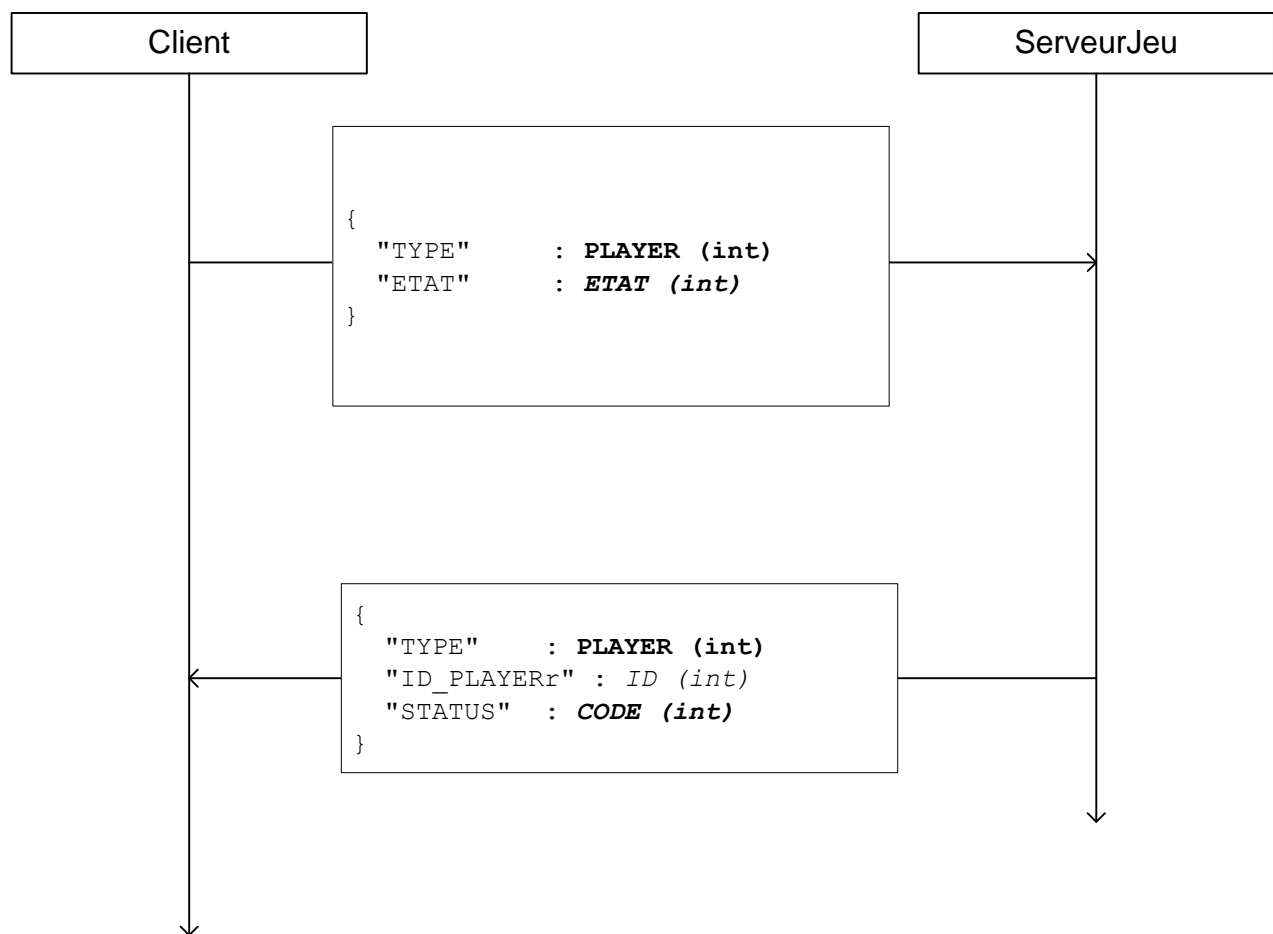
1. Envoi d'un objet de différent type avec différentes propriétés à tous les clients.

Envoi d'un message (TCP)



Scénario d'utilisation	
Action Client	Action Serveur
1. Envoi d'un message texte au serveur avec un éventuel destinataire.	2. Réception du message et traitement interne.
2. Réception du message texte du serveur et traitement interne.	1. Envoi d'un message texte à un destinataire spécifique.

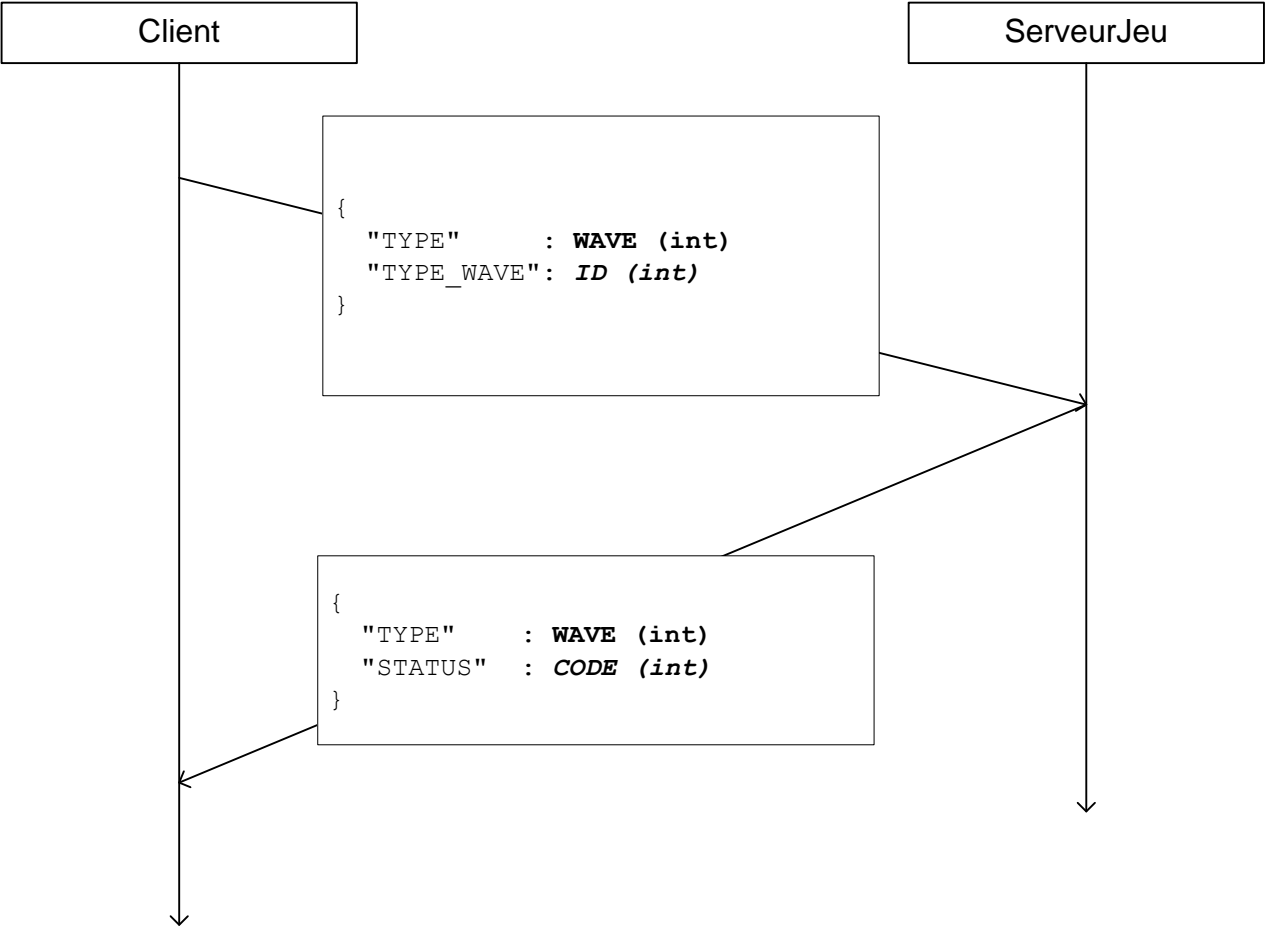
Etat d'un joueur : changements signalés (TCP)



Scénario d'utilisation

Action Client	Action Serveur
1. Envoi du changement d'état du joueur courant au serveur.	2. Réception du changement d'état et traitement interne.
2. Réception du changement d'état du joueur identifié.	1. Envoi du changement d'état d'un joueur à l'ensemble des joueurs

Lancement d'une vague (de client à serveur) (TCP)



Scénario d'utilisation

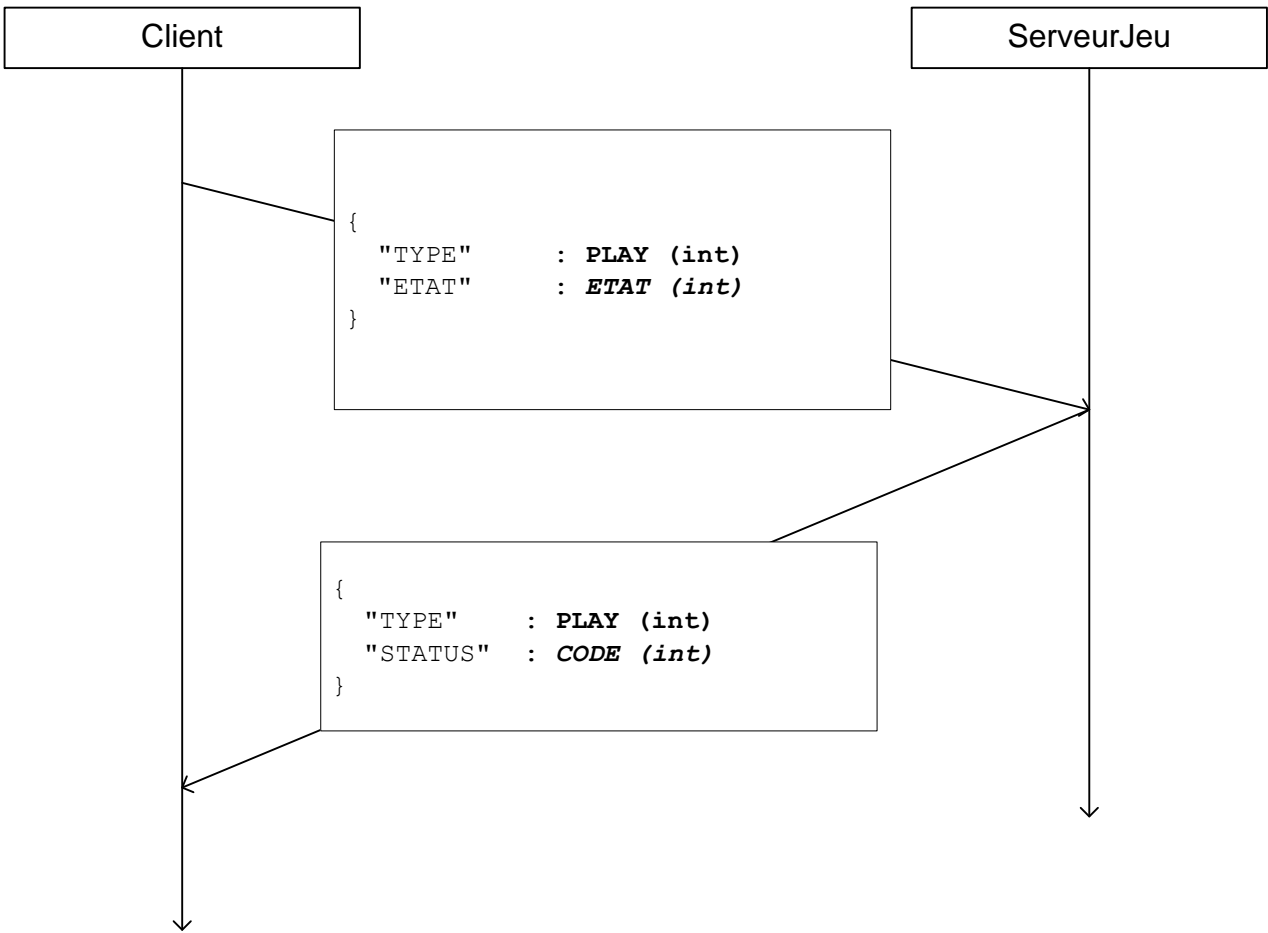
Action Client

1. Envoi d'une demande de lancement d'une vague au serveur avec différents paramètres discriminants.
3. Réception du message d'état et traitement interne.

Action Serveur

2. Réception de la requête, traitement interne et envoi d'une message de statu avec un code indiquant l'état de l'opération.

Changement de l'état d'une partie (TCP)



Scénario d'utilisation

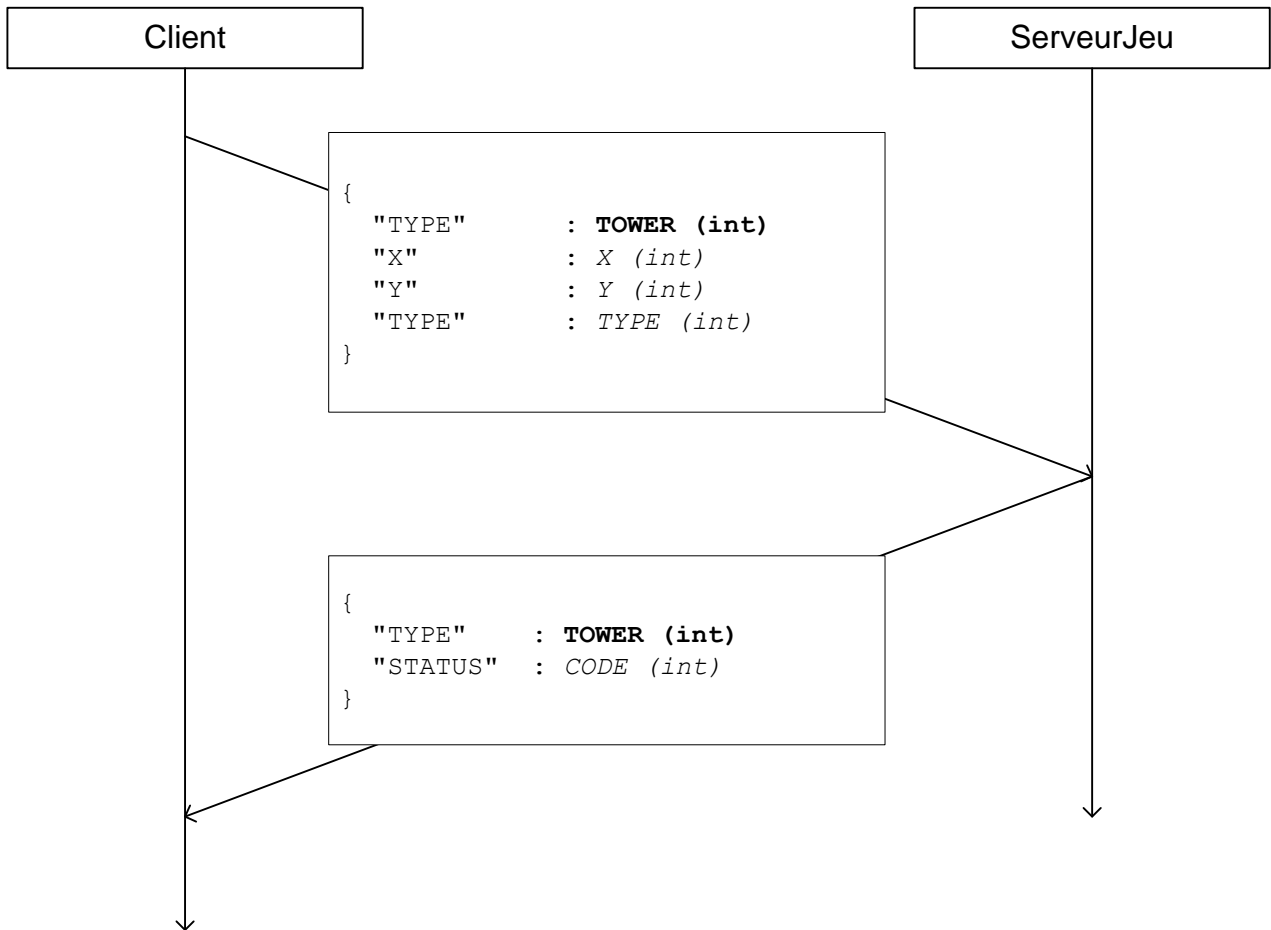
Action Client

1. Envoi de du changement d'état désiré de la partie au serveur.
3. Réception de la réponse du serveur et traitement interne.

Action Serveur

2. Réception de la requête, traitement interne, puis envoi au client la réponse en cas d'erreur ou de réussite.

## Création d'une Tour : demande du client (TCP)

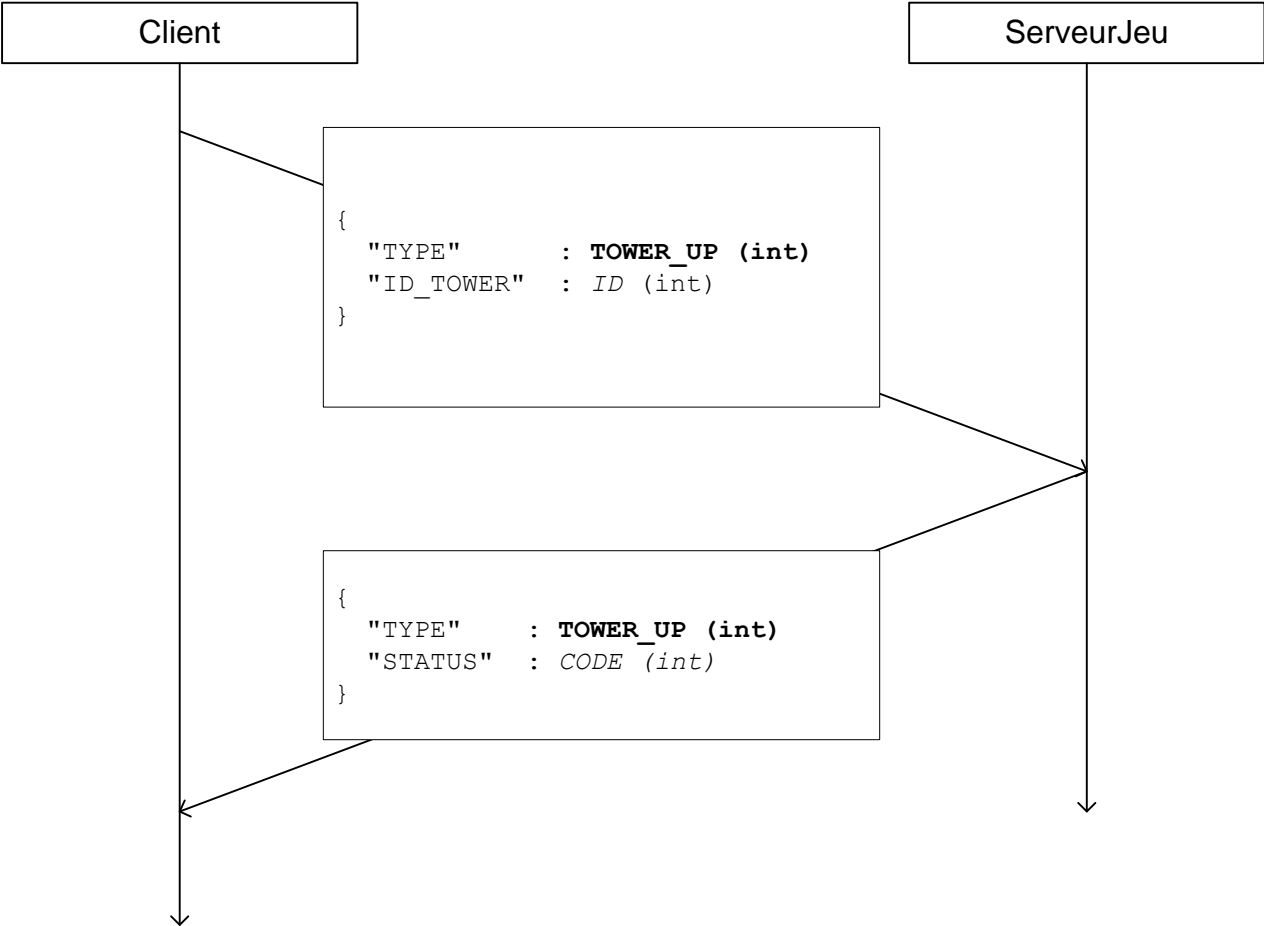


### Scénario d'utilisation

Action Client	Action Serveur
1. Envoi d'une demande de création de tour au serveur avec les différents paramètres.	2. Réception de la requête, traitement interne, puis envoi au client d'une réponse en cas d'erreur ou de réussite.
3. Réception de la réponse et traitement interne.	



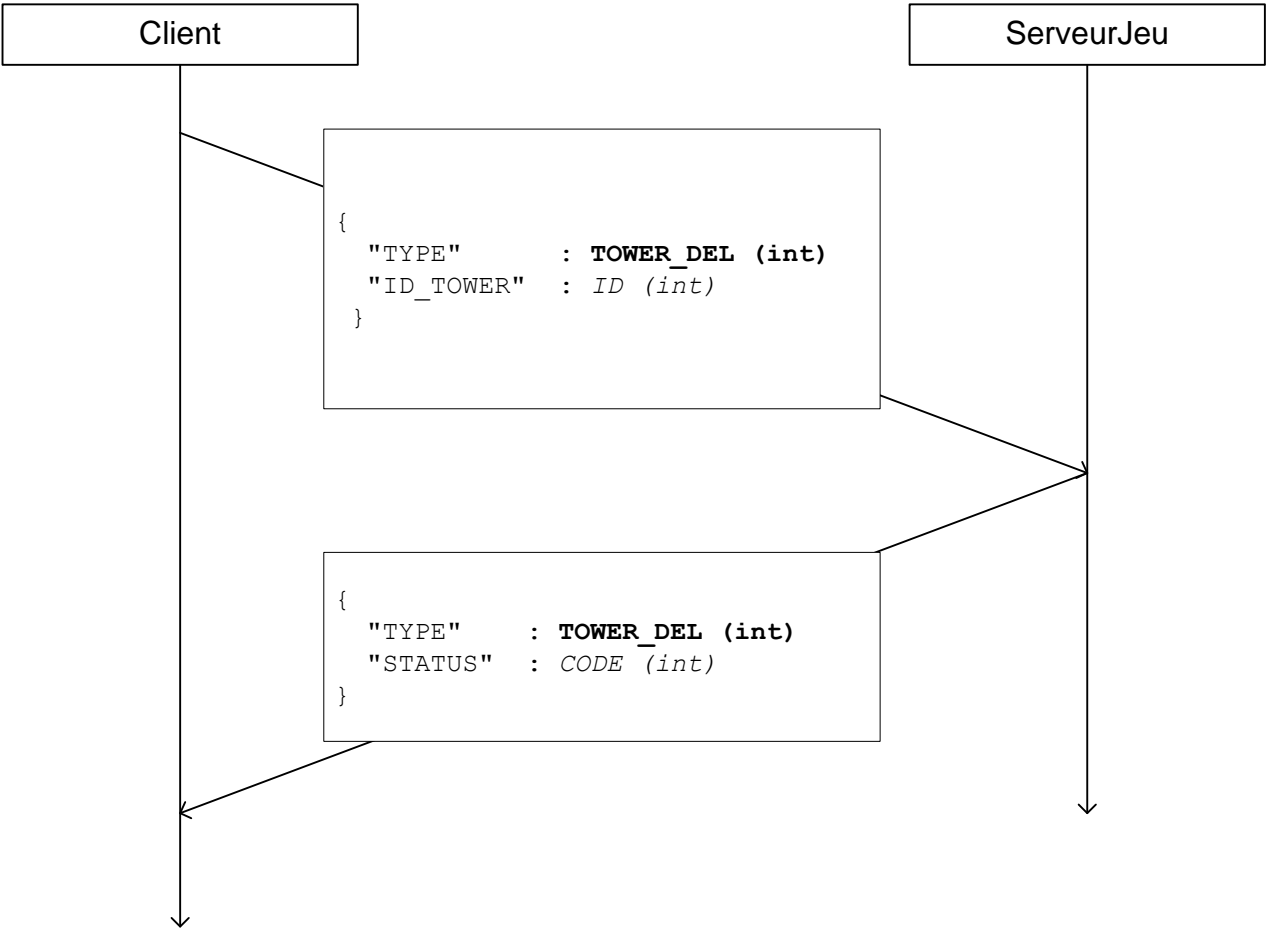
<b>Amélioration d'une Tour : demande du client (TCP)</b>
--



### Scénario d'utilisation

<i>Action Client</i>	<i>Action Serveur</i>
<p>1. Envoi au serveur d'une demande d'amélioration d'une tour.</p>     <p>3. Réception de la réponse du serveur et traitement interne.</p>	<p>2. Réception de la demande, traitement interne, puis envoi au client concerné un code en fonction de la réussite ou de l'échec de l'opération.</p>

Suppression d'une Tour : demande du client (TCP)



Scénario d'utilisation

Action Client

- 1. Envoi d'une demande de suppression de tour au serveur.
- 3. Réception de la réponse du serveur et traitement interne.

Action Serveur

- 2. Réception de la requête du client, traitement interne et envoi d'une réponse contenant un code de retour relatif au succès ou à l'échec de l'opération