

DAI - PW2

Pedro Alves da Silva

Gonçalo Carvalho Heleno

Contents

Overview du protocole	1
Description du jeu télévisé	1
Description du jeu numérique	2
Spécifications	2
Messages	3
END	3
FILL	4
GO	4
GUESS	4
INFO	5
JOIN	5
LAST	6
LETTERS	6
LOBBY	6
QUIT	7
ROUND	7
SKIP	7
START	8
STATUS	8
TURN	9
VOWEL	9
WINNER	10
Exemples	10
Connection d'un joueur	10
Tour d'un joueur	11
Fin de manche	12
Fin du jeu	12
Lifecycles	13
Client	13
Server	14

Overview du protocole

Ce protocole permet d'implémenter les interactions serveur ↔ client pour permettre à plusieurs clients de jouer à “Wheel of Fortune”.

Description du jeu télévisé

Dans le jeu télévisé, les joueurs se réunissent autour d'une roue divisée en plusieurs parties, appelées de *wedges*. Celles-ci peuvent soit contenir un prix, soit passer leur tour, soit leur enlever l'argent cumulé depuis le début de la partie. Les *wedges* contenant les prix peuvent contenir soit:

- Une somme d'argent fixe à gagner

- Une somme d'argent fixe, cachée à gagner
- Un voyage offert
- Le jackpot d'un million de dollars
- Quelques prix supplémentaires selon les éditions du jeu

Chaque manche commence avec l'affichage du puzzle à trouver, ainsi que sa catégorie. Lorsqu'un joueur tourne la roue et tombe sur une *wedge* prix, il devra deviner une consonne qui doit se trouver sur le puzzle de la manche. La consonne ne doit pas encore être utilisée. Si la consonne n'existe pas sur le puzzle, le tour fini, et le prochain joueur commence son tour. Si la consonne existe, le joueur a trois choix:

- Acheter une voyelle, dont le prix augmente à chaque manche
- Tenter de dire ce qui se cache derrière le puzzle
- Passer son tour s'il n'est pas possible d'acheter une voyelle, et le joueur ne veut pas tenter sa chance en remplissant le puzzle

Dans le cas où le joueur achète une voyelle, et que celle-ci existe sur le puzzle, le joueur gagne le prix fois le nombre de fois que ladite voyelle apparaît sur le puzzle. Son tour recommence. Si la voyelle n'existe pas, le prochain joueur commence son tour. Si le joueur devine correctement le puzzle, la manche se termine immédiatement, et le joueur remporte la manche.

Après plusieurs manches, seul le joueur ayant la somme d'argent la plus haute participe au dernier puzzle. Le puzzle contient 5 lettres déjà dévoilées: R, S, T, E, L, et N. Celles-ci sont les lettres les plus probables de trouver dans un texte en anglais. Le joueur doit, ensuite, donner 5 lettres de son choix pour qu'elles soient dévoilées sur le puzzle aussi. Ensuite, il y a une quantité maximale de temps pour trouver le puzzle au complet, sans pouvoir deviner d'autres lettres.

Afin d'éviter un long temps d'attente de la réponse du joueur en train de jouer, le jeu télévisé utilise des temps de réponse maximales. Si le joueur dépasse ce temps de réponse lors d'une manche normale, il perd son tour. S'il dépasse ce temps lors de la dernière manche, il perd la partie.

Description du jeu numérique

Lorsque les joueurs se connectent au serveur, ils se retrouvent dans une salle d'attente, en attendant qu'un joueur démarre la partie. Lorsque la partie est démarrée, un puzzle est choisi, et le premier joueur à se connecter au serveur reçoit une *wedge* aléatoirement choisie par le serveur. Selon son type, tout comme pour le jeu télévisé, il peut soit perdre son tour, soit faire faillite, soit tomber sur une *wedge* contenant un prix.

Ce protocole n'a pas été pensé pour inclure les *wedges* spéciales, telles que celles contenant un montant caché, ou un voyage offert.

Le joueur doit, alors, effectuer les mêmes étapes comme s'il se trouvait dans le jeu télévisé.

Au contraire du jeu de TV, ce protocole prévoit du temps maximales dans certaines des commandes, mais celles-ci sont inutilisées dans la présente version. Ceci veut dire que les joueurs n'ont pas un temps maximale pour présenter leur réponse, et le serveur ne le vérifie pas non plus.

Les restantes étapes du jeu se déroulent comme le jeu original. Toutefois, après la dernière manche, tous les joueurs se retrouvent à nouveau dans la salle d'attente, et peuvent reparticiper à une nouvelle partie.

Spécifications

Les communications entre les différentes parties sont échangées en texte clair, encodé en UTF-8.

Ces échanges sont faits utilisant le port **TCP** 1234. Le délimiteur est le caractère **new-line**, aussi connu par **\n**.

Les joueurs sont ceux qui doivent démarrer la communication.

Les joueurs ne peuvent pas envoyer arbitrairement des commandes au serveur, car ils devront attendre que le serveur interagisse avec eux. Exception faite pour les commandes **JOIN** et **QUIT**.

D'après les règles du jeu, la lettre 'Y' est une consonne, et ne peut, donc, pas être achetée avec la commande **VOWEL**.

Plusieurs joueurs peuvent se connecter à partir de la même IP. Ceci permet à des joueurs se trouvant dans la même maison de se connecter au serveur, mais permet également au même joueur de se connecter plusieurs fois depuis un même appareil.

Le protocole ne gère pas des permissions par joueur. Ceci fait que n'importe quel joueur peut démarrer la partie.

Si un joueur quitte la partie lorsque celle-ci est en cours, il ne parviendra pas à se reconnecter.

Les joueurs sont libres de choisir leur nom d'utilisateur, et peuvent être composés de plusieurs mots, délimités par des doubles guillemets ("). Le seul nom d'utilisateur qui n'est pas acceptable est un simple trait (-), qui est réservé pour la commande **END** dans le cas où aucun joueur n'a remporté la partie.

Messages

Table 1: Liste de tous les messages qui peuvent être échangés

Message	Server → Client	Server ← Client	Description
END	✓		Annonce de la fin du jeu, avec résultats
FILL		✓	Le joueur essaye de compléter le puzzle
GO		✓	Démarre la partie
GUESS		✓	Le joueur essaye devine une consonne
INFO	✓		Envoie les toutes dernières informations de la manche
JOIN		✓	Demande au serveur d'autoriser un joueur à rejoindre
LAST	✓		Demande au gagnant de deviner le puzzle
LETTERS		✓	Le joueur fourni 5 lettres à révéler pour la dernière manche
LOBBY	✓		Envoie la liste actuelle de joueurs dans la partie
QUIT		✓	Demande la déconnection d'un joueur
ROUND	✓		Fin de tour: le puzzle a été résolu
SKIP		✓	Saute le tour après avoir deviné une consonne
STATUS	✓	✓	Renvoie un code de status à l'autre bout
TURN	✓		Le serveur tourne la roue pour un joueur
VOWEL		✓	Le joueur achète une voyelle
WINNER	✓		Annonce le gagnant des manches. Début du dernier puzzle

END

Cette commande annonce la fin du jeu à tous les joueurs dans le lobby. Cette commande fournit les résultats finaux, permettant l'application Client de les afficher à l'écran pour que les joueurs puissent les lire.

Table 2: Tableau du format et réponse acceptées

Nom	Description
Format	END <winner> <player_1> <money1_> ... <player_n> <money_n>
Server → Client	✓
Client → Server	
Réponses acceptées	Aucune

Table 3: Paramètres de la commande

Nom	Description
winner	Username du joueur qui a gagné. Si aucun joueur a gagné, username = “-”
player_n	Username du joueur n de la liste
money_n	Argent gagné par le joueur n de la liste

FILL

Cette commande est utilisée quand le joueur veut tenter sa chance en devinant le puzzle.

Table 4: Tableau du format et réponse acceptées

Nom	Description
Format	FILL <puzzle>
Server → Client	
Client → Server	✓
Réponses acceptées	STATUS wrong_answer si la réponse est fausse STATUS right_answer si la réponse est correcte

Le paramètre **puzzle** doit être envoyé entre guillemets. Les lettres sont case-insensitive, mais doivent être envoyées en majuscule.

Table 5: Paramètres de la commande

Nom	Description
puzzle	Tentative de résolution du puzzle

GO

Cette commande démarre la partie.

Table 6: Tableau du format et réponse acceptées

Nom	Description
Format	GO
Server → Client	
Client → Server	✓
Réponses acceptées	Aucune réponse attendue

GUESS

Utilisée lors du début du tour, quand le joueur essaye de deviner une consonne qui peut se trouver dans le puzzle à résoudre.

Table 7: Tableau du format et réponse acceptées

Nom	Description
Format	GUESS <letter>
Server → Client	
Client → Server	✓
Réponses acceptées	STATUS timeout si le joueur n’a pas répondu à temps STATUS letter_missing si la lettre n’existe pas dans le puzzle

Nom	Description
	STATUS <code>letter_exists</code> si la lettre existe dans le puzzle
	STATUS <code>wrong_format</code> si la lettre n'est pas une consonne
	STATUS <code>already_tried</code> si la lettre a déjà été utilisée

Table 8: Paramètres de la commande

Nom	Taille [B]	Description
letter	1	Consonne à vérifier par le serveur

INFO

Fourni les toutes dernières informations de la manche à un joueur. L'envoi de cette commande démarre le timer d'attente de la réponse du joueur.

Table 9: Tableau du format et réponse acceptées

Nom	Description
Format	INFO <puzzle> <category> [used_letters]
Server → Client	✓
Client → Server	
Réponses acceptées	GUESS quand le joueur essaie de deviner une consonne

Le paramètre `puzzle` et `category` doivent être envoyés entre guillemets. Les lettres sont case-insensitive, mais doivent être envoyées en majuscule. De plus, les cases à trouver sont remplacées par `*`.

Table 10: Paramètres de la commande

Nom	Description
puzzle	Puzzle à trouver
category	Catégorie du puzzle
used_letters	Optionnel. Lettres déjà utilisées

JOIN

Demande au serveur l'autorisation de connection d'un joueur.

Table 11: Tableau du format et réponse acceptées

Nom	Description
Format	JOIN <username>
Server → Client	
Client → Server	✓
Réponses acceptées	STATUS <code>ok</code> si le serveur a accepté la connection
	STATUS <code>full</code> si la partie est déjà au complet
	STATUS <code>wrong_format</code> si l'username n'est pas valable
	STATUS <code>duplicate_name</code> si un autre joueur avec ce username est déjà dans la partie

Table 12: Paramètres de la commande

Nom	Description
username	Username souhaité

LAST

Informe un joueur que ça à son tour, et que c'est le dernier puzzle de la partie.

Table 13: Tableau du format et réponse acceptées

Nom	Description
Format	LAST <timeout> <puzzle> <category>
Server → Client	✓
Client → Server	
Réponses acceptées	FILL

Le paramètre **puzzle** et **category** doivent être envoyés entre guillemets. Les lettres sont case-insensitive, mais doivent être envoyées en majuscule. De plus, les cases à trouver sont remplacées par *.

Table 14: Paramètres de la commande

Nom	Description
timeout	Nombre de secondes, avant que le joueur perde la partie
puzzle	Puzzle à résoudre
category	Catégorie du puzzle

LETTERS

Le joueur fourni 5 lettres à révéler sur le puzzle de la dernière manche

Table 15: Tableau du format et réponse acceptées

Nom	Description
Format	LETTERS <letters>
Server → Client	
Client → Server	✓
Réponses acceptées	STATUS ko si le format est faux STATUS already_tried si les lettres ont déjà été utilisées ROUND si les lettres sont acceptées

Table 16: Paramètres de la commande

Nom	Taille [B]	Description
letters	5	Lettres demandées

LOBBY

Fourni la liste de tous les joueurs actuellement connectés dans la partie.

Table 17: Tableau du format et réponse acceptées

Nom	Description
Format	LOBBY <player_1> ... <player_n>
Server → Client	✓
Client → Server	
Réponses acceptées	Aucune

Table 18: Paramètres de la commande

Nom	Description
player_n	Username du joueur dans le lobby

QUIT

Informe le serveur de la déconnection d'un joueur.

Table 19: Tableau du format et réponse acceptées

Nom	Description
Format	QUIT
Server → Client	
Client → Server	✓
Réponses acceptées	Aucune

: Paramètres de la commande

ROUND

Informe tous les joueurs de la fin de la manche.

Table 20: Tableau du format et réponse acceptées

Nom	Description
Format	ROUND <puzzle>
Server → Client	✓
Client → Server	
Réponses acceptées	Aucune

Le paramètre **puzzle** doit être envoyé entre guillemets. Les lettres sont case-insensitive, mais doivent être envoyées en majuscule.

Table 21: Paramètres de la commande

Nom	Description
puzzle	Le puzzle complété

SKIP

Lorsque le joueur qui joue actuellement envoie cette commande, son tour est sauté. Il n'est possible de l'envoyer qu'après avoir deviné une consonne.

Table 22: Tableau du format et réponse acceptées

Nom	Description
Format	SKIP
Server → Client	
Client → Server	✓
Réponses acceptées	Aucune réponse attendue

START

Informe tous les joueurs du début de une nouvelle manche.

Table 23: Tableau du format et réponse acceptées

Nom	Description
Format	START <round_number> <puzzle> <category>
Server → Client	✓
Client → Server	
Réponses acceptées	Aucune

Le paramètre **puzzle** et **category** doivent être envoyés entre guillemets. Les lettres sont case-insensitive, mais doivent être envoyées en majuscule. De plus, les cases à trouver sont remplacées par *****.

Table 24: Paramètres de la commande

Nom	Description
round_number	Le numéro de la manche qui a commencé
puzzle	Le puzzle à compléter
category	La catégorie du puzzle

STATUS

Un des clients fournit une réponse à une information venant du serveur, ou le serveur fournit une réponse à une information venant du client. La liste des codes de réponse est connue par les deux parties. Dans un projet plus conséquent, une vérification de version entre Serveur ↔ Client s'imposerait pour s'assurer que les deux ont des versions compatibles. Toutefois, pour un laboratoire de cette taille, nous avons jugé ce point dénécessaire.

Table 25: Tableau du format et réponse acceptées

Nom	Description
Format	STATUS <status>
Server → Client	✓
Client → Server	✓
Réponses acceptées	Aucune

Table 26: Paramètres de la commande

Nom	Description
Status	Nom du code de status

Les valeurs marquées comme prioritaires doivent toujours exécuter le code lui appartenant, et ensuite revenir à l'état précédent, à l'exception de CLOSING qui oblige le joueur à revenir à l'état "Idle".

Table 27: Valeur acceptées pour la commande **STATUS**

Valeur	Nom	Prio.?	Description
0x00	OK		Réponse acceptée ou acceptable. Requête traitée correctement
0x01	KO		Erreur, mais l'erreur ne peut pas être spécifiée
0x02	PLAYER_JOINED	✓	Un joueur a rejoint la partie. La commande LOBBY sera envoyée aussi
0x03	PLAYER_QUIT	✓	Un joueur a quitté la partie. La commande LOBBY sera envoyée aussi
0x04	LETTER_EXISTS		La lettre fournie existe. Le joueur peut continuer son tour
0x05	LETTER_MISSING		La lettre fournie n'existe pas. Le tour du joueur est terminé
0x06	TIMEOUT		Le temps est écoulé. Le tour du joueur est terminé
0x07	ALREADY_TRIED		La lettre a déjà été utilisée dans le puzzle
0x08	WRONG_ANSWER		La tentative de résolution du puzzle a échoué. Fin du tour
0x09	RIGHT_ANSWER		La tentative de résolution du puzzle a réussi. Fin de la manche
0x0A	DUPLICATE_NAME		Un autre joueur avec ce nom est déjà présent dans le lobby
0x0B	NO_FUNDS		Le joueur n'a pas assez d'argent pour effectuer l'achat
0x0C	FULL		La partie est au complet
0x0D	LOST_A_TURN		Le joueur a perdu son tour
0x0E	BANKRUPT		Le joueur a perdu tout son argent, et son tour est passé

TURN

Informe un joueur que ça à son tour.

Table 28: Tableau du format et réponse acceptées

Nom	Description
Format	TURN <money> <total_money>
Server → Client	✓
Client → Server	
Réponses acceptées	Aucune

Table 29: Paramètres de la commande

Nom	Description
money	Somme d'argent gagnée pour cette manche
total_money	Somme d'argent cumulée depuis le début de la partie

VOWEL

Le joueur demande l'achat d'une voyelle spécifique.

Table 30: Tableau du format et réponse acceptées

Nom	Description
Format	VOWEL <vowel>
Server → Client	
Client → Server	✓
Réponses acceptées	STATUS letter_exists si la lettre existe dans le puzzle STATUS letter_missing si la lettre n'existe pas dans le puzzle STATUS no_funds si le joueur n'a pas assez d'argent

Nom	Description
-----	-------------

La lettre est case-insensitive, mais doit être envoyée en majuscule.

Table 31: Paramètres de la commande

Nom	Taille [B]	Description
vowel	1	Voyelle demandée

WINNER

Déclare le gagnant des manches précédentes. Le gagnant participe au dernier round.

Table 32: Tableau du format et réponse acceptées

Nom	Description
Format	WINNER <username>
Server → Client	✓
Client → Server	
Réponses acceptées	Aucune

Table 33: Paramètres de la commande

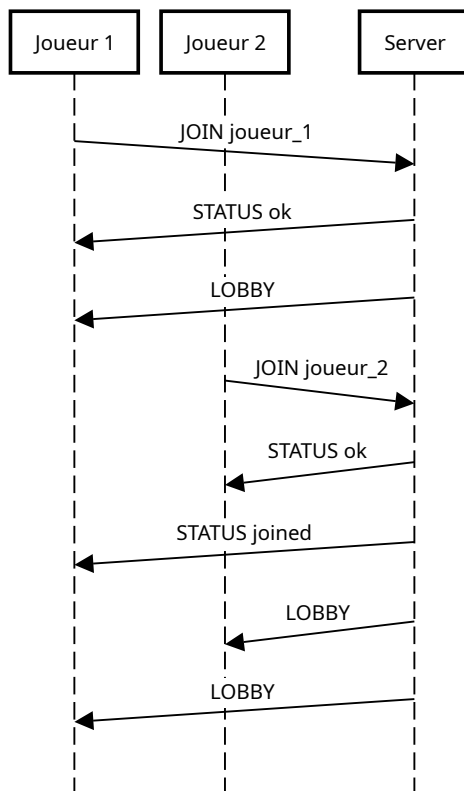
Nom	Description
username	Username du joueur qui participe à la dernière manche

Exemples

Connection d'un joueur

Un joueur envoie son username comme paramètre de la commande JOIN. Si la requête est acceptée par le serveur, il répondra avec un STATUS acceptable. La commande STATUS est envoyée aux autres joueurs dans le lobby pour les informer qu'un joueur a rejoint la partie. La command LOBBY est ensuite envoyée à chacun des joueurs dans le lobby.

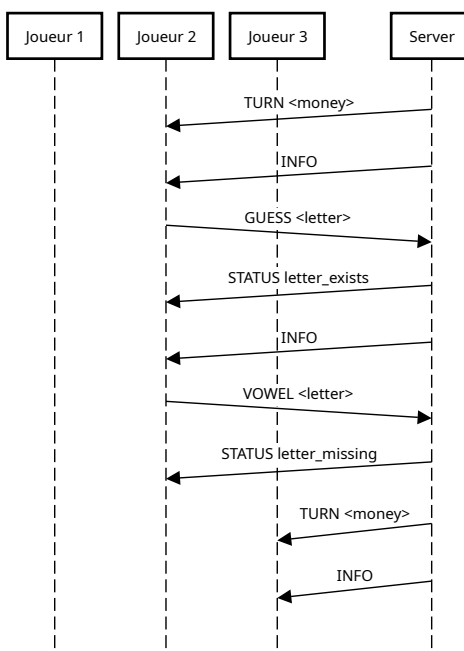
Joueurs rejoignent le lobby



Tour d'un joueur

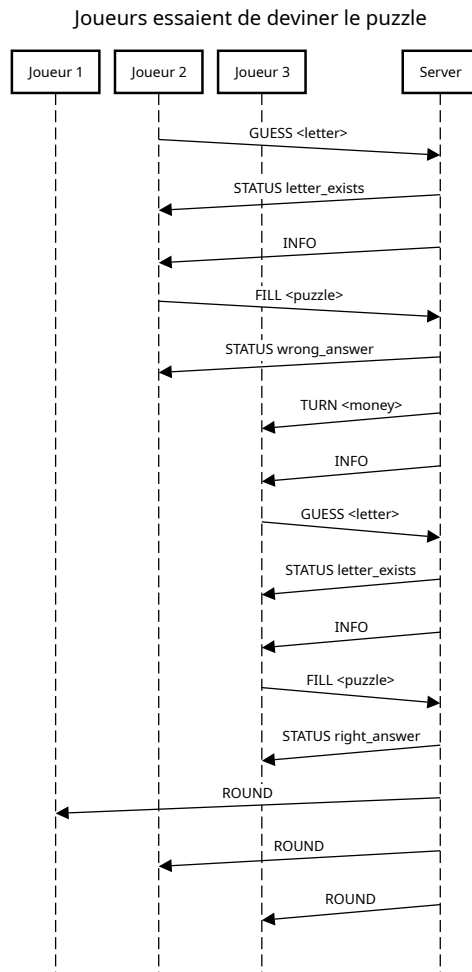
Le serveur informe un joueur que son tour a commencé. Il reçoit la somme d'argent obtenue, ainsi que le puzzle à résoudre. À la fin de son tour, le serveur effectuera le même échange avec le prochain participant dans la liste des joueurs.

Tour d'un joueur



Fin de manche

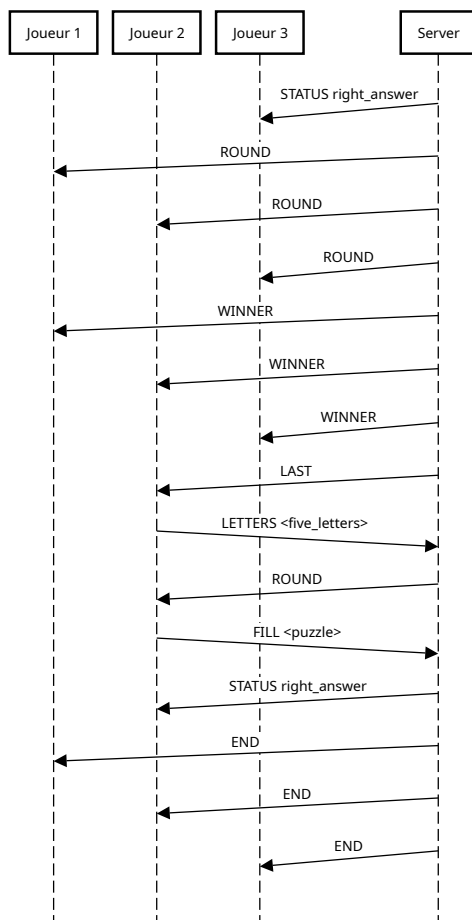
Lorsqu'un joueur devine correctement le puzzle, la fin de la manche est annoncée à chacun des joueurs dans la partie.



Fin du jeu

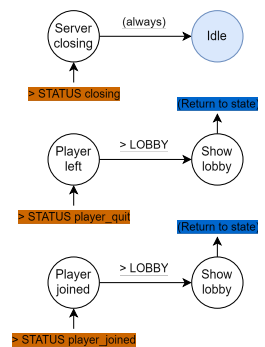
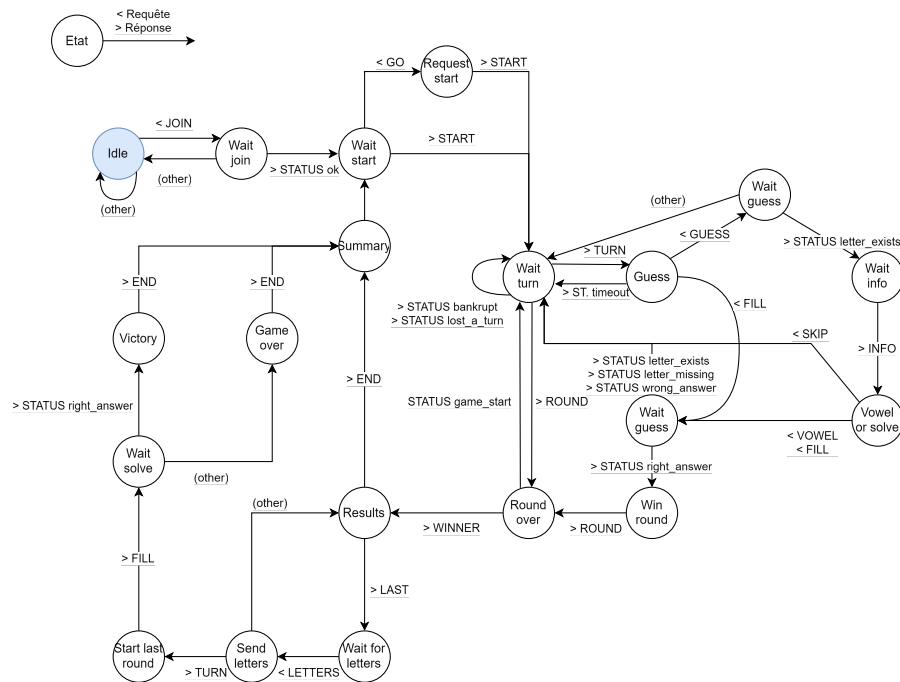
Lors de la dernière partie du jeu, le gagnant de toutes les manches précédentes est annoncé, et doit trouver le puzzle qui lui est affiché à l'écran. S'il parvient à la résoudre correctement dans le temps qui lui a été imparti, il gagne le jeu. Même s'il ne le gagne pas, tous les joueurs reçoivent le résultat du jeu.

Début du dernier puzzle, fin du jeu



Lifecycles

Client



Server

