

# SAÉ Socket

Création d'un jeu du pendu en C

## Sommaire :

- Introduction.
- Fonctionnement des versions.



# Introduction

Ce projet est un jeu du pendu codé en langage **C**, utilisant une architecture **client / serveur** avec des sockets.

Il est disponible en plusieurs versions, chacune apportant de nouvelles fonctionnalités et une complexité croissante.

Nous vous conseillons, dans un premier temps, de compiler l'ensemble des versions du jeu avant de les tester.

Toutes les commandes suivantes sont à exécuter depuis le répertoire **./saesocket**.

```
gcc ./src/V0/serveur_v0.c -o ./build/serveur_v0
gcc ./src/V0/client_v0.c -o ./build/client_v0
```

```
gcc ./src/V1/serveur_v1.c -o ./build/serveur_v1
gcc ./src/V1/client_v1.c -o ./build/client_v1
```

```
gcc ./src/V2/serveur_v2.c -o ./build/serveur_v2
gcc ./src/V2/client_v2.c -o ./build/client_v2
```

```
gcc ./src/V3/serveur_v3.c -o ./build/serveur_v3
gcc ./src/V3/client_v3.c -o ./build/client_v3
gcc ./src/v3/partie_v3.c -o ./build/partie_v3
```

```
gcc ./src/V4/serveur_v4.c -o ./build/serveur_v4
gcc ./src/V4/client_v4.c -o ./build/client_v4
```

Pour ce qui est de l'exécution des jeux, il faut exécuter pour chaque version successivement les commandes suivantes :

```
./build/serveur_vX
#X -> le numéro de la version
./build/client_vX *adresse_du_serveur* *port_du_serveur*
#X -> le numéro de la version
```

Pour une meilleure qualité, le lien suivant permet de visualiser nos diagrammes sur le site [draw.io](https://draw.io) :

[DiagrammesSAE.drawio - draw.io](https://draw.io)

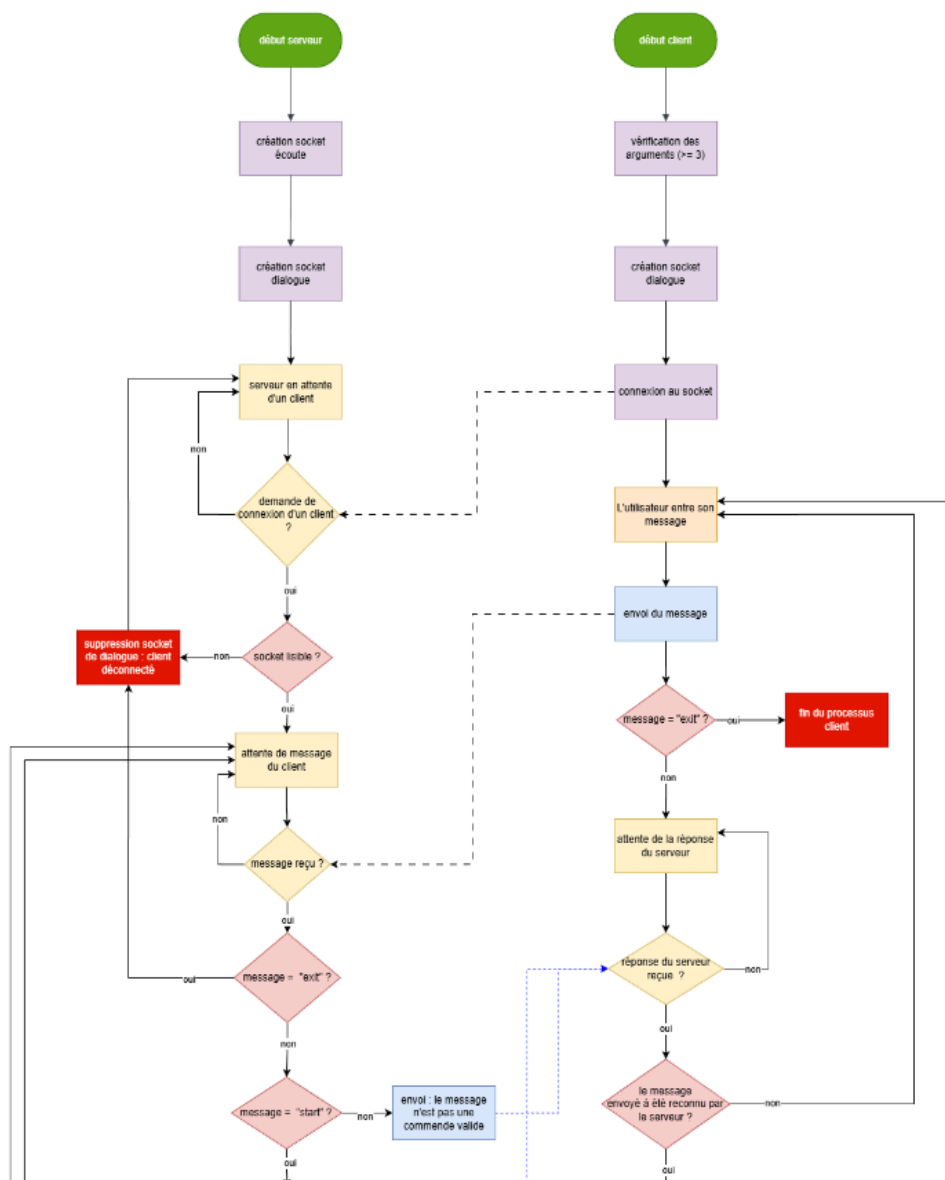
# Version 0 : Jeu solo

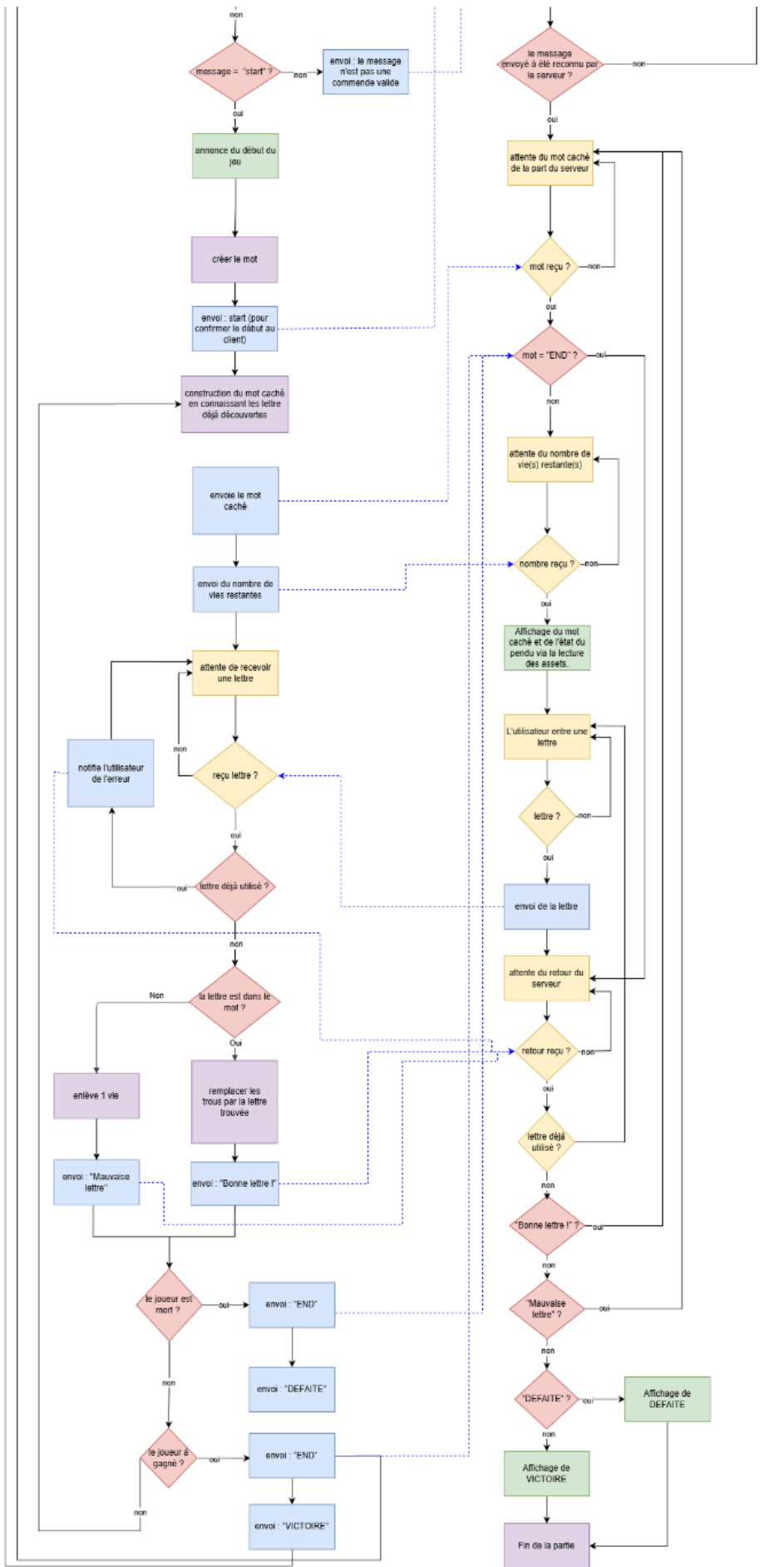
Cette version correspond à un jeu du pendu classique à **un seul joueur**.

## Fonctionnement

- Le serveur attend la connexion d'un client.
- Une fois connecté, le joueur accède à un écran d'accueil en mode terminal.
- Deux commandes sont disponibles :
  - **start** : lancer une partie
  - **exit** : quitter le serveur
- Le serveur choisit automatiquement un mot.
- Le joueur propose des lettres jusqu'à la fin de la partie (victoire ou défaite).
- Une fois la partie terminée, le joueur revient à l'écran d'accueil.

Ci-dessous, le diagramme de cette version :





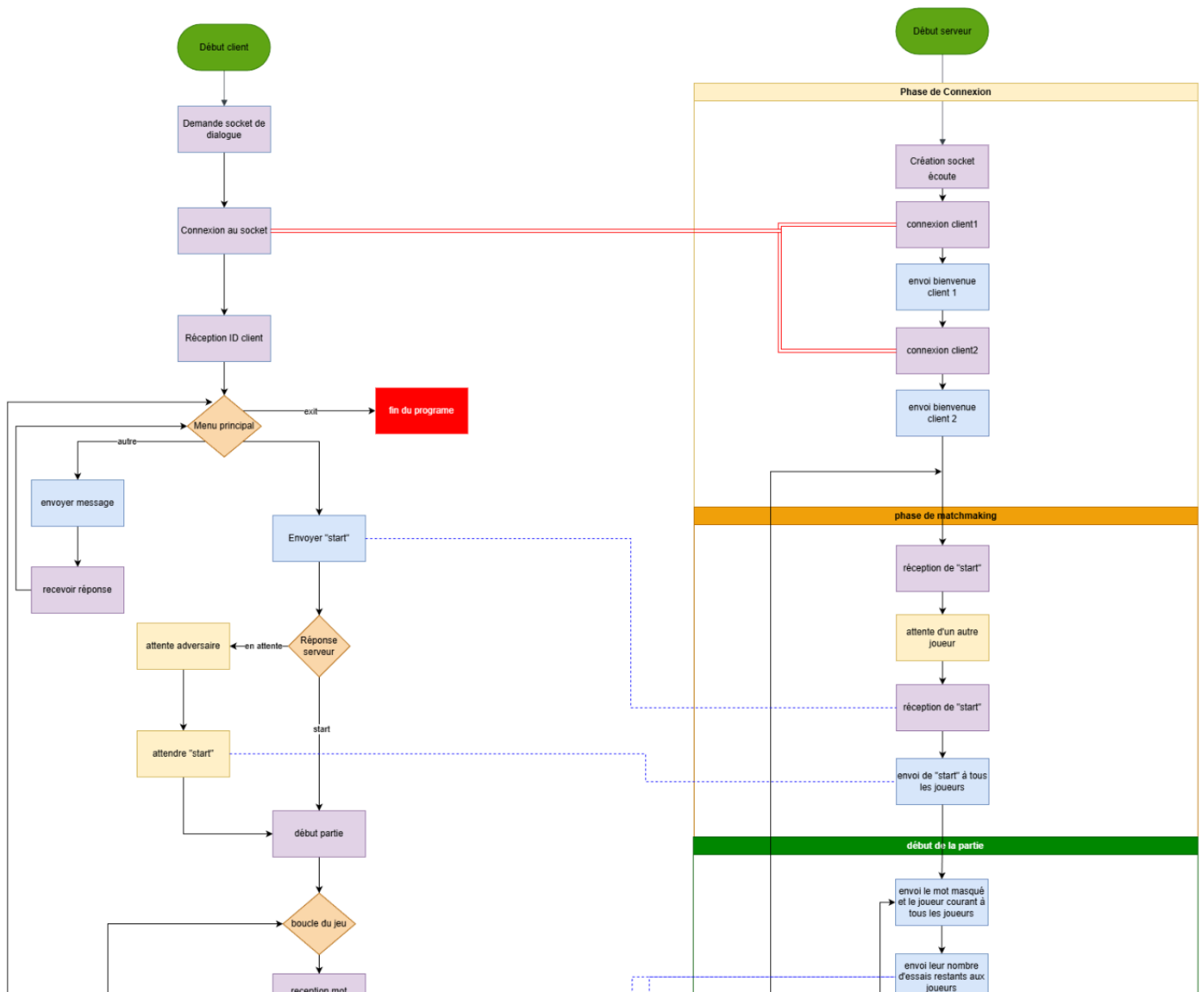
# Version 1 : Jeu à deux joueurs (compétitif)

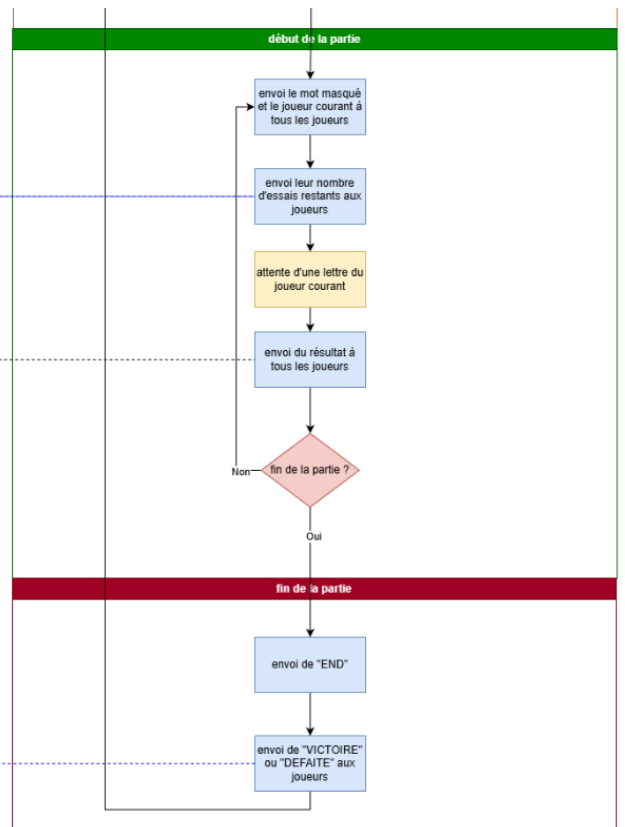
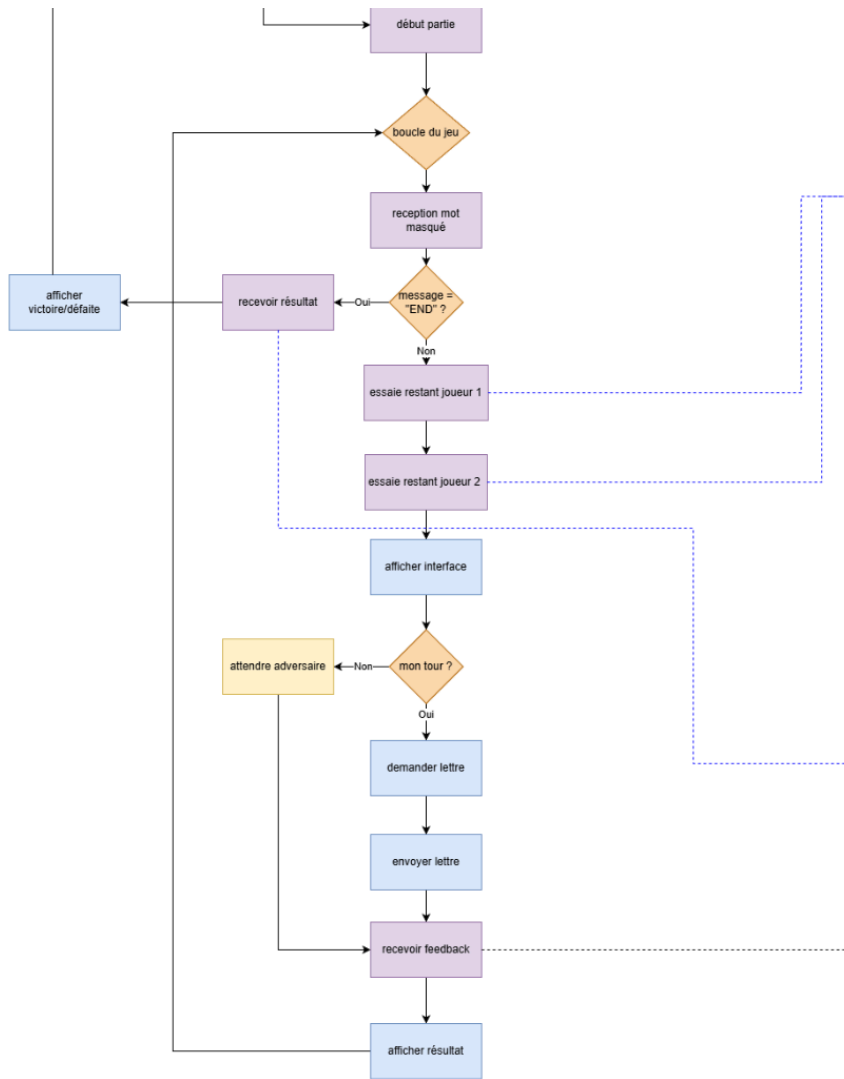
Cette version permet à **deux joueurs** de jouer simultanément en compétition.

## Fonctionnement

- Les deux clients se connectent au serveur via un terminal.
- Chaque joueur entre la commande **start**.
- Lorsque le serveur reçoit les deux commandes **start**, la partie commence.
- Les deux joueurs cherchent le **même mot** et voient les lettres déjà trouvées.
- Chaque joueur possède son propre pendu.
- Si un joueur gagne, l'autre perd.
- Le fonctionnement général reste similaire à la version 0.

Ci-dessous, le diagramme de cette version :





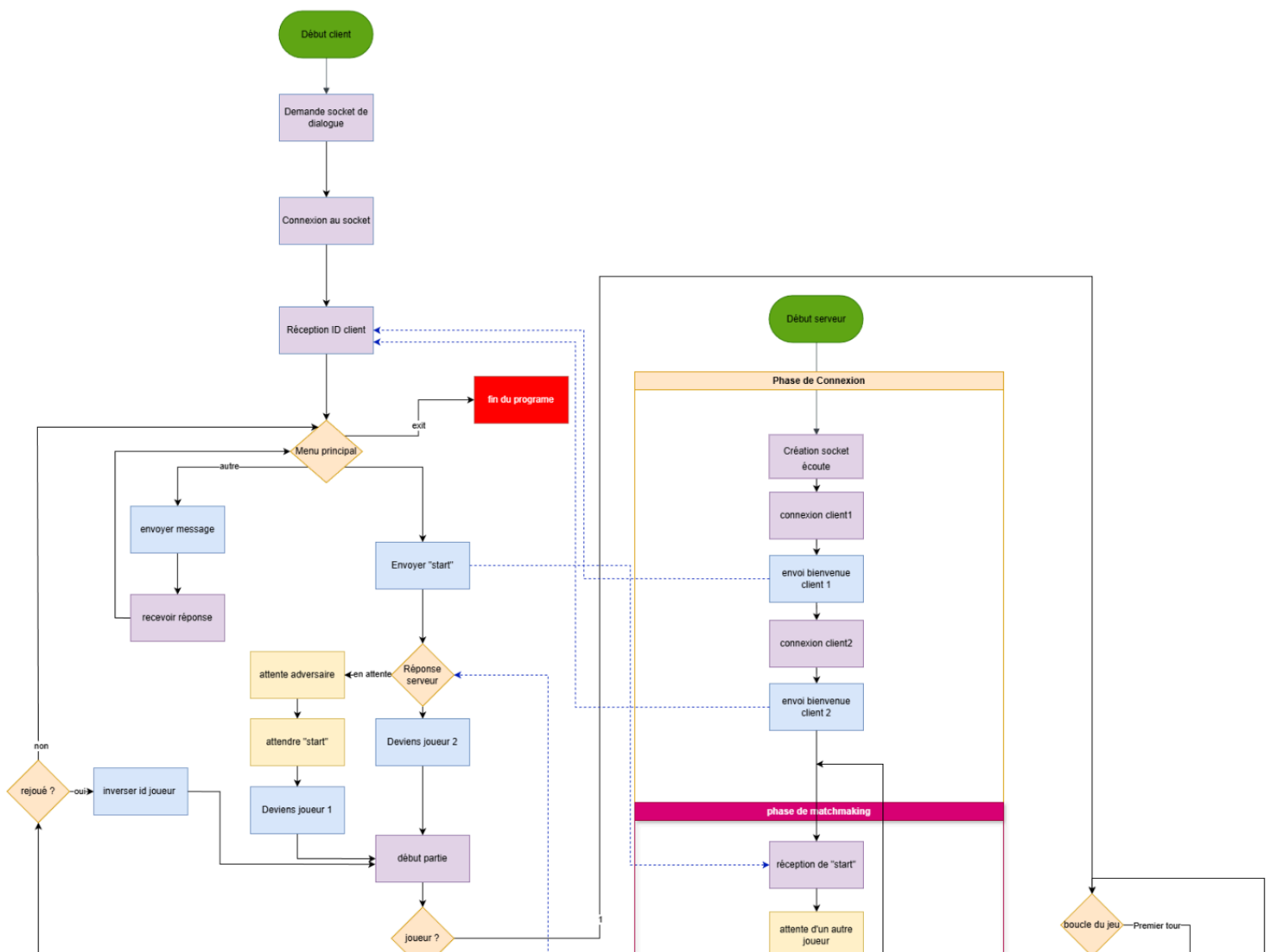
## Version 2 : Jeu à deux joueurs (coopératif)

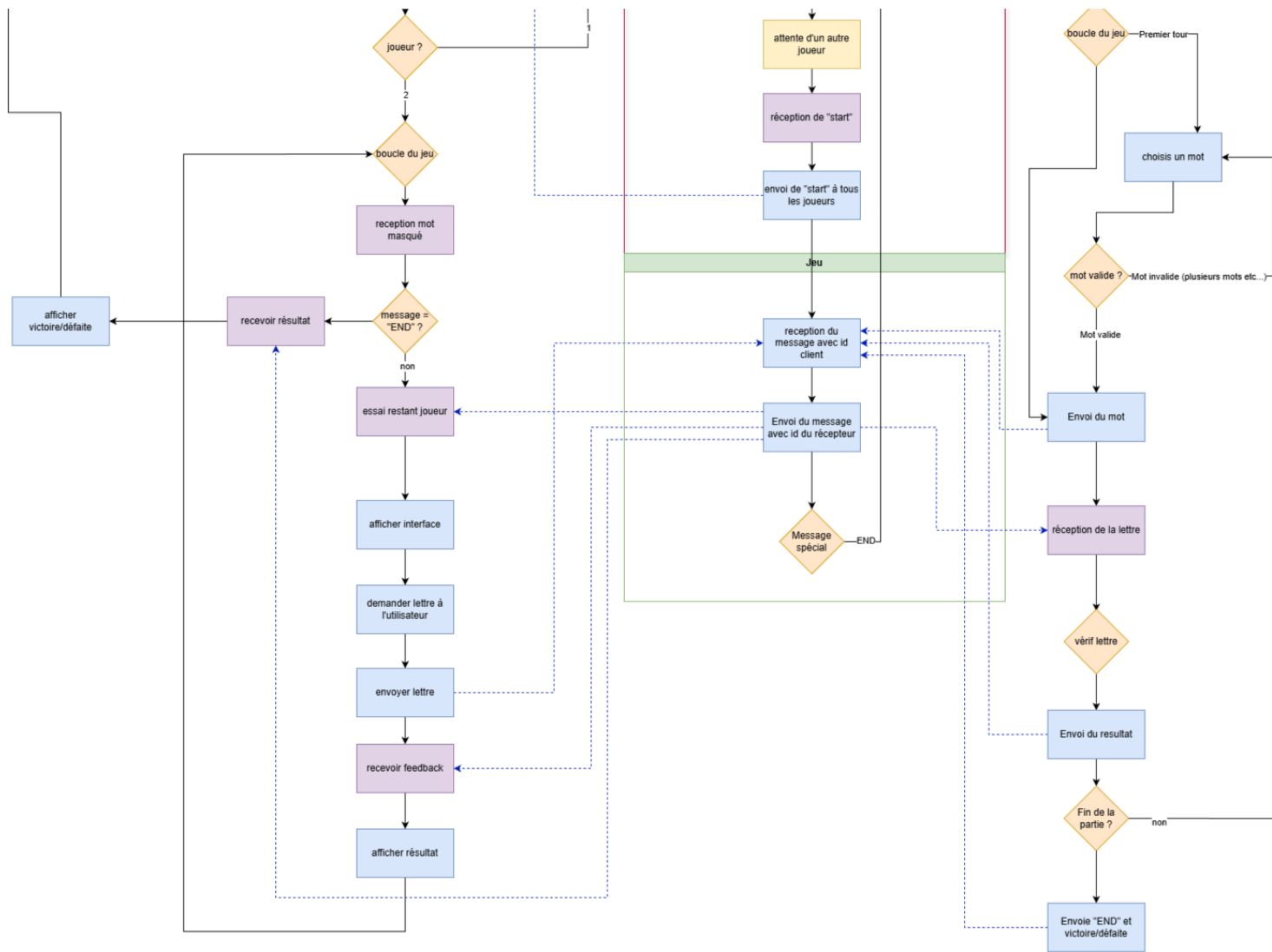
Cette version est également un jeu à **deux joueurs**, mais sans compétition.

## Fonctionnement

- Les deux joueurs se connectent au serveur.
- Chaque joueur entre la commande `start`.
- Le premier joueur connecté choisit le mot à faire deviner.
- Le second joueur doit deviner le mot.
- Une fois la partie terminée, les joueurs peuvent :
  - relancer une partie
  - quitter le serveur

Ci-dessous, le diagramme de cette version :







# Version 3 : Gestion de plusieurs parties simultanées

Cette version reprend le fonctionnement de la version 2, mais permet au serveur de gérer **plusieurs parties en parallèle**.

## Fonctionnement

- Les joueurs se connectent et entrent la commande **start**.
- Le premier joueur choisit le mot, le second le devine.
- Chaque partie est gérée par un **processus fils** du serveur principal.
- Plusieurs paires de joueurs peuvent jouer en même temps, sans interférence.
- De nouveaux clients peuvent se connecter à tout moment.

