# Sherlock 13 – Projet Réseau & SDL2

Rapport de projet – Module OS USERS

Auteur : **TOURE SEKOUBA**

## 1. Objectif

— Adapter le jeu *Sherlock 13* en application multijoueur réseau (TCP).
— Fournir une interface graphique SDL2 indépendante pour chaque joueur.

## 2. Organisation du dépôt

— `server.c` – serveur TCP (logique du jeu).
— `sh13.c` – client SDL2 (rendu + communication).
— `Makefile` – compilation automatique.
— Ressources – images PNG, police TTF.
— `README.md` – guide utilisateur.
— `rapport_SH13_TOURE_SEKOUBA.pdf` – présent document.

## 3. Compilation et lancement

— **Prérequis** : `build-essential`, `pkg-config`, `libsdl2-dev`, `libsdl2-image-dev`, `libsdl2-ttf-dev`.
— Compilation : `$ make`
— Serveur : `$ ./server 8080`
— Client : `$ ./sh13 <IPserv> 8080 <IPcli> <PortCli> <Nom>`

## 4. Architecture logicielle

— Processus serveur unique + quatre processus clients.
— Chaque client possède deux threads :
  — Principal : boucle SDL (rendu, *input*).
  — Réseau : socket TCP locale, réception non bloquante.
— Synchronisation via `mutex` + variable volatile `synchro`.

## 5. Notions mises en œuvre

— Sockets TCP (`bind`, `listen`, `accept`, `connect`, `read`/`write`)
— Processus multiples
— Threads POSIX côté client
— Mutex pour section critique
— SDL2 pour le graphisme

## 6. Protocole d'échange

— `C ip port name` – connexion
— `I id` – id attribué
— `L n1 n2 n3 n4` – liste joueurs
— `D c1 c2 c3` – distribution cartes
— `V i j val` – valeur (ou 100 pour )
— `M id` – joueur courant

— `O id obj` – question globale
— `S id j obj` – question ciblée
— `G id suspect` – accusation

## 7. Fonctionnement du jeu

— Au lancement, les 4 joueurs se connectent (bouton *CONNECT*).
— Le serveur distribue et annonce le joueur 0 comme premier.
— Le bouton *GO* apparaît uniquement pour le joueur actif.
— Actions : question globale, question ciblée, accusation.
— La partie s'arrête dès qu'une accusation est correcte.

## 8. Difficultés rencontrées

— Tester seul : 4 instances SDL, gestion des ports.
— Maintenir la réactivité SDL tout en écoutant le réseau.
— Protocole minimaliste mais suffisant pour toutes les actions de jeu.

## 9. Perspectives d'amélioration

— Envoyer le nom du vainqueur aux clients (message `W`) et afficher un pop-up de fin de partie.
— Ajouter la règle « dernier joueur sans accusation erronée gagne ».
— Implémenter une IA simple pour partie solo.
— Créer un lobby réseau et un chat texte.
— Packaging (`.deb`) et script d'installation.