

# Rapport Projet OS User

## SHERLOCK 13

**RÈGLES À 2 JOUEURS**  
Les règles sont les mêmes que pour 3 et 4 joueurs à quelques exceptions près :  
1. En préparation, on distribue 5 cartes à chaque joueur et on place 3 cartes face cachée (au lieu d'une seule) en ligne au centre de la table. La carte du milieu est la carte Criminel.  
2. ① L'Enquête 1 est différente : Choisissez l'une des deux cartes adjacentes au criminel et échangez-la avec une carte de votre main. Placez la carte qui vient de votre main FACE VISIBLE.  
3. Lorsque les deux cartes adjacentes au criminel ont été révélées (par n'importe qui), on ne peut plus faire l'enquête 1.

**VARIANTE EXPERT**  
Avec ce mode de jeu, l'enquête sera plus compliquée.  
1. Lorsque vous recevez vos cartes en début de partie, classez-les de la première à la dernière. Lorsque vous répondez aux questions ne prenez pas en compte les symboles de la dernière carte.  
**Note: Ne confondez pas avec la feuille d'enquête. Seules les cartes EN MAIN comptent !**  
Exemple : Lorsqu'on vous demande de lever la main pour le symbole « Ampoule », si votre dernière carte est la seule à avoir une « Ampoule », ne levez pas la main. Si vous avez 3 cartes avec un « Crâne » en main dont une en dernière et que l'on vous demande combien de « Crâne » vous avez, répondez que vous en avez 2.  
2. Au début de son tour, chaque joueur doit placer la première de ses cartes en dernière position. Les joueurs doivent faire ça à chaque tour même lorsque leurs tours sont sautés après qu'ils aient raté une accusation.  
\* À 2 joueurs, lorsque l'on fait l'enquête 1, placez la carte prise dans la position de la carte échangée.

**SHERLOCK 13**  
Sherlock Holmes, John Watson, James Moriarty, Irène Adler...  
De célèbres personnages sont devenus les suspects d'une sombre affaire.  
Le coupable est forcément l'un d'entre eux ! Qui sera capable de le démasquer en premier ?

13 Cartes personnage • 4 Paravents • 1 Carnet de feuilles d'enquête  
Chaque joueur a besoin d'un crayon.

**CRÉDITS**  
Auteur: Hope S. Hwang  
Illustrateur & Graphiste: Vincent Dutrait  
Adaptation PAO : Kiwi World  
www.letheia-games.com  
Droits réservés pour les éditions française, néerlandaise, italienne  
© 2017 Letheia • Tous droits réservés  
© 2016 BoardM Factory

**Groupe A**  
**JIANG Zimeng**

# Table des matières :

- [I. Description du jeu](#)
- [II. Réalisation du projet - Implémentation technique](#)
  - [1. Communication TCP](#)
    - [A. Modèle Client-Serveur](#)
    - [B. Communication via Sockets TCP](#)
      - [Appels Système Clés \(Serveur\)](#)
      - [Appels Système Clés \(Client\)](#)
    - [C. Protocole de Communication](#)
  - [2. Gestion des États du Serveur \(Machine à États Finis - FSM\)](#)
  - [3. Gestion des Conflits](#)
    - [Contexte Multithread](#)
    - [Solution](#)
  - [4. Affichage Graphique \(SDL\)](#)
  - [5. Gameplay - Comment Jouer ?](#)

## I. Description du jeu

**Sherlock 13** est un jeu de déduction dans lequel les joueurs incarnent des enquêteurs cherchant à identifier le criminel parmi 13 suspects célèbres issus de l'univers de Sherlock Holmes. Chaque suspect est caractérisé par une combinaison de **symboles** (par exemple : loupe, chapeau, pipe, crâne...). Le but est d'utiliser la logique et les indices pour démasquer le criminel avant les autres.

*Règles pour 4 joueurs :*

- a. Préparation du jeu :
  - On mélange les 13 cartes Personnage en gardant la dernière en tant que criminel.
  - Les 12 autres cartes sont distribuées entre les 4 joueurs, soit **3 cartes chacun**.
  - Chaque joueur garde secrètes ses cartes et utilise **une feuille d'enquête** pour noter les symboles observés et les indices récoltés.
- b. Déroulement d'un tour :

*Chaque joueur, à son tour, choisit une action :*

- Enquête 1 (**O**): Tous lèvent la main s'ils ont un symbole demandé.
- Enquête 2 (**S**): Un joueur ciblé révèle combien il a d'un symbole précis.
- Accusation (**G**): Nommer un personnage.
  - Correct : Victoire.
  - Faux : Élimination.

- c. Fin de partie :

Un joueur accuse juste **OU** un seul joueur reste.

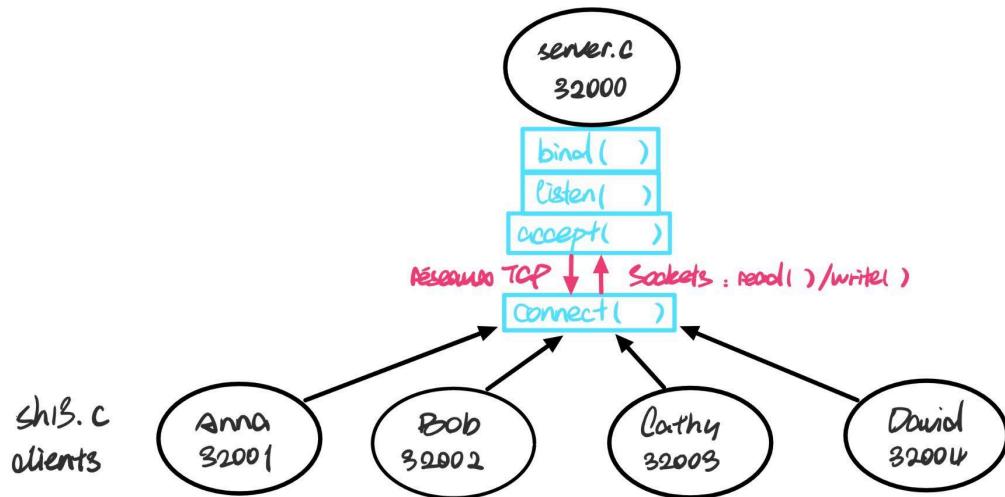
## II. Réalisation du projet - Implémentation technique

### 1. Communication TCP

#### A. Modèle Client-Serveur

**Serveur** : Centralise la logique du jeu (distribution des cartes, gestion des tours, vérification des accusations).

**Clients** : Représentent les joueurs (affichage graphique, envoi/réception des actions).



## B. Communication via Sockets TCP

Les échanges reposent sur des sockets TCP pour garantir une connexion fiable et ordonnée.

Appels Système Clés (Serveur)

Fonction	Rôle	Exemple dans <code>server.c</code>
<code>socket()</code>	Crée un socket TCP/IPv4	<code>sockfd = socket(AF_INET, SOCK_STREAM, 0);</code>
<code>bind()</code>	Lie le socket à un port/IP	<code>bind(sockfd, (struct sockaddr*)&amp;serv_addr, ...)</code>
<code>listen()</code>	Met le socket en écoute pour des connexions entrantes	<code>listen(sockfd, 5);</code>
<code>accept()</code>	Accepte une connexion client et crée un nouveau socket dédié	<code>newsockfd = accept(sockfd, ...);</code>
<code>read()</code> <code>/write()</code>	Lit/envoie des messages via le socket	<code>n = read(newsockfd, buffer, 255);</code>

### Appels Système Clés (Client)

Fonction	Rôle	Exemple dans <code>sh13.c</code>
<code>connect()</code>	Établit une connexion au serveur	<code>connect(sockfd, (struct sockaddr*)&amp;serv_addr, ...)</code>
<code>pthread_create()</code>	Lance un thread pour gérer la réception asynchrone des messages	<code>pthread_create(&amp;thread_serveur_tcp_id, ...);</code>

## C. Protocole de Communication

*Format des messages :*

- *Côté joueur*
- 'C' : Inscription et connexion au serveur ('C <IP> <port> <nom>')
- 'O' : Enquête « objet » – demande à tous les autres joueurs s'ils possèdent un symbole spécifique ('O <id\_joueur> <id\_objet>')
- 'S' : Enquête « standard » – demande à un joueur ciblé le nombre exact d'un symbole ('S <id\_joueur> <id\_ciblé> <id\_objet>')
- 'G' : Accusation – accusation contre un suspect ('G <id\_joueur> <id\_suspect>')

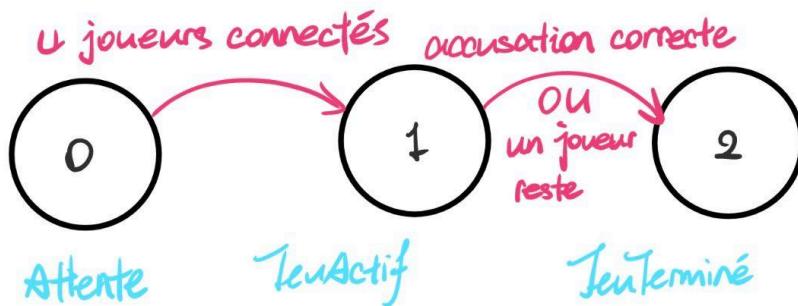
- *Côté serveur*
- 'I' : Attribution d'un identifiant unique au joueur ('I <id>')
- 'L' : Diffusion de la liste des joueurs connectés ('L <nom1> <nom2> <nom3> <nom4>')
- 'D' : Distribution des cartes aux joueurs ('D <cartes\_joueur>')
- 'M' : Notification du joueur courant ('M <id\_joueur>')
- 'V' : Réponse à une enquête (valeur d'un symbole) ('V <id\_joueur> <valeur>')
- 'F' : Échec d'une accusation ('F <id\_joueur>')
- 'W' : Victoire d'un joueur ('W <id\_gagnant>')

## 2. Gestion des États du Serveur (Machine à États Finis - FSM)

*Le serveur utilise une machine à états finis (FSM) pour gérer les phases du jeu. La variable `fsmServer` contrôle l'état courant, avec trois états principaux :*

État (fsmServer)	Nom	Description
0	Attente	Le serveur attend la connexion des 4 joueurs.
1	Jeu Actif	Tous les joueurs sont connectés, la partie est en cours.
2	Jeu Terminé	Un joueur a gagné ou tous sauf un ont échoué. Le serveur ne traite plus de nouvelles actions.

fsmServer: Machine à États Finis



### 3. Gestion des Conflits

#### Contexte Multithread

Un **thread** est une séquence d'instructions qui peut être exécutée **indépendamment** par le processeur. Chaque thread partage la même mémoire que les autres threads du même processus, mais il possède son propre contexte d'exécution (comme son propre compteur de programme et sa pile d'appels).

Dans notre architecture client-serveur avec 4 joueurs connectés simultanément, le serveur et les clients utilisent des threads pour gérer les interactions **en temps réel** :

- Serveur : Un thread principal accepte les connexions, et un thread par client traite ses actions.
- Client : Un thread gère l'interface graphique (SDL), un autre écoute les messages réseau.

Dans un environnement **multithread** ou réseau, plusieurs threads/processus peuvent accéder simultanément à des ressources partagées (ex: liste des joueurs, état du jeu). Cela peut entraîner des **race conditions** (incohérences de données).

## Solution

**Mutex** (Mutual Exclusion) : Verrous pour protéger l'accès aux variables critiques. (ex: pthread\_mutex\_t pour synchro)

**Synchronisation** : Mécanismes pour coordonner les threads :

- Les messages M (tour courant) et W (victoire) sont envoyés via broadcast pour garantir que tous les clients ont un état cohérent.
- Le serveur attend un accusé de réception implicite (pas de réponses explicites des clients).

## 4. Affichage Graphique (SDL)

*Structure du Code Client (sh13.c) :*

1. **Initialisation** SDL :

- Fenêtre et rendu
- Chargement des ressources :
  - Textures : Cartes, boutons, symboles  
(ex: IMG\_Load("assets/images/S13\_0.png")).
  - Polices : Utilisation de TTF\_OpenFont pour le texte (ex: sans.ttf).

2. **Boucle** Principale (while (!quit)) :

- a). Gestion des événements: les clics souris, les touches, et la fermeture de la fenêtre.
- b). Traitement des messages réseau (**synchro == 1**).
- c). Mise à jour de l'interface: affichage dynamique des cartes, symboles, et messages via `SDL_RenderCopy`.

3. **Nettoyage** : Libération des ressources SDL.

## 5. Gameplay - Comment Jouer ?

1) **Lancement** du serveur et des clients (terminaux séparés) (voir [README.md](#))

2) **Connexion** - Cliquer sur le bouton "**Connect**" (en haut à gauche).

*Le serveur attend 4 connexions: une fois complet, les cartes sont distribuées automatiquement et la ligne du joueur dans **tableCartes** est remplie (symboles visibles).*

3) **Déroulement** d'un Tour

*Quand c'est votre tour : Le bouton "**Go**" apparaît (en bas à droite).*

a). Actions Possibles :

- Enquête 1 (**O**): Sélectionner un symbole.
- Enquête 2 (**S**): Sélectionner un symbole + un joueur (colonne de gauche).
- Accusation (**G**): Sélectionner un personnage (liste de droite).

b). Cliquer sur "Go".

	U	L	H	S	D	O	E	K
	5	5	5	5	4	3	3	3
Anna	1	1	1	1	1	1	1	0
Bob								
Cathy								
David								

💀💡	Sebastian Moran	
💀💡💡	Irene Adler	
💡💡💡💡	inspector Lestrade	
💡💡💡💡	inspector Gregson	
💡💡💡💡	inspector Baynes	
💡💡💡💡	inspector Bradstreet	
💡💡💡💡	inspector Hopkins	
💡💡💡💡	Sherlock Holmes	
💡💡💡💡	John Watson	
💡💡💡💡	Mycroft Holmes	
💡💡💡💡	Mrs. Hudson	
💡💡💡💡	Mary Morstan	
💀💡💡	James Moriarty	

#### 4) Résultat

- Accusation correcte - Écran de victoire s'affiche ("VICTOIRE !").
- Accusation incorrecte - Élimination (plus de tours).

	U	L	H	S	D	O	E	K
	5	5	5	5	4	3	3	3
Anna								
Bob								
Cathy	2	2	2	1	2	0	0	0
David								

💀💡	Sebastian Moran	
💀💡💡	Irene Adler	
💡💡💡💡	inspector Lestrade	
💡💡💡💡	inspector Gregson	
💡💡💡💡	inspector Baynes	
💡💡💡💡	Inspector Bradstreet	
💡💡💡💡	inspector Hopkins	
💡💡💡💡	Sherlock Holmes	
💡💡💡💡	John Watson	
💡💡💡💡	Mycroft Holmes	
💡💡💡💡	Mrs. Hudson	
💡💡💡💡	Mary Morstan	
💀💡💡	James Moriarty	

VICTOIRE ! Vous avez gagné !