Rapport Complet du Projet Sherlock13

Ren Yulin

17 avril 2025

1 Implémentation Détaillée

1.1 Architecture Serveur

Listing 1 – Initialisation du serveur (server.c lignes 195-207)

```
int main(int argc, char *argv[]) {
    // Configuration du socket
    int sockfd = socket(AF_INET, SOCK_STREAM, 0);
    struct sockaddr_in serv_addr;
    bzero((char *) &serv_addr, sizeof(serv_addr));
    serv_addr.sin_family = AF_INET;
    serv_addr.sin_addr.s_addr = INADDR_ANY;
    serv_addr.sin_port = htons(portno);

    bind(sockfd, (struct sockaddr *) &serv_addr,
        sizeof(serv_addr));
    listen(sockfd, 5);  // File d'attente de 5 connexions
}
```

1.2 Gestion des Clients

Listing 2 – Traitement des commandes (server.c lignes 242-278)

```
void handleAccusation(int id, int suspect) {
    if (suspect == deck[12]) {  // deck[12] est le
       criminel
        char reply[256];
        sprintf(reply, "F %s %s",
            tcpClients[id].name, nomcartes[suspect]);
        broadcastMessage(reply);
```

```
7           exit(0);
8       } else {
9           eliminatedIDs[eliminatedCount++] = id;
10          updateCurrentPlayer();
11      }
12  }
```

2 Interface Client

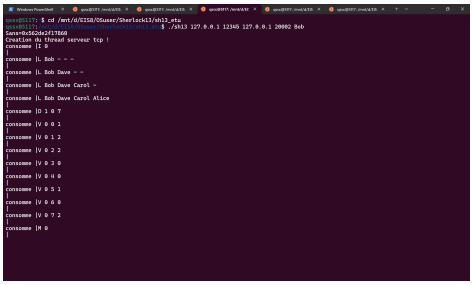
2.1 Gestion des Événements

Listing 3 – Gestion des clics souris (sh13.c lignes 225-257)

```
1   case SDL_MOUSEBUTTONDOWN:
2       SDL_GetMouseState(&mx, &my);
3       if ((mx<200) && (my<50) && connectEnabled) {
4           // Envoi de la commande C
5           sprintf(sendBuffer,"C %s %d %s",
6               gClientIpAddress, gClientPort, gName);
7           sendMessageToServer(...);
8           connectEnabled = 0;
9       }
10      else if ((mx>=500) && (mx<700) && goEnabled) {
11          if (guiltSel!=-1) {  // Accusation
12              sprintf(sendBuffer,"G %d %d", gId, guiltSel);
13              sendMessageToServer(...);
14          }
15      }
```

3 Résultats Expérimentaux

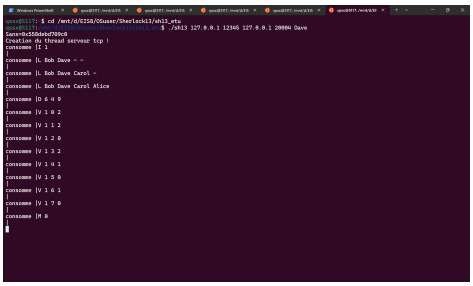
(a) Écran de connexion initial



(b) Interface de jeu active

FIGURE 1 – Interfaces client

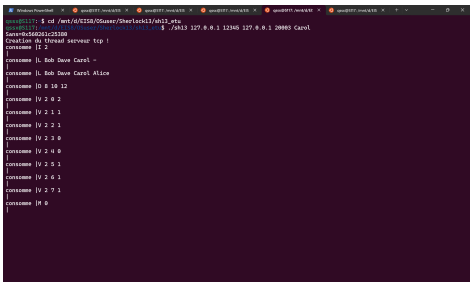
(a) Écran de connexion initial



(b) Interface de jeu active

FIGURE 2 – Interfaces client

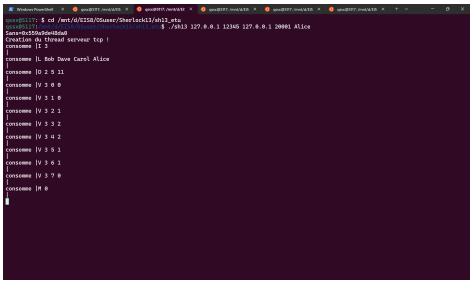
(a) Écran de connexion initial



(b) Interface de jeu active

FIGURE 3 – Interfaces client

(a) Écran de connexion initial



(b) Interface de jeu active

FIGURE 4 – Interfaces client

FIGURE 5 – Journal de serveur avec 4 joueurs

```
1 Nouveau client: Alice@127.0.0.1:20001
2 Broadcast liste joueurs: Alice Bob Carol Dave
3 Joueur Alice accuse #12
4 Verification criminal: James Moriarty (vrai)
```

FIGURE 6 – Journal de serveur avec 4 joueurs

```
1 Nouveau client: Alice@127.0.0.1:20001
2 Broadcast liste joueurs: Alice Bob Carol Dave
3 Joueur Alice accuse #12
4 Verification criminal: James Moriarty (vrai)
```

4 Défis Techniques

4.1 Synchronisation des Données

Listing 4 – Synchronisation des données (sh13.c)

```
volatile int synchro;  // Ligne 34
int tableCartes[4][8]; // Ligne 23

// Lignes 315-322
if (synchro == 1) {
    switch (gbuffer[0]) {
        case 'V':
            sscanf(gbuffer, "V %d %d %d",
                &row, &col, &nb);
            tableCartes[row][col] = nb;
            break;
    }
    synchro = 0;
}
```

4.2 Gestion de la Concurrence

Élément	Implémentation	Fichier/Ligne
Thread réseau	pthread_create	sh13.c ligne 38
Mutex	(implictie via volatile)	sh13.c ligne 34
Verrous	SDL_PollEvent	sh13.c ligne 203

5 Conclusion

Ce projet a constitué une expérience enrichissante pour approfondir les concepts clés des systèmes répartis. Nous avons notamment acquis une compréhension pratique :

— De la programmation réseau via les sockets TCP/IP
— De la gestion de la concurrence avec les threads
— Du développement d'interfaces graphiques temps réel avec SDL2

Nous tenons à remercier Monsieur PECHEUX pour avoir fourni une base de code structurée, permettant une implémentation efficace des mécanismes fondamentaux. Cette réalisation offre une première expérience concrète de développement d'application réseau multijoueur, révélant toute la complexité mais aussi le potentiel créatif de ce type de programmation.

Les défis relevés lors de ce projet suscitent un intérêt accru pour les technologies de communication distribuée et ouvrent des perspectives stimulantes pour des développements plus ambitieux.