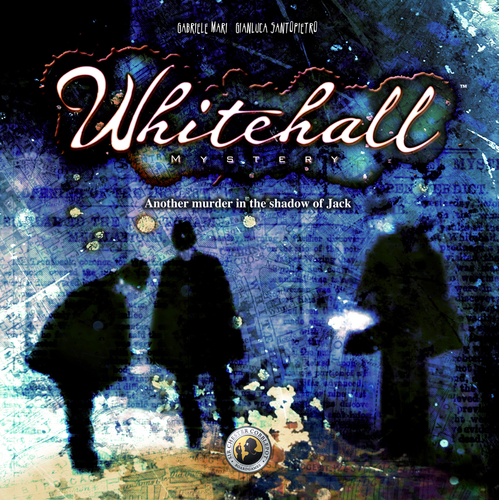
Whitehall



Professeur : Mr François Pecheux

Table des matières

[I) Introduction 2](#_Toc503610072)

[II) Le jeu 3](#_Toc503610073)

[Le but de la partie 3](#_Toc503610074)

[Etat 1 : La connexion 3](#_Toc503610075)

[Etat 2 : L’initialisation 3](#_Toc503610076)

[Etat 3 : Le jeu 4](#_Toc503610077)

[Etat 4 : La fin du jeu 4](#_Toc503610078)

[III) Diagramme de communication client-serveur 5](#_Toc503610079)

[IV) Pour aller plus loin 6](#_Toc503610080)

[V) Conclusion 6](#_Toc503610081)

# Introduction

Ce projet a pour but de démontrer nos connaissances des applications client-serveur et des threads. Nous avions le choix entre 3 jeux mais nous n’étions pas limités et pouvions entreprendre une idée personnelle.  
Dans ce rapport nous parlerons des règles du Jeu, des fonctionnalités que nous avons implémentées, et nous parlerons des échanges client-serveur dans un diagramme. Enfin, nous aborderons les points que nous aurions aimé améliorer ou ajouter au jeu pour le rendre plus drôle à jouer.  
Ce projet qui au départ avait été démarré individuellement en cours a été ensuite réalisé en binôme par Louis Buchert et Marco Longépé après avoir participé avec le professeur et la classe à la création de certains fichiers de positionnement sur la carte du jeu.

# Le jeu

## Le but de la partie

Jack : C’est le méchant, il est un peu dérangé et tue des gens. Le problème c’est que c’est interdit, et la police est à ses trousses. Son but est donc de faire les 3 victimes restantes qu’il a planifiées dans son plan machiavélique en moins de 15 coups pour chacune prise séparément, pour ne pas louper l’heure du thé.

Les policiers : Les piliers de la justice, de vrais héros. Londoniens malgré eux, ils vont devoir s’acquitter de leur travail s’ils veulent être payé. Ils se lancent donc aux trousses de Jack et doivent le coincer pour gagner.

Le jeu peut se diviser en 4 étapes principales, décrivant l’état, l’avancée de la partie :

## Etat 1 : La connexion

La première étape consiste à s’assurer que la connexion des différents clients au serveur se déroule bien.  
Le script exe.sh permet de tester le jeu car il lance le serveur de jeu et 4 fenêtres : En effet ce jeu se joue à 4 joueurs, à savoir un impitoyable Jack qui sera attribué au premier client initiant la connexion, et 3 vaillants policiers.  
Le jeu ne démarre pas tant que tous les clients ne sont pas connectés, c’est-à-dire que la variable fsm\_server décrivant l’état du jeu reste à 0, étape de connexion. Pour ce faire, chaque joueur doit cliquer sur le bouton « connexion », fsm\_server passe alors à 1.

## Etat 2 : L’initialisation

La seconde étape sera d’initialiser la partie. La variable fsm\_server passe donc à 1, et y restera tant que l’étape d’initialisation ne sera pas validée par le serveur. Bien entendu toutes les vérifications qui vont suivre peuvent se faire dans n’importe quel ordre, c’est-à-dire que les clients peuvent tous placer leurs pions en même temps. Attention cependant à la latence du jeu, il faudra laisser la souris fixe après un clic afin de laisser le serveur procéder à la validation de la position.

Au début de la partie, Jack choisit secrètement 4 cercles (rouges) sur le plateau de jeu en cliquant dessus, sachant que chacun doit appartenir à un quart différent du quadrillage, bien que cette restreinte ne soit pas vérifiée par le serveur. A chaque décision prise, un cercle rouge apparait comme par magie, ces quatre cercles sont les Lieux de Découverte, là où il assouvira sa soif de sang.  
Une fois la mise en place terminée, Jack choisit secrètement son point de départ, par un cinquième clic : il doit s’agir d’un des quatre cercles numérotés qu’il a notés en haut de sa feuille comme Lieux de Découverte.   
Un marqueur Lieu de Découverte rouge est alors visible par les enquêteurs, afin qu’ils sachent où débute la fuite de Jack.

La dernière vérification pour le serveur sera de vérifier que chaque policier a choisi sa position de départ, car ils sont obligés de commencer sur un carré entouré de jaune, vers le centre de la carte.

Durant toutes ces étapes, si un clic de placement n’est pas validé par le serveur, il ne se passe rien et le serveur attend un clic valide de la part du client.

## Etat 3 : Le jeu

La troisième étape permet d’attribuer les tours de jeu aux clients et de contrôler leurs coups.

Jack se déplace du dernier cercle numéroté noté sur sa feuille vers un autre cercle numéroté adjacent, en suivant les lignes en pointillés noires.   
Si le numéro du cercle vers lequel il se déplace correspond à celui d’un nouveau Lieu de Découverte, Jack l’annonce à la fin du tour pour commencer un nouveau round. Mais s’il réalise ses quinze déplacements sans avoir atteint un nouveau Lieu de Découverte, il a perdu la partie.

Quand Jack déclare avoir atteint une nouvelle scène de crime, le round en cours s’achève et un nouveau débute ; le suivi de déplacement reprend alors à 0. Rien n’oblige Jack à rallier les Lieux de Découverte dans un ordre précis : l’ordre dans lequel les quatre numéros apparaissent en haut de sa feuille n’a aucune influence sur le déroulement de la partie et ne doit pas limiter sa stratégie.

Après s’être concertés, les Enquêteurs peuvent déplacer leurs figurines au clic gauche.  
Les figurines Enquêteur peuvent se déplacer jusqu’à 2 Croisements, en suivant les lignes en pointillés (elles ignorent les cercles numérotés quand elles se déplacent).

Dans l’ordre de priorité, chaque Enquêteur peut décider de questionner une case adjacente, en cliquant droit : ou Jack est sur cette case (point rouge apparait), ou il s’agit d’un point de Découverte (point rouge apparaît également), ou Jack est passé par là (point jaune apparaît). Ces types d’actions peuvent être entrepris dans les cercles numérotés adjacents au Croisement occupé par la figurine Enquêteur (c’est-à-dire les cercles numérotés directement reliés au Croisement par une ligne en pointillés, sans autre Croisement ou cercle numéroté entre eux.).

Chaque Enquêteur ne peut entreprendre qu’une seule action (questionner une case au clic gauche ou avancer au clic droit). Un joueur qui contrôle plusieurs figurines Enquêteur peut cependant choisir une action différente pour chacune d’elles.

Ainsi, si Jack n’atteint pas en moins de 15 coups une zone de découverte, si toutes les zones de découvertes ont été trouvées avant lui par les policiers, ou s’il a été trouvé par les policiers, il perd. Au contraire, s’il trouve toutes ses zones de crime, il gagne.

## Etat 4 : La fin du jeu

Finalement, la dernière étape permet de clore la partie. C’est l’état de fin, fsm\_server vaut 3, le serveur de prend plus en considération les requêtes des clients.

# Diagramme de communication client-serveur

Nous allons voir un exemple de partie "type" pour comprendre les différents échanges client-serveur qui surviennent lors de la partie :

server-white.c  
APP Server  
TCP

Etat

t7.c  
APP Client/Server  
SDL2, TCP  
Policiers (3)

t7.c  
APP Client/Server  
SDL2, TCP  
JACK

Clic Connexion SDL, @IPClient1, N°Port1, Jack

I id

Clic Connexion SDL, @IPClient2-3-4, N°Port2-3-4, Sherlock

I id

3 premiers clics gauche SDL: zone meurtre 1-2-3

1 cercle rouge SDL après chaque clic de Jack

1 clic gauche / policier : position

1 policier placé SDL / client si emplacement valide

Dernier clic gauche : Départ de Jack

Jack placé SDL

Clic gauche SDL : Se déplacer

Déplacement de Jack SDL, si tour

Clic gauche SDL: Se déplacer

Déplacement des policiers SDL, si tour

Clic gauche SDL : Se déplacer

Déplacement de Jack SDL

Policier 1 : clic droit SDL : révéler une zone

Cercle jaune car Jack s’y est rendu

Policier 2 : clic droit SDL sur la zone ou est Jack

Les policiers  
ont gagné, cercle rouge chez Policiers à l’emplacement

# Pour aller plus loin

Dans cette partie nous allons essayer de ne pas nous étendre car il y aurait une infinité de possibilité d’ajouter ou de modifier notre jeu afin de le rendre plus drôle, plus dynamique, plus ergonomique…

La première chose que nous aurions aimé renforcer est le contrôle des actions des joueurs. Par exemple, lorsque Jack veut aller sur les carrés (il ne devrait pas pouvoir), ou encore lorsque les policiers veulent bloquer Jack en l’encerclant : rien de permet de contrôler l’adjacence et l’identification des points du jeu.

La seconde amélioration que nous aurions aimé mettre en place est la reconnexion au serveur après timeout. Si le client se déconnecte, le serveur qui attend une action de sa part s’en rend compte. Ainsi il peut mettre le jeu en pause et passer à une phase de reconnexion. Si le joueur tente la reconnexion avec la même adresse IP, le serveur relance la partie là où il s’est arrêté.

La dernière amélioration aurait été de créer plus d’éléments graphiques, notamment :

* Un menu principal
* Des fenêtres pop-ups d’action erronées
* Une fenêtre de victoire (resp. défaite) pour le/les gagnants (resp. perdants).

# Conclusion

Pour conclure, l’objectif du projet est atteint car nous avons réussi à modéliser une application client-serveur, à travers un jeu programmé en C utilisant des threads, et les protocoles TCP-IP.

Tout au long de ce projet, nous avons été capables de démontrer nos compétences avec les 4 étapes du déroulement de notre jeu : Tout d’abord avec la connexion, puis avec l’initialisation du jeu, en passant bien sûr par le cœur du jeu, pour terminer avec la fin du jeu.

Nous avons pu également nous faire la main sur les différentes fonctionnalités que la librairie SDL propose.