|  |  |
| --- | --- |
|  | **Projet LO23 ChessP2P** |
|  | Auteur(s) :   * Communication et traitements * Gestion de données * IHM Connexion * IHM Grille |

|  |
| --- |
| **Dossier de Réalisation**  **Réalisation d’un jeu d’échec en réseau décentralisé** |
|  |

Table des matières

[1 Introduction 3](#_Toc341188506)

[2 Communication et traitement 4](#_Toc341188507)

[2.1 Introduction 4](#_Toc341188508)

[2.2 Choix technologiques 4](#_Toc341188509)

[2.2.1 La communication avec l'adversaire 4](#_Toc341188510)

[2.2.2 La découverte des joueurs 4](#_Toc341188511)

[2.3 Difficultés rencontrées 4](#_Toc341188512)

[2.3.1 Synchronisation des threads 4](#_Toc341188513)

[2.3.2 Thread MAJ interface graphique 5](#_Toc341188514)

[2.3.3 Apprentissage de JAVA 5](#_Toc341188515)

[2.4 Points particuliers 5](#_Toc341188516)

[2.4.1 Polymorphisme 5](#_Toc341188517)

[2.4.2 Récupération adresse IP machine 5](#_Toc341188518)

[3 Gestion des données 6](#_Toc341188519)

[3.1 Introduction 6](#_Toc341188520)

[3.2 Choix technologiques 6](#_Toc341188521)

[3.3 Difficultés rencontrées 6](#_Toc341188522)

[3.4 Points particuliers 6](#_Toc341188523)

[4 IHM connexion 7](#_Toc341188524)

[4.1 Introduction 7](#_Toc341188525)

[4.2 Choix technologiques 7](#_Toc341188526)

[4.3 Difficultés rencontrées 7](#_Toc341188527)

[4.4 Points particuliers 7](#_Toc341188528)

[5 IHM grille 8](#_Toc341188529)

[5.1 Introduction 8](#_Toc341188530)

[5.2 Choix technologiques 8](#_Toc341188531)

[5.3 Difficultés rencontrées 8](#_Toc341188532)

[5.4 Points particuliers 8](#_Toc341188533)

[6 Conclusion générale 9](#_Toc341188534)

# Introduction

# Communication et traitement

## Introduction

Suite à l’établissement du dossier de conception, l’équipe du module « communication et traitement » a entamé la phase de réalisation de son module. Nous avons veillé à répartir le travail en fonction des ressources humaines disponibles au sein de notre groupe. Au cours de cette phase de développement, nous avons du faire face à diverses problématiques.

Dans un premier temps, nous allons mettre en lumière les divers choix technologiques que nous avons faits. Ensuite nous nous attacherons aux principales difficultés rencontrées. Et pour finir, nous évoquerons divers points spécifiques.

## Choix technologiques

Nous présenterons dans cette partie les divers choix techniques auxquels nous avons du faire face tout en argumentant nos choix.

### La communication avec l'adversaire

Pour cette communication avec l'adversaire, nous avons décidé d'utiliser java.net.Socket et java.net.ServerSocket qui permet de communiquer une mode connecté (TCP) assez simplement. Par exemple, on peut envoyer et recevoir directement des objets Java sérialisables via java.io.ObjectOutputStream et java.io.ObjectInputStream et une exception se produit s'il se produit un problème avec la connexion.

### La découverte des joueurs

Après avoir fait quelques recherches et quelques tests, nous avons décidé d'utiliser le mode multicast pour l'envoie d'un message à tout les joueurs disponibles. Il nous a semblé que le mode multicast était plus simple à mettre en œuvre que le mode broadcast car on n’a pas besoin de connaître précisément la topologie du réseau pour envoyer un message. Il faudra donc utiliser java.net.MulticastSocket pour lire les paquets reçus en mode multicast et java.net.DatagramSocket pour envoyer un paquet en mode non-connecté (UDP). Enfin, il faudra également utiliser une adresse IP multicast comprise entre 224.0.0.0 et 239.255.255.255 en vérifiant bien que cette adresse n'est pas réservée.

## Difficultés rencontrées

Nous évoquerons dans cette partie les différentes difficultés que nous avons rencontrées et qui nous ont amenées à adapter notre code.

### Synchronisation des threads

Quand on veut récupérer des messages du réseau, les appels des méthodes concernées sont bloquants, c'est-à-dire que le thread est en attente jusqu’à la réception effective du message. Il faudra donc mettre en place des threads dédiés à la réception des messages mais aussi faire très attention aux problèmes de synchronisation entre les threads. Par exemple, si on envoie un message et qu’au même moment on a reçu un message du réseau alors il est possible d’avoir des problèmes d’accès concurrent à certaines ressources.

### Thread MAJ interface graphique

Un autre problème vient de Swing qui ne peut mettre à jour son interface graphique que dans un thread particulier, le Event Dispacher Thread, appelé aussi EDT alors que nous avons différents threads dédiés au réseau. Nous avons décidé de traiter cette problématique dans notre module pour ne pas la répandre dans les autres modules notamment les modules IHM. Nous devrons donc utiliser des méthodes et des objets spécifiques comme SwingUtilities ou SwingPropertyChangeSupport pour exécuter les interfaces des autres modules dans le thread EDT.

### Apprentissage de JAVA

Il s’est avéré que certain membres du groupe n’ayant pas pratiqué le langage JAVA, une période d’apprentissage (auto-formation) a été prise en compte.

## Points particuliers

Cette partie traitera de divers points ayant été soulevés avant, pendant et après la phase de réalisation du module communication et traitement.

### Polymorphisme

En effet, au lieu de tester la classe de chaque message reçu afin d’appeler le traitement correspondant, nous aurions pu utiliser le concept de polymorphisme. Nous aurions défini une classe abstraite « Message »possédant une méthode « traiter » et nous aurions redéfini cette méthode dans chacune des classes filles afin de spécifier le comportement à adopter vis-à-vis de chaque type de message. Ceci nous aurait évité de tester explicitement la classe du message reçu dans les Handlers liés à la réception de messages.

### Récupération adresse IP machine

L’application cliente étant identifiée sur le réseau via son profil (contenant une adresse IP) et le fait qu’un ordinateur muni de plusieurs cartes réseaux puisse disposer de plusieurs adresses IP a soulevé une question : quelle adresse IP devons-nous récupérer lorsque plusieurs sont disponibles (cas de la majorité des pc portables munis d’une interface réseau WIFI et Ethernet). Afin de répondre à cette question, il a été décidé que nous écrirons une fonction permettant de sélectionner la première adresse IP locale étant différente de celle de loopback.

# Gestion des données

## Introduction

## Choix technologiques

## Difficultés rencontrées

## Points particuliers

# IHM connexion

## Introduction

## Choix technologiques

## Difficultés rencontrées

## Points particuliers

# IHM grille

## Introduction

## Choix technologiques

## Difficultés rencontrées

## Points particuliers

# Conclusion générale