

# Guide d'utilisation de l'interface

Jalal AZOUZOUT — Saghar YAZDANI — Ali BA FAQAS — Leila BOUROUF

May 12, 2023

## 1 Introduction

Voici le guide d'utilisation pour bien utiliser l'interface, il est conseillé de commencer par la création du fichier de configuration de l'interface graphique puis l'implémentation de la connexion dans votre programme à l'aide du fichier "interface.h" fournis. Prenez soin de bien choisir votre nombre d'états et aux couleurs associées, ce qui facilitera la compression durant le fonctionnement de votre programme.

## 2 Initialisation de l'interface : Creation du fichier de configuration

Avant de connecter votre programme, ou de lancer l'interface, il faut d'abord configurer le fichier .txt placé dans le dossier "Files", s'il n'existe pas, créé-le.

Ensuite entré ses informations dans le fichier : 1.Ajouter le nombre de sites lancé 2.Nombre d'états 3. Choisir si les couleurs associées aux états sont générées (0) ou choisies par vous (1). 4. n lignes d'état avec ou sans le numéro des couleurs au format, id-etat : nom-etat valeur-de-couleur.

Si vous avez choisi d'initialiser les couleurs par vous-même, il faudra ajouter 3 chiffres allant de 0.0 à 1.0.

Pour le noir : 0 0 0 Pour le blanc : 1 1 1

exemple avec choix des couleurs :

```
#Nombre de sites
4
#Nombre d'états
4
#Choix(O ou 1) 1
#Liste d'états
5:attente 0.1 1.0 0.0
1:fermer 0.1 0.0 0.0
3:traitement 0.1 1.0 1.0
4:initialisation 0.3 1.0 0.3
```

exemple avec couleur générer :

```
#Nombre de sites
4
#Nombre d'états
4
#Choix(O ou 1) 0
#Liste d'états
5:attente
1:fermer
3:traitement
4:initialisation
```

## 3 Connection du processus à l'interface

Après avoir configuré le fichier, vous allez devoir implémenter l'api dans votre programme.

Veillez bien à suivre les étapes dans l'ordre dans votre programme, sinon il ne fonctionnera pas.

### 3.1 Initialisation de la connexion

Premièrement, il faut inclure le fichier interface.h dans votre programme. Inclure "interface.h"

Deuxièmement, il faut initialiser un struct qui contiendra l'adresse de la machine où est situé le MainServer, et un port que vous choisirez.

Exemple : `struct Server inter; inter.addr = "127.0.0.1"; inter.port = 3000;`

### 3.2 Etablir la connexion à l'interface

Ensuite, pour établir la connexion avec le MainServer, il faudra ajouter cette fonction qui prend en paramètre la struct précédemment définie et qui retourne un int de valeur 1 si la connexion est établie, -1 sinon.

`int connectConnectionInterface(struct Server* inter);`

### 3.3 Envoyer de l'id du processus

Attention : Assurez-vous bien que la connexion au MainServeur est bien établie.

Après avoir établi la connexion, pour que l'interface puisse identifier le processus courant qui lui enverra les données, il faudra lui envoyer un identifiant, ainsi vous devez ajouter cette fonction à la suite, qui prend en paramètre l'adresse de la struct et le numéro d'identifiant et retourne un entier, -1 si erreur lors de la connexion, 1 sinon.

`int sendId(struct Server* inter, int id);`

Remarque : Se connecter à l'interface et envoyer l'identifiant du site sont les deux premières étapes que votre programme doit accomplir.

### 3.4 Envoyer d'une structure de donnée

Pour indiquer à l'interface quel processus est connecté à celui-là via une connexion TCP, vous devez envoyer les sockaddr\_in de toutes les connexions une par une à l'aide de cette fonction :

`int sendStruct(struct Interface* inter, struct sockaddr_in id);`

Ainsi vous devrez ajouter cette fonction qui prend en paramètre l'adresse de la struct Server et une structure de type sockaddr\_in, -1 si erreur lors de l'envoi du message, 1 sinon.

### 3.5 Envoyer un état à l'interface

Maintenant que la connexion est établie avec le MainServer, Vous pouvez ajouter une ou plusieurs fonctions "sendStatus" à des points-clés où votre programme change d'états, et qui enverra une valeur de type int que vous avez précédemment définie dans le fichier de configuration.

La fonction prend en paramètre l'adresse de la struct et le code précédemment initialisé associé à l'état de votre programme et retourne un entier, 1 si le code est bien reçu, -1 sinon.

`int sendStatus(struct Server* inter, int code);`

### 3.6 Fermeture de la connexion

```
int closeConnectionInterface(struct Server* inter);
```

Pour fermer la connexion avec l'interface, il faut ajouter cette fonction qui prend en paramètre l'adresse de la struct initialisée et qui retourne un entier, -1 si erreur lors de la connexion 1 sinon.

## 4 Visualisation

Maintenant, nous allons voir comment utiliser l'interface graphique pour visualiser les processus.

### 4.1 Compilation du programme

Pour compiler il faut utiliser le fichier compile.script

commande: `bash compile.script`

### 4.2 Lancement du programme

Pour ne pas avoir de problème, prenez bien soin de lancer en premier l'interface grâce à la commande `"/MainServeur Port-utiliser chemin-fichier-config"` puis votre programme.

### 4.3 Zoom sur un processus

Pour pouvoir zoomer sur l'activité d'un processus en particulier, vous pouvez en remplissant la barre de saisie située en haut à droite le numéro du processus puis appuyé sur submit, afficher son activité.