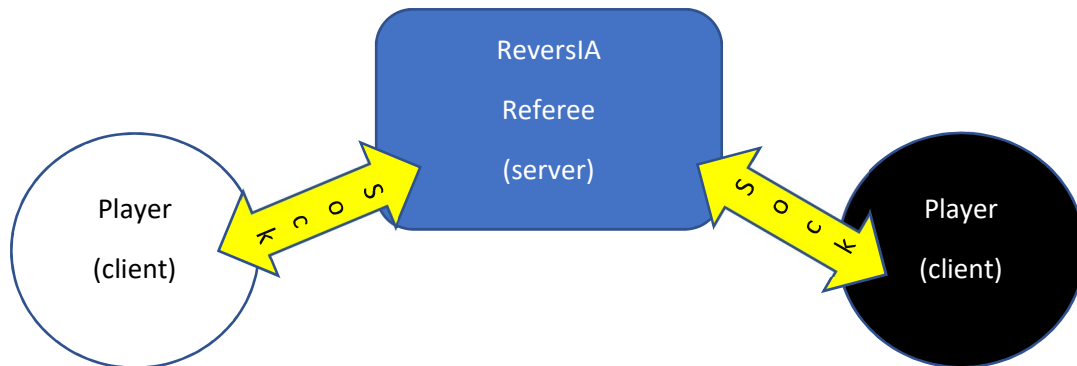


ReversiA Quick Start

Introduction



ReversiA propose à deux programmes, algorithmes, IA, de se mesurer entre elles dans une partie de Reversi (aussi connu sous le nom d'Othello mais ce dernier est une marque commerciale déposée par « tsukuda original »). Le système se compose d'un arbitre et de deux joueurs. Les joueurs peuvent dialoguer avec l'arbitre via des « socket ». L'arbitre est un serveur et se chargera de gérer la partie :

- Accueil des joueurs
- Affectation de la couleur
- Démarrage de la partie
- Affichage du tablier (plateau de jeu)
- Comptage du temps de jeu
- Vérification du mouvement de chaque joueur
- Comptage du score

L'arbitre a été écrit en langage JAVA pour sa simplicité au niveau de l'affichage graphique.

Les joueurs devront se connecter, envoyer un message de connexion et attendre les requêtes de l'arbitre. Ils devront être écrit en C sous Windows. Les sockets vont permettre à 3 programmes différents d'interagir entre eux. Ils peuvent communiquer sur la base de messages prédéterminés à l'avance (défini dans les slides ReversiA) Ces programmes peuvent être écrit en langage différents ou tourner sur des machines différentes. Lors du développement les trois programmes tourneront sur votre machine. Lors du tournoi nous utiliserons un routeur/point d'accès wifi permettant de mettre en lien toutes les machines participantes.

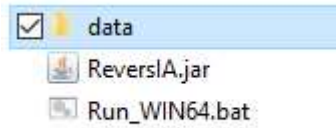
Description de l'archive :

Dans l'archive vous trouverez deux répertoires :

Referee_ReversiA : arbitre en mode entraînement

Player_ConnectMessageSample : 2 joueurs exécutables et un exemple de code permettant de se connecter à l'arbitre.

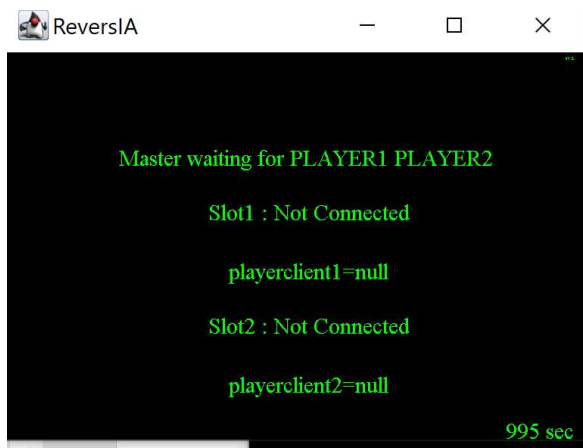
Referee_ReversIA :



L'arbitre est un programme java archivé dans un « .jar » pour le lancer il suffit de double cliquer sur « Run_WIN64.bat ». Il faut aussi qu'une machine virtuelle java soit installée sur votre machine. Si votre machine a déjà une JVM : rien à faire. Sinon :

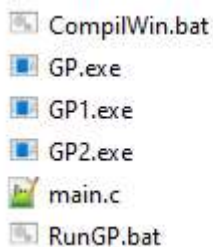
Aujourd'hui il y a un scission du java. Oracle, le propriétaire de Java, a décidé que Java redevenait payant. Mais il laisse les développeurs open source le droit de continuer à fournir des JVM. Pour ma part j'ai choisi « AdoptOpenJDK » : <https://adoptopenjdk.net/>

Lors que vous double cliquer sur Run_WIN64.bat vous devriez vous apparaitre la fenêtre :



L'arbitre est lancé et il attend la connexion de deux joueurs.

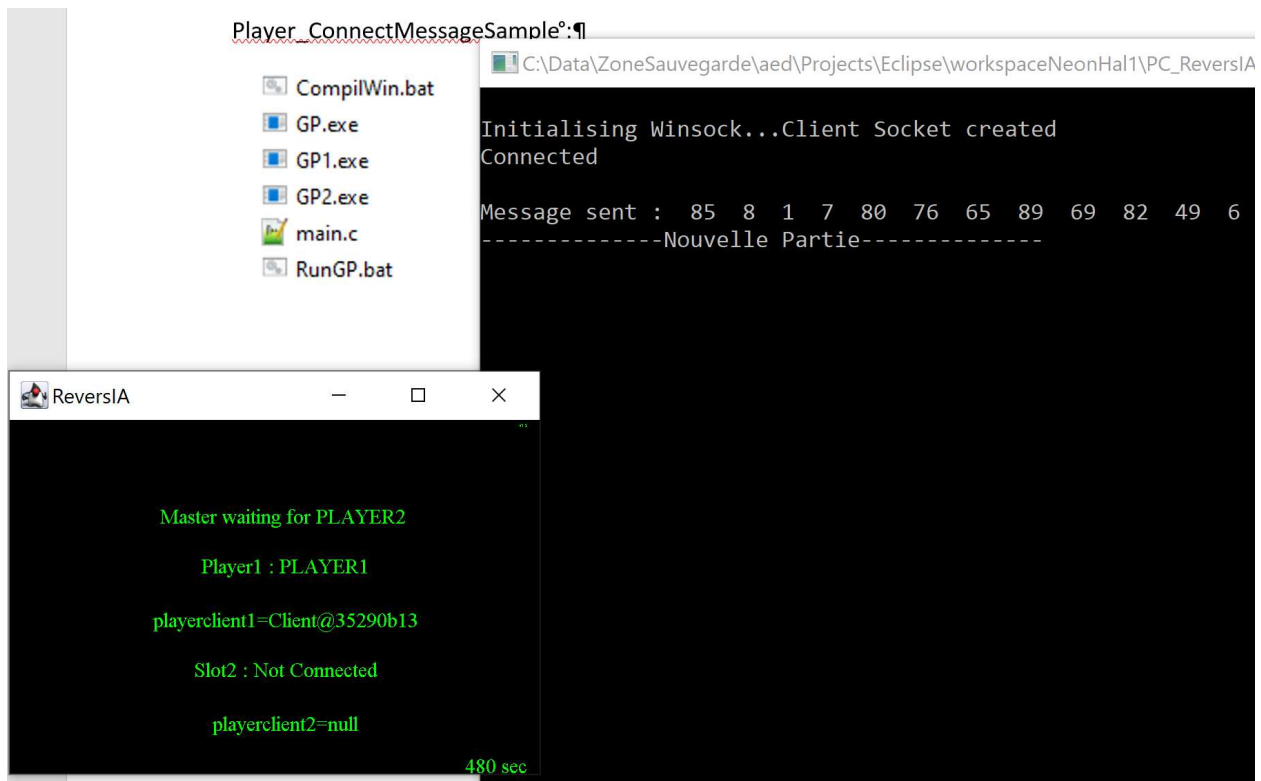
Player_ConnectMessageSample :



C'est ici que vous allez développer votre Player (n'oubliez pas de lui donner un petit nom, c'est avec ce nom que l'arbitre pourra reconnaître le Player).

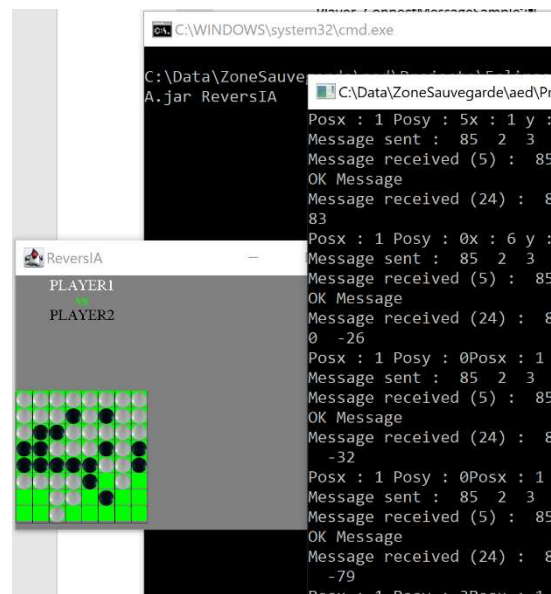
Il y a deux joueurs prêts à être exécutés : GP1.exe et GP2.exe. GP1 et GP2 sont les deux mêmes algorithmes. Ils jouent la première position « jouable ». Ils n'offrent donc pas un grand challenge mais permettent de vous fournir une base et aussi de vérifier que votre environnement fonctionne. Ils diffèrent aussi par leur « petit nom » : GP1 se nomme « PLAYER1 » et GP2 « PLAYER2 ».

Si l'on double clique sur GP1.exe, l'arbitre affiche :



On voit dans une fenêtre de commande le message qui a été envoyé à l'arbitre pour se connecter. Ça pourra vous servir.

On double clique sur GP2.exe :



Dans la console qui a lancé l'arbitre vous pouvez voir aussi toute la partie en version « ASCII ».

Cette contient aussi tous les messages reçus. Vous pourrez vous en inspirer pour tracker vos bug.

Maintenant que le système est en place, vous trouverez dans ce même répertoire un « main.c » qui contient le code source permettant d'ouvrir une socket et de se « présenter » à l'arbitre. Pour compiler le main.c, il suffit de double cliquer sur CompilWin.bat. Ce fichier contient la ligne de commande GCC pour windows :

Remarquez le `-DMINGW32` qui est équivalent à `« #define MINGW32 »` dans le code. En effet, vous verrez dans le code de `#ifdef MINGW32` car les fonctions concernant les socket ne sont pas tout à fait identiques entre Linux et Windows. De même la ligne de commande indique à GCC d'intégrer la librairie `winsock` : `« -lw2 32 »` ce qui ne sera pas nécessaire si vous utilisez Linux.

Evidemment vous devez avoir un GCC sur votre machine... Pour ma part j'ai compilé avec celui-ci (mais d'autre peuvent convenir) :

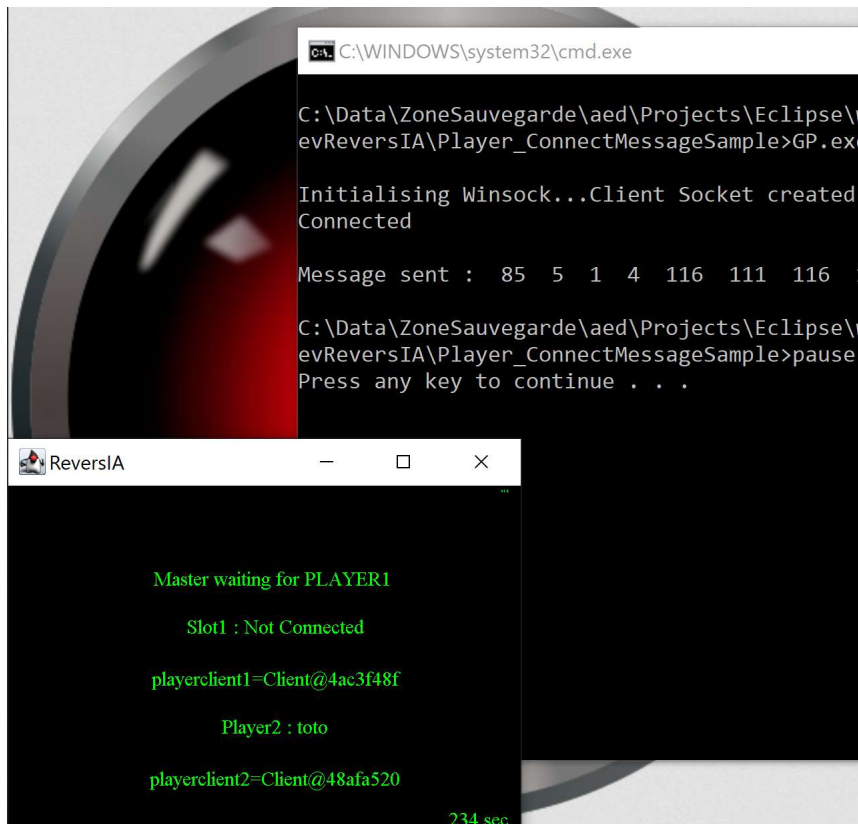
```
ctMessageSample> gcc --version
gcc.exe (x86_64-posix-seh-rev0, Built by MinGW-w64 project) 8.1.0
Copyright (C) 2018 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Lorsque vous double cliquez sur CompilWin.bat vous devriez vous apparaitre GP.exe dans le répertoire.

Pour lancer GP.exe il suffit de double cliquer sur RunGP.bat qui contient la ligne suivante :

GP.exe toto

« toto » est un argument qui est récupéré par GP.exe et utilisé comme nom pour se présenter à l'arbitre. Donc lorsque vous lancez RunGP.bat, l'arbitre affiche :



Toto a été enregistré dans le slot2 de l'arbitre.

Maintenant, à vous de jouer (façon de parler...). Que le meilleur gagne !

FIGHT