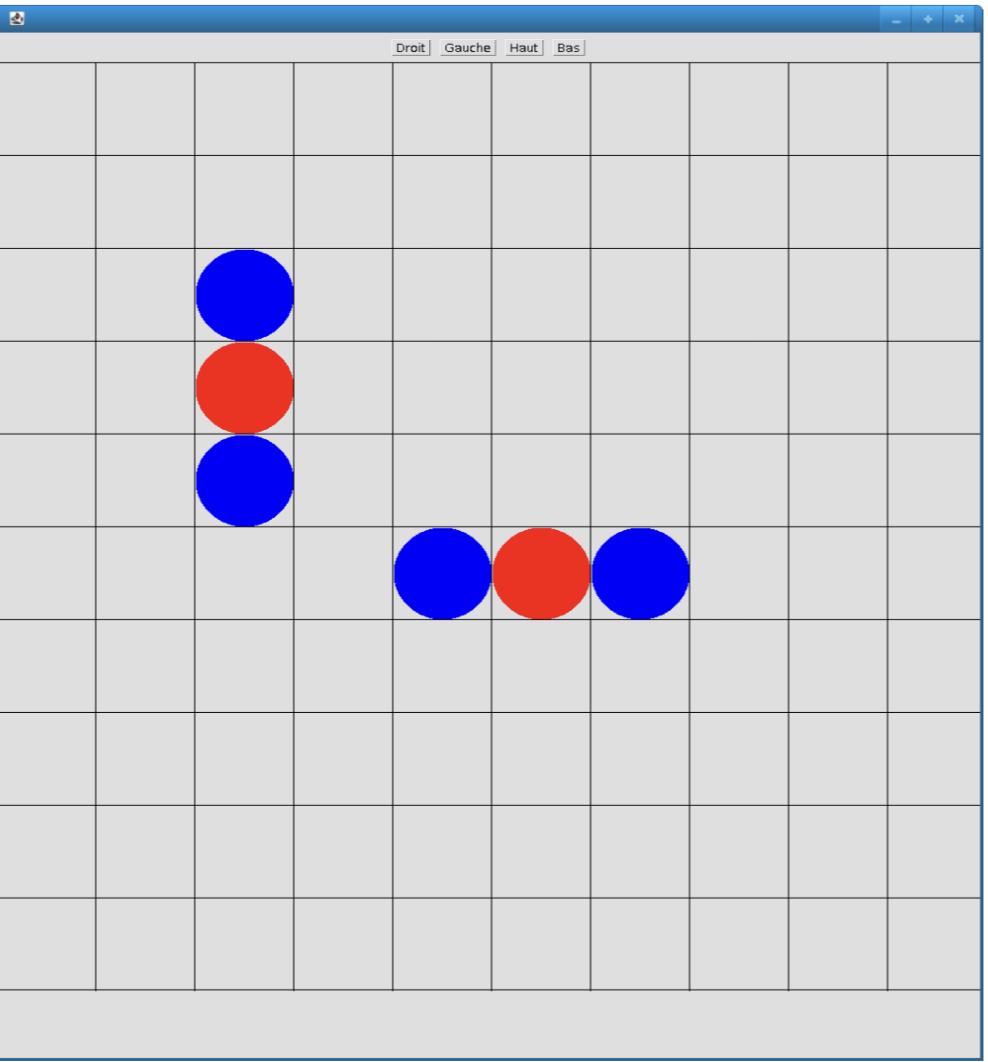
**Compte rendu : Réseau et Transmission**

**Introduction :**

Dans le cadre de la matière nous avons réalisé un projet qui consiste en un jeu. Il consiste à bloquer le joueur adverse horizontalement ou verticalement pour gagner. Il y a 2 équipes : rouge et bleu. Le client devra ici interagir avec le client pour traiter les mouvements demandés par les joueurs, les mouvements peuvent être les suivants : gauche, droite, haut ou bas.



Mon dépôt contient les 2 exécutables : le client et le serveur qui sont les éléments principaux pour que le jeu puisse accueillir 2 joueurs différents. Ensuite il y a un code source commenté pour pouvoir être compréhensible pour n’importe quel utilisateur qui souhaite le modifier par exemple.

La présence d’un README est également pour expliquer comment exécuter le jeu.

Pour exécuter le jeu nous avons simplement à écrire la commande suivante dans le terminal : « java -jar ./serveurJeu.jar » qui permet de lancer le serveur java et ensuite il faudra effectuer la commande : « java -jar ./reseauxJeu.jar » qui elle sert à lancer la partie client du serveur donc le jeu.

**Partie client :**

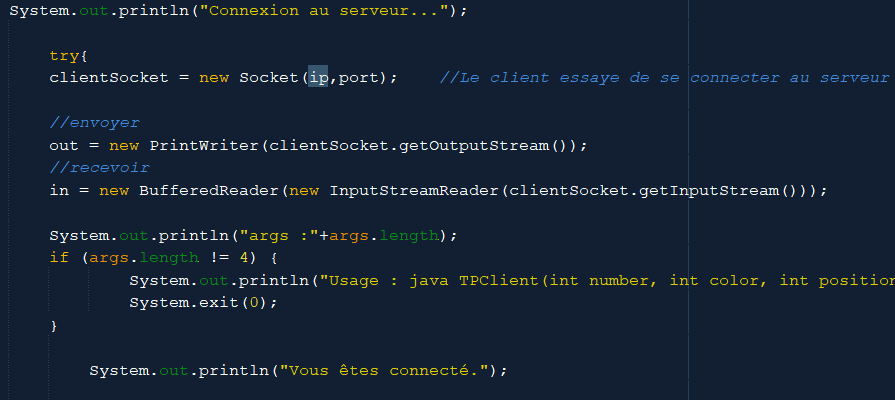
Tout d’abord la partie client se découpe en plusieurs classes. La classe TPPanel gère les boutons de déplacements, la classe TPCanvas gère l’affichage de la grille et des joueurs et la classe TPClient gère le main et l’exécution des déplacements, communication avec le serveur.

Le but de notre client est de choisir une équipe, une position d’apparition, de se connecter au serveur et pouvoir jouer. C’est pourquoi il nécessite quatre arguments au lancement:

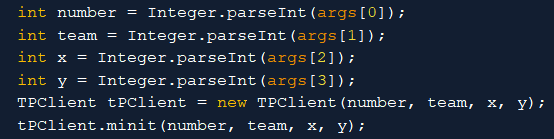
Le numéro du joueur, son équipe (1 étant bleu, 2 étant rouge), sa position x et sa position y.



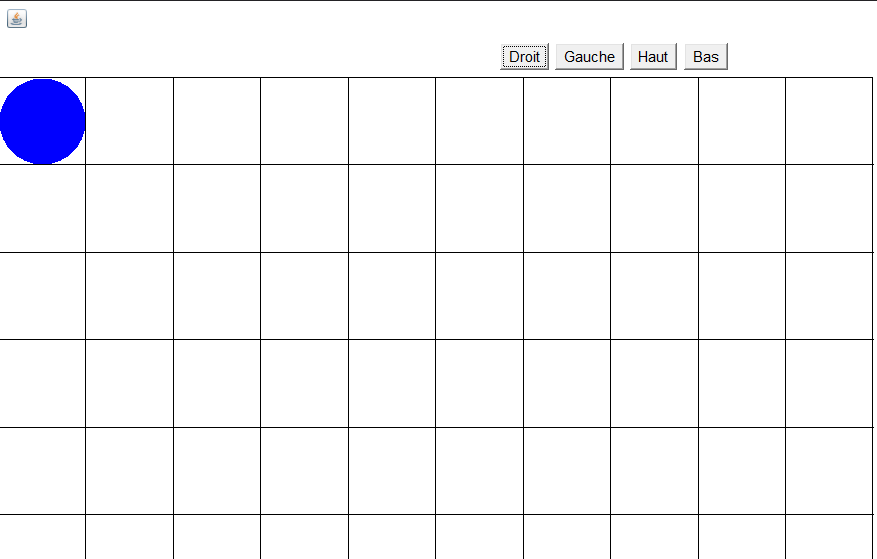
Il est important de d’abord lancer le serveur avant le client. On exécute le client avec la commande ci-dessus. Le client essaye alors de se connecter au serveur.



Si la connexion est réussie , alors on va initialiser le joueur avec les paramètres précisés.

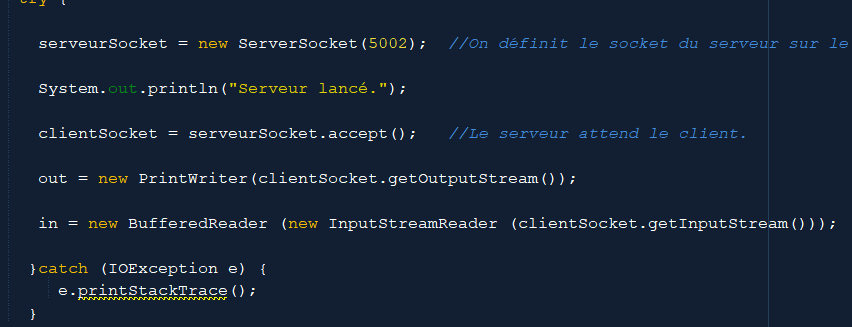


Les arguments sont convertis en int

Le joueur apparait bel et bien en position indiquée. 

**Partie Serveur :**

Lorsque la partie serveur est exécutée le serveur se lance dans un port qui a été précisé dans le code. (Ici le port 5002)



Ensuite le serveur passe en mode écoute ce qui signifie qu’il attend la connexion d’un client. Lorsque le serveur reçoit le socket client la connexion est établie et la partie de jeu peut commencer mais s'il y a une erreur cela renvoie une exception.

**Conclusion :**

Pour conclure je n’ai pas réussi à aller au bout du sujet car la partie communication des positions avec le serveur a posé problème et cela m’a empêché d’avancer. Il était compliqué d'envoyer une position au serveur et d'actualiser avec le timer. Hors j’ai réussi à mettre en place une communication avec le serveur. Donc malgré le non-achèvement du projet cela à été tout de même enrichissant et intéressant à travailler.