

# Projet de réseau

## Bataille navale

L'objectif du projet est de réaliser en binômes un jeu de bataille navale en réseau.

## 1 Sujet

### 1.1 Description générale

#### 1.1.1 Jeu de la bataille navale

Le jeu de la bataille navale est décrit très clairement sur cette page wikipédia [https://fr.wikipedia.org/wiki/Bataille\\_navale\\_\(jeu\)](https://fr.wikipedia.org/wiki/Bataille_navale_(jeu)).

### 1.2 Information relative au code source

Le projet est packagé de la manière suivante et disponible à <http://dept-info.labri.fr/~thibault/Reseau/bataille.tgz> :

```
game.py
main.py
```

Ce code vous permet de jouer à un joueur contre la machine (qui joue simplement des coups aléatoires).

## 2 Passage en réseau

Il s'agit maintenant de jouer à 2 joueurs se connectant sur un même serveur...

Dans un premier temps, faites une version simple avec un support pour seulement 2 joueurs (clients) à la fois. Sur le serveur, on lance le programme sans argument, et le programme doit alors ouvrir une socket TCP en écoute. Sur les clients, on lance le programme avec le nom du serveur en paramètre, le programme doit, dans ce cas, se connecter en TCP à la machine passée en paramètre.

Maintenant que l'on a une socket établie entre chaque client et le serveur, il s'agit de transmettre les informations sur les coups joués et l'état de la grille du joueur (ce sont les clients qui feront le rendu du jeu chacun de leur côté). Il faut donc établir un protocole pour les exprimer. Il est recommandé d'établir un protocole textuel ascii, qu'il sera ainsi facile de déboguer à la main. Réfléchissez et implémentez dans un premier temps votre propre protocole. Puisque la bataille navale se joue chacun son tour, le serveur pourra simplement réceptionner tour à tour depuis chaque joueur.

## 3 Extensions

Plusieurs extensions sont possibles, à vous d'essayer d'en implémenter !

- On peut vouloir jouer plusieurs parties, en comptant les scores.
- Un des joueurs peut être un robot qui joue aléatoirement des coups.
- Si un troisième client se connecte, c'est un observateur : il voit tous les coups joués.
- Lorsque la partie est finie, le serveur demande qui veut jouer la partie suivante parmi les 3 clients, et le 3e se retrouve observateur.
- Il peut y avoir bien plus que 3 clients, les autres étant aussi observateurs.

- Il se peut que les connections entre le serveur et les clients soient interrompues. On ne veut pour autant pas perdre la partie en cours. Du côté du serveur, on doit donc simplement mettre en pause la partie en attendant que le client perdu se reconnecte. Du côté du client, on essaie simplement de se reconnecter. Il s'agit en fait d'implémenter la couche Session du modèle ISO. Pour tester, vous pouvez simplement ajouter un `s.close()` qui survient au bout d'un certain temps (utiliser `time.time()` pour obtenir l'heure en secondes).
- Il se peut qu'un client soit tué (crash de la machine, control-C inattendu, ...), il a donc perdu l'état du jeu. Lorsqu'on le relance, il faudrait que le serveur lui redonne l'état du jeu pour pouvoir reprendre la partie.
- et tout autre idée intéressante, discutez-en avec votre chargé de TD!

## 4 Organisation

Le projet est travaillé et étudié en binôme mais la notation est individuelle. Le projet *doit* être réalisé avec un outil de gestion de révision (svn, git, ...) : directement dans votre *home* ou sur la savanne du CREMI (<https://services.emi.u-bordeaux1.fr/projet/savane/>). Les enseignants évalueront la contribution de chacun des éléments du binôme au travail commun. Les enseignants se réserveront la possibilité de modifier la composition de chacun des binômes. Afin de permettre un travail profitable, il est conseillé de ne pas créer de groupes avec des niveaux trop différents.

## 5 Rapport

Il s'agit de mettre en valeur la qualité de votre travail à l'aide d'un petit rapport. Pour cela le rapport doit explicitement faire le point sur les fonctionnalités du logiciel (lister les objectifs atteints, lister ce qui ne fonctionne pas et expliquer - autant que possible - pourquoi). Attention bien sûr au style, la grammaire, l'orthographe et la présentation.