TP2: Projet socket

1.Objectif

On souhaite réaliser une bataille navale entre plusieurs clients à l’aide du protocole socket.

|  |  |
| --- | --- |
| Consignes | Statut |
| Les donnes du jeu doivent être coté serveur | Consigne réalisée |
| Les clients doivent mémoriser leurs connaissances de la partie | Consigne réalisée |
| Possibilité de parties multiples | Consigne réalisée |

Principe : le Serveur et les clients sont lancés.

* Le serveur demande les Pseudo aux clients.
* Les clients entre les pseudos.
* Le serveur demande d’entrer M pour placer les navires de façon manuelle ou alors A pour les placer de façon Automatique et aléatoire (c’est un plus pour le projet et très adapté pour la situation en question de rapidité)
* Les clients choisissent la méthode et valident
* Le serveur affiche les grilles de chaque client et ensuite demande au joueur 1 de tirer puis au tour du joueur 2 de tirer.
* Le joueur 1 tire et le serveur affiche le résultat du tir (touché ou raté). Pareil pour le joueur 2
* L’un des clients gagne quand les navires de l’autre ont été tous coulées.
* Le système notifie les clients du gagnant
* Le système propose de rependre la partie
* Les clients choisissent ‘oui’ ou ‘non’ et le système se relance ou pas.

Les classes :

* Client
* Serveur
* ListeningThread
* Parties
* Grille
* ThreadBataille

Les classes importantes ici sont Grilles et ThreadBataille.

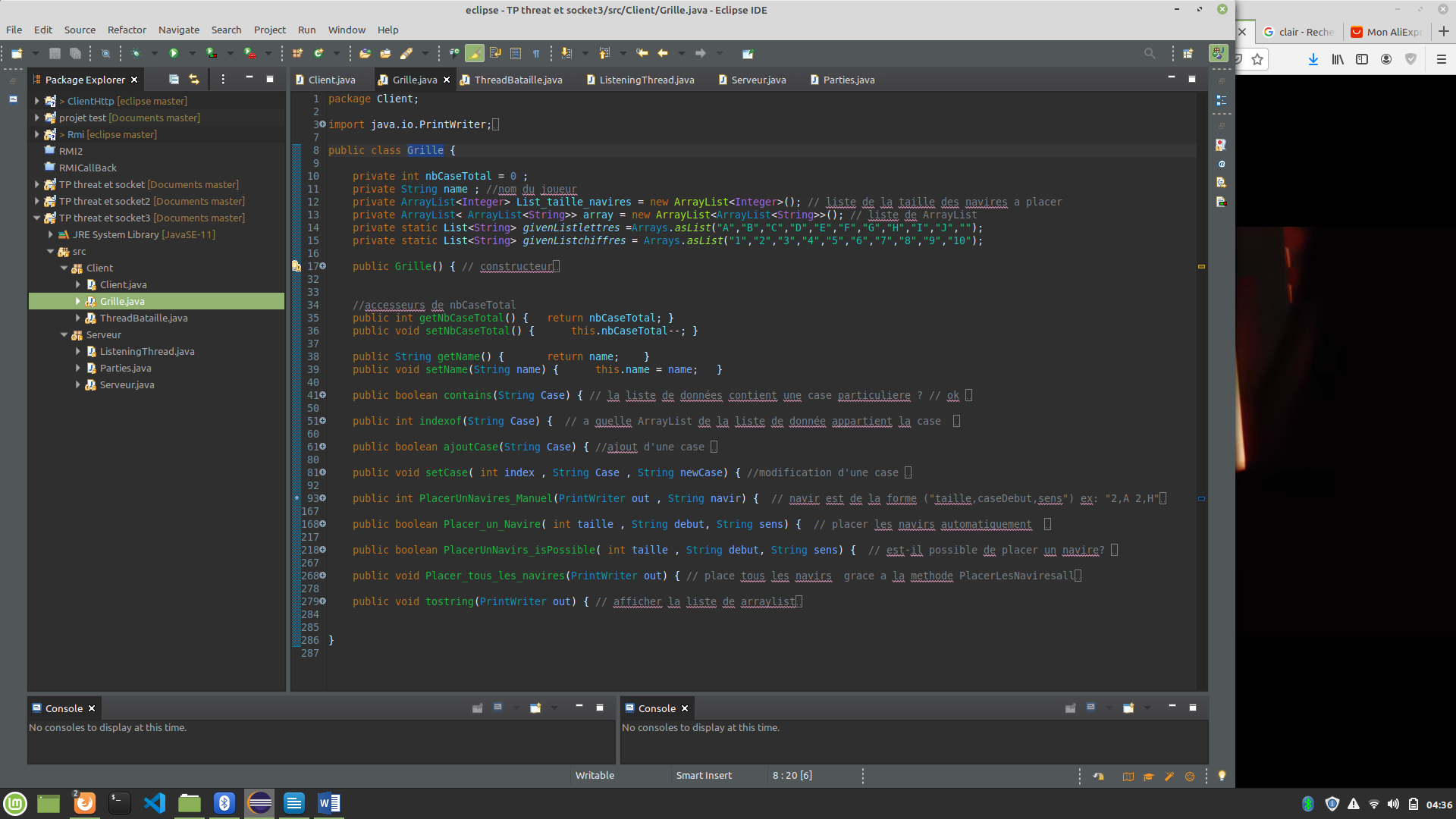
* Class Grille

Pour former les grilles, on cree une ArrayList< ArrayList<String>>.

* PlacerUnNavires\_isPossible() teste la possibilité de placer un navire
* PlacerUnNavires\_Manuel() place un navire manuellement. Elle s’assurre aussi que toutes les erreures pouvant lever des exeption sont traites. Toutes les conditions sont reunis pour que les navires soient placées comme il faut et ou il faut, en fontion des borures et de la taille du navire

* PlacerUnNavires()place un navire Automatiquement
* Placer\_tous\_les\_navires() place tous les navires Automatiquement et traite toutes les exeption et reglementations que la methode PlacerUnNavires\_Manuel().

NB : il y’a d’autres methodes tels que Containt(), setCase, tostring, indexOf() etc qui sont pour la plupart des methodes secondaires.



* Classe ThreadBataille
* Les méthodes PlacerNavires() et PlacerNavires\_aleatoirement() récupèrent les données client et placent les navires en fonction des joueurs
* La méthode Brodcast\_messages\_Serveur() envoie des messages a tous les client
* Tir se charge d’effectuer des tirs en fonction des clients.
* Run() met en scène le jeu : place les navires et effectue les tir et notifie en fonction du gagnant et relance si nécessaire la partie

La classe Parties qui définit les différentes parties n’est pas indispensable. Elle aurait pu être implémenté directement dans la classe serveur, mais elle demeure utile si on veut créer une instance de la classe partie afin de créer une partie.

