

La réalisation du projet s'articule autour de trois fichiers dont deux classes :

- `MonteurInference.java` qui contient l'ensemble des fonctionnalités relatives à la gestion des faits et des règles ainsi que la prise de décision.
- `Regle.java` qui est la représentation en objet d'une règle d'inférence.
- `Battleship.rules` qui contient l'ensemble des règles utilisées par le moteur d'inférence sous un format spécifique : `premise1,... ,premiseN => conséquence`.

La classe `Regle` comprend notamment les méthodes suivantes qui servent au traitement :

- `satisfaitCondition(fait)` qui permet de savoir si un fait correspond à une prémisse d'une règle.
- `satisfaitConditions(faits)` qui permet de savoir si un ensemble de faits satisfait tous les prémisses d'une règle.

La classe `MoteurInference` dont le constructeur se charge de lire le fichier des règles et de les instancier. Cette classe contient aussi les méthodes notables suivantes :

- `estSolution(fait)` qui vérifie si un fait est solution, c.-à-d. qu'il ne fait pas partie des faits connus.
- `int[] calculCoup()` qui contient toute la logique de décision du moteur d'inférence. Il utilise un chaînage avant simple pour décider quel sera le coup suivant à réaliser en fonction des faits de base contenu dans la base de faits du moteur d'inférence. Cette fonction est aussi capable de revenir un coup en arrière si l'avant dernier coup a été une réussite et si le dernier coup a été un échec. Le chaînage avant simple consiste à parcourir l'ensemble des faits initiaux et voir si pour un fait donné il existe une règle dans le cas où le fait n'est déjà pas une solution. Si le fait donné est une solution alors il est ajouté dans la base de faits courante du moteur. Une fois l'ensemble des faits parcourus, la base de faits du moteur contient alors le fait lui permettant de prendre la décision.