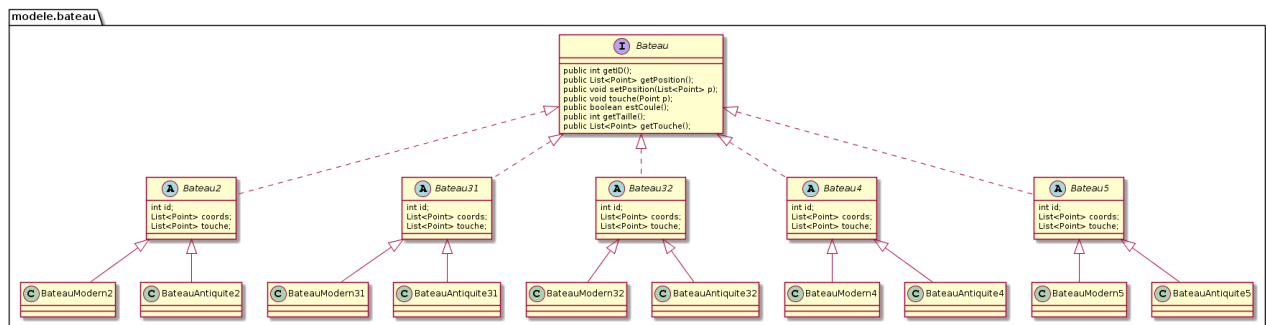


Rapport

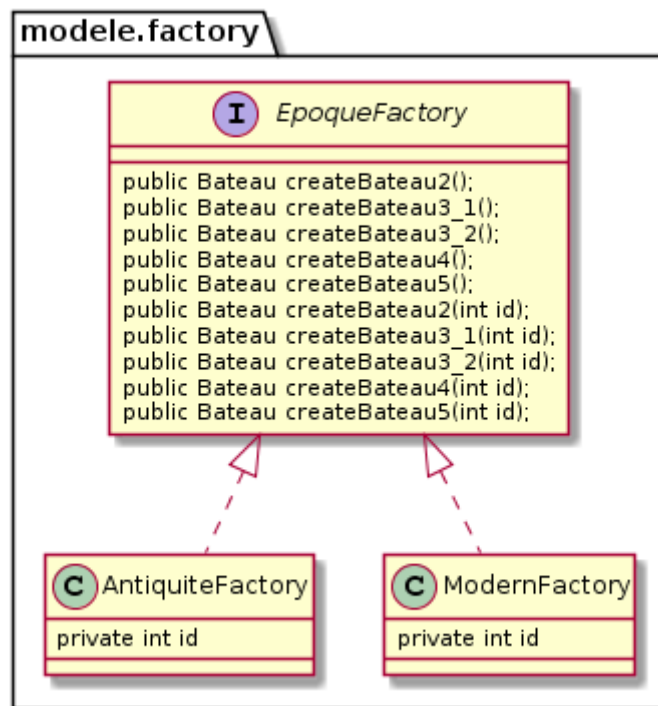
Les sources sont disponibles sur git :

<https://github.com/damienlevy/Navaleproject2018>

Dans notre conception, nous avons décidé de créer une interface Bateau puis créer des classes abstraites correspondant à la taille de chaque bateau afin de normaliser les fonctions pour permettre l'extensibilité de l'application.

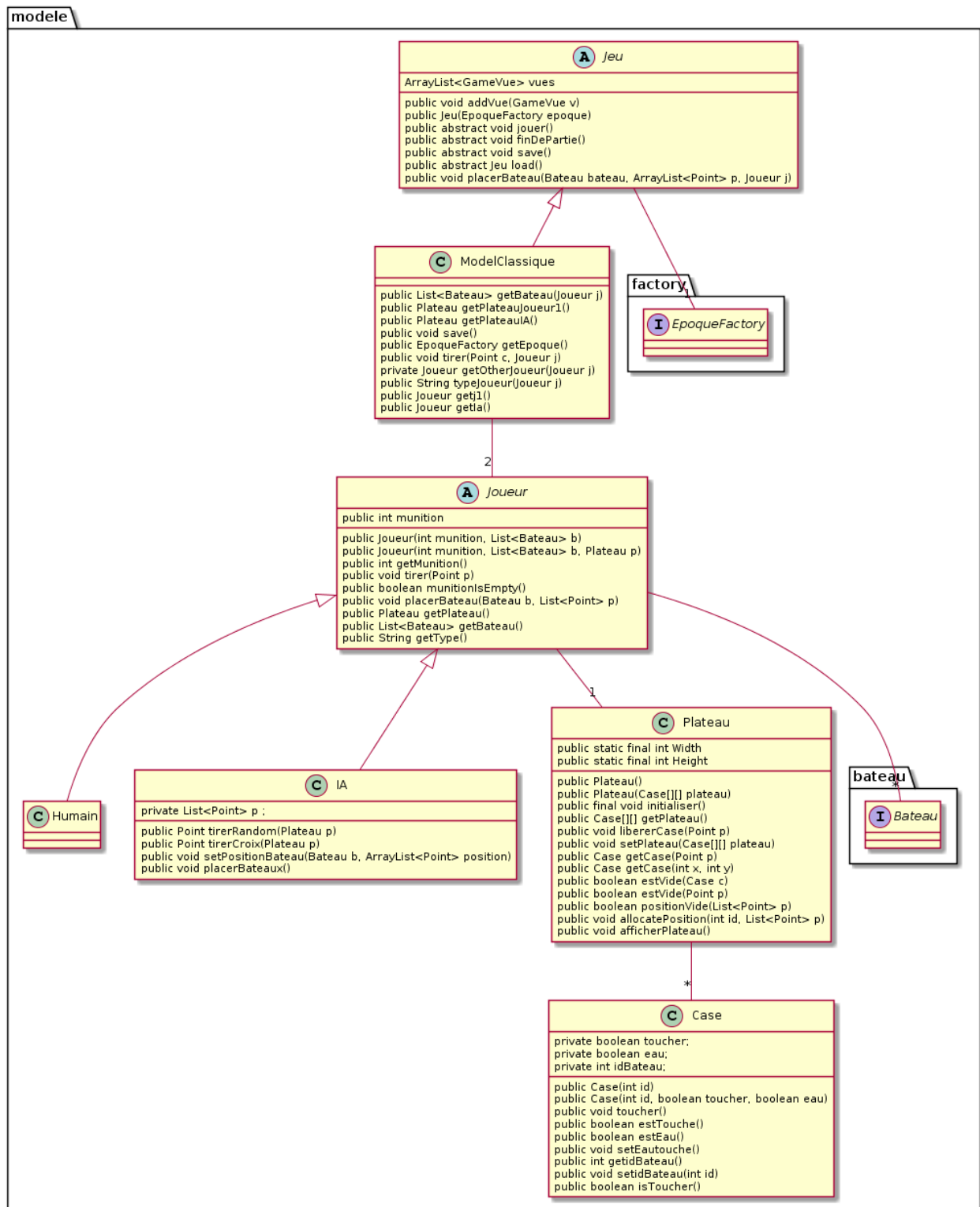


Nous avons décidé de créer une factory par époque. Pour ce faire nous nous sommes basés sur le design pattern « factory methods ». Il nous a permis de créer facilement les bateaux liés à une époque. Cela permet aussi d'ajouter une nouvelle époque sans avoir à modifier toutes les classes existantes.



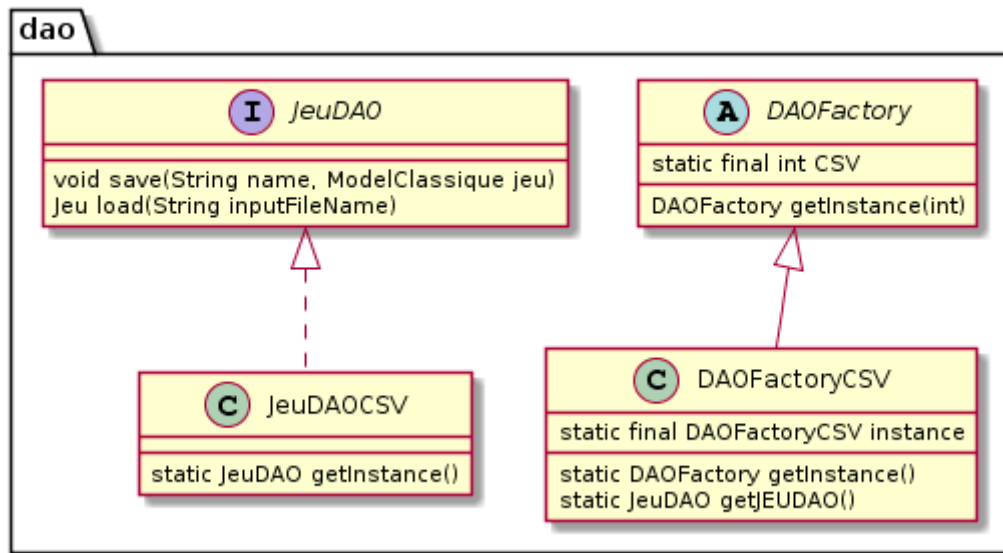
LEVY Damien
TRABELSI Illef
CHOFFÉ Damien

Nous avons fait une classe abstraite Jeu qui permet de définir un certain nombre de méthodes utiles au jeu. Cela permet de pouvoir créer plusieurs modes de jeu différents. Nous en avons créé un que nous avons nommé ModelClassique. Nous avons créé une classe abstraite Joueur qui définit toutes les méthodes communes à l'IA et au Joueur Humain.

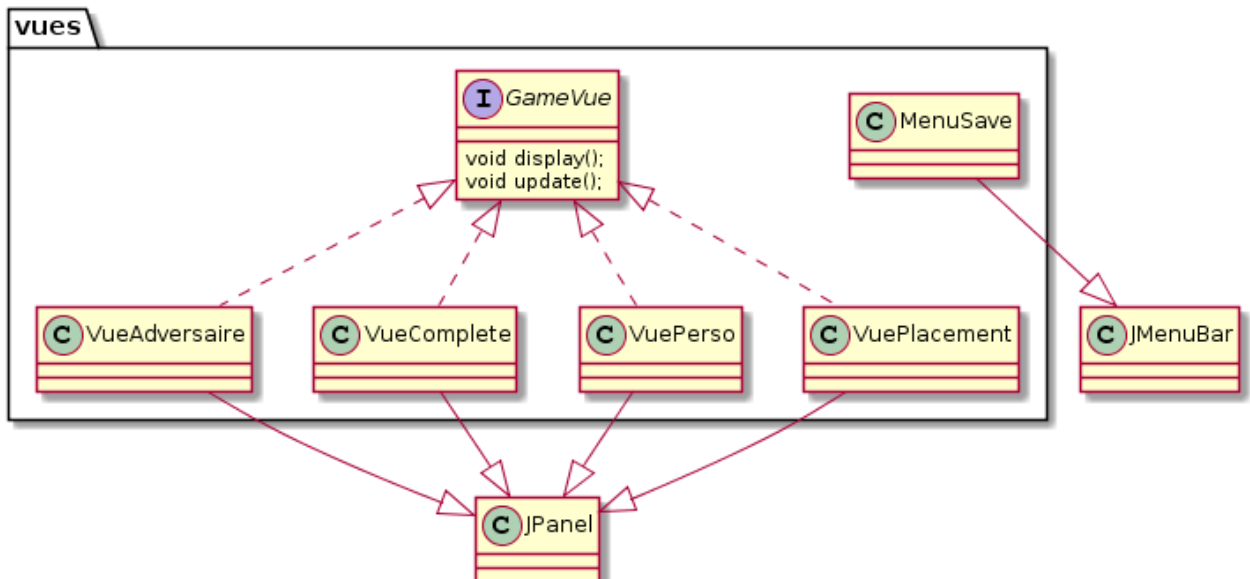


LEVY Damien
TRABELSI Ildefonso
CHOFFÉ Damien

Afin de pouvoir créer une sauvegarde et dans l'hypothèse où nous voudrions utiliser plusieurs formats de fichiers différents pour sauvegarder nous avons décidé d'utiliser le design patterns DAO.



Pour l'interface graphique, nous avons défini une interface afin que chacune des vues aient les mêmes méthodes. Cela nous a permis d'ajouter chacune des vues facilement et de pouvoir en ajouter d'autres si nous le voulions.



LEVY Damien
TRABELSI Ildefonso
CHOFFÉ Damien

