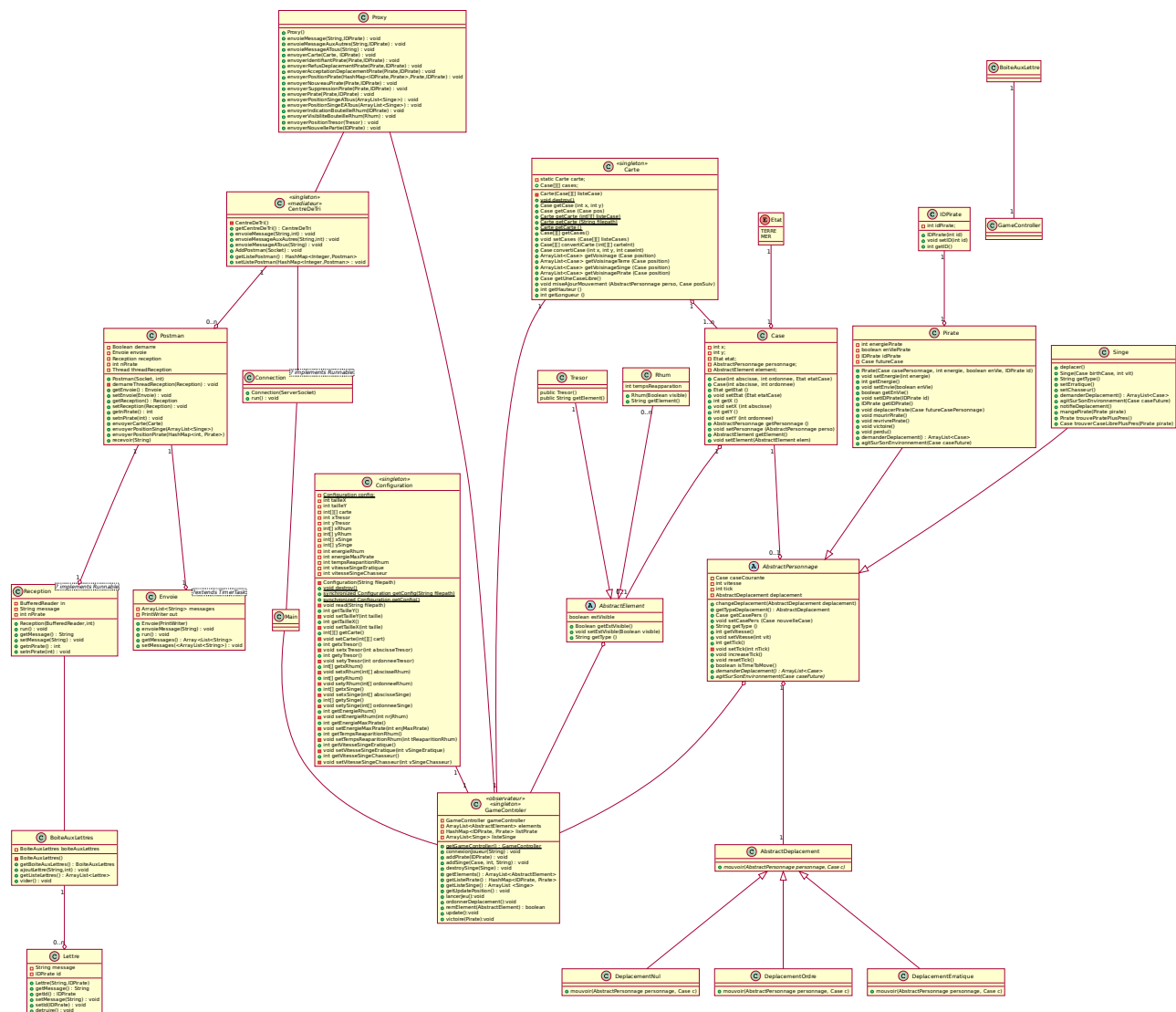


réalisé par:
 Regnaud Clément, Chaillou Clément, Njamfa Clémence, Sorlie Pierre-François

1. Diagramme de classe



2. Patron de conception Singleton

Nous avons choisi d'implémenter ce patron de conception sur les classes GameController, Carte, Config, et CentreDeTri.

Le GameController est la classe cerveau qui dirige l'instance de jeu, nous avons donc décidé d'appliquer ce patron de conception pour s'assurer d'en avoir qu'un seul.

De même, notre instance de jeu n'a qu'une et une seule Carte, le patron Singleton s'y applique également.

La classe Config s'occupe de charger les fichiers de configuration en début d'instance de jeu. L'instancier en plusieurs exemplaires n'est pas nécessaire, cette classe suit également le pattern Singleton.

Le CentreDeTri fait passer les informations du Proxy au Postman et nous avons jugé intéressant de n'en avoir qu'une instance.

3. Patron de conception Stratégie

Nous avons appliqué le patron de conception Stratégie sur la classe AbstractPersonnage et ses classes héritées. Selon la stratégie choisie, un Singe se comportera en tant qu'Erratique, ou en tant que Chasseur.

De même, ce patron permet de définir une stratégie de comportement pour un Pirate.

Si le pirate n'est pas ivre, il se déplace à la case indiquée par le joueur, comme un singe chasseur mais pour le singe c'est lui qui choisit sa case d'arrivée.

Si le pirate est ivre, il se déplace de manière aléatoire comme un singe erratique.

4. Patron de conception Patron de Méthode

Ce patron est complémentaire à notre implémentation du patron Stratégie. Les Pirates et les Singes sont tous des AbstractPersonnage et comportent des similitudes. Cependant, ils n'agissent pas de la même façon, notamment concernant leurs règles de déplacement et leurs interactions avec les cases et les éléments de la carte.

Pour ce faire, la stratégie de déplacement appelle les méthodes implémentées selon les types de personnage.

5. Patron de conception Médiateur

Nous avons appliqué le Patron de conception Médiateur sur la classe CentreDeTri. Ce patron sert à envoyer au Postman les informations utiles reçues par le Proxy. Cette classe implémente trois commandes : envoyer un message à un client choisi, envoyer un message à tous les clients sauf à un choisi, envoyer un message à tous les clients.

6. Patron de conception Observateur

Notre Classe GameController observe chaque Pirate afin de prévenir le proxy lorsqu'il se déplace via l'appel de la méthode notifieDéplacement (boolean acceptation). Nous avons pensé à faire la même chose sur les singes, mais ne l'avons pas implémenté dû à un manque de temps.

7. Patron de conception Fabrique Abstraite

Ce patron n'a pas été implémenté car nous l'avons découvert trop tard. La conception et l'implémentation avaient déjà été faites sans ce patron et donc le mettre en place aurait été coûteux en temps.

Ce patron aurait pu être implémenté pour fabriquer des AbstractPersonnages, tels que les Pirates ou les Singes.