

## TP 4 - HashSets

Resp. UE : Érik Martin-Dorel & Jean-Paul Bodeveix

Sujets P00: Christelle Chaudet & Jean-Paul Bodeveix

 Ce TP fait suite au TP 3. Vous devez donc avoir terminé le TP 3 pour passer au TP 4.

Pour rappel, les règles du jeu sont [ici](#).

On se place maintenant dans un package jeu gérant la boucle de jeu et le comportement des joueurs.

### Coups et Joueur

1. Définir la classe **Coup**. Un coup est défini par une carte et le joueur cible. Donner le constructeur et les accesseurs en lecture sur ces deux champs. On notera que la cible peut être **null**, ce qui signifie que le coup consiste à remettre la carte dans le sabot.
2. Définir dans la classe **ZoneDeJeu** la méthode **boolean estDepotAutorise(Carte carte)** renvoyant true dans les cas suivants:
  - a. si la carte est une borne : si le joueur n'est pas bloqué, la borne ne dépasse pas la vitesse limite et que la somme des bornes ne dépasse pas 1000
  - b. si la carte est une botte
  - c. si la carte est une limitation de vitesse : si le joueur n'a pas la botte prioritaire et n'a pas déjà une limite de vitesse,
  - d. si la carte est une fin de limite de vitesse : si le joueur n'a pas de botte prioritaire et a une vitesse limitée
  - e. si la carte est de type Bataille, on récupère dans la variable **top** le sommet de la pile de batailles du joueur. Si la pile est vide, **top** désignera le feu vert lorsque le véhicule est prioritaire ou que la carte jouée est le feu rouge, et le feu rouge sinon. On examine ensuite les cas suivants :
    - si **top** est une attaque et qu'il n'y a pas de botte du même type, la carte jouée peut être une parade du même type.
    - si **top** est une parade, une attaque peut être jouée si le joueur n'a pas déposé la botte correspondante.
3. Un dépôt étant autorisé dans la zone ciblée,, le joueur ne peut jouer une carte d'attaque ou une limitation de vitesse que sur un autre joueur sinon il ne peut la déposer que dans sa propre zone. Ajouter à la classe Coup la méthode **estValide(Joueur joueur)** ne renvoyant **true** que si ces conditions sont vérifiées.

4. Définir dans la classe **Joueur** la méthode **coupsPossibles(Set<Joueur> participants)** renvoyant l'ensemble de tous les coups valides. L'ensemble sera géré par un HashSet. Il faudra donc redéfinir les méthodes **equals** et **hashCode** de **Coup**. Pour obtenir cet ensemble, pour chaque participant, pour chaque carte que le joueur courant a en main, créer un coup et s'il est valide, l'ajouter à l'ensemble.
5. Définir dans la classe **Joueur** la méthode **coupsDefausse**. Cette méthode crée un coup pour chaque carte de la main avec une cible nulle, ce qui représente la remise de la carte dans le sabot.

### Jouer une carte

1. Ajouter à **ZoneDeJeu** puis à **Joueur** la méthode **boolean déposer (Carte c)** renvoyant **true** si la carte a pu être déposée dans la zone de jeu. On suppose que le dépôt est autorisé. Le traitement associé à chaque type de carte est décrit ci-dessous :
  - a. si la carte est une borne,, la carte est ajoutée aux bornes.
  - b. si la carte est une botte, elle est ajoutée aux bottes du joueur. Si elle répond à une attaque, l'attaque est supprimée.
  - c. si la carte est une limitation de vitesse,, la carte est ajoutée.
  - d. si la carte est une fin de limite de vitesse,, la carte est ajoutée.
  - e. si la carte est de type Bataille, la carte est ajoutée.
2. Ecrire des tests en faisant varier le contenu de la zone du joueur et la carte appliquée. On produira par exemple la trace suivante :

```
déposer feu rouge : dépôt ok ? true, bloqué ? true
déposer accident : dépôt ok ? false, bloqué ? true
déposer as du volant : dépôt ok ? true, bloqué ? true
déposer panne essence : dépôt ok ? false, bloqué ? true
déposer essence : dépôt ok ? false, bloqué ? true
déposer feu vert : dépôt ok ? true, bloqué ? false
déposer 100 : dépôt ok ? true, bloqué ? false
déposer limite: dépôt ok ? true, bloqué ? false
déposer 100 : dépôt ok ? false, bloqué ? false
déposer 25 : dépôt ok ? true, bloqué ? false
déposer fin limite : dépôt ok ? true, bloqué ? false
déposer 100 : dépôt ok ? true, bloqué ? false
```

3. Ajouter au package jeu la classe **Sabot** implantée comme un tableau de cartes dont la taille est fournie par le constructeur. Il devra être possible d'ajouter une carte isolée (**ajouterCarte**) ou une famille de cartes (**ajouterFamilleCarte**) en prenant en compte le nombre de cartes d'un même type. Définir la méthode **piocher()** qui renvoie et supprime la première carte du sabot.
4. Ajouter au package **jeu** la classe **Jeu**. Le jeu comprend l'ensemble des joueurs et un sabot de 106 cartes. Ajouter la méthode **inscrire(Joueur jour)**. Ajouter la méthode

**remplirSabot(JeuDeCartes jeu)** qui remplit le sabot à partir d'un jeu passé en paramètre. Le nombre de cartes du sabot sera la taille du jeu de carte

5. Toujours dans la classe **Jeu**, définir la méthode **distribuerCartes()**. Donner à tour de rôle une carte à chaque joueur jusqu'à ce qu'ils aient chacun **NBCARTES=6** cartes.
6. Dans la classe **Joueur** ajouter la méthode **retirerDeLaMain(Carte carte)** qui enlève la carte demandée de sa main..

### Un tour de jeu

1. Définir dans la classe **Jeu** la méthode **jouerTour()** qui fait jouer une fois chaque participant. Un joueur prend une carte, sélectionne un coup ou si ce n'est pas possible une carte à rendre. On affichera la carte piochée et le coup joué après avoir redéfini la méthode **toString()** de Coup.
2. Définir une classe **TestJeu** qui crée un jeu, crée et inscrit 2 joueurs, distribue les cartes, affiche la main de chaque joueur et réalise un tour de jeu. On obtiendra par exemple la trace suivante:

Le joueur J1 a en main: [Feu Vert, 50KM, Feu Rouge, Feu Vert, 50KM, Réparation]

Le joueur J2 a en main: [Feu Vert, panne d'essence, Increvable, 25KM, 200KM, 100KM]

Le joueur J1 a pioche 100KM

Le joueur J1 a choisi le coup "déposer la carte Feu Rouge dans la zone de jeu de J2"

Le joueur J2 a pioche 50KM

Le joueur J2 a choisi le coup "déposer la carte Feu Vert dans la zone de jeu de J2"