

Projet Info2A : Kartel

Introduction

Kartel est un jeu de société développé par Reiner Knizia et édité par la société suisse Helvetiq : <https://www.helvetiq.com/kartel>

Dans ce jeu, vous incarnez un détective qui enquête sur les 7 gangs qui terrorisent la ville. Son objectif est de mettre en prison les membres des différents gangs et parfois (pas trop souvent) d'accepter les pots-de-vin. Mais attention, il faudra choisir judicieusement vos alliés (car un Boss en prison refusera de payer ses pots-de-vin) et vos ennemis (en effet, s'ils parviennent à s'enfuir, les Boss sauront venger leur gangsters incarcérés).

>>>> Les règles du jeu expliquées en vidéo : <https://youtu.be/b5Vi8tK56GI> <<<<

I) Règles du jeu :

La particularité de ce jeu est que tous les **Joueur** avancent avec le même pion représentant le **Detective**. Votre objectif est de faire plus de points que vos adversaires en mettant dans votre **Reserve** les **Gangster** dont le **Boss** finira en **Prison** et en acceptant les **PotDeVin** de ceux qui parviendront à s'enfuir.

a) Début de la partie :

Le **Plateau** est construit en mélangeant les 42 **Jeton** qui le composent.

Chaque **Jeton** est associé à l'un des 7 gangs (rouge, bleu, vert, jaune, rose, orange, et violet).

Il y a 6 **Jeton** différents par gang :

- 1 **Boss**
- 1 **Gangster** unique
- 2 duo de **Gangster**
- 1 trio de **Gangster**
- 1 **PotDeVin**

b) Description d'un tour de jeu :

- Le **Joueur** lance un **De** à 6 faces dont les faces sont 2, 2, 3, 3, 4, et 4.
- Le **Joueur** décide de combien de **Jeton** il veut avancer le **Detective** (le nombre maximum étant celui indiqué par le **De**).
- Le **Jeton** sur lequel le **Detective** atterrit est retiré du **Plateau** :
 - Si ce **Jeton** est un **Boss**, celui-ci va dans la **Prison**
 - Sinon il va dans la **Reserve** du **Joueur**
- C'est au tour du **Joueur** suivant

c) Fin du jeu et calcul des scores :

La partie se termine lorsque 5 des 7 **Boss** sont en **Prison**. Le joueur ayant le plus de points remporte la partie. La valeur de chaque **Jeton** ramassé par un Joueur dépend de l'état du **Boss** de son gang

État du Boss	Gangster unique	Duo de Gangster	Trio de Gangster	PotDeVin
Libre	-1	-2	-3	+3
En Prison	+1	+2	+3	0

II) Conception du projet :

Vous devrez créer un projet Java qui permet de jouer à Kartel. Votre projet doit tirer parti de la Programmation Orientée Objet pour séparer les différents aspects du jeu. Vous êtes libre de construire le projet comme il vous plaît mais pour vous aider une liste de classes vous est proposée :

- **Jeton** qui est associé à un gang et qui a pour sous-classes : **Boss**, **Gangster**, et **PotDeVin**
- **Gangster** qui a une taille d'équipe associée (1, 2, ou 3)
- **Prison** dans laquelle vont les **Boss** arrêtés
- **Plateau** qui contient l'ensemble des cases au début de la **Partie**, ainsi que le **Detective**
- **Detective** qui a une position et la possibilité de se déplacer
- **Joueur** qui contient son nom et sa **Reserve**
- **Reserve** où est stockée les **Jeton** ramassés par le **Joueur** et qui calcule son score
- **De** qui possède une liste de faces dont la face courante, ainsi qu'une méthode de lancer
- **Partie** qui contient l'ensemble des **Joueur**, le **Plateau**, la **Prison**, et le **De**

Les fonctionnalités requises sont :

- L'affichage de l'état de la **Partie**, c'est à dire l'état du **Plateau** (avec les **Jeton** restants et la position du **Detective**) celui de la **Prison** (liste des **Boss** incarcérés) et celui de chacun des **Joueur** (nom du joueur et contenu de sa **Reserve**). Pensez à déléguer l'affichage aux différentes composantes. Par exemple :

```
*****
2C - 2A - 2E - 2B - 2F - 3C - $A - [B] - [D] - 3G - [A] - 1C - <> - 1G - [E] - 2E -
Prison : [F] - [G] - [C] -
Alice(-8) : 2C - $D - 3D - 1F - 3E - 2D - 3B - 3A -
Bob(1) : $E - 2F - 2D - 1B - 2B - 2G - 1E -
Charles(4) : 1A - $B - $G - 1D - $C - 2A - 3F - $F - 2G -
C'est à Alice de jouer
Résultat du dé : 2, que jouez-vous ? : □
```

*Le **Detective** est représenté par <>, le **Boss** du gang X par [X],
ses **Gangster** (en équipe de taille N) par NX et son **PotDeVin** par \$X*

- Le paramétrage de la **Partie** : permettre à l'utilisateur de rentrer le nom des différents **Joueur** puis de lancer la **Partie**
- La gestion d'un tour de jeu : lancer du **De**, avancée du **Detective** (attention à bien revenir au début du **Plateau** une fois la fin dépassée), ramassage de la **Jeton** sur lequel il atterrit
- la fin de la **Partie** : vérification du nombre de **Boss** en **Prison**, calcul du score de chaque **Joueur** et affichage du gagnant

Si toutes les fonctionnalités ci-dessus ont été implémentées, vous pouvez en rajouter selon votre envie. Voici quelques suggestions mais vous êtes libre de faire vos propres propositions :

- lors de l'affichage d'un **Joueur**, afficher son score actuel (par rapport à une **Prison** incomplète)
- à la fin, proposer de rejouer avec les mêmes **Joueur** ou de recréer une nouvelle **Partie**
- ajout de nouveaux types de **Jeton** (selon votre imagination)
- possibilité de paramétrer complètement la partie (nombres de gangs, nombres et types de **Jeton** par gang)

Et si vous avez vraiment du temps à perdre, la création d'une "Intelligence Artificielle" :

- d'abord aléatoire
- puis qui joue selon le meilleur coup (celui qui rapporte le plus de points) "en l'état"
- et pour finir qui "anticipe" les prochains coups

III) Modalités d'évaluation :

Le projet est à faire en binôme et se doit d'être original. Vous pouvez vous inspirer d'autres binômes pour différentes idées de mise en œuvre mais chaque ligne de code devra avoir été écrite par vos soins ; il pourra vous être demandé d'expliquer certaines parties du code.

Dans un premier temps, vous rendrez un pré-rapport au format PDF. Ce pré-rapport devra contenir un *diagramme de classes*, c'est-à-dire un schéma qui montre les relations entre les différentes classes ainsi que les attributs principaux de chacune d'entre-elles. Il contiendra également un *descriptif de chaque classe* pour expliquer le choix des attributs et des méthodes, et vous détaillerez les fonctionnalités les plus complexes. Il n'est pas nécessaire d'avoir commencé à programmer pour écrire ce pré-rapport : il porte sur ce que vous comptez faire et comment mais il ne vous engage en rien sur la suite du projet. Vous remettrez ce pré-rapport à votre chargé de TP qui vous rendra un avis constructif pour vous permettre d'éviter différents problèmes de conception.

Une fois le projet terminé, il faudra le présenter à votre chargé de TP lors d'une démonstration. Celle-ci dure 20 minutes lors desquelles vous décrirez votre projet (structures et principales fonctionnalités) et où vous répondrez aux questions. Cette démonstration doit avoir été préparée au préalable, c'est-à-dire :

- votre projet se doit de fonctionner sur l'ordinateur de la démonstration, ne le transférez pas au dernier moment sur un autre ordinateur ce qui pourrait entraîner des problèmes de compatibilité
- vous avez préparé à l'avance un jeu d'essai pour montrer les différentes fonctionnalités
- ne perdez pas de temps à expliquer des détails techniques, concentrez vous sur la structure globale de votre projet et les fonctionnalités développées.
- ayez un code organisé (chaque classe dans un fichier séparé), bien structuré (indentation propre), et commenté sur les fonctions les plus complexes. Si une question vous est posée sur une fonctionnalité particulière vous devrez être capables d'aller rapidement au bon endroit dans le code source et de l'expliquer
- les deux membres du binôme devront prendre la parole lors de la démonstration de manière égale. Soyez cohérents dans votre discours (en ayant préparé à l'avance les points importants à mettre en avant) et dans la répartition de la parole (chacun présente la partie qu'il a personnellement programmée)

Suite à la démonstration, vous remettrez le code source de votre projet (sous format ZIP) ainsi qu'un rapport (semblable au pré-rapport mais en plus détaillé et représentant le projet tel que réalisé). Vous pourrez également rendre des documents supplémentaires si leur communication vous semble importante.

Dates importantes :

- Semaine du 23-27 mars : rendu du pré-rapport
- Semaine du 6-10 avril : remise des corrections du pré-rapport
- Semaine du 4-7 mai : démonstration du projet