Projet Kartel

Agustín Cartaya

Objectif principal :

- Créer le jeu de Kartel pour PC sous Java

Objectifs spécifiques :

- Créer une structure de classes pour modéliser le projet
- Utilisez java et la programmation et orientée aux objet (POO) pour créer le jeu
- Créer une intelligence artificielle pour le jeu

Ressources utilisées :

- Langage de programmation : JAVA, version 1.8.0-241
- IDE : NET BEANS 8.2

Structure du projet :

Le projet Kartel compte actuellement avec 16 classes publiques, dont 2 classes abstraites et une classe principale avec la méthode principale, et dispose également de 2 interfaces (Showable, Reutilisable). Le schéma graphique UML des relations et des héritages des classes et des interfaces peut être consulté dans le fichier KartelUML.pdf

Liste des classes (c), classes abstraites (ca), interfaces (i), classe principale (cm):

Boss (c)
De (c)
Detective (c)
Gang (c)
GangMember (ca)
Gangster (c)
Joueur (c)
KartelIntelligence (c)
LesGangs (c)
LesJoueurs (c)
Lire (c)
Partie (cm)
Plateau (c)
PotDeVin (c)
Prison (c)
Recompense (ca)
Reutilisable (i)
Showable (i)

Spécifications détaillées de classe :

GangMembre

En-tête

classe abstraite publique GangMember implements Showable

Définition

Cette classe est responsable de la modélisation de tous les membres de gangs [un membre d'un gang est tout ce qui appartient à un gang indépendamment de sa forme ou son statuts. Un gang peut avoir beaucoup de membres, mais un membre ne peut appartenir à un seul gang, un membre d'un gang est immuable, c'est-à-dire qu'il ne peut pas changer de type de membre] parmi eux les Boss, Gangsters et les PotsDeVin

Attributs

String type
définit le type de membre, par exemple "BOSS", "GANGSTER" ou "POT DE VIN"

Gang gang
fait référence à la bande à laquelle ce membre appartient

Méthodes

Constructeur

GangMember public (Gang gang, String type)
Initialise le GangMembre appartenant à un gang et avec son type passé par paramètre

Getters et Setters

public Gang getGang ()
Renvoie la référence à l'objet de gang qui appartient

public String getType()
Retourne le type de gang membre

Boss

En-tête

public class Boss extends GangMember

Définition

Cette classe est responsable de la modélisation de tous les Boss d'un gang [le Boss d'un gang est un membre de cela avec la qualité qu'il est le Boss et en fonction de son état de liberté les autres membres peuvent changer leurs valeurs]

Attributs

boolean prisonnier
Définit le statut de Boss vrai si elle est en prison

public static final String TYPE = "BOSS"
Chaîne de texte avec le type de membre de gang réglé à BOSS

Méthodes

Constructeur

public Boss (Gang gang)
Initialise le Boss en appelant le constructeur de sa classe parent (GangMembre) en le passant par le paramètre le gang et le type. Par défaut, le Boss n'est pas un prisonnier

Getters et Setters

boolean public isPrisonnier()
Renvoie si le Boss est un prisonnier ou non

public void setPrisonnier(boolean prisonnier)
Définit si le Boss est un prisonnier ou non

Override

public String toString()
Retourne une chaîne de texte avec le mot "Boss" et le nom de gang

public String show()
Retourne la représentation du Boss pour le jeu (abréviation de gang enfermée dans les parenthèses carrées)

Récompense

En-tête
Public abstract class Recompense extends GangMember

Définition
Cette classe est responsable de la modélisation de tous les membres qui peuvent avoir une récompense [une récompense est une valeur que le membre peut contribuer au joueur qui l'a, la récompense varie en fonction de l'état de liberté du Boss de gang]

Attributs
int valeurBossPrisonnier
Valeur de la récompense si le Boss de gang est en prison

int valeurBossNonPrisonnier
Valeur de la récompense si le Boss de gang n'est PAS en prison

Méthodes
Constructeur
public Recompense(Gang gang, String type, int valeurBossPrisonnier, int valeurBossNonPrisonnier)

Initialise la Récompense en appelant le constructeur de sa classe parent (GangMembre) en le passant par paramètre le gang et le type, et initialise les différentes valeurs de récompense avec les paramètres correspondants

Autres
public int getRecompense()
Retourne la valeur de la récompense selon que le statut du Boss de gang

Getters et Setters
public int getValeurBossPrisonnier()
Retourne la valeur de la récompense si le Boss de gang est en prison

public void setValeurBossPrisonnier(int valeurBossPrisonnier)
Etablit si la valeur de la récompense si le Boss de gang est en prison

public int getValeurBossNonPrisonnier()
Retourne la valeur de la récompense si le Boss de gang n'est pas en prison

public void setValeurBossNonPrisonnier(int valeurBossPrisonnier)
Stable si la valeur de la récompense si le Boss de gang n'est pas en prison

Gangster

En-tête
public class Gangster extends Recompense

Définition
Cette classe est responsable de la modélisation de tous les gangsters de gangs [un gangster est un membre d'un gang qui a un niveau et en fonction de ce niveau a une valeur de récompense]

Attributs
int niveau;
contient le niveau du gangster

public static final String TYPE = "GANGSTER"
chaîne de texte avec le type de membre de gang réglé à GANGSTER

Méthodes
Constructeur
public Gangster(Gang gang, int niveau)
initialise le Gangster en appelant le constructeur de sa classe parent (Recompense) en le passant par paramètre le gang, le type et les valeurs des récompenses en fonction de son niveau

Getters et Setters
public int getNiveau ()
Retourne le niveau du gangster

public void setNiveau(int niveau)
Etablit le niveau gangster

Override
public String toString()
Retourne une chaîne de texte avec le mot "Gangster", sa valeur et le nom du gang

public String show()
Retourne la représentation du Gangster pour le jeu (le niveau le plus abrégé de la bande)

PotDeVin

En-tête
public class PotDeVin extends Recompense

Définition
Cette classe est responsable de la modélisation de tous les PotDeVins dans un gang [un PotDeVIn est un membre d'un gang qui a une valeur qui fixe sa récompense]

Attributs
public static final String TYPE = "POT DE VIN"
chaîne de texte avec le type de membre de gang réglé à POT VIN

Méthodes

Constructeur
public PotDeVin (Gang gang, int récompense)
Initialise le PotDeVin en appelant le constructeur de sa classe parent (Récompense) en le passant par paramètre le gang, le type et les valeurs de récompense en fonction de leur valeur

public PotDeVin (Gang gang)
Appelle le constructeur précédent en le passant par le paramètre le gang et 3 comme une récompense

Override
public String toString()
Retourne une chaîne de texte avec les mots "Pot de vin", sa valeur (si le Boss n'est pas en prison) et le nom du gang

public String show()
Retourne la représentation de potDeVin pour le jeu ($ plus d'abréviation de gang)

Gang

En-tête

public class Gang implements Reutilisable

Définition

Cette classe est responsable de la modélisation de tous les gangs dans le jeu [un gang est un groupe de GangMembres avec un nom et une abréviation]

Attributs

String nom
Nom de gang

String abb
Pseudonyme de gang

ArrayList<GangMember> gangMembers
Liste qui contiendra tous les membres de gangs

Méthodes

Constructeur

public Gangster(Gang, int niveau)
Initialise le gang avec un nom et une abréviation, puis initialise la liste avec le Boss de gang ajouté en tant que premier membre

Autres

initDefaultGangMembres()
Ajoute 4 Gangster et 1 PotDeVin à la liste des membres

public void bossInPrision(boolean etat)
Méthode utilise pour changer l'état de liberté du Boss

public boolean isBossInPrison()
Retourne si le Boss de gang est en prison ou non

public void addGangMember(GangMember g)
Ajoute un nouveau membre à la bande, cela ne peut pas être un Boss

public Boss getBoss ()
Ren retour au chef de gang

public ArrayList<Gangster> getGangsters()
retourne une liste de tous les gangsters dans le gang

public ArrayList<Gangster> PotDeVin()
retourne une liste de tous les PotDeVin dans le gang

Getters et Setters
public ArrayList<GangMember> getGangMembresList()
Retourne la liste avec tous les membres de gangs

public String getNom()
Retourne le nom du gang

Chaîne publique getAbb ()
Retourne l'abréviation des gangs

Override
public void reset()
Réinitialise le gang pour l'utiliser à nouveau, ce qu'il fait est de dire au Boss que non est plus en prison

public String toString()
Retourne une chaîne de texte avec le nom, l'abréviation et tous les membres de gangs

public int hashCode()
Modifie le hasCode de l'instance, le calculant en ce qui concerne le pseudonyme du gang

public boolean equals(Object obj)
compare deux gangs pour voir s'ils sont les mêmes [deux gangs sont les mêmes si c'est le même gang ou s'ils ont le même pseudonyme]

LesGangs

En-tête
public class LesGangs

Définition
Cette classe est responsable de la modélisation d'un groupe de gangs avec les méthodes nécessaires pour leur traitement le plus confortable dans d'autres classes

Attributs
ArrayList<Gang> gangs
Liste qui contiendra tous les gangs

Méthodes
Constructeur
public LesGangs ()
initialise le groupe avec une liste de gangs vides

Autres
public boolean addGang(Gang g)
Ajoute un gang à la liste s'il n'est pas déjà ajouté et renvoie true s'il était possible de l'ajouter

public int addGangs (LesGangs gs)
Ajouter un groupe gangs à la liste et retourner le nombre d'agrégats

public Gang getGang (indice int)
Renvoie le gang qui se trouve à la position de la liste indiquée par paramètre, si la position n'est pas trouvée renvoie null

public Gang getGang (String abb)
Retourne le Gang de la liste des pseudonymes abb, s'il n'existe pas de renvoie null

public int getNbGangs ()
Retourne le nombre de gangs dans la liste

private boolean gangRepete(Gang g)
Vérifie si le gang existe sur la liste

public void resetGangs()
Fait l'appel de la méthode de réinitialisation de tous les gangs dans la liste

public ArrayList<GangMember> getGangsMembers()
Renvoie une liste de tous les membres de tous les gangs dans la liste

Getters et Setters
ArrayList<Gang> getGangs()
Renvoie la liste de tous les gangs

Override
public String toString()
Un String avec tous les gangs et leurs spécifications

Joueur

En-tête
public class Joueur implements Comparable, Reutilisable, Showable

Définition
Cette classe est responsable de la modélisation des joueurs, ils auront un nom et un groupe de récompenses à partir de laquelle leur score sera calculé.

Chaque fois qu'un joueur gagne un jeu cela ajoute un point aux jeux gagnés

Un joueur peut être automatique [Un joueur est appelé automatique quand il est kartelIntelligence intelligence qui le contrôle] ou peut avoir une intelligence du niveau désiré et avec un volume désiré

Attributs
String nom
Nom du joueur

ArrayList<Recompense> recompenses
Liste qui aura les récompenses du joueur dans le jeu

int jeuxGagnes
Nombre de jeux gagnés

int iNiveau
Niveau d'intelligence de joueur

int iVolumen
Niveau de volume d'intelligence de joueur

Méthodes
Constructeur
public void addJeuGange()
initialise un joueur avec un nom vide et une liste de récompenses, les jeux gagnés par défaut sont nuls, le joueur n'est pas automatique et le volume et le niveau d'intelligence est égal à zéro

Autres
vide public addJeuGange ()
Ajoute au joueur un autre jeu gagné

public void addRecompense(Recompense recompense)
Ajouter la récompense à la liste de récompenses du joueur

public int getScore ()
Retourne le score du joueur [l'escore est la somme de la valeur de toutes les récompenses du joueur a]

Getters et Setters
public ArrayList<Recompense> getRecompenses()
Retourne la liste de récompenses du joueur

public String getNom()
Retourne le nom du joueur

public int getJeuxGagnes ()
Retourne le nombre de matchs gagnés

boolean public isAutomatique ()
Retourne true si le joueur est automatique

public void setAutomatique(boolean automatique)
Définit si le joueur est automatique

public int getiNiveau ()
Retourne le niveau d'intelligence des joueurs

public int setiNiveau (int iNiveau)
Définit le niveau d'intelligence des joueurs
public int getiVolumen ()
Retourne le niveau de volume de l'intelligence du joueur

public int setiVolumen (int iVolumen)
Définit le niveau de volume de l'intelligence du joueur

Override
public String toString()
Retourne une chaîne de texte avec le nom et le score du joueur

public int compareTo (Object o)
Comparez deux joueurs à leur score, cette méthode est utilisée pour trier les joueurs à la fin du jeu et obtenir le gagnant

public void reset()
Réinitialise le joueur pour l'utiliser à nouveau, ce qu'il fait est de supprimer toutes les récompenses de la liste

public String show()
Retourne une chaîne avec le texte du joueur affiché dans le jeu, ce texte contient son nom, son score et sa liste de récompenses

public int hashCode()
Modifie le hasCode de l'instance, en le calculant en ce qui concerne le nom du joueur

public boolean equals(Object obj)
Compare les joueurs pour voir s'ils sont les mêmes [deux joueurs sont les mêmes s'ils sont le même joueur ou s'ils ont le même nom]

LesJoueurs

En-tête
Public class LesJoueurs

Définition
Cette classe est responsable de la modélisation d'un groupe de joueurs avec les méthodes nécessaires pour leur traitement le plus confortable dans d'autres classes

Attributs
private ArrayList<Joueur> joueurs
Liste qui contiendra tous les joueurs

private int indexJoueurActuel
index pour se référer au joueur actuel dans la liste [Le joueur actuel se réfère au joueur qui a le tour de jeu]

Méthodes
Constructeur
public LesJoueurs ()
Initialise le groupe de joueurs avec une liste vide

Autres
vide public nextJoueur ()
Changer l'index du joueur actuel pour le suivant dans la liste, si l'indice du joueur actuel est le dernier, le prochain sera zéro

public String afficherPositions (int ctt)
Retourne une chaîne qui montrera toutes les positions des joueurs dans la liste,
La chaîne est calculée en triant la liste des joueurs dans l'ordre décroissant par rapport à leur score, puis ajouté avec leur position par rapport aux joueurs, s'il ya deux joueurs répétitifs ceux-ci apparaîtront dans la même position

public ArrayList<Joueur> getListResultats()
Retourne une liste de joueurs triés dans l'ordre décroissant en ce qui concerne leur score

public boolean addJoueur (Joueur j)
Ajoute un joueur à la liste des joueurs si le joueur n'est pas répété, retourne vrai si le joueur a été ajouté

public int getNbJoueurs ()
Retourne le nombre de joueurs sur la liste

public Joueur getJoueur (index int)
Renvoie le gang de la position indiquée dans la liste, si cette position n'existe pas de déclarations nulles

public int getIndexJoueur (Joueur j)
Retourne la position dans la liste du joueur passée par paramètre

boolean public remove (index int)
Supprime le joueur de la position indiquée par paramètre s'il existe, si tous les joueurs sont supprimés le joueur actuel devient -1, retourne vrai si le joueur a été éliminé

boolean public remove (Nom string)
Supprime le joueur dont le nom est passé par paramètre, retourne vrai si le joueur a été éliminé

private boolean joueurRepete (Joueur j)
Vérifiez si le joueur existe sur la liste

public Joueur getJoueurActuel()
Retourne le joueur actuel

public Joueur getJoueurGagnant()
Retourne le premier joueur avec le score le plus élevé, (TEST SEULEMENT)

public ArrayList-lt;Joueur-gt; getGagnant()
Retourne une liste de joueurs ordonnant aux joueurs par leur score (NON UTILISE)

public void removeAll()
Supprimer tous les joueurs de la liste

public void resetJoueurs()
Appelle la méthode de réinitialisation de tous les joueurs sur la liste

public String showJoueurs ()
Renvoi une chaine de texte en faisant appel au méthode show de chaque joueur

Getters et Setters
public ArrayList<Joueur> getJoueurs()
retourne la liste des joueurs

public int getIndexJoueurActuel()
Retourne la position du joueur actuel

public void setIndexJoueurActuel(int index)
Définit la position du joueur actuel si cette position est autorisée

Override
public String toString()
Retourne une chaîne avec tous les joueurs et leurs spécifications

Détective

En-tête

public class Detective implements Showable

Définition

Cette classe est responsable de la modélisation du détective de jeu [le détective est le jeton qui déplace tous les joueurs sur la plaque de jeu pour capturer les membres du groupe]

Attributs

int position
Position du détective dans le jeu

int max
Position maximale que le détective peut atteindre

Méthodes

Constructeurs

public Detective(int max)
initialise le détective avec la position maximale donnée par paramètre et la position égale à zéro

Autres

public void avancer(int deplacement)
Méthode qui fait avancer le détective au conseil d'administration en tenant compte du retour au poste de départ si la position maximale est dépassée

Getters et Setters

public int getPosition()
Retourne la position actuelle du détective

public void setPosition(int pos)
Établit la position actuelle du détective si elle est autorisée

public int getMax ()
Retourne la position maximale que le détective peut atteindre

public void setMax (int max)
Définit la position maximale que le détective peut atteindre

Override
public String toString()
Retourne une chaîne de texte avec le mot avec les spécifications du détective contenied sa position actuelle et sa position maximale

public String show()
retourne une chaîne de texte avec le texte affiché pour représenter le détective dans le jeu [le détective est représenté par "<>"]

De

En-tête
public class De

Définition
Cette classe est responsable de la modélisation du De du jeu

Attributs
final int []faces
Array d'entiers qui contiendra tous les faces du De

int indexFaceCourante
Indique que la position dans l'Array de faces la visage qui sera affichée

Méthodes
Constructeurs
public De (int [] visages)
Initialise les dés avec les visages passés par paramètre, et définit l'indice de la face actuelle à zéro

public De()
Appelle le constructeur précédent en lui passant un tableau de visage (2,2,3,3,4,4)

Autres
public int lancer()
Choisit au hasard l'un des visages de mourir et le retourne, et définit l'index de ce visage comme l'indice actuelle

Getters et Setters
public int getFaceCourante()
retourne le visage actuel

public int getIndexFaceCourante()
Retourne l'indice de la face actuelle

public void setIndexFaceCourante(int indexFaceCourante)
définit l'indice de la face actuelle

Override
public String toString()
Retourne une chaîne de texte avec le mot "De" qui contient également le nombre de visages et tous leurs visages

Prison

En-tête

public class Prison implements Reutilisable, Showable

Définition

Cette classe est responsable de la modélisation de la prison du jeu [la prison est l'endroit où les Bosss vont après avoir été capturé par le détective]

Attributs

ArrayList<Boss> bosses
Liste qui contiendra tous les Bosss de la prison

int maxPrisonniers
Nombre maximum de prisonniers autorisés en prison

Méthodes

Constructeurs

public Prison(int maxPrisonniers)
initialise la prison avec un nombre maximum de prisonniers et la liste des prisonniers vides

Autres

public boolean addBoss (Boss boss)
Ajouter un boss à la prison si elle n'est pas complète et revient fausse si la prison est complète

public int getNbBosses ()
Retourne le nombre de prisonniers dans la prison

boolean public isPrisonComplete()
Rendement vrai si le nombre de Bosss dans la liste égale au maximum accepté par la prison

public Boss getBoss (indice int)
Retourne le Boss avec l'indice regardé sur la liste des prisonniers

Getters et Setters

public int getMaxPrisoniers()
renvoie le nombre maximum de prisonniers acceptés par la prison

public void setMaxPrisonniers(int maxPrisonniers)
établit le nombre maximum de prisonniers acceptés par la prison

public ArrayList<Boss> getBosses()
Retourne la liste du boss en prison

Override
public void reset()

Réinitialise la prison pour l'utiliser à nouveau, ce qu'elle fait est de supprimer en disant aux Bosss qu'ils ne sont plus prisonniers et les retirer de la liste

public String toString()
Renvoie une chaîne de texte avec le mot avec les caractéristiques de la prison, y compris le nombre maximum de prisonniers acceptés et tous les Bosss qu'il contient

public String show()
Retourne le texte à afficher de la prison dans le jeu.

Plateau

En-tête

public class Plateau implements Reutilisable, Showable

Définition

Cette classe est responsable de la modélisation du plateau de jeu, il contiendra tous les jetons (membres de tous les gangs), la prison et le détective

Attributs

Gangs LesGangs
Groupe de gangs qui va être dans le jeu

ArrayList <GangMember> jetons
Liste des jetons qui seront présents sur le plateau

Detective detective
Détective de jeu

Prison prison
Prison du jeu

Méthodes

Constructeurs

public Plateau(LesGangs gangs, int maxPrisonniers)
initialise le conseil avec un groupe de gans et autant de prisonniers que possible dans la prison, si la liste des gangs n'est pas vide, il crée les jetons, puis initialise le détective

public Plateau(LesGangs gangs)
Appelle le constructeur antérieur lui passant le groupe Gangs et le maximum de prisonniers comme 5

Autres

public void setGangs(LesGangs gangs)
supprime le groupe Gang actuel et définit le passage par paramètre, puis crée les Jetons

private void createJetons()
Crée les jetons du plateau en les obtenant du groupe gangs, puis mélange-les pour les mettre au hasard et définissez les paramètres du détective comme leur position actuelle et leur position maximale

public int getNbJetons()
Retourne la quantité de jetons présents sur le tableau

public GangMember next(int pas)
Avance le détective sur le plateau à une certaine position sur la liste jeton et supprime le jeton de cette position, retourne le jeton éliminé et définit les nouveaux paramètres du détective

public void melanger(List<GangMember> list)
Utilisez la classe Collections pour mélanger la liste des jetons à l'aide de la classe Random

Getters et Setters
public ArrayList<GangMember> getJetons()
Retourne la liste des jetons qui sont sur le tableau

public Detective getDetective()
Retournez le détective présent sur le tableau

public Prison getPrison()
Retourner la prison du jeu

public LesGangs getgangs()
Retourne le groupe gangs présents dans le jeu

Override
public String toString()
Retourne une chaîne de texte qui modélise le plateau

public String show()
Retourne le texte qui représentait le plateau dans le jeu, celui-ci contient la représentation des Jetons et de la prison

public void reset()

Réinitialisez le plateau pour l'utiliser à nouveau en créant à nouveau les Jetons et en réinitialisant la prison

Partie

En-tête
public class Plateau implements Reutilisable

Définition
Il s'agit de la classe principale du jeu contenant la méthode principale et le menu qui vous permet d'interagir avec le ou les utilisateurs, plus il contient une instance de KerteIntelligence qui vous permet d'aider à jouer les joueurs ou jouer contre elle

Attributs
Plateau
Plateau du jeu

Joueurs LesJoueurs
Groupe de joueurs présents dans le jeu

De de
De du jeu

KartelIntelligence intelligence
Intelligence présente dans le jeu

boolean rejouer = false
true après avoir joué la première fois pour activer l'option rejouer

Méthodes
Constructeurs

public Partie()
Initialise le jeu, initialise le groupe de joueurs vides, un conseil et une intelligence, ainsi que le démarrage d'un décès par défaut et de faire l'appel de menu

Autres
public void menu()
Crée le menu de jeu et permet à l'utilisateur de choisir les différentes options qu'il contient pour paramétriser le jeu, faire des statistiques ou commencer à jouer

private boolean ajouterJoueur(String nom, boolean automatique, int iNiveau, int iVolimen)

Permet de créer un nouveau joueur avec un nom, et de définir les paramètres de votre intelligence, tels que votre niveau, et le volume, vous permet également de définir si oui ou non le joueur sera contrôlé par l'intelligence, puis l'ajoute à la liste

private void setGangs(LesGangs gangs, int maxP)
Établit le groupe de gangs au conseil et paramélise la prison en fixant le nombre maximum de prisonniers et en la réinitialisant

private void setGangs(LesGangs gangs)
Place le groupe de gangs sur le conseil

private void chargerIntelligencesTest()
Élimine tous les joueurs actuels et crée 4 joueurs totalement contrôlés par des intelligences de niveau différent et avec un volume zéro

private void chargerDefaultsJoueurs()
Élimine tous les joueurs actuels et crée 3 joueurs aidés par des intelligences de différents niveaux

private void chargerDefaultsGangs()
Crée 7 Gangs avec ses membres par défaut et appelle la fonction setGangs pour les ajouter au conseil d'administration

public void modeStatistiques(int ctt)
Initialise le mode statistique pour configurer les intelligences et mesurer leurs niveaux, les intelligences de charge et les gangs par défaut, puis faire le nombre de jeux passés par paramètre Ensuite, vous obtenez une liste classée avec les joueurs en fonction du nombre de jeux gagnés grâce à la classe Comparateur et la méthode de comparaison, puis montre les joueurs avec un pourcentage de jeux gagnés

private void modeVs()
Initialise le mode 1 à 1 qui met l'utilisateur à jouer contre l'intelligence de niveau 3

private void modifierPrison()
Permet à l'utilisateur de modifier le nombre maximum de prisonniers dans la prison

private void suprimerJoueur()
Permet à l'utilisateur d'éliminer les joueurs

public void createJoueurs()
Permet à l'utilisateur de créer de nouveaux joueurs

public void createGangs()
Permet à l'utilisateur de créer de nouveaux gangs

private void deModification()
Permet à l'utilisateur de modifier le dé, son nombre de visages et les valeurs de ces

public void afficerLesParametres()
Permet de montrer les paramètres actuels du jeu, tels que les joueurs, la prison, les dés...

private void jouer()
Méthode qui vous permet de lire l'explication détaillée dans le code

Override
private void jouer()
Permet de redémarrer les joueurs et le conseil d'administration pour jouer avec les mêmes paramètres actuels