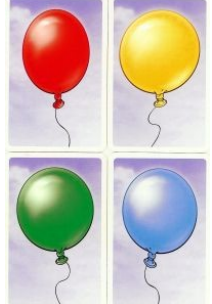


Présentation du jeu

« Ballons » est un jeu de cartes pour enfants à partir de 3 ans. Le jeu est très simple et ne demande aucune stratégie. Il est composé de 2 paquets de cartes.

	Les cartes « Ballon » représentent un ballon de couleur (il y a 5 couleurs : Rouge, Jaune, Vert, Bleu et Violet)		Les cartes « Action » représentent soit un ballon d'une certaine couleur qui est cassé ou perdu soit une « maman » qui peut « réparer » un ballon.
---	--	--	--

Le jeu peut se jouer de 2 à 5 joueurs.

Le paquet de cartes « Ballon » est mélangé et chaque joueur reçoit 5 cartes qu'il pose devant lui, face visible. Le reste du paquet est mis de côté et ne servira plus.

Le paquet de cartes « Action » est mélangé et posé au milieu du jeu, face cachée.

A tour de rôle, un joueur pioche une carte « Action », exécute l'action qui s'y trouve et puis la défausse.

- Pour une carte de type « casser un ballon d'une certaine couleur », le joueur retourne (la met face cachée) un de ses ballons de la couleur indiquée. Par exemple, si il tire la carte montrant un ballon violet qui s'envole, il doit retourner sa carte qui représente un ballon violet
 - Si il ne possède pas de ballon violet, la carte est sans effet.
 - Si il en possède plusieurs, il en retourne une au choix.
- Avec une carte « Maman », il peut « réparer » (remettre face visible) une carte préalablement retournée
 - Si il ne possède pas de ballon retourné, la carte est sans effet.
 - Si il en possède plusieurs, il en « répare » un seul, au choix.

Lorsqu'un joueur n'a plus de ballon visible, il a perdu. Le dernier joueur à rester en jeu a gagné.

Programmation du jeu

On vous demande d'implémenter ce jeu. Comme pour le projet, on va vous présenter une à une chaque classe à écrire.

La Couleur

Ecrire l'énumération définissant la couleur (il y en a 5 en tout) des ballons. La méthode toString() affiche le nom de la couleur en clair.

Les cartes « Ballon »

Ecrire la classe CarteBallon qui définit une carte de type « Ballon ».

Elle possède 2 attributs :

- Le premier représente la couleur du ballon
- Le second indique si la carte est face cachée ou visible (retournée ou non)

Ecrivez aussi

- Un constructeur recevant la couleur en paramètre (elle est face visible)
- Les assesseurs associés aux 2 attributs (mais la couleur ne peut pas être modifiée)
- Une méthode toString().

Les cartes « Action »

Ecrire la classe CarteAction qui définit une carte de type « Action ».

Elle possède 2 attributs :

- Le premier est un booléen indiquant si il s'agit d'une carte « Maman » ou pas.
- Le deuxième est une couleur. Cet attribut n'a de sens que si il ne s'agit pas d'une carte « Maman ».

Ecrivez aussi

- Un constructeur sans paramètre créant une carte de type « Maman »
- Un constructeur a un paramètre (la couleur) créant une carte de type « Casser un ballon ».
- Les getteurs (pas de setteurs)
- Une méthode toString().

La main d'un joueur

Ecrire la classe Main qui représente les 5 cartes « Ballon » d'un joueur.

Je vous laisse définir les attributs nécessaires. Ecrivez aussi

- Un constructeur qui reçoit en paramètre un tableau de 5 cartes « Ballon », la main du joueur.
- Une méthode « void cacher(Couleur c) » qui retourne une carte de la couleur indiquée si il en existe une (n'importe laquelle si il en existe plusieurs)
- Une méthode « void réparer() » qui répare une carte cachée si il en existe une (n'importe laquelle si il y en a plusieurs)
- Une méthode « boolean aPerdu() » qui indique si le joueur a perdu (toutes ses cartes sont cachées)
- Une méthode toString().

Tests

Ecrivez une classe de test pour la main de joueur. Testez de façon pertinente la validité de la méthode « cacher ».

Documentation

Ecrivez et générez la documentation de la classe Main.

Le Paquet de cartes « Ballon »

La classe PaquetBallon représente le paquet des cartes de type « Ballon ». Le paquet est représenté par une Pile (Stack en anglais, voyez la documentation)

- Le constructeur (sans paramètre) crée un paquet contenant 5 exemplaires de chaque carte (5 rouges, 5 verts, ...). Vous ferez appel à une méthode privée « remplir » pour remplir le paquet.
- La méthode mélanger() mélange le paquet. Pour simplifier votre travail, ne mettez rien dans cette méthode pour l'instant (elle ne mélange donc pas le paquet).
- La méthode piocher() retire et retourne la première carte du paquet. Si le paquet est vide, on le remplit au préalable.

Le Paquet de cartes « Action »

La classe PaquetAction est très proche de la classe précédente mais concerne les cartes Actions. A nouveau, il y a 5 exemplaires de chaque type de carte.

Le joueur

Nous ne définissons pas la notion de Joueur. Chaque joueur sera représenté par un numéro et c'est tout. Inutile donc d'en faire une classe.

Le jeu proprement dit

La classe JeuBallons implémente le jeu proprement dit. La logique succincte est la suivante (bien sûr nous vous conseillons de modulariser votre code).

Demande le nb de joueurs.
Le nombre de joueurs en lice est initialisé à ce nombre.
Créer les 2 paquets
Créer les mains
Le premier joueur commence
Tantque la partie n'est pas finie faire
 Le joueur courant pioche une carte action et l'exécute.
 Si il a perdu, diminuer le nombre de joueurs en lice
 Passer au joueur suivant
 (attention, il est possible que certains joueurs ne soient plus dans la course)
Fin tantque
Le joueur courant est le gagnant.

Amélioration du code

Le code des classes PaquetBallon et PaquetAction est très proche. Pour éviter cela, on peut.

- Ecrire une classe abstraite Carte ne contenant que l'attribut Couleur
- Les classes CarteBallon et Carte Action héritent à présent de cette classe Carte
- Ecrire une classe abstraite Paquet qui définit explicitement les méthodes piocher() et mélanger(). Attention la méthode piocher() fait appel à la méthode remplir() qui, elle, ne peut pas encore être écrite pour l'instant.
- Adapter les classes PaquetBallon et PaquetAction.

Mélanger la paquet

Vous avez vu au cours qu'il existe une méthode dans la librairie standard pour trier une Collection. Il existe également une méthode pour le mélanger (shuffle en anglais). Utilisez-là pour implémenter votre méthode mélanger.