

Exercices Python Objet

Jeu du Memory

Considérer un paquet de $2*n$ cartes, organisées par paires, qui seront présentées, faces cachées, à un joueur. Ce joueur doit retrouver les paires en retournant deux cartes, l'une après l'autre. Lorsqu'il constitue une paire, elle est retirée. Le programme doit compter le nombre de coups effectués pour retirer toutes les paires et permettre de choisir le nombre de cartes à générer.

Une carte possède deux attributs : une valeur entière et une lettre (minuscule ou majuscule). Toutes les valeurs sont différentes, toutes les lettres sont différentes. Les cartes sont mélangées avant d'être disposées sur une grille à peu près carrée où chaque case est représentée par des coordonnées (lettre, nombre).

Évolution : faire évoluer votre code pour permettre de choisir le nombre de cartes identiques à retrouver, plutôt que ce nombre soit égal à 2.

Jeu du plus ou du moins

Nous allons considérer une autre sorte de paquet de cartes contenant une valeur comme attribut et une couleur. Créer une classe commune à ces deux paquets de cartes qui va contenir toutes les informations et fonctions liées à la gestion de la valeur de la carte. Modifier votre classe Memory pour créer une classe CarteMemory qui ajoute une lettre et les fonctionnalités liés aux cartes du memory. Vérifier que votre programme précédent fonctionne toujours très bien.

Créer une classe Carte32 qui ajoute une couleur à une Carte et permet de gérer les cartes d'un jeu de 32 cartes (valeurs de 7 à 14, couleur rouge ou noir). Proposez un jeu qui permet de deviner, sur un tirage aléatoire du paquet de 32 cartes, si la carte suivante sera plus grande ou plus petite que la précédente et qui enregistre la plus grande suite de bonnes prédictions réalisée sans erreur sur le paquet de 32 cartes.

Yams / Yathzee

Faire les constructeurs permettant de générer un gobelet contenant 5 dés .

Faire une méthode estYams() qui indique si un gobelet contient 5 fois le même chiffre sur les dés.

Faire une méthode compter(valeur) qui compte le nombre d'apparition de "valeur" dans un gobelet de dés.

Faire une méthode sommeValeurs(valeur) qui retourne la somme des dés du gobelet de cette "valeur".

Faire une méthode qui indique si un gobelet contient au moins un breton (3)

Faire une méthode qui indique si un gobelet contient au moins un carré (4)

Faire une méthode qui indique si un gobelet contient au moins un full (3 et 2)

Faire une méthode qui indique si un gobelet contient une petite suite (4 valeurs)

Faire une méthode qui indique si un gobelet contient une grande suite (5 valeurs)

Faire une méthode change(position) qui permet de relancer le dé situé en "position" dans un gobelet.

Faire une méthode changeCertains(listeDePositions) qui permet de relancer les dés situés aux positions indiquées par listeDePositions.

Faire une méthode nouveauLance() qui présente à un utilisateur un nouveau gobelet et lui demande lesquels il souhaite relancer.

Faire une méthode qui permet de jouer un tour : faire un jet de dés, le relancer partiellement trois fois et choisir une combinaison valide pour ce lancer.

Faire une méthode qui joue au Yams en solitaire. La feuille de points est un objet, pouvant être

représentée par un dictionnaire Python qui associe une mission à une valeur qui par défaut vaut -1.

Faire une méthode pour jouer au Yams à plusieurs joueurs.