

M1 E3A

2023-2024

Projet informatique

On souhaite écrire un programme permettant de jouer à un jeu de carte dérivé du Black Jack.. Le principe général du jeu est que les joueurs jouent seuls contre le croupier (la banque). Un joueur gagne s'il totalise 21 points. S'il dépasse 21 points il perd tout l'argent qu'il a misé. Au départ, le croupier dispose d'un (gros) paquet de cartes composé de plusieurs jeux de 52 cartes mélangés. La valeur des cartes est la suivante : As : 1 point, 2 : 2 points, 9 : 9 points, 10 : 10 points, Valet, Dame, Roi : 10 points.

Déroulement d'une partie : Chaque joueur dispose au départ d'une certaine somme d'argent. En début de partie les joueurs choisissent combien ils souhaitent miser. Il y a une somme minimale pour avoir le droit de miser (définie en début de partie par le croupier). Le croupier tire une carte à chaque joueur (visible par tout le monde). Ensuite à tour de rôle, chaque joueur réalise une action parmi les 3 actions possibles : tirer carte(s), passer son tour, arrêter. Si le total d'un joueur atteint exactement 21 points, le joueur a gagné et la partie s'arrête. La banque lui verse 1,5 sa mise de départ. Des règles plus précises seront introduites dans la suite du projet.

Conseil : Durant tout le projet il vous est conseillé de compléter au fur et à mesure un diagramme UML des classes.

Partie 1

Il s'agit dans cette partie de définir l'environnement permettant de manipuler des cartes et de les afficher à l'écran ainsi que de créer le jeu de carte du croupier et de l'afficher.

1. Une carte est définie par 3 attributs : figure, couleur, valeur. On définit la classe `Carte` contenant les attributs précédemment définis, auquel on ajoutera les getter et les setter habituels ainsi que la méthode `__str__` de sorte que le programme principal suivant affiche :

```
if __name__ == '__main__':
    C=Carte('as','pique',1)
    print(C)
```

AS de PIQUE : valeur 1

On souhaite afficher l'image de la carte à l'écran. Pour cela, vous disposez dans le dossier « cartes » des images au format png. Le nom des fichiers est du type (pour un 3 de Trèfle) : test-03-trefle-img.png.

Ajouter la méthode `plot` qui devra charger l'image correspondante et l'afficher à l'écran. Le code (incomplet) de la méthode est détaillé ci-dessous :

```
def plot(self, axes=None):
    .....
    img=
    if axes is not None:
        axes.axes.clear()
        imgplot = axes.imshow(img)
        axes.axis('off')

    else:
        plt.clf()
        imgplot = plt.imshow(img)
        plt.axis('off')
```

Le programme principal ci-dessous devra afficher l'image d'un as de pique à l'écran :

```
if __name__ == '__main__':
    C=Carte('as','pique',11)
    C.plot()
```

2. On définit un paquet de cartes à l'aide d'une classe appelée `PaquetCartes` composée d'un seul attribut de type liste appelé `listeCartesDuPaquet`. A la création d'une instance `PaquetCartes`, le paquet sera vide.

Les méthodes seront les suivantes :

`__str__()` : liste l'ensemble des cartes du paquet
`__len__()` : retourne le nombre de cartes du paquet
`melanger()` : mélange les cartes du paquet (voir fonction `shuffle` du module `random`)
`tirerCarte()` : tire la première carte du paquet et la renvoie (renvoie `None` si paquet vide)
`ajouterCarteDansPaquet(carte)` : ajoute carte à la fin du paquet de carte
`getValeurDuPaquet()` : retourne la valeur du paquet
`clearPaquet()` : supprime toutes les cartes du paquet

Le programme principal ci-dessous devra afficher :

```
if __name__ == '__main__':
    C1=Carte('as','pique',1)
    C2=Carte('as','coeur',1)
    C3=Carte('Valet','Trefle',10)
    C4=Carte('3','Pique',3)
    P=PaquetCartes()
    P.ajouterCarteDansPaquet(C1)
    P.ajouterCarteDansPaquet(C2)
    P.ajouterCarteDansPaquet(C3)
    P.ajouterCarteDansPaquet(C4)
    print('Le paquet contient {} cartes et sa valeur est {}'.format(len(P),P.getValeurDuPaquet()))
    print(P)
    carte=P.tirerCarte()
    print('Le {} a été retiré du paquet'.format(carte))
```

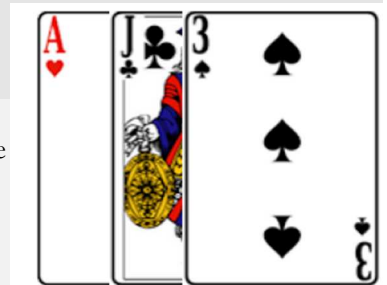
```
Le paquet contient 4 cartes et sa valeur est 15
AS de PIQUE : valeur 1,
AS de COEUR : valeur 1,
VALET de TREFLE : valeur 10,
3 de PIQUE : valeur 3
Le AS de PIQUE : valeur 1 a été tiré du paquet
```

On souhaite également afficher les images des cartes du paquet. Pour cela, on ajoute la méthode `plot` ci-dessous :

```
def plot(self,fig=None,left=0,bottom=0,width=0.2,height=0.2,shift=0.05):
    # Affiche l'ensemble des cartes du paquet en les décalant
    if fig==None:
        fig=plt.figure()

    for c in self.listeCartesDuPaquet:
        axes=fig.add_axes([left,bottom,width,height])
        c.plot(axes)
        left+=shift
        if left >0.9:
            left=0
            bottom +=0.15
```

Ajouter cette méthode et compléter le programme principal afin d'afficher le paquet de carte. Vous devez obtenir quelque chose ressemblant à la figure ci-contre.



3. Le jeu de cartes du black Jack est composé d'un ou plusieurs jeux de 52 cartes mélangés.

On finit la classe `JeuCartesBlackJack` qui dérive de la classe `PaquetCartes` permettant de représenter l'ensemble des cartes. A la création d'une instance on devra préciser combien de jeux de 52 cartes on utilise. Le paquet de cartes créé devra donc contenir $52 \times N$ cartes (avec N le nombre de paquets de 52 cartes utilisé). Il faudra penser à mélanger les cartes. Il n'y a pas de nouveaux attributs ni de nouvelles méthodes à créer hormis le constructeur.

Exemple d'affichage obtenu avec le programme principal suivant :

```
if __name__ == '__main__':
    jeu=JeuCartesBlackJack(2)      # creation d'un jeu de black Jack à partir de 2 jeux de 52 cartes
    jeu.plot()
```

