

Dernière mise à jour	Informatique	Denis DEFAUCHY
25/03/2021	7 - Matrices de pixels et images	TD 7-5 - Tour de magie

Informatique

7

Matrices de pixels et images

TD 7-5

Tour de magie

Subplot

Dernière mise à jour	Informatique	Denis DEFAUCHY
25/03/2021	7 - Matrices de pixels et images	TD 7-5 - Tour de magie

Contexte

Le principe de ce tour de magie consiste à prendre 21 cartes et à en faire choisir une au public, on appellera Nicolas notre cobaye. Il peut juste la regarder de loin, pas besoin de la prendre, mais il peut aussi la prendre, la remettre dans le paquet et le mélanger.

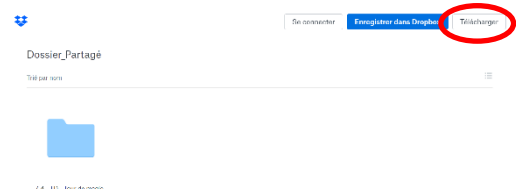
Le magicien dispose alors les cartes ligne par ligne, dans l'ordre, sur 3 colonnes, et demande dans quelle colonne est la carte choisie. Nicolas répond, le magicien regroupe alors les 3 paquets de 7 cartes en veillant à mettre celui désigné par la personne entre les deux autres.

Le magicien répète cette opération 2 fois supplémentaires.

A la fin, le magicien enlève 12 cartes en citant les lettres A B R A C A D A B R A, quel que soit le sens de vidange du paquet, et oh magie, la carte correspondant au 1 est la carte choisie par Nicolas.

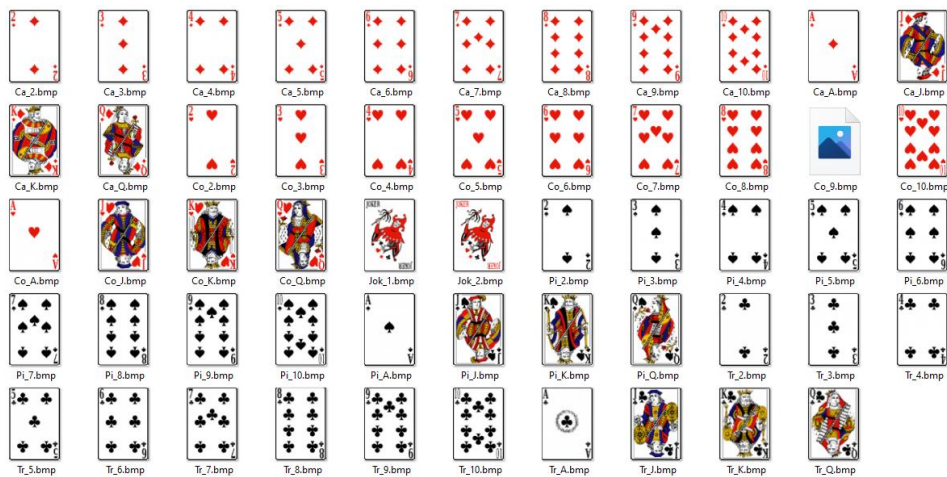
Téléchargement du dossier élèves

Téléchargez le dossier [derrière ce lien](#). N'ouvrez pas le dossier nommé « Dossier élèves », mais téléchargez le directement. Vous aurez les images sources, les images recherchées, et un code Python à compléter.



Création des cartes

Vous avez à votre disposition un dossier nommé « Cartes » contenant toutes les cartes d'un jeu de 52 cartes + 2 joker.



Vous remarquerez la logique des noms des cartes.

Question 1: Créer un code permettant d'établir presque automatiquement la liste des noms des cartes du dossier « Cartes » comme ci-contre

```
>>> Jeu
['Co_A.bmp', 'Co_2.bmp', 'Co_3.bmp',
 'Co_4.bmp', 'Co_5.bmp', 'Co_6.bmp',
 'Co_7.bmp', 'Co_8.bmp', 'Co_9.bmp',
 'Co_10.bmp', 'Co_11.bmp', 'Co_12.bmp',
 'Co_13.bmp', 'Co_14.bmp', 'Co_15.bmp',
 'Co_16.bmp', 'Co_17.bmp', 'Co_18.bmp',
 'Co_19.bmp', 'Co_20.bmp', 'Co_21.bmp',
 'Co_22.bmp', 'Co_23.bmp', 'Co_24.bmp',
 'Co_25.bmp', 'Co_26.bmp', 'Co_27.bmp',
 'Co_28.bmp', 'Co_29.bmp', 'Co_30.bmp',
 'Pi_1.bmp', 'Pi_2.bmp', 'Pi_3.bmp',
 'Pi_4.bmp', 'Pi_5.bmp', 'Pi_6.bmp',
 'Pi_7.bmp', 'Pi_8.bmp', 'Pi_9.bmp',
 'Pi_10.bmp', 'Pi_11.bmp', 'Pi_12.bmp',
 'Pi_13.bmp', 'Pi_14.bmp', 'Pi_15.bmp',
 'Pi_16.bmp', 'Pi_17.bmp', 'Pi_18.bmp',
 'Pi_19.bmp', 'Pi_20.bmp', 'Pi_21.bmp',
 'Pi_22.bmp', 'Pi_23.bmp', 'Pi_24.bmp',
 'Pi_25.bmp', 'Pi_26.bmp', 'Pi_27.bmp',
 'Pi_28.bmp', 'Pi_29.bmp', 'Pi_30.bmp',
 'Tr_1.bmp', 'Tr_2.bmp', 'Tr_3.bmp',
 'Tr_4.bmp', 'Tr_5.bmp', 'Tr_6.bmp',
 'Tr_7.bmp', 'Tr_8.bmp', 'Tr_9.bmp',
 'Tr_10.bmp', 'Tr_11.bmp', 'Tr_12.bmp',
 'Tr_13.bmp', 'Tr_14.bmp', 'Tr_15.bmp',
 'Tr_16.bmp', 'Tr_17.bmp', 'Tr_18.bmp',
 'Tr_19.bmp', 'Tr_20.bmp', 'Tr_21.bmp',
 'Tr_22.bmp', 'Tr_23.bmp', 'Tr_24.bmp',
 'Tr_25.bmp', 'Tr_26.bmp', 'Tr_27.bmp',
 'Tr_28.bmp', 'Tr_29.bmp', 'Tr_30.bmp',
 'Jok_1.bmp', 'Jok_2.bmp']
```

Dernière mise à jour	Informatique	Denis DEFAUCHY
25/03/2021	7 - Matrices de pixels et images	TD 7-5 - Tour de magie

Affichage

On rappelle que l'ouverture au format array d'une image se réalise avec la commande :

```
plt.imread(Chemin)
```

Question 2: Créer une fonction Lecture(Chemin) qui renvoie l'array associé à l'image de chemin 'Cartes\\'+Chemin où chemin est l'un des noms de la liste Jeu

On donne le code suivant :

```
import matplotlib.pyplot as plt
plt.close('all')

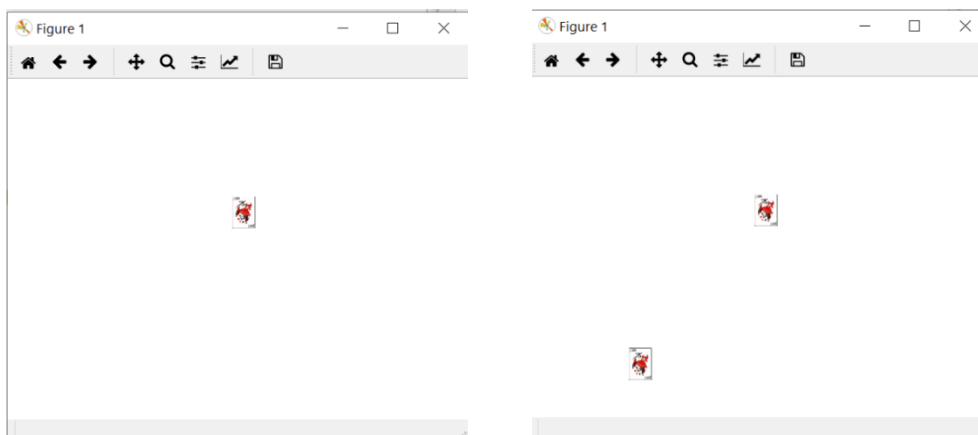
def Affiche(fig,Chemin):
    plt.figure(fig)
    Carte = Lecture(Chemin)
    plt.imshow(Carte)
    plt.axis('off')
    plt.show()
    plt.pause(0.5)
```

A partir du chemin d'une image, la fonction Affiche ouvre et affiche cette image sur une figure numéro fig, et fait attendre 0,5 secondes.

Question 3: Créer une fonction Affiche_lc(Chemin,l,c,L,C) ressemblant à la fonction Affiche, et permettant de mettre et d'afficher l'image à la ligne l et la colonne c d'un quadrillage de L lignes et C colonnes (cf subplot) sur une figure quelconque (pas de plt.figure) – On ajoutera une petite pause afin de rendre la distribution de cartes visuellement réaliste (0.1 par exemple)

Vérifiez dans l'ordre :

`Affiche_lc('Jok_1.bmp',3,2,7,3)` `Affiche_lc('Jok_1.bmp',7,1,7,3)`



Choix de la carte

Question 4: Créer un programme qui demande à l'utilisateur de retenir une carte parmi les 21 premières cartes du jeu, alors affichées successivement sur la figure 0

Dernière mise à jour	Informatique	Denis DEFAUCHY
25/03/2021	7 - Matrices de pixels et images	TD 7-5 - Tour de magie

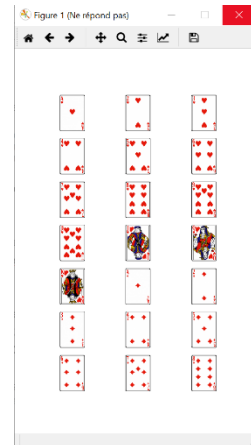
Manipulation des cartes

On suppose que Colonnes est une liste de 3 listes de 7 chemins d'images contenus dans « Jeu ».

```
>>> Colonnes
[['Co_A.bmp', 'Co_4.bmp', 'Co_7.bmp', 'Co_10.bmp', 'Co_K.bmp', 'Ca_3.bmp', 'Ca_6.bmp'],
 ['Co_2.bmp', 'Co_5.bmp', 'Co_8.bmp', 'Co_J.bmp', 'Ca_A.bmp', 'Ca_4.bmp', 'Ca_7.bmp'],
 ['Co_3.bmp', 'Co_6.bmp', 'Co_9.bmp', 'Co_Q.bmp', 'Ca_2.bmp', 'Ca_5.bmp', 'Ca_8.bmp']]
```

Question 5: Créer la fonction Affiche_Colonnes(fig,Colonnes) qui, après avoir vidé le contenu de la figure numéro fig, y affiche une image comme celle proposée ci-contre – Pour plus de réalisme, on respectera l'ordre de la distribution des cartes

Rappel : plt.figure(fig,figsize=(5,8)) permet de définir la taille de l'image créer en pouces



Question 6: Créer une fonction Empaquetage(Colonnes,c) qui renvoie une liste de 21 images (cartes) contenues dans les 3 colonnes (listes) de « Colonnes » en mettant au milieu la colonne d'indice Python c (0, 1 ou 2)

A partir de la liste Colonnes de l'exemple ci-dessus, j'obtiens :

```
>>> Empaquetage(Colonnes,2)
['Co_A.bmp', 'Co_4.bmp', 'Co_7.bmp', 'Co_10.bmp', 'Co_K.bmp', 'Ca_3.bmp', 'Ca_6.bmp',
 'Co_3.bmp', 'Co_6.bmp', 'Co_9.bmp', 'Co_Q.bmp', 'Ca_2.bmp', 'Ca_5.bmp', 'Ca_8.bmp',
 'Co_2.bmp', 'Co_5.bmp', 'Co_8.bmp', 'Co_J.bmp', 'Ca_A.bmp', 'Ca_4.bmp', 'Ca_7.bmp']
```

Question 7: Créer une fonction Distribution(Paquet) qui à partir d'une liste de 21 chemins d'images, les distribue ligne par ligne dans 3 colonnes (listes) et renvoyant la liste de ces 3 listes

A partir de la liste Colonnes de l'exemple ci-dessus, j'obtiens :

```
>>> Paquet = Empaquetage(Colonnes,2)
>>> Distribution(Paquet)
[['Co_A.bmp', 'Co_10.bmp', 'Ca_6.bmp', 'Co_9.bmp', 'Ca_5.bmp', 'Co_5.bmp', 'Ca_A.bmp'],
 ['Co_4.bmp', 'Co_K.bmp', 'Co_3.bmp', 'Co_Q.bmp', 'Ca_8.bmp', 'Co_8.bmp', 'Ca_4.bmp'],
 ['Co_7.bmp', 'Ca_3.bmp', 'Co_6.bmp', 'Ca_2.bmp', 'Co_2.bmp', 'Co_J.bmp', 'Ca_7.bmp']]
```

Tour de magie

Question 8: Ecrire un code permettant de créer la liste initiale « Colonnes » contenant les 21 premières cartes du jeu sous la forme de 3 listes (colonnes) de 7 chemins chacune - Cf exemple ci-dessus

Question 9: Créer un code réalisant les 3 étapes du tour de magie en affichant, à chaque fois, les 3 colonnes sur une figure 1, et demandant à l'utilisateur dans la console dans quelle colonne (1, 2 ou 3), se trouve la carte qu'il a choisie

Nous disposons maintenant d'un paquet de 21 cartes que l'on va vider depuis l'un de ses côtés.

Question 10: Créer un code affichant les cartes tirées dans le paquet sur une figure 2, et en parallèle une lettre du mot ABRACADABRA dans la console, et s'arrêtant enfin à la carte choisie par l'utilisateur avec un petit message « La voilà »