

Réussite familles de couleurs

1	La réussite de cartes « Familles de couleurs »	2
2	Modélisation de la réussite « familles de couleurs »	3
3	Travail à réaliser	7
3.1	Objectif	7
3.2	Cahier des charges.....	7
3.3	Bonus possible.....	8
4	Évaluation.....	8

Réussite familles de couleurs

<p>Thématique : Algorithmique</p> <p>Niveau : 1^{ère}</p> <p>Type d'activité : projet</p> <p>Point(s) du programme traité(s) : Types construits (p-uplets, tableaux de tableaux pour représenter des matrices, dictionnaire)</p> <p>Résumé : Ce projet fait référence à la réussite de cartes « Familles de couleurs » qui se joue avec 32 cartes.</p> <p>Deux matrices qui représentent à tout instant l'état du jeu.</p>	<p>Durée : 8h</p> <p>Matériel / logiciels :</p> <ul style="list-style-type: none">• Environnement de développement intégré Spyder avec le package Pygame. <p>Documents ressources :</p> <ul style="list-style-type: none">• Visualisation d'une partie sur https://www.youtube.com/watch?v=kJOT6ffL8hs• Cours sur les types construits disponible sur http://www.astrovirtuel.fr/nsi/premiere/premiere2022-2023/premiere2022-2023.htm• Le programme initial http://www.astrovirtuel.fr/nsi/premiere/premiere2022-2023/REUSSITE_FAMILLES_DE_COULEURS/reussite_familles_de_couleurs_eleve.zip
---	--

1 La réussite de cartes « familles de couleurs »

La réussite se joue seul. Le joueur dispose de 32 cartes.

Il commence par mélanger le paquet.

Il dispose ensuite, sur le plateau de jeu, quatre lignes de sept cartes dos visible.

Il reste quatre cartes qu'il place devant lui également dos visible : c'est la pioche.

Chaque carte placée représente une couleur et une valeur.

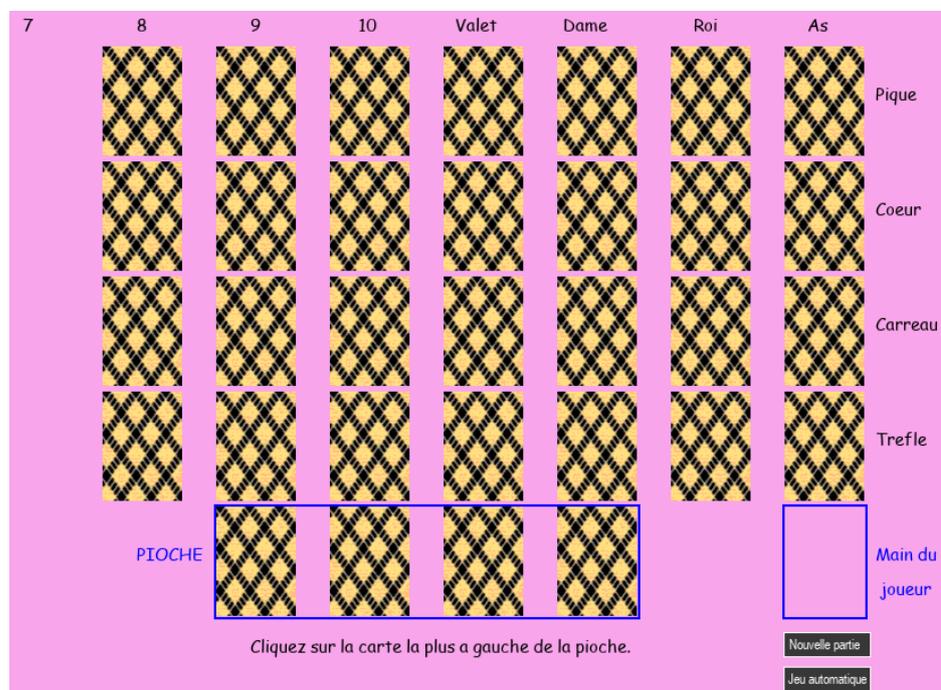


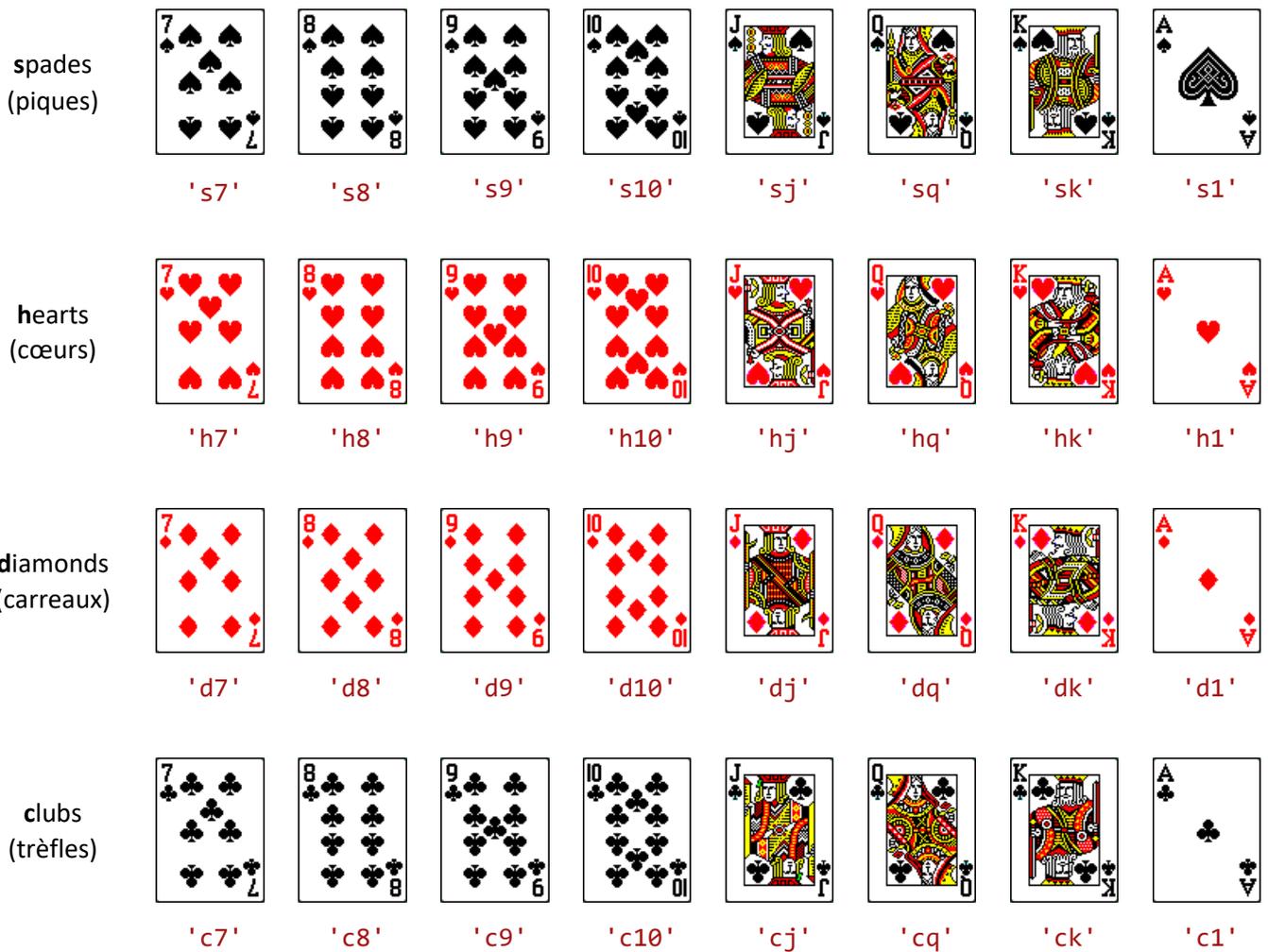
Figure 1

Nous utiliserons la disposition visible sur la figure 1 : lignes pique, cœur, carreau, trèfle et pioche.

2 Modélisation de la réussite « familles de couleurs »

L'objectif de ce mini-projet est de **modéliser un plateau de réussite Familles de couleurs** à partir de **deux matrices** de 5 lignes et 8 colonnes. Votre création numérique devra être en mesure :

- d'afficher la **matrice du jeu** dans la console Python. Les 32 cartes sont nommées d'après les appellations anglaises des couleurs et d'après leur valeur. Il y a une 33^e carte nommée '**vide**' qui représente une position vide sur le plateau de jeu.



- d'afficher la **matrice des dos** qui contient seulement des 0 (la carte est face visible ou vide) et des 1 (la carte est dos visible ou vide).

Sur les pages suivantes sont donnés des exemples permettant de comprendre le rôle des deux matrices qui décrivent l'état du jeu.

Exemple 1

Partie non terminée

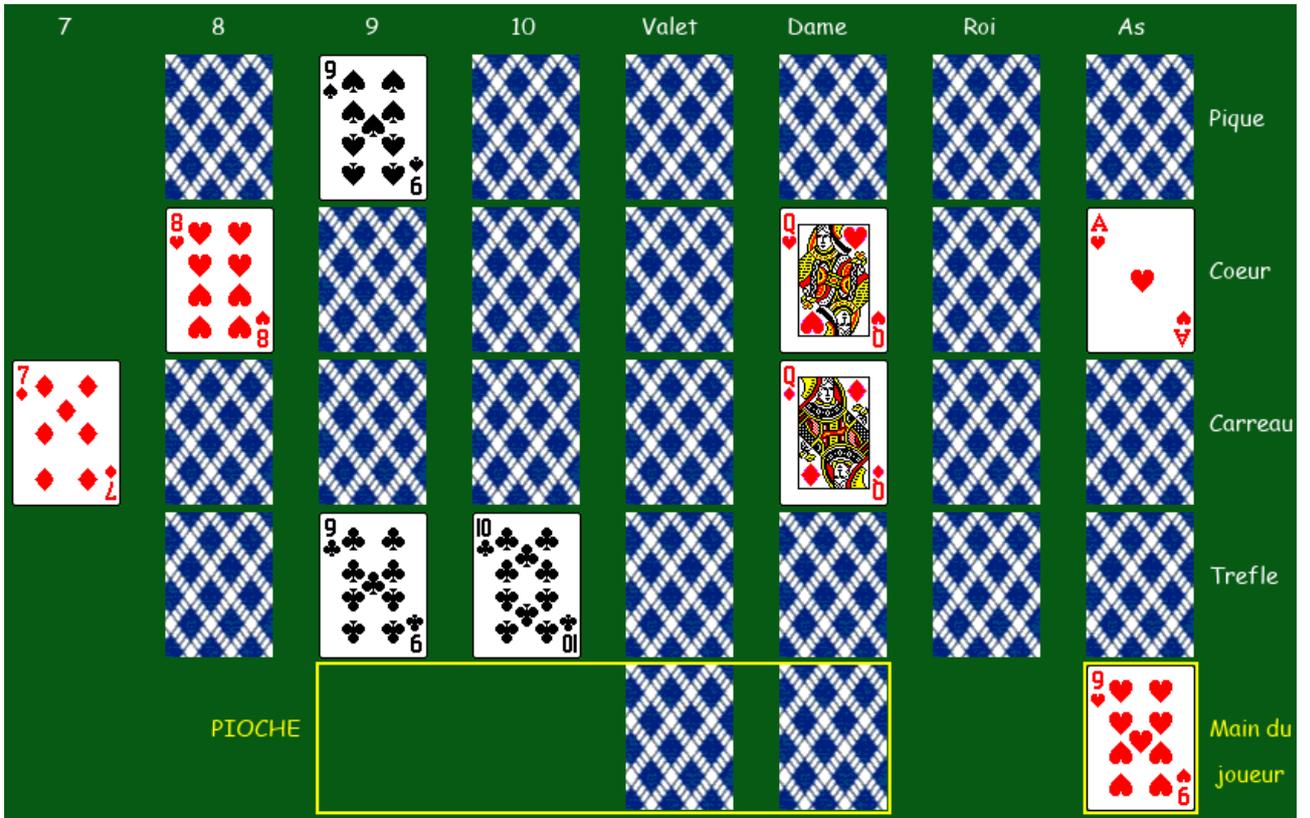


Figure 2

```

Matrice du jeu
[[ 'vide', 's10', 's9', 'sk', 'd8', 'h10', 'cj', 'h7' ]
 [ 'vide', 'h8', 'd10', 'ck', 's7', 'hq', 'cq', 'h1' ]
 [ 'd7', 'c7', 'c1', 'dk', 'd9', 'dq', 's8', 'sq' ]
 [ 'vide', 'sj', 'c9', 'c10', 'hk', 'dj', 'd1', 'hj' ]
 [ 'vide', 'vide', 'vide', 'vide', 'c8', 's1', 'vide', 'h9' ]

Matrice des dos
[[ 0, 1, 0, 1, 1, 1, 1, 1 ]
 [ 0, 0, 1, 1, 1, 0, 1, 0 ]
 [ 0, 1, 1, 1, 1, 0, 1, 1 ]
 [ 0, 1, 0, 0, 1, 1, 1, 1 ]
 [ 0, 0, 0, 0, 1, 1, 0, 0 ]

```


Exemple 3

Partie terminée sur une défaite

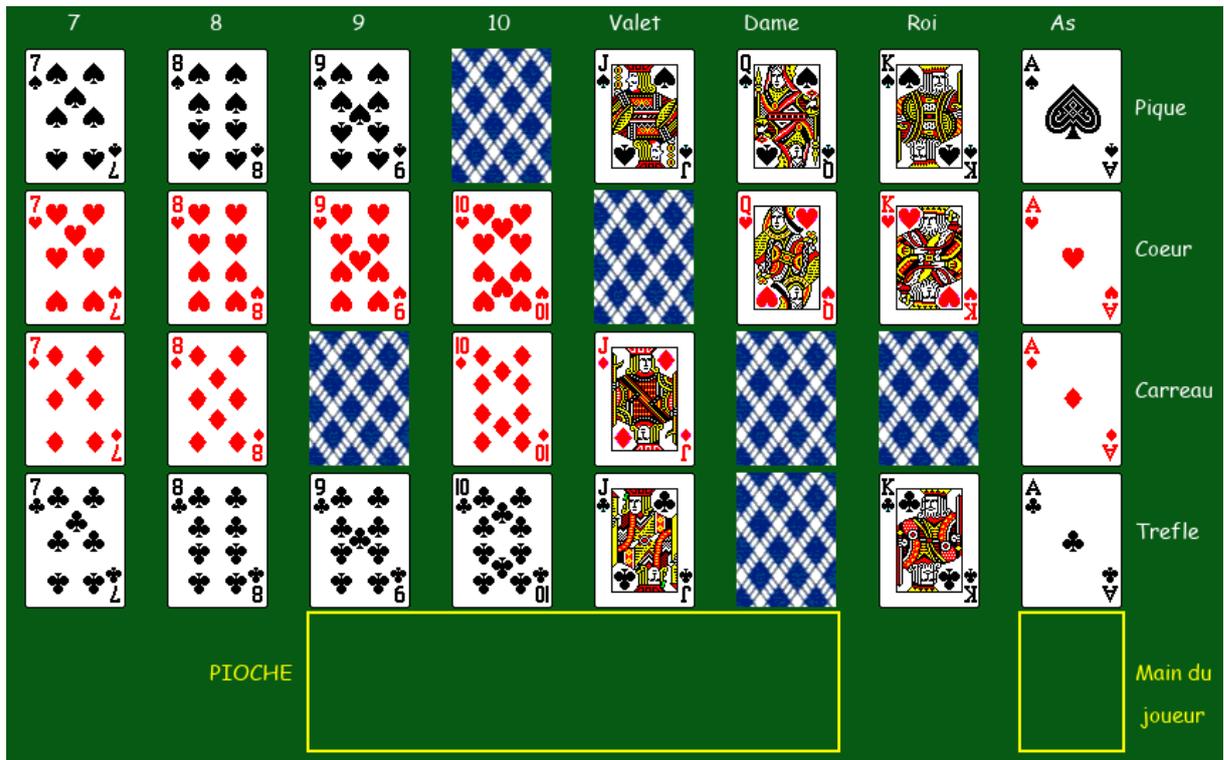


Figure 4

Matrice du jeu

```

[[ 's7', 's8', 's9', 's10', 'sj', 'sq', 'sk', 's1' ]
[ 'h7', 'h8', 'h9', 'h10', 'hj', 'hq', 'hk', 'h1' ]
[ 'd7', 'd8', 'cq', 'd10', 'dj', 'dq', 'dk', 'd1' ]
[ 'c7', 'c8', 'c9', 'c10', 'cj', 'd9', 'ck', 'c1' ]
[ 'vide', 'vide', 'vide', 'vide', 'vide', 'vide', 'vide', 'vide' ]

```

Matrice des dos

```

[[ 0, 0, 0, 1, 0, 0, 0, 0 ]
[ 0, 0, 0, 0, 1, 0, 0, 0 ]
[ 0, 0, 1, 0, 0, 1, 1, 0 ]
[ 0, 0, 0, 0, 0, 1, 0, 0 ]
[ 0, 0, 0, 0, 0, 0, 0, 0 ]

```

- de détecter la fin de la partie.
- de détecter une victoire éventuelle. La partie est finie et toutes les cartes du plateau sont face visible.
- de détecter une défaite éventuelle. La partie est finie et au moins une carte est dos visible.

Vous devrez donc :

- Construire un premier module contenant les fonctions :
 - o obtenir_paquet_32_cartes(couleurs, valeurs)
 - o melange(jeu)
 - o formate(liste_chaines)
- Construire un deuxième module contenant les fonctions :
 - o construit_dico_emplacement(couleurs, valeurs)
 - o affichage_console(matrice_chaines, matrice_masque)
- Construire un troisième module contenant les fonctions :
 - o detection_victoire(matrice_masque)
 - o detection_jeu_termine(matrice_chaines)

3 Travail à réaliser

3.1 Objectif

Téléchargez et décompressez le fichier http://www.astrovirtuel.fr/nsi/premiere/premiere2022-2023/REUSSITE_FAMILLES_DE_COULEURS/reussite_familles_de_couleurs_eleve.zip

dans votre dossier personnel sur le réseau du lycée.

Vous obtenez cinq fichiers : constantes.py, reussite.py, premier_module.py, deuxieme_module.py, troisieme_module.py. Les fichiers constantes.py, reussite.py ne doivent pas être modifiés.

Votre groupe doit se répartir les trois modules et compléter les fonctions qui s’y trouvent. Chaque membre du groupe est responsable de la réalisation de son module.

Exécutez dans Spyder le programme principal et allez dans la fenêtre Pygame. Constatez que si on clique sur la carte la plus à gauche de la pioche alors la carte ne va pas dans la main du joueur. Lorsque les modules seront complétés, ce défaut disparaîtra.

3.2 Cahier des charges

Votre module contient deux ou trois fonctions à compléter pour correspondre à la docstring déjà présente. Vos fonctions sont donc déjà parfaitement documentées dans leur docstring. Il vous suffira de les rendre opérationnelles. Vous pouvez prévoir des tests de vos fonctions dans la console.

L’exécution du programme présent dans le fichier reussite.py vous permettra également de contrôler le bon fonctionnement de vos trois modules lorsqu’ils auront été tous les trois complétés.

Votre projet doit être déposé sur l’espace de travail du Groupe Teams Première NSI dans votre dossier de groupe.

Remarques

- Un compte-rendu individuel au format Word portant votre nom devra également être déposé dans votre dossier de groupe sur Teams.
- Vous respecterez les recommandations données sur la maquette téléchargeable sur http://www.astrovirtuel.fr/nsi/premiere/premiere2021-2022/maquette_de_compte_rendu.pdf

3.3 Bonus possible

Compléter la fonction `jeu_automatique(coup_actuel, matrice_chaines, matrice_masque)` présente dans le fichier `reussite.py` de manière à pouvoir faire jouer l'ordinateur automatiquement.

4 Évaluation

- Le compte-rendu
- Le code avec les critères d'évaluation suivants :
 - Respect du cahier des charges.
 - Bonnes pratiques PEP-8.
 - Utilisation d'outils collaboratifs numériques.
 - Implication, organisation du travail en équipe.
 - Bonus.

Attention : Tout plagiat est sanctionné par un zéro !