

Le Memory



a. Objectif :

Créer un programme en langage Python qui permette de jouer au **jeu Memory** à deux joueurs.

b. Règles et déroulement :

Le jeu se compose de paires d'images portant des illustrations identiques. L'ensemble des images est mélangé, puis étalé face caché.

À son tour, chaque joueur retourne deux images de son choix.

S'il découvre deux images identiques, il les ramasse et les conserve. Ce même joueur peut alors rejouer.

Si les images ne sont pas identiques, il les retourne faces cachées à leur emplacement de départ.

Le jeu se termine quand toutes les paires d'images ont été découvertes et ramassées. Le gagnant est le joueur qui possède le plus de paires.

c. Cahier des charges :

Écrire un programme en python permettant à deux joueurs de jouer à ce jeu via une interface graphique créé avec Tkinter.

- Le jeu proposé sera constitué de 16 cartes soit 8 paires de cartes identiques (grille 4x4) ;
- Le programme fera appel à une ou plusieurs fonctions ;
- Le programme comptera le nombre de paires trouvées par chaque joueur et indiquera le joueur gagnant en fin de partie. (vous pourrez afficher le nombre de points de chaque joueur dans la fenêtre du jeu au fur et à mesure de la partie) .

Améliorations possibles :

- Mettre un temps limité pour jouer à chaque joueur ;
- En fin de jeu, vous pourrez nous proposer soit une nouvelle partie soit de quitter le jeu.

Le Motus

S	E	R	P	E	N	T
S	E	R	R	U	R	E
S	O	U	R	I	R	E
S	O	U	L	I	E	R
S	O	U	C	I	E	R

Code des couleurs:

Lettre bien placée

Lettre mal placée

Lettre absente du mot

a. Objectif :

Créer un programme en langage Python qui permette de jouer au **jeu Motus** en solo.

b. Règles et déroulement :

Le but du jeu est de retrouver un mot de 7 lettres (pas de verbe conjugué) en un nombre d'essais limités.

La première lettre du mot vous est donnée. Il faut alors proposer un mot et en déduire les lettres qui le composent à l'aide du code couleur suivant :

- une lettre colorée en **ROUGE** est dans le mot et est bien placée
- une lettre colorée en **JAUNE** est dans le mot mais est mal placée
- une lettre **NON COLORÉE** n'est pas dans le mot

Le joueur a le droit à 10 essais, après quoi il a perdu et le mot mystère lui est donné.

c. Cahier des charges :

- Écrire un programme en python permettant à l'utilisateur de jouer à ce jeu via une interface graphique créé avec Tkinter. *Vous pourrez avoir une fenêtre de saisie et en-dessous faire apparaître au fur et à mesure les mots proposés avec les codes couleurs décrits comme précédemment.*
- Vous devrez créer un fichier texte contenant un certain (grand !) nombre de mots à 7 lettres. Un mot sera alors choisi au hasard dans cette liste. L'utilisateur devra alors trouver ce mot avec pour indice de départ la première lettre.
- Le joueur a alors 10 essais en tout avec en indices les lettres bien placées ou celles présentes dans le mot.
- Si le joueur trouve le mot en 10 coups ou moins, lui annoncer qu'il est gagnant et lui proposer une nouvelle partie.
- Si le mot n'est pas trouvé au bout des 10 essais, dire au joueur qu'il a perdu et lui donner le mot mystère.
- Il est fortement conseillé de décomposer votre programme à l'aide de fonctions.

Amélioration possible : Mettre un temps limité pour proposer une réponse à chaque coup.

Tester le jeu au lien suivant : <http://www.funmeninges.com/masterkid-mot.html#sp>

La cryptographie



a. Objectif :

Créer un programme en langage Python qui permette de coder, décoder ou décrypter un message par le principe du codage César.

b. Règles et déroulement :

A l'aide d'une interface graphique, l'utilisateur pourra :

- soit donner un texte clair et une clé pour obtenir le texte codé correspondant ;
- soit donner un texte codé et une clé pour obtenir le texte clair ;
- soit donner un texte codé sans connaître la clé et obtenir la clé du codage ainsi que le texte en clair.

c. Cahier des charges :

- Écrire un programme en python permettant de coder un texte, la clé étant donnée ; ou bien de décoder un texte connaissant ou non la clé du codage.
- Le programme devra faire appel à une ou plusieurs fonctions.
- La fenêtre d'échange avec l'utilisateur se fera via une interface graphique créée avec le module Tkinter.

Voir lien suivant (pour vous donner des idées) : <http://www.dcode.fr/chiffre-cesar#1>

Le compte est bon



a. Objectif :

Créer un programme en langage Python qui permette de jouer au **compte est bon** du célèbre jeu « Des chiffres et des lettres ».

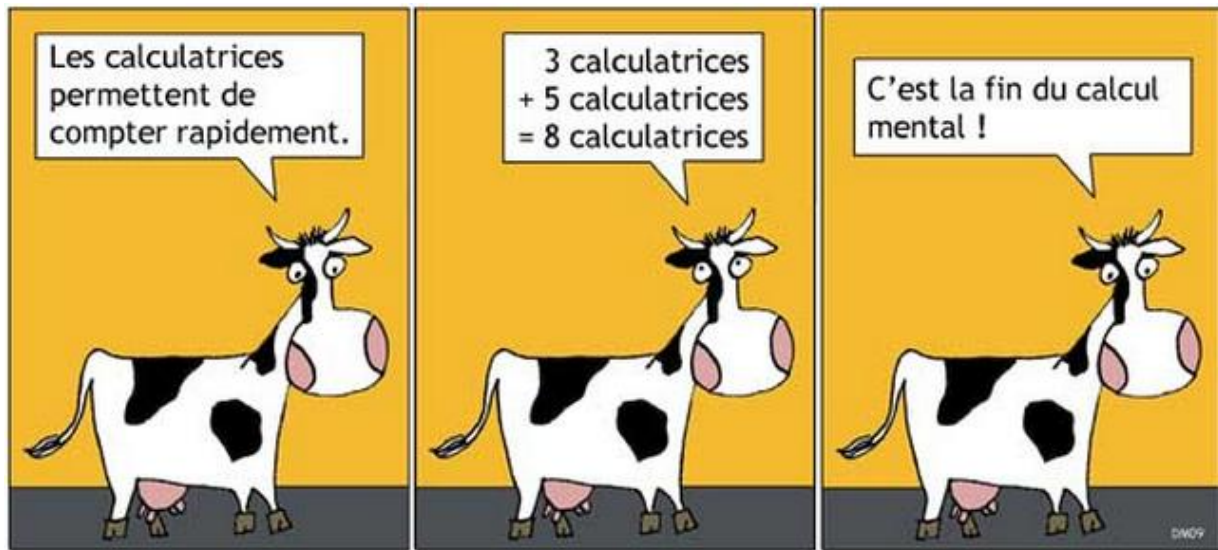
b. Règles et déroulement :

- Le programme génère une liste de six nombres au hasard ainsi que le nombre total à trouver par l'utilisateur.
- L'utilisateur doit déterminer ce total uniquement avec les quatre opérations de base (+, -, ×, ÷) mais peut les utiliser autant de fois qu'il veut.
- Une fois un nombre utilisé dans un calcul, il ne peut plus l'être une 2^e fois.
- Le résultat d'un calcul peut être utilisé comme nouveau nombre de la liste.
- Le jeu prend fin soit à la demande de l'utilisateur, soit si le compte est trouvé (vérifié par le programme), soit si tous les nombres ont été utilisés sans trouver le compte.

c. Cahier des charges :

- Les six nombres font nécessairement partie de la liste des entiers {1 ; 2 ; ... ; 9 ; 10 ; 25 ; 50 ; 75 ; 100}.
- Le compte à trouver est un entier compris entre 101 et 999 inclus.
- Le programme devra faire appel à une ou plusieurs fonctions.
- La gestion du jeu (saisies de l'utilisateur, résultats, nombres utilisés de la liste...) se fera via une interface du t-kinter.

Calcul mental



a. Objectif :

Créer un programme en langage Python qui soit un **générateur de calcul mental** avec différents niveaux de difficultés.

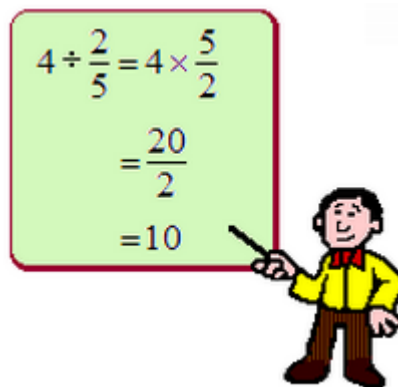
b. Règles et déroulement :

- Le programme propose un calcul à l'utilisateur.
- L'utilisateur doit saisir sa réponse pour vérification et le programme répond si la proposition est juste ou fausse.
- Le programme doit proposer des séries de 10 calculs et afficher le score après chaque série.
- Le programme doit pouvoir proposer au moins trois niveaux de difficultés.
- Le programme prend fin à la demande de l'utilisateur.

c. Cahier des charges :

- Les calculs doivent faire appel aux quatre opérations (+, -, ×, ÷) ainsi qu'aux puissances de 2 et de 3 et à la racine carrée.
- Les nombres négatifs doivent être pris en compte.
- Le programme devra faire appel à une ou plusieurs fonctions.
- La gestion du programme (saisies de l'utilisateur, réponse Juste/Faux, score...) se fera via une interface du t-kinter.

Calcul fractionnaire



a. Objectif :

Créer un programme en langage Python qui permette **le calcul fractionnaire en valeurs exactes**.

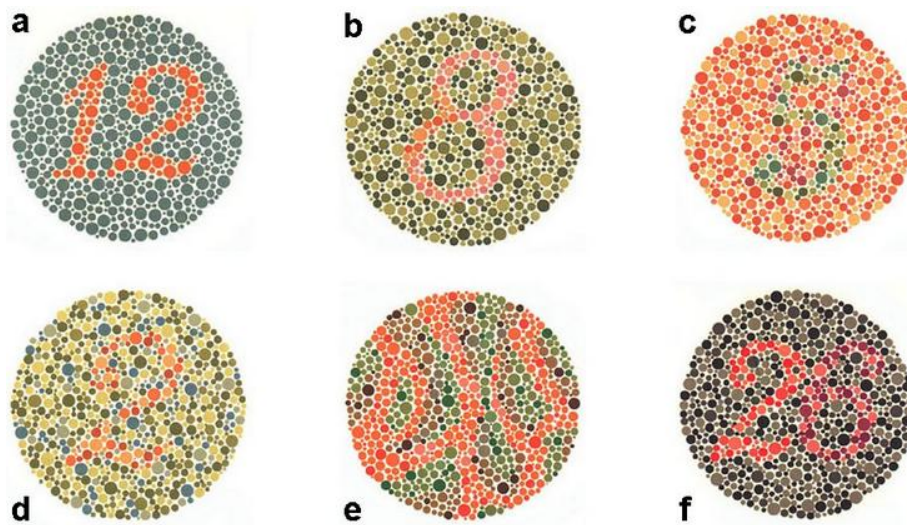
b. Règles et déroulement :

- Le programme demande la saisie d'un calcul par l'utilisateur.
- La saisie d'un calcul consiste en la saisie d'une fraction suivie d'une opération suivie d'une 2^e fraction.
- Le programme doit différencier les quatre cas d'opérations et traiter les calculs avec les règles mathématiques puis retourner le résultat.
- Le programme doit pouvoir gérer les cas particuliers de saisie et/ou de calculs (division par zéro, résultats entiers, simplifications de signes...).
- Le programme prend fin à la demande de l'utilisateur.

c. Cahier des charges :

- Les calculs doivent faire appel aux quatre opérations (+, -, ×, ÷).
- Les fractions saisies seront du type « a/b » et « c/d » où a, b, c et d sont des entiers relatifs.
- Le programme doit pouvoir gérer les cas particuliers de saisie et/ou de calculs (division par zéro, résultats entiers, simplifications de signes...).
- Le programme devra faire appel à une ou plusieurs fonctions.
- La gestion du programme (saisies de l'utilisateur, affichage de la réponse, affichage d'exception...) se fera via une interface du t-kinter.

Le daltonisme



a. Objectif :

Créer un programme en langage Python qui simule **la perception d'une image par un daltonien**.

b. Déroulement :

- Le programme demande à l'utilisateur le fichier image à modifier.
- L'utilisateur renseigne le chemin de l'image et indique également selon quel type de daltonisme on souhaite modifier l'image.
- Le programme affiche l'image telle qu'elle serait perçue par un daltonien.

c. Cahier des charges :

- On intégrera les quatre cas suivants de daltonisme : l'achromatopsie, la deutéranopie, la protanopie et la tritanopie (effectuer des recherches pour les significations).
- Le programme devra faire appel à une ou plusieurs fonctions.
- L'utilisateur peut souhaiter l'affichage de plusieurs images modifiées selon un ou plusieurs des cas de daltonisme.
- La gestion du programme (accueil, chemin d'accès à l'image, choix du ou des types de daltonisme...) se fera via une interface du t-kinter.

Page web sur Python



a. Objectif :

Créer une page web pour aider un internaute à faire ses premiers pas en Python.

b. Cahier des charges :

- La page web sera écrite en langage HTML avec une feuille de style en CSS

Voir lien suivant : <http://fr.openclassrooms.com/informatique/cours/apprenez-a-creeer-votre-site-web-avec-html5-et-css3>

- Elle se présentera de la manière suivante :

