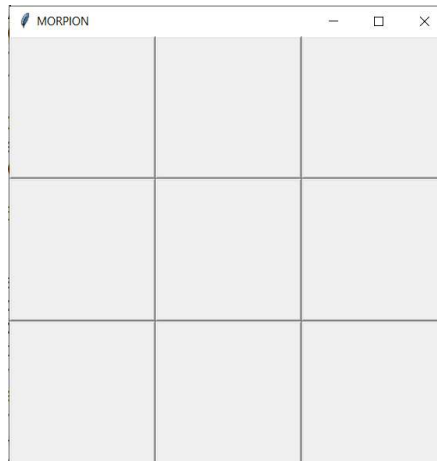
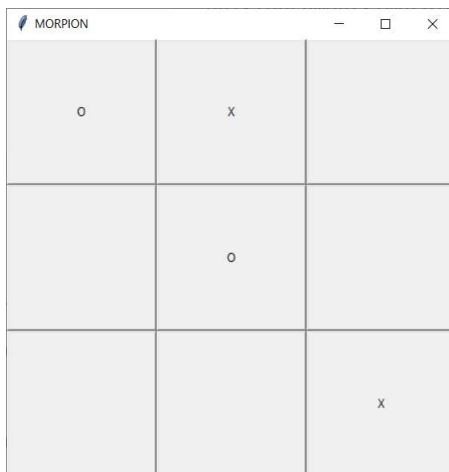


Création d'un jeu de Morpion en Python avec la bibliothèque graphique Tkinter.

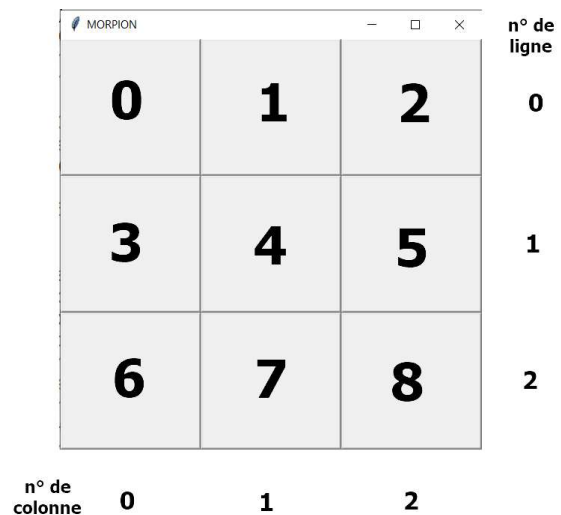
Les cases du jeu seront représentées par une grille de boutons.



Quand le joueur 1 (respectivement 2) clique sur un bouton libre, le texte affiché sur le bouton sera 'O' (resp. 'X').



On a numéroté les boutons de 0 à 9. Chaque bouton est repéré par ses coordonnées (i,j) où i est le n° de ligne et j le n° de colonne.



Le programme utilisera 4 fonctions: **game**, **update**, **check** et **reset**.

- Fonction **game** (evt, i, j)

La fonction **game** est appelée lors du clic d'un bouton. Elle a pour paramètres l'événement bouton evt et les coordonnées (i, j) du bouton et pour variables globales: la liste des boutons, le code du joueur (-1 ou 1), le nombre de coups joués et le code gagnant (-1, 0 ou 1).

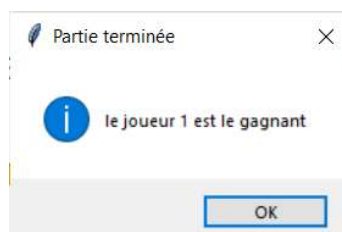
Le joueur 1 aura le code -1, le joueur 2 le code 1. Le code gagnant sera le code du joueur. Le code gagnant 0 indique qu'il n'y a pas de gagnant.

Elle vérifie si le coup est jouable (i.e. si la case est libre) et affiche alors le symbole du joueur sur le bouton.

La fonction **game** appellera la fonction **update** et lancera une boîte d'information (**messagebox**) indiquant le joueur gagnant.

Pour afficher une messagebox, il faut ajouter l'importation:

from tkinter import messagebox et utiliser la méthode *messagebox.showinfo (titre_de_la_fenetre, message_à_afficher)*



- Fonction **update(i, j)**

La fonction **update** où (i, j) sont les coordonnées du bouton met à jour les listes suivantes:

M=[[0,0,0],[0,0,0],[0,0,0]] # la grille de jeu (0= libre, -1=joueur 1, 1= joueur 2)

L=[[],[],[]] # représente les lignes

C=[[],[],[]] # représente les colonnes

D=[[],[]] # représente les diagonales

Les listes L, C et D vont permettre de savoir si on a un gagnant. Si l'une des sous-listes est de taille 3, il y a la possibilité d'un gagnant. Si la somme de ses éléments est -3, le joueur 2 est gagnant. Si elle vaut 3 alors le joueur 1 est gagnant sinon on a un match nul.

Exemple:

```
Joueur1=-1
Joueur2=1
M=[[0,0,0],      L=[[ ],[ ],[ ]] C=[[ ],[ ],[ ]] D=[[ ],[ ]]
  [0,0,0],
  [0,0,0]]

Le joueur 1 clique sur le bouton 4.
M=[[0,0,0],
  [0,-1,0],
  [0,0,0]]

L =[[ ],[-1],[ ]] C=[[ ],[-1],[ ]] D=[[-1],[-1]]

Le joueur 2 clique sur le bouton 0.
M=[1,0,0,0,-1,0,0,0,0]
M=[[1,0,0],
  [0,-1,0],
  [0,0,0]]

L =[[1],[-1],[ ]] C=[[1],[-1],[ ]] D=[[-1,1],[-1]]

Le joueur 1 clique sur le bouton 2.
M=[[1,0,-1],
  [0,-1,0],
  [0,0,0]]

L=[[1,-1],[-1],[ ]] C=[[1],[-1],[-1]] D=[[-1,1],[-1,-1]]
```

Le joueur 2 clique sur le bouton 8.

```
M=[[1,0,-1],
  [0,-1,0],
  [0,0,1]]

L=[[1,-1],[-1],[1]] C=[[1],[-1],[-1,1]] D=[[-1,1],[-1,-1]]
```

Le joueur 1 clique sur le bouton 6.

```
M=[[1,0,-1],
  [0,-1,0],
  [-1,0,1]]

L=[[1,-1,-1],[-1],[1]] -> la première sous-liste a une taille 3,
la somme de ses éléments n'est ni -3 ni 3. Ce n'est pas
une ligne "gagnante".
C=[[1,-1],[-1],[-1,1]]
D=[[-1,1],[-1,-1,-1]] -> la deuxième sous-liste a une taille
3, la somme de ses éléments est -3 donc la diagonale est
"gagnante" pour le joueur 1.
```

update est appelée par la fonction **game**.

- Fonction **check(liste)**

La fonction **check** qui prend en paramètre une sous-liste de L, C ou D vérifie si un joueur est gagnant. Elle est appelée par la fonction **update**.

- Fonction **reset()**

La fonction **reset** qui remet la grille à l'état initial.

- La partie graphique constitue le corps du programme. Elle interagit avec les fonctions précédentes.

Travail à faire

Etape 1:

Compléter la partie graphique pour que lors de l'exécution la fenêtre avec les boutons apparaisse.

Etape 2:

Compléter une partie de la fonction **game** et vérifier que lors du clic sur le bouton le symbole du joueur apparaît.
`nom_du_bouton.configure(text= nouveau_texte)` permet de modifier le texte apparaissant sur la bouton.

Etape 3:

Compléter la fonction **check**.

`check([-1,1,1])` renvoie False

`check([-1,-1,-1])` renvoie True et `code_winner=1`

`check([1,1,1])` renvoie True et `code_winner=2`

`check([-1,1,-1])` renvoie False et `code_winner=0`

`check([-1,1])` renvoie False et `code_winner=0`

`check([1])` renvoie False et `code_winner=0`

Etape 4:

Compléter la fonction **update**.

Etape 5:

Compléter entièrement la fonction **game**.

Etape 6:

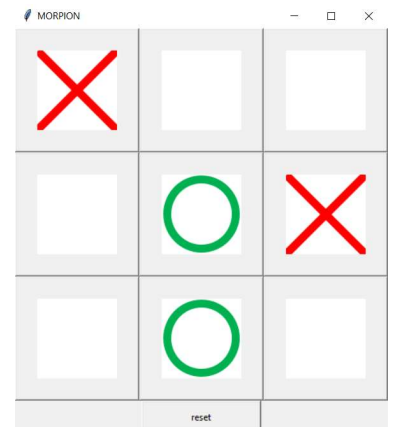
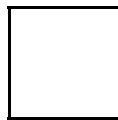
Compléter la fonction **reset**.

Pour aller plus loin

■ Ajouter des images aux boutons

Sur les boutons, remplacer les symboles des joueurs par des images.

Vous avez à votre disposition les images: blanc.png, croix.png et rond.png.



Les méthodes:

`nom_de_l_image = PhotoImage(file='nom_du_fichier')` récupère l'image.

`Button(fenetre, image= nom_de_l_image, width= largeur en pixels, height= hauteur en pixels)`

`nom_du_bouton.configure(image= nom_de_l_image)` permet de changer l'image.

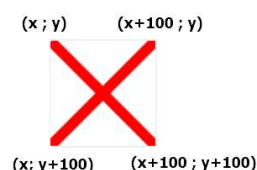
■ Remplacer la grille de boutons par un panneau de dessin

On commencera par créer une fonction **gamegrid()** qui crée la grille vierge. Chaque carré aura une taille de 100 pixels.

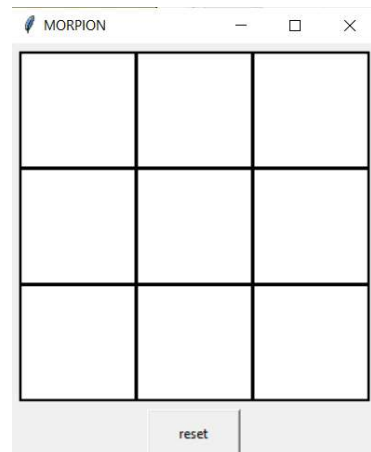
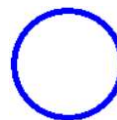
Créer une fonction **croix(i,j)** qui dessine une croix dans la case (i,j)

(i,j) sont les coordonnées de la case

(x;y) les coordonnées en pixels du point en haut à gauche de la case. (cf figure)



Créer une fonction **rond(i,j)** qui dessine un rond.



Les méthodes à votre disposition:

`panneau_de_dessin = Canvas(fenetre, bg="couleur", width= largeur en pixels, height= hauteur en pixels)` crée un panneau de dessin

`panneau_de_dessin.create_line(xA, yA, xB, yB, width = largeur de la ligne, fill = 'couleur_de_remplissage')`

`panneau_de_dessin.create_oval(xA, yA, xB, yB, width = largeur de la ligne, outline = 'couleur de la ligne extérieur')`

`panneau_de_dessin.delete(ALL)` efface tous les éléments du panneau.

<http://tkinter.fdex.eu/doc/caw.html>

`panneau_de_dessin.bind('<Button-1>', nom_de_la_fonction)` pour que le panneau de dessin interagisse avec la souris.

`nom_de_la_fonction` a pour paramètre **evt** qui récupère la souris.

evt.x récupère l'abscisse de la souris (en pixel)

Remarque: on peut bien sûr changer le nom du paramètre **evt**.

<http://tkinter.fdex.eu/doc/event.html>