**I - Programmation d'un jeu morpion en Python**

**Mise en forme du jeu :**

1. Création des joueurs
2. Création du plateau de jeu
3. Vérification du gagnant
4. Limites des joueurs
5. Lancement du jeu
6. Fin du jeu

**Utilisation du Framework:** VSCode

**Etapes de la création du jeu morpion sans interface graphique:**

On définit **les joueurs** avec leurs **symboles**, , **J1=X**, **J2=O.**

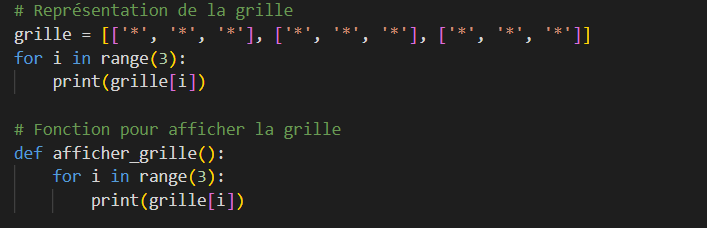
- Classer les joueurs par la fonction define:

**def joueur1joue()**

**def joueur2joue()**

De plus, on crée la grille pour que les joueurs puissent déposer leurs symboles approprié et que via le terminal on observe le rendu de la grille.

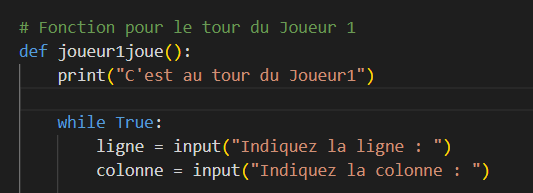
**- Modélisation de la grille:**



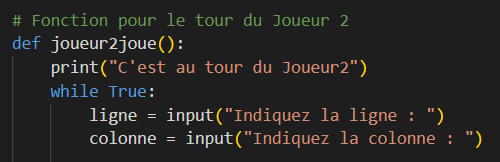
**NB:** la position de notre liste est indexé de 0 à 2, dans Python, on crée la grille sous forme de liste et on utilise une **boucle FOR** qui va réitérer trois fois la liste et ensuite l'affiché via print(nomDeLaListe(Nomde Variable).

nomdeVariable conoté **'I'.**

Egalement, on s'intéresse au bloc du joueur, ci-dessous on trouvera les éléments du Joueur(1):



Pareillement pour le **Joueur2**, on inclut le même nom de la ligne & de la colonne, car il se forme une et unique grille. Cependant, on séparera bien la fonction du Joueur 1 et du Joueur 2,



**VERIFICATION DU GAGNANT**

On va crée une fonction qui va permettre de verifier si il y a une victoire. Pour cela on utilise la commande "def" qui va définir une fonction qui s'appellera "verif\_victoire"

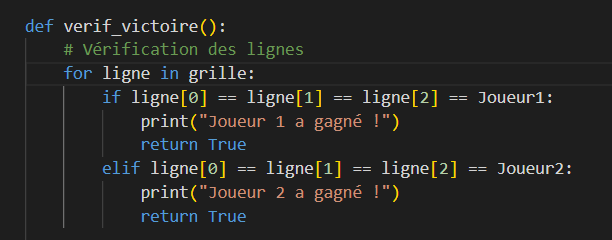
**Verification des lignes:**

La boucle **for ligne in grille** lis à travers chaque ligne de la grille (il y en a 3)

Ensuite, elle vérifie si tous les éléments de cette ligne sont égaux au Joueur1. Si oui, le joueur 1 à gagné en remplissant une ligne horizontale.

Elle fait aussi la vérification pour le joueur2.

Si l'un des joueurs à gagné, la fonction retourne True, ce qui signifie que le jeu est terminé.



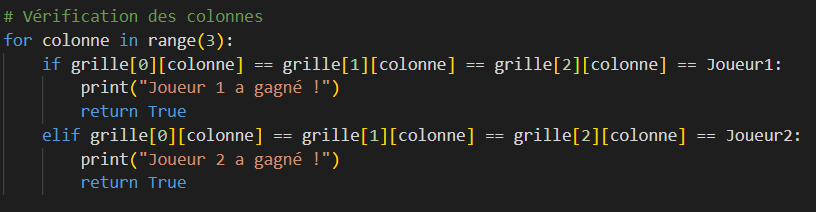
**Verification des colonnes:**

Ensuite, la boucle **for colonne in range(3)** lis à travers chaque colonne de la grille(il y en a3)

Elle vérifie si tous les éléments dans cette colonne sont égaux à Joueur1. Si c'est le cas, le Joueur1 a gagné en remplissant une colonne verticale.

Elle fait la même vérification pour le Joueur2.

Encore une fois, si l'un des joueurs à gagné, la fonction retourne True.



**Verification des diagonales:**

Enfin, la fonction vérifie les deux diagonales. **La première condition** vérifie si les élements de la diagonale supérieure gauche à la diagonale inférieure droite (de gauche à droite) sont tous égaux à Joueur1. **La deuxième condition** vérifie la même chose poiur le Joueur2.

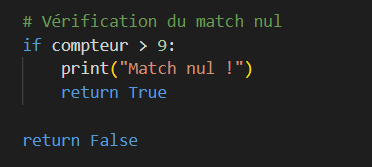
**La troisième condition** vérifie si les élèments de la diagonale supérieure droite à la diagonale intérieure gauche (de droite à gauche) sont tous égaux à Joueur1. **La quatrième condition** vérifie la même chose pour le Joueur2.

Si l'une de ces conditions est vérifiée, cela signifie que l'un des joueurs à gagné en remplissant une diagonale.

**Verification du match nul:**

Enfin, après avoir vérifié les lignes, les colonnes et les diagonales, la fonction vérifie si le nombre de coups joués **(compteur) dépasse 9**. Si c'est le cas, la grille à été remplie sans aucun gagnant.

Si un match nul est détecté, la fonction retourne True.



**Retour par défaut:**

Si aucune des conditions de victoire ou de match nul n'est vérifiée, la fonction retourne False, indiquant que la partie se poursuit...

**LES LIMITES DES JOUEURS**

Le choix des conditions telles que: if, elif et else ont pour objectif de prendre des décisions, dans notre cas, on s'intéresse sur les limites du jeu.

Si, on souhaite qu'un des joueurs tape un symbole; que ça soit un chiffre ou une lettre > '2' ou < '0' concernant les lignes et les colonnes, il ne pourra pas car on précise bien que la valeur attendu sera de 0 à 2.

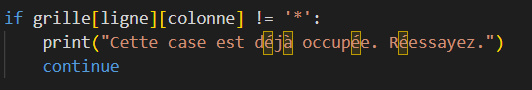
**De cette manière, la condition s'emploie:**



**print("Les coordonnées sont incorrectes.")**

**continue.**

D'autant plus que, si un des joueurs tentent de taper un symbole(**VALEUR OU LETTRE**) dans la case occupé par un des joueurs ils n'y pourront pas car on précise bien la fonction Si c'est différent de '\*'', alors la cordonné est fausse ce qui oblige au joueur 1 de retenter et de placer son symbole dans une case AUTRE que le joueur numéro 2.



Applicable que ça soit pour le Joueur 1 ou le Joueur 2, ils auront les mêmes conditions.

**LANCEMENT DU JEU ET FIN DE PARTIE.**

**compteur = 1:** Ce compteur est utilisé pour suivre le nombre de tours joués.

**enCours = True:** Ce qui signifie que le jeu est en cours.

**La boucle while enCours and compteur <= 9** : C'est une boucle qui continue à s'exécuter tant que le jeu est en cours (enCours est True) et que le nombre de tours (compteur) est inférieur ou égal à 9.

**À chaque tour de la boucle,** le programme vérifie si c'est le tour du joueur 1 ou du joueur 2 en utilisant l'opération compteur % 2 == 1. Si compteur est impair, c'est le tour du joueur 1, sinon c'est le tour du joueur 2.

**Si c'est le tour du joueur 1**, la fonction joueur1joue() est appelée pour permettre au joueur 1 de jouer. Si c'est le tour du joueur 2, la fonction joueur2joue() est appelée pour permettre au joueur 2 de jouer.

Ensuite, la fonction **afficher\_grille()** est appelée pour afficher l'état actuel du jeu.

**Le programme vérifie s'il y a un gagnant** en appelant **la fonction verif\_victoire().** Si un joueur a gagné, la boucle s'arrête avec l'instruction break, et le jeu se termine.

Après chaque tour, le compteur est augmenté de 1 avec l'instruction compteur += 1.

Une fois que la boucle se termine (soit parce qu'il y a un gagnant, soit parce que 9 tours ont été joués), le message "Fin de la partie." est affiché.

