

Compte rendu : Projet – Les tours de Hanoi.

Remarque :

Vous trouvez le programme principal de chaque fonction en dessous de sa définition.

Tous les programmes principaux sont en commentaire.

Partie A :

Dans cette partie, nous avons défini les fonctions de base pour notre jeu.

La fonction `init(n)` :

Elle crée le plateau initial de n disque.

Exemple : pour n = 3 ;

```
>>> init (3)
```

```
[[3, 2, 1], [], []]
```

La fonction `nombre_disques (plateau, numtour)` :

Elle renvoie le nombre des disques dans la tour indiquée en argument.

Exemple :

```
>>> plateau = [[1], [3, 2], []]
```

```
>>> nombre_disques (plateau, 1)
```

```
2
```

La fonction `disque_superieur (plateau, numtour)` :

Elle renvoie le numéro du disque le plus grand dans la tour indiquée en argument.

Exemple :

```
>>> plateau = [[4, 1], [5, 3, 2], []]
```

```
>>> disque_superieur(plateau, 2)
```

```
5
```

La fonction `position_disque (plateau, numdisque)` :

Cette fonction renvoie la position du disque demandé (numéro du disque en argument), La position est une des trois tours (0, 1 ou 2).

Exemple :

```
>>> plateau = [[12,11,10], [8,7,6], [5,4,3,2,1]]
```

```
>>> position_disque (plateau, 5)
2
```

La fonction `verifier_deplacement (plateau, nt1, nt2)` :

Elle vérifie si le déplacement d'un tour (nt1) à un autre (nt2) est possible ou pas, elle renvoie 'True' si c'est possible et 'False' sinon.

Exemple :

```
>>> plateau = [[12, 11, 10], [8, 7, 6], [5, 4, 3, 2, 1]]
>>> verifier_deplacement (plateau, 1, 0)
True
```

La fonction `verifier_victoire (plateau, n)` :

Cette fonction vérifie si tous les disques sont dans la dernière tour (2) triées dans l'ordre décroissant, si oui elle renvoie 'True' sinon 'False'.

Exemple :

```
>>> plateau = [[], [], [12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1]]
>>> verifier_victoire (plateau, 12)
True
>>> plateau2 = [[], [2], [12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 1]]
>>> verifier_victoire (plateau2, 12)
False
```

Partie B :

Cette partie était basée sur l'animation -sur 'turtle'- de jeu (dessin de disque, plateau... etc.)

Pour cette partie, nous mettrons des photos des résultats finals avec plusieurs nombres de disque (3, 5 et 10) après la dernière fonction.

La fonction `disque (L, M)` :

Elle dessine un disque de longueur L et largeur M (pour notre cas L dépend de nombre des disques et M = 20).

Remarque :

Cette fonction dessine des disques colorer, donc nous avons défini une fonction jumelle `white_disque` dessine des disques sans couleur qui nous servira à effacer des disques prochainement.

La fonction `dessine_plateau(n)` :

Elle dessine le plateau du jeu. Ces paramètres dépendent du nombre des disques.

La fonction `dessine_disque(nd, plateau, n)` :

Elle reçoit le numéro de disque, le plateau et le nombre des disques, en utilisant la fonction de la partie A `position_disque`, et dessine ce disque dans sa position dans le plateau.

La fonction `efface_disque(nd, plateau, n)` :

Cette fonction efface le disque du plateau.

Remarque :

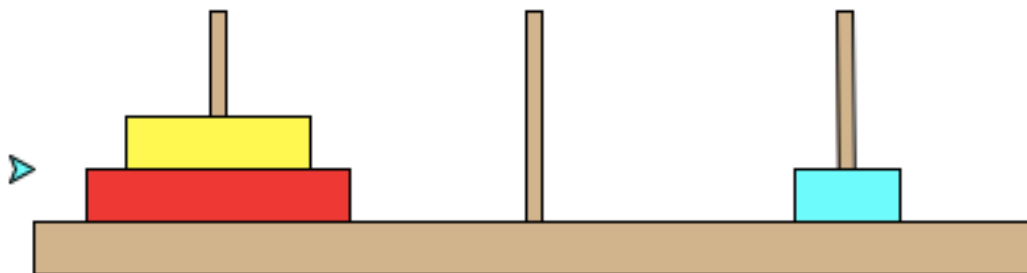
Comme la fonction `disque`, nous avons fait 2 versions de cette fonction, la deuxième version s'appelle `efface_disque2`, parce que la première fonction redessine le plateau à chaque efface, donc elle prend plus de temps que la deuxième qui nous servira dans la fonction `efface_tout`.

Les fonctions `dessine_config(plateau, n)` et `efface_tout(plateau, n)` :

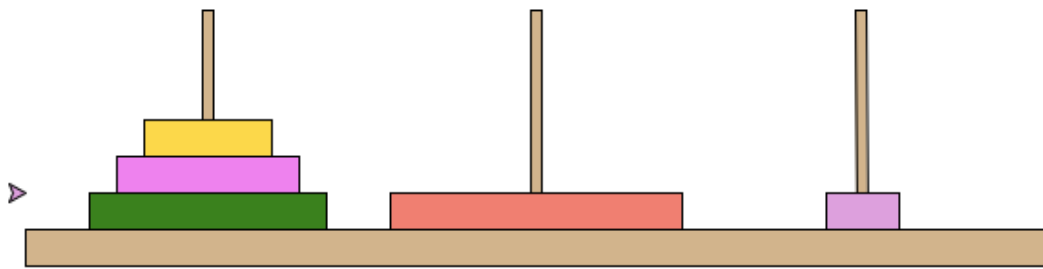
Elles dessinent et effacent la configuration du plateau respectivement.

Exemples :

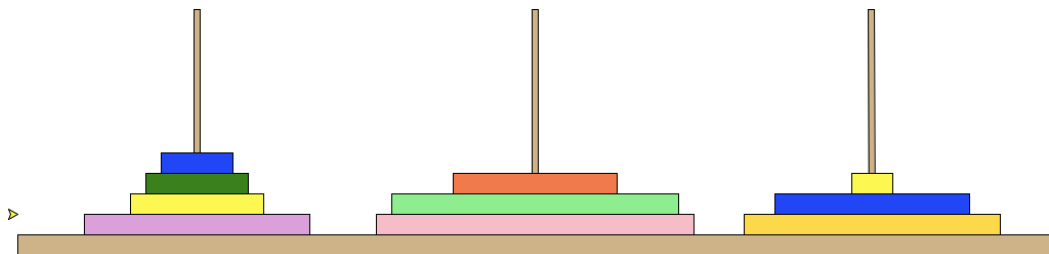
```
>>> plateau = [[3,2], [], [1]]
>>> dessine_plateau(3)
>>> dessine_config(plateau,3)
```



```
>>> plateau = [[4, 3, 2], [5], [1]]
>>> dessine_plateau(5)
>>> dessine_config(plateau, 5)
```

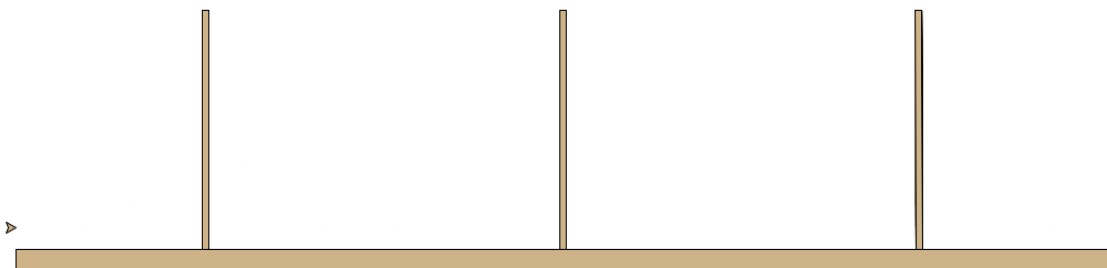


```
>>> plateau = [[7, 4, 3, 2], [10, 9, 5], [8, 6, 1]]
>>> dessine_plateau (10)
>>> dessine_config (plateau,10)
```



Pour effacer ces disques on utilise la fonction `efface_tout`, mais il faut faire attention à la couleur choisie dans la fonction `efface_disque`, elle doit être la même que celle d'arrière-plan.

```
>>> efface_tout (plateau, 10)
```



Remarque :

Pour les exemples on a choisi la couleur blanche comme arrière-plan mais pour le résultat final on préfère le rose.

Partie C :

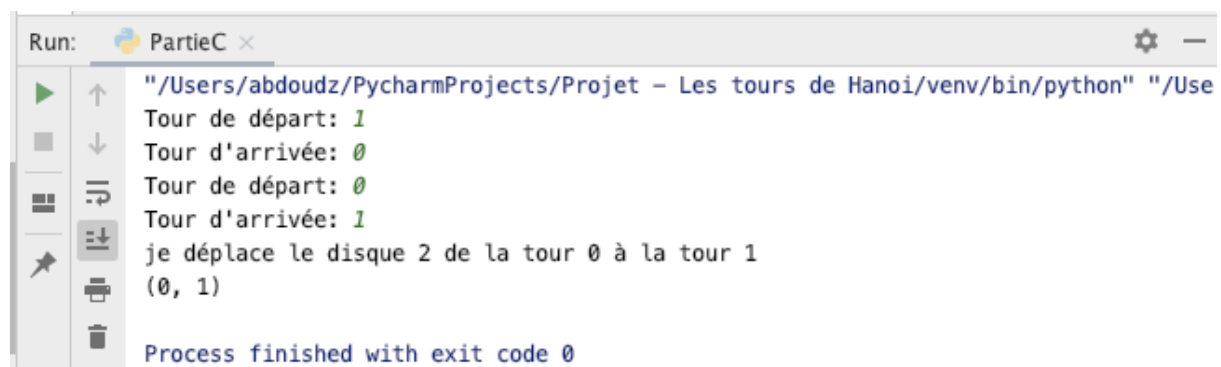
Dans cette partie on entre dans le cœur du sujet, jouer.

La fonction **lire_coords (plateau)** :

Cette fonction demande de l'utilisateur d'entrer les deux tours de déplacement (0, 1 ou 2), la fonction vérifie si les valeurs entrées sont parmi les tours sinon elle redemande. Si le déplacement est autorisé d'après les règles du jeu elle renvoie les deux tours, sinon elle redemande au joueur de réessayer.

Exemple :

```
>>> plateau = [[6, 5, 3, 2], [7, 4], [8, 1]]
>>> lire_coords (plateau)
```

A screenshot of a PyCharm Run console window. The window title is "Run: PartieC x". The console output shows the execution of the Python script. It displays the input for the lire_coords function: "Tour de départ: 1", "Tour d'arrivée: 0", "Tour de départ: 0", "Tour d'arrivée: 1". Then it shows the action: "je déplace le disque 2 de la tour 0 à la tour 1" and the coordinates "(0, 1)". At the bottom, it says "Process finished with exit code 0".

```
Run: PartieC x
"/Users/abdoudz/PycharmProjects/Projet - Les tours de Hanoi/venv/bin/python" "/Use
Tour de départ: 1
Tour d'arrivée: 0
Tour de départ: 0
Tour d'arrivée: 1
je déplace le disque 2 de la tour 0 à la tour 1
(0, 1)
Process finished with exit code 0
```

La fonction **jouer_un_coup (plateau, n)** :

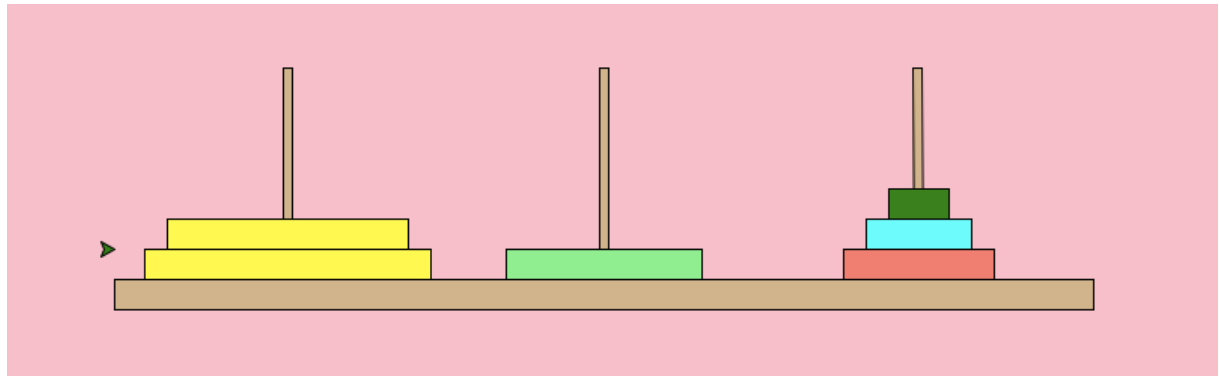
Cette fonction reçoit un plateau et le nombre des disques et utilise la fonction **lire_coords** pour demander au joueur de déplacer un disque et elle le déplace en animation en utilisant les fonctions **efface_disque** et **dessine_disque** et renvoie la nouvelle composition du plateau.

Exemple :

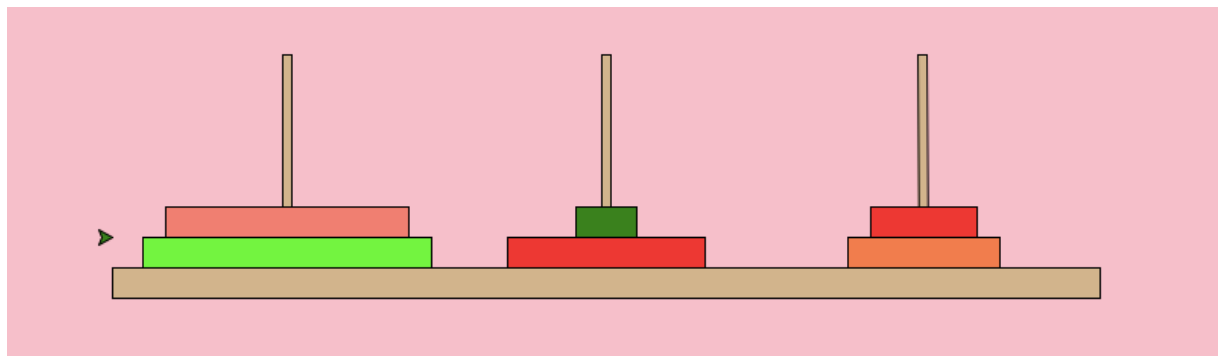
```
>>> plateau = [[6, 5], [4], [3, 2, 1]]
>>> # on dessine le plateau d'abord.
>>> dessine_plateau (6)
>>> dessine_config (plateau, 6)
>>> jouer_un_coup (plateau, 6)
La console nous affiche ça :
```

```
PartieC x
"/Users/abdoudz/PycharmProjects/Projet - Les tours de Hanoi/venv/bin/python" "/Use
Tour de départ: 2
Tour d'arrivée: 1
je déplace le disque 1 de la tour 2 à la tour 1
('', [[6, 5], [4, 1], [3, 2]])
```

Après il dessine le plateau :



il devient avec la fonction `jouer_un_coup` :



La fonction `boucle_jeu (plateau, n)` et Le programme principal :

Cette fonction joue une partie entière finie par une victoire ou un abandon, elle utilise la fonction `jouer_un_coup`.

Exemple :

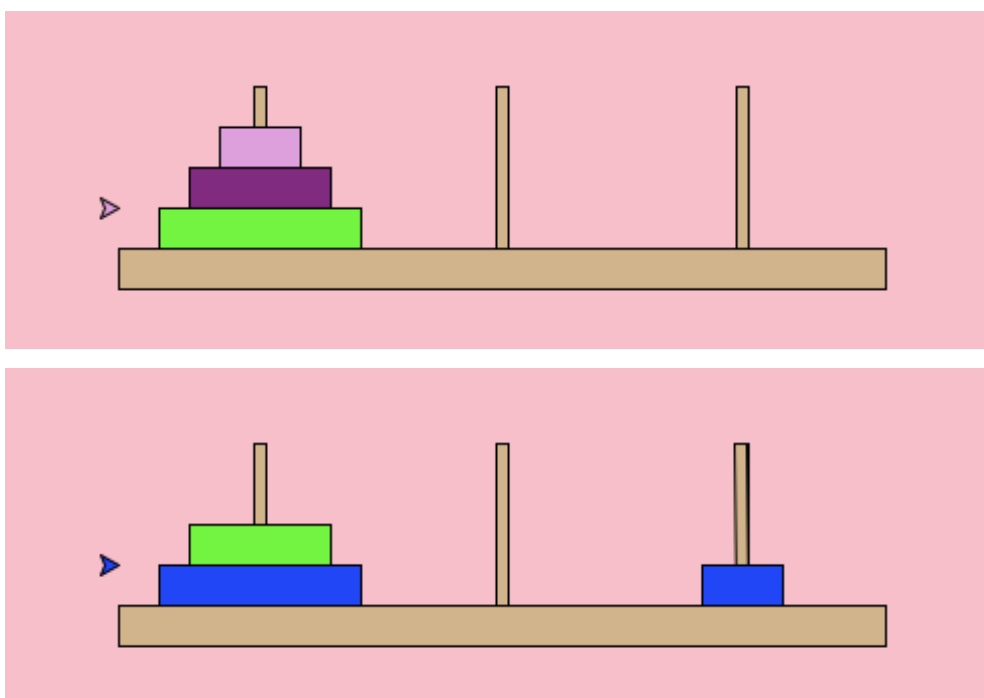
Une partie avec 3 disques :

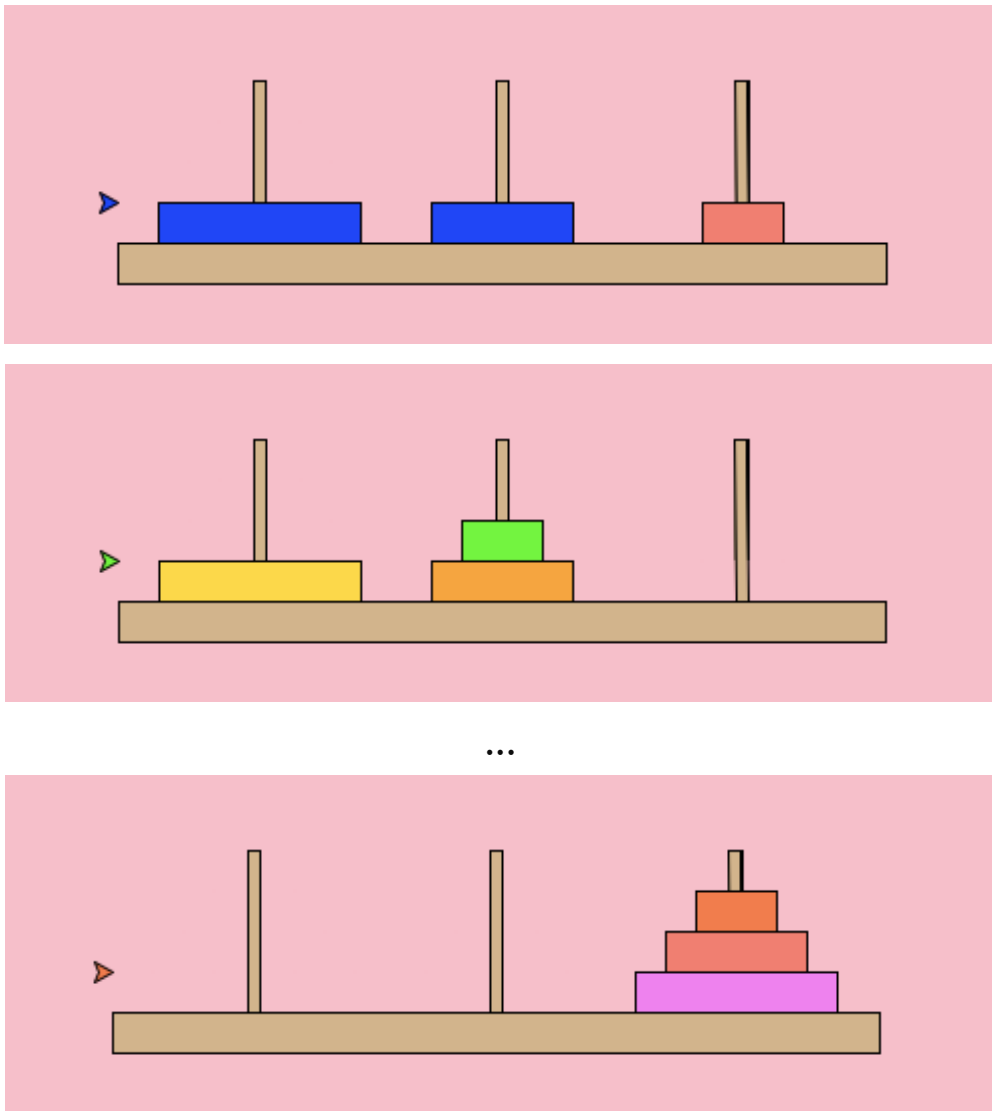
En utilisant le programme principal, on obtient :

Cas de victoire :

```
PartieC x
"/Users/abdoudz/PycharmProjects/Projet - Les tours de Hanoi/venv/bin/python" "/Use
Bienvenue dans les Tours de Hanoi
Combien de disques?3
remarque :
vous pouvez abandonner à tout moment en entrant "-1"
Tour de départ: 0
Tour d'arrivée: 2
je déplace le disque 1 de la tour 0 à la tour 2
Tour de départ: 0
Tour d'arrivée: 1
je déplace le disque 2 de la tour 0 à la tour 1
Tour de départ: 2
Tour d'arrivée: 1
je déplace le disque 1 de la tour 2 à la tour 1
Tour de départ: 0
Tour d'arrivée: 2
je déplace le disque 3 de la tour 0 à la tour 2
Tour de départ: 1
Tour d'arrivée: 0
je déplace le disque 1 de la tour 1 à la tour 0
Tour de départ: 1
Tour d'arrivée: 2
je déplace le disque 2 de la tour 1 à la tour 2
Tour de départ: 0
Tour d'arrivée: 2
je déplace le disque 1 de la tour 0 à la tour 2
Vous avez gagné!
Félicitation
```

Et pour l'animation :





Cas d'abandon :


```
PartieC x
"/Users/abdoudz/PycharmProjects/Projet - Les tours de Hanoi/venv/bin/python" "/Use
Bienvenue dans les Tours de Hanoi
Combien de disques?3
remarque :
vous pouvez abandonner à tout moment en entrant "-1"

Tour de départ: 0
Tour d'arrivée: 2
je déplace le disque 1 de la tour 0 à la tour 2
Tour de départ: 0
Tour d'arrivée: 1
je déplace le disque 2 de la tour 0 à la tour 1
Tour de départ: -1
vous souhaitez abandonner (o/n) ? o
T.T
Abandon de la partie après 2 coups.
Au revoir
```

Partie D :

Nous souhaitons maintenant de rajouter l'option d'annulation d'un coup.

La fonction **dernier_coup** (coups, drn_c) :

Cette fonction reçoit un dictionnaire et le numéro du dernier coup et renvoie les tours de déplacement du dernier disque déplacer (tour de départ, tour d'arriver).

Exemple :

```
>>> coups = {1: [[3, 2], [], [1]], 2: [[3], [2], [1]], 3: [[3, 2], [], [1]]}
>>> dernier_coup(coups,3)
(1, 0)
```

La fonction **annuler_dernier_coup** (coups, drn_c) :

En utilisant la fonction **dernier_coup** on récupère les deux tours du dernier déplacement alors on peut modifier le dictionnaire reçu en paramètre et annuler ce coup. Et renvoyer le nouveau dernier coup.

Exemple :

```
>>> coups = {1: [[3, 2], [7, 4, 1], [6, 5]], 2: [[3], [7, 4, 1], [6, 5, 2]], 3: [[3], [7, 4], [6, 5, 2, 1]]}
>>> annuler_dernier_coup(coups,3)
({1: [[3, 2], [7, 4, 1], [6, 5]], 2: [[3], [7, 4, 1], [6, 5, 2]]}, 2)
```

La fonction **boucle_jeu2 (plateau, n)** :

C'est la même fonction de la partie D mais nous l'avons amélioré.

On a rajouté un compteur de secondes en utilisant la fonction **time ()** du module **time** (qui nous sert à la partie E), et l'option essentielle de cette partie 'l'annulation d'un coup' avec la fonction définie précédemment.

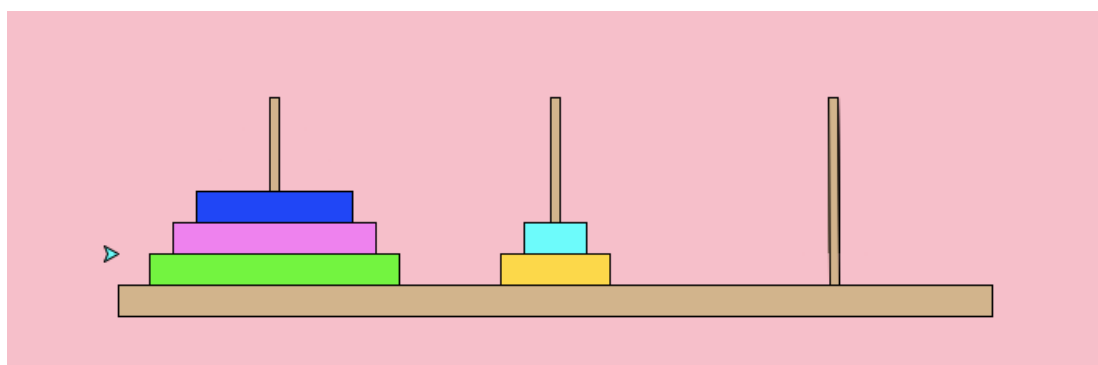
Exemple :

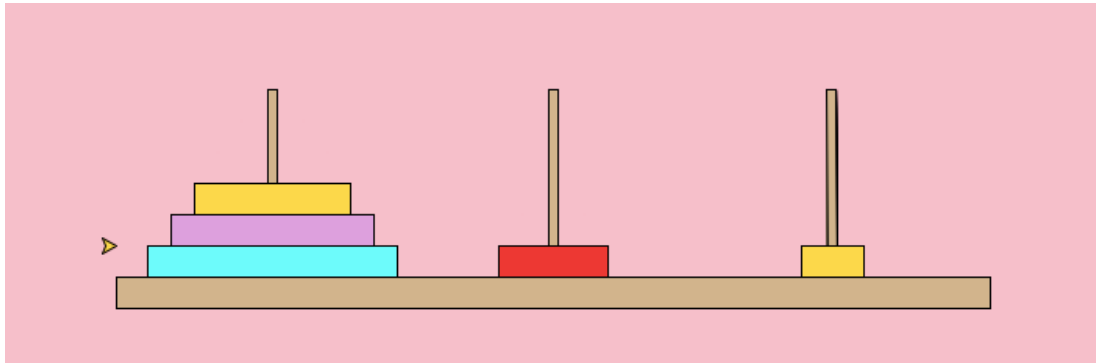
On lancera une partie de 5 disques.

En utilisant le programme principal écrit avant :

```
Run: PartieD x
"/Users/abdoudz/PycharmProjects/Projet - Les tours de Hanoi/venv/bin/python" "/Use
Bienvenue dans les Tours de Hanoi
Combien de disques?5
ramarque :
pour abandonner ou annuler un coup entrez: "-1"
Tour de départ: 0
Tour d'arrivée: 2
je déplace le disque 1 de la tour 0 à la tour 2
Tour de départ: 0
Tour d'arrivée: 1
je déplace le disque 2 de la tour 0 à la tour 1
Tour de départ: 2
Tour d'arrivée: 1
je déplace le disque 1 de la tour 2 à la tour 1
Tour de départ: -1
vous souhaitez abandonner ou annuler ? annuler
Tour de départ: |
```

Et pour l'animation ça donne :





Partie E :

La fonction **enregistrer** (**nom_joueur**, **nb_disques**, **nb_coups**, **duree**) :

Elle reçoit le nom du joueur, nombre de disques, nombre de coups et la durée du jeu récupérer par la fonction **boucle_jeu2** et elle crée un fichier '**Les tours de Hanoi.txt**' qui contient un tableau des parties jouer.

Exemple :

```
>>> enregistrer('Emilia', 3, 8, 345)
>>> enregistrer('Abdessalem', 9, 550, 970)
>>> enregistrer('Ludovic', 10, 1040, 1878)
>>> enregistrer('Sophie', 6, 65, 525)
>>> fichier = open('Les tours de Hanoi.txt')
>>> partie = fichier.read()
>>> print(partie)
```

Nom du joueur	Nombre de disques	Nombre de coups	La durée de la partie (secondes)
Emilia	3	8	345
Abdessalem	9	550	970
Ludovic	10	1040	1878
Sophie	6	65	525

La fonction **meilleur_score** (**nom_fichier**) :

Elle prend un fichier, l'ouvre et classe les parties jouer selon leur pourcentage (varie en variant le nombre de disques et les coups jouer), plus que la différence (pourcentage - 100) est petite plus que le score est meilleur.

Exemple :

(La suite du programme que nous avons utilisé avant)

```
>>> meilleur_score ('Les tours de Hanoi.txt')
```

```
Ludovic 10 1040 1878  
Sophie 6 65 525  
Abdessalem 9 550 970  
Emilia 3 8 345  
None
```

Si on calcule les pourcentages, on trouvera que le joueur Ludovic à le meilleur pourcentage (le plus proche de 100).

Programme principal :

Nous avons rajouté une option d'entrer le nom en cas de victoire, pour noter cette partie dans le fichier, en utilisant la fonction **enregistrer**.

Exemple :

On teste avec une partie de 2 disques.

On exécute le programme principal après :

```
>>> fichier = open('Les tours de Hanoi.txt')
```

```
>>> partie = fichier.read()
```

```
>>> print (partie) #ça nous donne les 6 dernières lignes.
```

```

Bienvenue dans les Tours de Hanoi
Combien de disques?2
remarque :
pour abandonner ou annuler un coup entrez: "-1"
Tour de départ: 0
Tour d'arrivée: 1
je déplace le disque 1 de la tour 0 à la tour 1
Tour de départ: 0
Tour d'arrivée: 2
je déplace le disque 2 de la tour 0 à la tour 2
Tour de départ: 1
Tour d'arrivée: 2
je déplace le disque 1 de la tour 1 à la tour 2
Vous avez gagné!
Félicitation
entrez votre nom: Nezuko
None
C'est noté! Merci, Au revoir!

```

Nom du joueur	Nombre de disques	Nombre de coups	La durée de la partie (secondes)
Emilia	3	8	345
Abdessalem	9	550	970
Ludovic	10	1040	1878
Sophie	6	65	525
Nezuko	2	3	28

La fonction **moyenne_temps(nom_fichier)** :

Cette fonction prend en argument le nom du fichier et renvoie affiche la durée moyenne de chaque joueur.

Exemple :

Cette fois ci on a gardé les résultats précédents et on a modifier la durée de chaque joueur pour recevoir la moyenne de chacun.

```

la duree moyenne par partie de Emilia est: 245.0
la duree moyenne par partie de Abdessalem est: 758.0
la duree moyenne par partie de Ludovic est: 1111.5
la duree moyenne par partie de Sophie est: 322.5
la duree moyenne par partie de Nezuko est: 28.0

```

Remarques :

Vous pouvez vérifier en modifiant les durées de chaque joueur.

On a rajouté une ligne au début du programme :

« os.remove('Les tours de Hanoi.txt') »

Qui nous sert à initialiser le fichier à chaque exécution, donc pour rajouter des nouvelles parties avec des nouvelle durées il faut le mettre en commentaire.

La fonction **classement_duree(nom_fichier)** :

Cette fonction joue le même rôle que la fonction `meilleur_score`, mais elle classe les joueurs selon leur durée.

Exemple :

```
>>> print (classement_duree ('Les tours de Hanoi.txt'))
```

```
1 . Sophie - 120 secondes
2 . Emilia - 145 secondes
3 . Ludovic - 345 secondes
4 . Abdessalem - 546 secondes
None
```

Partie F :

La fonction `hanoi` (`n`, `source = 0`, `auxiliary = 1`, `destination = 2`, `deplacements=[]`) :

Cette fonction est basée sur l'algorithme qui résout le problème de la tour de Hanoi en utilisant la fonction récursive.

Ces étapes sont :

- Déplacer `n-1` disques de la tour 0 à 1 à l'aide de la tour 2.
- Déplacer un disque de la tour 0 à 2.
- Déplacer les `n-1` disques de la tour 1 à 2 à l'aide de la tour 0.

Remarque :

On l'expliquera les détails dans la présentation.

Alors, elle renvoie une liste 'deplacements' qui a les listes des déplacements comme éléments.

La fonction `jouer_un_coup_auto` (`plateau`, `n`, `i`, `j`) :

On a juste modifier la fonction `jouer_un_coup` pour qu'elle devient automatique (au lieu de demander à l'utilisateur d'entrer les déplacements, elle utilise la fonction `hanoi` pour les récupérer).

La fonction `solution_Tour_de_Hanoi`(`plateau`, `n`):

Elle résout le problème de la tour de Hanoi automatiquement avec l'animation, quel que soit le nombre des disques `n` entré dans le programme principal.

GamePlay :

On a rajouté un fichier qui s'appelle « `GamePlay.py` », où vous pouvez jouer une partie en propre (propre = juste un programme principal +

les fonctions de la partie F aussi dedans, pour afficher ton classement).