



Les tours de Hanoï

I- Présentation

Définition wikipédia :

https://fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF

Il s'agit d'un jeu que vous connaissez tous comportant trois tours, composées d'un empilage de pièces de diamètres différents.



Flickr / Photo © lilszeto

Les tours de Hanoï sont un jeu de réflexion imaginé par le mathématicien français Édouard Lucas, et consistant à déplacer des disques de diamètres différents d'une tour de « départ » à une tour d'« arrivée » en passant par une tour « intermédiaire », et ceci en un minimum de coups, tout en respectant les règles suivantes :

- on ne peut déplacer plus d'un disque à la fois ;
- on ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

II- L'interface graphique

On vous fournit une interface graphique à travers la classe `GUIhanoi` que vous pouvez importer depuis le module fourni. Si le répertoire `GUI_hanoi` est dans le même répertoire que votre fichier :

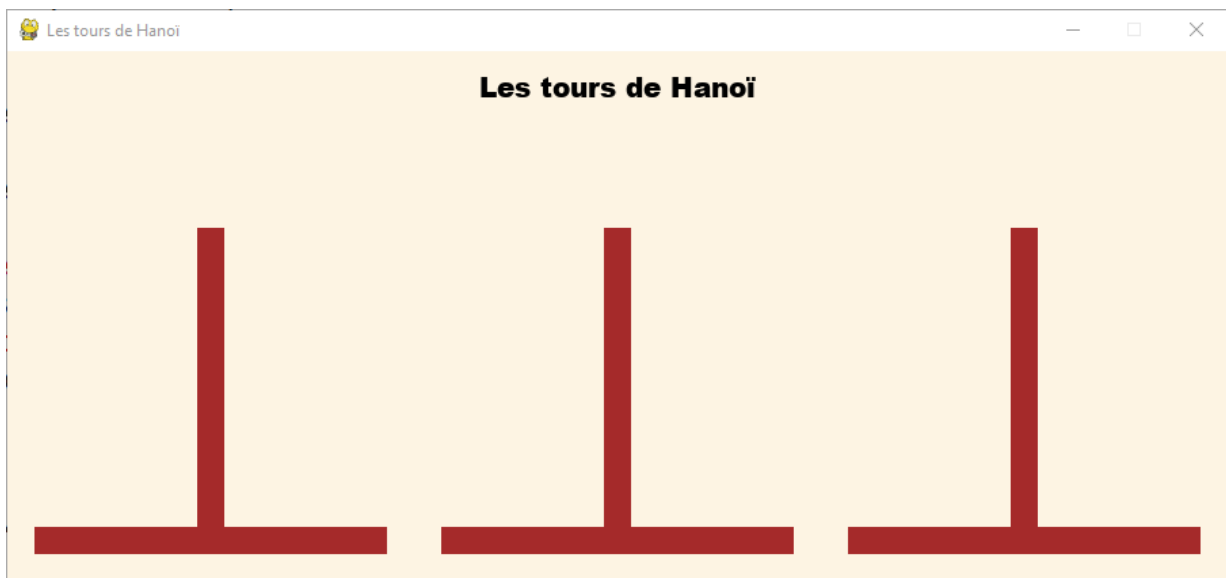
```
from GUI_hanoi .guiHanoi import GUIhanoi
```

Constructeur de la classe :

- Le constructeur prend en argument le nombre de pièces du jeu. Il permet de créer un objet graphique vierge.

Par exemple pour des tours de cinq pièces :

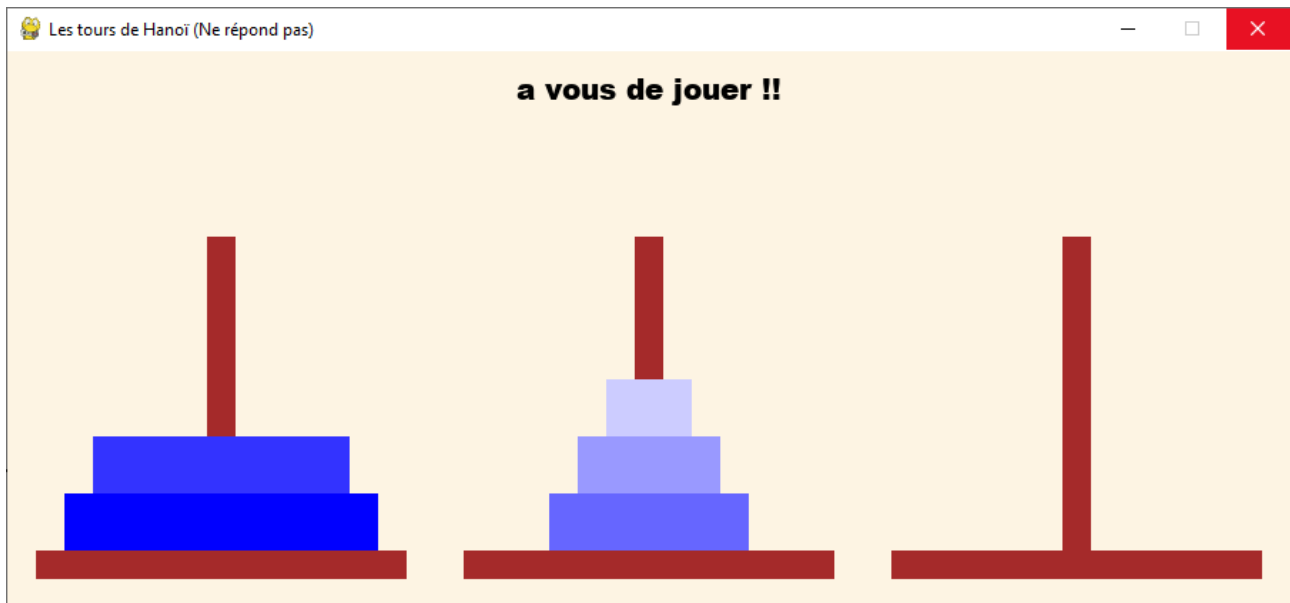
```
GUI = GUIhanoi(5)
```



- Les méthodes :
 - **refresh(g, t = "")** : Cette méthode rafraichie l'affichage du jeu conformément à la grille passée en argument.
 - **g** est une liste contenant les 3 tours. Une tour est une structure de type list ou tout autre objet qui possède les méthodes `__len__` et `__getitem__`, le rendant indexable. La première pièce d'une tour (un int > 0) est a l'index 0 de la tour.
`g[0][0]` est l'emplacement en bas de la tour de gauche ;
`g[2][4]` est la cinquième pièce empilée sur la tour de droite.
 - **t** est un texte affiché a destination du joueur, sous le score.

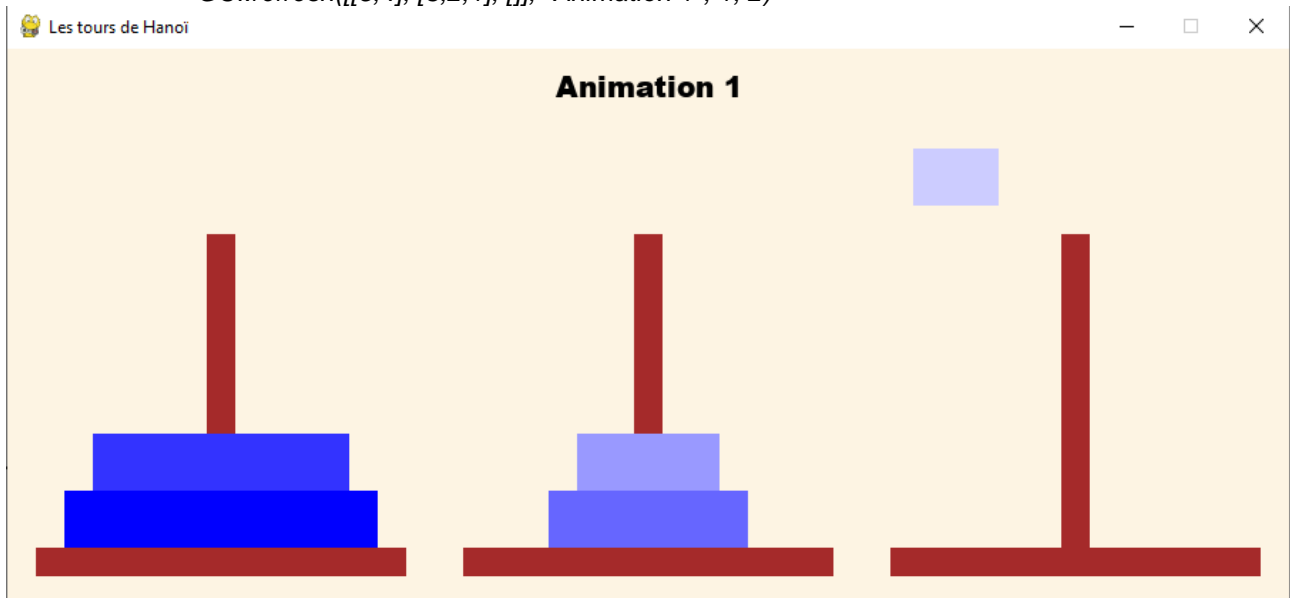
Par exemple :

```
GUI.refresh([[5,4], [3,2,1], []], "a vous de jouer !!")
```



- **refresh(g, t, colonneD, colonneA)** : C'est la même méthode que la précédente mais les deux arguments supplémentaires, si ils sont renseignés, déclenche l'animation de la pièce du haut de la colonneD vers la colonneA. Attention, g n'est pas modifiée par cette méthode, il s'agit juste d'une animation.
 - **colonneA** est le numéro de la colonne dans laquelle tombe la nouvelle pièce.
 - **colonneD** est le numéro de la colonne depuis laquelle bouge la pièce du haut si elle existe.

```
GUI.refresh([[5,4], [3,2,1], []], "Animation 1", 1, 2)
```



- **messCentre(mess)** : Cette méthode permet d'afficher le message mess plein écran, au centre.

Par exemple :

`GUI.messCentre('GAGNE !!')`



- **waitClick()** : Cette méthode attend l'action d'un joueur. Elle gère trois types d'actions :
 - **demande fermeture de la fenêtre** : fermeture propre de la fenêtre pygame et fin du programme python.
 - **click sur la fenêtre** : retourne un entier correspondant au numéro de colonne choisie. La tour de gauche porte le numéro 0.
 - **appui sur des touches spéciales** :
 - fleche RIGHT : retourne '_R';
 - fleche LEFT : retourne '_L';
 - fleche DOWN : retourne '_D';
 - fleche UP : retourne '_U';
 - touche BACKSPACE : retourne '_B';
 - touche RETURN : retourne '_E';
 - touche ESCAPE : retourne '_S';
 - **appui sur une autre touche du clavier** : retourne le caractère unicode correspondant.

Attention, une fois exécutée, on ne peut sortir de cette méthode que par l'une de ces quatre actions.