

# Cahier des charges : Tours de Hanoi

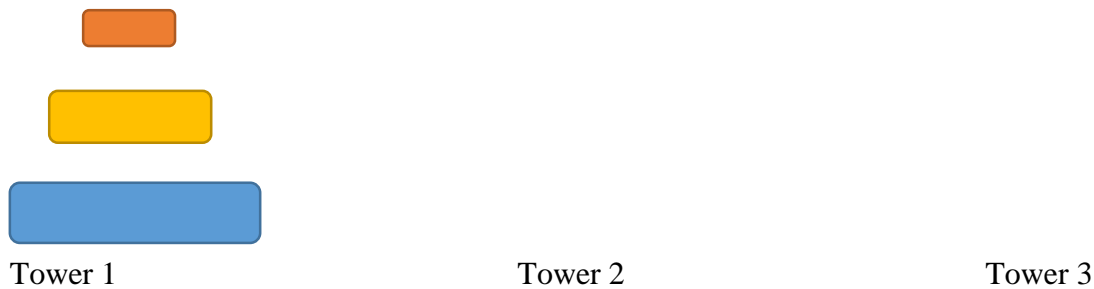
Guille Agathe  
Martinez Arnaud  
Guyot Nathan  
Remnant Fabio

Les tours de Hanoi 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 :

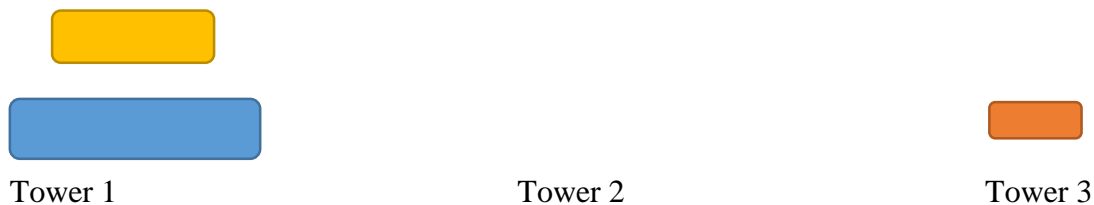
- On ne peut déplacer plus d'un disque à la fois
- On ne peut placer un disque que sur un autre disque plus grand que lui ou sur un emplacement vide.

Voici le fonctionnement du jeu avec trois disques :

## Position initiale



## Mouvement 1



## Mouvement 2



Tower 1



Tower 2



Tower 3

### Mouvement 3



Tower 1



Tower 2

Tower 3

### Mouvement 4

Tower 1



Tower 2



Tower 3

### Mouvement 5



Tower 1



Tower 2



Tower 3

### Mouvement 6



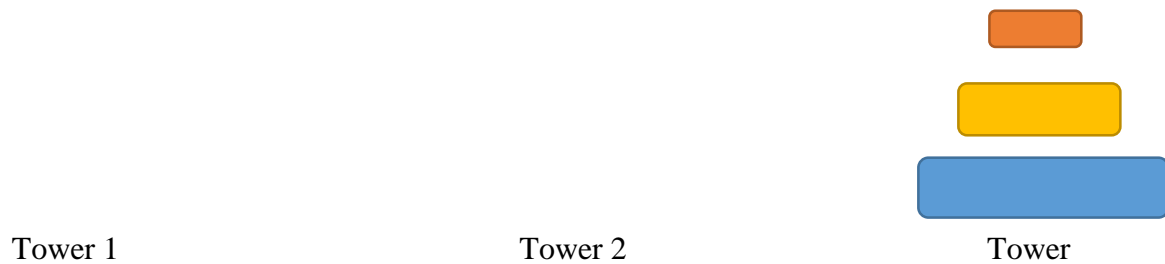
Tower 1

Tower 2



Tower 3

## Mouvement 7



Nous nous sommes reparti le projet en plusieurs parties :

Tout d'abord, Arnaud a créé le code avec Nathan, puis le design assed avec Fabio. Agathe a créé le diaporama, le cahier des charges puis le design du jeu de Hanoi et pour finir Fabio et Nathan a regardé une vidéo donnée par Arnaud pour comprendre un peu plus le fonctionnement du jeu.

## Le programme :

```
resolution.py x
resolution.py > resolution
1 def resolution():
2     def hanoi(n, de, a, par):
3
4         if n>0:
5             hanoi(n-1,de, par, a)
6             a.append(de[len(de)-1])
7             de.pop()
8             print("""
9
10            Etape suivante :
11            """)
12             print("Base 1"+str(A))
13             print("Base 2"+str(B))
14             print("Base 3"+str(C))
15             hanoi(n-1, par, a, de)
16     Disks=["Disque 5","Disque 4","Disque 3","Disque 2","Disque 1"]
17     A=[]
18     B=[]
19     C=[]
20     print("""
21     Tours de Hanoi
22     Voici les la liste des déplacements pour faire passer les disques de la tour A vers la tour C
23     """)
24     n = int(input("Nombre de disques (de 1 à 5) : "))
25     for i in range(n):
26         A.append(Disks[i+(len(Disks)-n)])
27     print(A)
28     hanoi(n,A,C,B)
29     print("Bon jeu.")
```

**Le principe du programme :**

En résumé, déplacer  $n$  disques de A vers C en passant par B consiste à :

1. Déplacer  $(n-1)$  disques de A vers B (en passant par C);
2. Déplacer le plus grand disque de A vers C ;
3. Déplacer  $(n-1)$  disques de B vers C (en passant par A).

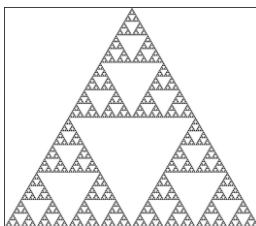
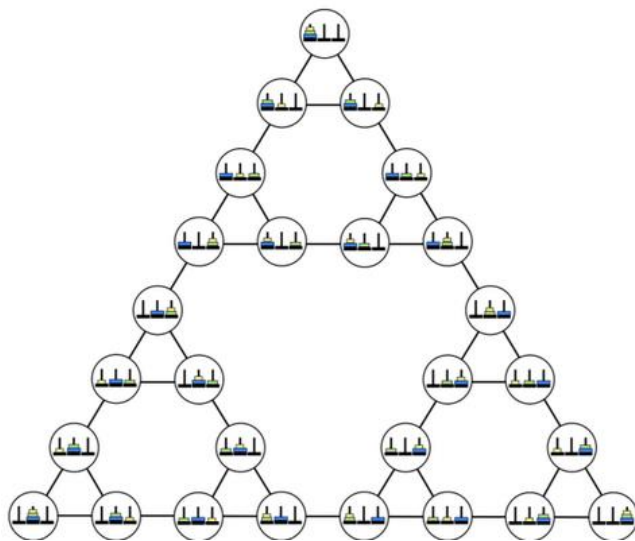
Les étapes 1 et 3 peuvent elles-mêmes se décomposer selon le même principe, sauf que les rôles des paquets sont intervertis. Par exemple, dans la première, on va de A vers B, donc forcément par l'intermédiaire de C. Voici la marche à suivre, donnée sur deux niveaux :

1. déplacer  $(n-1)$  disques de A vers B (en passant par C) ;
  - 1.1. Déplacer  $(n-2)$  disques de A vers C (en passant par B) ;
  - 1.2. Déplacer un disque de A vers B ;
  - 1.3. Déplacer  $(n-2)$  disques de C vers B (en passant par A).
2. Déplacer le plus grand disque de A vers C ;
3. Déplacer  $(n-1)$  disques de B vers C (en passant par A).
  - 3.1. Déplacer  $(n-2)$  disques de B vers A (en passant par C) ;
  - 3.2. Déplacer un disque de B vers C ;
  - 3.3. Déplacer  $(n-2)$  disques de A vers C (en passant par B).

Et ainsi de suite...

Cela nous fait penser à une lecture en largeur des feuilles d'un arbre de hauteur  $n$ .

D'autres résolutions sont faisables, en supposant qu'on ne peut pas revenir à un état déjà rencontré, voici tous les états possibles que l'on peut représenter par ce schéma :



On peut observer des similitudes avec le triangle de Sierpiński.

**Sources :**

- [www.lycee-beausejour.fr](http://www.lycee-beausejour.fr)
- [https://fr.wikipedia.org/wiki/Tours\\_de\\_Hano%C3%AF](https://fr.wikipedia.org/wiki/Tours_de_Hano%C3%AF)
- <https://youtu.be/rOnRbPKvGQg>