

# Utiliser le modèle PAC pour visualiser (puis rendre interactif) l'algorithme de résolution du problème des Tours de Hanoï

TD Master 2 GL et Mitic, ESIR 2 LSI et IN

Année 2013/2014

## Description du problème des tours de Hanoï

- N anneaux de tailles différentes sont empilés autour d'un piquet (ce qu'on appelle une tour), dans l'ordre de taille décroissant, du plus grand (tout en dessous) jusqu'au plus petit (tout au dessus).
- On dispose également de 2 autres tours, vides.
- On doit reproduire cet empilement sur l'une des autres tours (la tour d'arrivée), avec les restrictions suivantes concernant les manipulations des anneaux :
  - on ne peut manipuler qu'un anneau à la fois,
  - on ne peut manipuler qu'un anneau se trouvant au sommet d'une tour,
  - on ne peut déposer un anneau qu'au sommet de l'une des 3 tours,
  - on doit par conséquent replacer l'anneau manipulé en le déposant sur le sommet d'une tour avant de pouvoir prendre un autre anneau,
  - on ne peut pas déposer un anneau sur un anneau plus petit que lui.
- On souhaite offrir :
  1. une démonstration de cette résolution à un utilisateur en lui montrant graphiquement les déplacements successifs des anneaux pour arriver à la solution finale.
  2. une application graphique interactive permettant à un utilisateur de résoudre "à la main" le problème des tours de Hanoï.

## 1 Modélisation PAC de l'application de démonstration

### 1.1 Algorithme de résolution proposé

```
// résolution du problème : comment emmener n anneaux d'une tour de départ d,  
// vers une tour d'arrivée a, en utilisant une tour intermediaire i pour placer  
// temporairement certains anneaux  
  
procédure hanoï (Tour d, Tour i, Tour a, entier n)  
début  
  // il faut au moins un disque à déplacer...  
  si (n > 0) alors  
    // on résoud d'abord le problème des n-1 plus petits anneaux, qu'on emmène  
    // de la tour d vers la tour i, en utilisant la tour a comme tour intermédiaire  
    hanoï (d, a, i, n-1)  
    // on règle ensuite le problème du dernier anneau : on l'emmène directement  
    // de la tour d vers la tour a  
    Anneau anneau  
    anneau <- d.getSommet ()  
    d.dépiler ()  
    a.empiler (anneau)  
    // on termine en ramenant les n-1 plus petits anneaux  
    // de la tour i vers la tour a,  
    // en utilisant la tour d comme tour intermédiaire  
    hanoï (i, d, a, n-1)  
  fin si  
fin
```

## 1.2 Modélisation PAC

- Donner l'architecture PAC d'une application graphique permettant de visualiser cette résolution du problème des tours de Hanoï :
  - lister et décrire succinctement les méthodes des classes ATour et AAnneau,
  - présenter les architectures PAC (sous forme de diagrammes de classes) des agents [ATour-CTour-PTour] et [AAnneau-CAnneau-PAnneau],
  - lister les méthodes des composants contrôle et présentation de ces 2 types d'agents,
  - donner les algorithmes (en “pseudo-code”) des méthodes de ces 2 composants contrôles.
  - donner les algorithmes (en “pseudo-code”) des méthodes de ces 2 composants présentations.
- Remarques :
  - la classe AAnneau est très très simple. . .
  - les classes PAnneau et PTour pourront hériter d'une classe de l'API graphique utilisée (la classe JPanel de Swing par exemple),
  - les déplacements des anneaux devront être perçus comme des “glissements” d'une position à une autre,
  - lors d'un dépilement, on placera l'anneau au dessus de la pile dont il est dépilé,
  - lors d'un empilement, on placera l'anneau au sommet de la pile dans laquelle on va l'empiler, puis on le placera à sa place au sommet de la pile. . .
  - on déplacera un anneau de sa position courante vers une position (x, y) qui devra avoir été calculée. . .
  - les anneaux n'ont pas forcément tous la même hauteur (graphique). . .
  - la procédure de résolution du problème des tours de Hanoï présentée auparavant devrait pouvoir être réutilisée telle quelle. . .

## 2 Modélisation PAC de l'application graphique interactive

- Proposer un mode de fonctionnement offrant une rétroaction sémantique à l'utilisateur (comment les différents objets de l'application doivent-ils réagir pour guider le mieux possible l'utilisateur dans la réalisation de sa tâche?).
- Lister les modifications à apporter aux composants PAC précédents :
  - ajouts d'attributs (correspondant à des états) et de leurs méthodes de gestion dans les composants abstractions,
  - ajouts de méthodes de visualisation de ces états dans les composants de présentation,
  - ajouts de méthodes liées à la gestion des événements graphiques dans les composants de présentation,
  - ajouts de méthodes gérant l'interaction à plus haut niveau dans les composants contrôles,
  - . . .
- Donner les algorithmes des principales méthodes ajoutées à ces composants :
  - méthodes de la présentation de la tour qui vont devoir capter les événements graphiques de bas niveau,
  - méthode du contrôle de la tour appelée lors de la prise en compte d'une désignation d'anneau en vue d'un déplacement,
  - méthode du contrôle de la tour appelée lors de l'arrivée d'un anneau en vue d'un empilement,
  - méthode du contrôle de la tour appelée lors du départ d'un anneau après un survol de la tour,
  - méthode du contrôle de la tour appelée lors du dépôt d'un anneau,
  - . . .