



INSA - CVL

RAPPORT DE DÉVELOPPEMENT

## Projet application

23 mai 2015

Auteurs : AUBRY Lisa, CHAZOT Alban, COLAS Korlan, GOURD Anthony -  
Promotion 2017  
*Tuteur : P. CLEMENTE*

## **Résumé**

Compte rendu portant sur le projet application effectué lors de la première année de formation à la Sécurité et aux Technologies de l'Informatique (STI) à l'INSA Centre Val de Loire

# Table des matières

<b>I</b>	<b>Analyse du sujet</b>	<b>3</b>
1	Présentation du casse-tête	4
2	Description des objectifs	6
<b>II</b>	<b>Résolution du casse-tête</b>	<b>7</b>
3	Principe de la résolution	8
4	Algorithme de résolution	9
<b>III</b>	<b>Organisation</b>	<b>10</b>
5	Organisation humaine	11
6	Architecture logiciel	12
<b>IV</b>	<b>Développement</b>	<b>13</b>
7	Core	14
8	Graphique	15
9	Portabilité	16
<b>V</b>	<b>Résultat</b>	<b>17</b>
10	Réalisations	18
11	Tests	19

# Introduction

Première partie

**Analyse du sujet**

# Chapitre 1

## Présentation du casse-tête

Le Snake Cube (ou cube serpent en français) est un casse-tête géométrique à trois dimensions appartenant à la famille des casse-tête mécaniques. Cette famille, dont fait parti le Rubik's Cube, recouvre l'ensemble des jeux de réflexion basés sur la manipulation d'un objet tridimensionnel, afin de lui donner un agencement précis.

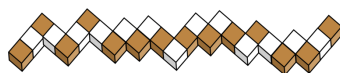


FIGURE 1.1 – Snake Cube déplié

Ce casse-tête se présente généralement comme une succession de 27 petits cubes en bois, reliés les uns aux autres par un fil élastique qui les traverse de part en part. Lorsque tous les cubes sont mis sur le même plan, la forme du puzzle s'apparente à un serpent (Figure 1 - Snake Cube déplié). Pour résoudre le puzzle, il faut manier le serpent afin de le ramener à une forme entièrement cubique telle que présentée ci-dessous (Figure 2 - Snake Cube résolu). Dans la suite de cette partie et pour éviter les ambiguïtés, les petits cubes qui forment le serpent seront appelés **unités** tandis que le mot cube désignera plutôt le volume final visible sur la figure 2.

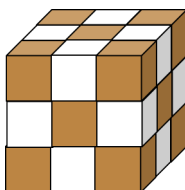


FIGURE 1.2 – Snake Cube résolu

Comme mentionné précédemment, les unités sont reliées entre elles ce qui les rend totalement indissociables. Ainsi, les manipulations pouvant être effectuées sur le puzzle sont réduites à des rotations. La Figure 3 - Exemple de manipulation du Snake illustre une manipulation réalisée sur le serpent présenté figure 1. De plus, on remarque que les rotations conduisant à modifier la forme de la figure s'effectuent toujours au niveau des coins du serpent. Cette particularité sera notamment exploitée lorsque l'on traitera de la résolution du casse-tête.

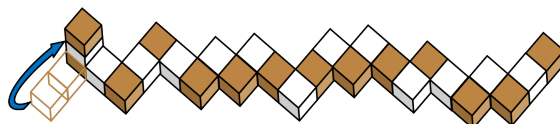


FIGURE 1.3 – Exemple de manipulation du Snake Cube

Notons également que pour une figure finale donnée (ici une cube  $3 \times 3 \times 3$ ), il existe une multitude de formes de serpents correspondantes. Les serpents les plus connus seront listés en annexe de ce document, associés à leurs spécificités (difficulté, nombre de solutions, temps de calcul correspondant). De manière moins classique, on trouve aussi dans le commerce des serpents permettant de réaliser des cubes  $4 \times 4 \times 4$ . À notre connaissance il n'existe pas de casse-tête correspondant à des Snake Cubes de tailles supérieures.

## Chapitre 2

# Description des objectifs

Dans un premier temps, notre objectif consiste en l'implémentation d'une application capable de résoudre des Snake Cubes. L'utilisateur fournit au résolveur deux paramètres essentiels : la définition d'un serpent initial ainsi qu'un volume final. Ensuite l'application automatise la tâche de recherche pour présenter à l'utilisateur la liste de solutions correspondant au problème. Les solutions devront être présentées de manières claires avec, pour chacune d'entre elles, la possibilité pour l'utilisateur de la visionner pas-à-pas ou bien encore de revenir en arrière. Une première contrainte évidente liée à cet objectif concerne le temps nécessaire au calcul. Ainsi, il va de soi que le résolveur devra à la fois s'employer à utiliser des méthodes de calcul pertinentes et être capable de ne pas rechercher des solutions redondantes. Cette redondance traduit en fait le caractère symétrique de plusieurs solutions.

Dans un second temps, nous proposons à l'utilisateur un mode interactif où celui-ci pourra tenter de résoudre virtuellement le casse-tête. En termes de contraintes, ce versant de l'application nécessite très peu de calcul. Néanmoins, il exige une réflexion accrue concernant la représentation graphique. En effet, il faut ici proposer au joueur un environnement en 3 dimensions à la fois clair et maniable ainsi qu'un panel de fonctionnalités rendant l'application intuitive.



Deuxième partie

Résolution du casse-tête

## Chapitre 3

# Principe de la résolution

## Chapitre 4

# Algorithme de résolution

## Troisième partie

# Organisation

## Chapitre 5

# Organisation humaine

## Chapitre 6

# Architecture logiciel

Quatrième partie

Développement

## Chapitre 7

### Core



# Chapitre 8

## Graphique

## Chapitre 9

# Portabilité

Cinquième partie

Résultat

## Chapitre 10

# Réalisations

# Chapitre 11

## Tests

Dans le but de vérifier l'efficacité de certains de nos algorithmes, de connaître les performances ainsi que les limites de notre application, nous avons effectué plusieurs benchmark donc voici les résultats et conclusions.

### **Benchmark n°1**

# Table des figures

1.1	Snake Cube déplié . . . . .	4
1.2	Snake Cube résolu . . . . .	4
1.3	Exemple de manipulation du Snake Cube . . . . .	5

# Listings