

FACULTE DES SCIENCES - Année 2018-2019

DEPARTEMENT INFORMATIQUE

Rapport de projet TER  
Projet Informatique HLIN405

## **Projet Sudoku en Réalité Augmentée**

Encadrant:

---

Etudiants:

Simione Jérémy, Henriksen Leif, Besson Thomas

# Contents

<b>1</b>	<b>Organisation du projet</b>	<b>2</b>
1.1	Organisation du travail . . . . .	2
1.2	Répartition du travail dans le temps . . . . .	2
1.3	Outils de travail collaboratif . . . . .	3
<b>2</b>	<b>Conception du Sudoku</b>	<b>3</b>
2.1	. . . . .	4
<b>3</b>	<b>Implémentation du Sudoku</b>	<b>4</b>

# 1 Organisation du projet

## 1.1 Organisation du travail

Pour le développement de notre application Sudoku , nous avons décidé de travailler chacun de notre côté et de temps en temps ensemble suivant la difficulté des choses à réaliser. Pour que le projet nous apporte a tous des connaissances dans les différents domaines auquel il touche nous avons essayé de répartir les tâches de sorte a ce que chaque membre du groupe ai vu chaque domaine (android,java,openCV).

Afin d'être les plus efficace et d'avancer le plus rapidement possible nous nous sommes réunis quotidiennement. Durant les jours de la semaine, nous nous sommes vus souvent afin de connaître l'avancée de chacun dans le projet, faire le point sur l'avancement du projet, définir de nouveaux objectifs et de les réaliser.

A chaque étape réalisée nous avons postés sur un depot Github créé pour le projet chaque nouvelle partie afin que tout chaque memebre puisse s'informer et voir. A chaque étape importante nous nous sommes réunis avec notre encadrant M. Boudet afin de faire le point sur l'état d'avancement de l'application.

## 1.2 Répartition du travail dans le temps

Nous avons découpé cette période de travail en plusieurs phases.

1. Préparation du projet. Nous avons réalisé le cahier des charges de l'application, choisi les outils de travail et les principales technologies utilisées. Nous avons fait une première version du diagramme de répartition des tâches dans le temps, et une première modélisation de l'architecture de l'application.

2. Développement du projet. Nous avons implanté les fonctionnalités de l'application en raffinant la modélisation au fur et à mesure. Pour chaque module implanté, nous nous sommes efforcés d'écrire des tests afin de s'assurer de leur bon fonctionnement.

3. Finalisation du projet. Cette phase a consisté en la correction de bogues afin d'obtenir une version suffisamment stable pour pouvoir être présentée en vue de la soutenance et du rendu du projet T.E.R.

### 1.3 Outils de travail collaboratif

Nous avons choisi d'utiliser Github qui permet la gestion des versions du projet et facilite la collaboration a distance.

Enfin, pour éditer le code du projet , nous nous sommes servis d'Android Studio.

Il etait en effet plus facile de commencer sur cet éditeur car il ne fonctionne pas de la meme facon que les autres éditeurs et notre code source final passe obligatoirement par cet IDE. Nous avons aussi créer un diagramme de gantt afin de planifier les tâches pour avoir des dates limites pour chaque partie ce qui nous a permis de réaliser le projet dans son ensemble et dans les temps.

Pour rédiger les différents documents, y compris ce rapport, nous avons utilisé  $\text{\LaTeX}$  pour sa capacité à produire des documents de bonne qualité.

## 2 Conception du Sudoku

La partie conception était toute nouvelle pour nous car il s'agissait de créer une application Android. Nous avons donc divisé le projet en plusieurs parties à savoir:

Premièrement une classe Sudoku qui va représenter notre grille de sudoku mais aussi toutes les methodes qui vont nous permettre de récupérer la grille ainsi que l'algorithme de résolution d'une grille (Solveur).

Deuxièmement il s'agissait de faire une partie qui se charger de récupérer les données de la classe Sudoku et qui allait l'afficher dans une application android c'est a dire réaliser l'affichage graphique de notre classe sudoku en passant par une application android.

Il fallait ensuite pour le projet pouvoir prendre une image en photo afin de pouvoir la faire lire ensuite par une intelligence artificielle capable de lire des nombres a partir d'une image.

Cette partie consistait encore a développer une interface android avec des boutons et un acces a la camera du smartphone de l'utilisateur pour pouvoir l'enregistrer dans une galerie de photos afin de retrouver le chemin de l'image et procéder à la reconnaissance.

Ensuite nous avons besoin d'une partie qui nous permettrait de lire l'image ainsi enregistrée a savoir l'intelligence artificielle. Cette partie a été de loin la plus dure et la plus longue car il s'agissait d'appliquer des traitements a l'image en fonction afin de faciliter la lecture par l'IA.

## 2.1

### 3 Implémentation du Sudoku

La première partie consistait à créer une classe java de sudoku pour pouvoir implémenter la résolution; la solution la plus simple était de choisir une structure de données de type tableau.

La deuxième étape consistait à créer des fonctions secondaires qui nous serviraient pour la résolution de la grille entière; nous avons donc créé des fonctions qui testent si une valeur est absente d'un bloc, d'une colonne ou d'une ligne de la grille ; voici un exemple de l'une de ces fonctions:

---

**Algorithme 1** : absentSurColonne(Entier valeur,Grille g,Entier j)

---

**Données** : Entier valeur,Grille grille,Entier j

**Résultat** : Renvoi vrai si la valeur n'est pas dans la colonne, faux sinon

**pour** *i allant de 0 à 9 faire*

**si** *grille[i][j]=k alors*

**retourner** *Faux*

**fin**

**retourner** *Vrai*

**fin**

---

La troisième étape consistait à implémenter le backtrack<sup>1</sup>. Cette fonction doit prendre une grille en entrée la résoudre et nous informer de l'état du résultat en nous renvoyant un booléen. Il fallait donc vérifier dans la descente récursive en énumérant tous les chiffres possibles pour observer si nous arrivions à un résultat correct ou un blocage tout cela en remplissant la grille dans la descente et si nous arrivions à un blocage nous réinitialisions la case correspondante à zéro.

Voici l'algorithme en question :

---

<sup>1</sup> Aussi nommé le retour sur trace en français

---

**Algorithme 2 : estValide(Grille grille,Entier position)**

---

**Données :** Grille grille,Entier position **fin**  
**Résultat :** Renvoi vrai si la grille a été résolue, renvoi faux sinon  
**si**  $position=9*9$  **alors**  
    **retourner** *Vrai*  
**fin**  
 $i \leftarrow position \div 9$   $j \leftarrow position \% 9$   
**si**  $grille[i][j] \neq 0$  **alors**  
    **retourner**  
    *estValide(grille,position+1);*  
**pour**  $k$  allant de 1 à 9 **faire**  
    **si** *absentSurLigne(k,grille,i)* et *absentSurColonne(k,grille,j)* et  
        *absentSurBloc(k,grille,i,j)* **alors**  
             $grille[i][j] \leftarrow k$ ;  
            **si** *estValide(grille,position+1)* **alors**  
                **retourner** *Vrai*  
            **fin**  
    **fin**  
     $grille[i][j] \leftarrow 0$   
    **retourner** *Faux*  
**fin**

---

Cette classe étant terminée nous sommes ensuite passés à la partie interface graphique de la grille sur l'application.  
Cette interface graphique