FACULTE DES SCIENCES - Année 2018-2019

DEPARTEMENT INFORMATIQUE

Rapport de projet TER
Projet Informatique HLIN405

Projet Sudoku en Réalité Augmentée

Encadrant: Boudet Vincent

Etudiants:
Simione Jérémy, Henriksen Leif

Contents

1 Introduction

1.1 Objectifs du projet et cahier des charges

L'objectif est de permettre à un utilisateur de prendre une photo d'une grille de Sudoku afin qu'il puisse obtenir la solution de la grille, le second objectif est qu'il puisse jouer sur l'application.

Il existe déjà des applications Android capables de réaliser cet objectif, mais la plupart ont du mal a obtenit un résultat correct.

Voici le cahier des charges que nous avons crée pour notre application :

Création d'un algorithme de résolution
Création d'une classe Sudoku qui va représenter notre grille de sudoku mais aussi toutes les méthodes qui vont nous permettre de récuperer la grille ainsi que l'algorithme de résolution d'une grille.

Création de l'interface graphique de l'application
L'interface graphique nous permettra de utiliser l'application en appuyant sur des boutons. Nous allons créer une classe Grille qui prend une grille en entrée qui vient de l'algorithme de résolution ou de l'analyse d'image et qui va l'afficher sur l'application. Cette classe Grille nous permettra d'utiliser notre grille.

Mise en place d'un accès a la caméra
Création d'une classe dédiée à la caméra pour que l'utilisateur puisse prendre des photos d'une grille. Cette partie nous servira a envoyer l'image a notre derniere partie.

Mise en place d'un système d'analyse d'images
Création de différentes classes qui vont nous servir à traiter l'image en utilisant la bibliothèque OpenCV et ainsi pouvoir procéder à une reconnaissance des chiffres et l'afficher sur la grille de l'application dans le cas ou l'utilisateur voudrait résoudre une grille ou jouer.

Création du jeu
Création d'une partie jouable en utilisant l'interface graphique de l'application ainsi que la classe Sudoku.

Rassemblement des différentes parties
S'assurer de la compatiblite de tous les classes pour avoir le comportement dessire, et faire les modifications necessaires à la interface graphique pour pouvoir accerder a tous les fonctionalites de l'application.

2 Organisation du projet

2.1 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 repartir 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.

2.2 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.

2.3 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 LaTeXpour sa capacité à produire des documents de bonne qualité.

3 Conception du Sudoku

Deuxiemement il s'agissait de faire une partie qui se charger de récuperer 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 cappable 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 avions 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.

3.1

4 Implémentation du Sudoku

4.1 Création de l'algorithme de résolution

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

La deuxième étape consitait a créer des fonctions secondaires qui nous servirait pour la résolution de la grille entière; nous avons donc créer 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[1].Cette fonction doit prendre une grille en entrée la résoudre et nous informer de l'état du réultat en nous renvoyant un booléen. Il fallait donc vérifier dans la descente récursive en énumérerant 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 reinitialisions la case correspondante à zéro.
Voici l'algorithme en question :

Algorithme 2 : estValide(Grille grille,Entier position)

```
Données : Grille grille,Entier position
Résultat : Renvoi vrai si la grille a été résolue, renvoi faux sinon
si position=9*9 alors
    retourner Vrai
fin
i ⟵ position ÷ 9 j ⟵ position%9
si grille[i][j]!=0 alors
    retourner estValide(grille,postion+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] ⟵ k;
            si estValide(grille,position+1) alors
                retourner Vrai
            fin
        fin
        grille[i][j] ⟵ 0
        retourner Faux
    fin
fin
```

[1]Aussi nommé le retour sur trace en français

4.2 Mise en place d'un système de reconnaissance des chiffres

Pour réaliser le système de reconnaisance des chiffres, nous avons du utiliser la bibliothèque graphique OpenCV. La plupart des problèmes que nous pouvons rencontrer avec cette bibliothèque ne sont très bien documentés car c'est une bibliothèque technique et utilisé surtout par des gens qualifiés.

Nous avons opté pour une implémentation permettant d'afficher sur une grille la reconnaissance d'une image provenant d'un traitement.Le but est que du moment ou l'image a été traitée elle est directement envoyé a l'IA pour qu'elle soit ensuite directement affichée sur l'application.

Lorsque l'utilisateur va sélectionner l'image elle sera directement envoyé au programme de traitement de l'image qui lui-même envera le résultat a l'IA qui essaiera de reconnaitre les chiffres puis d'envoyer un résultat a la classe Sudoku qui l'affichera par la suite.

Pour cette partie nous avons décider de nous faciliter la tâche en utilisant l'IDE eclipse car la gestion des chemins d'accès des images[2] était plus aiséé et ainsi pouvoir observer les traitements résultants afin d'obtenir une image lisible par l'ordinateur.

Pour implémenter ce traitement il y avait un cahier des charges à suivre :
Premièrement il fallait détecter la grille en appliquant une série de traitements à l'image a savoir:

- Convertir notre image de base avec des teintes grises
- Appliquer un flou gaussien afin de réduire le bruit et les détails de l'image
- Appliquer un seuillage d'image c'est a dire convertir notre résultat précedant en noir et blanc
- Trouver les bordures de la grille
- Détecter les lignes de la grille

Voici ci-dessous des algorithmes de traitement de l'image utilisés par OpenCV:
Il fallait créer ensuite un système d'intelligence artificielle qui allait essayer de comparer chaque chiffre de la grille avec des ressources et renvoyer une grille en résultat.
La derniere partie consistait a lier les trois classes c'est a dire notre classe Sudoku, celle qui manipulait l'image et l'intelligence artificielle et adapter notre code afin qu'il soit effectif sur un terminal android.
Voici ci-dessous des algorithmes de traitement de l'image utilisés par OpenCV:

[2]En effet lors de la création d'une application Android les chemin d'accès aux images doivent provenir exclusivement du téléphone.

5 Bilan et difficultés rencontrées

5.1 Avancement du projet

5.2 Difficultés rencontrées

Pour la partie "Mise en place d'un systeme de reconnaissance de chiffres" nous avons été confrontés pour l la première fois a une bibliotheques de traitement d'image et la gestion d'une intelligence artificielle.

Nous avons donc du nous former a cette bibliothèque et après de nombreux essais nous nous ne savions pas si notre système d'IA allait réussir à détecter notre image traitée c'est pourquoi après concertation avec notre encadrant nous avons décidé d'utiliser une classe déja existante nommée ImageManipulator qui traitait l'image comme nous avions besoin afin d'être sûr que le résultat serait correct et que l'image sera ainsi facilement lisible par l'IA.Nous sommes donc parvenus a surmonter cette difficulté en utilisant des classes deja existantes a savoir une classe qui nous a permis de manipuler l'image et une classe qui nous a servi d'intelligence artificielle.

Ensuite malgré la grande aide que nous ont apportés ces deux classes l'adaptation du code d'eclipse à Android Studio a été très difficile car il fallait déjà réussir a installer une version d'openCV sur l'IDE Android ce qui n'a pas été chose aisée, mais aussi une fois ceci fait, trouver une version qui compilait à l'appel de la bibliothèque et vérifier qu'openCV était bien fonctionnel.
Ensuite plus rien ne fonctionnait de la même façon il a donc fallu créer des classes qui nous ont permis de passer outre cette difficulté et pouvoir faire l'adaptation à Android Studio.

Cette partie a été de loin la plus longue et la plus dure.