

Master 2 Informatique : Image et Multimédia

Débruitage de signaux Doppler

...

Recette

Équipe :

Benjamin AZZINI

Lucien MAHOT

Aline LEPAILLEUR

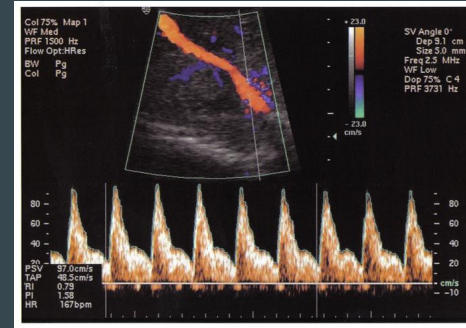
Kévin LEPAN

Client et superviseur :

Dr. Denis KOUAME

Introduction

- ❖ Doppler foetal : enregistrer le son du cœur du bébé



- ❖ Société qui propose une utilisation chez les particuliers



COCOON LIFE
EN CONNEXIONS DOUCES AVEC BÉBÉ

Plan de présentation

- ❖ Présentation du travail effectué
 - Couverture des spécifications
 - Conception détaillée et description du code
 - Tests unitaires et finaux
 - Outils utilisés
- ❖ Retour sur le déroulement du projet
 - Problèmes rencontrés et solutions
 - Améliorations
 - Déroulement du projet
 - Évolution du planning
- ❖ Documentation du projet
- ❖ Démonstration

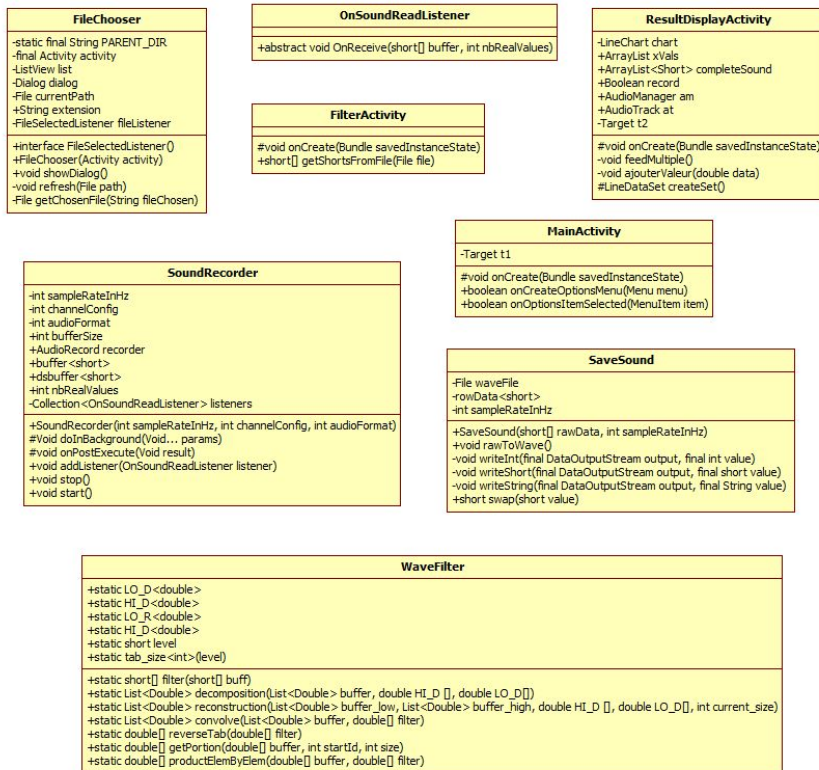
Couverture des spécifications

Description des fonctionnalités de l'application Android :

- ❖ Lecture du son
- ❖ Enregistrement du son
- ❖ Filtrage du son -> pas en temps réel
 - Après comparaison de différentes techniques de débruitage
- ❖ Sauvegarde dans un fichier sur le smartphone
- ❖ Calcul du rythme cardiaque du fœtus -> pas opérationnel
- ❖ Affichage d'informations

Conception détaillée et description du code

Diagramme de classe



Description des classes

- ❖ MainActivity
 - Lancement des processus des différentes fonctionnalités de l'application
- ❖ OnSoundReadListener
 - Classe lançant le traitement à chaque nouveau buffer du son récupéré
- ❖ SoundRecorder
 - Concaténation des buffers pour former le son que récupère l'utilisateur
 - start()/stop() : lancement et arrêt de l'enregistrement du son
 - doInBackground(...) : récupération en parallèle au thread principal UI
 - onPostExecute(...) : ferme correctement la tâche après avoir terminé l'enregistrement
- ❖ ResultDisplayActivity
 - Gère la réception du son du microphone et le place dans un buffer
 - S'occupe de tous les affichages de l'application dont le spectrogramme
 - Gère l'action des boutons d'enregistrement, de filtrage et d'aide

Description des classes

❖ SaveSound

- Sauvegarde du son dans un fichier au format WAV
- WriteInt(...)/WriteShort(...) : évite de devoir fournir les données en bytes

❖ FileChooser

- Permet à l'utilisateur de choisir un enregistrement

❖ FilterActivity

- Grâce à FileChooser, l'enregistrement choisi est découpé en buffer et envoyé au filtrage

❖ WaveletFilter

- Permet le filtrage par ondelettes et seuillage d'un buffer de données au format PCM en paramétrant l'ordre de décomposition -> améliorations apporter au niveau de la rapidité d'exécution.

Description du code

❖ Entrées / Sorties

- Lecture du son du microphone grâce à la classe `AudioRecord`
 - `audioRecorder.startRecording()` après instantiation
 - `AudioRecord(MediaRecorder.AudioSource.DEFAULT, sampleRateInHz, channelConfig, audioFormat, bufferSize) ;`
 - `while(recorder.getRecordingState() == AudioRecord.RECORDSTATE_RECORDING)`

```
nbRealValues = recorder.read(buffer, 0, bufferSize);
```


Description du code

❖ Entrées / Sorties

- Playback du son grâce à la classe `AudioTrack`
 - `at = new AudioTrack(AudioManager.STREAM_MUSIC, sampleRateInHz, AudioFormat.CHANNEL_OUT_MONO, recorder.getAudioFormat(), bufferSize, AudioTrack.MODE_STREAM);`
 - `at.play();`
 - `at.write(buffer, 0, nbRealValues);` (en boucle)
- Création d'un header pour enregistrement en WAV
 - Conversion data Big-endian -> Little-endian

Description du code

❖ Filtrage

- Choix des ondelettes discrètes
 - Résultats Matlab cohérents (comme les autres filtres)
 - Temps d'exécution minimal pour les filtres testés
- Implémentation en Matlab sans utiliser une seule fonction Matlab
 - Faciliter passage Matlab <-> Java
- Jusqu'à 4 niveaux de décompositions
 - Seuillage doux sur les hautes fréquences

Tests unitaires

- ❖ Filtrage/débruitage
 - Nos résultats + Matlab
 - Soustraction signal bruité avec débruité
- ❖ Récupération du son
 - Fonction de lecture audio des données
- ❖ Enregistrement
 - Lecture de l'enregistrement avec le lecteur basique du téléphone

Tests unitaires

❖ Transformée de Fourier

- Comparaison de nos résultats de l'application avec Matlab

❖ Rythme cardiaque

- Tests impossibles, il nous manque le véritable matériel médical pour cette tâche
- En revanche, la vérification de la cohérence des données est possible (~110..160 bpm)

❖ Affichage

- Essentiellement visuels
- Contrainte de fluidité, cohérence

Tests finaux

- ❖ Comparaison de l'application CocoonLife et la nôtre



Outils utilisés

❖ Développement



❖ Documentation



Problèmes rencontrés et solutions

❖ Problèmes matériels :

- compatibilité avec l'application CocoonLife
 - 2 téléphones sur 5 compatibles
- dispositif avec une prise TRRS
 - fabrication d'un adaptateur puis récupération de la sonde

❖ Problèmes logiciels :

- installation avec les mêmes configurations et version du SDK
- problèmes de gestion des version sur git et de merge

❖ Problèmes organisationnels :

- rythme et charge de travail du master élevés
- problèmes du filtrage en temps réel qui aurait pu être détecté plus tôt
- disponibilité des salles de travail pour un temps consécutif important

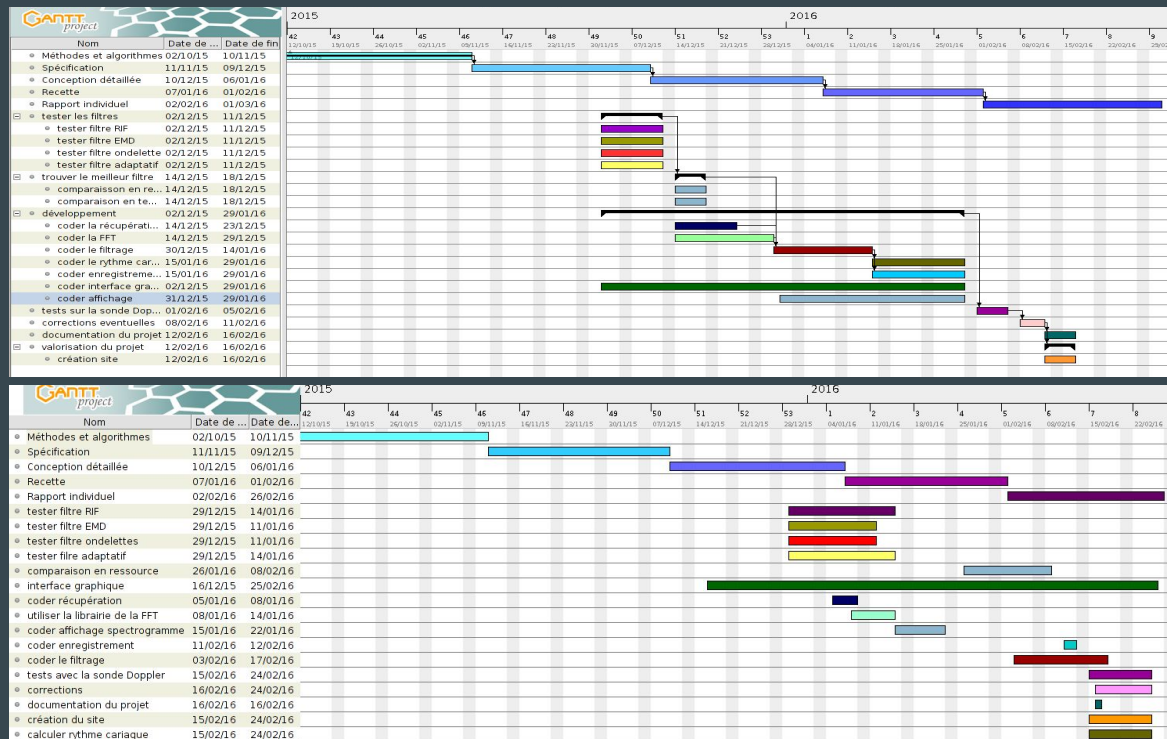
Améliorations

- ❖ Calcul du rythme cardiaque foetal
 - Pour calculer le rythme cardiaque foetal il faudrait séparer les fréquences positives et négatives du signal puis utiliser une fonction d'autocorrelation qui permet de trouver la période et donc la fréquence.
- ❖ Optimisation de la fonction de filtrage -> temps réel
 - Parallélisation de certaines opérations (2 convolutions en parallèle par exemple)
 - Sous-échantillonnage puis sur-échantillonnage (à tester car perte possible)
- ❖ Laisser le choix du filtre à appliquer à l'utilisateur
 - Le filtre utilisé peut ne pas être adéquat suivant les conditions d'enregistrements
- ❖ Partage vers les réseaux sociaux

Déroulement du projet

- ❖ Première phase latente, en parallèle des cours :
 - Prise de connaissance du sujet
 - Installation et paramétrisation de l'environnement de développement
 - Recherche et tests du filtre à utiliser
 - Choix du filtre
 - Code jetable
- ❖ Deuxième phase active
 - Codage de la structure de l'application (entrées, sorties, etc)
 - Codage du filtrage
 - Réorientation du projet vers une solution non temps réel
 - Tests

Évolution du planning



Documentation du projet

❖ Manuel d'utilisation

❖ Site web



The screenshot shows a website with a dark grey header. The header contains a navigation menu with the following items: 'Home' (highlighted in red), 'Ressources', 'Contact', 'TCI', and a logo for 'COCOON LIFE'. Below the header, the main content area has a white background. The title 'Chef d'oeuvre du Master 2 Images et Multimédia - Université Paul Sabatier' is centered in a bold, black font. Below the title, the subtitle 'Débruitage d'un signal Doppler' is also centered in a bold, black font. The text 'Introduction à l'échocardiographie et à l'effet Doppler:' is followed by a paragraph explaining the technique. The 'Objectif du projet :' section describes the project's goal and mentions 'LATL TECHNOLOGY' and 'CocoonLife'. The 'Quelques screenshot :' section is partially visible at the bottom. The bottom of the image shows a Windows taskbar with several open browser windows and system icons.

Home Ressources Contact TCI COCOON LIFE

Chef d'oeuvre du Master 2 Images et Multimédia - Université Paul Sabatier

Débruitage d'un signal Doppler

Introduction à l'échocardiographie et à l'effet Doppler:

L'échocardiographie ou écho-doppler est une technique d'imagerie médicale permettant d'étudier les battement du coeur. Cette technique utilise une sonde Doppler qui envoie et reçoit des ultrasons.

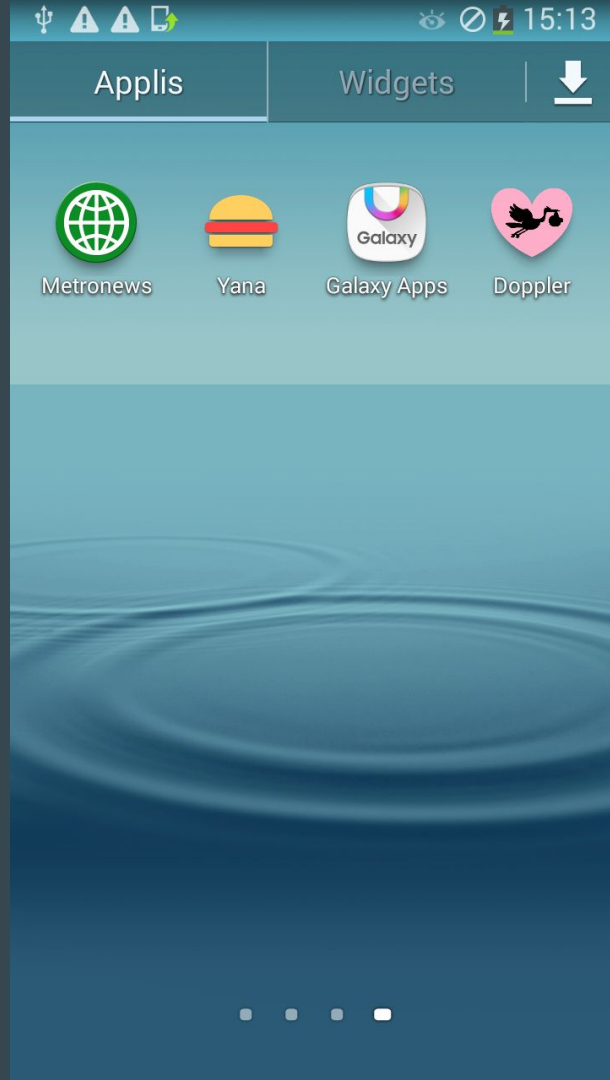
Objectif du projet :

Dans le cadre de notre projet de "Chef d'oeuvre" du Master 2 Image et multimédia, la société LATL TECHNOLOGY nous a mis à disposition, une sonde Doppler provenant de leur système CocoonLife. Ce système permet aux femmes enceintes de pouvoir écouter les battement du coeur de leur foetus depuis leur smartphone. Afin d'améliorer l'enregistrement, la société a fait appel à l'université Paul Sabatier afin de trouver une manière de débruité le signal correspondant au battement du feotus.

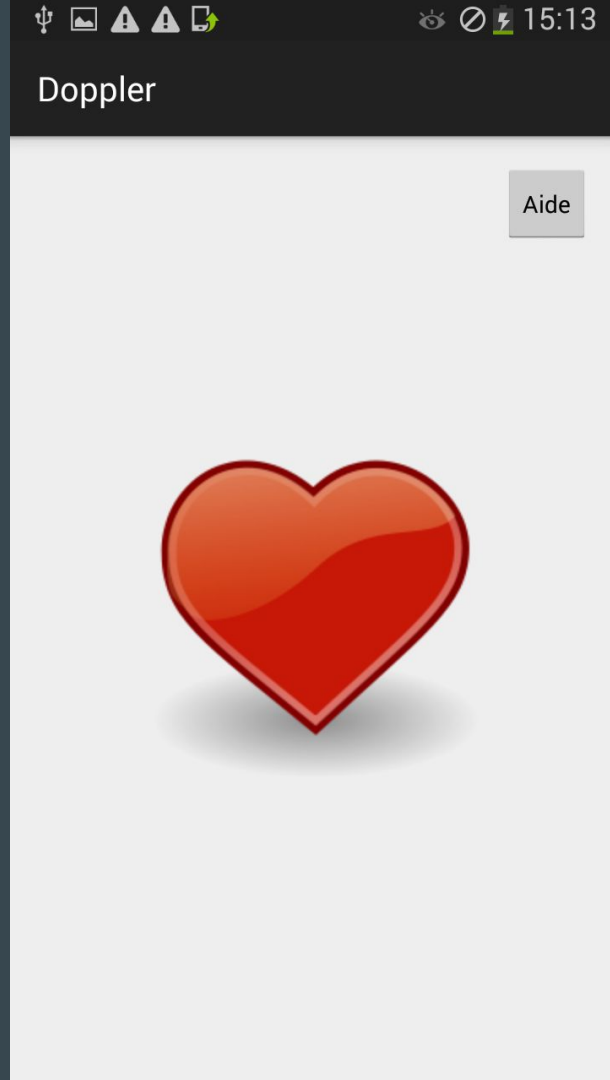
Ainsi, l'objectif de notre chef d'oeuvre est de re-créeer une application Android qui puisse enregistrer un son grâce à la sonde Doppler, mais aussi de trouver une manière de débruite ce signal.

Quelques screenshot :

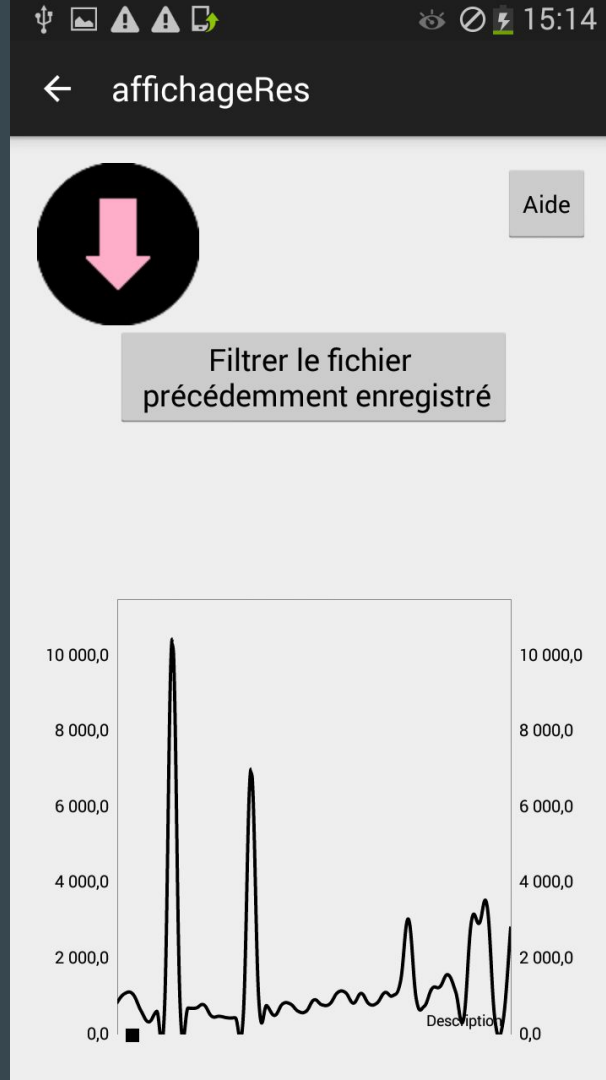
Démo



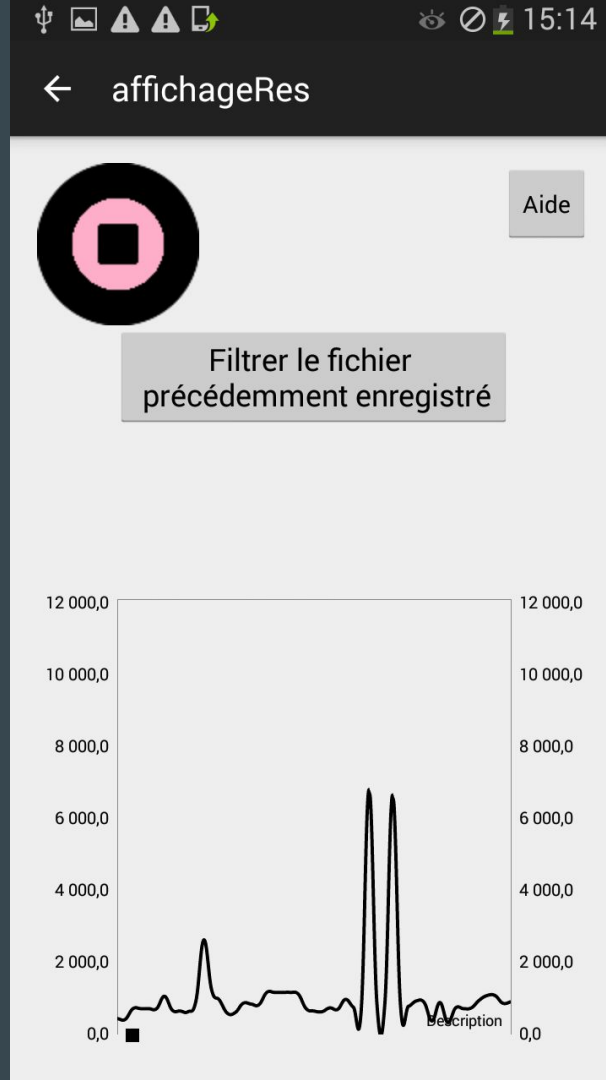
Démo



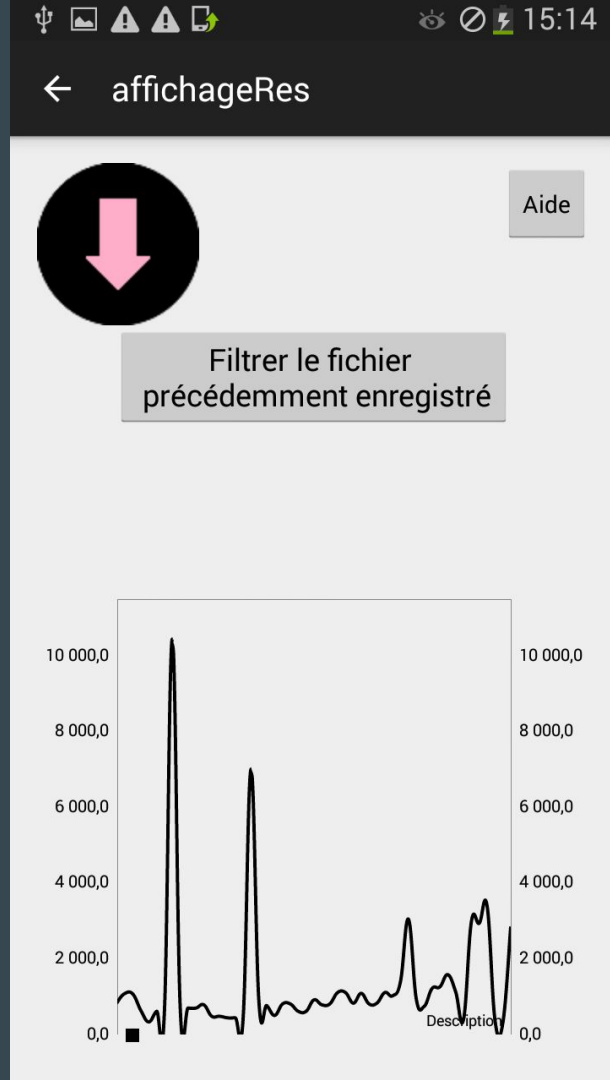
Démo



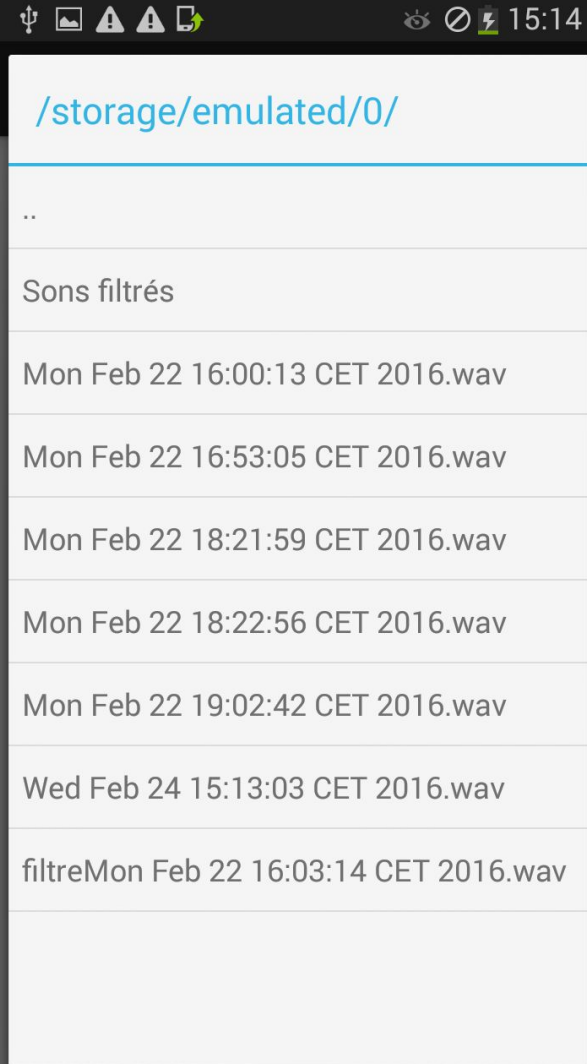
Démo



Démo



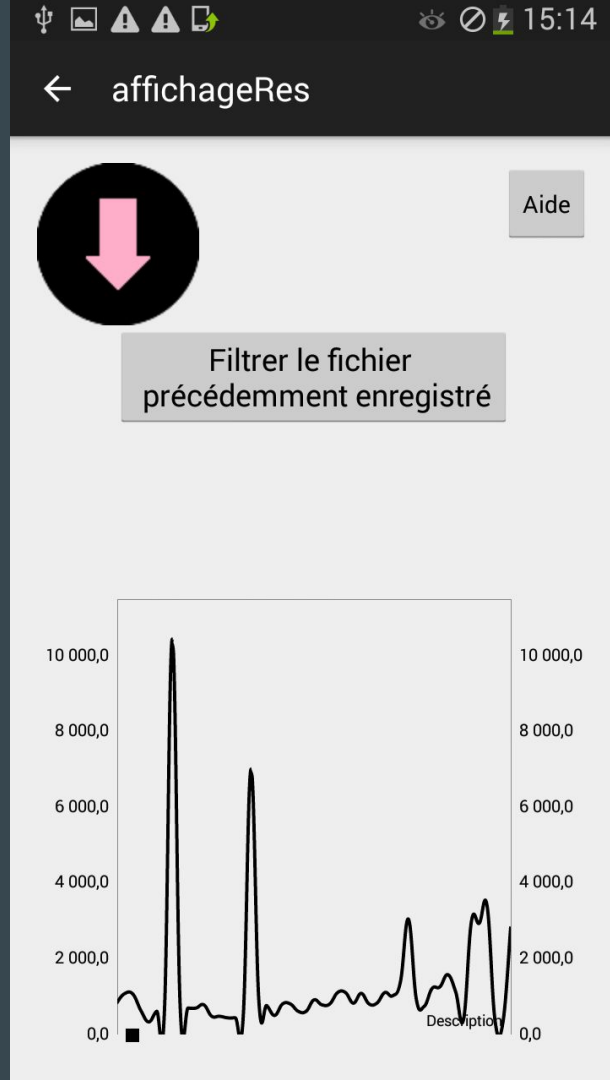
Démo



Démo



Démo



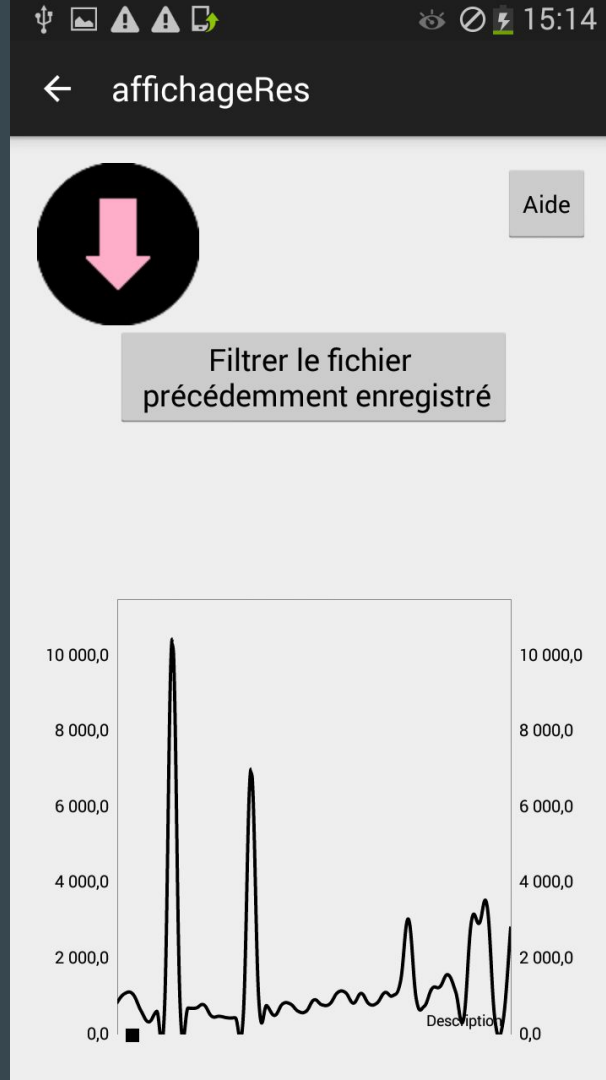
Démo



Démo



Démo



Conclusion

- ❖ Objectif principal atteint :
 - Application fonctionnelle de débruitage d'un signal
- ❖ Améliorations avec de l'optimisation :
 - Pour une meilleure interaction avec les utilisateurs
- ❖ Objectif secondaire : Regret pour le rythme cardiaque
- ❖ Acquis et connaissances développés
- ❖ Expérience personnelle en traitement du signal

Questions ?