

# Projet de Programmation

## Une surface télécommande mains libres partagée

Guillaume Béchade, Raphaël Jorel, Craig Josse, Antoine Laulan  
Client : M. Serge Chaumette

Université de Bordeaux

Année 2014/2015

# Plan

## Introduction

- Présentation du projet

- État de l'art

- Environnement de travail

## Analyses des besoins

- Besoins fonctionnels

- Besoins non-fonctionnels

## Architecture et fonctionnalités

- Diagrammes statiques

- Fonctionnalités implémentées

- Quelques détails techniques

## Limites, tests et améliorations possibles

- Limites et tests unitaires

- Performances et robustesse

- Améliorations et conclusion

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

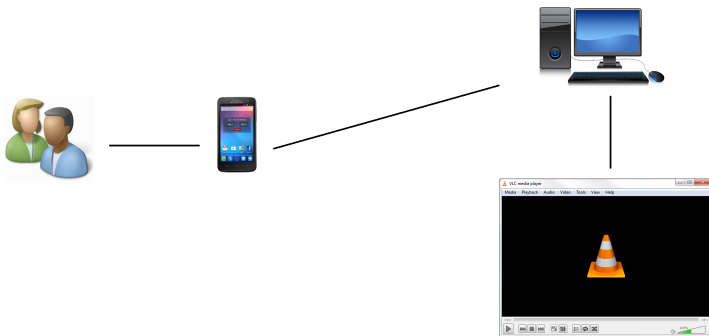
## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

# Contexte



# Domaine



- ▶ Logiciel à produire : application mobile + application PC,
- ▶ Système d'exploitation : Android,
- ▶ Type de connexion : connexion Wi-Fi,
- ▶ Utilisation des capteurs d'un smartphone,
- ▶ Échange de données.

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Télécommande Android

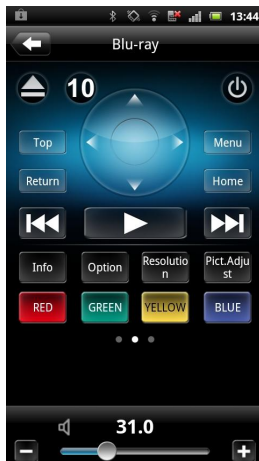


Figure: Denon Remote app

# Analyse de données

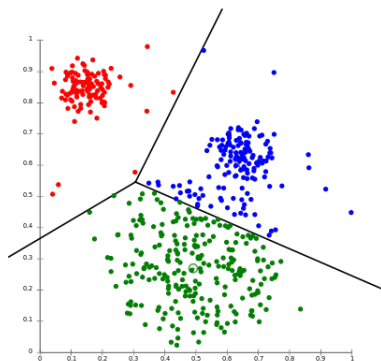


Figure: Kmeans



## Remarques sur l'existant

- ▶ De nombreuses télécommandes,
- ▶ Plusieurs algorithmes d'analyse de données utilisant les capteurs,
- ▶ Apparemment pas d'application mélangeant les deux.

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Cadre

université  
de **BORDEAUX**



## Technologies utilisées

- ▶ Langages de programmation : Java, XML,
- ▶ Integrated Development Environment : Eclipse
- ▶ Outils : Android SDK, JUnit, Emma,
- ▶ Gestionnaire de version : SVN

# Plan

## Introduction

- Présentation du projet

- État de l'art

- Environnement de travail

## Analyses des besoins

- Besoins fonctionnels

- Besoins non-fonctionnels

## Architecture et fonctionnalités

- Diagrammes statiques

- Fonctionnalités implémentées

- Quelques détails techniques

## Limites, tests et améliorations possibles

- Limites et tests unitaires

- Performances et robustesse

- Améliorations et conclusion

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

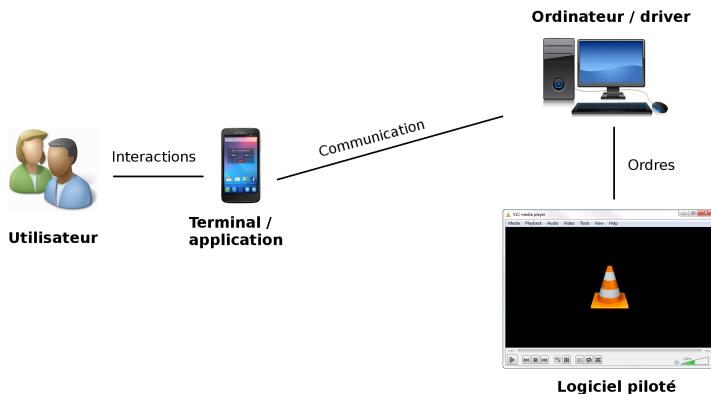
Limites et tests unitaires

Performances et robustesse

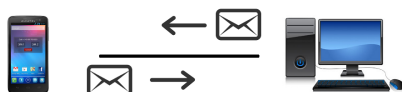
Améliorations et conclusion

- └ Analyses des besoins
- └ Besoins fonctionnels

## Contexte



## Communication

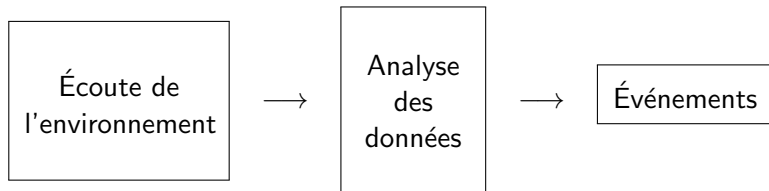


Type de messages :

- ▶ initialisation de connexion (application et driver),
- ▶ spécification des séquences d'événements attendues (driver),
- ▶ informations quant aux séquences effectuées par l'utilisateur (application).



## Application : écoute et analyse



---

Écoute de l'environnement : les capteurs font des mesures de quantités physiques de l'environnement.

---

Analyse des données : utilisation d'algorithmes pour détecter des variations sur les mesures et en déduire des actions effectuées par l'utilisateur.

---

Événements : résultats des analyses, l'utilisateur a-t-il provoqué une modification de l'environnement ?

## Application : Interface Homme Machine



- ▶ Connexion au driver,
- ▶ Aide textuelle pour guider l'utilisateur,
- ▶ Mode capture et analyse pour contrôler le programme tiers.

## Driver



- ▶ Démarrer une session serveur (modèle client-serveur),
- ▶ Récupérer l'information venant de l'application,
- ▶ Contrôler un programme tiers et lui donner des ordres.

- └ Analyses des besoins
  - └ Besoins non-fonctionnels

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

**Besoins non-fonctionnels**

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Adaptabilité

- ▶ Support des dernières versions d'Android, 4.x et 5.x au minimum,
- ▶ Fonctionnement sur différents matériels, dont les sensibilités des capteurs sont différentes.

## Vélocité générale

- ▶ Structures de données appropriées,
- ▶ Séquençage d'événements,
- ▶ Envoi des informations sur le canal de communication.

# Plan

## Introduction

- Présentation du projet

- État de l'art

- Environnement de travail

## Analyses des besoins

- Besoins fonctionnels

- Besoins non-fonctionnels

## Architecture et fonctionnalités

- Diagrammes statiques

- Fonctionnalités implémentées

- Quelques détails techniques

## Limites, tests et améliorations possibles

- Limites et tests unitaires

- Performances et robustesse

- Améliorations et conclusion

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

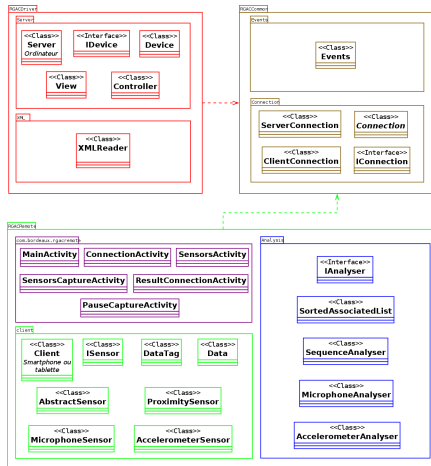
Limites et tests unitaires

Performances et robustesse

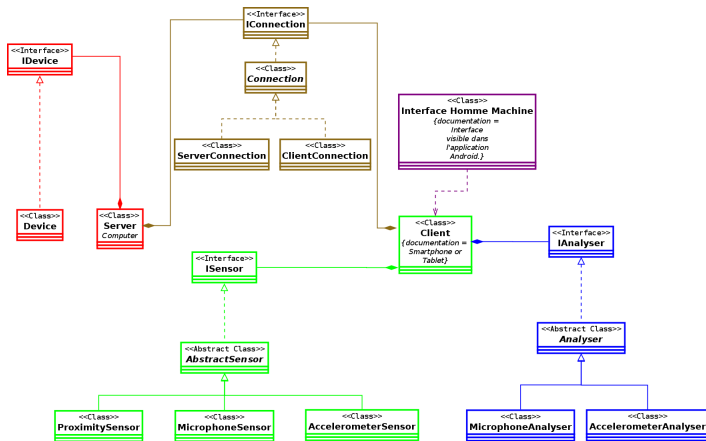
Améliorations et conclusion



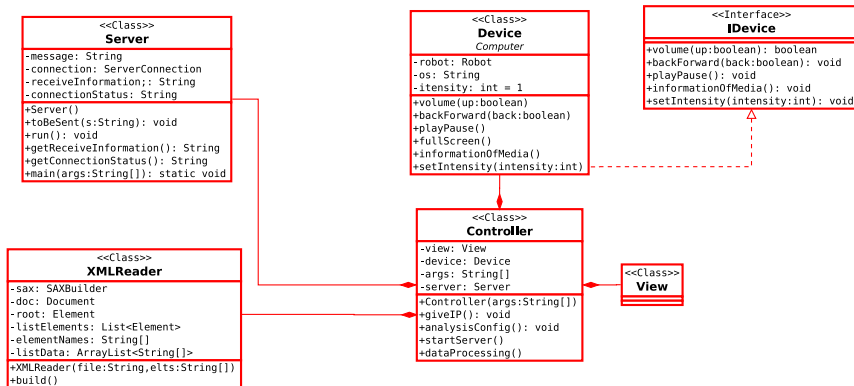
## Diagramme de paquetages



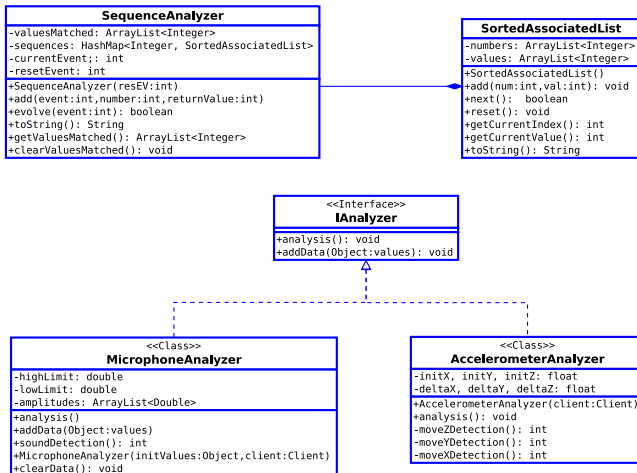
# Diagramme de classe général



# Paquetage Driver



# Paquetage Analysis



- └ Architecture et fonctionnalités
- └ Fonctionnalités implémentées

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

**Fonctionnalités implémentées**

Quelques détails techniques

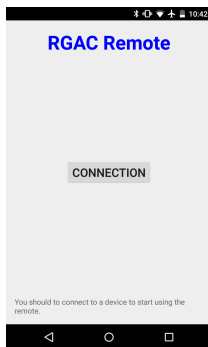
## Limites, tests et améliorations possibles

Limites et tests unitaires

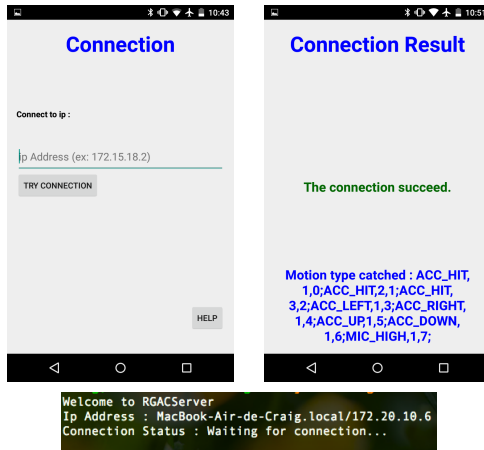
Performances et robustesse

Améliorations et conclusion

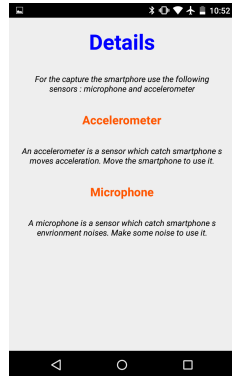
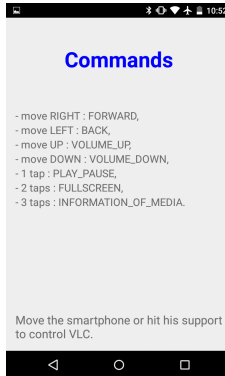
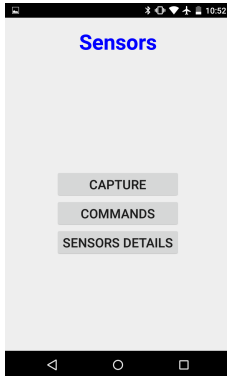
## Fonctionnalités implémentées



## Connexion

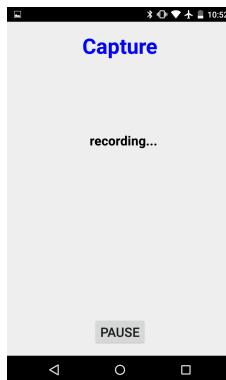


# Capteurs





# Capture



```
Welcome to RGACServer
Ip Address : MacBook-Air-de-Craig.local/172.20.10.6
Connection Status : Waiting for connection...
Receive Information : 0
PLAY_PAUSE
```

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Algorithme d'analyse

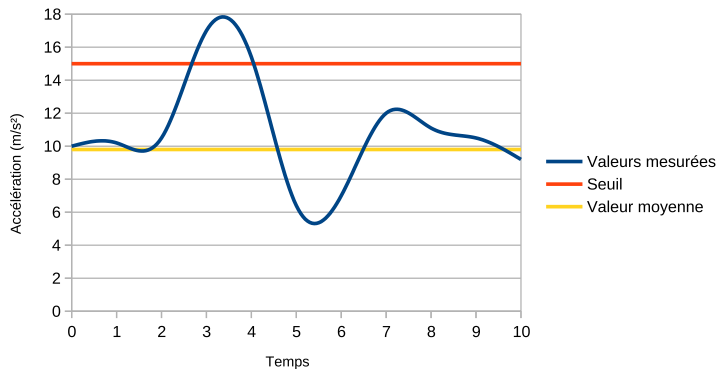


Figure: Accélération sur l'axe z

## Algorithme de séquençage

*aaaabRbaaaRabb*

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$

# Algorithme de séquençage

*aaaabRbaaaRa*  $\underbrace{bb}_2$

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$

# Algorithme de séquençage

*aaaabRbaaaR*  $\underbrace{a}_0$

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$

# Algorithme de séquençage

*aaaabRbaaa*  $\underbrace{R}_{\text{reset}}$

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$

# Algorithme de séquençage

$aaaabRb \underbrace{aaa}_1$

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$



# Algorithme de séquençage

$$aaaabR \underbrace{b}_{\emptyset}$$

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$

## Algorithme de séquençage : exemple de spécification

$$\left\{ \begin{array}{ll} a^1 & \longrightarrow 0 \\ a^3 & \longrightarrow 1 \\ b^2 & \longrightarrow 2 \\ b^5 & \longrightarrow 3 \end{array} \right.$$

```
<?xml version="1.0" encoding="UTF-8" ?>
<root>
  <entry>
    <event>a</event>
    <number>1</number>
    <id>0</id>
  </entry>

  <entry>
    <event>a</event>
    <number>3</number>
    <id>1</id>
  </entry>

  <entry>
    <event>b</event>
    <number>2</number>
    <id>2</id>
  </entry>

  <entry>
    <event>b</event>
    <number>5</number>
    <id>3</id>
  </entry>
</root>
```

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Limites

- ▶ Comparaisons de valeurs par rapport à des seuils,
- ▶ Les valeurs sont fixées en dur dans le code, elles sont adaptées pour le Nexus 4.



Figure: Google LG Nexus 4

# Tests unitaires

Outils utilisés : JUnit4 et EclEmma

TestAccelerometerAnalyzer (6 avr. 2015 17:28:39)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
Analyzer	26,3 %	552	1 547	2 099
src	26,3 %	552	1 547	2 099
tests	22,4 %	260	901	1 161
analysis	35,3 %	292	535	827
SequenceAnalyzer.java	0,0 %	0	214	214
SortedAsAssociatedList.java	0,0 %	0	210	210
MicrophoneAnalyzer.java	0,0 %	0	109	109
AccelerometerAnalyzer.java	99,3 %	292	2	294
AccelerometerAnalyzer	100,0 %	231	0	231
AccelerometerAnalyzer(IClient)	100,0 %	47	0	47
addData(Object)	100,0 %	6	0	6
analysis()	100,0 %	9	0	9
moveXDetection()	100,0 %	63	0	63
moveYDetection()	100,0 %	63	0	63
moveZDetection()	100,0 %	43	0	43
events	0,0 %	0	111	111

- L'analyse des données,
- La partie connexion,
- L'application PC.

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Consommation batterie

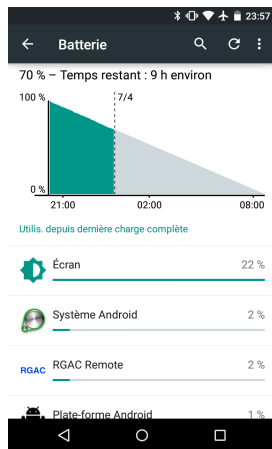


Figure: Après 3H30 d'utilisation



## Consommation CPU

```
Mem: 1718984K used, 160828K free, 0K shrd, 103288K buff, 976952K cached
CPU: 13.5% usr 25.2% sys  0.0% nic 61.1% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.74 1.37 1.73 1/1211 17068
```

PID	PPID	USER	STAT	VSZ	%VSZ	CPU	%CPU	COMMAND
1780	1460	system	S <	1622m	88.1	0	12.3	system_server
16821	1460	app_144	S	1485m	80.6	0	10.5	{eaux.rgacremote} com.bordeaux.rgacremote
1461	1	nobody	S	9560	0.5	0	6.1	/system/bin/sensors.qcom
16760	16720	app_97	R <	940	0.0	0	3.5	top -d 1

Figure: Application smartphone en action

```
Mem: 1722392K used, 157420K free, 0K shrd, 103304K buff, 976960K cached
CPU: 30.0% usr 69.9% sys  0.0% nic  0.0% idle  0.0% io  0.0% irq  0.0% sirq
Load average: 0.85 1.10 1.57 4/1177 17429
```

PID	PPID	USER	STAT	VSZ	%VSZ	CPU	%CPU	COMMAND
17284	1460	app_144	S	1485m	80.6	1	97.3	{eaux.rgacremote} com.bordeaux.rgacremote
17253	16720	app_97	R <	940	0.0	0	0.9	top -d 1

Figure: Application smartphone après l'avoir mis en pause

- └ Limites, tests et améliorations possibles
- └ Performances et robustesse

## Robustesse

- ▶ Perturbations lors de la capture,
- ▶ Taux de réussite de reconnaissance de mouvements : 78.4%.
  - ▶ succession de LEFT / RIGHT : 86%,
  - ▶ succession de UP / DOWN : 72%,
  - ▶ succession de “un tap” : 73%,
  - ▶ succession de “deux taps” : 81%,
  - ▶ succession de “trois taps” : 79%,
  - ▶ succession de MIC LOW : 88%,
  - ▶ succession de MIC HIGH : 70%.

# Plan

## Introduction

Présentation du projet

État de l'art

Environnement de travail

## Analyses des besoins

Besoins fonctionnels

Besoins non-fonctionnels

## Architecture et fonctionnalités

Diagrammes statiques

Fonctionnalités implémentées

Quelques détails techniques

## Limites, tests et améliorations possibles

Limites et tests unitaires

Performances et robustesse

Améliorations et conclusion

## Améliorations possibles

- ▶ Des algorithmes plus performants,
- ▶ Un meilleur séquençage,
- ▶ Fichier XML pour ne pas avoir de valeurs en dur,
- ▶ Consommation CPU de notre application en veille.

## Conclusion



Figure: Notre application en action

- └ Limites, tests et améliorations possibles
- └ Améliorations et conclusion

# Merci pour votre attention

