SYM : Labo 3 - Environnement I (Codes-barres, iBeacons et NFC)

Auteurs : Julien Béguin, Robin Cuénoud & Gaëtan Daubresse
Date : 07.01.2021
Classe : B

1. Introduction

Ce laboratoire est la seconde partie du travail concernant l'utilisation de données environnementales, celui-ci est consacré aux capteurs disponibles sur les smartphones (accéléromètre et magnétomètre principalement) ainsi qu'à la communication Bluetooth Low Energy.

2. Capteurs

> Une fois la manipulation effectuée, vous constaterez que les animations de la flèche ne sont pas fluides, il va y avoir un tremblement plus ou moins important même si le téléphone ne bouge pas. Veuillez expliquer quelle est la cause la plus probable de ce tremblement et donner une manière (sans forcément l'implémenter) d'y remédier.

Nous avons effectivement constaté un tremblement sur le flèche.

Cela est probablement dû à la précision des capteurs. En effet, les valeurs retournés par les capteurs comporte du "bruit" car même si l'appareil ne bouge pas, les valeurs retournées sont légèrement différentes. Ces différences peuvent être plus ou moins grande. A cet effet, chaque capteur indique son niveau de précision avec les valeurs suivantes :

```
SENSOR_STATUS_ACCURACY_HIGH : 3
SENSOR_STATUS_ACCURACY_MEDIUM : 2
SENSOR_STATUS_ACCURACY_LOW : 1
SENSOR_STATUS_UNRELIABLE : 0
SENSOR_STATUS_NO_CONTACT : -1
```

Ainsi, plus la précision est basse, plus le bruit peut être élevé.

Pour avoir un rendu plus précis, une solution serait d'accepter uniquement les données des capteurs indiquant une haute précision. Cela peut être implémenté avec la méthode `onAccuracyChanged` qui est appelé dès que la précision d'un capteur que l'on écoute est changé. Il nous est alors possible de stocker la précision de nos capteurs ce qui nous permettra de filtrer uniquement les valeurs des capteurs indiquant un précision haute. Le problème avec cette solution est que si, pendant un periode de temps, aucun capteur n'indique un précision haute, la boussole ne sera pas du tout mise à jour. Même en cas de mouvement de l'appareil.

Une autre solution serait de "lisser" le bruit en appliquant une moyenne des valeurs sur le temps. Ainsi, une valeur qui change brusquement sans raison serait adoucie à l'affichage.

Sources :

- onAccuracyChanged : https://developer.android.com/reference/android/hardware/SensorEventListener#onAccuracyChanged(android.hardware.Sensor,%20int)
- SENSOR_STATUS_ACCURACY : https://developer.android.com/reference/android/hardware/SensorManager#SENSOR_STATUS_ACCURACY_HIGH

3. Communication *Bluetooth Low Energy*

> La caractéristique permettant de lire la température retourne la valeur en degrés Celsius, multipliée par 10, sous la forme d'un entier non-signé de 16 bits. Quel est l'intérêt de procéder de la sorte ? Pourquoi ne pas échanger un nombre à virgule flottante de type float par exemple ?

La précision de la valeur est suffisante à 0.1 le capteur n'est probablement pas plus précis. Donc envoyer un entier puis le diviser par 10 permets d'utiliser moins de bit pour la précision de la virgule et ainsi de couvrir un plus grand interval avec le même nombre de bits.
Avec 8 bits on pourrait aller jusqu'a une température de 25.6 degré (256/10) ce qui est insufisant. 16 bits sont alors nécessaire. Il n'est pas possible de représenter des valeurs négative mais dans le cas d'utilisation la température est rarement en dessous de 0 car pour le laboratoire c'est à l'intérieur qu'on le fait sinon on a trop froid.. On pourrait choisir 16bits signé pour couvrir les valeurs de −3276,8 a 3276,7 ce qui pourrait couvrir les valeurs négative ainsi que toutes les positives probable.

> Le niveau de charge de la pile est à présent indiqué uniquement sur l'écran du périphérique, mais nous souhaiterions que celui-ci puisse informer le smartphone sur son niveau de charge restante. Veuillez spécifier la(les) caractéristique(s) qui composerai(en)t un tel service, mis à disposition par le périphérique et permettant de communiquer le niveau de batterie restant via Bluetooth Low Energy. Pour chaque caractéristique, vous indiquerez les opérations supportées (lecture, écriture, notification, indication, etc.) ainsi que les données échangées et leur format.

Pour obtenir le niveau de charge du périphérique, nous pouvons nous baser sur la spécification GATT du battery service disponible à l'adresse suivante : https://www.bluetooth.org/docman/handlers/downloaddoc.ashx?doc_id=245138

Opérations	**Exigences**
Broadcast	exclu
Read	obligatoire
Write without response	exclu
Write	exclu
Notify	optionnel
Indicate	exclu
Signed write	exclu
Reliable write	exclu
Writable auxiliaries	exclu

Le service est lu conformément à la procédure GATT et retourne le pourcentage de niveau de batterie entre 0% et 100%.

4. Conclusion

Nous avons particulièrement apprécié ce laboratoire car il nous a permis de nous familiariser avec la communication BLE ainsi qu'avec les données environnementales. Le fait d'utiliser de vrais écrans externes a permis de rendre le travail plus concret. Nous avons également pu mieux nous rendre compte des possibilité liées à l'usage du protocole BLE.