

# TP 2

## Les Capteurs

Chamy Kaci  
Ingo Diab

Mars 2023



## Table des matières

<b>1</b>	<b>Exercice 1</b>	<b>3</b>
<b>2</b>	<b>Exercice 2</b>	<b>4</b>
<b>3</b>	<b>Exercice 3</b>	<b>5</b>
<b>4</b>	<b>Exercice 4</b>	<b>6</b>
<b>5</b>	<b>Exercice 5</b>	<b>7</b>
<b>6</b>	<b>Exercice 6</b>	<b>8</b>
<b>7</b>	<b>Exercice 7</b>	<b>9</b>

## 1 Exercice 1

### Liste de capteurs

Dans cet exercice, nous pouvons lister tous les capteurs présents dans notre téléphone en utilisant `getSensorList(Sensor.TYPE_ALL)` sur le sensor manager.

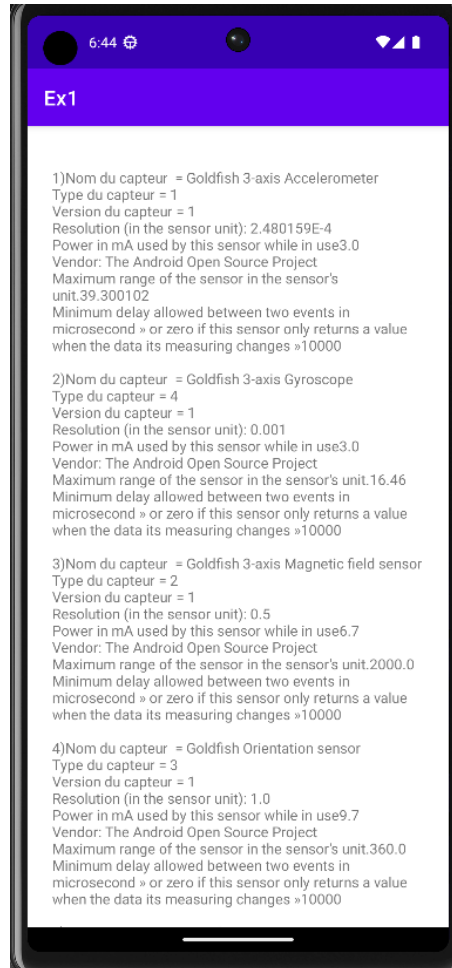


FIGURE 1 – Liste des capteurs présent dans le téléphone

## 2 Exercice 2

### Détection de présence/absence de capteurs

Dans cet exercice, nous avons demandé au téléphone s'il possède un capteur de type TYPE\_HEART\_RATE, ce qui n'est pas le cas donc l'application Santé ne peut pas être lancée

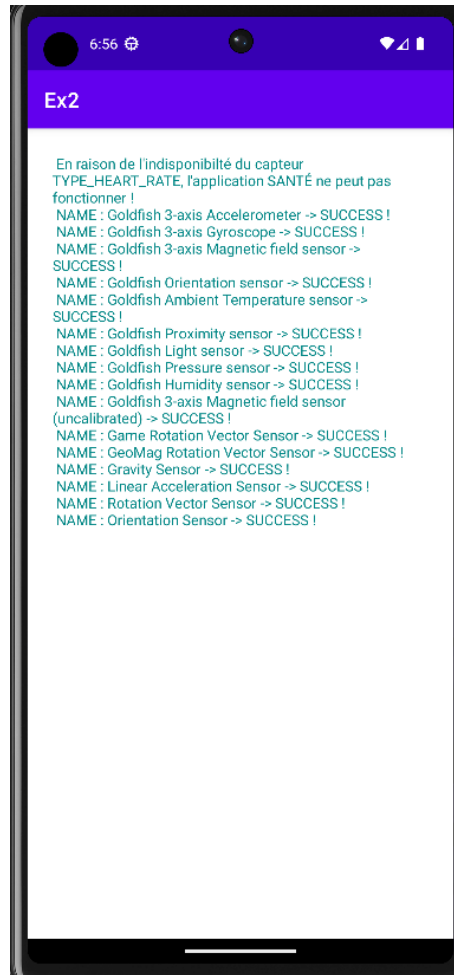


FIGURE 2 – Absence d'un capteur dans le téléphone

### 3 Exercice 3

#### Accéléromètre

Dans cet exercice, nous utilisons l'accéléromètre. Nous calculons la norme  $N$  du vecteur accélération avec  $x$ ,  $y$  et  $z$  les valeurs fournis par l'accéléromètre. Nous utilisons un `LINEAR_ACCELERATION` qui est un accéléromètre qui ne tient pas compte de la gravité ni de la rotation de l'écran car on veut que l'application ne détecte que les mouvements du téléphone.

$$N = \sqrt{x^2 + y^2 + z^2} \quad (1)$$

Nous pouvons contrôler la borne minimal et la borne maximal. Si la norme du vecteur accélération est inférieur à la borne minimal, le background est vert, si la norme est entre la borne minimal et la borne maximal, le background est jaune, si la norme est supérieur à la borne maximal le background est rouge.

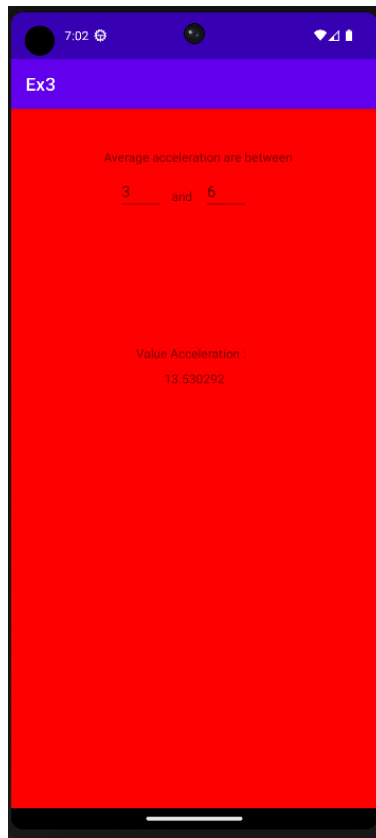


FIGURE 3 – Accélération supérieur à la borne maximal (6)

## 4 Exercice 4

### Direction

Dans cet exercice, nous utilisons aussi un `LINEAR_ACCELERATION` pour ne pas tenir compte de la gravité ni de la rotation du téléphone. Nous récupérons les valeurs de l'accéléromètre sur l'axe X et Y. Si la valeur sur l'axe X est positive alors le téléphone va vers la droite, si la valeur est négative alors il va vers la gauche. Sur l'axe Y, si la valeur est positive alors le téléphone va vers le haut, si la valeur est négative alors il va vers le bas. On peut aussi combiner les deux axes, quand l'accéléromètre a une valeur positive sur X ET Y, alors le téléphone va vers le haut à droite (en diagonale), etc.



FIGURE 4 – Accéléromètre avec une valeur positive sur l'axe X et sur l'axe Y

## 5 Exercice 5

### Secouer un appareil

Dans cet exercice, nous utilisons aussi un `LINEAR_ACCELERATION` pour ne pas tenir compte de la gravité ni de la rotation du téléphone. A chaque changement de valeur de l'accéléromètre, nous calculons la norme  $N$  du vecteur accélération avec  $x$ ,  $y$  et  $z$  les valeurs fournis par l'accéléromètre.

$$N = \sqrt{x^2 + y^2 + z^2} \quad (2)$$

On calcule ensuite la différence entre la norme de ce vecteur accélération et la norme du vecteur accélération précédent. Si la différence est supérieur à une certaine sensibilité, alors le téléphone a été secoué et le flash de la caméra arrière s'allume.

Android Studio ne gère pas le flash de la caméra donc on ne peut pas montrer le résultat mais voici la View de notre application où nous pouvons changer la sensibilité nécessaire pour que le flash s'allume. En effet, selon le constructeur du téléphone, l'accéléromètre semble être plus ou moins sensible.

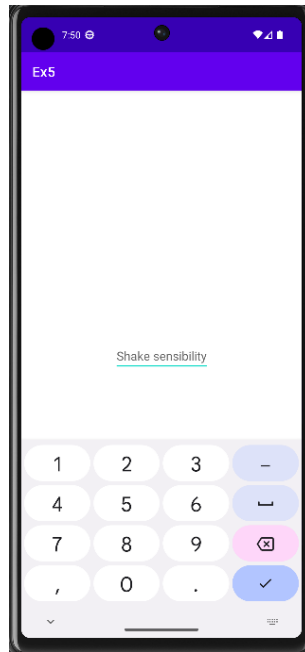


FIGURE 5 – View de l'application permettant de choisir la sensibilité nécessaire pour allumer le flash

## 6 Exercice 6

### Proximité

Dans cet exercice, nous allons utiliser le capteur de type `TYPE_PROXIMITY`. La View nous permet de choisir une borne minimale en dessous de laquelle l'objet est considéré comme trop proche et une borne maximale au dessus de laquelle l'objet est considéré comme trop éloigné.

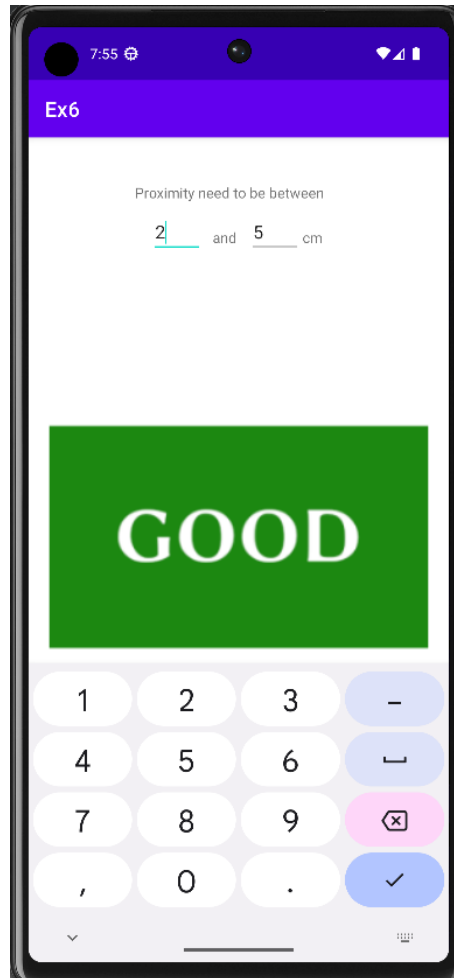


FIGURE 6 – Capteur de proximité détectant un objet à 3 cm



## 7 Exercice 7

### Géolocalisation

Dans cet exercice, nous avons accès à Google Map. Si l'utilisateur accepte de partager sa géolocalisation (nous utilisons le `LocationManager.GPS_PROVIDER`), alors l'application créera un marqueur à sa position et la caméra zoomera à cette position. On peut bouger la caméra autour de la position et il existe un bouton (coin inférieur droit de la View) permettant de recentrer la caméra sur la position de l'utilisateur.

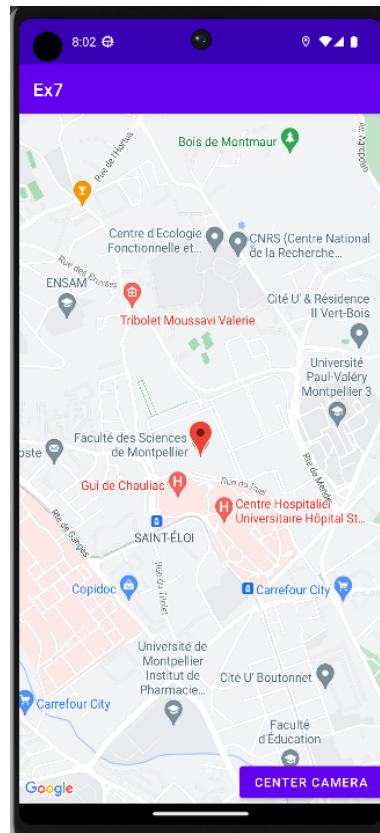
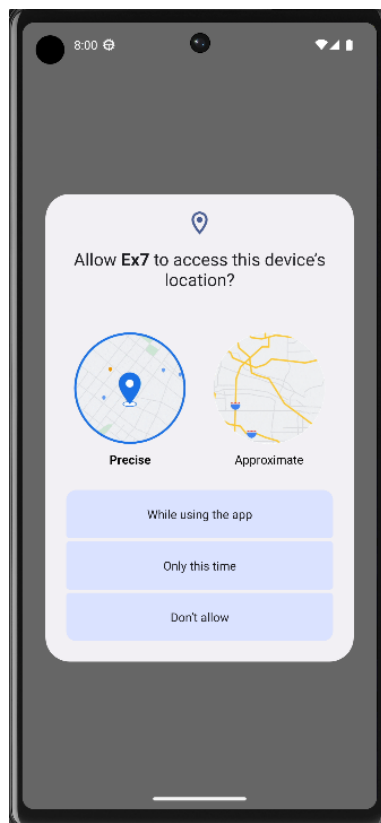


FIGURE 7 – Application demandant la permission d'utiliser la géolocalisation

FIGURE 8 – Géolocalisation d'un utilisateur étant sur le campus de la fac