



UNIVERSITÉ DES SCIENCES DE MONTPELLIER
HAI811I : PROGRAMMATION MOBILE

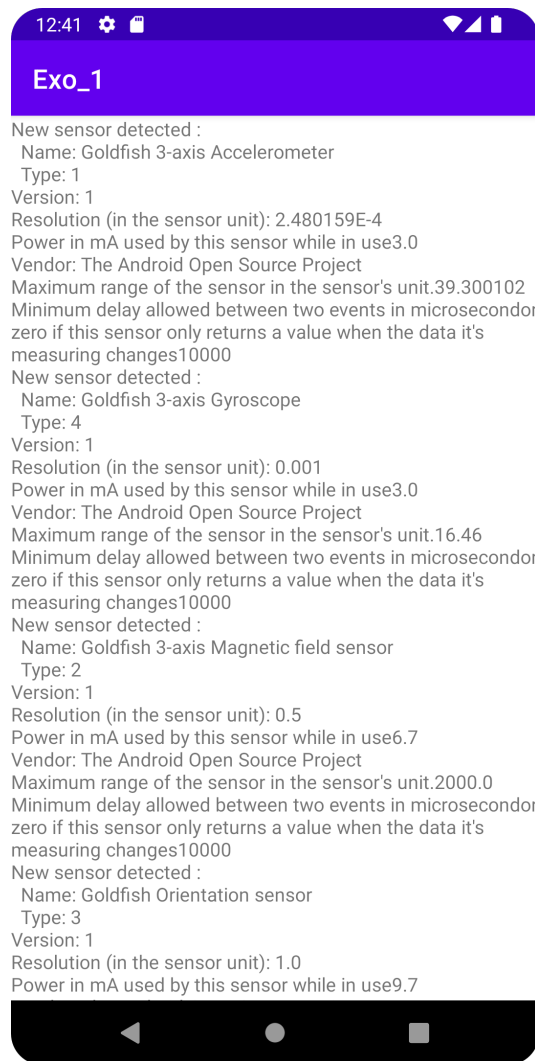
Rendu TP2

Anyaknine, Lucas Rouquairol

20/02/2023

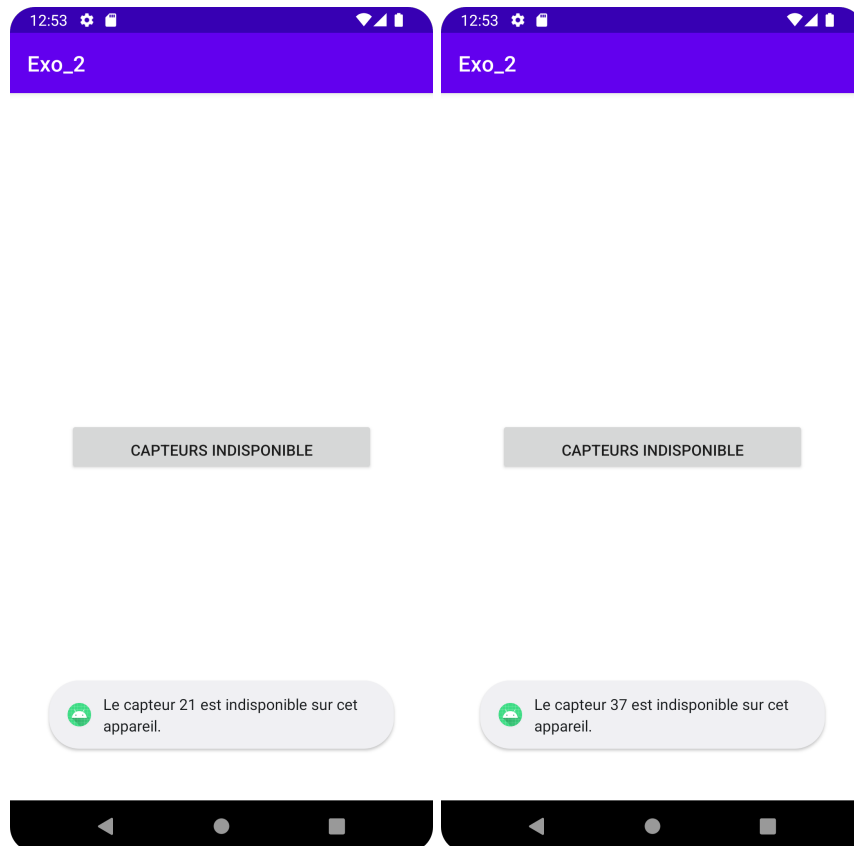
Exercice 1 : Liste de capteurs

On stocke toutes les infos de tous les types de sensors dans un StringBuffer (exemple du cours) puis on l'affiche dans un **TextView** lui même incorporé dans un **ScrollView** afin d'affichers toutes les infos sur un seul linéar layout.



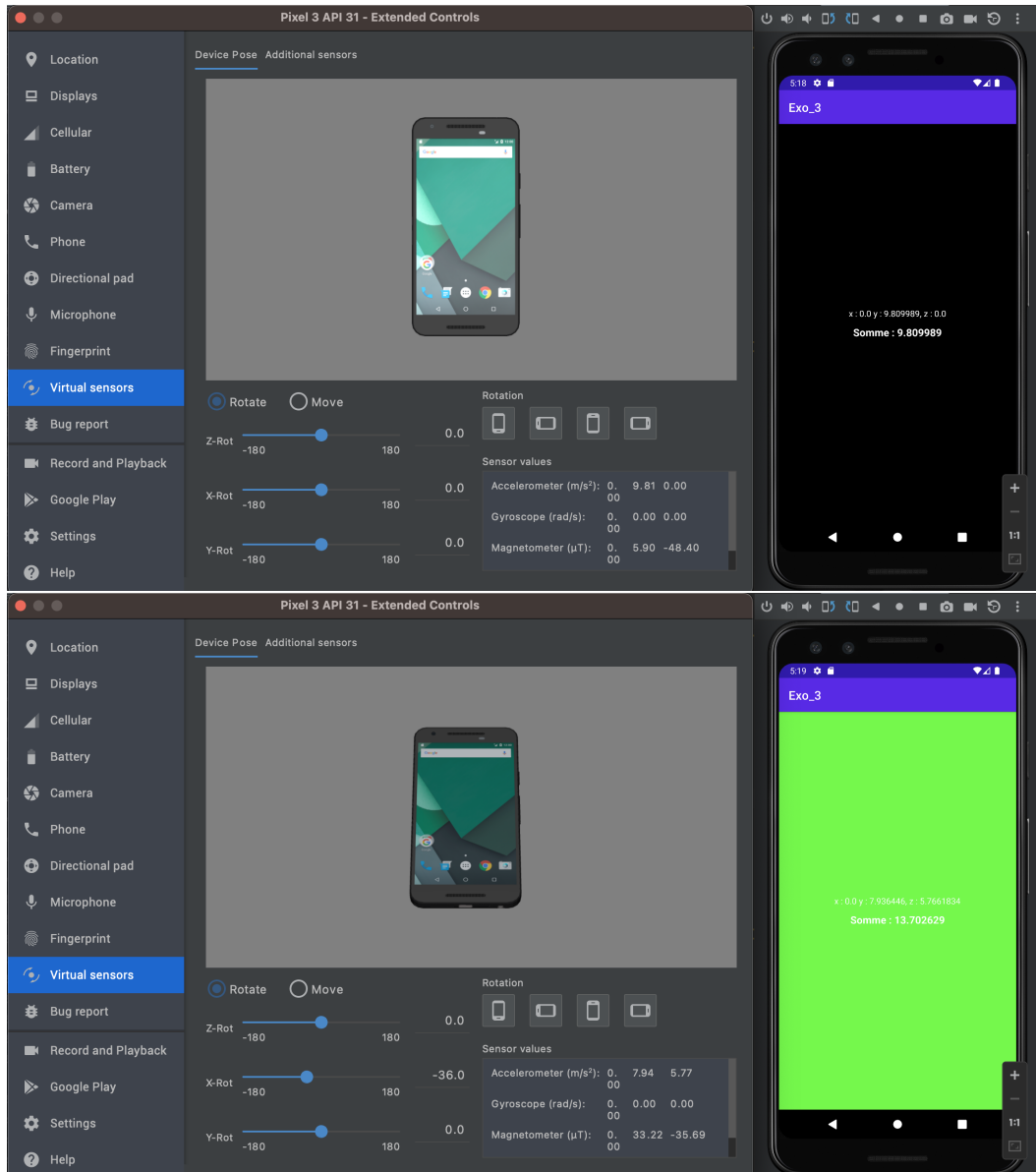
Exercice 2 : Détection de présence/absence de capteurs

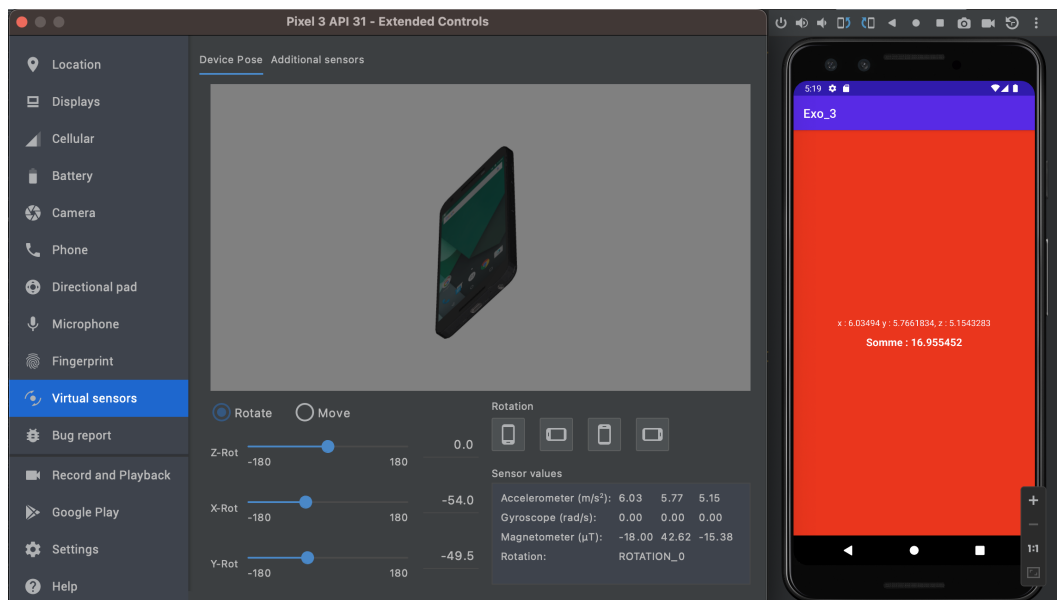
On crée une liste de types de capteurs pour tester notre application. Sachant que sur l'appareil que j'utilise pour les tests, les sensors HEAD_TRACKER et HEART_RATE ne sont pas disponibles, on les ajoute à la liste. On a simplement à tester chaque sensor de la liste pour voir si ils sont "null". Si oui, cela veut dire qu'ils ne sont pas disponibles. On affiche alors un **Toast** pour prévenir l'utilisateur.



Exercice 3 : Accéléromètre

Pour cette exercice, on récupère simplement les trois valeurs x, y et z absolue du sensor ACCELEROMETRE et on vérifie si la somme des trois dépasse une certaine valeur associé au couleurs : noir(0 - 13), vert(13 - 16) et rouge(16 et plus).





Exercice 4 : Direction

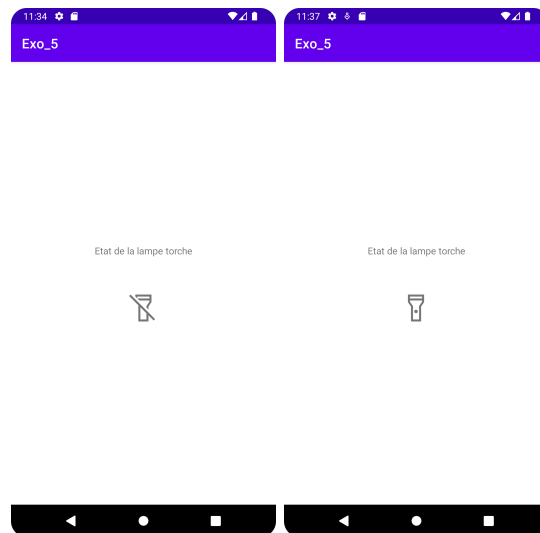
Pour calculer la direction du déplacement, on utilise aussi le sensor ACCELEROMETRE sans calculer les valeurs absolues. On utilise le fait que :

- Si x est négatif \Rightarrow roulis à droite
- Si x est positif \Rightarrow roulis à gauche
- Si y est négatif \Rightarrow bascule vers le haut
- Si y est positif \Rightarrow bascule vers le bas

Il suffit maintenant de comparer deux valeurs de chaque axe à un instant t et $t-1$ en rajoutant un biais pour ne capter que les variations significatives ([vidéo exemple en complément du rendu](#)).

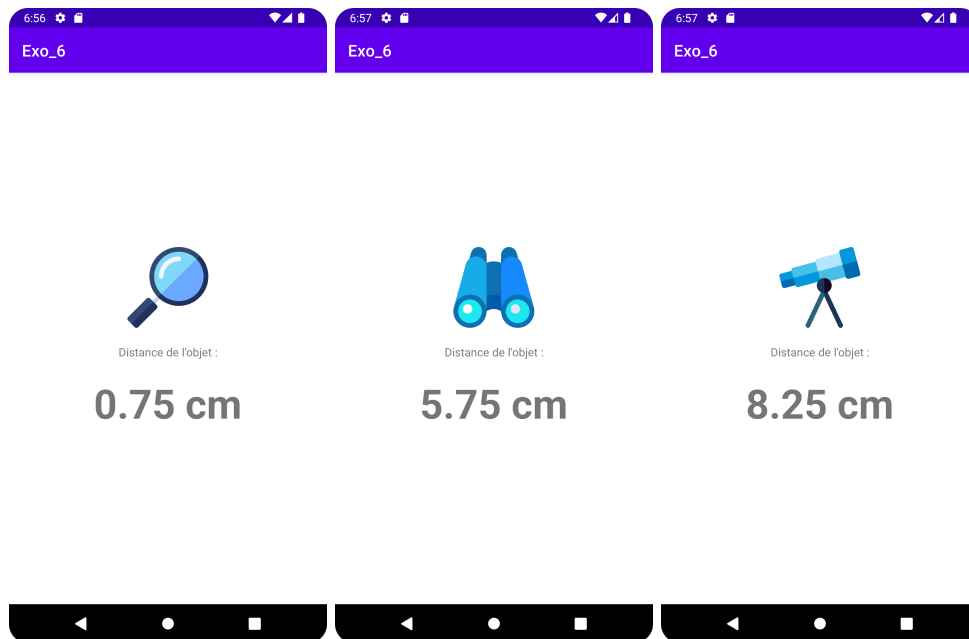
Exercice 5 : Secouer un appareil

On reprend la même base que l'exercice 3 avec une seule valeur plus élevée qui va permettre de déclencher le flash. On déduit l'allumage en comparant la valeur de l'accéléromètre calculé précédemment plus un biais avec la valeur actuelle de l'accéléromètre. On utilise le service CAMERA_SERVICE pour accéder au flash que l'on utilisera comme lampe torche. On rajoute une image qui change en fonction de l'état du flash pour tester dans l'émulateur ([vidéo exemple en complément du rendu](#)).



Exercice 6 : Proximité

On crée une liste de types de capteurs pour tester notre application. Sachant que sur l'appareil que j'utilise pour les tests, les sensors HEAD_TRACKER et HEART_RATE ne sont pas disponibles, on les ajoute à la liste. On a simplement à tester chaque sensor de la liste pour voir si ils sont "null". Si oui, cela veut dire qu'ils ne sont pas disponibles. On affiche alors un **Toast** pour prévenir l'utilisateur ([vidéo exemple en complément du rendu](#)).



Exercice 7 : Geolocalisation

Le but de l'exercice est d'afficher une carte centrée sur les coordonnées gps du téléphone. Pour cela, on utilise la bibliothèque *"osmdroid"* qui est open-source et qui permet la visualisation de cartes en utilisant les données OpenStreetMap. Cette fois, au lieu de récupérer un **sensor**, on va utiliser le service **LOCATION** qui va permettre de récupérer la position GPS du téléphone. Après avoir rajouté les autorisations de géolocalisation dans le manifest, on peut afficher la position du téléphone sur une **MapView** grace à un **Marker**.

