

**ANCIEN DEVOIR FINAL**

Classe : SEM31

Matière : *Développement Mobile Avancé*

Nb pages : 4

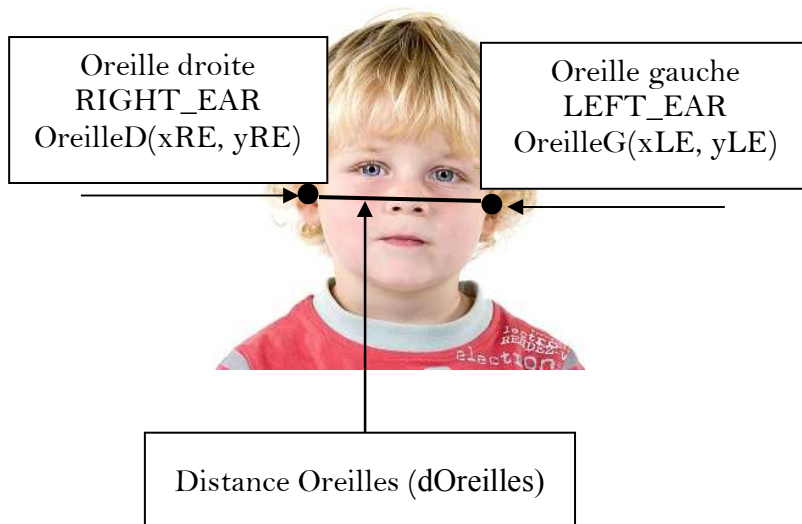
Enseignant : *Souissi Hafedh*

Documents Non Autorisés

Durée : 1 heure 30 minutes

**Exercice 1**

"Stat Visage" est une application Android qui permet de détecter les visages d'une image et d'afficher un ensemble de statistiques sur les visages détectés (Nb Visages, Nb Yeux Droits fermés, Nb Yeux Gauches Fermés et Distance Oreilles Moyenne). "Stat Visage" contient une seule activité.



L'œil droite d'un visage est fermé si son "is right eye open probability" est inférieur à 0.5.

L'œil gauche d'un visage est fermé si son "is left eye open probability" est inférieur à 0.5.

Pour obtenir la distance "dOreilles" entre les deux points OreilleD(xRE, yRE) et OreilleG(xLE, yLE), utiliser la formule :

«  $dOreilles = \text{Math.sqrt}(\text{Math.pow}(xRE - xLE, 2) + \text{Math.pow}(yRE - yLE, 2))$  ».

La Distance Oreilles Moyenne = somme des distances dOreilles / nombre de visage.

La méthode "analyser()" permet de détecter les visages de l'image, de calculer et d'afficher :




- le nombre de visages,
- le nombre d'yeux droits fermés,
- le nombre d'yeux gauches fermés,
- La distance moyenne entre les oreilles des visages détectés.

1- Donner le code de la méthode "analyser()" de la classe "MainActivity".

## Exercice2 (Utiliser Annexe II)

L'objectif de cet exercice est de réaliser une application android qui s'utilise dans une voiture pour connaître l'état du revêtement de la route. L'état de revêtement de la route peut être "BON" (pas de fissure et pas de trous), "Mauvais" (fissures mais pas de trous) ou "Très Mauvais" (fissures et trous).

Le smartphone qui contient l'application est placé sur le tableau de bord de la voiture écran vers le haut. C'est l'accélération sur l'axe des Z qui permet de déduire l'état de revêtement de la route. Le tableau suivant présente l'état de la route et l'image à afficher en fonction de l'accélération sur l'axe des Z :

| Accélération sur l'axe des Z (accZ)         | Etat         | Image à afficher  | Id de l'image à afficher    |
|---|--------------|---|-----------------------------|
| $\text{Math.abs}(\text{accZ}) < 13$         | BON          |    | R.drawable.rev_bon          |
| $13 \leq \text{Math.abs}(\text{accZ}) < 16$ | MAUVAIS      |   | R.drawable.rev_mauvais      |
| $16 \leq \text{Math.abs}(\text{accZ})$      | TRES MAUVAIS |  | R.drawable.rev_tres_mauvais |

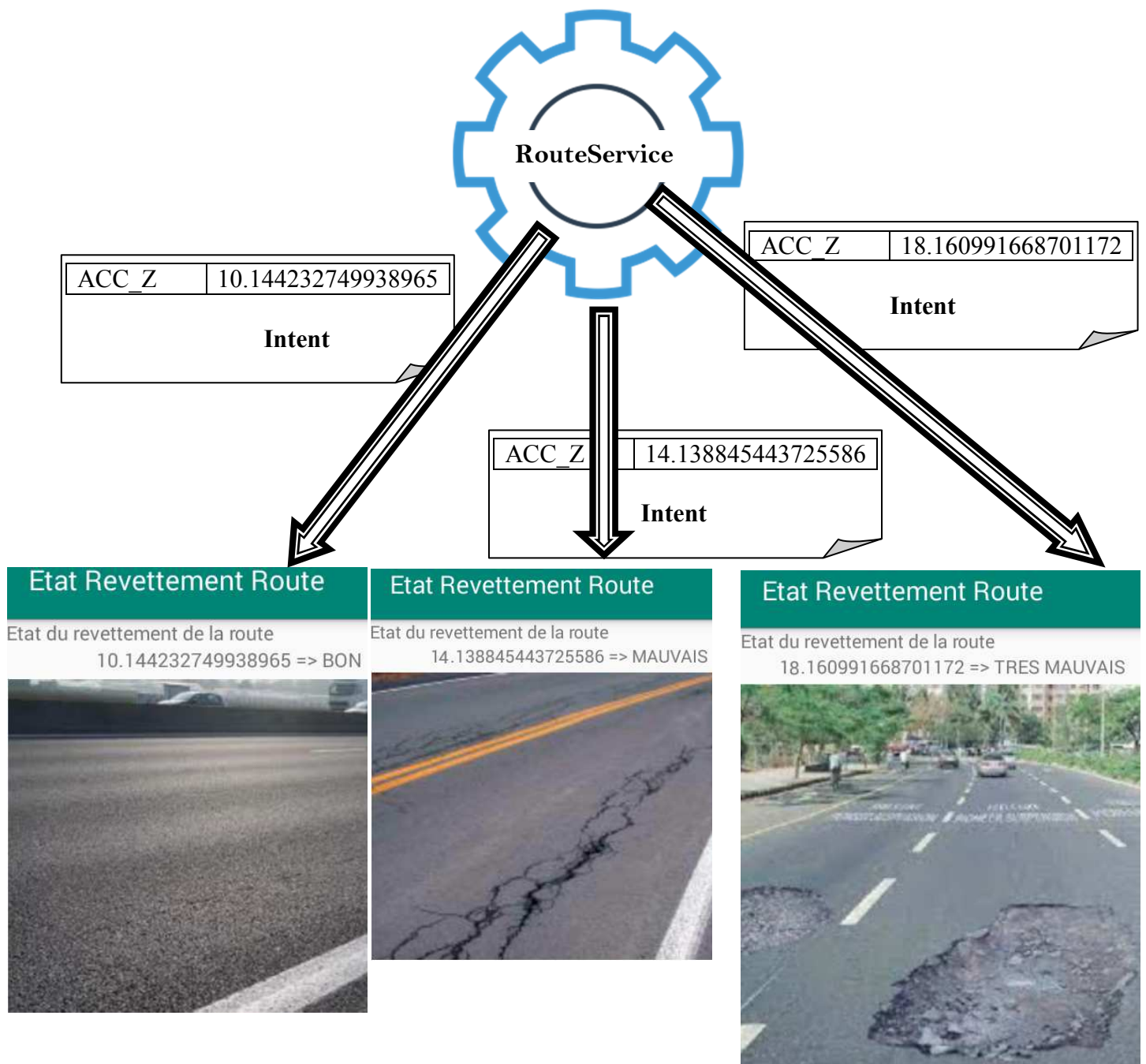
"**EtatRevettementRoute**" est une application Android qui permet d'afficher l'accélération sur l'axe des **Z** interceptée par le capteur d'accélération et de modifier l'image affichée suivant la valeur interceptée en respectant le tableau précédent.

"**EtatRevettementRoute**" est composée d'une activité "**MainActivity**" et d'un service "**RouteService**". "RouteService" utilise un capteur de type TYPE\_ ACCELEROMETER pour intercepter les modifications de l'accélération sur l'axe des Z "accZ", par la suite il la diffuse (broadcast) dans un Intent.

Lorsque "MainActivity" fait un "Resume" elle démarre le service et lorsqu'elle fait une "Pause" elle arrête le service. "MainActivity" utilise un BroadcastReceiver pour lire la valeur "accZ" envoyée par le service et l'afficher avec son interprétation dans "tvEtat". Suivant la valeur de "accZ", "imgEtat" affiche IMG\_BON ou IMG\_MAUVAIS ou IMG\_TRES\_MAUVAIS.

Le texte affiché dans "tvEtat" est de la forme suivante : accZ => Etat, par exemple pour une accZ=12.59837652, le texte affiché est "12. 59837652 => BON".

Ci-dessous, trois exemples de fonctionnement de l'application :



1- Donner le code des méthodes "ajouterEcouleur()", "demarrerService()", "arreterService()", "onResume()", "onPause()" et "actualiser(...)" de la classe "MainActivity".

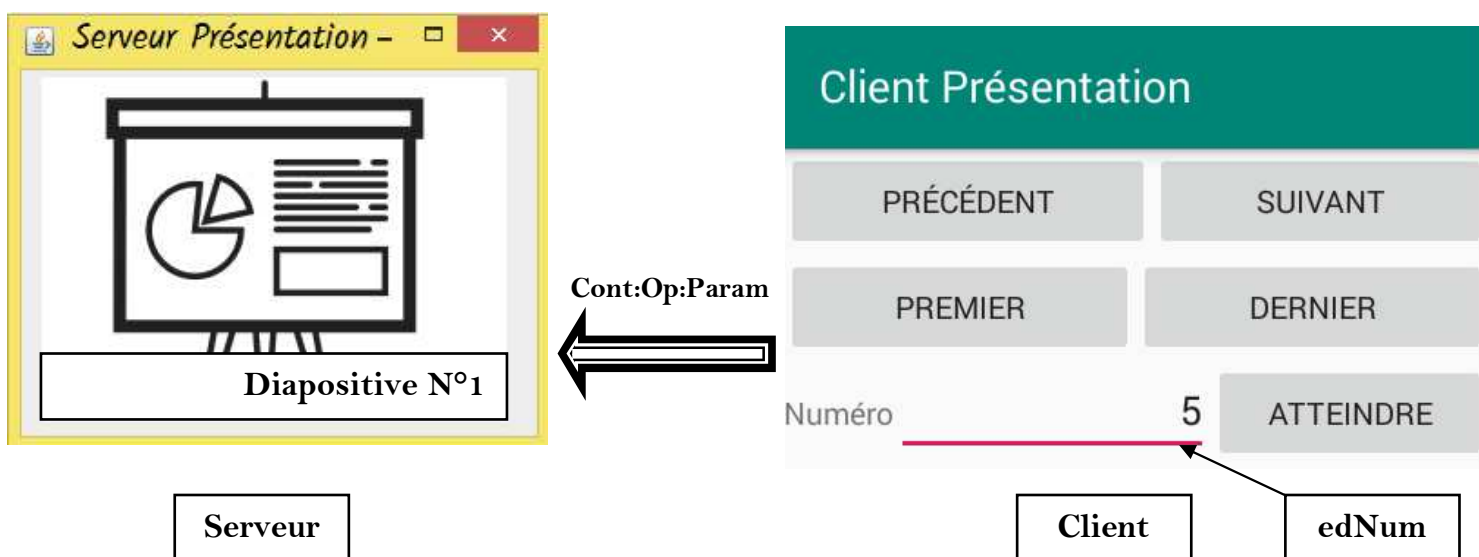
2- Donner le code des méthodes "init()", "onDestroy()" et de "onSensorChanged()" de la classe "RouteService".

### Exercice3 (Utiliser Annexe III )

On désire développer une application qui permet de contrôler une présentation composée de plusieurs diapositives par un smartphone. Cette application utilise les sockets. Le programme serveur est installé sur un ordinateur d'adresse "192.46.10.20". Le programme client est installé sur un Smartphone android pour envoyer les commandes de l'utilisateur au serveur. Les contrôles qu'on peut effectuer sur la présentation et les messages envoyés par le socket clients sont résumés dans le tableau suivant :

| Contrôle (Cont) | Opération (Op) | Paramètre (Param) | Message envoyé |
|-----------------|----------------|-------------------|----------------|
| PrecSuiv (0)    | Précédent (0)  |                   | 0:0            |
|                 | Suivant (1)    |                   | 0:1            |
| PremDern(1)     | Premier(0)     |                   | 1:0            |
|                 | Dernier (1)    |                   | 1:1            |
| Att (2)         | Atteindre (0)  | numero            | 2:0:numero     |

Le fonctionnement de l'application est comme suit :



- 1-Donner le code des méthodes " lancerThreadServeur()", "demarrerServeur()" et "executerCommande(...)" de la classe "**ServPresentation**" (coté serveur).
- 2-Donner le code des méthodes " lancerThreadClient()", "demarrerClient()", "envoyer() ", "precedent() ", "suivant() ", "premier() ", "dernier()" et "atteindre()" de la classe "**MainActivity**" (coté client).