

Systèmes mobiles

Laboratoire n°4 : Android Wear

18.12.2018

Cet exercice est constitué de manipulations de laboratoire qui vont vous permettre de vous familiariser avec l'utilisation de la plateforme *Wear OS*. Il n'y a pas de rapport à rendre pour ce laboratoire, uniquement votre code correctement commenté. La date de rendu est le **dimanche 27.01.2019** à 23h55.

1. Environnement de développement

Afin de pouvoir tester vos applications *Android Wear*, il faut disposer au minimum d'un smartphone physique ainsi que d'un émulateur de montre *Wear OS*. La Fig. 1 indique comment créer l'émulateur d'une montre. Il faut obligatoirement installer la version de l'OS la plus récente (Pie 9.0 – API 28).

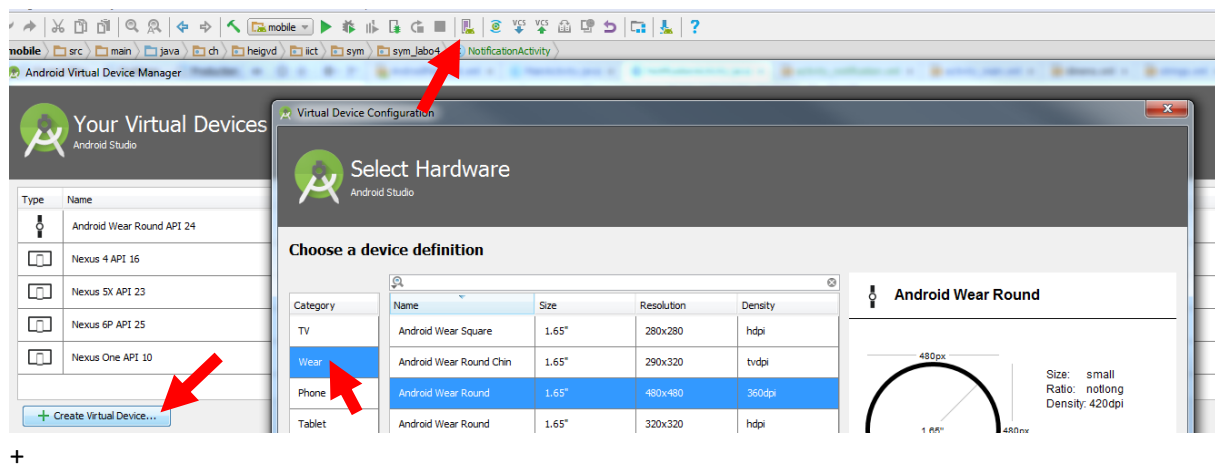


Figure 1 - Créer un émulateur Android Wear

L'appairage d'un émulateur de smartphone avec un émulateur de montre n'est pas supporté officiellement. Il vous faudra donc disposer au moins d'un téléphone physique.

Une fois l'émulateur Android Wear démarré et le smartphone branché par USB, vous pouvez exécuter la commande `adb devices`, vous permettant de lister les devices (physiques + émulés) branchés sur votre machine. La Fig. 2 présente un résultat de cette commande, on peut voir l'émulateur (emulator-5554) ainsi que le téléphone réel (CB5A28DUJ9).

```
PS C:\Users\fabien.dutoit> adb devices
List of devices attached
emulator-5554    device
CB5A28DUJ9      device
```

Figure 2 - Liste des périphériques Android connectés

Vous devrez installer sur votre téléphone l'application *Android Wear*, qui permettra d'appairer et de gérer la montre connectée : le QR code de la Fig. 3 vous permet de l'installer plus facilement, le lien sur le Play Store : <https://play.google.com/store/apps/details?id=com.google.android.wearable.app>



Figure 3 - Installation de l'application *Android Wear* sur votre téléphone

Vous devrez ensuite exécuter la commande

```
adb -d forward tcp:5601 tcp:5601
```

qui permettra aux deux devices de communiquer par USB.

Ouvrez l'application *Wear OS* sur le téléphone, cliquez sur configurer, acceptez les demandes d'autorisations. Vous allez arriver sur une page de recherche, cliquez sur le bouton menu (trois petits points, en haut à droite) puis sélectionnez *Associer à un émulateur*. Continuez la procédure selon les indications présentes à l'écran. Pour tester que tout est fonctionnel, vous pourrez sélectionner différents cadrans de montre depuis votre téléphone, l'émulateur doit se mettre à jour en conséquence.

La documentation officielle vers l'installation et la configuration de l'environnement de développement : <https://developer.android.com/training/wearables/apps/creating.html>.

Vous trouverez la documentation générale et officielle sur le développement Android Wear ici : <https://developer.android.com/training/building-wearables.html>

Nous vous fournissons un template d'application que vous importerez dans *Android Studio*. Vous noterez que le projet comporte 3 modules gradle, un pour l'application smartphone (mobile), un pour la montre (wear) et un avec le code en commun (common). Pour la première manipulation, nous allons uniquement nous intéresser à l'application téléphone.

2. Notifications

Le mode le plus simple pour intégrer une smartwatch à votre application est l'utilisation des notifications. Votre application ne saura pas si une montre est disponible, les notifications générées sont compatibles avec *Wear OS* et si l'utilisateur en possède une il pourra en profiter automatiquement. La documentation officielle est disponible ici :

<https://developer.android.com/training/wearables/notifications/creating.html>

Vous devrez compléter l'activité *NotificationActivity*, vous ajouterez plusieurs boutons permettant de lancer différents types de notifications :

- Notification simple (avec *PendingIntent*)
- Notification avec des *Action Buttons* (visibles sur le téléphone et la montre)
- Notification avec des *wearable-only Actions*.

Nous vous fournissons le code permettant de créer des *PendingIntent* qui afficheront un *Toast* sur le téléphone une fois une action cliquée sur la montre (ou le téléphone).

L'activité que vous proposerez devra permettre de tester facilement ces différents types de Notifications.

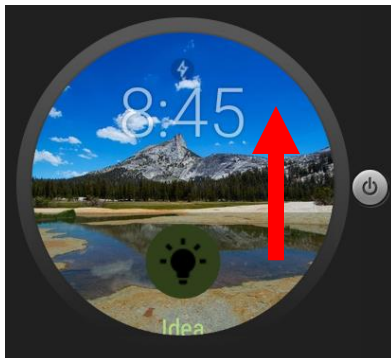


Figure 4 - Les notifications ne sont pas forcément visibles sur la montre, il faut glisser vers le haut pour les faire apparaître. Un vrai device vibrera pour notifier l'utilisateur

3. Application dédiée

Le but de cette manipulation est de faire communiquer une application dédiée (≠ autonome) sur une montre connectée avec l'application correspondante sur le smartphone à l'aide de la *Wearable Data Layer API*¹. Nous vous fournissons un template d'application pour montre (module wear du projet fourni) permettant la saisie d'une couleur à l'aide de 3 sliders, voir la capture d'écran en Fig. 5.



Figure 5 - Capture d'écran de l'application "montre" fournie avec les 3 sliders

Vous devrez ensuite utiliser la *Wearable Data Layer API* pour permettre à l'activité *MainActivityWear* (module wear) d'envoyer les composantes couleurs (RGB) à l'activité *WearSynchronizedActivity* (module mobile). L'activité sur le mobile modifiera sa couleur de fond en fonction de la couleur sélectionnée sur les sliders de la montre et vice-versa.

Vous ajouterez 3 sliders sur l'activité « smartphone », ceux-ci seront à synchroniser avec ceux de l'activité « montre » (dans les deux sens). Les fonds des deux activités changeront de couleur en fonction des trois composantes couleurs sélectionnées (on mettra à jour uniquement lorsque l'utilisateur lâche le slider). Vous veillerez aussi à ce que les sliders (et la couleur de fond) soient mis à jour si l'utilisateur quitte et revient sur l'activité ou lors d'un changement de contexte (rotation du smartphone).

¹ Documentation *Wearable Data Layer API*: <https://developer.android.com/training/wearables/data-layer/index.html>

Quelques remarques

- Il est important d'avoir la dernière version des *Google Play Services* sur votre montre ou sur l'émulateur. Pour réaliser la mise à jour, il faut synchroniser son compte Google depuis l'application *Android Wear*, les téléphones mis à disposition sont configurés avec le compte iict.sym@gmail.com, vous pouvez demander à l'assistant d'entrer le mot de passe si demandé.
- Nous possédons quelques montres physiques, celles-ci sont déjà appairées avec un téléphone physique. Pour installer les 2 applications il faut préparer un apk signé (build -> Generate Signed APK) et l'installer sur le téléphone à l'aide d'adb (adb -d install -r /path/to.apk). L'application montre sera ensuite disponible depuis l'application play store sur la montre.
- Un appel à la méthode *getDataItems()* de la classe *DataClient* va retourner les dernières valeurs enregistrées sur tout les périphériques (montre + smartphone). A vous de garder que la plus récente.