

Lors d'un projet pluridisciplinaire de 3 semaines en IUT Informatique, j'ai eu l'occasion de travailler sur le développement d'une application Android permettant à l'utilisateur de sélectionner des images et de les faire défiler sur l'écran. L'image actuelle est alors retransmise sur tous les appareils connectés à une page Web d'un serveur. Ce projet nous a permis d'avoir une première approche du développement sur Android, mais nous avons eu beaucoup de difficultés à effectuer des manipulations d'images.

Cette année, j'ai pu rapidement "reprenre en main" le SDK Android, en utilisant cette fois-ci Android-studio, amené à devenir le SDK officiel, et de développer deux applis de démonstration exploitant des fonctionnalités du SDK indispensables à la réalisation d'une application permettant la prise de contrôle d'un ordinateur.

Lors du développement de la première application de démo, j'ai pu me familiariser avec le module de gestion de version intégré à android-studio, à la manipulation de "fragments", bien utiles pour que l'affichage s'adapte sur tablettes et smartphone, par exemple.

J'ai pu remarquer une nouvelle fois la qualité de la documentation Android:

<http://developer.android.com/guide/index.html>

Avec elle, j'ai pu récupérer les infos de nombreux capteurs présents sur le téléphone, et les afficher.

Dans le cas de notre projet, il faudra néanmoins interpréter correctement les valeurs retournées par les capteurs comme l'accéléromètre par exemple, afin de déterminer où pointe l'appareil par exemple.

En ce qui concerne la deuxième application de démonstration, elle doit prouver la possibilité d'établir une communication par socket entre un PC et un appareil Android, connectés au même réseau local. Il a d'abord fallu créer un serveur en JAVA qui attend une connexion. Une fois la communication Android-PC établie, je suis parvenu à utiliser la classe "Robot" de JAVA côté serveur, pour déplacer la souris lorsque l'application Android envoie une commande.

Le développement de ces 2 applications a permis de mettre en évidence la faisabilité d'une communication entre appareil et PC d'un même réseau, la possibilité de récupérer les informations des capteurs d'Android, et enfin, la capacité de la classe Robot à simuler des entrées systèmes (clavier et souris).

Après le développement de ces deux applications de démonstration, et la lecture de documentations sur internet, il est apparu que le projet devra s'articuler autour de ces composants:

-Classe Robot

<http://docs.oracle.com/javase/7/docs/api/java/awt/Robot.html>

Cette classe permet de générer des événements d'entrées systèmes, et sera ainsi exploitée pour contrôler les événements de la souris ou du clavier. Elle est également capable de renvoyer un "screenshot" de l'écran, ce qui nous servira pour déporter l'écran du PC sur l'appareil Android et effectuer plusieurs niveaux de zooms.

- Les classes *Sensor*, *SensorManager* et *SensorEvent*

<http://developer.android.com/reference/android/hardware/Sensor.html>

Pour récupérer les informations des capteurs. Il faut cependant les interpréter correctement par des calculs, pour obtenir la position pointée à l'écran par le téléphone, comme pour une "wiimote".

- La classe *Socket* <http://docs.oracle.com/javase/7/docs/api/java/net/Socket.html>

Pour établir une communication entre le téléphone et le PC et transmettre les données.

- L'accéléromètre, il sera utilisé pour repérer le point vers lequel l'appareil est dirigé, à la manière d'une "wiimote". Il faudra effectuer des calculs pour déterminer la position de l'appareil. Il n'est pas simple encore de savoir s'il faudra utiliser les données de l'accéléromètre et d'autres capteurs comme le gyroscope pour plus de précision. Il faut trouver une documentation valable.

- Les input events d'Android, qui permettent de gérer les événements liés au clic, au glissé, au multitouch. Sera utile pour gérer toutes les actions relatives à l'écran tactile. Voir <https://developer.android.com/training/gestures/index.html>

- Concepts fondamentaux d'Android, tels que les "Views", Activities, Services, Intents, Intent filters...

Une activité correspond à un écran, elle est le "cœur" de chaque écran de l'application, elle réagit en conséquence d'entrées utilisateur. Les Vues définissent quand à elles les éléments qui sont affichés à l'écran, définis dans des "layouts".

Les services correspondent à des programmes qui peuvent être lancés en arrière-plan pour une période donnée.

Les intents définissent des intentions de notre application. Si nous voulons par exemple faire appel à une application permettant de lire une vidéo, il suffit de créer un intent demandant d'ouvrir une vidéo. Il faut également créer un intent pour chaque passage

entre activités de notre application. On peut transmettre des données entre activité grâce à ces intents.

Il existe déjà une application permettant le contrôle de la souris avec l'accéléromètre du téléphone, voir <https://sites.google.com/site/accelerometermouse/>