# Mon application : Capture Photo

## C:\Users\Steven\Dropbox\Captures d'écran\Screenshot_2014-02-24-21-44-02.pngExplications

Ce TP a pour but de vous faire manipuler un capteur : l’appareil photo.

Pour complètement y arriver, vous devrez :

* Manipuler correctement l’objet Camera en respectant le cycle de vie de l’activité (comme pour le TP lampe torche).
* Avoir un layout permettant de prévisualiser la caméra (SurfaceView)
* Injecter un second layout comportant un bouton permettant le déclenchement de l’enregistrement d’une photo
* Détecter l’orientation du périphérique afin d’avoir les meilleurs paramètres de prévisualisation

Nous vous conseillons de lire le contenu de ce lien pour résoudre ce TP : <http://developer.android.com/training/camera/photobasics.html>

Ainsi que : <http://developer.android.com/training/camera/cameradirect.html>

## Ajout des permissions

Cette application nécessite deux autorisations : la première pour manipuler l’appareil photo, la seconde afin de pouvoir écrire nos photos sur le périphérique.



## Layout

Comme indiqué précédemment, le layout ne contient qu’un objet « SurfaceView ».

### Fullscreen

Afin d’avoir un rendu plein écran, il suffit d’appliquer le flag « fullscreen » à la fenêtre ainsi que demander une fenêtre sans titre :



### SurfaceView

C’est un moyen rapide et abstrait de manipuler une vue pixel par pixel.

Dans un premier temps, l’objet doit être initialisé ainsi que son « holder » :



L’objet précédemment initialisé nécessite la création du callback « SurfaceHolder.Callback », celui-ci va se charger de l’initialisation de la prévisualisation :



### « Overlay » : Injecter une vue

Afficher une vue au-dessus d’une autre, dans notre cas au-dessus de « SurfaceView », il suffit d’appeler la méthode « addContentView » de l’activité courante en précisant l’objet « View » préalablement généré à partir du fichier xml :



## OrientationEventListener

Afin de gérer correctement la prévisualisation de l’appareil photo, il est important de prendre en compte l’orientation du périphérique grâce à un « OrientationEventListener » :



Tachez de mettre à disposition l’orientation du périphérique aux autres méthodes (dans notre cas « orient »).

## Camera

La surface de prévisualisation est liée à l’appareil photo. Il est ensuite configuré sur deux aspects :

* L’autofocus, si cette fonctionnalité est disponible, nous l’activons
* La sélection de la taille du flux de prévisualisation. Il existe un nombre important de résolutions différentes. La fonction suivante permet de trouver la surface la plus importante en prenant en compte l’orientation du téléphone :



Code pour initialiser le prévisualisation :



## Prendre une photo

Il existe plusieurs méthodes pour prendre une photo, comme l’indique la documentation Android.

Dans notre cas, il est préférable d’utiliser la version avec le callback permettant d’avoir une meilleure maîtrise et pourquoi pas appliquer des filtres sur l’image avant sa sauvegarde.



Une fois la photo prise, la prévisualisation est automatiquement coupée. Vous devez la relancer manuellement.

« createImageFileName() » est une méthode permettant de générer un nom de fichier *unique* pour votre photo. (Il s’agit d’une version dérivée du portail développeur d’Android) :



## Mettre à jour la bibliothèque

Voici un petit morceau de code permettant de faire prendre en compte votre nouvelle photo par le système :



## Résultat

L’ensemble du projet « PhotoCapture » est disponible sur le dépôt github : <https://github.com/steven-martins/tp-android>

Une très grande partie du code contenu dans ce TP est issu de la page : <http://developer.android.com/training/camera/photobasics.html>