|  |
| --- |
| PST : Introduction à Google Tango |
| Acquisition d’une image de profondeur avec Google Tango |
|  |
| DELANGLE Flavien et SEURRE Maxime |
|  |

PST : Introduction à Google Tango

Acquisition d’une image de profondeur avec Google Tango

# Présentation de l’application :

Le but premier de notre PST était de se familiariser à l’environnement Tango et de proposer une ébauche d’application permettant d’accéder à la Depth Map afin de pouvoir y appliquer des traitements dans le futur.

La majeure partie de la logique logicielle de notre application a été développé en C++. Nous avions dans un premier temps essayé d’effectuer les traitements liés à Tango avec l’API Java mais il s’est avéré que nous ne pouvions pas accéder à la caméra de couleur et au capteur de profondeur simultanément. Nous avons donc basé notre projet sur un des exemples C++ nommé *cpp\_rgb\_depth\_sync\_example*.

# Fonctionnalités :

* Affichage au choix de la vidéo RGB ou de la Depth Map en cours d’enregistrement sur l’écran du smartphone
* Enregistrement de la vidéo RGB et de la Depth Map sur la carte SD du téléphone sous forme de fichiers JPEG
* Détecteur de visage mais on est pas sur parce qu’on a potentiellement la flemme voilà

# Communication Java ⬄ C++ : La JNI

## Appeler des fonctions Java depuis un fichier C++ :

**// Access the class *your.package.class***

jclass cls = env->FindClass("your/package/class");

if (cls == nullptr) {

LOGE("*Class not found*");

}

else {

// **Access the default constructor with <init>**

// **()V : no arguments, void return**

jmethodID constructor = env->GetMethodID(cls, "<init>", "()V");

if (constructor == nullptr) {

LOGE("*Constructor not found*");

}

else {

**// Create an instance of the class**

jobject object = env->NewObject(cls, constructor);

**// Access the method yourMethod**

**// Here, (I)I means 1 parameter of type Integer, return Integer)**

jmethodID method = env->GetMethodID(cls, "*yourMethod*", "(I)I");

if (method == nullptr) {

LOGE("*Method not found*");

}

else {

**// Call the method with the needed arguments**

env->CallVoidMethod(object, method, 5);

}

}

}

Remarque : Il est nécessaire d’avoir accès au pointeur d’environnement pour ces fonctions (*FindClass, NewObject, etc*). Pour ce faire il est conseillé, par exemple dans une méthode *onCreate* de stocker un pointeur *JavaVM* de la manière suivante :

JavaVM\* javaVM;

env->GetJavaVM(&javaVM);

On récupèrera ensuite, au moment souhaité, le pointeur d’environnement comme suit :

JNIEnv \*env;

javaVM->AttachCurrentThread(&env, NULL);

# A l’attention des prochains groupes :

## Informations :

Afin de build l’application à partir de votre ordinateur, il est nécessaire de modifier le fichier *PSTTango/TangoDepthMap/userPaths.mk* pour chaque utilisateur afin de spécifier le chemin absolu pointant vers votre dossier *PSTTango*.

Tous les modules locaux (ex : OpenCV) ont été placé dans le dossier *PSTTang/Modules/.* Il pourrait être intéressant de les retirer du projet Github afin de l’alléger.

## Conseils :

1) Essayez de comprendre les principes de base de la JNI (Java Native Interface) qui est la technologie permettant de faire communiquer des codes Java et C / C++

2) Réduire au maximum le nombre d’appel de Java vers C++ et encore plus de C++ vers Java, ces derniers rendent le code particulièrement incompréhensible et ne devraient être utilisés qu’en cas de nécessité

3) Utilisez l’API C++ de Tango plutôt que l’API Java, nous aurions préféré effectuer la totalité du projet avec Java mais les limitations d’interopérabilité entre la caméra OpenCV (*CvCameraViewListener2*) et la caméra Tango (*TangoUpdateCallback* ou *OnTangoUpdateListener* pour les codes plus anciens) nous on fait perdre plusieurs jours et sont a priori difficilement contournable.

4) Utilisez ou inspirez-vous des exemples proposés par Google (lien à la fin du dossier), ils couvrent la plupart des situations que nous avons rencontré et sont dans l’ensemble bien réalisés (bien que parfois inconsistants avec des approches différentes pour effectuer la même action d’un exemple à un autre).

# Liens utiles :

* API Tango en C++ : <https://developers.google.com/tango/apis/c/>
* Exemples officiels en C++ : https://github.com/googlesamples/tango-examples-c/