

# Contraintes des systèmes embarqués

## Application à un système vidéo (suite)

Romain Raveaux

### Sommaire :

But du TP2.....	1
Travail à faire.....	1
à Rendre :.....	7

### But du TP2

- 1°) Visualiser les images issues de la camera
- 2°) Passer de l'espace couleur YUV à l'espace couleur RGB en utilisant du code C++.
- 3°) Monitorer l'exécution en calculant le nombre de Frame Per Second
- 4°) Conclure sur l'intérêt du code natif pour des applications temps réel

### Travail à faire

1°) Installer cygwin sur votre disque 'c' : <http://cygwin.com/setup.exe> (Pas besoin sous linux)

2°) Installer NDK dans votre répertoire <cygwin>/home/ (Déjà installé dans votre VM. NDK est dans le repertoire /home/polytech/dev/android-ndk-r10c)  
<http://developer.android.com/sdk/ndk/index.html>

NDK est un compilateur C/C++ pour le système d'exploitation Android.

3°) Reprenez le projet du TP précédent.

4°) Créer un répertoire nommé 'jni' à la racine de votre projet.

5°) Créer un répertoire nommé 'libs' à la racine de votre projet.

6°) Pour NDK, créer un fichier de configuration s'appelant Application.mk dans votre répertoire « jni ». Ce fichier doit contenir les informations suivantes :

```
APP_STL := gnuSTL_static
APP_CPPFLAGS := -frtti -fexceptions
APP_ABI := armeabi-v7a
```

*APP\_ABI* correspond à l'architecture cible, arm ou x86. *armeabi* correspond à une architecture ARM bi-processeurs. *V7a* correspond au type d'architecture ARM, ici un cortex A7.

7°) Adapter le paramètre *APP\_ABI* suivant votre terminal.

8°) Pour NDK, créer un fichier de configuration s'appelant Android.mk dans votre répertoire « jni ». Ce fichier doit contenir les informations suivantes :

```
LOCAL_PATH := $(call my-dir)

include $(CLEAR_VARS)

LOCAL_MODULE := native_sample
LOCAL_SRC_FILES := jni_part.cpp
LOCAL_LDLIBS += -llog -ldl

include $(BUILD_SHARED_LIBRARY)
```

LOCAL\_MODULE : le nom de la librairie qui sera générée.

LOCAL\_SRC\_FILES : le nom des fichiers c/cpp à compiler.

LOCAL\_LDLIBS : Les dépendances.... Libraires pour les logs dans notre cas.

9°) Dans votre répertoire jni, créer un fichier cpp, jni\_part.cpp

10°) Voici la souche de votre fichier :

```
#include <jni.h>
#include <math.h>

using namespace std;

extern "C" {
    JNIEXPORT void JNICALL Java_epu_d2i_android_FrameProcessing_ProcessFast(JNIEnv* env, jobject thiz,
    jint width, jint height, jbyteArray yuv, jintArray bgra)
    {
        jbyte* _yuv = env->GetByteArrayElements(yuv, 0);
        jint* _bgra = env->GetIntArrayElements(bgra, 0);

        //stuf todo here

        env->ReleaseIntArrayElements(bgra, _bgra, 0);
        env->ReleaseByteArrayElements(yuv, _yuv, 0);
    }
}
```

Le nom de la fonction est très long, il correspond à une syntaxe précise :

JNIEXPORT void JNICALL Java\_nom des packages\_nom de la classe\_nom de la fonction

JNIEnv\* env :

Correspond à l'environnement JNI permettant certaines conversions de types JAVA vers C/C++

jobject thiz

Correspond à la classe qui a appelé la méthode.

jint width, jint height

Largeur et Hauteur de l'image

jbyteArray yuv

Le tableau de données issues de la caméra dans l'espace couleur YUV

jintArray bgra

Un tableau résultat contenant les valeurs converties Rouge, Vert et Bleu à partir de la représentation YUV.

Les appels suivants :

```
jbyte* _yuv = env->GetByteArrayElements(yuv, 0);
jint* _bgra = env->GetIntArrayElements(bgra, 0);
```

Ces appels servent à obtenir des pointeurs sur les objets issus du framework JAVA (dans le but de les modifier). A la fin il faut libérer les objets alloués :

```
env->ReleaseIntArrayElements(bgra, _bgra, 0);  
env->ReleaseByteArrayElements(yuv, _yuv, 0);
```

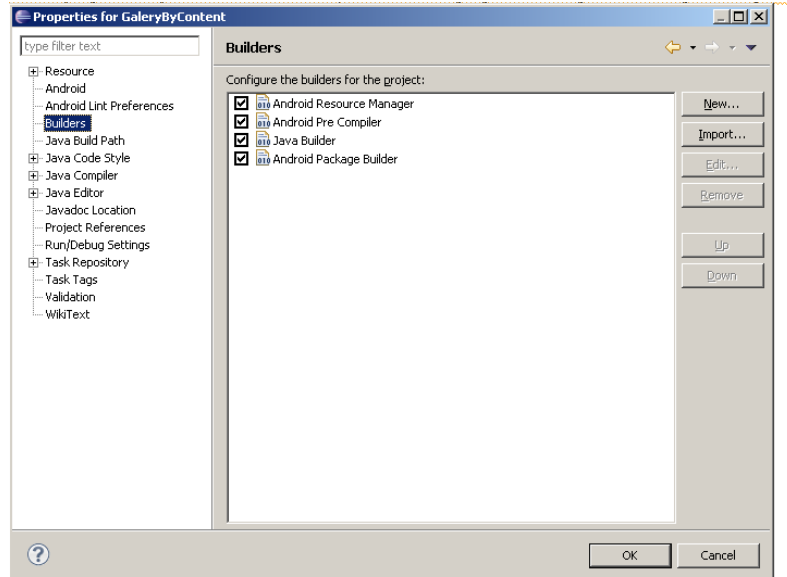
11°) Quelle est le nom de la méthode dans notre cas ?

12°) Quelle est le nom la classe possédant cette méthode ?

13°) Modifier le nom de la méthode JNI afin de correspondre à votre arborescence de projet.

14°) Pour compiler le code C/C++, créer un nouveau Builder pour votre projet.

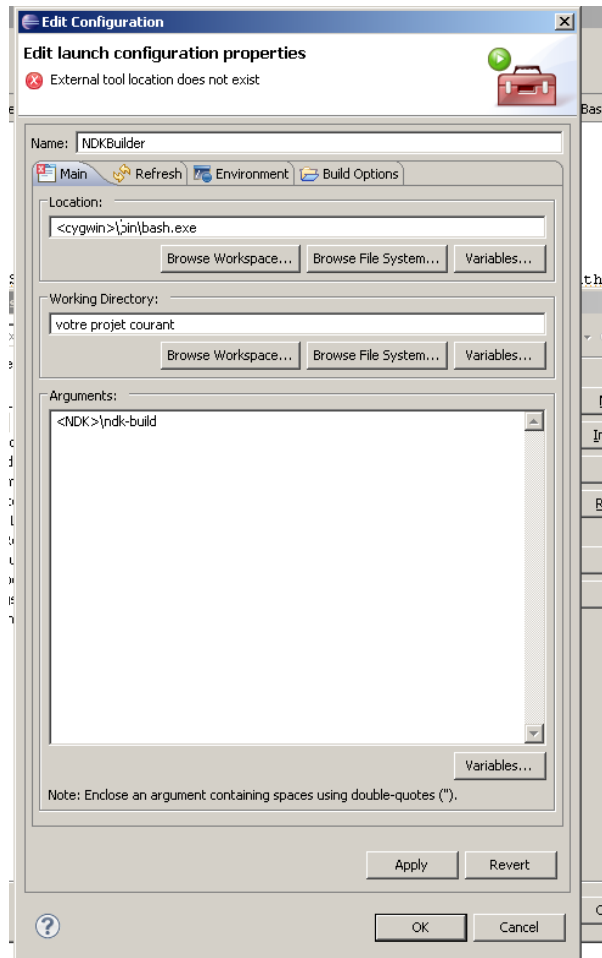
Clic droit sur votre projet. Ensuite Propriété du projet :



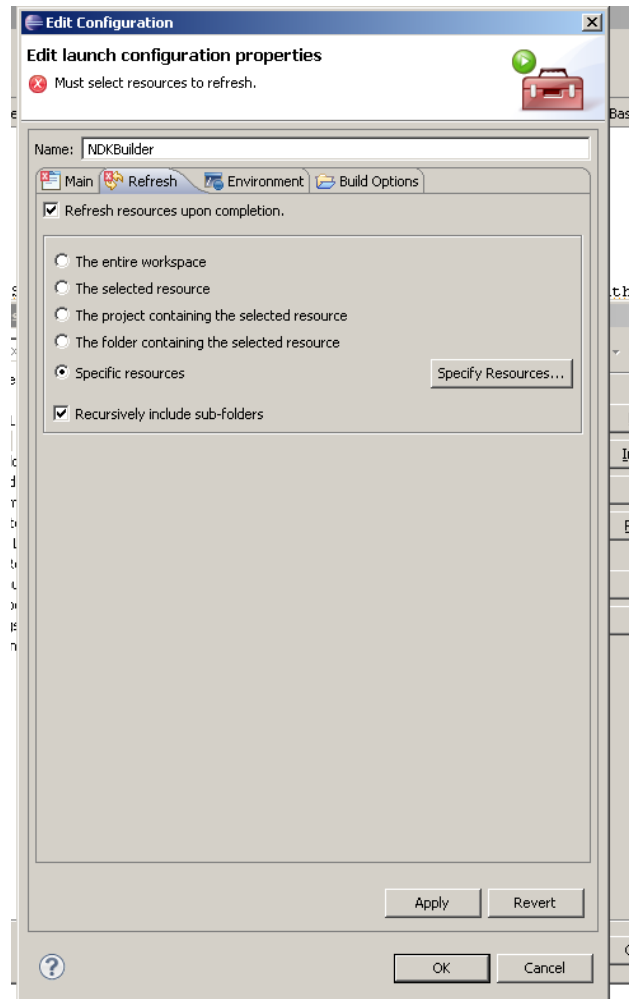
- New

- Configuration de type « program »

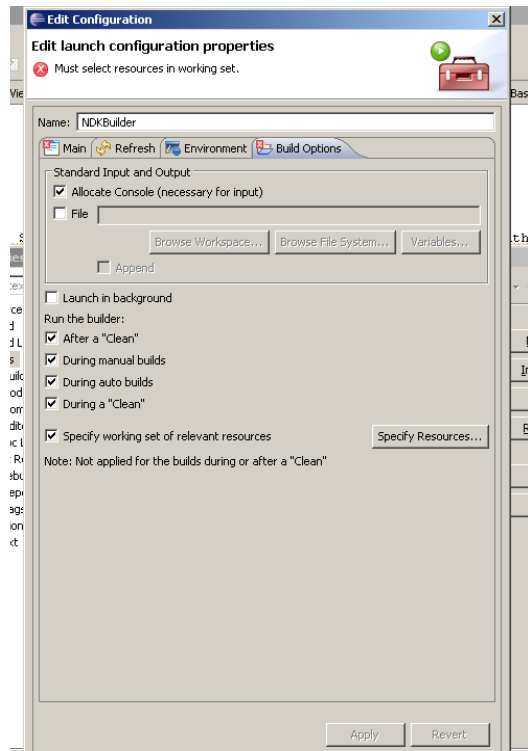
- Configurer votre Builder comme suit : (Sous Linux le champ Location est égal à <repertoire NDK>/NDKBuilder également le contenu champ argument doit être supprimé)



Ensuite passer à l'onglet Refresh :



- Cliquer sur *Specify Ressources* et choisissez le répertoire « libs ». Ensuite cliquez sur l'onglet *environment* : établissez une nouvelle variable : `PATH = <cygwin>\bin` (Pas besoin sous linux)
- Cliquer sur l'onglet *Build Option* :



- Cliquer sur les cases à cocher comme dans l'image ci-dessus.
- Cliquer sur Specify Ressources et choisissez le répertoire JNI.
- Voilà votre compilateur NDK est prêt.

15°) Dans votre classe FrameProcessing, placer le code suivant :

```
public native void ProcessFast(int width, int height, byte yuv[], int[] rgba);

static {
    System.loadLibrary("native_sample");
}
```

16°) Que fait le code ci-dessus ?

17°) Dans la fonction processFrame, commenter le code JAVA écrit précédemment (dans le TP précédent) et faites un appel à la fonction ProcessFast.

19°) Compiler et exécuter sur un émulateur.

20°) Que conclure sur le nombre de Frame par seconde ?

21°) ajouter le code C/C++ dans la méthode ProcessFast permettant de passer du modèle de représentation YUV au modèle RGB ? aidez vous du code JAVA écrit au TP précédent.

22°) Compiler et exécuter sur un émulateur.

23°) Que conclure sur le nombre de Frame par seconde ?

24°) Au lieu de calculer chaque valeur Y, U et V pour chaque pixel, implémenter une lookup table pour précalculer les valeurs Y, U, V.

25°) Que conclure sur le nombre de Frame par seconde ?

## **à Rendre :**

1°) Le code source commenté

2°) Un rapport décrivant le fonctionnement et le but de ce TP. Qu'avez-vous compris ? Avez-vous rencontré des problèmes ? Ecrivez les éléments qui permettraient à un de vos collègues de faire ce TP facilement. Que conclure sur l'apport du C/C++ dans les applications Android ?

3°) Le tout doit être déposé sur moodle pour la semaine suivant le TP. (ou si moodle était récalcitrant par email à votre enseignant)