

# PROGRAMMATION NDK

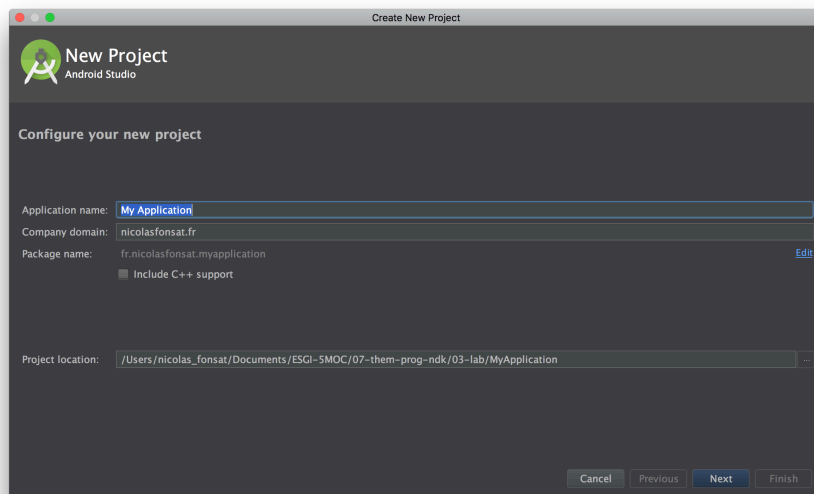
Remarque : tous les ateliers ont été réalisés avec Android Studio, parce qu'ils nous étaient impossible d'installer la machine Docker.

**Github :** <https://github.com/nFonsat/ESGI-thematique-Android-NDK>

## Atelier N°1

### Introduction

Pour ce TP, nous avons simplement créer un projet de base sur Android Studio et lié un code C++. « Include C++ support »



### Code C

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

extern "C"
{
    JNIEXPORT jstring JNICALL
    Java_com_example_akiro_hellofromc_MainActivity_stringFromJNI(JNIEnv *env, jobject /* this */) {
        std::string hello = "Hello from C++";
        return env->NewStringUTF(hello.c_str());
    }
}
```

Nous avons encadré le code par « extern « C » » pour que le code soit lisible par un compilateur C.

## Code JAVA

```
// Used to load the 'native-lib' library on application startup.
static {
    System.loadLibrary("native-lib");
}

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    final Handler handler = new Handler();

    TimerTask myTimerTask = () -> {
        handler.post(new Runnable() {
            @Override
            public void run() {
                TextView tv = (TextView) findViewById(R.id.sample_text);
                tv.setText(stringFromJNI());
            }
        });
    };

    Timer timer = new Timer();
    timer.schedule(myTimerTask, 2000);
}

/**
 * A native method that is implemented by the 'native-lib' native library,
 * which is packaged with this application.
 */
public native String stringFromJNI();
```

Nous avons choisi d'ajouter un Timer avant l'affichage du texte.

## Problème et difficulté

Pour ce projet, aucun problème particulier, à part l'installation du NDK.

## Atelier N°2

### Introduction

Pour ce TP, nous avons aussi utiliser AndroidStudio sans difficulté particulière. Le but de cet atelier est de résoudre le problème des Oods. Il consiste à calculer les parts équitables à fournir pour acheter un cadeau en fonction des budgets de chacun.

### Code C

```
#include <jni.h>
#include <string>
#include <stdlib.h>

extern "C"
{
    int cmpfunc (const void * a, const void * b)
    {
        return ( *(int*)a - *(int*)b );
    }
}

JNIEXPORT jintArray JNICALL
Java_fr_nicolasfonsat_oodsphere_MainActivity_solution(
    JNIEnv *env, jobject,
    jint nbParticipant, // Le nombre de participant
    jint gift, // Le prix du cadeaux
    jintArray budgets /*// Un tableau des budget de chaque participant */ ) {

    jint * budgetAr = env->GetIntArrayElements(budgets, NULL); //jintArray -> jint *
    jintArray result = env->NewIntArray(nbParticipant);
    jint * resultAr = env->GetIntArrayElements(result, NULL); //jintArray -> jint *

    // On trie le tableau par le plus petit budget
    qsort(budgetAr, (size_t) nbParticipant, sizeof(int), cmpfunc);

    //Vérification que le budget max soit suffisant
    int sumMax = 0;
    for (int i = 0; i < nbParticipant; ++i) {
        sumMax += budgetAr[i];
    }

    if (sumMax < gift) {
        return result;
    }
}
```

```
int nbWithoutPay = nbParticipant; // Participant qui n'ont pas encore payer
int due = gift; // Reste à payer
for(int i = 0; i < nbParticipant; ++i){

    // Si le budget du participant est inférieur au budget équitable par participant
    // alors on prend tout son budget
    // sinon on prend une somme équitable avec le reste des participants
    // Cas particulier: S'il reste un participant alors on prend la somme qui manque..
    if(budgetAr[i] < due/nbWithoutPay){
        jint toPay = budgetAr[i];
        resultAr[i] = toPay;
        due -= toPay;
    } else {
        if ( nbWithoutPay == 1) {
            jint toPay = due;
            resultAr[i] = toPay;
            due -= toPay;
        }
        else {
            jint toPay = due/nbWithoutPay - (due % nbWithoutPay);
            resultAr[i] = toPay;
            due -= toPay;
        }
    }
    nbWithoutPay--;
}

env->ReleaseIntArrayElements(result, resultAr, NULL); //jint * -> jintArray
return result;
}
```

Voici le code détaillé pour le calcul des parts équitables. Nous avons documenté le code afin d'expliquer, notre algorithme de calcul. Nous avons testé sur plusieurs jeux de données et les résultats semblent cohérents.

### Problème et difficulté

La plus grosse difficulté fut de comprendre l'algorithme et de pouvoir trouver les bons jeux de données.

## Atelier N°3

### Introduction

Pour cet atelier, nous allons mesurer les performances de calcul de l'algorithme de fibonacci.

- En code C natif
- En code Java
- En callback C depuis JAVA

### Code fibonacci

```
int fibonacci (int n) {  
    if (n<=1) return 1;  
    return (fibonacci(n-1))+(fibonacci(n-2));  
}
```

### Code C natif

```
JNIEXPORT jint JNICALL  
Java_fr_nicolasfonsat_helloTestndk_MainActivity_fibonacciC(JNIEnv *env, jobject, jint n) {  
    return fibonacci(n);  
}
```

### Callback JAVA

```
Integer nb = Integer.parseInt(text.getText().toString());  
  
25 Platform >] java.lang  
s String extends Object  
alizable, Comparable<String>, CharSequence  
  
String timeC = String.format(Locale.getDefault(), "Native speed : %d ms", t1End.getTime() - t1Begin.getTime());  
tv_fibonacci_native.setText(timeC);  
Log.d("XXXXXX", timeC);  
  
Date t2Begin = new Date();  
fibonacciJava(nb);  
Date t2End = new Date();  
  
String timeJAVA = String.format(Locale.getDefault(), "Java speed : %d ms", (t2End.getTime() - t2Begin.getTime()));  
tv_fibonacci_java.setText(timeJAVA);  
Log.d("XXXXXX", timeJAVA);  
  
Date t3Begin = new Date();  
fibonacciCallback(nb);  
Date t3End = new Date();  
  
String timeCallback = String.format(Locale.getDefault(), "C/callback speed : %d ms", (t3End.getTime() - t3Begin.getTime()));  
tv_fibonacci_callback.setText(timeCallback);  
Log.d("XXXXXX", timeCallback);
```

Lors d'un appuie en JAVA, il est possible de récupérer tous les résultats.

### Problème et difficulté

Pas de difficulté particulière pour cet atelier.

## Atelier N°4

### Introduction

Pour cet atelier, nous allons réaliser trois petits projets sous SDL. Le premier consiste à la création d'un Cercle sous SDL, puis, un carré suiveur et la réalisation d'un damier.

### Code Cercle

```
while (1) {
    // Set render color to red ( background will be rendered in this color )
    SDL_SetRenderDrawColor( renderer, 0, 0, 0, 255);

    // Clear window
    SDL_RenderClear( renderer );

    int width;
    int height;
    int radius = 100;
    SDL_GetWindowSize(window, &width, &height);

    SDL_SetRenderDrawColor(renderer, 0, 0, 255, 255);
    for (int w = 0; w < radius * 2; w++)
    {
        for (int h = 0; h < radius * 2; h++)
        {
            int dx = radius - w; // horizontal offset
            int dy = radius - h; // vertical offset
            if ((dx*dx + dy*dy) <= (radius * radius))
            {
                SDL_RenderDrawPoint(renderer, width/2 + dx, height/2 + dy);
            }
        }
    }

    // Render the rect to the screen
    SDL_RenderPresent(renderer);
}
```

Voici le code C qui permet de créer, un cercle Bleu sur fond noir remplis.

## Code Mouse Tracker

Une première version avec l'utilisation de GetMouseState

```
// Set render color to red ( background will be rendered in this color )
SDL_SetRenderDrawColor( renderer, 0, 0, 0, 255);

// Clear window
SDL_RenderClear( renderer );

SDL_GetMouseState(          // Sets mouse_position to...
    &mouse_position.x,      // ...mouse arrow coords on window
    &mouse_position.y
);

SDL_Log("Mouse position: x=%i y=%i", // Print mouse position
    mouse_position.x, mouse_position.y
);

SDL_SetRenderDrawColor( renderer, 0, 0, 255, 255);

SDL_Rect rect;
rect.x = mouse_position.x;
rect.y = mouse_position.y;
rect.w = 100;
rect.h = 100;

SDL_RenderFillRect(renderer, &rect);

// Render the rect to the screen
SDL_RenderPresent(renderer);
```

Une version avec PollEvent

```
// Set render color to red ( background will be rendered in this color )
SDL_SetRenderDrawColor( renderer, 0, 0, 0, 255);

// Clear window
SDL_RenderClear( renderer );

SDL_Event event;
if (SDL_PollEvent(&event)) {
    switch (event.type) {
        case SDL_QUIT:
        case SDL_KEYDOWN:
        case SDL_FINGERDOWN:
        case SDL_FINGERMOTION:
            SDL_SetRenderDrawColor( renderer, 0, 0, 255, 255);

            SDL_Rect rect;
            rect.x = (int)event.tfinger.x;
            rect.y = (int)event.tfinger.y;
            rect.w = 100;
            rect.h = 100;

            SDL_RenderFillRect(renderer, &rect);
            break;
    }
}

// Render the rect to the screen
SDL_RenderPresent(renderer);
```

### Problème et difficulté

Installation de SDL avec Android Studio.

Problème de Cross compilation, l'application générée n'est pas utilisable par tous les téléphones.