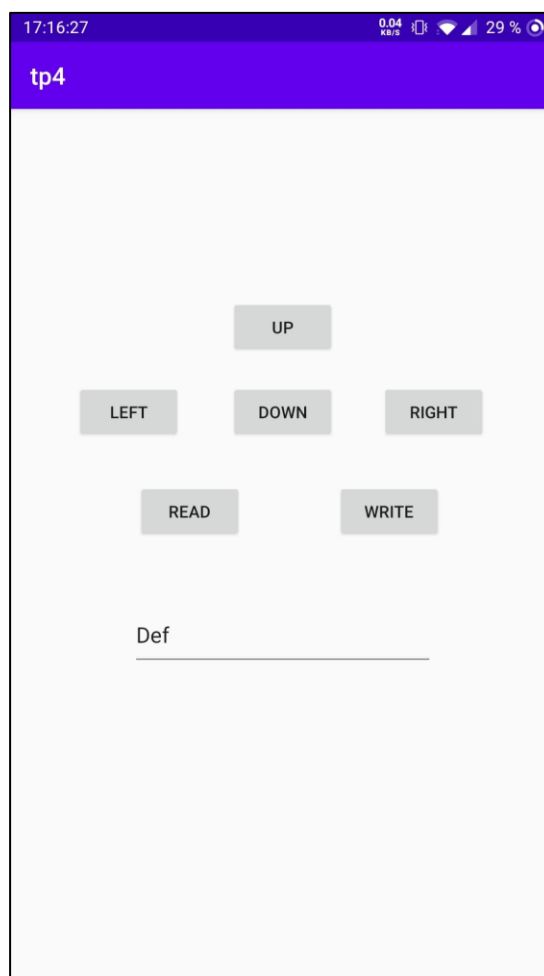


TP4 : Utilisation de JNI

GUILLEMAILLE Timothée - MONTANGE Aimeric

Tout d'abord nous avons démarré Android Studio et créé une application C++ native en Java. Une fois cela fait l'IDE nous demandais d'installer le NDK Android nécessaire pour travailler sur ce type d'application. Car en effet celui-ci permet de développer directement dans le langage cible ici le C++ avec quelques fonctions.

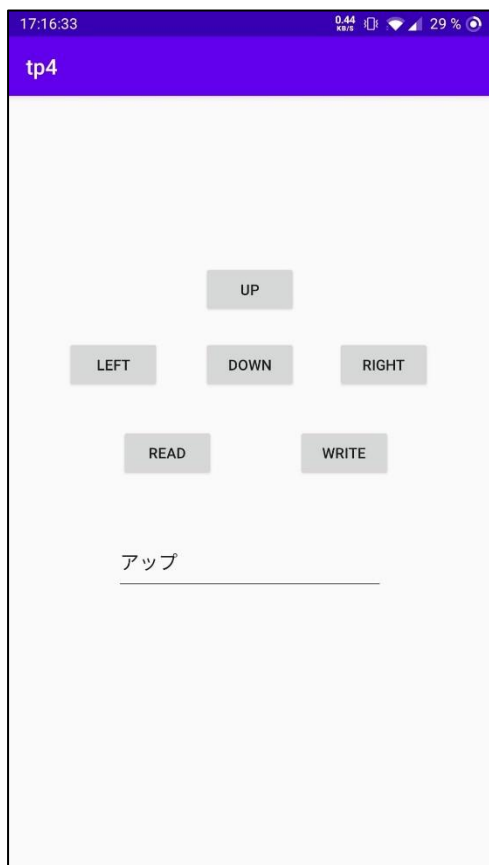
Premièrement nous avons créé les objets (boutons, textView) de l'application. 6 boutons et un PlainText est alors ajouté, la valeur du PlainText est mis par défaut sur Def :



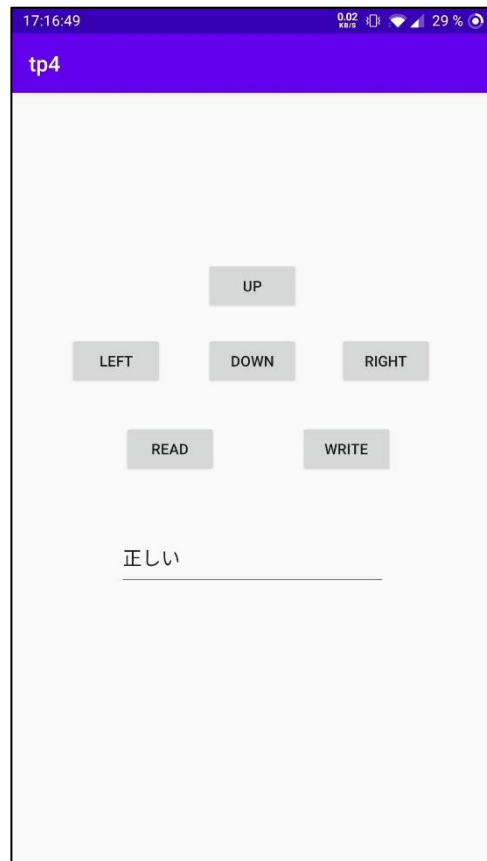
Nous avons ensuite implémenté les trois fonctions événements qui vont se déclencher lors de l'appui d'un des 6 boutons : OnClick, OnRead et OnWrite.

La première fonction Java permet de récupérer la valeur du texte présent sur le bouton appuyé pour ensuite envoyer cette information à la fonction native C++. Cette fonction permet avec cette information l'affichage du bon résultat dans la PlainText en fonction du bouton. Deux exemples :

Click UP :

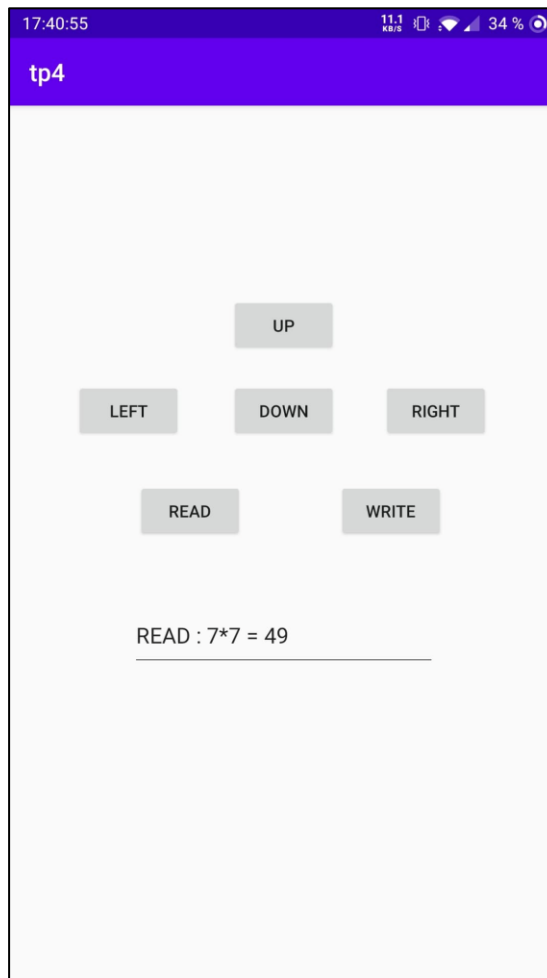


Click RIGHT :



La fonction native DirectionFunction récupère le texte du bouton et en fonction de celui-là, envoie la traduction japonaise correspondante.

La deuxième fonction Java, OnRead est appelée lors de l'appui sur le bouton Read, elle créera un nombre aléatoire entre 0 et 10 et affichera dans le PlainText le résultat de la fonction native C++ ReadFunction. Cette fonction retournera le carré du nombre avec un formalisme d'affichage.



La troisième fonction Java, OnWrite est appelée lors de l'appui sur le bouton Write, elle créera un nombre aléatoire entre 0 et 10 et affichera dans le PlainText le résultat de la fonction native C++ WriteFunction. Cette fonction retournera le cube du nombre avec un formalisme d'affichage.

