Annexe 2– Importance d’un buffer / méthode du boomerang

Dans l’annexe 1B, on a pu pratiquer la lecture et l’écriture dans des fichiers texte sauvegardés en mémoire interne; nous avons utilisé un Buffer pour accélérer le processus mais pour quelques strings, le gain en temps est minime.

On va s’attaquer à lire un fichier volumineux ( un mp3 ) et voir la différence de vitesse en le lisant, octet par octet, avec ou sans buffer.

On devra lire le fichier se situant dans le **stockage partagé** de l’appareil, soit dans les dossiers Music, Images ou Téléchargements.

Utiliser le projet LireFichierMp3 présent dans LEA

But du projet : lire un fichier présent dans le dossier Musique pour prouver l’importance du Buffer.

1. Créer un nouveau projet Kotlin. Placer le fichier mp3 présent sur LÉA ( grand fichier ) dans le dossier storage/self/primary/Music de l’émulateur / de votre téléphone à l’aide du Device Explorer

(POSSIBLE DE DEVOIR FERMER L'ÉMULATEUR APRÈS AVOIR AJOUTER LE .MP3 ET LE REDÉMARRER )

Intégrer le fichier kt et le fichier xml de positionnement à votre projet

1. Regarder le [processus de permission](https://developer.android.com/training/data-storage/shared/media#request-permissions):

Scoped Storage / stockage étendu

* Sous le principe du stockage étendu ( scoped storage ) , plus besoin de demander des permissions dans les dossiers permis ( download, music… )
* On utilise le stockage étendu par opposition au stockage interne, visible que par l’app qu’on développe et que nous avons utilisé dans les annexes précédentes

1. Gestion de l’événement
2. Processus startActivityForResult, pourquoi ?

Prendre des notes :

1. Quel est le boomerang ?
2. Codez la méthode qui va lire un à un les octets du fichier mp3 et calculer le temps que ça prend avec ou sans tampon.

Quels sont les bons flux de données à utiliser ?Cliquez ou appuyez ici pour entrer du texte.

Comment récupérer l’heure système ?

Temps de lecture du fichier ( sans Buffer ) :

Temps de lecture du fichier ( avec Buffer ) :