

Rapport Projet Android

Lou LE GALL - Rémi UHARTEGARAY

IAI3

Sommaire

Sommaire	2
Introduction	3
Notre réalisation	4
Figure 1 : Page d'accueil	4
Figure 2 : Page initialisation Bluetooth	5
Figure 3 : Appareillage et connexion Bluetooth	5
Figure 4 : Popup de création	6
Figure 5 : Création nouveau bouton raccourci	6
Figure 6 : Page d'accueil avec un nouveau bouton	7
Figure 7 : Bouton "Supprimer"	7
Figure 8 : Alerte de suppression	7
Figure 9 : Stream decks par défaut	8
Les difficultés	9
Conclusion	10

Introduction

Après des cours magistraux et des travaux pratiques, nous avons initié un projet Android visant à développer une application Android/Kotlin. L'objectif principal est de créer une application qui utilise la technologie Bluetooth pour appareiller, configurer et simuler un périphérique d'interface humaine (HID).

L'interface utilisateur sera conçue pour permettre la découverte, l'appairage et la connexion via Bluetooth avec un hôte HID, tel qu'un autre appareil Android ou même un Mac. De plus, l'application devra offrir la possibilité de définir et de conserver plusieurs profils HID, avec une interface permettant de définir de nouvelles touches et raccourcis grâce à une fonctionnalité de liaison de touches (key binding).

Pour mener à bien ce projet, nous prévoyons d'utiliser la documentation Bluetooth d'Android Developer, ainsi que les fonctionnalités fournies par Android Room pour le stockage local et les coroutines Kotlin pour gérer les opérations asynchrones.

Cette approche nous permettra de créer une interface HID sur notre téléphone, nous permettant de commander à distance un autre appareil via la liaison Bluetooth.

Pour concrétiser ce projet, Lou Le Gall et Rémi Uhartegaray somment un binôme. Nous collaborons en utilisant un seul téléphone Android, ce qui signifie que nous ne pouvons pas travailler simultanément sur le projet. De plus, le lien Bluetooth fonctionne uniquement avec un de nos deux ordinateurs.

Notre réalisation

Examinons de manière approfondie les différentes étapes de notre projet et les fonctionnalités que nous avons implémentées.

Tout d'abord, nous avons choisi de travailler avec Android Studio sur deux plateformes distinctes, à savoir un Mac et un PC Linux. Pour tester et compiler notre code, nous avons utilisé à la fois l'émulateur Android Studio et un véritable téléphone Android, garantissant ainsi une vérification exhaustive de la performance de notre application dans des environnements variés.

La première phase de notre projet a été marquée par la manipulation de la base Bluetooth. Nous avons commencé par essayer de réaliser par nous-mêmes la partie d'appareillage et de connexion Bluetooth, en vain. C'est pour cette raison que nous sommes repartis de zéro et utilisé comme base le code "bluetoothsample" fournis.

Nous avons étendu cette base de code en intégrant une HomePage, qui constitue l'interface principale de notre application. Cette page d'accueil présente un bouton Bluetooth, servant de point d'accès à la fonction de connexion Bluetooth.

Sur notre page d'accueil, nous retrouvons un bouton "Bluetooth" :

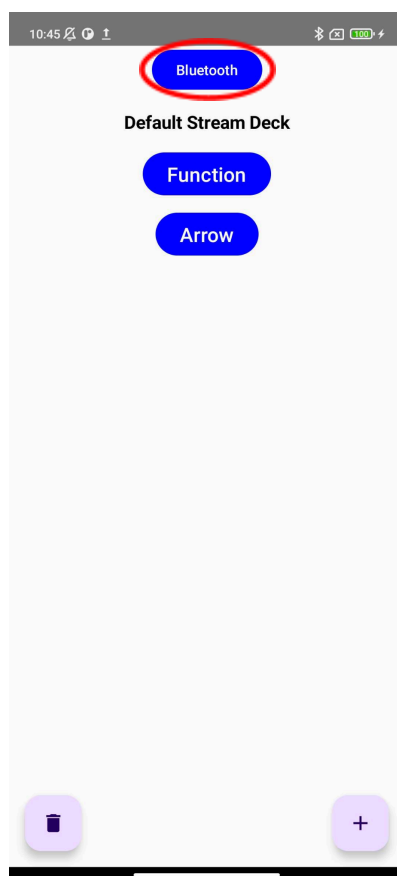


Figure 1 : Page d'accueil

Cela nous permet d'arriver sur la page suivante :

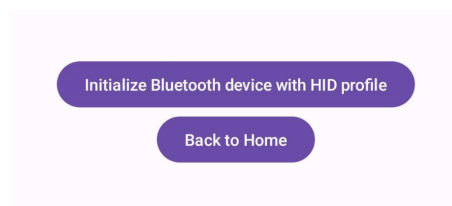


Figure 2 : Page initialisation Bluetooth

Une fois sur cette page, nous avons la possibilité d'activer le Bluetooth ou de retourner à la page d'accueil. Si l'intention est de poursuivre l'initialisation du Bluetooth, il suffit de cliquer sur "Initialiser le périphérique Bluetooth avec le profil HID".

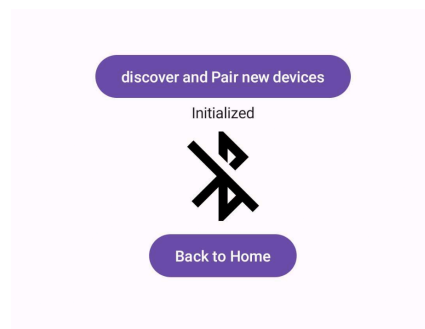


Figure 3 : Appareillage et connexion Bluetooth

Sur cette page, il est possible de détecter un nouvel appareil, ce qui indique que nous ne sommes pas encore connectés à aucun autre appareil en Bluetooth. Ensuite, nous remarquons un bouton intitulé "Connect to host". Initialement, ce bouton remplace le texte "Initialized".

Ce problème est survenu lors de nos tests, et pour une raison inconnue, le bouton a disparu. Nous n'avons pas réussi à résoudre ce problème, cela même avec le téléphone de notre professeur. Ce bouton est destiné à être utilisé pour se connecter à un appareil auquel nous nous sommes déjà connectés précédemment.

Lorsqu'une connexion est établie avec succès, le logo Bluetooth barré devient bleu, et le bouton "Connect to host" disparaît.

Passons maintenant à la gestion du Stream Deck. Sur notre page d'accueil, nous avons ajouté un bouton en bas à droite. Une fonctionnalité clé que nous avons intégrée est le bouton "Ajout" sur la page d'accueil. Ce bouton déclenche l'affichage d'une fenêtre contextuelle, permettant à l'utilisateur d'ajouter au "stream deck" un bouton associé à un raccourci. Dans cette fenêtre, l'utilisateur peut saisir le nom du shortcut, choisir une commande (shortcut) dans une liste déroulante, et sélectionner une icône associée.

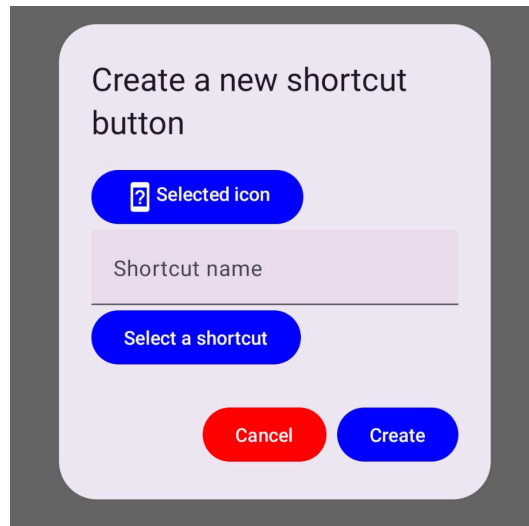


Figure 4 : Popup de création

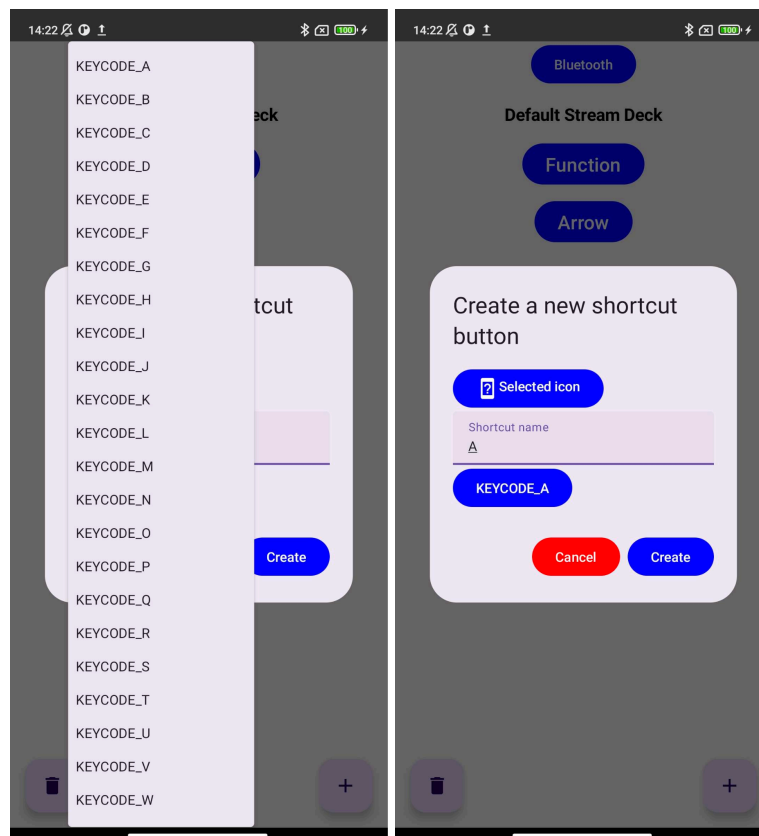


Figure 5 : Création nouveau bouton raccourci

Lorsque l'on valide les informations de la popup, on peut cliquer sur "Créer". Ce qui nous permet d'afficher le nouveau bouton sur la page d'accueil. L'intérêt de cette fonctionnalité réside dans la création dynamique de boutons sur la HomePage. Chaque bouton nouvellement créé représente un shortcut personnalisé qui peut être utilisé pour envoyer des commandes spécifiques à un ordinateur distant.

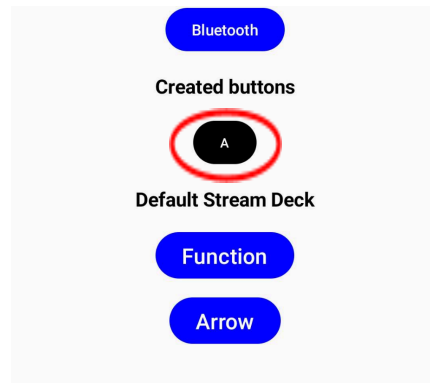


Figure 6 : Page d'accueil avec un nouveau bouton

Si un bouton de raccourci créé ne vous satisfait pas, vous avez la possibilité de le supprimer en utilisant le bouton "Supprimer". Lors de la suppression, un message "Toast" s'affiche pour confirmer la suppression du bouton ainsi que le nom du bouton supprimé.

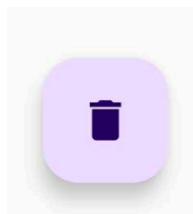


Figure 7 : Bouton "Supprimer"

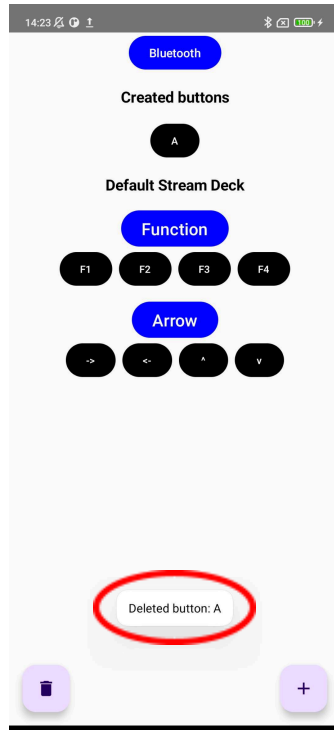


Figure 8 : Alerte de suppression

Pour finir, à la création de notre application, nous avons créé deux “stream deck” par défaut qui contiennent plusieurs boutons shortcut.

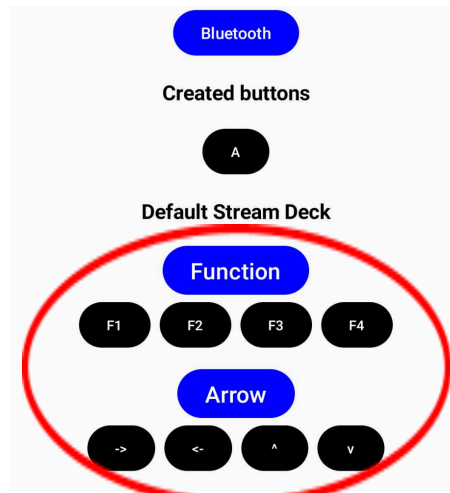


Figure 9 : Stream decks par défaut

Ensuite, afin de garantir la persistance des variables locales même lorsque l'application est fermée, nous avons judicieusement intégré la bibliothèque Room à notre projet. Room offre une solution efficace pour la gestion du stockage local, permettant à notre application de conserver les données nécessaires entre les sessions.

Dans notre projet, Room est utilisé pour sauvegarder les boutons shortcut que nous créons à partir du bouton “Ajout”.

Les difficultés

Nous avons rencontré plusieurs problèmes tout au long de ce projet. Tout d'abord, la limitation de puissance de nos ordinateurs s'est avérée être un obstacle majeur. Au sein de notre binôme, nous disposions d'un seul PC capable de faire tourner l'émulateur d'Android Studio. De plus, nous ne possédions pas de téléphone Android pour tester notre code directement sur un appareil réel (car non fournis par l'école), ce qui est essentiel pour évaluer la fonctionnalité Bluetooth, une composante cruciale du projet.

En outre, après avoir perdu trois séances à tenter de créer un code fonctionnel pour la communication Bluetooth, nous avons finalement reçu un code préexistant pour cette partie du projet. N'ayant pas réussi à produire un code opérationnel précédemment, nous avons dû recommencer à zéro avec ce nouveau code. Nous avons ainsi perdu trois séances à développer un code qui ne fonctionnait pas correctement, suivies d'une séance dédiée à la mise en œuvre du code sur un téléphone Android, aimablement prêté par un camarade, avec un ordinateur Mac.

Après ces quatre séances, nous avons finalement pu débiter le projet. Cependant, de nouveaux problèmes liés au Bluetooth ont émergé lors de nos heures de travail en dehors des séances prévues. En effet, notre téléphone Android refusait de se connecter en Bluetooth au PC sous Linux, puis refusait également la connexion avec le PC sous Mac. Pour une raison inconnue et soudainement (sans avoir touché au code spécifique), nous nous sommes retrouvés sans Bluetooth.

La dernière difficulté rencontrée a été la contrainte de respecter les consignes dans le délai imparti. En raison des problèmes évoqués précédemment, de l'ampleur du projet ainsi que du manque de connaissances, les nombreux bugs récurrents liés au Bluetooth ont entravé notre progression dans le projet.

Conclusion

Pour conclure, notre projet permet la création d'un "stream deck" associé à des raccourcis. Ce dernier comprend des sous-decks par défaut, auxquels on peut ajouter de nouveaux raccourcis pour personnaliser davantage notre "stream deck". Les boutons que nous créons sont stockés localement grâce à Room, et nous avons la possibilité de nous connecter en Bluetooth à un appareil pour utiliser notre "stream deck".

Ainsi, nous avons réussi à respecter les principaux objectifs du projet malgré certaines difficultés et de nombreuses heures de travail.

Cette expérience nous a confrontés à un défi majeur, celui de maîtriser le langage Kotlin pour la première fois. L'intégration de ses concepts s'est avérée être un véritable défi. Malgré les séances de travaux pratiques, nous avons rencontré des difficultés dans l'apprentissage du langage et de son architecture.