DAA - Laboratoire 02

Développement d'Applications Android

Martins Alexis, Saez Pablo

HEIG VD

29 octobre 2023

Table des matières

1. Les activité

Que se passe-t-il si l'utilisateur appuie sur « back » lorsqu'il se trouve sur la seconde *Activité* ?

On y voit simplement `Bienvenue <espace>` sans aucun prénom. La raison est que dans notre implémentation, on vérifie si un nom est entrée dans la seconde activité, s'il n'y en a pas, on renvoie simplement rien (`<espace>`). Cette vérification est nécessaire sinon l'application cash.

Au niveau du cycle de vie des activités, l'activité contenant le formulaire est mise en pause. L'activité principale repasse en phase `Start`, puis `Resume` et finalement l'activité avec le formulaire est détruite.

Voici le diagramme pour le premier cas :

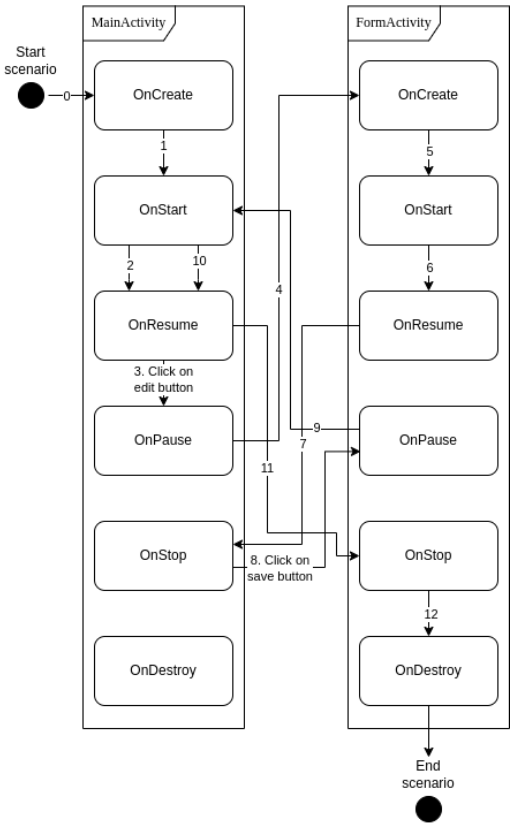

Figure 1: Scénario 1

Voici le diagramme pour le second cas :

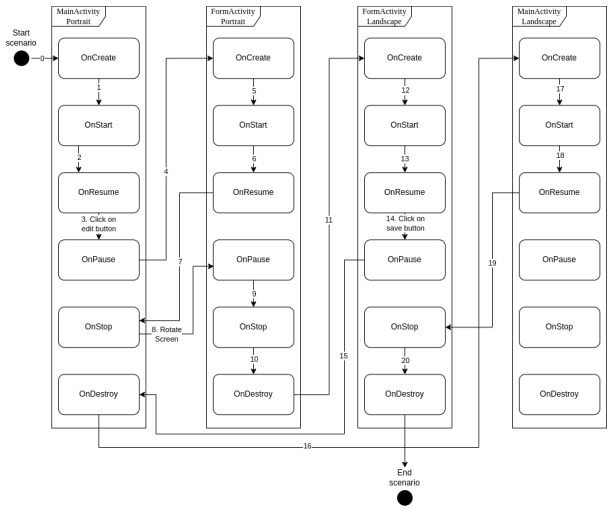

Figure 2: Scénario 2

Que faut-il mettre en place pour que vos *Activités* supportent la rotation de l'écran ? Est-ce nécessaire de le réaliser pour les deux *Activités*, quelle est la différence ?

Dans l'activité principale, on doit overrride les fonctions `onSaveInstanceState` et `onRestoreInstanceState` afin d'éviter que le texte avec le nom soit écraser. Cela vient du fait que l'activité, lorsque l'on change son orientation, est écrasée et recrée comme on a pu le voir dans le diagramme d'activité.

2. Les Fragments, premiers pas

Les deux *Fragments* fournis implémentent la restauration de leur état. Si on enlève la sauvegarde de l'état sur le *ColorFragment* sa couleur sera tout de même restaurée, comment pouvons-nous expliquer cela ?

On sauvegarde directement les couleurs dans la `SeekBar` et lors de la restauration de l'état des barres de progressions, on fait appels à `setOnSeekBarChangeListener` pour chacune des barres. Cela fonctionne avec la plupart des widgets fournis par Android.

Si nous plaçons deux fois le *CounterFragment* dans l'*Activité*, nous aurons deux instances indépendantes de celui-ci. Comment est-ce que la restauration de l'état se passe en cas de rotation de l'écran ?

L'état sauvegardé dans le `bundle` du mode portait sera réutilisé dans le mode paysage puisque dans les deux layouts l'identifiant `fragmentContainerView` sera identique.

3. Le FragmentManager

A l'initialisation de l'*Activité*, comment peut-on faire en sorte que la première étape s'affiche automatiquement ?

On utilise `backStackEntryCount` dans la fonction `onCreate` et si la valeur de la stack retourné est 0, on ajoute un nouveau *fragment*.

Comment pouvez-vous faire en sorte que votre implémentation supporte la rotation de l'écran ? Nous nous intéressons en particulier au maintien de l'état de la pile de *Fragments* et de l'étape en cours lors de la rotation.

L'état de la pile est géré par le `FragmentManager` directement et la valeur de l'étape courante est dans le bundle. Le texte compte à lui est rechargé dans la fonction `OnViewCreated`.

Dans une transaction sur le *Fragment*, quelle est la différence entre les méthodes *add* et *replace* ?

La principale différence réside dans le comportement de la méthode `add`, qui empile les fragments, tandis que la méthode `replace` remplace le fragment existant par le nouveau. La première va laisser les deux fragments visibles à l'écran et la seconde va retirer le fragment courant de l'affichage pour y mettre le nouveau.