

Programmation des Applications Mobiles avec Android
TP/Td n°2 : Composants graphiques de base - Gestion des événements

Exercice 1:

Créer une application Android, placer deux boutons (*Button* avec comme labels respectif “Pousse-Moi” et “Clique Moi”) et une zone de texte (un *TextView* ayant come texte “Ceci est un Test”) sur la fenêtre. En utilisant l'approche basée sur XML, vous pouvez spécifier le label des boutons ainsi que le contenu de la zone de texte dans le fichier *activity_main.xml* (vous n'avez pas encore besoin d'utiliser *strings.xml*). Pour cela, utiliser la propriété *android:text*. Veuillez cependant noter qu'il est plus facile d'utiliser l'éditeur visuel dans un premier temps, puis d'éditer ensuite le fichier *activity_main.xml*. Naturellement, rien ne se produit lorsque vous appuyez sur les boutons. Tester dans un premier temps votre application sur l'émulateur puis dans un second temps, si vous avez un téléphone Android ou une tablette, tester y votre application. N'oubliez pas de mettre votre téléphone en mode "Débogage USB".

Exercice 2:

En reprenant l'exercice précédent, modifier le code de telle sorte que les labels des boutons et de la zone de texte puissent être spécifiés à l'aide du fichier *strings.xml*. Ajouter également une image de fond à la fenêtre.

Exercice 3:

Dans cet exercice, nous allons donner à vos boutons certains comportements aux cliques. Voici quelques options sur le comportement des boutons :

1. Faire apparaître une fenêtre popup *Toast* lorsqu'on appuie sur chacun des boutons avec le texte de votre choix.
2. Changer la couleur du label du bouton appuyé. Pour cela choisissez au hasard parmi les couleurs *Color.RED*, *Color.BLUE*, *Color.YELLOW*, etc. Pour changer la couleur utiliser la méthode *setTextColor* sur le bouton qui a été passé au gestionnaire d'événement (*event handler*). Pour générer un nombre aléatoire penser à utiliser la classe java *Random*.
3. Après la clique sur chacun des boutons, changer le label du bouton pour un texte du genre “*J'ai été cliqué n fois*” où *n* représente le nombre de fois où l'on a cliqué sur chacun des boutons.

Exercice 4:

Créer une nouvelle application similaire au premier, mais cette fois-ci, lorsque vous appuierez sur un bouton, vous devrez changer avec une couleur choisie au hasard celle de la zone de texte (*TextView*) au lieu de la couleur du label du bouton qui a été appuyé.

Exercice 5:

En utilisant une approche uniquement basée sur l'utilisation de la classe activité principale, créer une application avec deux boutons. Le premier doit faire apparaître un popup (*Toast*) ou bien insérer un texte dans un *TextView* qui affichera “*Hello*”. Le second bouton doit par contre afficher le texte “*Au revoir*”.

Exercice 6:

1. Créer une ligne contenant deux *ImageButtons*, puis placer un *TextView* vide en bas de cette dernière que nous allons utiliser dans toutes les questions de cet exercice. Pour le premier bouton, utiliser une image

rouge qui permettra de mettre la couleur de fond du *TextView* en rouge lorsqu'on appuie dessus. Pour le premier bouton, utiliser une image bleue qui permettra de mettre le texte du *TextView* en bleu lorsqu'on appuie dessus. Pour trouver une image bleue ou rouge vous pouvez aller sur la page <http://findicons.com/search/red> ou <http://findicons.com/search/blue>. Pour changer les couleurs, utiliser la méthode *setBackgroundColor* de la classe *TextView* ayant pour argument *Color.RED* ou *Color.BLUE*.

2. Ajouter une ligne avec trois *RadioButtons* ayant comme labels Rouge, Jaune et Bleu qui permettront selon le choix de l'utilisateur de mettre le *TextView* à la couleur appropriée.
3. Ajouter une ligne avec trois *ToggleButtons* ayant respectivement pour labels en état allumé et éteint Rouge/Non Rouge, Jaune/Non Jaune, et Bleu/Non Bleu qui permettront d'appliquer au *TextView* la couleur appropriée lorsqu'on clique sur chacun d'eux. Pour le cas "Non Couleur", utiliser la couleur noire.
4. Ajouter une nouvelle ligne avec des *RadioButtons* ayant les mêmes labels qu'à la question n°2. Seulement cette fois-ci il ne doit rien se passer lorsque le choix est effectué sur un des boutons radion. Au lieu de cela, placez un bouton en dessous ayant pour label "Appliquer la couleur choisie ci-dessus". Changer la couleur quand ce bouton est cliqué. Astuce: *RadioGroup* a une méthode *getCheckedRadioButtonId*. Notez que ce n'est pas l'indice (0 pour le premier bouton radio, 1 pour le deuxième, etc.) Au lieu de cela, il s'agit d'un entier qui représente l'id (i.e, l'identifiant tel que défini par android: id dans le fichier XML). Donc, vous avez à faire un "switch" ou "if" comparant les identifiants (*R.id.red_radio_button*, *R.id.blue_radio_button*, etc.).

Exercice 7:

1. Créer un projet avec un *Spinner* qui affiche les choix Rouge, Jaune, Bleu, et Vert. Ajouter ensuite un *TextView* dont la couleur de fond correspondra à celle choisie sur le *Spinner*. Spécifier les différents choix avec XML.). Penser à utiliser *setOnItemSelectedListener*.
2. Ajouter un second *Spinner* avec les mêmes choix et comportement. Mais, cette fois-ci, définissez les choix dans la classe java idoine. Si vous voulez ajouter une invite (par exemple, titre en haut, lorsque le *Spinner* apparaît, utilisez la méthode *Spinner.setPrompt* en lui fournissant soit un identifiant ou une chaîne). Penser à utiliser un *ArrayAdapter* avec éventuellement la liste des couleurs et la méthode *setDropDownViewResource* (*android.R.layout.simple_spinner_dropdown_item*) pour formater la liste déroulante.
3. (Bonus) Si vous avez terminé les deux questions précédentes, vous pouvez débiter celle-ci sur l'utilisation de plusieurs activités dans un même projet Android. Créer un projet dont le premier écran affiche un *TextView* ayant pour contenu : "Activité 1" et qui dispose d'un bouton ayant pour label "Aller à l'activité 2" permettant de basculer dans une nouvelle fenêtre (celle de l'activité 2). Dans Activité 2 vous devrez mettre un *TextView* qui affiche "Activité 2" ainsi qu'un bouton avec le label «Allez à l'activité 1 » qui permettra d'arrêter l'activité courante pour retourner à la première.

Exercice 8 : Calculatrice

Développer une application ANDROID selon le modèle de la calculatrice de base.

Exercice 9 : Jeu des trois pions

Le jeu du morpion qui se joue ici sur un échiquier de 3 sur 3. Le jeu se joue à deux. Chaque joueur pose une pièce sur l'échiquier puis laisse son adversaire poser sa pièce (d'habitude, les pions sont 'X' et 'O'); celui qui réussit à aligner 3 de ses pions en ligne, colonne ou diagonale, a gagné. Développer une application ANDROID permettant de jouer à ce jeu contre le téléphone. Bien avant de démarrer, il est demandé de générer aléatoirement les types de pions pour chacun.