

# Application MyColorPicker

Une couleur est codée sur un entier de 32 bits, c'est à dire 4 octets :

- le premier octet, appelé canal alpha, représente le degré de transparence de la couleur , 0 signifie que la couleur est totalement transparente et 255 qu'elle est opaque,
- le deuxième octet permet de coder l'intensité du rouge de 0 à 255,
- le troisième octet permet de coder l'intensité du vert de 0 à 255,
- le quatrième octet permet de coder l'intensité du bleu de 0 à 255.

Sous Android et sous d'autres systèmes il est traditionnel de représenter une couleur par une chaîne de caractères commençant par # suivi du codage en hexadécimal des données mentionnées ci-dessus.

Exemples :

#00ff0000 correspond à un canal alpha valant 0 et une intensité maximum de rouge (255). Le résultat est donc invisible ! En effet, le canal alpha valant à 0, la couleur est totalement transparente.

#ff000000 correspond à du noir

#ff00ff00 correspond à du vert

#ffffffffff correspond à du blanc

#3aff00ff correspond à du violet légèrement transparent.

En utilisant la documentation en ligne de l'API Android :

<http://developer.android.com/>

définissez une application utilisant un **LinearLayout** dans lequel vous placerez 2 **TextView** et 4 barres de défilement (cf. Fig . 1).

Utilisez la ressource **android:layout\_weight** afin de spécifier que la deuxième **TextView** doit occuper toute la place restante une fois la première zone et les 4 barres de défilement placées.

Trouvez le nom de la ressource permettant de donner comme couleur de fond par défaut à la deuxième **TextView** la valeur #00000000

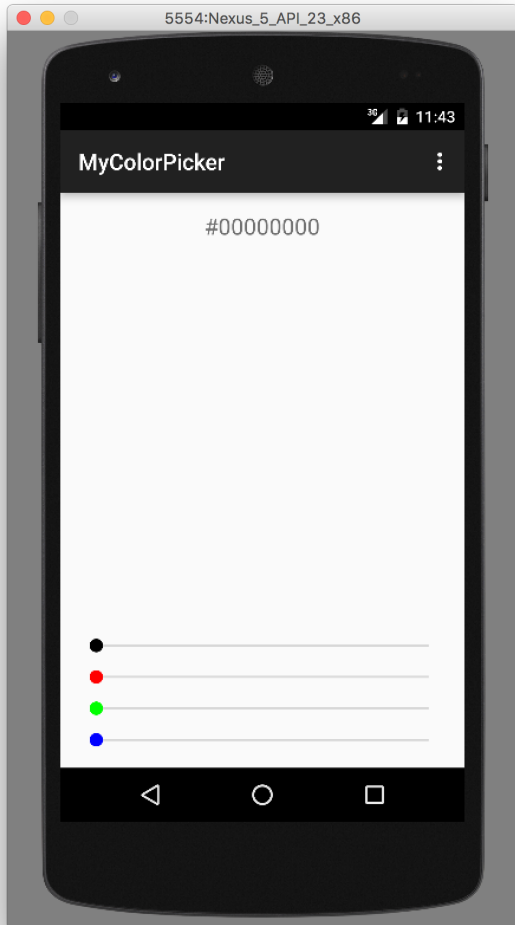
Trouvez le nom de la ressource permettant de spécifier que chaque barre de défilement permet de représenter au maximum la valeur 255.

Chaque barre de défilement représente l'une des 4 valeurs nécessaires à l'encodage d'une couleur. L'objectif de ce TP est d'afficher, lorsque l'utilisateur bouge les curseurs des différentes barres de défilement, un rectangle de la couleur correspondante dans la deuxième **TextView** et le code couleur sous forme de chaîne de caractères dans la première **TextView** (cf. Fig. 2).

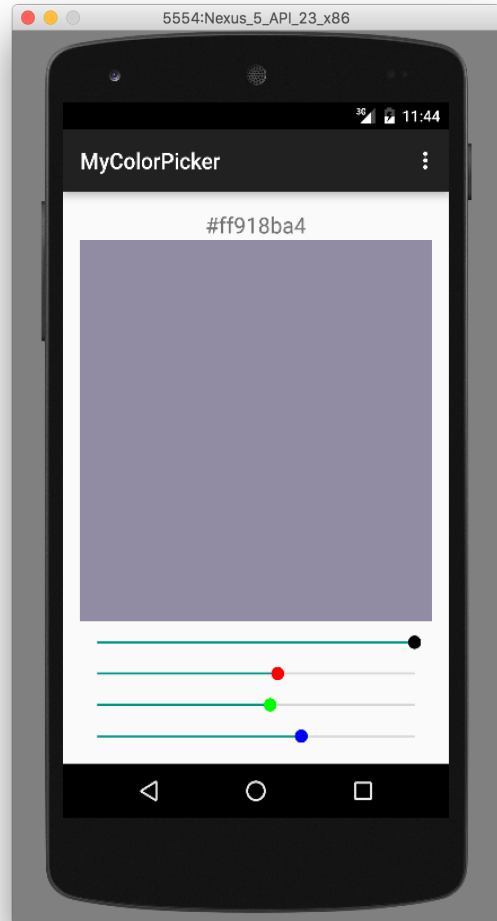
Au lancement de l'application, le code couleur par défaut affiché est #00000000 et la zone contient un rectangle noir totalement transparent (donc invisible, cf. Fig. 1).

Vous devrez placer un « écouteur » (listener) sur chaque barre de défilement. Votre classe principale devra donc implémenter l'interface `SeekBar.OnSeekBarChangeListener`. Regardez la documentation de la méthode `onProgressChanged` afin de déterminer comment récupérer la valeur associée à une position du curseur dans la barre de défilement.

Trouvez le nom de la ressource permettant de sélectionner le code couleur afin de le copier dans le presse-papier.



*Fig. 1*



*Fig. 2*