

Évaluation formative 06

Objectif

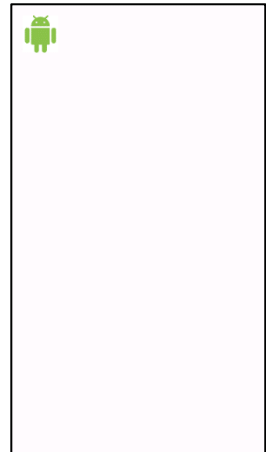
Se familiariser avec la gestion d'évènements `MotionEvent`.

Problème

Nous avons vu en classe comment capturer les mouvements de doigts sur l'écran dans une application *Android Studio*.

Dans le cadre de cet exercice, vous devez concevoir une application *Android* exploitant une telle gestion pour déplacer un widget dans l'activité :

1. Déposez un `ImageView` dans l'activité, et attribuez-lui l'image du *droid* mise à votre disposition par l'instructeur.
2. Exploitez l'évènement `onTouch()` du *layout* de l'activité afin pour repositionner le *droid* à l'emplacement touché. Pour repositionner un widget, on utilise ses attributs `X` et `Y`.
3. Assurez-vous que le *droid* soit centré à la position du toucher. Vous devez donc prendre en considération sa taille (via ses attributs `width` et `height`).



Une fois l'application opérationnel, vous devriez être capable déplacer le *droid* à volonté simplement en déplaçant votre doigt sur l'écran. Le *droid* devrait ainsi « suivre » votre doigt dans l'activité.

Amélioration

Affichez en tout temps la position du *droid* dans un `TextView` au centre de l'écran.

Matériel fourni

Aucun matériel n'est fourni avec cette évaluation formative.

Suggestions

- a) Attribuez un *listener* de type `Touch` au *layout* de l'activité.
- b) Lorsque l'évènement `onTouch` du *layout* est déclenché, vérifiez si la position touchée correspond à l'emplacement du `ImageView`.
- c) Exploitez un attribut membre de type booléen, initialisé à `false`, pour gérer le déplacement du *droid*. Lorsque l'évènement est de type `MotionEvent.ACTION_MOVE`, déplacez le `ImageView` seulement si cet attribut est `true`.
- d) Pour déterminer si la position du doigt est sur le *droid*, vous pouvez exploiter la fonction suivante :

```
// Fonction indiquant si la position donnée est sur le droid
private fun positionSurDroid(x: Float, y: Float): Boolean {
    val droidOrigineX: Float = binding.droid.x
    val droidOrigineY: Float = binding.droid.y
    val droidOpposéX: Float = droidOrigineX + binding.droid.width
    val droidOpposéY: Float = droidOrigineY + binding.droid.height
    return (x >= droidOrigineX) && (x <= droidOpposéX) &&
        (y >= droidOrigineY) && (y <= droidOpposéY)
}
```

À soumettre

Une fois l'évaluation solutionné, démontrez à l'instructeur que ça fonctionne bien.