



VIII – Réalité Virtuelle avec Processing sur Android

125

VIII.2 – Processing en mode Android

<https://android.processing.org/>

Processing for Android



[Home](#)
[Install](#)
[Reference](#)
[Tutorials](#)
[Books](#)

Use Processing for Android to create Android apps with ease, including live wallpapers, watch faces, and VR apps. You can run your Processing sketches on Android devices (phones, tablets, watches) with little or no changes in the code, and without worrying about installing SDK files or editing layout files. Processing for Android also lets you access the Android API to read sensor data, and exporting your sketch as a signed package ready to upload to the Google Play Store.

Follow [@p5android](#) and [@ProcessingOrg](#) on Twitter, and also [Processing Foundation Medium's posts](#), for news and updates!

126

VIII.2 – Processing en mode Android sur PC

- Ajout du mode « android »
 - autorise Processing à éditer du code Android (java)



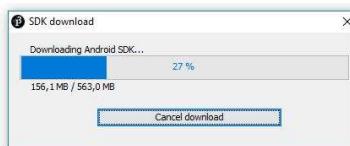
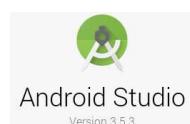
- Ajout du compilateur « android »
 - utilisation d'un SDK Android pour que Processing puisse compiler et exécuter du code Android

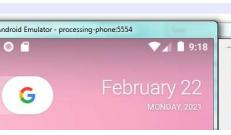


127

VIII.2 – Processing en mode Android sur PC

- Installation SDK Android
 - Solution 1 – par Processing :
 - « Download SDK automatically »
 - Solution 2 – par installation complète d'Android Studio IDE
 - Puis « SDK manager » d'Android Studio :
installation API 26 – Android 8.0 (Oreo)
 - « Locate SDK path manually » :
C:\Users\XXX\AppData\Local\Android\Sdk
 - Puis installation émulateur « Processing-phone » :
 - A mettre dans Documents >Processing > android > avd
 - Vérification sous « AVD manager » d'Android Studio :



```

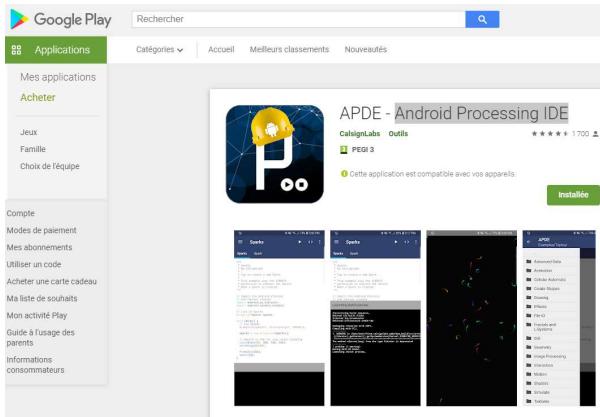
    android.emulator.image.abi=x86
    android.emulator.port.phone=5566
    android.emulator.resolution=480x800
    android.sdk.apkcompat=1.2.0
    android.sdk.arc=1.22.0
    android.sdk.gyr=1.180.0
    android.sdk.platform=Users\Emmanu
    android.sdk.sdk=~/Android/Sdk
    android.sdk.support=26.0.2
    android.sdk.target=26
    android.sdk.vr=false
    android.sdk.vrlegacy=1.0.0
    android.sdk.wear=false
    android.sdk.wear2=false
  
```

Attention : le numéro doit être **identique** à celui
du fichier « preferences.txt » de Processing !

128

VIII.2 – Processing en mode Android sur smartphone Android

- Nouveauté : Processing a été porté sur Android
 - Ecriture directe de sketch sur tablette / smartphone !
 - APDE : Android Processing IDE
<https://github.com/Calsign/APDE>



129

VIII.2 – Applications sous Android

- Différents types d'application Android
 - App : application « standard »
 - VR : application de Réalité Virtuelle
 - avec prise en compte des capteurs de mouvements
 - et de la déformation des lentilles !
 - AR : application de Réalité Augmentée
 - Google AR Core
 - Wallpaper : application de fond d'écran
 - fonds d'écran dynamiques
 - Watch Face : application de type montre
- Création (optionnelle) d'un APK
 - Package signé :
 - menu Fichier>Export Signed Package
 - Attention au nom et aux diverses contraintes imposées pour la « distribution »
<https://android.processing.org/tutorials/distributing/index.html>
 - ou bien, sans signer mais fonctionnel quand même :



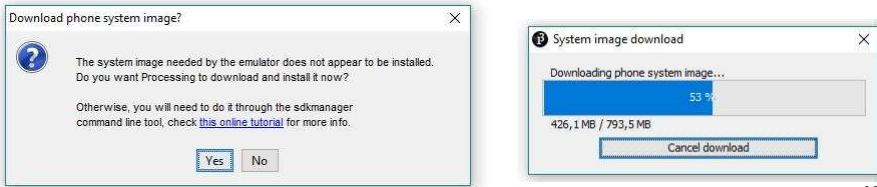
```
Building Android project...
Build folder: C:\Users\Emmanuel\AppData\Local\Temp\android8702084865744786144sketch
```

```
C:\Users\Emmanuel\AppData\Local\Temp\android8702084865744786144sketch\app\build\outputs\apk\debug
```

130

VIII.3 – Principe de compilation et d'exécution

- Vers le smartphone :
 - Smartphone relié à l'ordinateur
 - via l'USB,
 - avec mode développeur activé
 - Appui sur le bouton **play** :
Compilation (lente...) puis installation et exécution sur smartphone
(liste des périphériques autorisés dans menu *Android>Devices*)
- Vers l'émulateur :
 - Appui sur le bouton **play** avec **Majuscule** enfoncée :
Compilation (lente...) puis exécution sur émulateur
 - Rappel : APK disponible dans android...sketch > App > build > outputs > apk > debug
 - Attention : demande d'installation de l'image d'un système android pour téléphone lors de la première exécution :



131

VIII.4 – Ecriture d'applications *App* pour smartphone Android

- Modifications à apporter aux applications « Processing » : rien ! ou presque...
 - Processing en « Mode Android », et application de type « App »
 - **Retrait** des fonctions liées à l'objet « surface »
 - `surface.setTitle("Exemple ... // Impossible sur Android")`
 - Eventuellement, remplacement de `size()` par `fullScreen()`
 - valable en 3D également : `fullScreen(P3D) // Ouverture en mode 3D`
 - De ce fait, il est possible de :
 - restituer sur l'écran des objets simples (2D ou 3D), tel qu'un cube tournant !
 - gérer l'équivalent de la « souris » (`mousePressed`, ...)
- Reprise de l'exemple du cube texturé :



132

VIII.4 – Exemple 12 : application App du cube texturé

- Résultat identique sous émulateur :

The screenshot shows the Processing IDE with the sketch 'Exemple_10_Android_CubeTexture.pde'. Below it are two Android Emulator windows. The left window shows a green cube, and the right window shows a blue cube with a texture. The Processing code includes comments about drawing two cubes (simple and textured) and adapting Example 7 for Android.

Rotation écran possible :

The screenshot shows the Android Emulator with a rotated textured cube, demonstrating the rotation capability mentioned in the text.

133

TD 9 Application Android « Classique »

- Travail demandé :

 - Faire une application au format « App » qui permet de tester les fonctionnalités d'entrées/sorties basiques de toute application : touchscreen, clavier, souris, ...

- Spécifications de l'application « AndroidPaint » :

 - Ecrire du texte sur l'écran et permettre également la saisie sur clavier (voir `displayDensity` et `openKeyboard`)
 - Gérer le touchscreen pour effectuer le tracé du dessin :
 - dessiner un point au niveau du contact sur l'écran (voir `touchStarted`)
 - éventuellement, prévoir un bouton pour effacer l'écran,
 - un petit sélecteur de couleur pour égayer les tracés, ...
 - Demander la confirmation avant de quitter (voir `backPressed`)
 - Eviter le redémarrage à cause de l'auto-rotate
 - Appel à `orientation()` dans la fonction `setup` (paramètre LANDSCAPE ou PORTRAIT)

134