

1. Introduction

Présentation du plan de cours, aperçu de *Android Studio* et de l'émulateur d'appareils Android (AVD).

Android Studio HedgeHog Essentials, Kotlin Edition : chapitres 1 à 8

1.1. Plan de cours

1. Présentez le PDF du plan de cours à l'écran
2. Affichez dans le navigateur Web le livre via *Payload publishing*

<https://www.payloadbooks.com/product/android-studio-hedgehog-essentials-kotlin-edition/>

3. Décrivez comment seront évalués les travaux

1.2. Installation d'Android Studio (si ce n'est pas déjà fait)

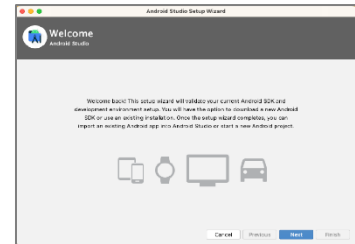
4. Énumérez les prérequis physiques
 - *Windows, Linux* ou *OSX*
 - 8 Go de RAM, 8 Go d'espace disque
 - Résolution minimale d'écran de 1280 x 800
5. Avec les étudiants, téléchargez puis installez la plus récente version de *Android Studio* à partir de <https://developer.android.com/studio/index.html>
 - Soulignez que le fichier d'installation est assez volumineux, donc requiert un temps de téléchargement et d'installation considérable
 - Expliquez que nous installons *Android Studio* sur un appareil Windows. Référez les étudiants possédant un appareil *MacOS* ou *Linux* au chapitre 2 du manuel de référence pour obtenir les instructions d'installation

6. Une fois le fichier d'installation téléchargé, double-cliquez dessus afin de démarrer le processus d'installation

- Suivre les étapes d'installation telles que proposées, en s'assurant de cocher les composants *Android Studio* et *Android Virtual Device* lorsque demandé

7. Expliquez que la procédure d'installation diffère sensiblement si une version antérieure d'*Android Studio* a déjà été installé sur l'appareil

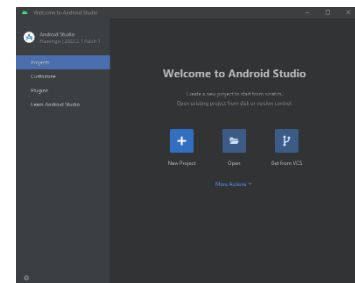
- Si c'est une première installation, le *Android Studio Setup Wizard* est affiché; ce script d'installation configurera l'appareil pour *Android Studio*. Suggérez aux étudiants d'accepter les options d'installations proposées par le script



- Si ce n'est pas une première installation, le *Android Studio Setup Wizard* ne sera pas affiché puisque l'appareil est déjà configuré pour accepter *Android Studio*

8. Lorsque le processus d'installation est complété, *Android Studio* affiche alors son écran de démarrage

9. Soulignez que trois thèmes de couleurs sont proposés via l'onglet « Customize » (l'exemple ci-contre exploite le thème « Dracula »)



10. Soulignez les éléments de configuration qui seront utilisés en laboratoire

- On développera des projets visant l'**API 26 (Oreo)** de *Android*, afin que celles-ci puissent être installées sur la plupart des appareils physiques au besoin
- On utilisera les émulateurs du **Pixel 5** (téléphone) et **Pixel C** (tablette) pour tester nos applications

11. Soulignez que *Android Studio* requiert BEAUCOUP de ressources

- Il faut donc utiliser un ordinateur puissant (CPU rapide, carte graphique performante, beaucoup d'espace disque)

- Proposez à ceux n'ayant pas d'ordinateur assez puissant le service en ligne *Macincloud* (<https://www.macincloud.com/>)

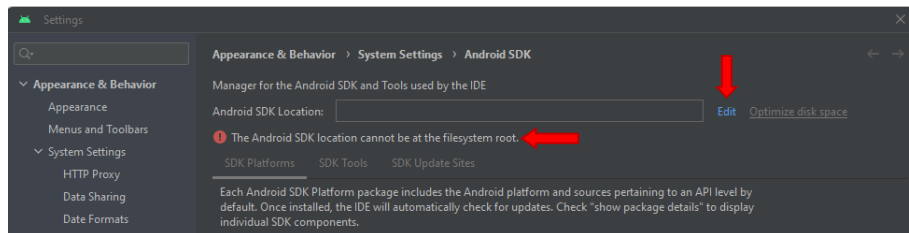
1.3. Configuration de Android Studio

12. Expliquez aux étudiants qu'on va maintenant configurer *Android Studio* sur leur appareil. Si ce n'est déjà fait, démarrez *Android Studio*

- Indiquez aux étudiants qu'ils doivent effectuer les mêmes opérations que l'instructeur et qu'ils devront démontrer que leur implantation résultante est fonctionnelle

13. Dans la fenêtre d'accueil d'*Android Studio*, cliquez le bouton « More Actions ▼ », sélectionnez l'option « SDK Manager » afin de configurer celui-ci

14. Si le panneau « Settings » affiché indique une erreur au niveau du répertoire d'installation du SDK, cliquez le bouton « Éditer » pour résoudre l'erreur



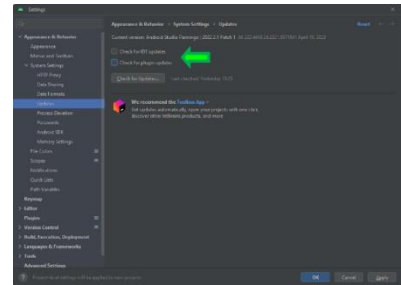
15. Dans le panneau « SDK Platforms », cochez « Android 8.0 ("Oreo") – API Level 26 » si celui-ci ne l'est pas déjà. Cliquez ensuite le bouton « Apply » afin de compléter l'installation du nouveau SDK sur l'appareil

16. Dans le panneau « SDK Tools », cochez les outils suivants s'ils ne sont pas installés ou si une mise à jour est disponible :

- Android SDK Build-tools 34
- Android SDK Platform-tools
- Android Emulator
- Intel x86 Emulator Accelerator (HAXM installer)

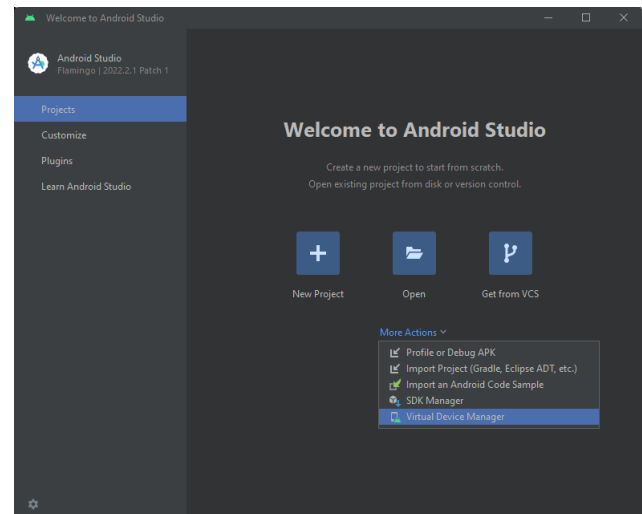
17. Avant de poursuivre la configuration, nous devons désactiver les mises à jour automatisées d'*Android Studio* afin de s'assurer que tous travaillent avec la même version de l'environnement de développement durant tout le trimestre

- Dans l'écran d'accueil d'*Android Studio*, cliquez l'onglet « Customize » à gauche, puis sélectionnez le lien « All settings... » à droite
- Via le panneau de droite, sélectionnez « Appearance & Behavior → System Settings → Updates », puis dans le panneau central désactivez les options « Check for IDE updates » et « Check for plugin updates »; appuyez ensuite « OK »

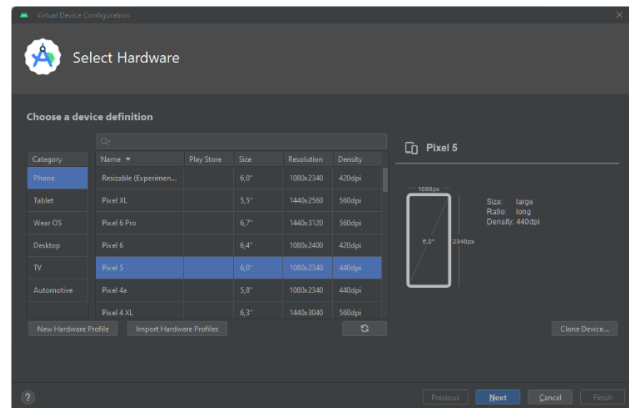


18. Enfin, la dernière étape de configuration d'*Android Studio* consiste à créer un téléphone et une tablette virtuels pour tester les applications développées en classe

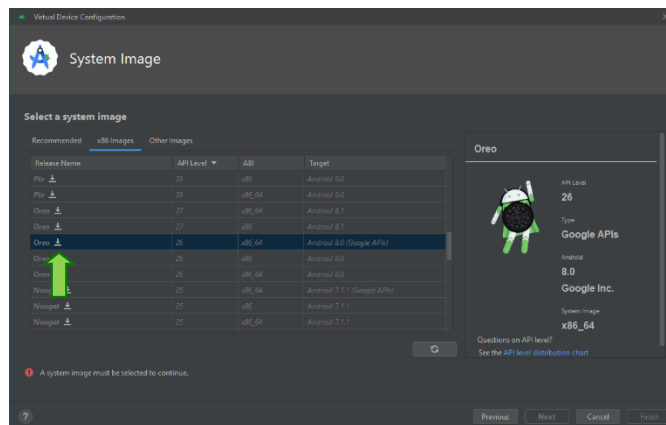
- Expliquez aux étudiants qu'il n'est pas nécessaire de posséder un appareil *Android* pour tester les applications développées puisque *Android Studio* offre des appareils virtualisés, que nous utiliserons durant le trimestre. Il est cependant relativement facile de tester nos applications sur un appareil *Android* physique; on y reviendra plus tard
- Dans la fenêtre d'accueil, sélectionnez « More Actions → Virtual Device Manager »
- Dans la fenêtre « Device Manager », cliquez le bouton « + », ce qui enclenche le processus de création d'un appareil virtuel



- Sélectionnez un téléphone de type « Pixel 5 » de la catégorie « Phone », puis appuyez « Next »



- Dans la prochaine fenêtre, sélectionnez l'onglet « x86 Images », puis cliquez le lien de téléchargement accompagnant l'image « Oreo / 26 / x86_64 / Android 26.0 (Google APIs) »

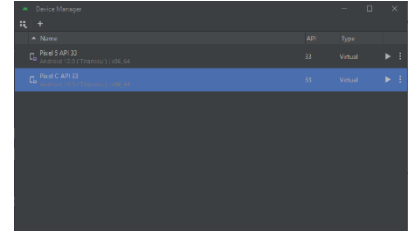


- Une fois le téléchargement et l'installation de l'image complétés, appuyez « Finish » pour retourner à la fenêtre « Virtual Device Configuration », assurez-vous que l'image « Oreo » sélectionnée précédemment l'est toujours (le lien de téléchargement ne devrait plus y être), puis appuyez « Next »
- La prochaine fenêtre récapitule la configuration de l'appareil virtuel à créer. Appuyez « Finish » pour procéder à sa création

19. La fenêtre « Device Manager » devrait maintenant afficher le nouvel appareil

20. Puisque nous allons aussi étudier pendant le trimestre les particularités du développement d'applications pour tablettes, vous devez maintenant répéter les étapes ci-dessus pour créer une tablette virtuelle

- Appuyez le bouton « Create Device... » de la fenêtre « Device Manager »
- Cette fois, sélectionnez la « Pixel C » de la catégorie « Tablet »
- Pour les étapes de création suivantes, sélectionnez les mêmes paramètres que lors de la création du téléphone virtuel; vous n'aurez cependant pas à télécharger à nouveau l'image de *Oreo* pour la tablette puisqu'elle l'est déjà



21. La fenêtre « Device Manager » devrait maintenant afficher les deux appareils virtuels; fermez cette fenêtre pour retourner à la fenêtre d'accueil d'*Android Studio*

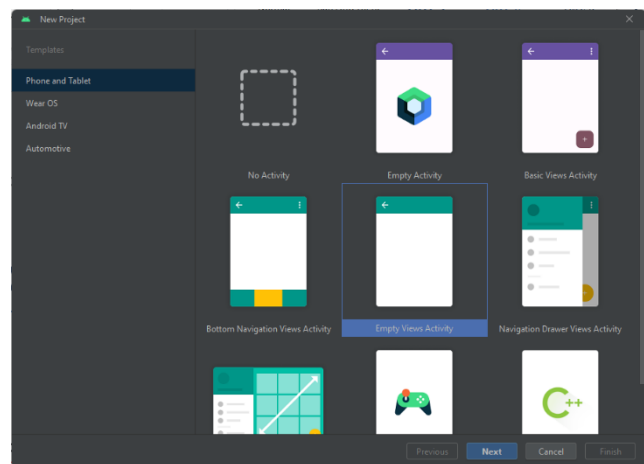
22. La configuration d'*Android Studio* est maintenant terminée

1.4. Première application Android

23. Démarrez *Android Studio* et expliquez les différentes options de gestion de projets offertes

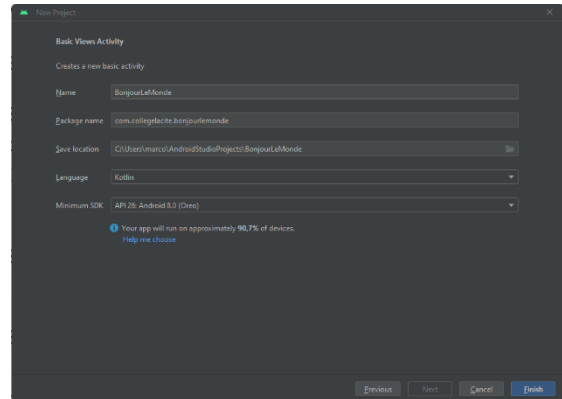
- Demandez aux étudiants de faire les mêmes opérations que vous

24. Sélectionnez le bouton « New Project », puis dans la fenêtre de gabarits, expliquez quelques types de projets d'applications *Android*. Sélectionnez « Empty Views Activity » de la catégorie « Phone and Tablet ». Appuyez ensuite le bouton « Next »



25. Configurez ensuite votre projet

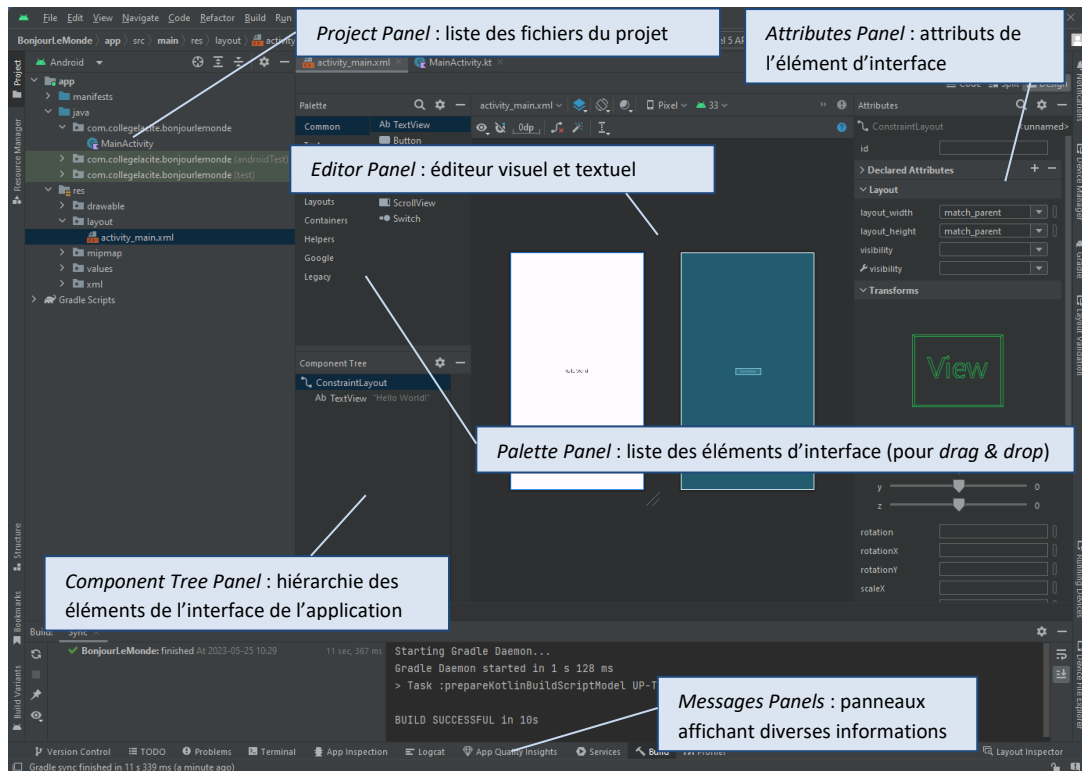
- Nommez le projet « *Bonjour Le Monde* » (sans les «*»*)
- Entrez le nom de domaine du collège en remplacement de celui proposé
- Sélectionnez un répertoire où créer le projet (vous pouvez accepter le répertoire proposé)
- Sélectionnez « Kotlin » comme langage de programmation
- Sélectionnez « API 26 ("Oreo"; Android 8.0) » comme niveau minimal d'API. Cliquez l'hyperlien « [Help me choose](#) » pour expliquer la raison qu'on sélectionne un API particulier
- Appuyez ensuite le bouton « Finish »



26. Soulignez aux étudiants qu'*Android Studio* va conserver leurs modifications à la configuration par défaut de projets, de sorte que les prochains projets seront automatiquement proposés avec la configuration ci-dessus

27. Expliquez qu'*Android Studio* va effectuer quelques téléchargements et installations supplémentaires lors de la création de ce premier projet, ce qui demande quelques minutes

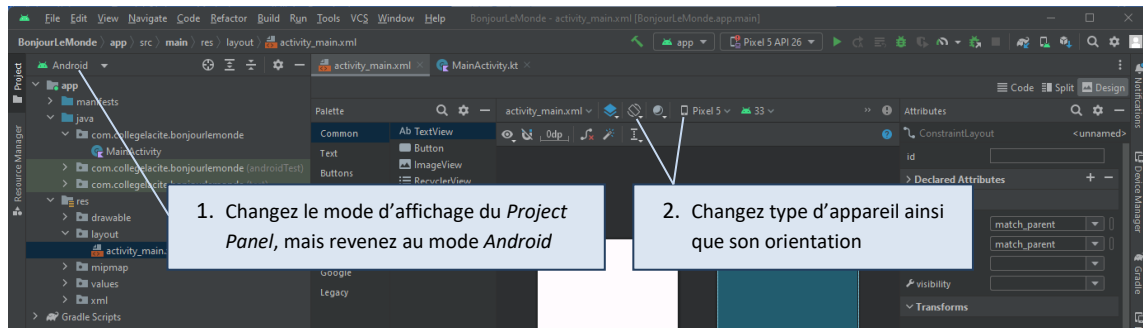
- Expliquez cependant que ces opérations ne sont effectuées qu'à la création du premier projet et consistent essentiellement à terminer la configuration d'*Android Studio* selon les caractéristiques de l'ordinateur

28. Donnez un bref aperçu des différents panneaux initialement affichés par *Android Studio*29. Expliquez qu'une *activité* (c.à.d. un écran de l'interface) est conçue à l'aide d'un fichier XML (accessibles via le *Project Panel* dans **app** → **res** → **layout**)

- **activity_main.xml** : fenêtre principale de l'application (en blanc par défaut dans le *Design*)

30. Expliquez qu'on peut créer des éléments d'interface par programmation, mais qu'il est beaucoup plus facile de le faire par *drag & drop* via la *Palette Panel*31. Montrez que le *Project Panel* peut afficher les fichiers projet selon différents aspects, mais qu'on s'en tient au mode **Android** pour l'instant

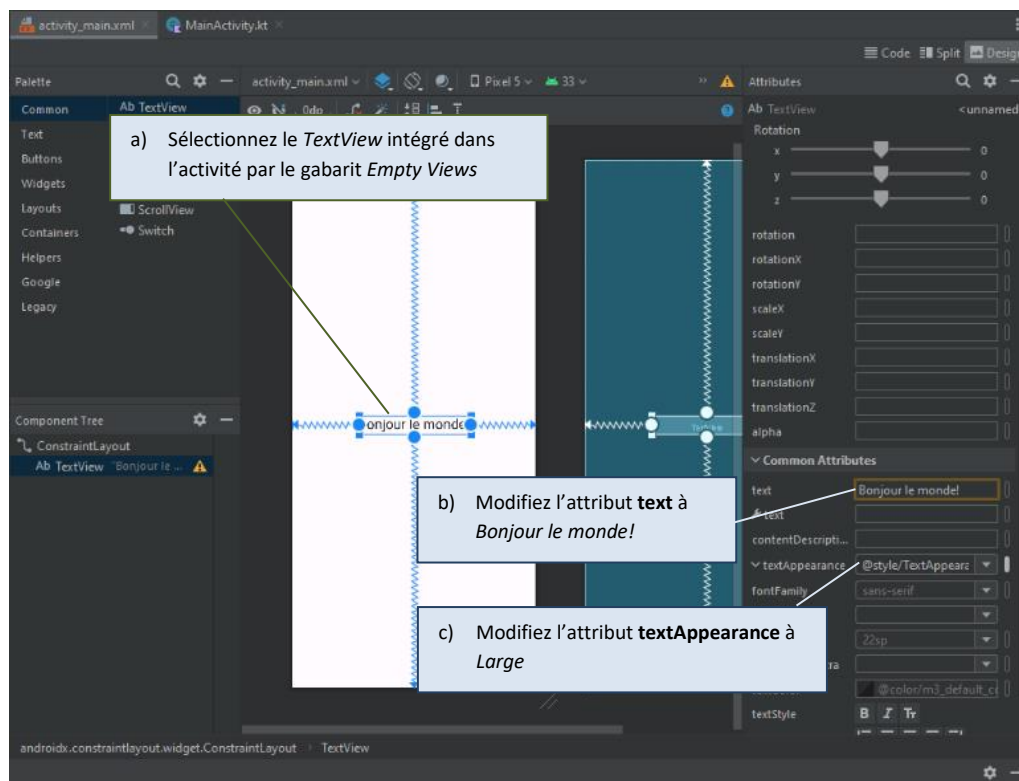
32. Démontrez que le *Designer Panel* peut afficher l'activité sur plusieurs types d'appareils ainsi que d'orientation. Sélectionnez cependant à **Pixel 5** en mode portrait.



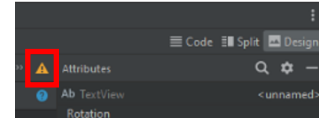
33. Affichez le panneau de droite (*Attributes Panel*) puis sélectionnez la scène pour afficher ses attributs et autres éléments du panneau de droite

34. On va modifier l'activité `activity_main.xml` via le *Designer Panel*.

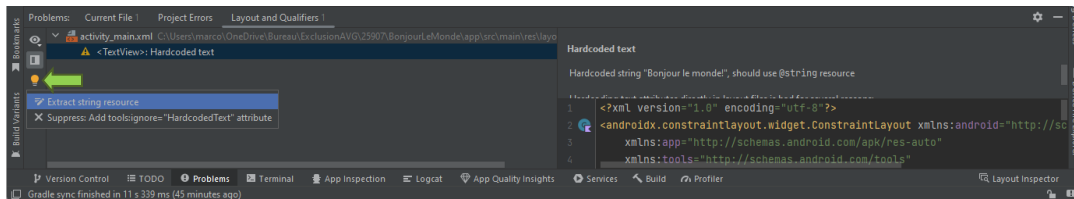
- Modifier l'attribut **text** du **TextView** à « Bonjour le monde ! » via le *Attributes Panel*, puis l'attribut **textAppearance** à **Large**.



35. Soulignez l'icône orange affichée par le *Designer Panel* en haut à droite

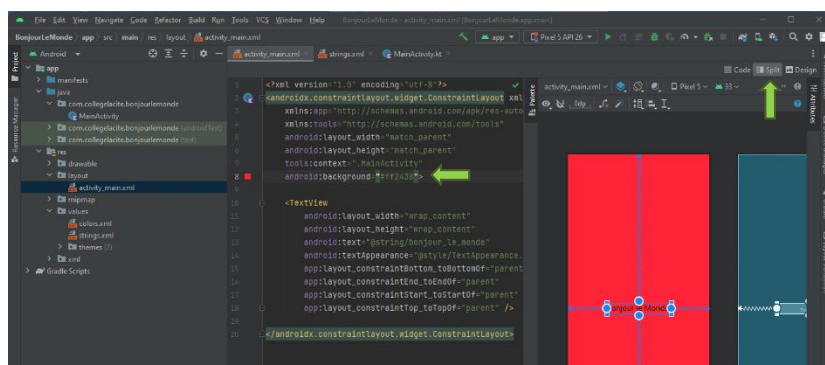


- Cliquez dessus et expliquez le message d'avertissement (<TextView>: Hardcoded text). Expliquez pourquoi on veut utiliser une telle ressource plutôt que « hardcoder » le texte directement dans sa propriété **text** (pour faciliter l'internationalisation)
- Sélectionnez le message d'avertissement et cliquez le bouton « Quick Fix » (c.à.d. 🛠️) en marge du panneau affichant l'avertissement, puis la commande « Extract string resource ».

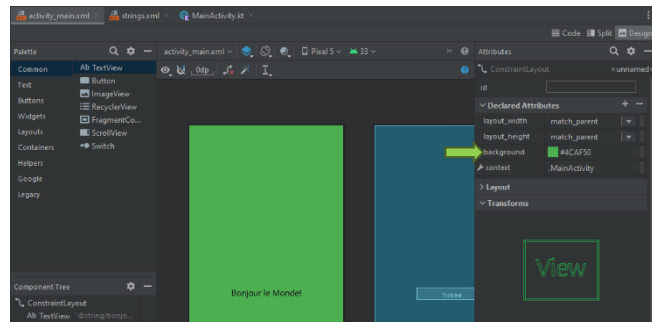


- Soulignez que la propriété **text** du widget **textView** a maintenant la valeur « @string/bonjour_le_monde » plutôt que « Bonjour le monde ! »
- Affichez le contenu du fichier **strings.xml** (accessibles via le *Project Panel* dans **app** → **res** → **values**) et soulignez la nouvelle ressource qui s'y trouve. Profitez-en pour changer sa valeur à « Bonjour le Monde ! » puis affichez à nouveau l'activité pour confirmer le changement à la ressource

36. Affichez le fichier **activity_main.xml** en mode Split, puis ajoutez une nouvelle balise au **ConstraintLayout** afin de changer la couleur de fond de l'activité : **android:background="#ff2438"**

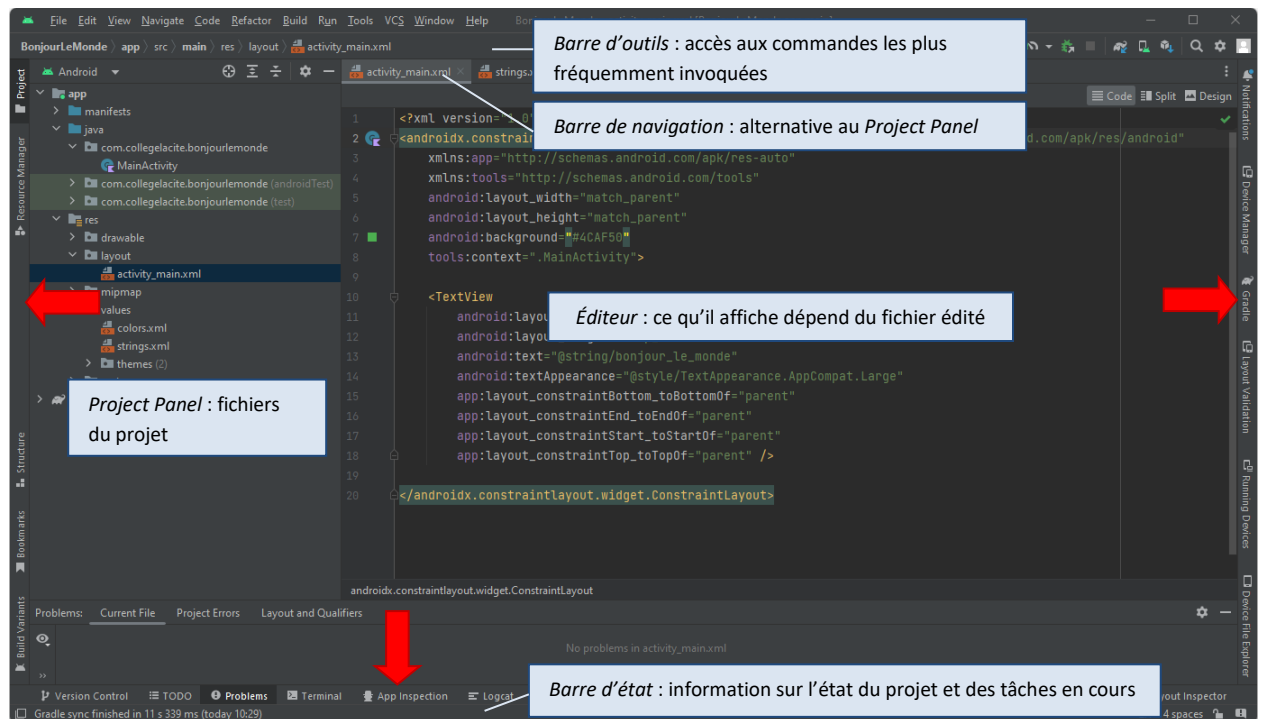


37. Changez à nouveau la couleur de fond de l'activité (en vert : #4CAF50), mais cette fois-ci via le *Attributes Panel*



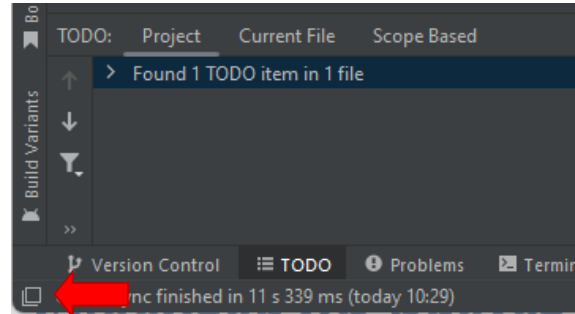
1.5. L'environnement de développement Android Studio

38. Décrivez les différents composants de l'environnement de développement



- Trois côtés de la fenêtre principale de *Android Studio* (désignées ci-dessus par les flèches rouges) affichent des boutons permettant d'afficher / cacher diverses fenêtres de l'environnement
- Montrez qu'on peut effectuer une recherche dans certaines de ces fenêtres (telle que le *Project Panel*) simplement en commençant à taper ce qu'on recherche

- Expliquez que **Gradle** est l'équivalent pour *Linux* de *make*
- Cliquez quelques fois le menu contextuel de la barre d'état afin de démontrez la facilité d'accès aux multiples fenêtres de l'environnement

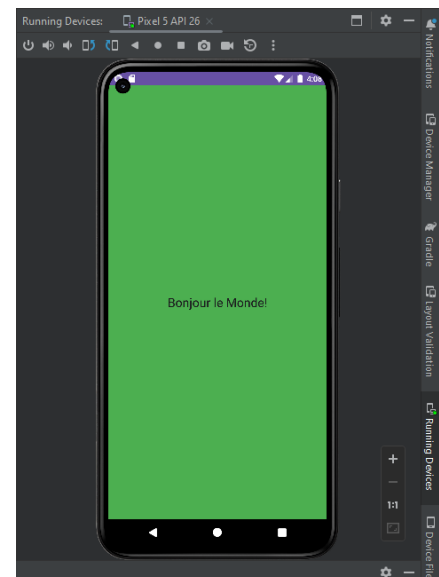
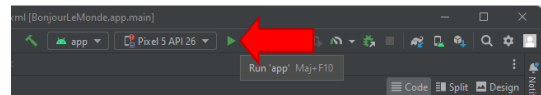


39. Décrivez ce qu'est la fenêtre *TODO* : énumération des blocs de commentaires de type *ToDo* dans le code source, permettant de les localiser rapidement. On y reviendra plus tard...

1.6. Exécuter un projet

40. Expliquez qu'un AVD est un émulateur pouvant simuler la plupart des appareils mobiles Android (téléphones et tablettes)

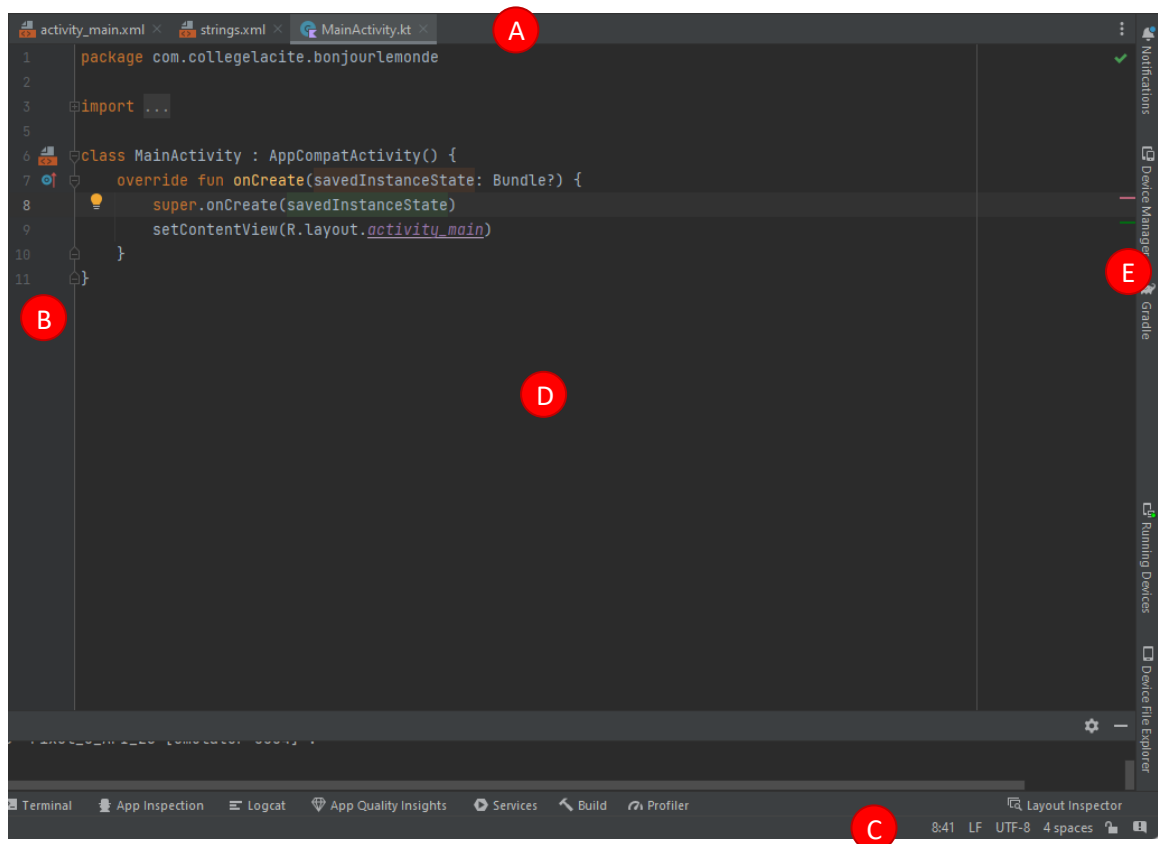
- Enfin, présentez le bouton « Run » dans la barre d'outils de *Android Studio* pour exécuter l'application. Sélectionnez au préalable l'AVD *Pixel 5 API 26* de la liste affichée au besoin, puis appuyez le bouton « Run »
- Expliquez que le démarrage de l'émulateur requiert beaucoup de temps, mais qu'une fois démarré, il est rapide d'exécution (il ne faut pas le fermer avant la fin de la classe)
- Explorez les boutons de l'appareil virtuel pour montrer qu'il se comporte comme un appareil physique
- Arrêtez l'exécution de l'application via le bouton à cet effet dans la barre d'outils de *Android Studio*



41. Explorez les différentes fonctionnalités de gestion de l'émulateur via les boutons de sa barre d'outils, généralement affichée au-dessus de l'appareil

1.7. L'éditeur de code

42. Décrivez les principaux éléments d'interface de l'éditeur de code



- A. **Onglets de fichiers** : illimités, peuvent être dragués pour détacher l'éditeur de la fenêtre principale
- B. **Marge d'indicateurs** : points d'arrêt, repli de code, modifications, numéros de ligne (clique droite pour menu contextuel permettant de les afficher)
- C. **Barre de statut** : position du curseur (clique gauche sur celle-ci pour aller à une position explicite)
- D. **Zone d'édition** : offre les fonctionnalités d'édition traditionnelles

E. **Barre de validation et marqueurs** : analyse de code en continu avec marqueurs informatifs et icônes d'état en haut de marge – carré vert, jaune ou rouge.

43. Décrivez brièvement les fonctionnalités les plus utiles de l'éditeur de code

- Split horizontal ou vertical (via le menu contextuel affiché en cliquant le bouton droit de la souris sur un onglet de fichier au haut de l'éditeur)
- Complétion de code (*Enter* ou *Tab* pour accepter la complétion suggérée, flèches pour une autre sélection)
- Repli de code à l'aide des icônes en marge gauche de l'éditeur
- Commenter les lignes de code sources sélectionnées via *Cmd+/,* et les décommenter avec *Cmd+Alt+/,*
- Reformatage de code avec *Cmd+Alt+L* (ou *Ctrl+Alt+Shift+L*)
- Expliquez que plusieurs raccourcis de clavier diffèrent selon la plateforme (*Linux*, *Windows* ou *OSX*)

Évaluation formative 01 Les étudiants doivent créer un projet *Android Studio* selon les spécifications données et démontrer à l'instructeur que leur application résultante est fonctionnelle.