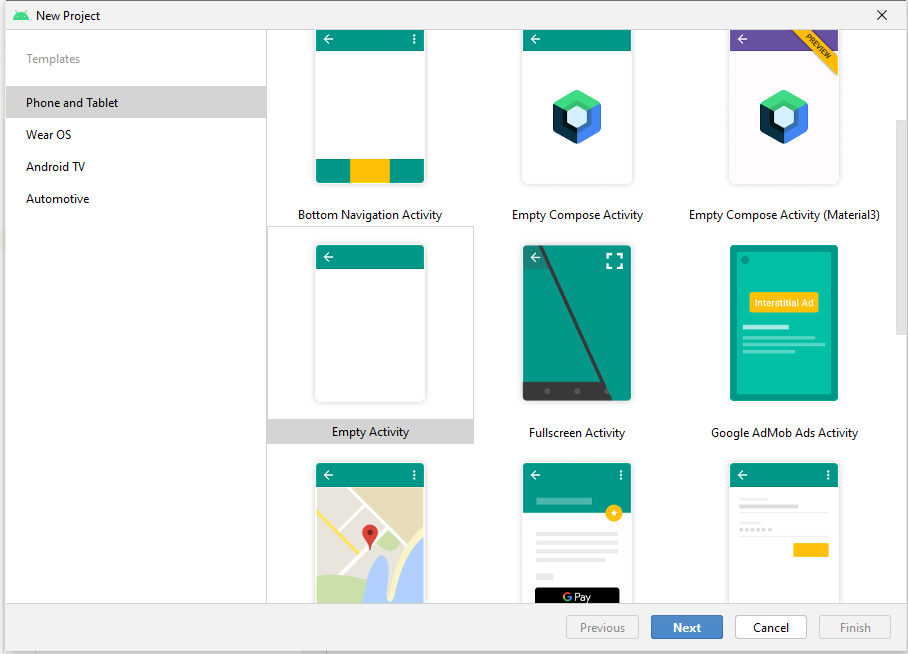
# Création d’un jeu Memory pour Android

## Description

Noël approche ! Et nous allons créer un petit jeu pour notre famille permettant de faire travailler la mémoire. Ce jeu consiste à retrouver des pairs de cartes identiques. Dans un premier temps, 20 cartes sont retournées (on ne voit que le dos des cartes). Dans la première version, une paire de deux chiffres comprit entre 0 et 9 sont utilisés. Initialement, tous les chiffres sont cachés. L’utilisateur retourne deux cartes, si les deux cartes sont identiques, alors ces cartes restent retournées sinon elles sont de nouveau retournées pour cacher les chiffres. L’objectif est simple : retourner toutes les cartes en un minimum d’essais. Cette version servira de base pour la seconde qui utilisera des images en lieu et place de chiffre. La troisième version permettra de sauvegarder le score dans une base de données puis la dernière version permettra le mode multijoueur en utilisant le Bluetooth. Pour créer ce jeu, nous utilisons l’IDE Android Studio et le langage JAVA.

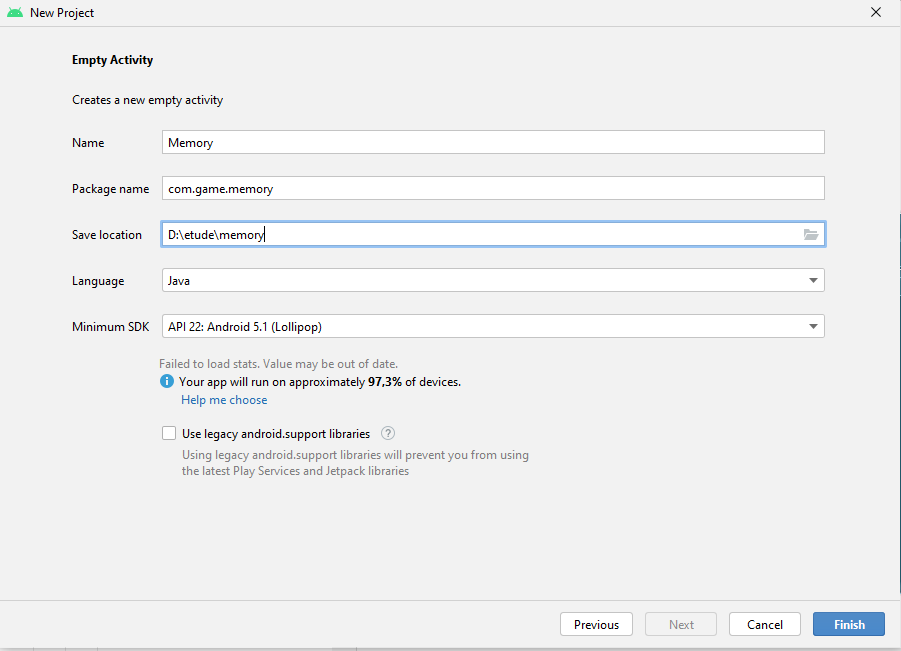
## Création du projet Android « Memory »

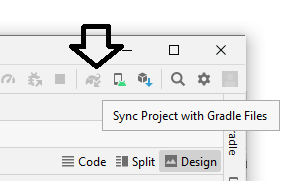
En utilisant Android Studio, il faut tout d’abord créer un projet vide « Empty Activity ».



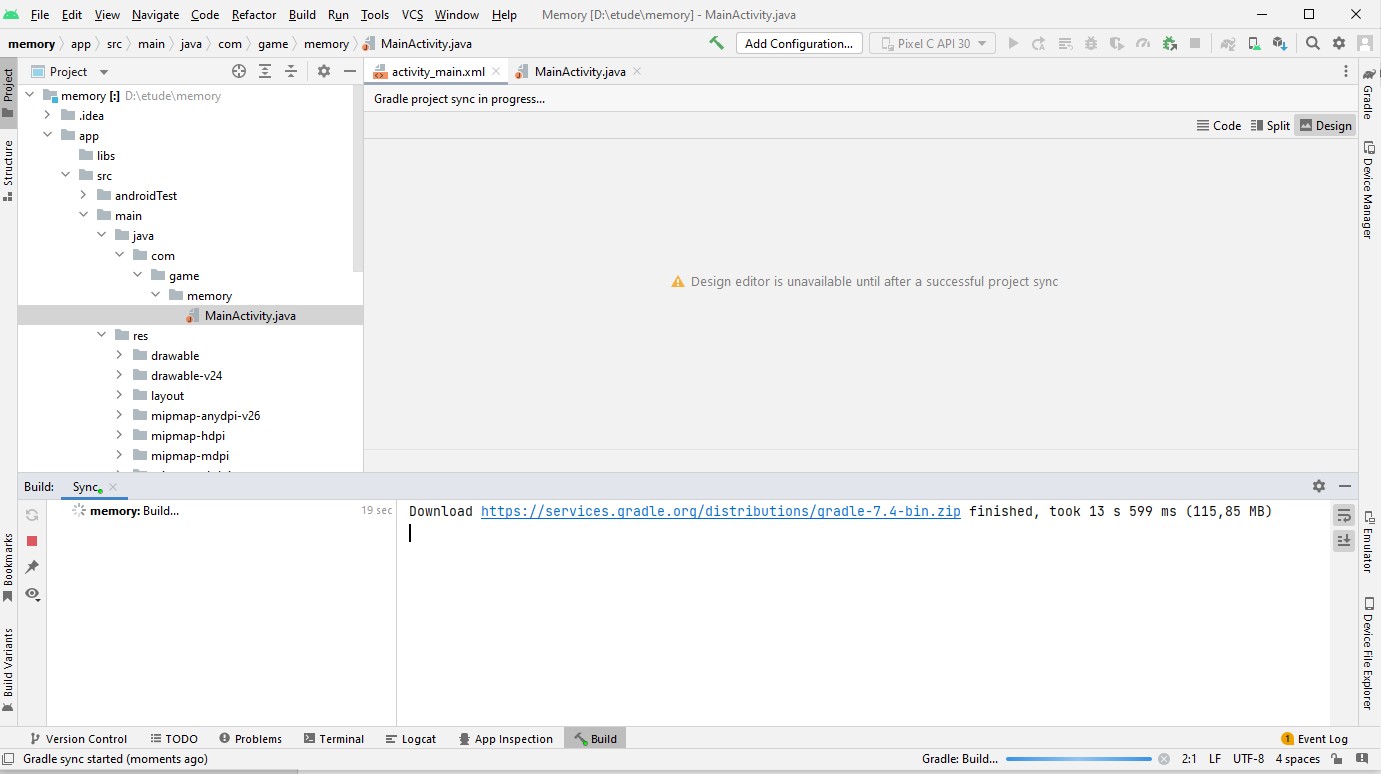
Empty Activity

Le nom de notre projet est « memory », pensez à utiliser le lecteur « D » comme répertoire de sauvegarde. Nous allons ensuite choisir le SDK **Lollipop** compatible à 97,3% à tous les devices actuels. Puis finaliser la création de votre projet en activant le bouton Finish.



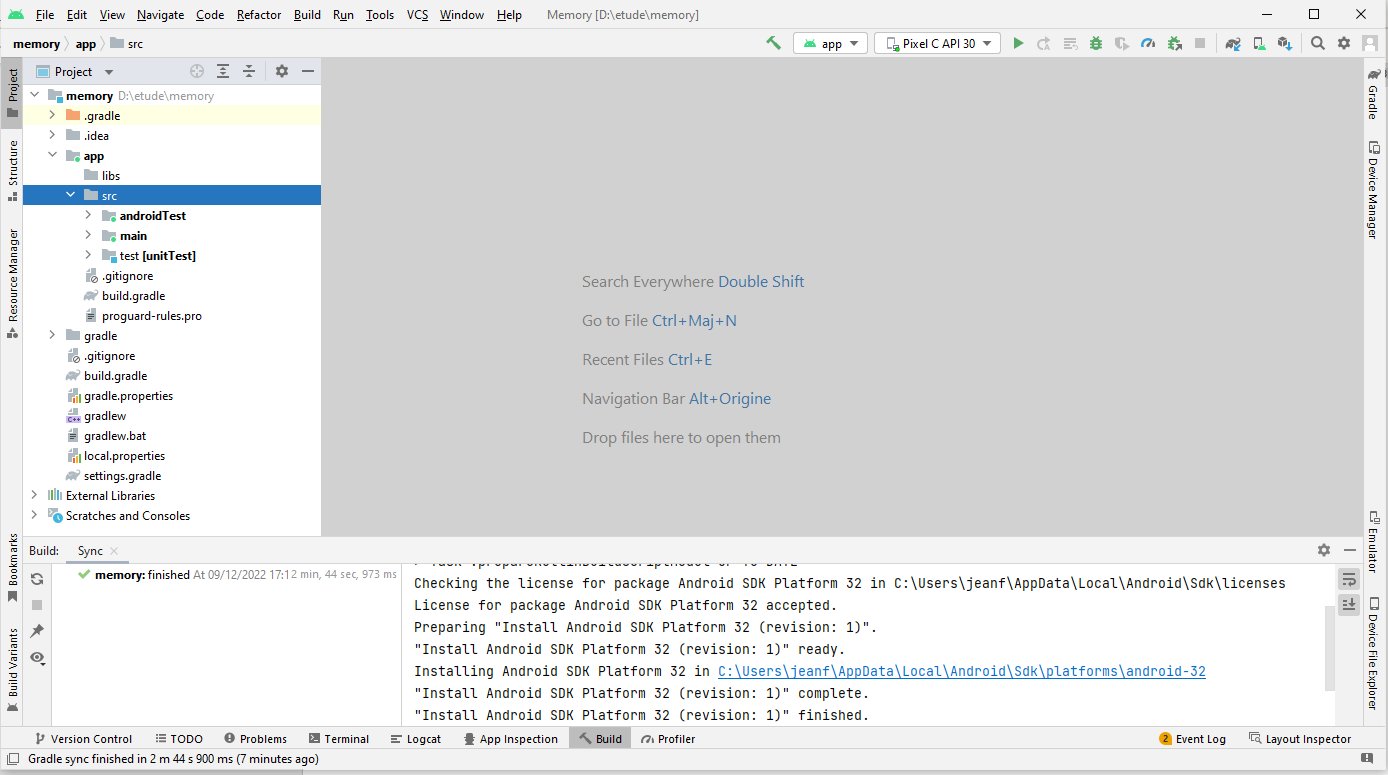
Si une erreur apparait lors du chargement du SDK, il faut simplement resynchroniser le **gradle** en cliquant sur le bouton en haut à gauche « **Sync Projet with Gradle File**s » ou par le menu « **File** ».

Une image contenant texte

Description générée automatiquement

## Interface Android Studio

Le projet memory est créé et nous pouvons commencer à l’implémenter à l’aide de l’IDE Android studio. Cet IDE se subdivise en quatre grandes parties :



4 Fenêtre d’information

3 Menu et barre d’action

2 Explorateur projet

1 zone de développement

Naviguez dans l’IDE pour le prendre en main.

## Ancroid Device Virtuel vs Environnement physique

Pour exécuter et vérifier le fonctionnement de notre application, il est possible d’utiliser sont smartphone ou de créer un device virtuel. Nous aborderons dans un premier temps la création d’un device virtuel puis dans un second temps l’utilisation de notre smartphone.

### Création d’un device virtuel

Pour créer un device virtuel, il faut tout d’abord cliquer sur « l’ADV Manager » (Android Device Virtual Manager) ou « **Device Manager** » puis le bouton « **create device** » permettant ainsi de configurer le device que nous souhaitons créer. Par ailleurs, il est possible de créer un téléphone, une tablette ou montre connectée…

Une image contenant texte

Description générée automatiquement

#### Création d’une réplication d’un Galaxy A12

Une image contenant texte

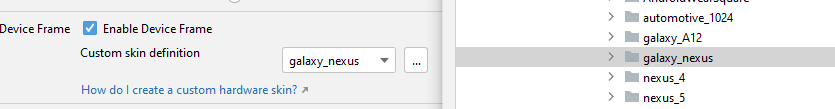
Description générée automatiquementHeureux possesseur d’un Galaxy A12 Sansung, je souhaite réaliser une duplication de mon téléphone dans Android Studio en VM. J’utilise le site du constructeur en tant que développeur puis je récupère les caractéristiques de mon téléphone tout en récupérant le « Skin » de mon smartphone. Le skin « (peau) permet dans au simulateur de donner le look extérieur de votre VM Smartphone à l’identique du vrai téléphone. Le skin représente le « background », « foreground », les « buttons/keyboard » du smartphone.

|  |  |
| --- | --- |
|  | Une image contenant texte  Description générée automatiquement |

Je dépose les caractéristiques du téléphone dans le répertoire Sdk/skins de mon IDE android studio, puis je configure mon ADV en mode avancé pour l’utiliser

#### Utilisation d’une device virtuel

Dans le contexte de notre projet, nous utiliserons le Skin d’un galaxy\_nexus



#### Configuration de notre ADV (VM) et exécution

La création d’un ADV n’est pas compliquée. Il faut identifier les caractéristiques de notre smartphone cible (RAM, SD…) pour ensuite créer votre ADV.

|  |  |
| --- | --- |
| Une image contenant texte  Description générée automatiquement | Une image contenant texte, capture d’écran, frappant  Description générée automatiquement |

Une image contenant texte, moniteur, équipement électronique, intérieur

Description générée automatiquement

## Utilisation d’un device physique

L’utilisation d’un device physique nécessite tout d’abord de le passer en mode développeur son smartphone pour ensuite y accèder en debogage en USB voire WIFI.

### Passer en mode développeur

L’activation du mode développeur d’un smartphone dépend du constructeur.

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

### Activer le mode développeur puis connecté le smartphone

Il faut ensuite aller sur le mode setting (paramètre) puis « Option développement », activer le mode « Debogage USB » et brancher votre smartphone à votre PC sur lequel vous avez Androi Studio 😉.

|  |  |
| --- | --- |
|  |  |

L’utilisation du mode debugage permet d’accéder aux ressources du smartphone.

Une image contenant texte

Description générée automatiquement

Explorateur de fichier du smartphone connecté

Caractéristique du smartphone connecté

Smartphone connecté

Vous pouvez à présent utiliser votre smartphone pour déboguer votre application

### Comment debuger notre solution

Vous pouvez analyser votre application à l’aide du **logcat** et de **l’App inspection**.

https://youtu.be/y5CyYhpoiho

#### logcat

Logcat permet d’afficher tous les messages de log provenant du système voire des applications. Il faut sélectionner en bas de votre IDE **Logcat** puis votre **device**, sur lequel vous souhaitez faire fonctionner votre application, tous les log du smartphone s’affichent.

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

Une image contenant texte, capture d’écran, intérieur

Description générée automatiquement

#### Build

Pour construire votre application et contrôler sa compilation, l’onglet **Build** et l’action de l’icône représentant un marteau déclenche le processus de compilation.

Une image contenant texte

Description générée automatiquementUne image contenant texte

Description générée automatiquement

#### App Inspection

L’onglet **App Inspection** permet d’inspecter la base de données de votre application, des échanges réseaux et l’état des tâches en arrière-plan

Une image contenant texte

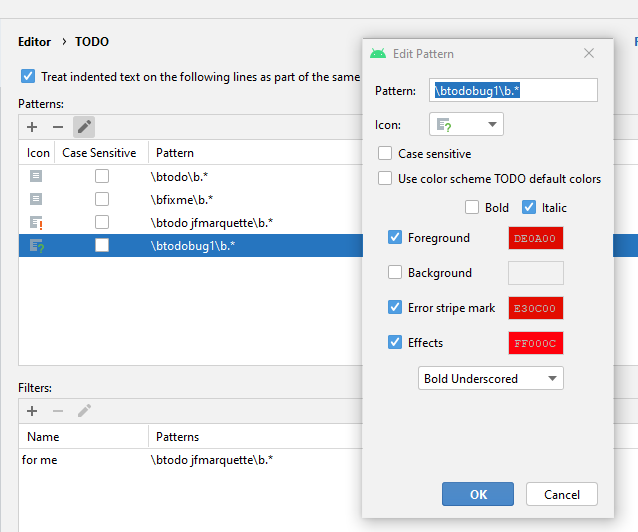
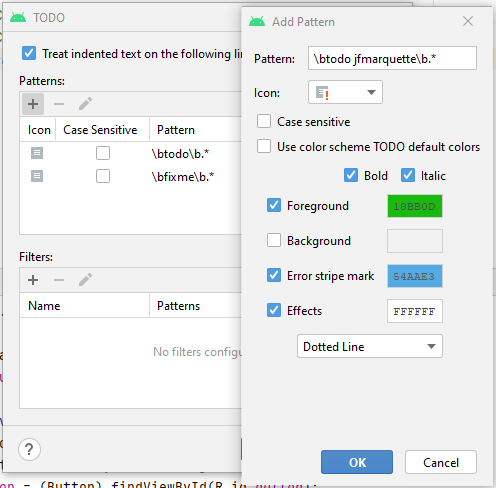
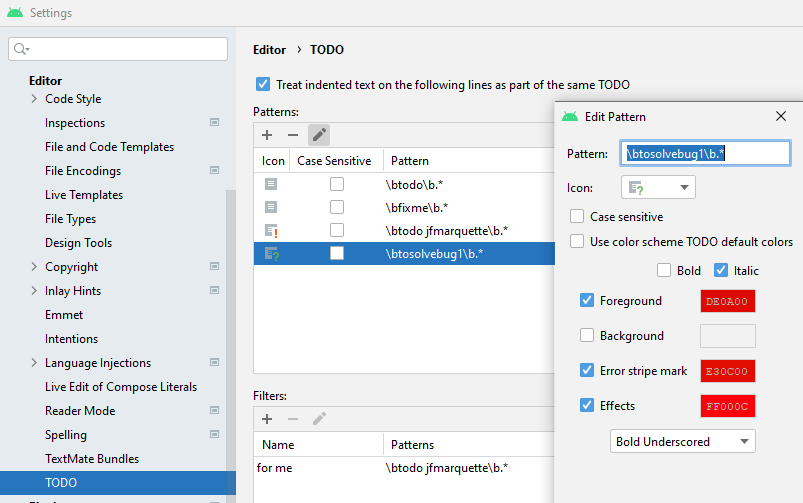
Description générée automatiquement

#### TODO

L’onglet **TODO** permet de dresser les actions à faire. Il est intéressant de configurer des Pattern en les personnalisant pour faciliter le processus de développement de votre équipe.

|  |  |
| --- | --- |
| **Pattern** | **Description** |
| \btodobug1\b.\* | Bug de priorité importante |
| \btodobug2\b.\* | Bug de priorité moyenne |
| \btodoevol1\b.\* | Evolution de priorité importante |
| \btodoevol2\b.\* | Evolution de priorité moyenne |
| \btodobug1 jfmarquette\b.\* | Bug de priorité importante attribué à jfmarquette |
| \btodobug2 jfmarquette \b.\* | Bug de priorité moyenne attribué à jfmarquette |
| \btodoevol1 jfmarquette \b.\* | Evolution de priorité importante attribué à jfmarquette |
| \btodoevol2 jfmarquette \b.\* | Evolution de priorité moyenne attribué à jfmarquette |

Pour configurer ces paternes, il faut ouvrir la fenetre des **setting** (**CTRL + ALT + S**) et les créer.

5 création

4 création

3 création

2 TODO

1 Editor

Les actions de développement à réaliser sont matérialisables via le filtre

Une image contenant texte

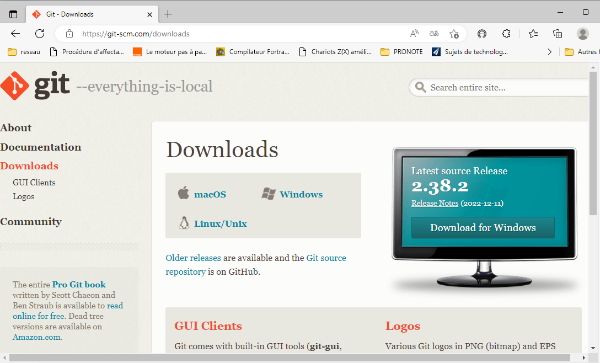
Description générée automatiquementUne image contenant texte

Description générée automatiquement

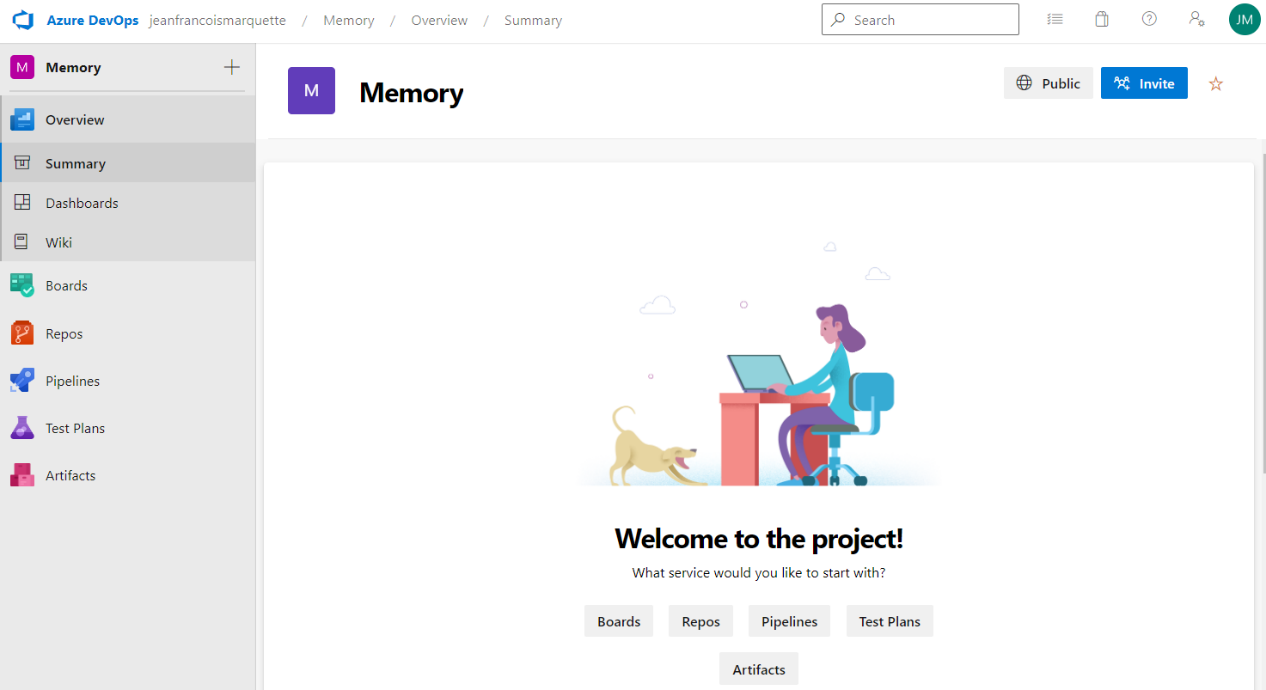
#### GIT (Gestion des versions)

##### Configuration Git Android Studio en relation avec DevOps Solution Microsoft

Nous allons gérer le versioning de note application à l’aide de la solution Git de la solution DevOps de Microsoft. Si **GIT n’est pas installé**, il faut l’installer **via ce site :** [Git - Downloads (git-scm.com)](https://git-scm.com/downloads)

Après l’installation, il faut se positionner dans le répertoire du projet à synchroniser puis utiliser la commande « git init » pour activer le versioning.

Une image contenant texte

Description générée automatiquementDans votre solution DevOps (ex. : <https://jeanfrancoismarquette.visualstudio.com/>), créez votre projet « memory » en cliquant sur « Repos » puis compléter les information et cliquer sur « Create » .

Pour accéder à ce Repos, il faut créer un token d’accès à l’aide du menu « 1 connexion » « 2 personal access tokens », puis New Token

Une image contenant texte

Description générée automatiquement

1

3

2

Retourner dans le répertoire projet (onglet Terminal) et utilisez la commande :  
**« git add origin** [**https://jeanfrancoismarquette.visualstudio.com/Memory/\_git/Memory**](https://jeanfrancoismarquette.visualstudio.com/Memory/_git/Memory)**»**

Une image contenant texte

Description générée automatiquementPour vérifier que le dépôt est correctement paramétré, utilisez la commande « git remote -v »

Vous venez de configurer GIT assurant le versioning de votre projet. Dès que vous accédez à l’onglet Git

Espace DevOps

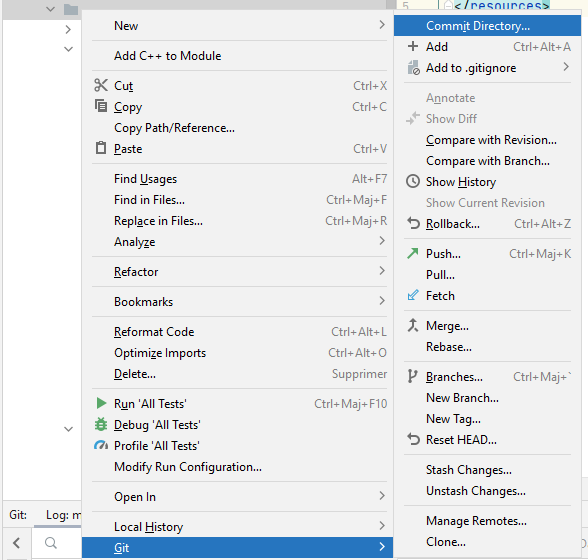
##### Utilisation de Git Android Studio

L’onglet Git permet d’accéder au nœud racine sur le serveur (remote) et sur le local, de visualiser l’évolution des versions de notre jeu Memory.

Une image contenant texte

Description générée automatiquement

Si nous souhaitons « valider» (commiter) nos modifications, il faut réaliser un click gauche (1) sur le répertoire de notre projet, puis sélectionner Git (2) et enfin « Commit Directory (3).

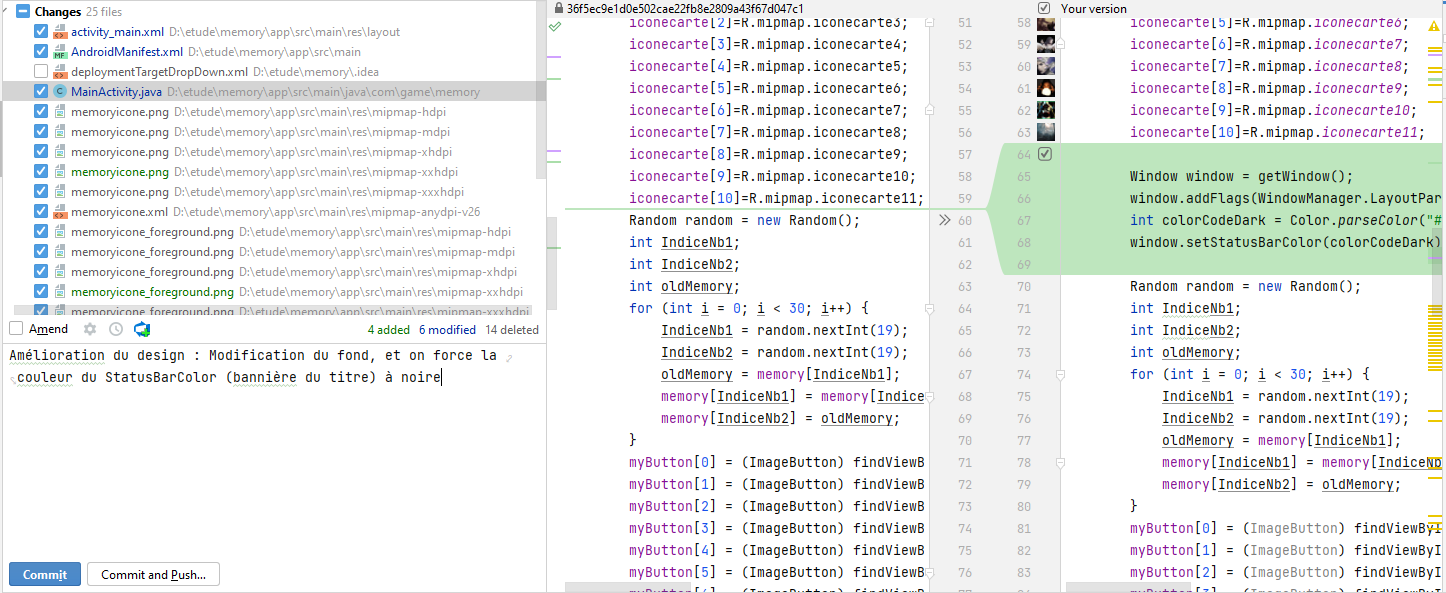


3

2

1

Automatiquement tous les fichiers modifiés sont listés (zone 1). En double cliquant sur un fichier source, il est possible d’obtenir la différence avec la version précédente (Zone2). Après vérification, il est possible de mettre un commentaire (Zone 3) et de valider par « commit + push ».



Nous observons après le push que le repository est à jour.

Une image contenant texte

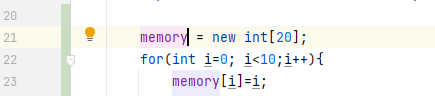
Description générée automatiquement

Sur la solution DevOps, il est également possible de visualiser les modifications.

##### Debuguer une application

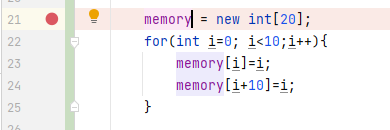
Une image contenant texte, moniteur, équipement électronique, téléphone

Description générée automatiquementPour débuger l’application, vous devez sur la première partie de votre application (voir vidéo).

Une image contenant texte

Description générée automatiquement

Positionner le curseur sur la ligne 21   
Puis activer un point d’arrêt à l’aide des touches CTRL + F8

Un rond rouge se positionne dans la colonne indiquant qu’un break point est inséré à la ligne 21.

Pour supprimer ce break point, il suffit de réitérer CTRL+F8. Pour lancer l’application en mode Debug, il faut MAJ + F9 ou cliquer sur le bouton 

Une image contenant texte

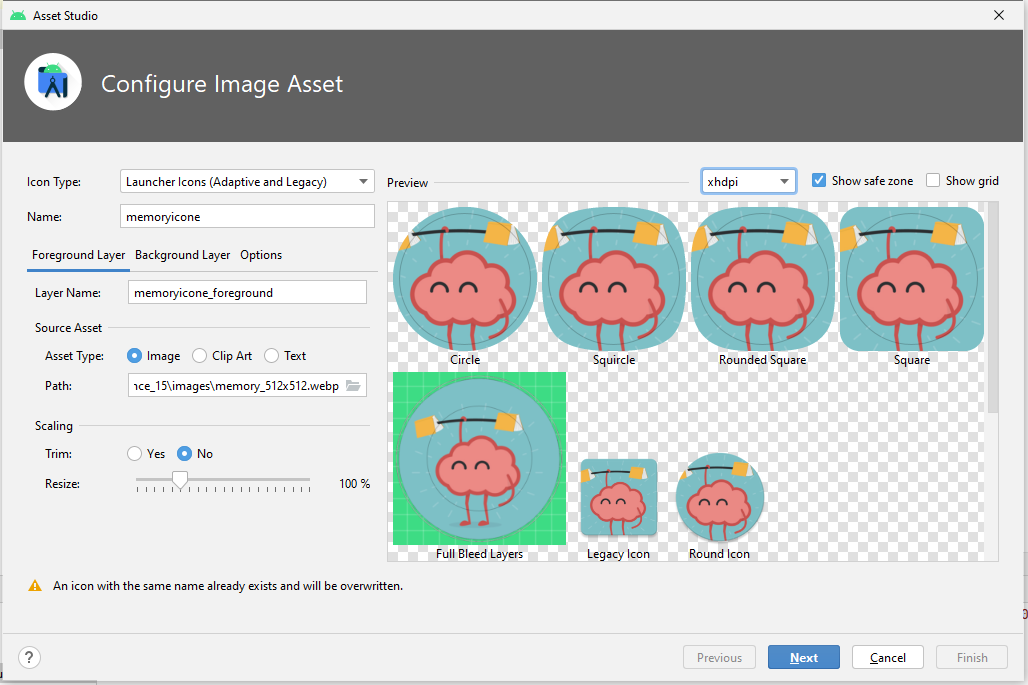
Description générée automatiquement

Puis pour dérouler le programme, il faut activer la touche F7

Raccourcis à connaître :



|  |  |  |  |
| --- | --- | --- | --- |
|  | F8 | Step over | Quand le debug est sur une fonction, il ne rentrera pas dans la fonction et restera dans la fonction mère. |
|  | F7 | Step into | Quand le debug est sur une fonction, il rentrera dans la fonction. |
|  | ALT+F9 | Run to cusor | Exécute le code jusqu-au curseur |
|  | MAJ + F8 | Step out | Sort de la fonction |



https://wallup.net/41209-2/

