

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique  
Université Mouloud MAMMERRI de Tizi-Ouzou  
Faculté Génie Electrique et Informatique  
Département Informatique



# MEMOIRE de fin d'étude

En vue de l'obtention du diplôme de Master en INFORMATIQUE

« Conduite de Projet Informatique »

## THEME :

Développement D'une Application ANDROID Offline  
Pour L'apprentissage de la Langue Française

Encadrer par :

M<sup>me</sup> G. Bourkache

Réalisé par :

- M<sup>elle</sup> Abderrahmani Sihem

- M<sup>elle</sup> AMIRECHE Célia

PROMOTION: 2014-2015



# Dédicaces

*Je dédie ce modeste travail à :*

*A mes chers parents (Hamza & Myassa) qui m'ont beaucoup aidé et soutenu durant ma vie et surtout dans mes études .Je prie dieu le tout puissant de les récompenser de toutes les peines et sacrifices données aux quels je ne rendrai jamais assez.*

*A ma très chère grand-mère.*

*A mes chers frères*

*A mes adorables sœurs et leur famille.*

*A tous mes amis(es) en exception M<sup>r</sup> Mezaguer.*

*A mon binôme Sihem.*

*A toute ma famille.*

*Gélia Amiréche.*





# *Dédicaces*

*Je dédie ce modeste travail à :*

*A mes chers parents qui m'ont beaucoup aidé et soutenu durant ma vie et surtout dans mes études et leur compréhension et leur soutien . Je prie dieu le tout puissant de les récompenser de toutes les peines et sacrifices données aux quels je ne rendrai jamais assez.*

*A mes très chers frères.*

*A tous mes amis(es).*

*A mon binôme Célia.*

*A toute ma famille.*

*A mon très cher mari.*

*Sihem Abderrahmani.*





# Remerciements

*Nous remercions le bon Dieu pour le courage et la patience qui nous ont été utiles tout au long de notre parcours.*

*Nous tenons à exprimer notre profonde gratitude à notre promotrice : Madame Bourkache Pour avoir accepté de nous encadrer et de nous orienter durant ce projet.*

*Nous remercions également les membres du jury qui nous en fait honneur de juger notre travail.*

*Nous tenons à remercier nos enseignants qui nous ont transmis le savoir et le bien.*

*Ces remerciements ne seraient pas complets si on n'a pas pensé à les destiner, avec notre profonde reconnaissance, à nos parents qui nous ont offert un environnement favorable mener à terme notre travail.*

*Enfin nous tenons à remercier tous ceux qui nous ont aidé et contribué de près et de loin à la réalisation de notre projet.*



# SOMMAIRE

Introduction générale.....	1
----------------------------	---

## Chapitre I : Système d'exploitation android

I. Introduction .....	2
II. Définition .....	2
III. Historique .....	3
III.1 Naissance d'android .....	3
III.2 Version d'android .....	4
III.3 Répartition des différentes versions sur le Google Play Store .....	5
IV. Fonctionnalités du système android .....	7
V. Architecture du système android .....	8
V.1 Applications.....	9
V.2 Application framework .....	9
V.3 Bibliothèque .....	10
V.4 Le moteur d'exécution android.....	10
V.5 Noyau linux .....	11
VI. Avantages et inconvénient d'android .....	11
VI.1 Avantages .....	11
VI.2 Inconvénients.....	12
VII. Conclusion .....	13

## Chapitre II : Langages de programmation

I. Introduction .....	14
II. Généralité .....	14
II.1. Présentation de langage java.....	14
II.2. Les outils nécessaires pour développer en java .....	15
II.3. Les versions de java.....	16
II.4. Les caractéristiques de java .....	17
III. Quelques packages de base .....	17
IV. Java sous Android .....	18
IV.1. L'IDE Eclipse .....	18
IV.2. Android Developer Tools .....	18
IV.3. Le SDK Android .....	18
V. Application Android .....	18
V.1. Types d'applications.....	18
V.2. Cycle de vie d'une application Android.....	19
VI. Les principaux composants d'une application .....	20
VII. Conclusion .....	23

### Chapitre III : Analyse et conception

I.	Introduction.....	24
II.	Objectif de l'application.....	24
III.	Présentation de l'UML.....	24
IV.	Processus de développement.....	25
IV.1.	Analyse.....	25
a)	Identification des acteurs.....	25
b)	Présentation de l'application.....	26
c)	Les cas d'utilisation.....	27
IV.2.	La conception.....	32
1.	Le niveau applicatif.....	32
2.	Diagramme d'activité.....	34
3.	Diagramme des classes.....	40
4.	Diagramme logique.....	43
V.	Conclusion.....	43

### Chapitre IV : Réalisation

I.	Introduction.....	44
II.	Outils nécessaires au développement.....	44
III.	Création d'un AVD.....	46
IV.	Création d'un nouveau projet.....	49
V.	Stockage de données.....	52
VI.	Le mode de fonctionnement de l'application.....	55
VII.	Conclusion.....	65
	Conclusion général.....	66
	Bibliographie.	

## TABLE DES FIGURES

<b>Figure 1 : Logo officiel du système d'exploitation Android.....</b>	<b>2</b>
<b>Figure 2 : Le nombre d'appareils Android connectés au Google Play Store.....</b>	<b>7</b>
<b>Figure 3 : Composants de la plate-forme Android .....</b>	<b>9</b>
<b>Figure 4: Les taux d'utilisation des environnements de développement Java .....</b>	<b>16</b>
<b>Figure 5 : Les différentes versions de Java.....</b>	<b>17</b>
<b>Figure 6: Menu d'applications Android.....</b>	<b>19</b>
<b>Figure7: Exemple d'un fichier Manifest.....</b>	<b>21</b>
<b>Figure 8: modélisation simple de l'application.....</b>	<b>26</b>
<b>Figure 9: Cycle de vie d'une activité.....</b>	<b>34</b>
<b>Figure 10 : diagramme d'activités des cas d'utilisation « suivre un cours» .....</b>	<b>35</b>
<b>Figure 11: diagramme d'activités des cas d'utilisation « effectuer un test » .....</b>	<b>36</b>
<b>Figure 12: diagramme d'activités des cas d'utilisation « lire un poème».....</b>	<b>37</b>
<b>Figure 13: diagramme d'activités des cas d'utilisation « Affiche un proverbe ».....</b>	<b>38</b>
<b>Figure 14: diagramme d'activités des cas d'utilisation « Afficher une citation».....</b>	<b>39</b>
<b>Figure 15 : diagramme de classe « suivre un cours » .....</b>	<b>40</b>
<b>Figure 16 : diagramme de classe « effectuer un test » .....</b>	<b>41</b>
<b>Figure 17 : diagramme de classe « effectuer un divertissement ».....</b>	<b>42</b>
<b>Figure 18 : modèle de la table de données score.....</b>	<b>43</b>
<b>Figure 19 : fenêtre de lancement d'Eclipse.....</b>	<b>45</b>
<b>Figure 20 : capture d'écran de la fenêtre d'android SDK Manager.....</b>	<b>45</b>
<b>Figure 21 : Capture d'écran de la fenêtre de validation de téléchargements .....</b>	<b>46</b>
<b>Figure 22 : Capture d'écran de la fenêtre Android Virtual Device Manager. ....</b>	<b>47</b>

<b>Figure 23 : Capture d'écran de la fenêtre « Create new AVD » .....</b>	<b>48</b>
<b>Figure 24: Capture d'écran de la fenêtre d'un AVD. ....</b>	<b>49</b>
<b>Figure 25 : Capture d'écran de la fenêtre Création d'un nouveau projet .....</b>	<b>50</b>
<b>Figure 26: Capture d'écran de la dernière fenêtre de création d'un nouveau projet ....</b>	<b>51</b>
<b>Figure 27: structure d'un projet Android. ....</b>	<b>52</b>
<b>Figure 28 : Modèle de la table de données score .....</b>	<b>52</b>
<b>Figure 29 : capture d'écran d'un fichier « .xml » .....</b>	<b>53</b>
<b>Figure 30 : Capture d'écran d'un code « xml » .....</b>	<b>54</b>
<b>Figure 31 : capture d'écran d'un code Java .....</b>	<b>54</b>
<b>Figure 32: l'icône de l'application dans le Menu de principal .....</b>	<b>55</b>
<b>Figure 33: Fenêtre principale de l'application .....</b>	<b>56</b>
<b>Figure 34: Menu principal de l'application .....</b>	<b>56</b>
<b>Figure 35: Le sommaire des cours .....</b>	<b>57</b>
<b>Figure 36: Exemple d'un cours .....</b>	<b>58</b>
<b>Figure 37: Menu des tests .....</b>	<b>58</b>
<b>Figure 38: Exemple d'une question .....</b>	<b>59</b>
<b>Figure 39: Affichage de score .....</b>	<b>60</b>
<b>Figure 40: Fenêtre « Voir tous les scores » .....</b>	<b>61</b>
<b>Figure 41: Menu divertissement .....</b>	<b>61</b>
<b>Figure 42: Sommaire des poèmes .....</b>	<b>62</b>
<b>Figure 43: Exemple d'un poème .....</b>	<b>63</b>
<b>Figure 44: Menu proverbes .....</b>	<b>63</b>
<b>Figure 45: Exemple de proverbes .....</b>	<b>64</b>
<b>Figure 46: Fenêtre citations .....</b>	<b>65</b>

## *INTRODUCTION GENERALE*

Il y a 10 ans avant, le téléphone mobile était juste un outil de communication pour émettre et recevoir des appels et des messages à partir d'autres mobiles.

Maintenant, avec l'arrivée des nouvelles générations (3eme, 4eme) le terminal mobile a retrouvé un nouvel aspect, il n'est plus ce téléphone simple qui 'existait avant.

A présent, on ne parle plus de téléphone mais de Smartphone, ce qui est plus juste grâce à son intelligence artificielle et un certain degré d'autonomie dont il est doté. Les

Smartphones permettent l'exécution d'une panoplie d'applications et offrent beaucoup de services. Il est un véritable ordinateur de poche qui permet une connexion avec un ordinateur pour stocker des données, connexion internet, un appareil relativement pratique grâce à son écran tactile avec, bien sûr, toutes les fonctionnalités d'un téléphone portable : enregistrement photo, vidéos, sons, envoi et réception de MMS, baladeur MP3 et radio, appareil photo et caméra vidéo.

Plusieurs types d'applications sont développées selon les plateformes propres aux Smartphones, parmi lesquelles on évoque « Android ». Android comprend un système d'exploitation basé sur Linux, une interface graphique, un navigateur Web et des applications qui peuvent être téléchargées par l'utilisateur.

L'extension technologique apportée au Smartphone fait de ce dernier un outil indispensable dans notre vie quotidienne. Par conséquent, nous nous éloignons de plus en plus des méthodes d'éducatons classiques. En se basant sur cet embarras, nous avons opté pour rassembler entre l'essor technologique que les téléphones ont vus, et les anciennes techniques d'enseignements, ainsi nous avons développé une application Android destinée à l'apprentissage.

Notre application à un aspect pédagogique, elle fournit d'une part, un moyen interactif pour améliorer son niveau linguistique en langue française grâce aux cours résumés et aux tests de différents niveaux qu'elle offre, d'autre part un univers de divertissement et d'enrichissement de connaissances avec des poèmes et des proverbes ainsi que des citations.

Pour vous mettre dans cadre de réalisation de notre application, nous avons procédé ainsi :

D'abord, puisque notre applications fonctionne sous le système d'exploitation Android, nous avons présenté une vue approfondie sur ce dernier en analysant son architecture interne et ses versions, les avantages et les inconvénients dans le premier chapitre.

Ensuite, comme Android est basé sur le système d'exploitation Linux, et l'ensemble de ses applications sera écrit en Java, nous avons consacré le deuxième chapitre pour détailler l'environnement de développement Java et les différents outils nécessaires pour l'obtenir.

En fin, dans les deux derniers chapitres, on a fait une description détaillée de processus de développement suivie lors de sa conception et réalisation, et une présentation globale de l'application en décrivant ses cas d'utilisations et son mode de fonctionnement.

## *I. Introduction :*

Android est un system d'exploitation pour Smartphones qui était initialement une startup (jeune entreprise) qui développa un système d'exploitation pour applications mobiles, racheté par Google en 2005.

Le 5 novembre 2007 Google annonça la création de l'OHA (Open Handset alliance) qui est un consortium réunissant 34 entreprises dans le but de favoriser l'innovation de l'application mobile en fournissant une plateforme véritablement ouverte et complète.

Une semaine plus tard ils annoncent la sortie du premier SDK (Software Développement Kit) permettant la création des applications sous la plateforme Android.

Cependant la principale différence d'Android face aux autres solutions est qu'il soit Open source. Cette particularité le rend donc gratuit et personnalisable par les constructeurs et les opérateurs de téléphonie mobile. Malgré des personnalisations accrues en terme d'interface utilisateur et de fonctionnalité, le cœur du système reste commun ce qui permet une interopérabilité des applications sans prendre en compte les difficultés liées aux différences matérielles des périphériques.

## *II. Définition :*

- ✚ Android est un système d'exploitation Open Source édité par Google pour appareils embarqués et/ou mobiles, comme les Smartphones ou les tablettes. On le retrouve aussi dans certains GPS, ordinateurs de bord, dans les téléviseurs, autoradios, et même des montres.
- ✚ Le personnage nommé Bugdroid est le petit robot vert utilisé par Google pour présenter Android. Ce personnage est sous licence «creative commons (3.0)» et peut donc être utilisé librement.

Le site En gadget annonce que Bugdroid, le logo d'android serait en fait un personnage d'un jeu des années 1990 sur Atari : Gauntlet : The Third Encounter.



**Fig.1: Logo officiel du système d'exploitation Android [01]**

- ✚ Le système Android est basé sur un « fork », un mécanisme de développement parallèle résultant de l'open source du noyau de Linux. Ce dernier a été modifié pour être plus adapté aux terminaux mobiles ayant peu de puissance de calcul, de mémoire et de batterie. De fait, certaines bibliothèques standards ne sont pas supportées par le système, et des améliorations ont été apportées pour une gestion optimale de l'énergie.
- ✚ Les applications sont écrites en Java, et fonctionnent grâce à une machine virtuelle Dalvik. Cette machine virtuelle a elle aussi été modifiée pour être la plus adaptée possible aux appareils de faible puissance. Ainsi, beaucoup d'efforts ont été fait sur la consommation de mémoire, qui a été largement diminuée par rapport à la machine virtuelle java classique. [02]

### *III. Historiques:*

#### **III.1. la naissance d'ANDROID**

Android fait immédiatement penser à Google, ce qui n'est pas faux à l'heure actuelle, mais cette multinationale n'est pas à l'initiative de projet. D'ailleurs, elle n'est même pas la seule à contribuer à plein temps à son évolution. À l'origine, Android était le nom d'une PME américaine, créée en 2003 puis rachetée par Google en 2005, qui avait l'intention de s'introduire sur le marché des produits mobiles.

Après le dévoilement de l'iPhone en 2007 par Apple, les constructeurs mobiles devraient s'aligner sur cette nouvelle création, ce qui demandait de longues années de recherches. Ceci a mené à la création de l'Open Handset Alliance en novembre 2007, et qui comptait à sa création 35 entreprises évoluant dans l'univers du mobile, dont Google. Cette alliance a pour but de développer un système open source (c'est-à-dire dont les sources sont disponibles librement sur internet) pour l'exploitation sur mobile et ainsi concurrencer les systèmes propriétaires, par exemple Windows Mobile et Ios. Cette alliance a pour logiciel vedette Android.

La possibilité de télécharger le code source du système Android, de le compiler, de l'installer, de l'exécuter et de le modifier est donnée au développeur à partir du 21 octobre 2008 quand Google et OHA annoncèrent la mise en disposition du code source de la plateforme Android en open source sous licence Apache 2.0.

Cette dernière possibilité à engendrer la création de diverses applications, et pour cela en novembre 2008 fut lancé Android Market, le magasin d'applications de Google pour proposer divers applications aux utilisateurs. Dès le 2 février 2011 il est devenu possible de télécharger, acheter et installer sur les appareils Android ces différentes applications.


Le premier smartphone Android (HTC Dream G1) est sorti en octobre 2008 aux Etat Unis, et en mars 2009 en France.


En mai 2009 apparait les premières tablettes tactiles Android, mais elles n'étaient utilisables qu'après la sortie de la version Android 3.0 spécialement conçue pour tablettes en 2011.


### III.2. Les version d'ANDROID


La progression des fonctionnalités, amélioration et correction de bogues, ont fait l'apparition de plusieurs versions d'Android de plus en plus sophistiquées, parmi lesquelles on cite les versions majeures suivantes :


Les différentes versions d'Android ont toutes des noms de desserts depuis la sortie de la version 1.5 et suivent une logique alphabétique (de A vers Z)


- 


la version 1.0 sortie le 23 septembre 2008, caractérisée par une version connue uniquement par des développeurs car c'est la version du SDK distribuée avant la sortie du premier téléphone Android, nommée Apple pie.
- 


la version 1.1 sortie le 9 février 2009, caractérisée par la version Beta incluse dans le premier téléphone le HTC G1/Dream, nommée Bananas split.
- 


la version 1.5 sortie le 30 avril 2009, caractérisée par de nouvelles fonctionnalités et la mise à jour de l'interface graphique, nommée Cupcake, basée sur le noyau Linux2.6.27.
- 

la version 1.6 sortie le 15 septembre 2009, caractérisée par de nouvelles fonctionnalités et la mise à jour de l'interface graphique, nommée Donut, basée sur le noyau Linux2.6.29.
- 






la version 2.0 sortie le 26 octobre 2009, caractérisée par de nouvelles fonctionnalités et la mise à jour de l'interface graphique, nommée Eclair basée sur le noyau Linux2.6.29.
- 

la version 2.2.x sortie le 20 mai 2010, caractérisée par une vitesse améliorée, nouvelles fonctionnalités et la mise à jour de l'interface graphique, nommée Gingerbread basée sur le noyau 2.6.35.
- 

Android 2.3 (Gingerbread, 6 décembre 2010)
- 

la version 3.x.x sortie le 22 février 2011, réservée aux tablettes tactiles et aux téléviseurs connectés, cette mise à jour comprend de nombreux changements dans l'interface, nommée Honeycomb.
- 

la version 4.0.x sortie le 19 octobre 2011, est fortement inspirée d'Honeycomb, unifiée pour Smartphones, tablettes et Google tv apporte de nombreux changements, nommée Ice Cream Sandwich, basée sur le noyau 3.0.1.

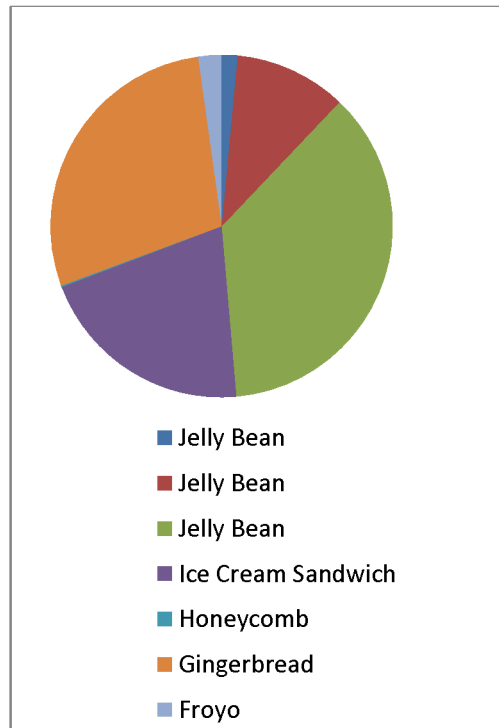
-  la version 4.1.x sortie le 9 juillet 2012, elle ajoute un système de notification améliorée, la reconnaissance vocale sans connexion internet et le projet Butté qui augmente la fluidité d'Android, nommée Jelly Bean, bases sur le noyau 3.0.31.
-  la version 4.2.x sortie le 13 novembre 2012, caractérisée par une nouvelle interface de l'appareil photo, et par un système multi-compte uniquement sur tablette et de type Gesture permettent d'écrire avec le clavier rien qu'en glissant le doigt, nommée Jelly Bean.
-  la version 4.3.x sortie le 24 juillet 2013, caractérisée par un support de Bluetooth smart, basse consommation et ajout de la norme AVRPC 1.3, gestion multiutilisateur plus poussée, support d'Open ES 3.0, nouvelle interface de l'appareil photo, mise à jour de sécurité et Slimport, nommée Jelly Bean.
-  la version 4.4.x sortie le 3 septembre 2013, caractérisée par un support de Bluetooth 4.0, consommation on ressource moins élevée nécessite moins de RAM, nouvelles icônes plus soignées, la barre de statut est maintenant transparente sur certain menus et changera de couleurs en fonction du contenu affiche, la barre en bas de l'écran est désormais translucide sur certains menus, nommée Kit Kat.
-  Android Lollipop est la version 5.0 du système d'exploitation mobile Android développé par Google. Son numéro de version et son nom de code ont été annoncés le 15 octobre 2014, pour une disponibilité publique le 3 novembre 2014, avec la sortie des Nexus 6, 9 et Player.

Les changements les plus importants d'Android 5.0 sont sa disponibilité sur les nouvelles plateformes Android TV et Android Auto, le renouveau de l'interface utilisateur nommée Material Design, l'amélioration du système de notifications, l'environnement d'exécution ART (en) qui remplace officiellement Dalvik pour améliorer les performances, ainsi que l'amélioration de l'autonomie de la batterie via le projet Volta

### III.3. Répartition des différentes versions sur le Google Play Store

Basée sur le nombre d'appareils Android qui se sont connectés au Google Play Store dans une période de 14 jours se terminant à la date de collecte des données.

Version	Nom de code	Date de sortie	API level	%
4.3.x	Jelly Bean	24 juillet 2013	18	1,5%
4.2.x	Jelly Bean	13 novembre 2012	17	10.6 %
4.1.x	Jelly Bean	9 juillet 2012	16	36.5 %
4.0.x	Ice Cream Sandwich	19 octobre 2011	14-15	20.6 %
3.x.x	Honeycomb	22 février 2011	11-13	0,1 %
2.3.x	Gingerbread	6 décembre 2010	9-10	28.5 %
2.2.x	Froyo	20 mai 2010	8	2.2 %



**Fig.2 : graphe représentant le nombre d'appareils Android qui se sont connectés au Google Play Store**

#### ***IV. Fonctionnalités du système Android :***

Android a été conçu pour intégrer au mieux les applications existantes de Google comme le service de courrier Gmail, l'agenda Google calendar ou encore la cartographie Google Maps. Voici quelques fonctionnalités proposées par Android classées par version :

##### **Android version 1.5 (Cupcake) :**

- ❖ Enregistrement et lecture des vidéos.
- ❖ Mise en ligne directe des vidéos sur YouTube.
- ❖ Mise en ligne directe des photos picasa.
- ❖ Prise en charge du Bluetooth A2DP.
- ❖ Dossiers dynamiques et widgets pour le home.
- ❖ Copier/coller étendu aux pages web.
- ❖ Nouvelle version du clavier virtuel.

**Android version 1.6 (Donut) :**

- ❖ L'application Galerie permet d'effacer plusieurs photos à la fois.
- ❖ Amélioration de l'Android Market.
- ❖ Amélioration de la vitesse de la recherche vocale et intégration étendue à plus d'applications natives.
- ❖ Prise en charge sur une seule application de la prise de photo et de l'enregistrement vidéo.
- ❖ Possibilité de recherche simultanément dans les favoris, les historiques, les contacts et sur Google depuis le home via widget recherche.
- ❖ Moteur Text-to-speech.
- ❖ Prise en charge de plusieurs résolutions d'écran.

**Android version 2.0/2.1 (Eclair) :**

- ❖ Interface utilisateur revue (lockscreen et lanceur d'application).
- ❖ Fonds d'écran animés.
- ❖ New browser interface avec prise en charge du HTML5.
- ❖ Prise en charge du protocole Microsoft Exchange.
- ❖ New contact lists.
- ❖ Prise en charge du Bluetooth 2.1.
- ❖ Amélioration du clavier virtuel.
- ❖ Prise en charge en natif du flash et du zoom numérique pour des appareils photos.
- ❖ Amélioration du ratio blanc/noir sur les fonds.
- ❖ Gestion multi-comptes Gmail et ajout de la synchronisation avec Facebook.
- ❖ Augmentation de la performance et de la vitesse.
- ❖ Fonctionnalité de Hot spot Wifi.
- ❖ Partage de contact sur Bluetooth.
- ❖ Mise à jour automatique des applications.

**V. Architecture du système ANDROID :**

Le système d'exploitation Android peut être résumé dans le diagramme suivant, qui illustre les composants principaux de chaque section décrite ci-dessous:

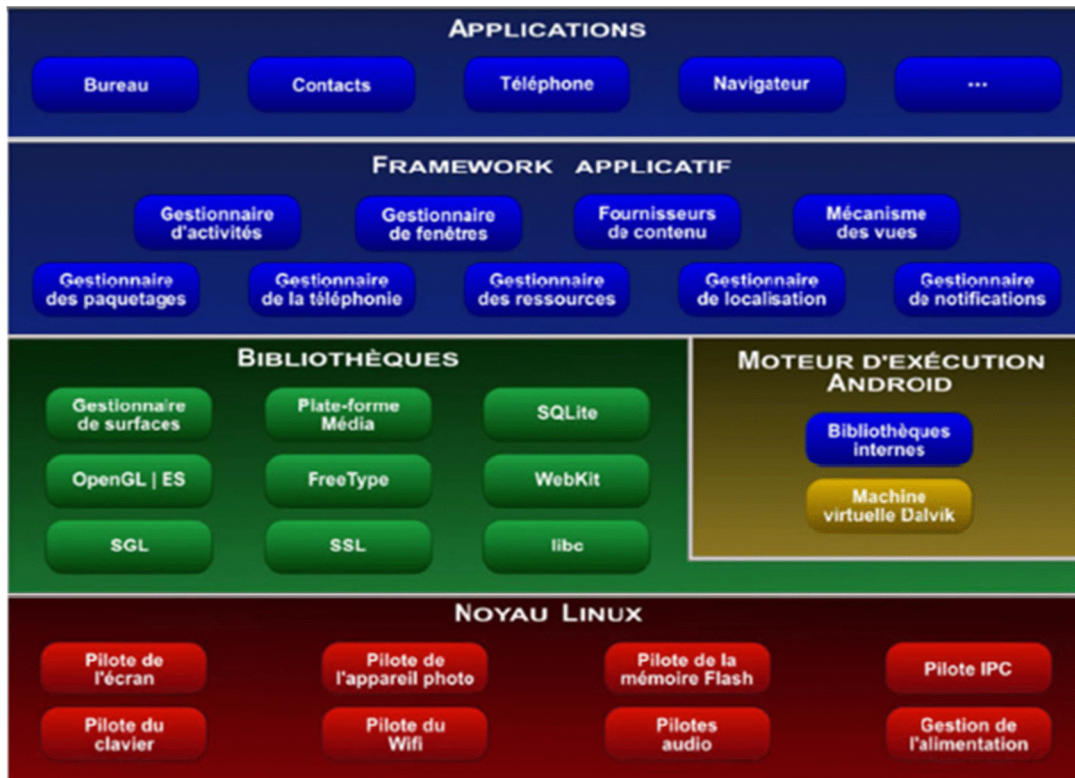


Fig.3 : Composants de la plate-forme Android [03]

## V.1. Applications

Android est livré accompagné d'un ensemble d'applications noyau incluant :

- + Un client mail,
- + Un programme pour les SMS,
- + Un calendrier,
- + Des cartes,
- + Un navigateur,
- + Des contacts personnels, voire d'autres sous-systèmes.

Toutes les applications sont écrites en utilisant le langage de programmation Java.

## V.2. Application Framework

Cette couche permet au programmeur de construire de nouvelles applications. Toute application est un ensemble de services et systèmes incluant :

- + Un ensemble riche et extensible de mécanismes de vues pouvant être utilisées pour créer l'application, incluant listes, grilles, cases à cocher, boutons, et même un navigateur web embarqué.

- ✚ Des fournisseurs « content providers » qui permettent aux applications d'accéder aux données d'autres applications (comme les contacts), ou de partager leurs propres données.
- ✚ Un Gestionnaire de Ressources qui donne accès aux ressources telles que des images et des fichiers « Layout » (disposition).
- ✚ Un Gestionnaire de Notification qui permet aux applications d'afficher des alertes personnalisées dans la barre de statuts.
- ✚ Un Gestionnaire d'Activités qui gère le cycle de vie des applications et propose une navigation commune.

### V.3. Librairies(Bibliothèques)

Android inclut un ensemble de bibliothèques C/C++ utilisées par de nombreux composants du système Android. Ces capacités sont proposées au développeur par le biais du Framework d'application d'Android. La plupart des bibliothèques noyau sont listées ci-dessous:

- ✚ La bibliothèque standard du système C adaptée pour les appareils fonctionnant avec le système Linux embarqué
- ✚ Les bibliothèques media responsables de la lecture et de l'enregistrement des données audio et vidéo sous plusieurs formats
- ✚ « *Surface Manager* » gère l'accès au sous-système d'affichage et aux couches graphiques 2D (2 dimensions) et 3D de multiples applications.
- ✚ « *SGL* » le moteur graphique 2D
- ✚ Les bibliothèques graphiques 3D basées sur « OpenGL » souvent utilisées pour développer des jeux 3D puissants.
- ✚ Des polices de caractères
- ✚ « *SQLite* » un puissant et léger moteur de base de données relationnel disponible pour toutes les applications, utilisé afin de gérer le stockage et la recherche des données. Son architecture simple et sa taille peu volumineuse le rendent parfaitement adapté aux téléphones mobiles ayant des ressources limitées.

### V.4. Le Moteur d'Exécution Android

Android inclut un ensemble de bibliothèques de base offrant la plupart des fonctionnalités disponibles dans les bibliothèques de base du langage de programmation Java.

Chaque application Android s'exécute dans son propre processus, avec son propre instance de la machine virtuelle Dalvik. Dalvik a été écrite pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. Elle exécute des fichiers dans l'exécutable Dalvik (.DEX), un format optimisé pour ne pas encombrer la mémoire. La machine virtuelle est la base de registres et fonctionne grâce aux classes compilées par un compilateur Java et transformées dans le format DEX.

La machine virtuelle Dalvik s'appuie sur le noyau Linux pour les fonctionnalités de base telles que le filetage et la gestion de la mémoire de bas niveau.

## V.5. Noyau Linux

Le noyau Linux version 2.6 est la couche la plus basse d'Android. Cette couche est le niveau d'abstraction matériel qui permet l'interaction des couches supérieures avec la couche matérielle à travers les pilotes des périphériques. Le noyau Linux fournit également les services les plus fondamentaux tels que la sécurité, la gestion de la mémoire, la gestion des processus ainsi que les fonctionnalités de communication réseau.

Linux est réputé pour sa gestion stable et performante de la mémoire et des processus, ainsi que par son modèle de sécurité robuste basé sur un contrôle d'accès discrétionnaire (DAC) qui n'a pas changé depuis les années soixante-dix(70).

Le noyau linux fournit un système de driver permettant une abstraction avec le matériel. Il permet également le partage de bibliothèques entre différents processus, le chargement et le déchargement de modules à chaud.

Le noyau linux est entièrement open source et il y a une communauté de développeurs qui l'améliore et rajoute des divers.

## VI. Avantages et Inconvénients d'ANDROID :

### VI.1. Avantages [03] [04]

Les développeurs s'orientent souvent vers le système d'exploitation Android pour créer leurs applications mobiles, grâce à ces nombreux avantages, à savoir:

#### **Open source**

Le contrat de licence Android respecte les principes de l'« open source », ce qui permet à n'importe qui de lancer sa propre version d'Android. De nombreuses personnes utilisent ainsi des "ROM custom", c'est à dire des versions modifiées par rapport au code de base fourni par l'éditeur.

#### **Simplicité**

Android s'est imposé par sa simplicité d'utilisation, sa souplesse, pas besoin d'être un connaisseur expérimenté pour l'utiliser.

Par exemple : pour transférer des fichiers, de la musique sur son Smartphone Android, il suffit de brancher l'appareil sur son ordinateur. Comme avec une clé USB. Aucun autre logiciel n'est nécessaire.

#### **Facilité de développement**

Toutes les API (Application Programming Interface) mises à disposition facilitent et accélèrent grandement le travail du développeur. Ces APIs sont très complètes et très faciles d'accès.

### Facilité de commercialisation

Le Play Store (anciennement Android Market) est une plateforme immense et très visitée, c'est une mine d'opportunités pour les développeurs ayant des idées originales et utiles.

### Flexibilité





Le système est extrêmement portable, il s'adapte à beaucoup de structures différentes, les Smartphones, les tablettes, la présence ou l'absence de clavier ou de *trackball*, et différents processeurs etc. On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android. En plus d'avoir autant d'opportunités, Android est construit de manière à faciliter le développement et la distribution en fonction des composants en présence dans le terminal (si une application nécessite d'utiliser le Bluetooth, seuls les terminaux équipés de Bluetooth pourront la voir sur le Play Store).

### Ingéniosité

L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leurs développements. Ces applications se trouvent essentiellement sur internet et leur principe est que vous pouvez combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS (système de positionnement global), on peut poster les coordonnées GPS des photos prises.

## VI.2. Inconvénients [03] [04]

Malgré le succès qu'a connu Android au bout de quelques années, il reste « incomplet », et on s'aperçoit au fur et à mesure de son utilisation des manques qui ne sont pas en sa faveur, parmi lesquels :

-  Android est un système Open Source, mais toutefois, on peut se demander si Google joue le jeu, le code source de la version 3.0 (Honeycomb), dédiée aux tablettes, n'a jamais été fourni.
-  Lors d'obtention d'une liberté de personnalisation de son téléphone cellulaire et l'accès à des applications téléchargeables illimités, il y a des fortes possibilités de tout détruire. Étant donné qu'Android est open source, n'importe qui peut développer et publier une application sur le magasin. Ce qui augmente les risques que l'appareil tombe sur un bogue avec l'application téléchargée. Un bogue peut donner accès à distance des informations stockées dans l'appareil à un pirate ou à un simple bogue qui peut détruire le téléphone mobile en surchargeant l'utilisation de la batterie par exemple.
-  Le souci de ne pas pouvoir se permettre constamment de mettre à jour son téléphone avec des versions plus récentes. Et le redémarrage impératif des téléphones avec des risques que l'appareil ne fonctionnera pas correctement après la mise à jour. Les différentes versions et combinaisons n'ont pas été testées.
-  Synchronisation automatique impossible de téléphone Android sur PC, Il faudrait installer une application de synchronisation tierce ou synchroniser manuellement le téléphone Android avec le bureau.

## **VII. Conclusion :**

Dans ce chapitre, nous avons fait une étude de l'art du système Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation, son architecture (principaux composants).

Et d'après cette étude nous avons déduit que l'Android a confirmé sa position de maître absolu au niveau mondial, car les constructeurs y trouvent un intérêt, les utilisateurs aussi peuvent changer de Smartphones sans perdre de données ou d'applications.

Il a pu conquérir sa place grâce à ses particularités qui s'adaptent aux développeurs et aux utilisateurs.

Dans le deuxième chapitre nous allons entamer le langage de programmation Java car toutes les applications sont écrites en utilisant ce langage (Java).

## I. Introduction :

Le langage Java a fait une entrée spectaculaire dans le monde du développement logiciel en 1995. Il combinait des options déjà popularisées par certains de ses concurrents comme C/C++ ou Objective C. Dans ce chapitre nous allons présenter le langage Java en général, ainsi que son historique, ses versions, ses caractéristiques techniques, les outils nécessaires pour son développement, et ensuite nous feront une petite présentation de Java sous Android.

## II. Généralité :

### II.1. Présentation de langage Java

Java est à la fois un langage de programmation et une plate-forme de développement fournissant un environnement d'exécution rendant les applications indépendantes de la plateforme « système d'exploitation ».

#### Le langage java

Le langage Java est un langage de programmation informatique orienté objet créé par James Gosling et Patrick Naughton, employés de Sun Microsystems, avec le soutien de Bill Joy (cofondateur de Sun Microsystems en 1982), présenté officiellement le 23 mai 1995 à Sun World.

La société Sun Microsystems a été ensuite rachetée en 2009 par la société Oracle qui détient la licence Java et maintient ainsi la plateforme.

Il est aussi connue par être un langage futur ou le langage de l'internet parce qu'il est venu presque avec l'apparition de l'internet. Il est dit idéal car il combine deux niveaux de langage, un bas niveau (assembleur) et un autre de haut niveau qui est le langage C++. [05]

#### La plateforme java

Java est une plate-forme se composant d'une machine virtuelle et d'un environnement d'exécution. La machine virtuelle est un processeur articulé autour d'un logiciel qui présente un ensemble d'instruction. L'environnement d'exécution se compose des bibliothèques pour des programmes courants et de l'interaction avec le système d'exploitation cible.

L'environnement d'exécution inclut une bibliothèque conséquente de « class files » qui effectuent des tâches communes, telles que des opérations mathématiques (par exemple, trigonométrie) et des communications réseau. Cette bibliothèque est désignée généralement sous le nom de la bibliothèque standard.

Un programme spécial de Java connu sous le nom de « compilateur » de Java traduit le code source en instructions (et données associées) qui sont exécutées par la machine virtuelle.

Ces instructions sont connues sous le nom de « byte code ». [06]

### Java SE, Java EE, Java ME

Les développeurs emploient différentes éditions de la plate-forme de Java pour créer les programmes de Java qui fonctionnent sur des ordinateurs de bureau, des navigateurs Web, des serveurs web, des applications mobiles, des dispositifs embarqués, etc.

- ❖ Java Platform, Standard Edition (Java SE): un environnement pour les applications de bureaux.
- ❖ Java Platform, Enterprise Edition (Java EE): un sur-ensemble de JavaSE, orienté transaction et centré sur les bases de données (ayant pour cible les entreprises)
- ❖ Java Platform, Micro Edition (Java ME): environnement d'exécution et une API pour les systèmes embarqués (téléphone, Smartphone, assistants, TV, etc.).

## II.2. Les outils nécessaires pour développer en Java

Pour développer une application Java il faut avoir un minimum de ressources à savoir un environnement d'exécution Java(JRE) et un éditeur de texte ou un Environnement de Développement Intégré (IDE).

Cependant, il est possible de télécharger et installer un Kit de Développement Java SE (JDK) qui fournit des outils importants pour le développement d'un programme Java.

### Environnement d'exécution Java

L'environnement d'exécution Java (JRE pour Java Runtime Environment), parfois nommé simplement 'Java', est une famille de logiciels qui permet l'exécution des programmes écrits en langage de programmation Java sur différentes plateformes informatiques. Il est distribué gratuitement par Oracle Corporation, sous forme de différentes versions destinées aux systèmes d'exploitation Windows, Mac OS X et Linux, toutes conformes aux Java Spécification Requêtes (JSR). [05]

JRE est souvent considéré comme une plateforme informatique au même titre qu'un système d'exploitation (OS). En effet, s'il n'est pas un OS, il offre les mêmes fonctionnalités par l'intermédiaire de ses bibliothèques et permet ainsi l'exécution des programmes écrits en langage Java sur de nombreux types d'appareils tel qu'ordinateurs personnels, ordinateurs centraux, téléphone mobiles, en faisant abstraction des caractéristiques techniques de la plateforme informatique sous-jacente qui exécute le JRE. [05]

JRE est un logiciel populaire, installé sur plus de 900 millions d'ordinateurs.

### Environnement de Développement Intégré

L'environnement de développement intégré est une collection de programmes interconnectés destinés à faciliter le développement logiciel.

Plusieurs IDE ont été développés pour programmer en Java, parmi lesquels on cite :

- ❖ Eclipse,
- ❖ NetBeans,
- ❖ Visual Studio,
- ❖ WinDev,
- ❖ Dreamweaver.

Mais les plus utilisées d'après la figure ci-dessous sont :

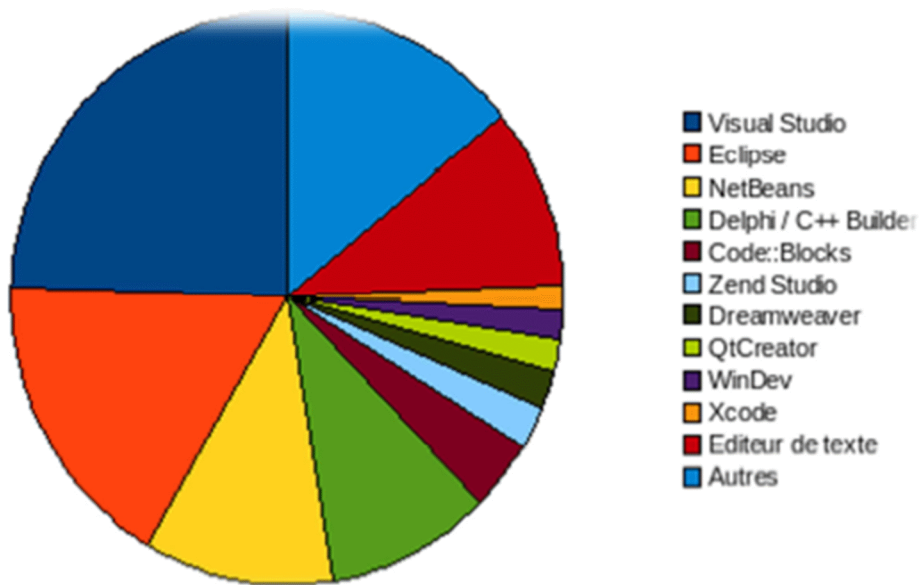


Fig.4 : Les taux d'utilisation des environnements de développement Java [07]

### Kit de Développement Java

Le Java développement Kit communément appelé JDK est le kit de développement de base que propose la société Oracle. Il comprend plusieurs outils, parmi lesquels :

- ❖ Javac : le compilateur Java,
- ❖ Java : un interpréteur d'application (machine virtuelle),
- ❖ Applet Viewer : un interpréteur d'applets,
- ❖ Jdb : un débogueur,
- ❖ Javap : un dé-compilateur, pour revenir du byte code au code source,
- ❖ Javadoc : un générateur de documentation,
- ❖ Jar : l'éditeur d'archives Java.

Il existe aussi des kits de développement matériels, utiles ou nécessaires notamment pour le développement d'applications sur les consoles de jeux.

### II.3. Les versions de Java

Le tableau ci-dessous illustre les différentes versions de java :

Année	Nom	Version	Nombre de classe de l'API
1996	Java JRE	1.0	170
1997	Java JRE	1.1	391

1998	Java 2 JRE	1.2	1462
2000	Java 2 JRE	1.3	1732
2002	Java 2 JRE	1.4	2367(54 packages)
2004	Java 2 JRE	1.5.0 ou 5.0	2800(166 packages)
2006	Java 6 JRE	1.6.0 ou 6.0	3793(203 packages)
2011	Java 7 JRE	1.7.0	4024(209 packages)

**Fig.5 : Les différentes versions de Java [05]**

## **II.4. Les caractéristiques de Java**

Certains des caractéristiques de base importantes de langage de programmation Java sont :

- ✚ Java est simple d'utilisation,
- ✚ Java est orienté objet,
- ✚ Java est sûre,
- ✚ Java est portable,
- ✚ Java est multitâche,
- ✚ Java est économe,
- ✚ Java est interprétée,
- ✚ Java est robuste,
- ✚ Java est dynamique,
- ✚ Java assure la gestion de la mémoire.

## **III. Quelques packages de base :**




Les fichiers sources peuvent être organisés en package. Les packages définissent une hiérarchie de noms, chaque nom étant séparé par le caractère point. Le nom d'un package est lié à une arborescence de sous répertoire correspondant à ce nom.

L'API Java est organisée en packages répartis en trois grands ensembles :

- ✚ Packages standards : ce sont les sous packages du package java
- ✚ Packages d'extensions : ce sont les sous packages du package javax
- ✚ Packages tiers : ces packages concernant notamment Corba et XML

#### **IV. Java sous Android :**

Pour développer avec java sous Android, on aura besoin des outils suivants :

-  L'IDE Eclipse,
-  Android Developer Tools (ADT),
-  Le SDK Android.

##### **IV.1. L'IDE Eclipse**

Eclipse est une plate-forme open source basé sur Java qui permet à un développeur de logiciel de créer un environnement de développement personnalisé (IDE) à partir de composants plug-in développés par des membres Eclipse. Eclipse est gérée et dirigée par le Consortium Eclipse.org.

Eclipse a fait ses débuts en 2001 quand IBM a fait don de trois millions de lignes de code de ses outils Java. L'objectif initial d'Eclipse était de créer et de favoriser une communauté IDE open source qui compléterait la communauté qui l'entoure Apache. [08]

##### **IV.2. Android Developer Tools**

ADT (Android Developer Tools) est un plugin pour Eclipse qui fournit une suite d'outils qui sont intégrés à l'IDE Eclipse. Il offre un accès à de nombreuses fonctionnalités qui permettent de développer des applications Android rapidement. ADT propose l'accès à l'interface graphique de la plupart des outils du SDK de ligne de commande ainsi que d'un outil de conception d'interface utilisateur pour le prototypage rapide, la conception et la construction de l'interface utilisateur de votre application.

Puisque ADT est un plugin pour Eclipse, vous obtenez les fonctionnalités d'un IDE bien établi, ainsi que des fonctionnalités spécifiques à Android qui sont regroupés avec ADT. [09]

##### **IV.3. Le SDK Android**

Ce kit de développement de logiciel permet aux développeurs de créer des applications pour la plateforme Android. Le SDK Android comprend des exemples de projets avec le code source, des outils de développement, un émulateur, et des bibliothèques nécessaires pour construire des applications Android. Les applications sont écrites en utilisant le langage de programmation Java et exécutés sur Dalvik, une machine virtuelle personnalisée, conçue pour une utilisation embarquée qui s'exécute au-dessus d'un noyau Linux. [10]

#### **V. Application Android :**

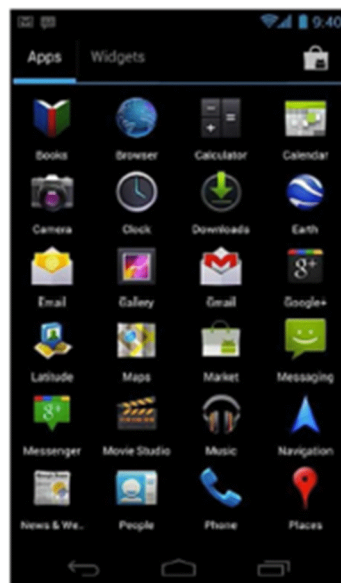
## V.1. Types d'applications

Une application Android est composée de deux parties

- ✚ Une série de fichiers XML
- ✚ Du code Java pour spécifier la partie dynamique de l'application

Les fichiers XML sont plus ou moins obligatoires et on trouve différents types d'application

- ❖ **Application de premier plan**, utilisable quand elle est visible et mise en pause lorsqu'elle ne l'est pas, comme les jeux.
- ❖ **Application d'arrière-plan**, en interaction limitée en dehors de sa configuration, comme le filtrage d'appels et réponses automatiques aux SMS.
- ❖ **Application intermittente**, interactive par moments, l'essentiel de sa tâche est en arrière-plan comme le lecteur de musique.
- ❖ **Widget**, qui est un composant visuel interactif que les utilisateurs peuvent ajouter à leur écran d'accueil comme l'affichage de l'heure et les prévisions météo.



**Fig.6 : Menu d'applications Android**

## V.2. Cycle de vie d'une application Android

Contrairement aux applications classiques, les applications Android n'ont pas un contrôle complet sur leur cycle de vie.

Les composants doivent être à l'écoute des changements d'états de l'application pour être prêt à une fin intempestive.

La mémoire et la gestion des processus est prise en charge exclusivement par le moteur d'exécution.

## VI. Les principaux composants d'une application :

### VI.1. Les services sous ANDROID

Un service est une composante essentielle d'une application, à la différence d'une activité, ne possède pas de vue (d'interface graphique) mais permet l'exécution d'un algorithme sur un temps indéfini. Il ne s'arrêtera que lorsque la tâche est finie ou que son exécution est arrêtée.

Il est nécessaire lorsque l'application souhaite effectuer des opérations ou des calculs en dehors de l'interaction utilisateur.

Il peut être lancé à différents moments :

- ✚ Au démarrage du téléphone.
- ✚ Au moment d'un événement (arrivée d'un appel, SMS, mail).
- ✚ Lancement de votre application.
- ✚ Action particulière dans votre application.

Il existe deux types de services : **Local Service** qui s'exécute dans le même processus que votre application. **Remote Service** il s'exécute dans des processus indépendants de votre application.

### VI.2. Broadcast Receiver, Intent

Un Broadcast Receiver (Récepteur d'émission) comme son nom l'indique permet d'écouter ce qui se passe sur le système ou sur votre application et déclencher une action que vous aurez prédéfinie. C'est souvent par ce mécanisme que les services sont lancés.

Les intents (intentions) permettent de communiquer entre les différentes activités définie dans les applications Java, mais aussi du téléphone. Ils sont en quelque sorte le "messenger" pour lancer une activité. Ainsi une activité peut en lancer une autre soit en passant un intent vide, soit en y passant des paramètres. Les Intent Filters jouent le rôle de filtre. Ils permettent de contrôler d'où provient l'Intent (ou d'autres paramètres) afin de lancer ou non l'activité.

L'interface **Broadcast Receiver** ne possède qu'une seule méthode **onReceive ()** que votre classe devra implémenter. Il ne vit que le temps de traiter **onReceive ()**. L'instance peut être supprimée par le « *Garbage Collector* ».

Les *Receivers* sont limités : ils ne peuvent pas ouvrir de boîte de dialogue par exemple.

### VI.3. Les Widgets

Terme résultant de la contraction des termes window et gadget.

Les widgets sont les éléments principaux de la création d'interfaces utilisateur. Ils peuvent afficher des données et des informations sur un état, recevoir des actions de l'utilisateur et agir comme un conteneur pour d'autres widgets qui doivent être regroupés. Un widget qui n'est pas intégré dans un widget parent est appelé une fenêtre.

Un widget peut désigner :

- ✚ **Un composant d'interface graphique** : un élément visuel d'une interface graphique (bouton, ascenseur, liste déroulante, etc.),
- ✚ **Un widget interactif** : un petit outil qui permet d'obtenir des informations (météo, actualité, dictionnaire, carte routière).

#### **VI.4. Le fichier Manifest**

Décrit les composants de l'application, c'est un point d'entrée potentiel dans l'application dont on trouve :

- ✚ Le nom du package de l'application. Il servira d'identifiant unique.
- ✚ Tous les composants de l'application (activités, services, broadcast receivers, content providers). On y décrit également les classes qui implémentent ces composants et leurs capacités (par exemple les Intents qu'elles attendent). Ces déclarations permettent à Android de savoir quels composants sont présents et dans quelles conditions ils s'exécutent.
- ✚ On détermine dans quels processus les composants de l'application seront contenus.
- ✚ Les permissions nécessaires pour le bon fonctionnement de l'application.
- ✚ Les permissions nécessaires pour que les autres applications utilisent les composants de votre application.
- ✚ Les informations contenant les versions de l'Android api requis pour exécuter votre application.
- ✚ Les bibliothèques utilisées par votre application.

```

<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.monspinner"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="3"
        android:targetSdkVersion="19" />

    <application
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.example.monspinner.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>

```

Fig.7 : Exemple d'un fichier Manifest

## VI.5. Les fichiers XML sous Android et Layout

Le langage de balisage extensible « *Extensible Markup Language* » est un format de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms.

Un Layout est le placement des composants graphiques dans l'IHM. Avec Android, il peut être décrit dans un fichier XML.

Un Layout Permet de concevoir des interfaces plus simplement qu'en langage Java, et d'instancier les widgets.

## VI.6. Interface Graphique sous Android

Les interfaces graphiques sous Android peuvent être :

- ✚ Déclarative (Fichiers XML dans le répertoire /res/layout/),
- ✚ Ou directement dans le code java.

Les vues (View) :

- ✚ Composant élémentaire des interfaces graphiques sous Android,
- ✚ Instance de la classe View,
- ✚ Sensible aux événements,
- ✚ Tous les composants graphiques (widgets) d'Android sont des vues : Button, EditText, TextView, Spinner, CheckBox, etc.

Regroupement de vues (ViewGroup)

- ✚ Le View Group permet de regrouper des vues,
- ✚ Les layouts disponibles sont : Linear Layout, Table Layout, Relative Layout, Frame Layout, Scroll View, etc.

### VI.7. SQLite

*SQLite* est une bibliothèque écrite en C qui propose un moteur de base de données relationnelle accessible par le langage SQL. SQLite implémente en grande partie le standard SQL-92 et des propriétés ACID (*Atomicity* (atomicité), *Consistency* (cohérence), *Isolation*, *Durability*(durabilité)).

Contrairement aux serveurs de bases de données traditionnels, comme MySQL, sa particularité est de ne pas reproduire le schéma habituel client-serveur mais d'être directement intégrée aux programmes. L'intégralité de la base de données (déclarations, tables, index et données) est stockée dans un fichier indépendant de la plateforme.

#### ✚ Principales fonctionnalités de SQLite

- ❖ Gestion de plusieurs bases de données,
- ❖ Gestion des bases de données attachées,
- ❖ Création, insertion, modification et suppression de tables et index,
- ❖ Gestion des Vues, possibilités de créer une vue à partir d'une requête SELECT,
- ❖ Gestion des Trigger,
- ❖ Gestion des fonctions utilisateurs, ces fonctions sont ensuite utilisable dans les requêtes manuelles, comme dans le formulaire d'insertion/modification de données,
- ❖ Requête manuelle et à partir d'un fichier, il est possible de préciser le format des requêtes, SQLite ou MySQL, une conversion est effectuée afin d'importer directement une base MySQL en SQLite.

### *VII. Conclusion :*

Dans ce deuxième chapitre on a vu une simple présentation de langage java et sa machine virtuelle, ses différentes versions ainsi que les outils nécessaires au développement sous Java. En ce qui va suivre nous allons entamer la partie conception de notre application

## *I. Introduction :*

La conception et le développement est une succession d'activités créatrices qui, partant des besoins exprimés et des connaissances existantes, aboutissent à la réalisation d'une application satisfaisant ces besoins.

La phase de conception est donc l'activité qui consiste à développer les meilleures solutions à partir d'un besoin donné. Cette phase suit l'étape d'analyse dans laquelle on met en évidence les différents acteurs ainsi que leurs besoins en illustrant les différents cas d'utilisation.

Dans ce chapitre, nous allons donc parler des principaux composants d'une application Android et expliquer le processus de développement et de conception de notre application et tout ça se basera sur la modélisation UML.

## *II. Objectif de l'application*

Les applications mobiles ont pris un espace important dans notre vie quotidienne, on passe la plupart du temps à jouer avec les différentes applications de notre Smartphone ce qui nous éloigne de méthodes éducatives classiques.

Notre application étant portable elle sert en quelque sorte d'une classe de cours de français voyageant avec l'utilisateur, à la fois dans tous ses déplacements, et dans un univers d'enseignement spécifique et interactif, avec des cours importants bien résumés, des tests à plusieurs niveaux de difficultés.

L'application contient tout ce qu'il faut pour améliorer votre niveau et découvrir de nouveaux mots en français en vous divertissant par des poèmes, proverbes et citations de différentes catégories.

## *III. Présentation de l'UML :*

Le langage UML (Unified Modeling Language) est un langage graphique de modélisation initialement conçu pour représenter, spécifier, concevoir et documenter les artefacts de systèmes logiciels, adapté par l'OMG (Object Management Group) en tant que standard. Il est devenu une référence incontournable dans le domaine de génie logiciel, sa richesse et sa puissance d'expression le rendent également éligible pour la modélisation de concepts et de processus « métier » (business modeling) et pour l'ingénierie des systèmes non logiciels.

L'UML intègre neuf types de diagrammes destinés à la caractérisation du système modélisé et à la représentation des éléments qui le contiennent.

- **Les diagrammes de cas d'utilisation :** permettent de décrire les fonctionnalités du système et de représenter les différents types de sollicitations auxquelles il doit pouvoir répondre.
- **Les diagrammes de classes :** sont destinés à décrire les propriétés structurelles des objets du monde réel, les concepts spécifiques du domaine considéré ou encore les notions abstraites que le système doit appréhender. Ce sont les diagrammes les plus utilisés en modélisation orientée objet. En phase de conception du logiciel, ils sont

exploités pour décrire l'architecture statique du système et les interdépendances entre ses constituants.

- **Les diagrammes d'objets** : offrent un moyen de représenter les objets (c'est-à-dire les instances des classes figurant dans les diagrammes de classes) ainsi que leurs relations.
- **Les diagrammes de collaboration** : permettent de formaliser les scénarios de mise en œuvre du système et de montrer comment les objets sont mis en jeu pour réaliser les cas d'utilisation. Ils décrivent les interactions entre objets.
- **Les diagrammes de séquences** : comme les diagrammes de collaboration, ces diagrammes décrivent les interactions entre objets. Ils mettent l'accent sur l'ordre chronologique dans lequel s'effectuent les échanges de messages entre objets.
- **Les diagrammes d'états** : apportent le complément nécessaire à la formalisation des aspects dynamiques : ils répondent au besoin de modéliser les processus d'exécution et les comportements des objets en réaction aux stimuli auxquels ils sont soumis.
- **Les diagrammes d'activités** : sont également dédiés à la représentation de l'exécution d'un processus : ils constituent une variante des diagrammes d'états.
- **Les diagrammes de composants** : sont destinés à la description des éléments de configuration qui constituent les logiciels (binaires exécutables, bibliothèques, unités de compilation, etc.) et à la formalisation de leurs dépendances.
- **Les diagrammes de déploiements** : permettent de représenter l'implantation des différents programmes et composants logiciels sur l'architecture physique du système.

#### *IV. Processus de développement*

##### *IV.1. Analyse*

###### *A. Identification des acteurs*

Après avoir étudié les besoins de notre système nous avons procédé à l'identification d'un seul acteur principal.

Un acteur étant une entité extrême qui interagit avec le système le voici pour notre cas :

- **Utilisateur** : toute personne disposant d'un mobile (Smartphone) Android, après un lancement volontaire par l'utilisateur, ce dernier aura le choix entre démarrer l'application ou l'abandonner en quittant, le bouton démarrer donne accès à un menu à multiple choix, à savoir enrichir ses connaissances en faisant des révisions grâce à la partie théorique de l'application, améliorer son orthographe et sa grammaire en passant par des tests à 03 niveaux et enfin l'utilisateur peut prendre une pose et se divertir en se cultivant par la poésie et les proverbes classés par catégories, et rien n'est plus divertissant qu'une musique douce accompagnés de belles citations, à tout moment il peut abandonner une tâche pour aller à une autre.

## B. Présentation de l'application

L'application « Enrichir votre français » est une application d'éducation de la langue française.

Elle contient des cours et des tests qui ont été pris de site original de formation suivant <http://www.ameliorer-votre-francais.com/>. En plus des poèmes de différents poètes, et des proverbes populaires classés en catégories.

La Figure suivante résume la hiérarchie de l'application.

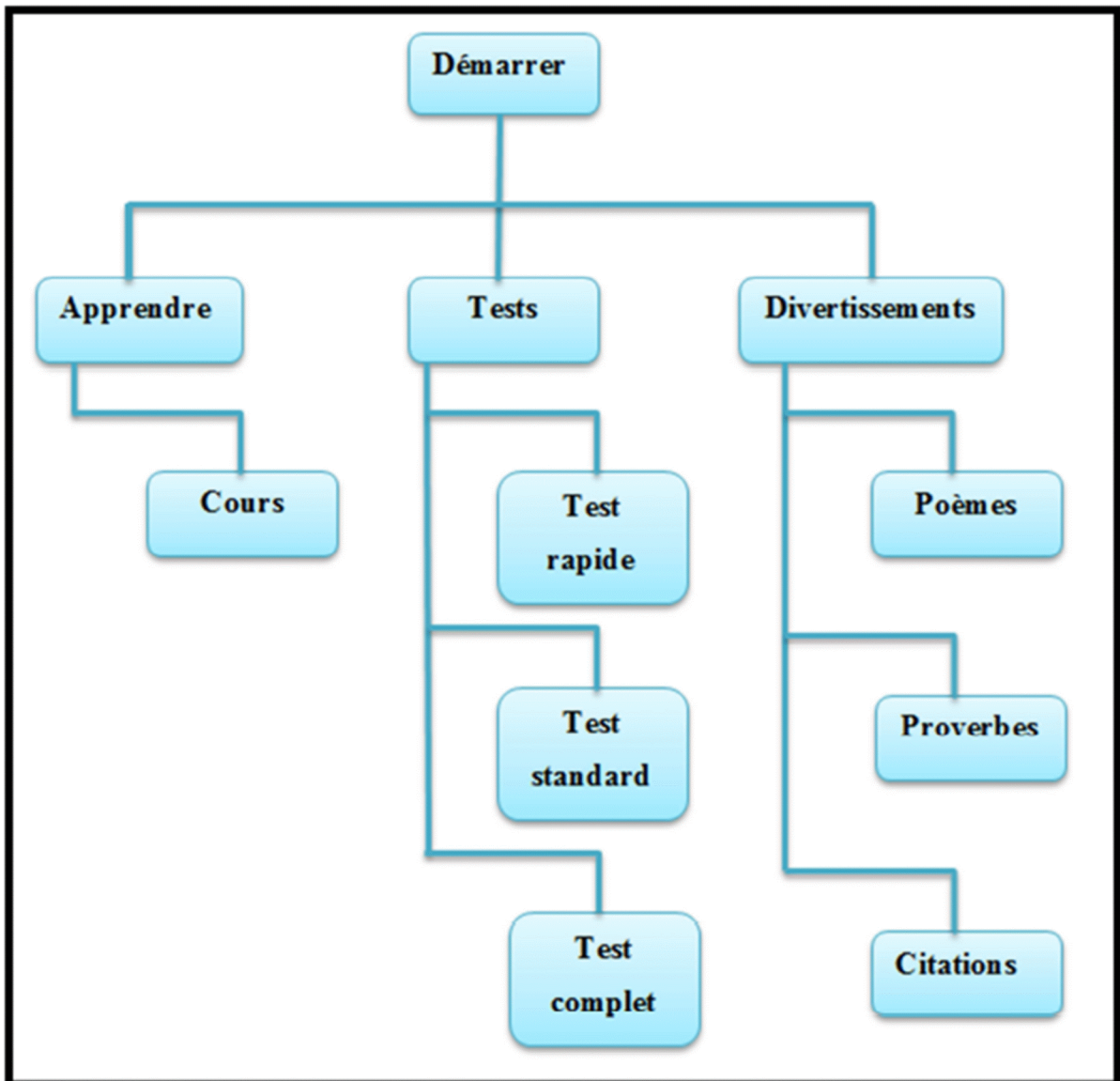


Fig.8 : Une modélisation simple de l'application

### C. Les cas d'utilisation:

C'est une représentation en général d'un ensemble d'actions qui sont réalisées par le système. Elle permet de représenter tous les besoins et les exigences que notre application doit respecter et suivre.

Dans notre cas on distingue beaucoup de cas d'utilisation, en voici certains :

#### 1) Suivre un cours

**Cas d'utilisation :** Suivre un cours

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « Démarrer ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « Apprendre ».
6. Le système affiche le sommaire des cours disponibles.
7. L'utilisateur choisit un cours et clique dessus pour y accéder.
8. Le système affiche le cours.

#### 2) Effectuer un test rapide

**Cas d'utilisation :** Effectuer un test rapide

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « Démarrer ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit sur le bouton « Test ».
6. Le système affiche la page du test principal.
7. L'utilisateur choisit le bouton « Test rapide » et clique dessus.
8. Le système affiche les 10 questions de test.
9. L'utilisateur répond aux questions, puis appui à chaque fois sur le bouton « valider » pour passer à la question suivante.
10. A la fin de la partie, le système affiche la note de l'utilisateur et lui propose s'il veut le sauvegarder.
11. En cas de oui, l'utilisateur saisie son nom et clique sur le bouton « sauvegarder ».
12. Le système sauvegarde la note dans la base de données, rien sinon.

### 3) Effectuer un test standard

**Cas d'utilisation** : Effectuer un test standard

**Acteurs** : Utilisateur

**Description** :

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « Démarrer ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Test** ».
6. Le système affiche la page du test principal.
7. L'utilisateur choisit le bouton « **Test standard** » et clique dessus.
8. Le système affiche les **40 questions** de test standard.
9. L'utilisateur répond aux questions, puis appui à chaque fois sur le bouton « **valider** » pour passer à la question suivante.
10. A la fin de la partie, le système affiche la note de l'utilisateur et lui propose s'il veut le sauvegarder.
11. En cas de oui, l'utilisateur saisie son nom et clique sur le bouton « **sauvegarder** ».
12. Le système sauvegarde la note dans la base de données, rien sinon.

### 4) Effectuer un test complet

**Cas d'utilisation** : Effectuer un test complet

**Acteurs** : Utilisateur

**Description** :

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « Démarrer ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Test** » et clique dessus.
6. Le système affiche la page du test principal.
7. L'utilisateur choisit le bouton « **Test complet** » et clique dessus.
8. Le système affiche les **100 questions** de test complet.
9. L'utilisateur répond aux questions, puis appui à chaque fois sur le bouton « **valider** » pour passer à la question suivante.
10. A la fin de la partie, le système affiche la note de l'utilisateur et lui propose s'il veut le sauvegarder.
11. En cas de oui, l'utilisateur saisie son nom et clique sur le bouton « **sauvegarder** ».
12. Le système sauvegarde la note dans la base de données, rien sinon.

### 5) Lire un poème

**Cas d'utilisation :** Lire un poème

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « **Démarrer** ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Divertissement** ».
6. Le système affiche la page de menu divertissement.
7. L'utilisateur clique sur le bouton « **Poèmes** ».
8. Le système affiche le sommaire principal des poèmes.
9. L'utilisateur clique sur un des poèmes qui sont proposés au sommaire.
10. L'utilisateur accède à un poème choisi.

### 6) Afficher un proverbe

**Cas d'utilisation :** Afficher un proverbe

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « **Démarrer** ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Divertissement** ».
6. Le système affiche la page de menu divertissement.
7. L'utilisateur clique sur le bouton « **Proverbes** ».
8. Le système affiche la page principale des **9 catégories** de proverbes.
9. L'utilisateur choisit une catégorie et clique dessus.
10. Le système affiche les proverbes.

### 7) Afficher une citation

**Cas d'utilisation :** Afficher une citation

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « **Démarrer** ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Divertissement** ».
6. Le système affiche la page de menu divertissement.
7. L'utilisateur clique sur le bouton « **Citations** ».
8. Le système affiche les citations en images sous forme d'une vidéo accompagné d'une musique.

### 8) Sauvegarder un score dans la base de données

**Cas d'utilisation :** Sauvegarder un score dans la base de données

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur est dans la fenêtre du résultat de test.
2. L'utilisateur saisie son nom et clique sur le bouton « **Sauvegarder** ».
3. Le système récupère la note et son nom et ajoute le niveau de test joué et les enregistre dans la base de données.

### 9) Afficher tous les scores

**Cas d'utilisation :** Afficher tous les scores

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur est à la fenêtre principale des tests.
2. L'utilisateur clique sur le bouton « **voir tous les scores** ».
3. Le système affiche la liste des scores sauvegardés, la liste est vide sinon.

### 10) Supprimer un score dans la base de données

**Cas d'utilisation :** Supprimer un score dans la base de données

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « **Démarrer** ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Test** » et clique dessus.
6. Le système affiche la page du test principal.
7. L'utilisateur choisit le bouton « **voir tous les scores** » et clique dessus.
8. Le système affiche la liste des scores obtenus par l'utilisateur, ou bien une liste vide s'il n'y a aucun score sauvegardé.
9. L'utilisateur choisit le score qu'il veut supprimer.
10. Une boîte de dialogue sera affichée.
11. L'utilisateur choisit « oui » pour confirmer la suppression, « non » pour annuler.
12. Le système supprime le score de la base de données.

### 11) Vider la base de données

**Cas d'utilisation :** vider la base de données

**Acteurs :** Utilisateur

**Description :**

1. L'utilisateur lance l'application en cliquant sur l'icône.
2. Le système affiche la page d'accueil.
3. L'utilisateur clique sur le bouton « **Démarrer** ».
4. Le système affiche la page du menu principal.
5. L'utilisateur choisit le bouton « **Test** » et clique dessus.
6. Le système affiche la page du test principal.
7. L'utilisateur choisit le bouton « **voir tous les scores** » et clique dessus.
8. Le système affiche la liste des scores obtenus par l'utilisateur, ou bien une liste vide s'il n'y a aucun score sauvegardé.
9. L'utilisateur clique sur le bouton « **supprimer tous** ».
10. Une boîte de dialogue sera affichée.
11. L'utilisateur choisit « oui » pour confirmer la suppression, « non » pour annuler.
12. Le système vide la base de données.

**IV.2. Conception:**

**1. Le niveau applicatif**

❖ **L'activité**

**Définition :**

Toute application Android nous donne la possibilité d'interagir avec les différentes fenêtres qu'elles nous offrent afin de naviguer entre celles-ci, une activité remplit tout l'écran, une application ne peut donc en afficher qu'une seule à la fois.

Une activité est un support sur lequel nous allons greffer une interface graphique qui est un ensemble d'éléments visuels avec lesquels peuvent interagir les utilisateurs ou bien leurs fournir des informations.

De plus une activité contient des informations sur l'état actuel de l'application : ces informations s'appellent les contextes.

❖ **Etat d'une activité :**

Sous Android il y a des applications plus prioritaires que d'autres, donc la plus prioritaire doit passer en premier lieu, pour assurer ce mécanisme, Android a eu recours à un système particulier :

- ❖ À tout moment une application peut laisser place à une autre application, qui a une priorité plus élevée. Si une application utilise trop de ressources système, alors elle empêchera le système de fonctionner correctement et Android l'arrêtera sans vergogne.
- ❖ Une activité existera dans plusieurs états au cours de sa vie, par exemple un état actif pendant lequel l'utilisateur l'exploite, et un état de pause quand l'utilisateur reçoit un appel.

Pour être plus précis, quand une application se lance, elle se met tout en haut de ce qu'on appelle la pile d'activités qui fonctionne avec le mécanisme « LIFO » (Last In First Out). Le tableau suivant récapitule les différents états que peut avoir une activité

Etat	Visibilité	Description
Active («active » Ou «running»)	L'activité est visible en totalité.	Elle est sur le dessus de la pile, c'est ce que l'utilisateur consulte en ce moment même et il peut l'utiliser dans son intégralité. C'est cette application qui a le focus, c'est-à-dire que l'utilisateur agit directement sur l'application.
Suspendue (« paused »)	L'activité est partiellement visible à l'écran. C'est le cas quand vous recevez un SMS et qu'une	Ce n'est pas sur cette activité qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application

	fenêtre semi-transparente se pose devant votre activité pour afficher le contenu du message et vous permettre d'y répondre par exemple.	<p>sus-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser de l'application qui l'obstrue, puis l'utilisateur pourra à nouveau interagir avec.</p> <p>Si le système a besoin de mémoire, il peut très bien tuer l'application (cette affirmation n'est plus vraie si vous utilisez un SDK avec l'API 11 minimum).</p>
Arrêtée («stopped»)	L'activité est tout simplement oblitérée par une autre activité, on ne peut plus la voir du tout.	<p>L'application n'a évidemment plus le focus, et puisque l'utilisateur ne peut pas la voir, il ne peut pas agir dessus.</p> <p>Le système retient son état pour pouvoir reprendre, mais il peut arriver que le système tue votre application pour libérer de la mémoire système.</p>

❖ **Le cycle de vie d'une activité :**

Une activité n'a pas de contrôle direct sur son propre état, il s'agit plutôt d'un cycle rythmé par les interactions avec le système et d'autres applications. La figure ci-dessous présente le cycle de vie d'une activité, c'est-à-dire qu'il indique les étapes que va traverser l'activité pendant sa vie, de sa naissance à sa mort. Chaque étape du cycle est représentée par une méthode.

Les transitions d'états sont captées par les méthodes suivantes :

- **onCreate()** : l'activité est en création.
- **onStart()** : l'activité va devenir visible.
- **onResume()** : l'activité est maintenant visible
- **onPause()** : l'activité va être mise en pause.
- **onStop()** : l'activité ne sera plus visible.
- **onDestroy()** : l'activité va être détruite.

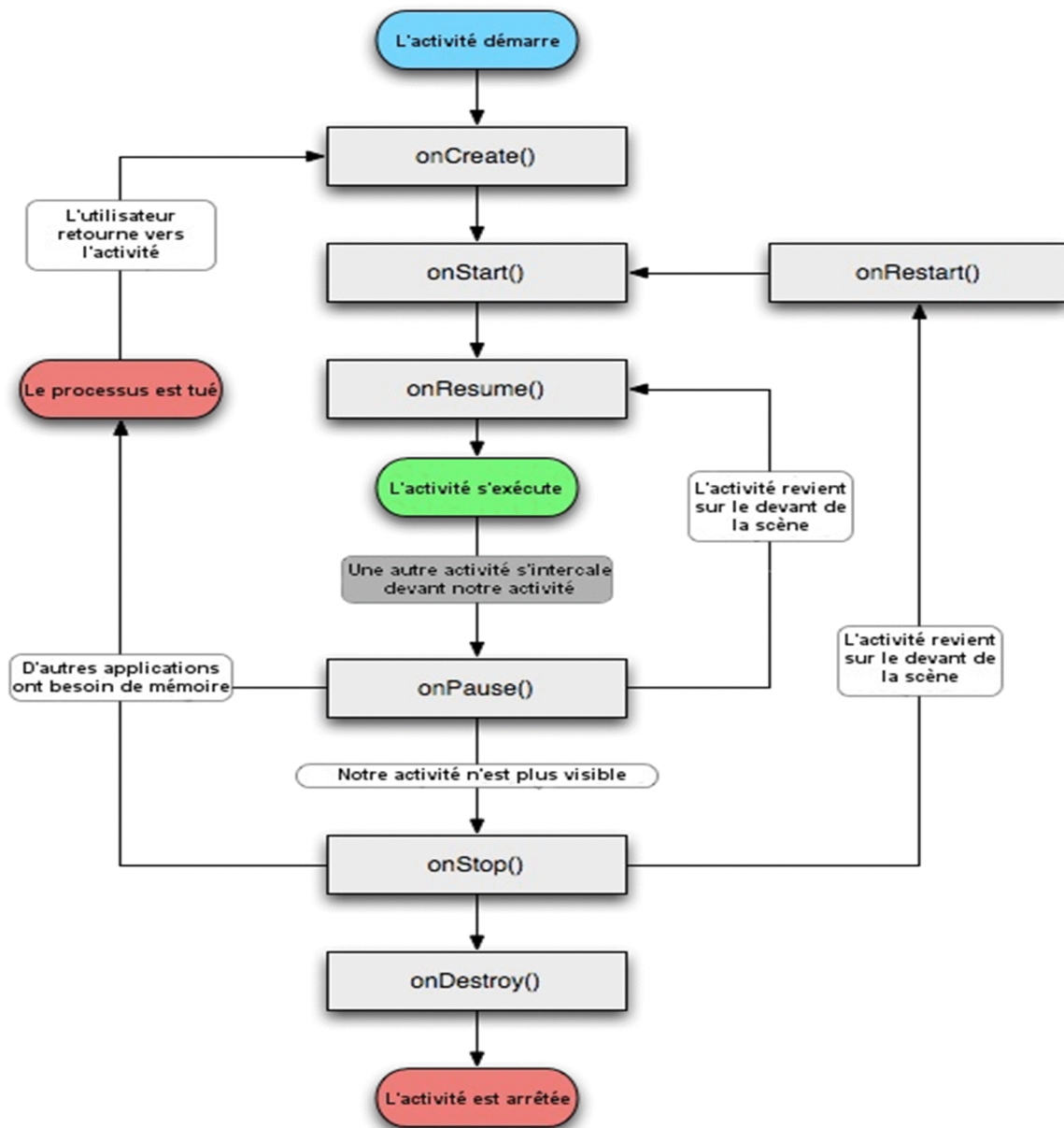


Fig.9 : Cycle de vie d'une activité

## 2. Diagrammes d'activités

Ils concernent le comportement interne des options ou des cas d'utilisation vu que les concepts action et activité sont au cœur de ce diagramme nous les définirons en quelques lignes :

Action : est un traitement qui modifie l'état du système. Elle est représentée par un rectangle dont les coins sont arrondis.

Activité : elle représente le comportement d'une partie du système interne d'action et de transition.

Les diagrammes d'activités dans notre application seront modélisés comme suit :

- Suivre un cours

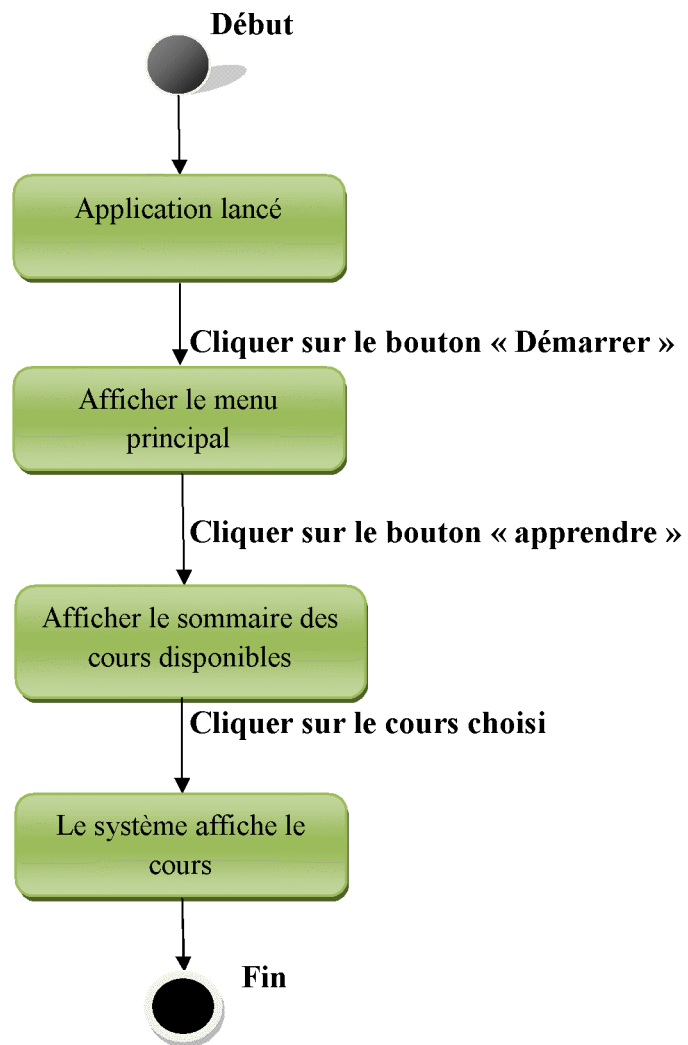


Fig.10 : diagramme d'activités du cas d'utilisation « suivre un cours»

- Effectuer un test :

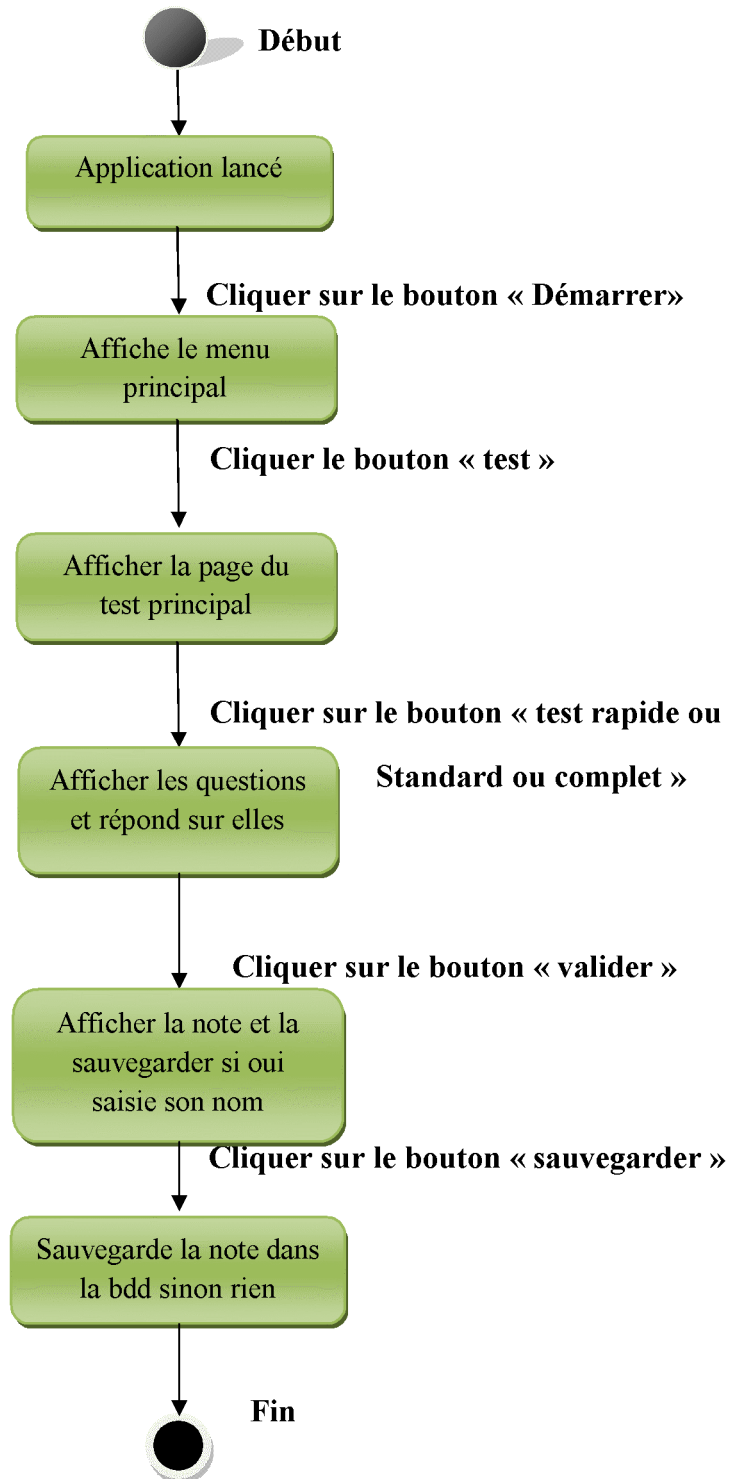


Fig.11 : diagramme d'activités du cas d'utilisation « effectuer un test rapide ou standard ou complet »

- Lire un poème :

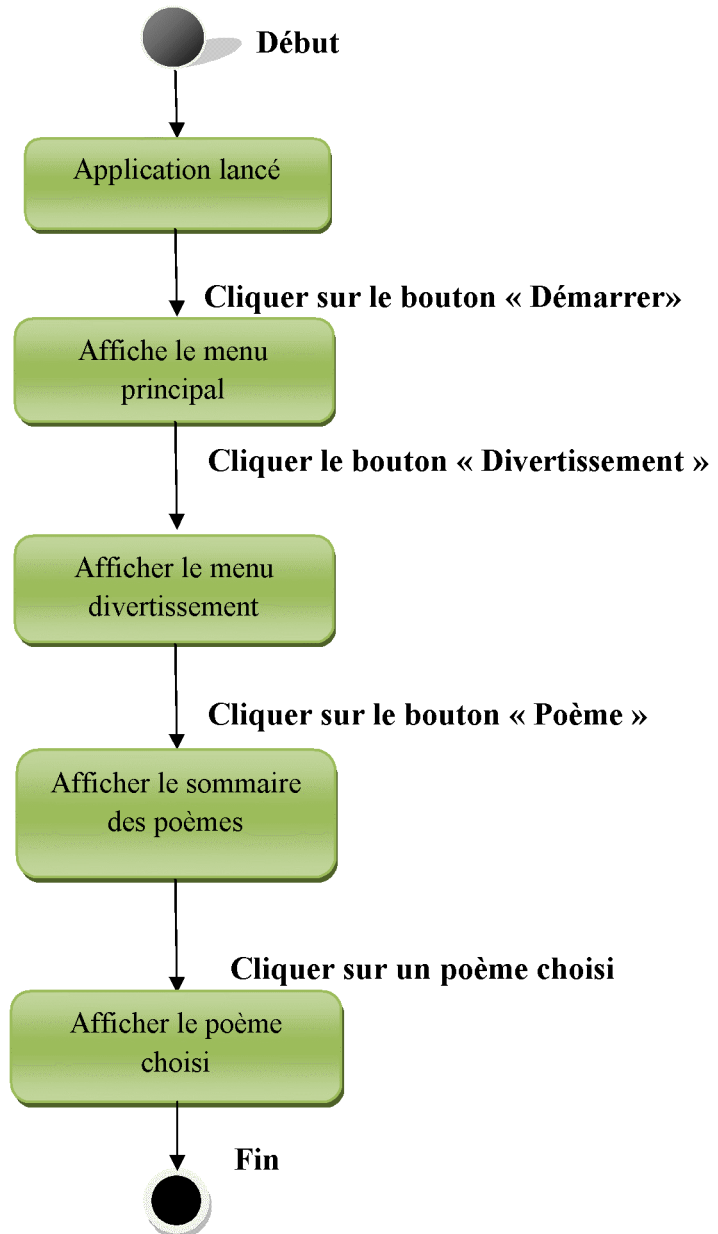


Fig.12 : diagramme d'activités du cas d'utilisation « lire un poème »

- Afficher un proverbe :

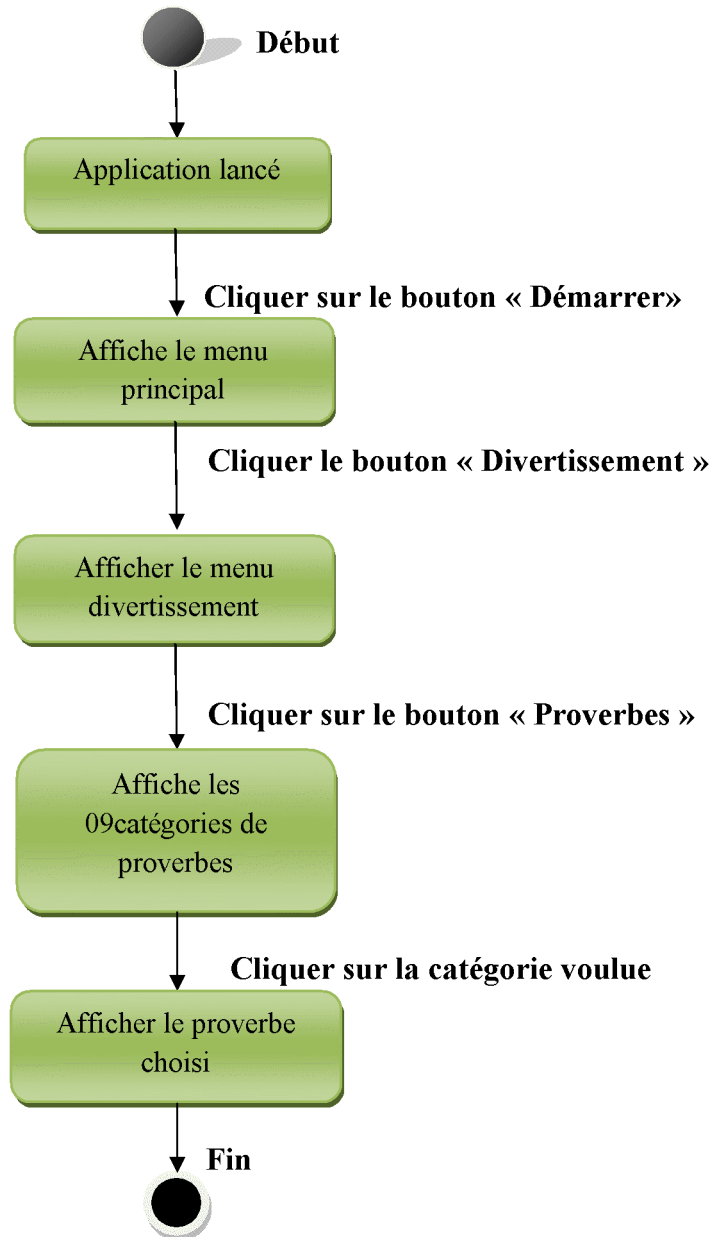


Fig.13 : diagramme d'activités du cas d'utilisation « Affiche un proverbe »

- Afficher une citation :

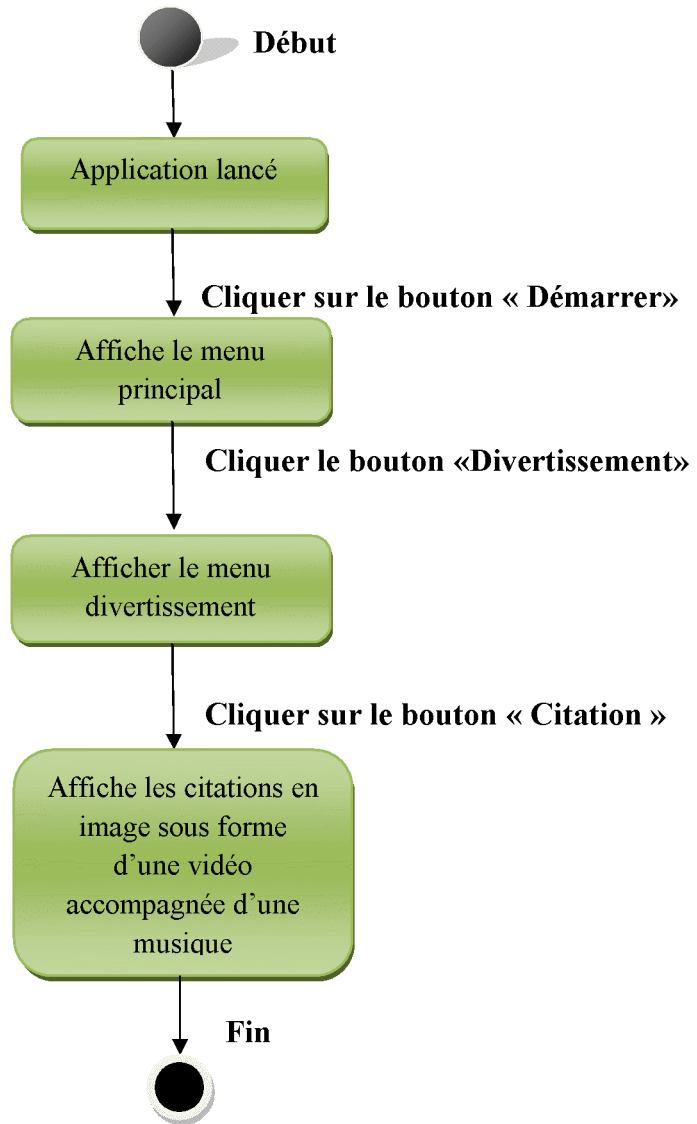


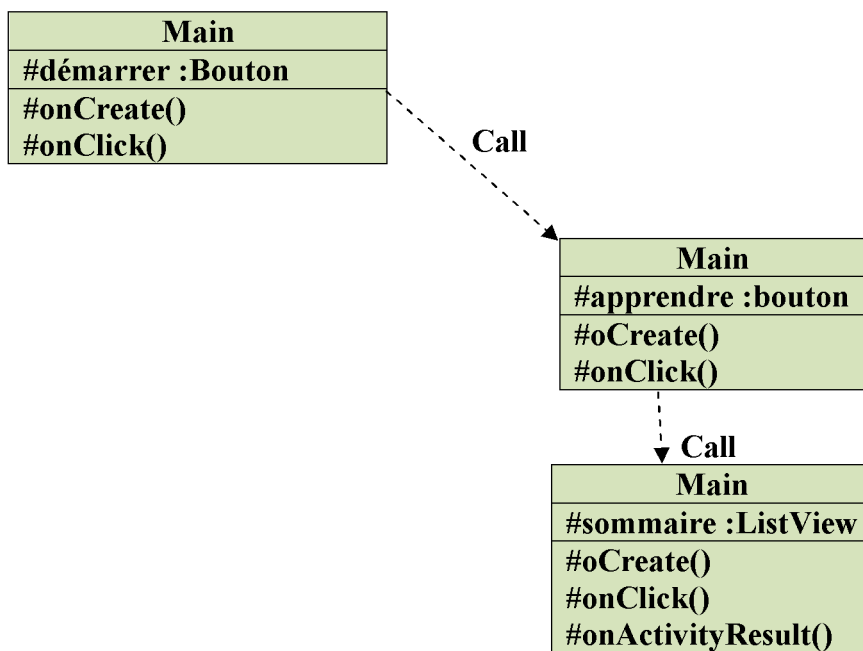
Fig.14 : diagramme d'activités du cas d'utilisation « Afficher une citation »

**3. Diagramme de classes :**

Constitue l'un des pivots essentiel de la modélisation UML il permet de donner la représentation statique du système à développer .Cette représentation est centrée sur les concepts de classes et d'associations.

Après avoir élaboré le diagramme d'activités pour les cas d'utilisation déjà décrit, nous allons élaborer des diagrammes de classes détaillés pour chacun d'eux.

- **Diagramme de classe Suivre un cours :**



**Fig.15 : Diagramme de classe « Suivre un cours »**

- Diagramme de classe effectuer un test:

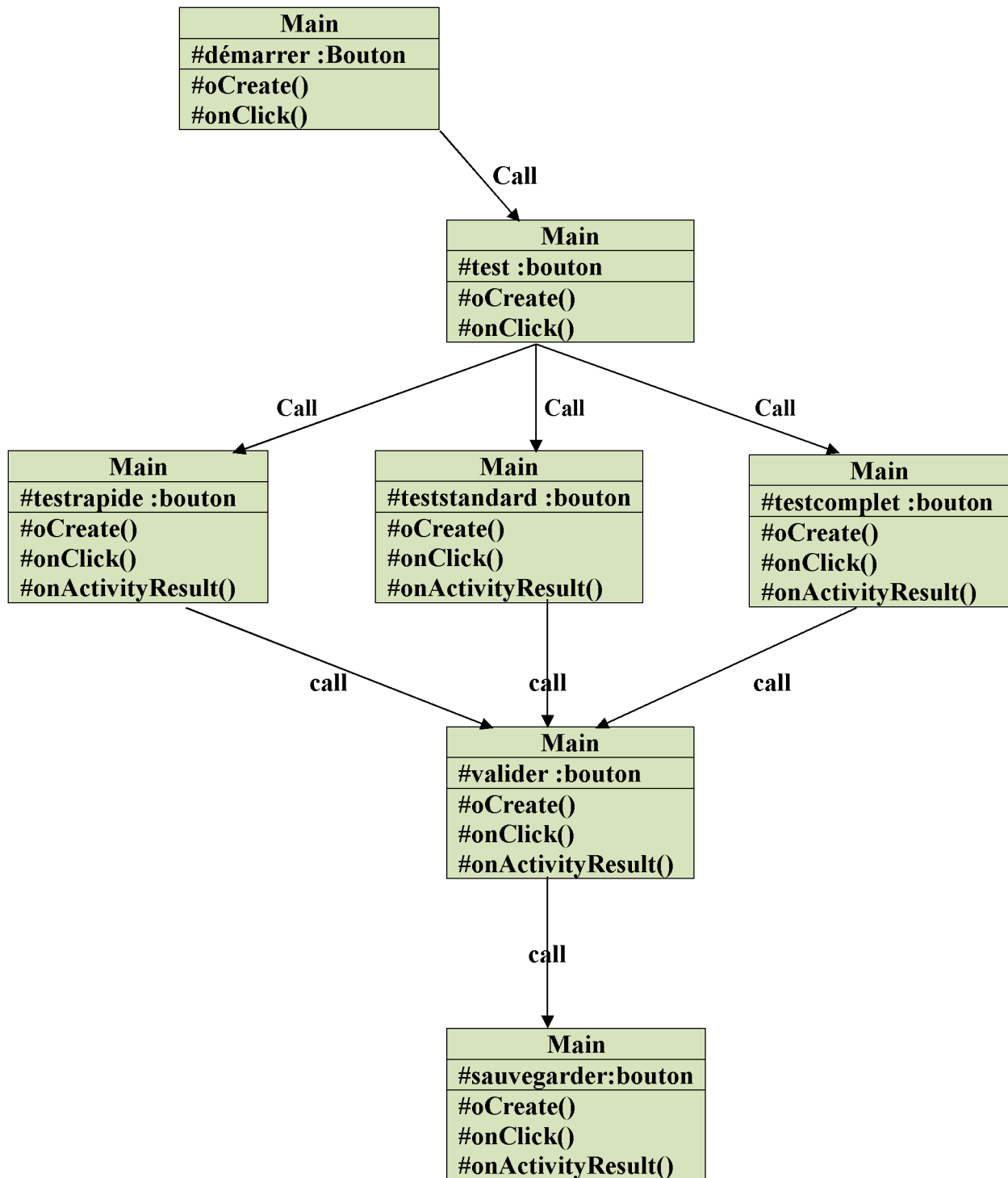


Fig.16 : Diagramme de classe « effectuer un test »

- Diagramme de classe des divertissements:

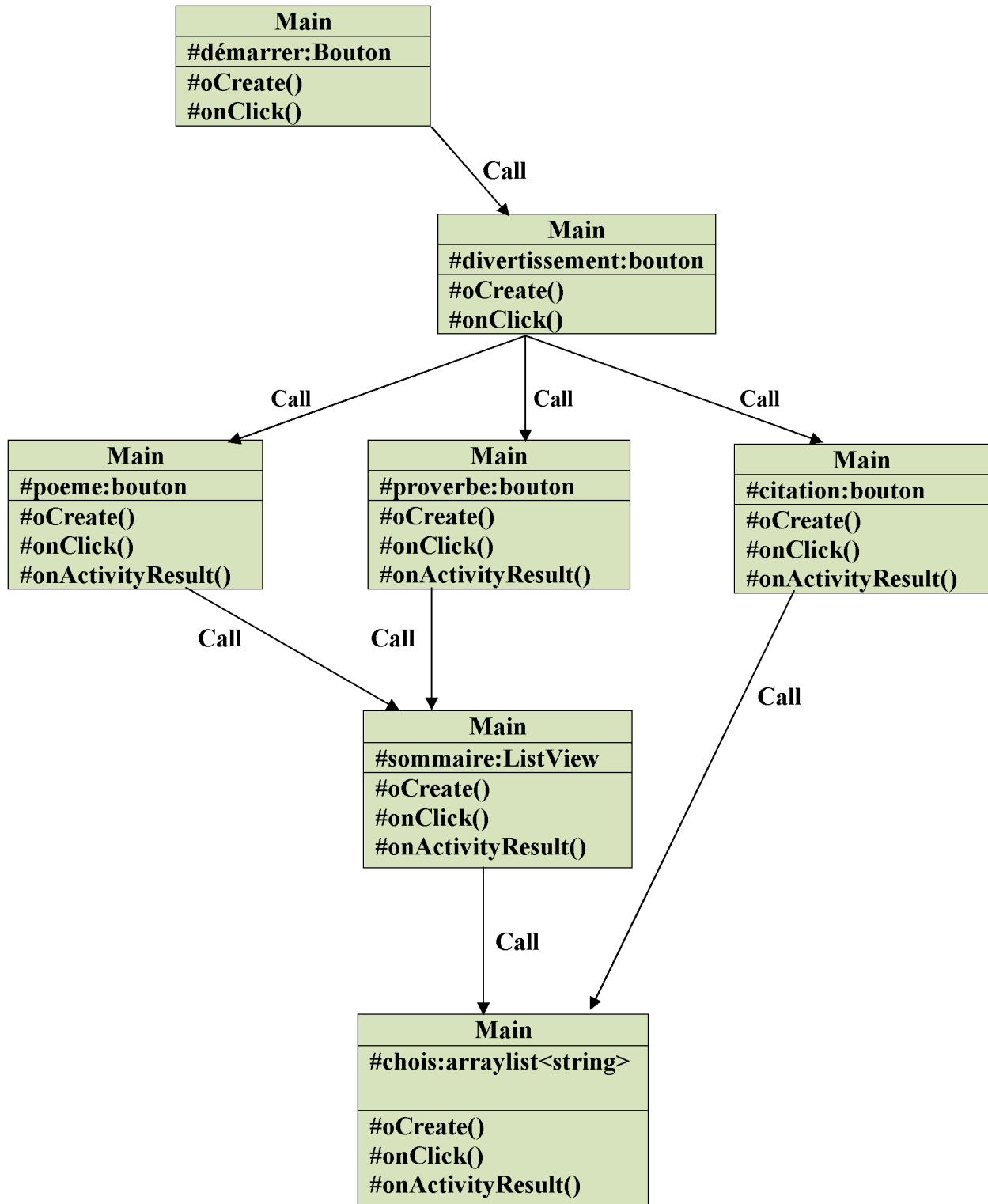


Fig.17 : Diagramme de classe « effectuer un divertissement »

#### 4. Diagramme logique (le niveau de données) :

L'application dans tous ces cas d'utilisation manipule les données stockées dans la base de données, et elle contient la table suivante :

La base de données SQLite est constituée d'une seule table pour sauvegarder les scores de l'utilisateur, elle est composée de quatre attributs comme est illustré :

Table score		
Champ	type	clé
ID	NUM	Primaire(PK)
Nom	VARCHAR	
Niveau	VARCHAR	
Note	NUM	

Fig.18 : Modèle de la table de données score

#### V. Conclusion:

A l'issue de ce chapitre, nous avons introduit les objectifs de l'application développée. Tout en donnant une description complète de ces diagrammes. Ce qui permet une meilleure compréhension du système.

Le chapitre suivant sera consacré à la réalisation et l'implémentation de notre application qui devra satisfaire les objectifs exprimés dans ce chapitre.

## **I. Introduction :**

Le développement d'applications Android se fait en un « atelier » spécifique avec les outils propres à cette plateforme.

Dans ce dernier chapitre seront énumérés les différents outils et moyens nécessaires au développement et tests de notre application durant toute la phase pratique de cette dernière, ensuite nous verrons une présentation illustrative des importantes parties de l'application accompagnée d'explications claires sur leurs modes de fonctionnement.

## **II. Outils nécessaires au développement :**

### **II.1. Android ADT Bundle**

Android ADT Bundle nous offre tous un « atelier » pour le développement d'application Android, il est composé de :

- ✚ Eclipse + ADT Eclipse plugin,
- ✚ Android SDK Tools,
- ✚ Android Platform-Tools,
- ✚ Android Platform (la dernière version),
- ✚ Android Virtual Device pour l'émulateur.

#### **II.1.1. Téléchargement**

L'ADT Bundle fournit tout ce dont on a besoin pour commencer à développer des applications, y compris une version de l'IDE Eclipse avec haut-ADT (Android Developer Tools) pour rationaliser le développement d'une application Android. Le téléchargement et la configuration d'environnement de développement Android se fait en une seule étape.

L'Android ADT Bundle est disponible sur le lien suivant :

<http://developer.android.com/sdk/index.html>.


#### **II.1.2. Configuration de l'ADT Bundle**

Après avoir téléchargé l'ADT, suivez les étapes suivantes pour le configurer :

- ❖ Décompressez le fichier ZIP (appelé adt-bundle-<os\_platform>. Zip) et enregistrez-le dans un endroit approprié, comme un répertoire "développement" dans votre machine
- ❖ Ouvrez le adt-bundle-<os\_platform> / eclipse / et lancez Eclipse, la figure ci-dessous vous apparaîtra.



Fig.19 : fenêtre de lancement d’Eclipse

- ❖ L'IDE est chargé avec l'Android Developer Tools plug-in et le SDK est prêt pour l'utilisation.
- ❖ Après avoir lancé Eclipse, vous choisissez cette icône  dans la barre d’outils pour lancer l’Android SDK Manager, la fenêtre ci-dessous sera apparue :

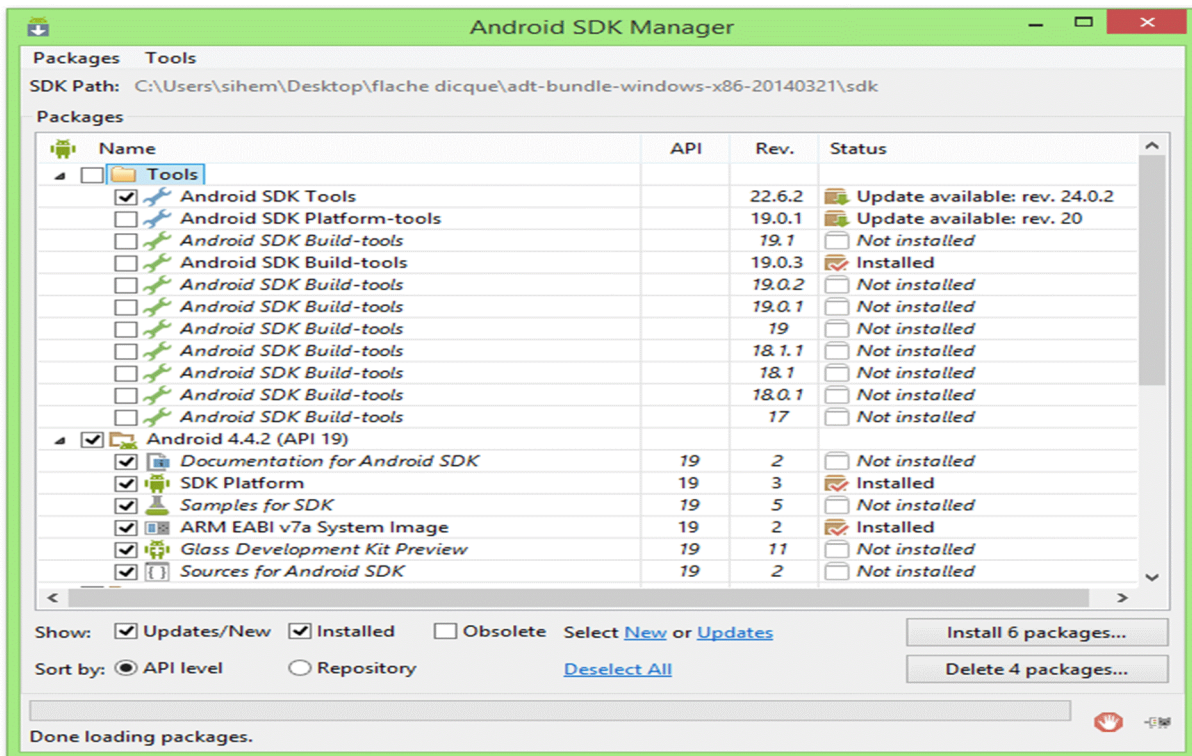


Fig.20 : Capture d’écran de la fenêtre d’Android SDK Manager

- ❖ Vous cochez « Tools » et « Extras » ainsi que la version d'Android que vous souhaitez utiliser, la version « 4.4.2 » dans notre cas. Ensuite vous cliquez sur le bouton « Install <nombre> packages » en bas.
- ❖ Vous acceptez la licence « Accept License» pour tous les packages sélectionnés puis vous validez par le bouton « Install » tel qu'expliqué à la figure ci-dessous :

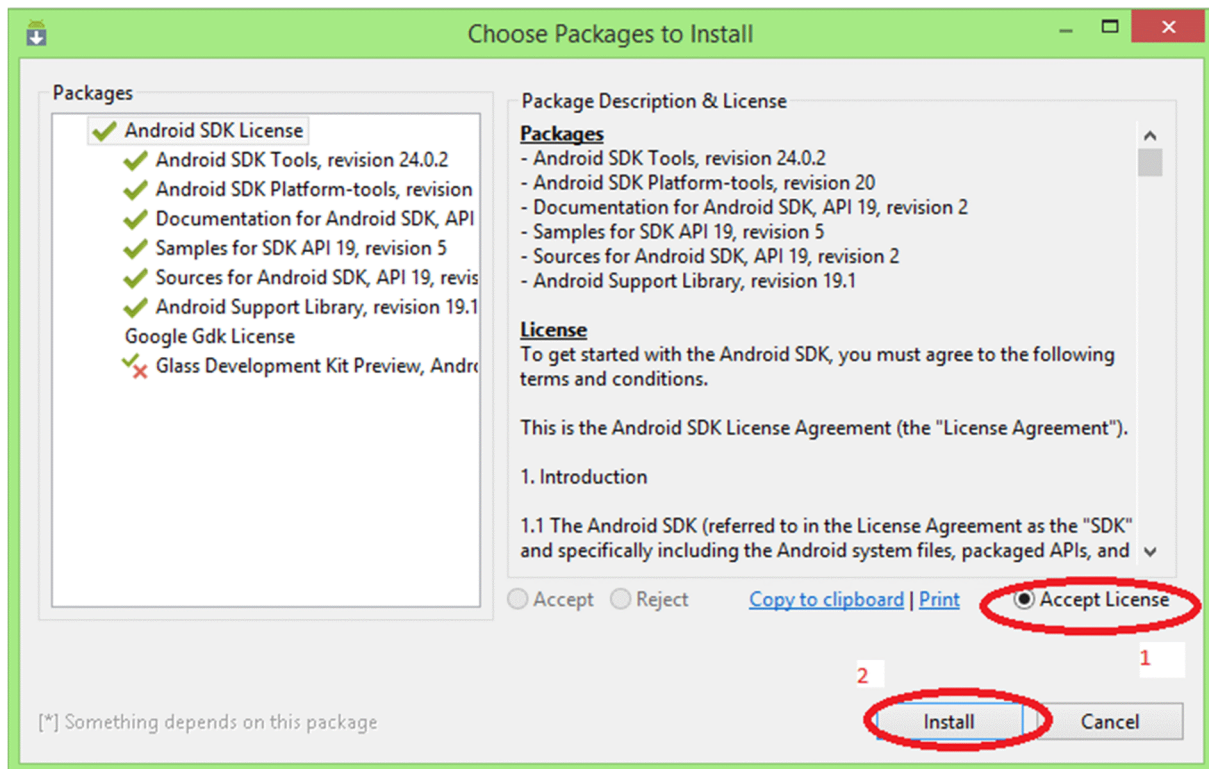


Fig.21 : Capture d'écran de la fenêtre de validation de téléchargements

### III. Création d'un AVD :

#### III.1. Définition

L'émulateur d'Android est un outil de test et de débogage d'application Android qui fournit une connexion réseau complète, une simulation d'envoi et de réception d'appels et de SMS.

C'est une implémentation de la machine virtuelle Dalvik, faisant de celle-ci une plateforme exécutant les applications Android comme le fait n'importe quel téléphone Android (Téléphone réel).

Intégré au Plugin ADT Eclipse, l'Emulateur est lancé automatiquement avec l'AVD sélectionné lors d'une exécution ou débogage.

Hors Eclipse l'émulateur peut s'exécuter via Telnet et le contrôler depuis une console :  
emulator -avd<avd\_name>.

L'émulateur n'implémente pas toutes les caractéristiques des matériels mobiles supportées par Android.

### III.2. Création

Afin de tester notre application, nous allons utiliser l'émulateur Android. Il faudra donc créer un Android Virtual Device (AVD), Un AVD décrit les paramètres systèmes et les composants de notre émulateur.

Pour créer un AVD :

- ✚ Lancer Eclipse.
- ✚ Aller sous « Window->Android SDK and AVD Manager » et sélectionner« Android Virtual Device Manager », ou bien choisir cette icône dans la barre d'outils.
- ✚ La fenêtre ci-dessous sera apparue.

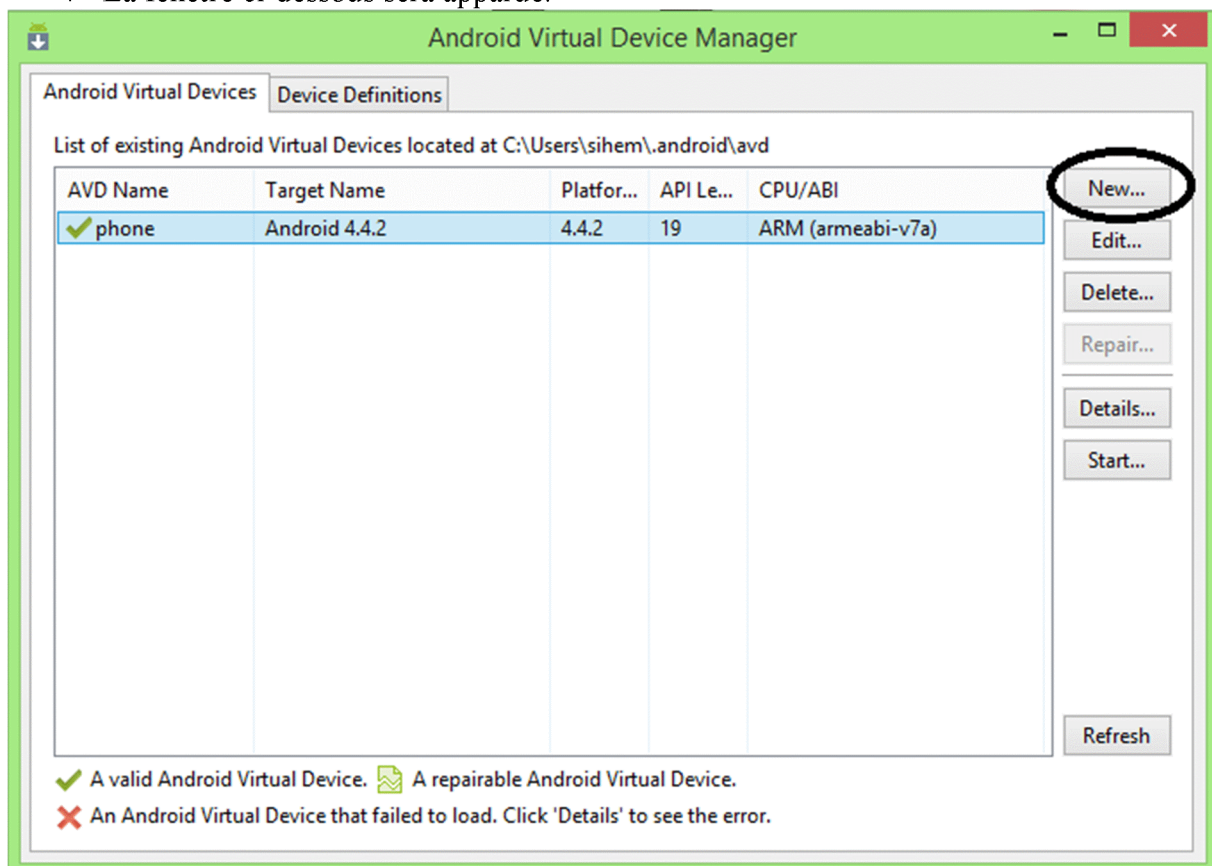
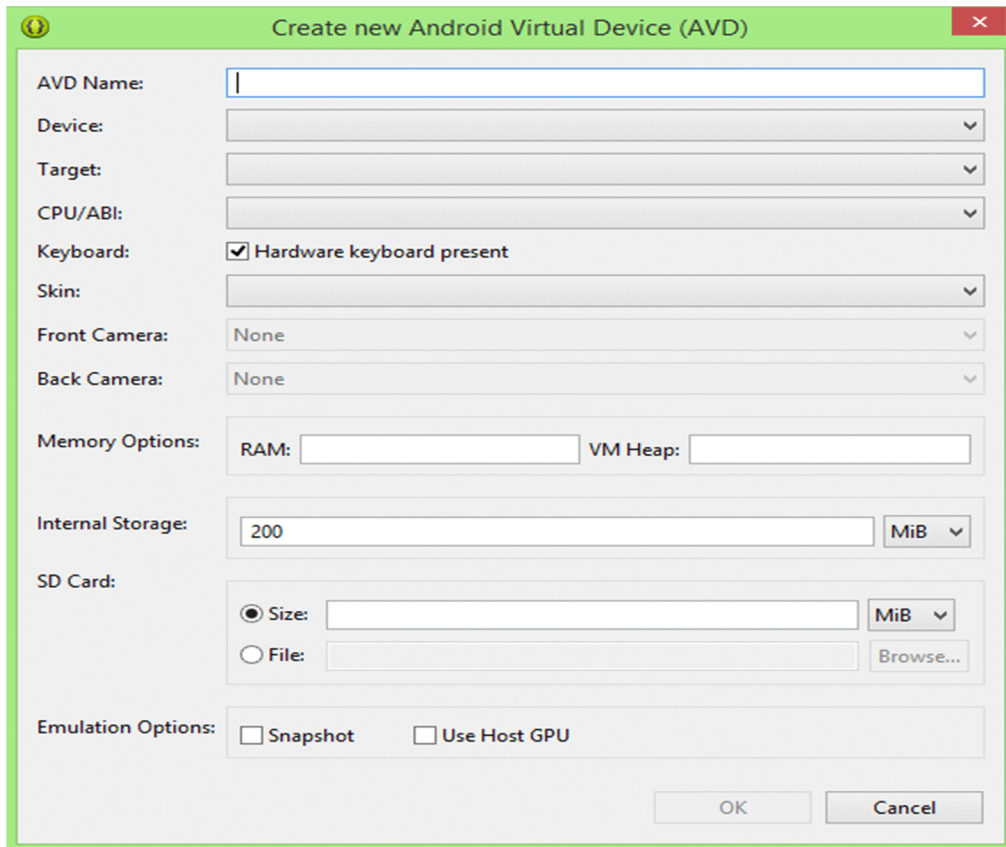


Fig.22 : Capture d'écran de la fenêtre Android Virtual Device Manager.

- ✚ Après un clic sur « New » encercler dans la figure ci-dessus, la figure suivante sera apparue :



**Fig.23: Capture d'écran de la fenêtre « Create new AVD »**

- ✚ Donner un nom pour l'AVD et choisir l'appareil « Device » et la version cible « Target».
- ✚ Cliquer sur « OK » et l'AVD sera créé.
- ✚ Sélectionner la ligne de l'AVD créée parmi la liste des émulateurs disponibles dans la fenêtre Android Virtual Device Manager et cliquer sur « Start »
- ✚ L'AVD se lancera comme le montre la figure suivante :



**Fig.24: Capture d'écran de la fenêtre d'un AVD.**

#### ***IV. Création d'un nouveau projet :***

##### **IV.1. Les étapes de création d'un nouveau projet**

Après avoir créé un émulateur Android, nous passons à la création d'un nouveau projet sous Eclipse.

Nous lançons Eclipse et nous allons sous File -> New-> Android Application Project, vous obtenez la figure suivante :

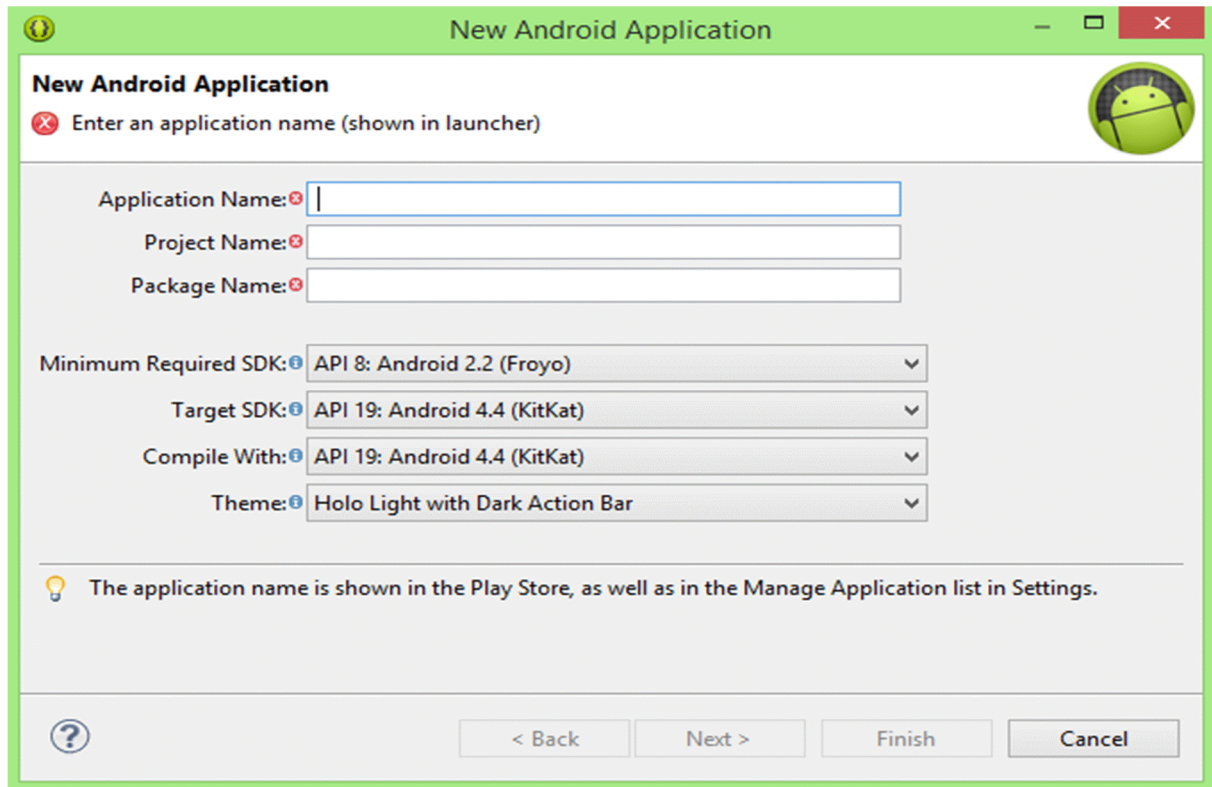


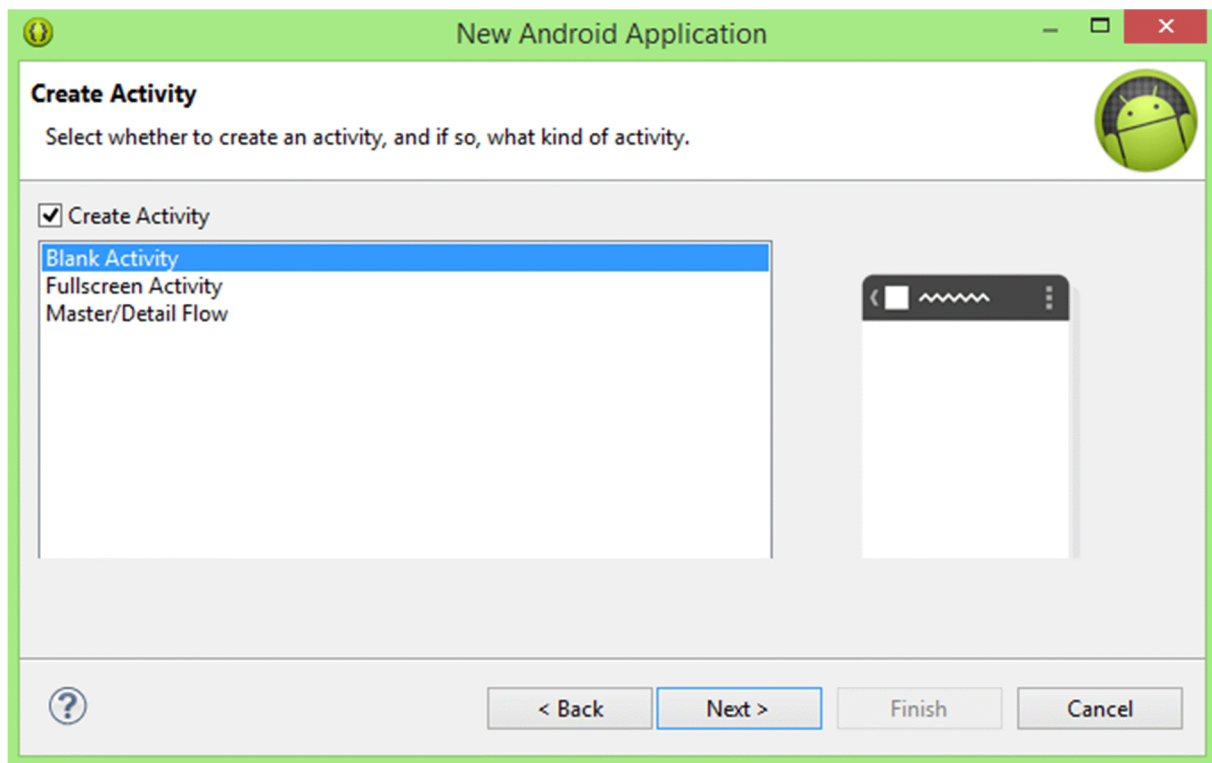
Fig.25 : Capture d'écran de la fenêtre Création d'un nouveau projet.

Vous remplissez les champs suivants :

- + « **Project Name** » : est le nom de votre projet pour Eclipse.
- + « **Application Name** » : le vrai nom de notre application.
- + « **Package Name** » : package où l'application sera sauvegardé.
- + « **Minimum Required SDK** » : est la version minimale pour laquelle votre application est destinée.
- + « **Target SDK** » : est la version maximale sous laquelle votre application fonctionne.
- + « **Compile With** » : la version du SDK avec laquelle sera compilée l'application, c'est la même version que celle de l'AVD créée.
- + « **Theme** » : c'est le thème de l'application, puisque aucun des thèmes proposés n'est disponible, le seul choix possible est « None » qui ne signifie aucun thème.

Ensuite vous cliquez successivement trois fois sur « **Next>** » vous aurez trois fenêtres dans lesquelles il est préférable de laisser tous les champs tels qu'ils sont par défaut.

La quatrième fois que vous cliquez sur « **Next>** » vous obtiendrez la fenêtre suivante :



**Fig.26: Capture d'écran de la dernière fenêtre de création d'un nouveau projet.**

« **Activity Name** » : permet d'indiquer le nom de la classe Java qui contiendra votre activité, ce champ doit donc respecter la syntaxe Java standard.

« **Layout Name** » : renseignera sur le nom du fichier qui contiendra l'interface graphique qui correspondra à cette activité.

« **Navigation Type** » : il permet de définir facilement comment s'effectueront les transitions entre plusieurs activités, mais il est préférable de le laisser tel qu'il est par défaut. Vous cliquez sur « **Finish** » et le projet sera créé selon une certaine structure.

## **IV.2. L'arborescence du projet**

Voici le résultat de la création de votre projet et l'arborescence de ce dernier

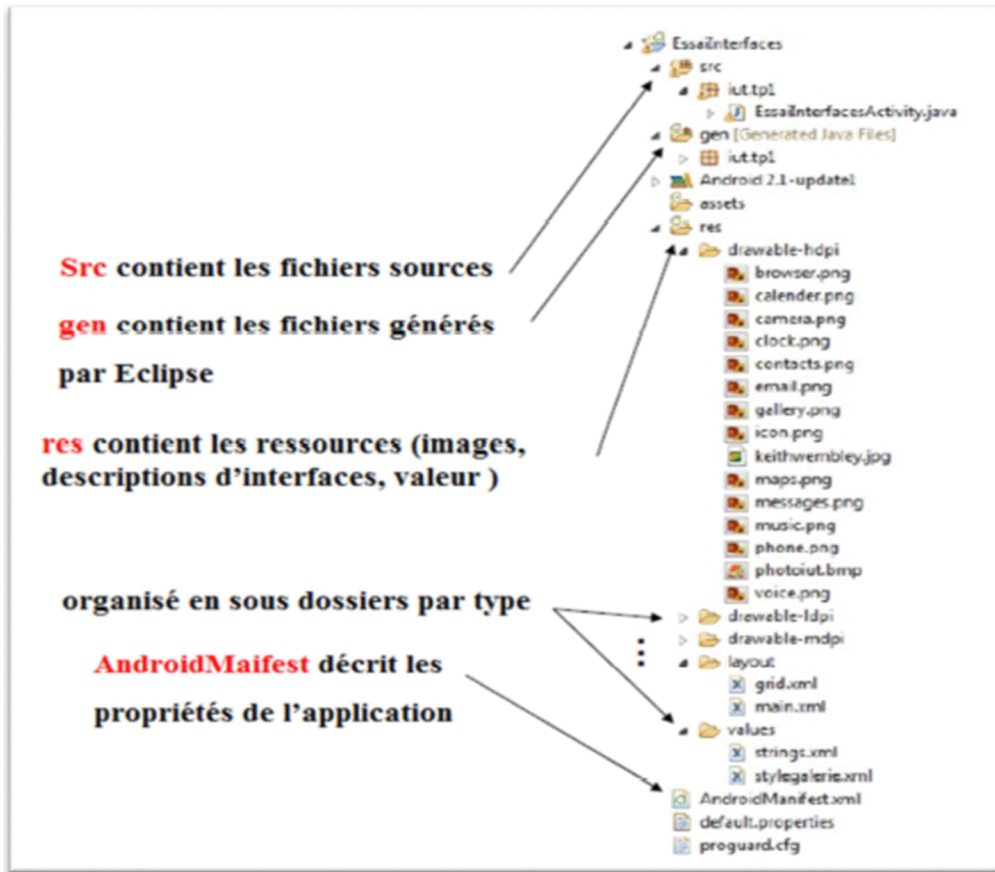


Fig.27: structure d'un projet Android.

V. Stockage de données :

Dans Framework Android les ressources sont des fichiers organisés d'une manière particulière de façon à ce qu'Android sache quelle ressource utiliser pour s'adapter au matériel sur lequel s'exécute l'application.

On découvre les ressources à travers une hiérarchie particulière de répertoires. Toutes les données de l'application sont sauvegardées et organisées selon cette hiérarchie.

Les images sont classées selon leurs tailles dans les sous répertoires « res>drawable », les vidéos utilisées sont stockés dans le sous répertoire « res>raw », Les données textuelles tels que poèmes et cours sont saisies dans les fichiers XML correspondants dans le sous répertoire «res>Layout », la base de données SQLite est constitués d'une seule table pour sauvegarder les scores de l'utilisateur, elle est composée de quatre attributs comme est illustré à la figure suivante :

Score
Id
Nom
Niveau
Note

Fig.28 : Modèle de la table de données score.

- ✚ **Id** : c'est un entier auto-incrément pour l'identification de chaque score.
- ✚ **Nom** : c'est une chaîne de caractère désignant le nom saisi par l'utilisateur.
- ✚ **Niveau** : c'est une chaîne de caractères qui désigne le niveau du test effectué par l'utilisateur, peut avoir pour valeur : « Facile », « Moyen », « Difficile ».
- ✚ **Note** : c'est un entier qui désigne la note obtenue par l'utilisateur dans un test joué.

A chaque Activité créée (.java) sous le répertoire « src » correspond un Layout (.xml), ces deux sont connectées entre elles par l'intermédiaire de la classe « R.java » située dans le répertoire « gen » qui utilise les identifiants des interfaces pour réaliser cette connexion, les activités définissent les actions et événements qui permettent une interaction effective entre l'utilisateur et l'application,

Les figures suivantes illustrent la manière dont les données textuelles sont stockées dans les fichiers « .xml » :

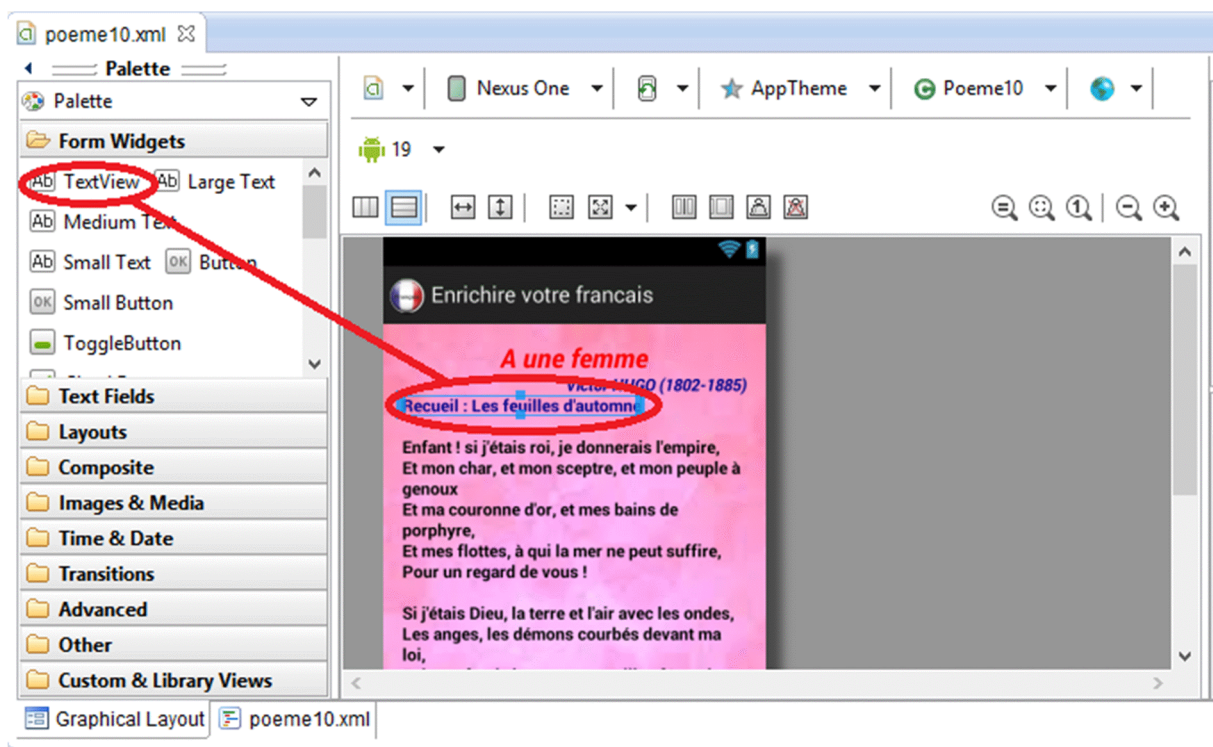


Fig.29 : capture d'écran d'un fichier « .xml ».

On peut utiliser la palette pour insérer des widgets différents (TextView, Button, ListView, etc.) dans la partie « GraphicalLayout », ou bien, pour plus de précision vous pouvez directement manipuler le code « xml » comme on l'aperçoit dans la figure suivante :

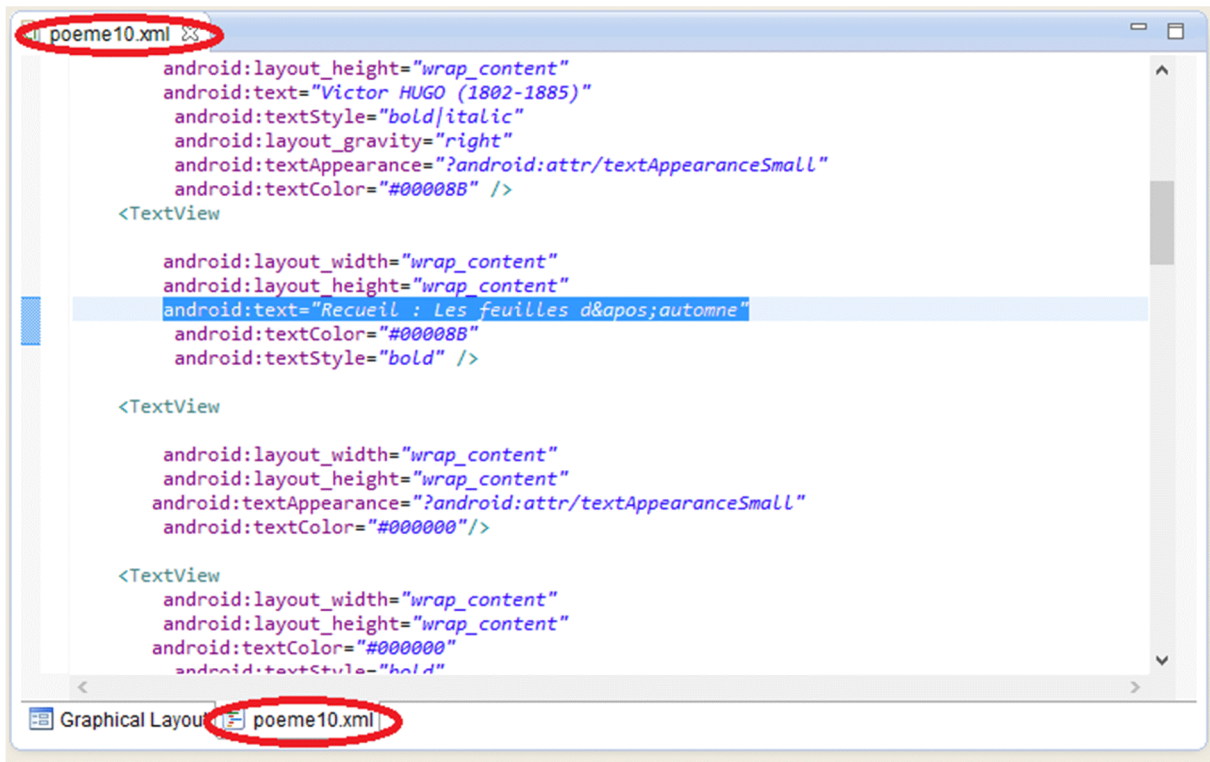


Fig.30 : Capture d'écran d'un code « xml ».

Les fichiers « .xml » sont très importants dans la programmation sous Android, ils permettent de diminuer le nombre de lignes dans code « .java », qui se chargera juste de traiter les évènements.

La figure suivante correspond au code « xml » de la figure précédente d'un poème :



Fig.31 : capture d'écran d'un code Java.

## VI. Le mode de fonctionnement de l'application :

### VI.1. Lancement de l'application

Pour lancer l'application « Enrichir votre français » dans le terminal mobile, vous choisissez dans le menu principal son icône spécifique, comme le montre cette figure :

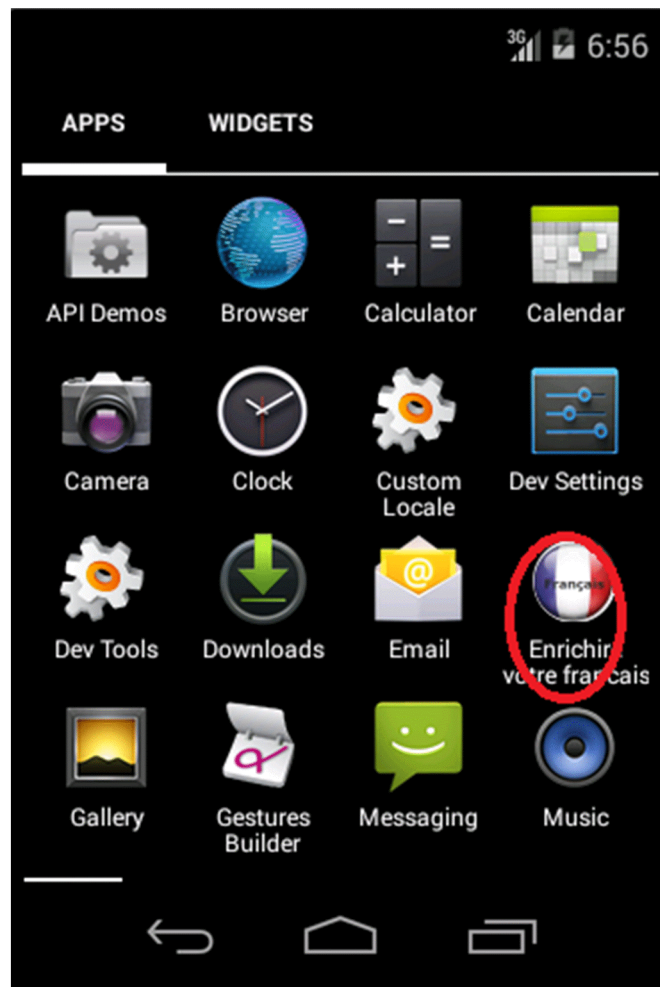


Fig.32 : l'icône de l'application dans le Menu de principal.

### VI.2. La fenêtre principale de l'application

Cette fenêtre nous offre deux boutons :

- ✚ Le premier permet d'ouvrir le menu principal de l'application.
- ✚ Le deuxième permet de quitter l'application.

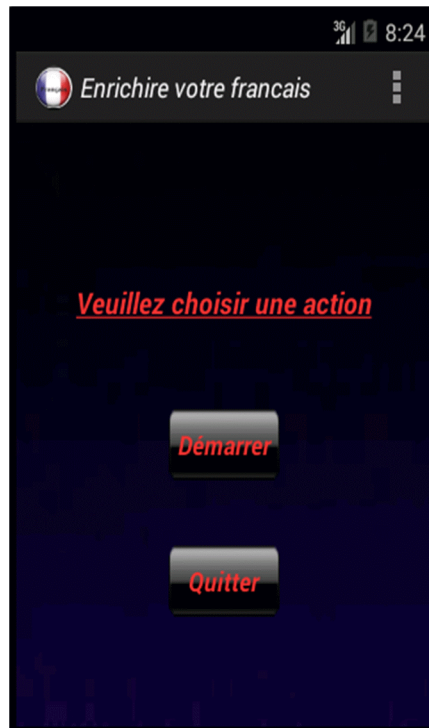


Fig.33 : Fenêtre principale de l'application.

### VI.3. Le menu principal de l'application

En appuyant sur le bouton « Démarrer », on aura accès au menu principal

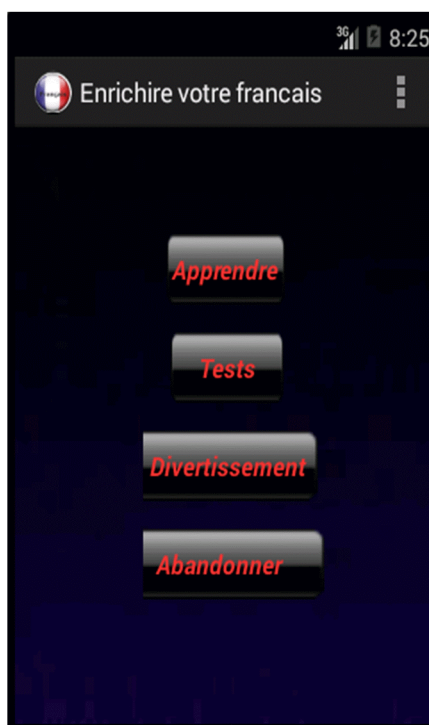


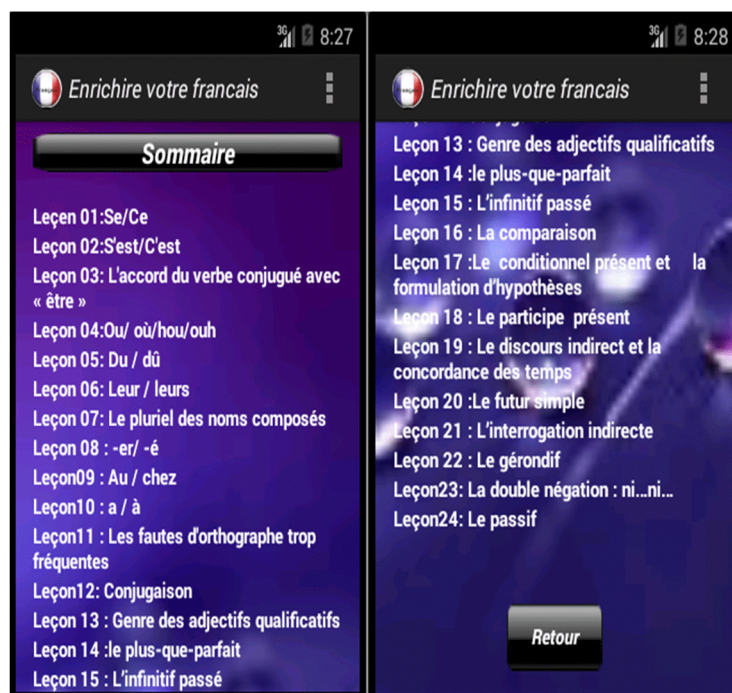
Fig.34 : Menu principal de l'application.

Cette fenêtre est composée de quatre boutons :

- ✚ Le premier nous mène vers les cours disponibles.
- ✚ Le deuxième nous mène vers le menu principal des tests.
- ✚ Le troisième nous mène vers le menu principal de la partie divertissement.
- ✚ Le quatrième nous permet de quitter l'activité en cours et revenir à la précédente.

#### VI.4. La fenêtre « sommaire des cours »

En appuyant sur le bouton « apprendre », on a l'accès à cette fenêtre



**Fig.35 : Le sommaire des cours.**

L'utilisateur choisi un cours dans le sommaire et clique ci-dessus.

A la fin de la fenêtre il y a un bouton « Sommaire » qui nous permet de revenir en arrière.

✚ **Exemple de cours :**

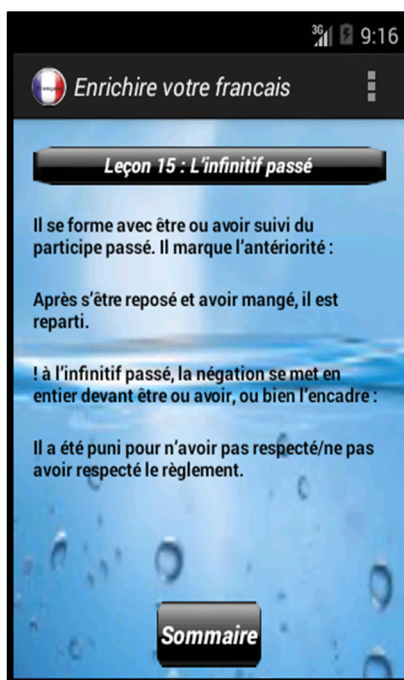


Fig.36 : Exemple d'un cours.

Pour suivre un cours long il suffit de défiler de haut en bas, et cliquer sur le bouton «Sommaire » pour revenir à la liste des cours disponible.

### VI.5. La fenêtre « Menu des tests»

En appuyant sur le bouton « Tests », on aura accès à cette fenêtre

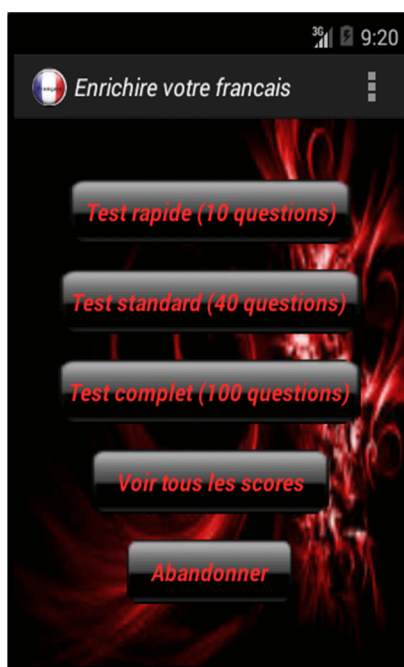


Fig.37 : Menu des tests.

Elle est composée de cinq boutons :

- ✚ Le premier nous mène vers la partie test rapide.
- ✚ Le deuxième nous mène vers la partie test standard.
- ✚ Le troisième nous mène vers la partie test complet.
- ✚ Le quatrième nous permet de voir tous les scores obtenus.
- ✚ Le dernier permet de quitter l'activité en cours et revenir à la précédente.

• **Exemple de « test rapide » :**

Nous proposons une liste de dix (10) questions dont les réponses sont au choix.

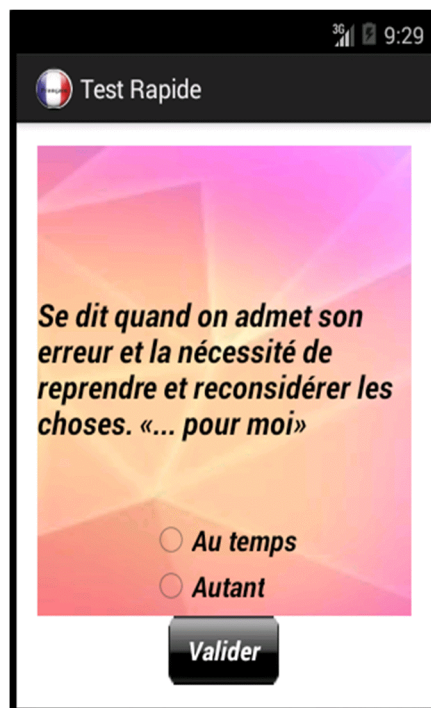


Fig.38 : Exemple d'une question.

### VI.6. La fenêtre « sauvegarder le score »

Une fois que le résultat est affiché test, le système nous propose de sauvegarder le score ou bien d'abandonner et faire un retour en arrière vers le « Menu des tests » comme l'illustre la figure suivante :



**Fig.39 : Affichage de score.**

Si l'utilisateur ne saisit pas de nom, un message sera afficher pour lui demander de le faire, sinon le score ne sera jamais sauvegardé.

Dans cette fenêtre, l'utilisateur peut aussi voir tous les scores sauvegardés dans la base de données.

### **VI.7. La fenêtre « Voir tous les scores »**

Grâce à cette fenêtre, l'utilisateur peut revoir l'intégralité de ces scores, et ainsi supprimer un, plusieurs ou tous les scores.

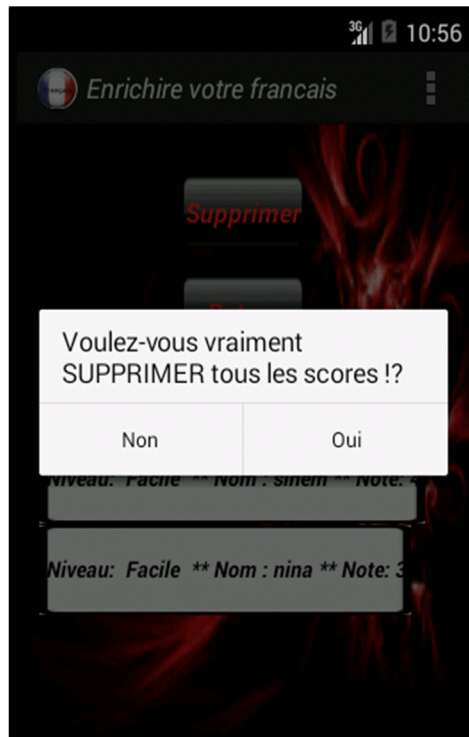


Fig.40 : Fenêtre « Voir tous les scores ».

### VI.8. La fenêtre « Menu divertissement »

On appuyant sur le bouton « Divertissement », on aura accès à cette fenêtre

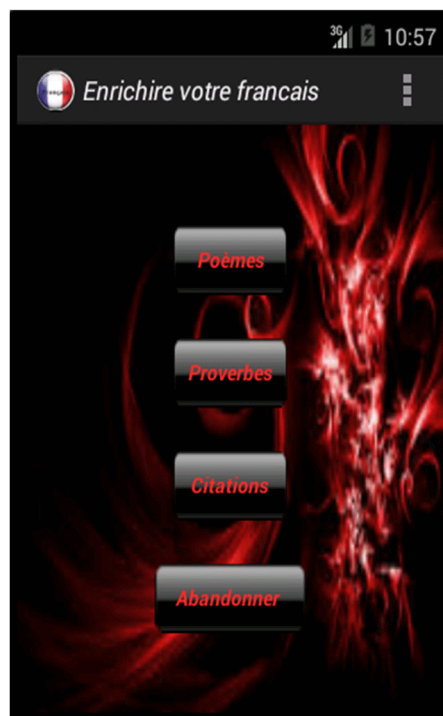


Fig.41 : Menu divertissement.

Le menu est composé de quatre boutons :

- ✚ Le premier nous donne accès au « sommaire des poèmes ».
- ✚ Le deuxième nous donne accès aux différentes « catégories proverbes ».
- ✚ Le troisième nous donne accès aux « citations ».
- ✚ Le dernier nous permet d'abandonner l'action en cours.

### VI.9. La fenêtre « poèmes »

Pour lire un poème il suffit de cliquer dessus dans le sommaire des poèmes disponibles, et de défiler de haut en bas pour lire la suite pour les poèmes longs.



**Fig.42 : Sommaire des poèmes.**

La figure ci-dessous nous donne un exemple d'un poème court.

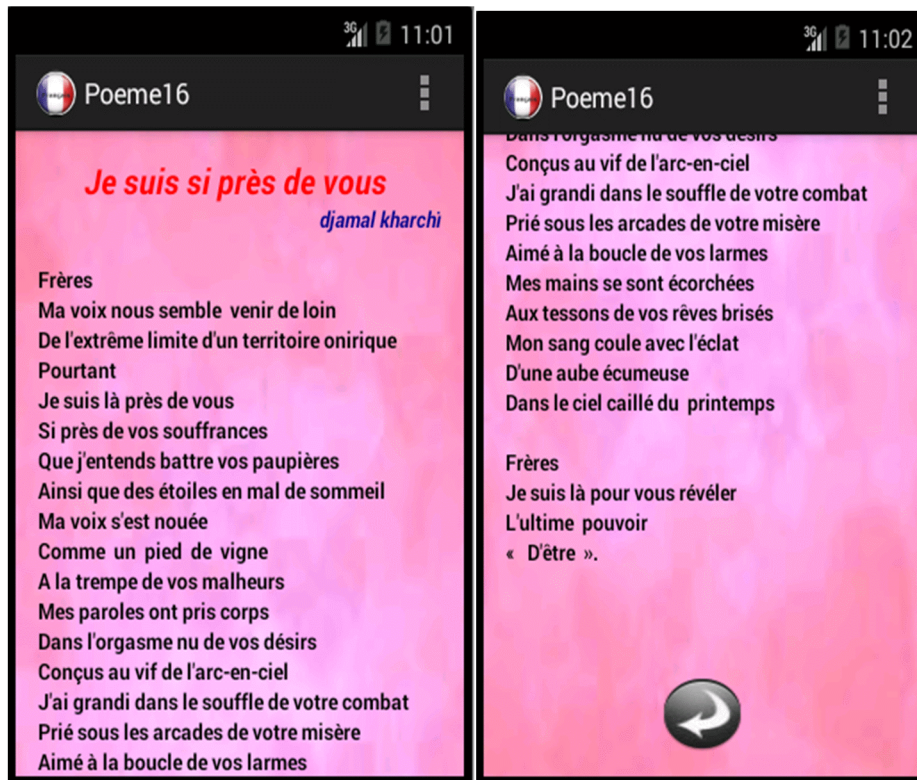


Fig.43 : Exemple d'un poème.

### VI.10. La fenêtre « catégorie proverbes »

Elle est composée de neuf catégories, et de deux boutons « Retour » qui permet d'abandonner la tâche en cours.

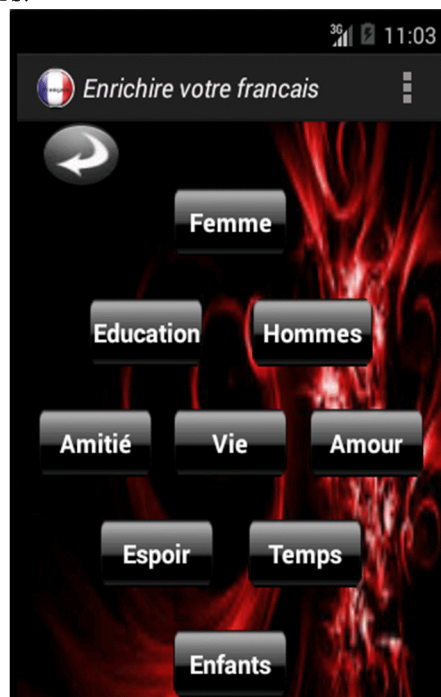


Fig.44 : Menu proverbes.

Pour lire un proverbe il suffit de choisir une catégorie et appuie dessus, puis défilé de haut en bas pour lire les différents proverbes.

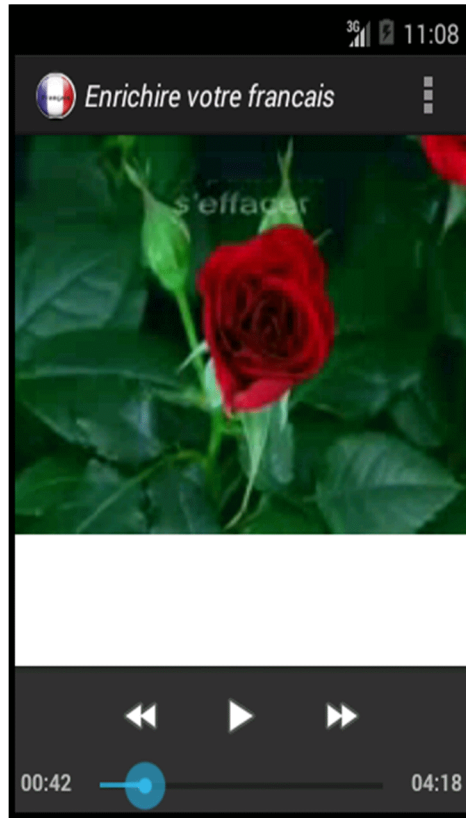
✚ Exemple de proverbes :



Fig.45 : Exemple de proverbes.

### VI.11. La fenêtre « citations »

Elle nous permet de suivre les citations en images animées sous forme d'une vidéo accompagnée d'une musique douce, pour la lire il suffit de choisir dans le menu divertissement le bouton citations comme le montre cette figure :



**Fig.46 : Fenêtre citations.**

## *VII. Conclusion :*

Dans ce chapitre nous avons présenté les différents outils que nous avons utilisés pour développer notre application, ensuite on a listé les étapes créatives d'un « AVD » et d'un nouveau projet.

Et enfin, on a clôturé par des captures d'écran des différentes vues de notre application.

## *CONCLUSION GENERALE*

Pour arriver à réaliser une application compréhensible et utilisable, qui est capable de répondre aux utilités de l'utilisateur, nous avons eu besoin de mieux s'adapter avec l'environnement des applications mobiles.

Ainsi, on a entamé un minutieux travail de recherches sur les applications mobiles et le système d'exploitation Android ainsi que le langage Java et son environnement de développement, et tous les outils et concepts dont nous avons besoins pour initier la démarche de notre projet. Ensuite nous avons entamé la partie pratique par une analyse soignée des nécessités capitales de l'application, pour y parvenir à une conception délicate, permettant une réalisation particulière et signifiante.

Notre application répond à des besoins avantageux, elle est bénéfique du côté culturel et éducatif, elle offre des moyens à la fois de divertissement et d'apprentissage. Assez utile et ergonomique avec des performances remarquables et un design plaisant, elle est aussi réflexive et d'usage commode.

Mais, elle reste incomplète à cause de la contrainte de la durée de développement limitée. Cependant, il y a des possibilités d'extensions susceptibles par ajouts de données (mises à jour) et modifications d'interfaces, etc.

La réalisation de ce projet nous a permis une bonne pratique en programmation, notamment en Java standard, et d'une découverte favorable de Java sous Android, le langage de balisage XML et SGBD SQLite, et plus particulièrement un essor important vers l'univers de développement mobile spécialement Android.

## RÉFÉRENCES BIBLIOGRAPHIQUES

- [01] <http://www.clubic.com/os-mobile/android/actualite-615662-google-samsung-accordentharmoniser-android-apps-pre-installees.html>
- [02] <http://www-igm.univ-mlv.fr/~dr/XPOSE2011/SDKAndroid/android.html>
- [03] <http://issatso.0fees.net/cours/cours.pdf> (Le jeudi 5 décembre 2013)
- [04] <http://www.igen.fr/iphone/android-finit-2013-sur-une-progression-generale-109854>
- [05] <http://fr.wikipedia.org/wiki/>
- [06] Jeff FRIESEN, « *Learn Java for Android Development* ». Deuxième édition APRESS, 2013.
- [07] <http://general.developpez.com/edi/>
- [08] <http://searchsoa.techtarget.com/definition/Eclipse>
- [09] <http://developer.android.com/tools/help/adt.html>
- [10] [http://www.webopedia.com/TERM/A/Android\\_SDK.html](http://www.webopedia.com/TERM/A/Android_SDK.html)

### ***Livre :***

- Serge Ungar, développer une application ANDROID (ed.ENI).
- L'art du développement ANDROID (Edition PEARSON)

### ***Mémoire :***

- Création d'application sous ANDROID gestion d'une base de données via code QR.
- Apprentissage interactif de l'anglais via une application ANDROID 2012/2013.
- Navigation GPS sous Android 2011/2012.
- Réalisation d'une application Android de navigation par GPS en utilisant des cartes OSM 2012/2013.