

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE**

*MINISTERE DE L'ENSEIGNEMENT SUPERIEURET DE LA RECHERCHE SCIENTIFIQUE*

*Université Mouloud Mammeri Tizi-Ouzou*

*Faculté De Génie Électrique Et Informatique*

*Département Informatique*

## *Mémoire de fin d'étude*

*En vue de l'obtention du diplôme de Master*

*Option : Conduite de projet informatique*

**Thème :**

**Conception et réalisation d'une application  
mobile sous Android**

**Cas : Jeu SUDOKU**

**Dirigé par : Mr. DIB**

**Réalisé par :**

**Mlle. BENMOUMENE Loubna**

**Mr. NADJEM Mounir**



**Promotion 2013-2014**

## *Remerciements*

*Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et béni par sa volonté durant toute cette période.*

*Nous tenons à exprimer notre profonde gratitude à notre encadreur Mr DIB Ahmed, pour tout ce qu'il nous a apporté comme aide, connaissances et conseils pour l'accomplissement de ce travail.*

*Nos vifs remerciements vont également aux membres de jury qui nous font l'honneur de juger notre travail et qui ont pris la peine de lire ce mémoire pour juger son contenu.*

*Nous aimerions aussi remercier nos familles, nos amis, ainsi que tous ceux qui ont contribué de prêt ou de loin à la réalisation de ce travail.*

## Table des matières

Table des matières .....	I
Liste des figures .....	IV
Liste des tableaux .....	V
Introduction Générale .....	1

### Chapitre I : Les systèmes d'exploitation mobiles

I.1. Introduction .....	4
I.2. les différents systèmes d'exploitation mobiles .....	5
I.2.1. BlackBerry .....	5
I.2.2. Symbian OS .....	6
I.2.3. iPhoneOS (iOS) .....	7
I.2.4. Windows Phone .....	8
I.3. Comparaison entre les différents Systèmes d'exploitation mobiles .....	9
I.4. Conclusion.....	10

### Chapitre II : Présentation d'Android

II.1.Introduction.....	12
II.2.Description.....	13
II.3.Open Handset Alliance .....	14
II.4.Bugdroïde .....	14
II.5.Historique des versions d'Android.....	15
II.6.Fonctionnalités d'Android .....	16
II.7.Architecture Android.....	18
II.7.1.Applications .....	22
II.7.2.Framework de développement .....	22
II.7.3.Bibliothèques.....	22

II.7.4.Android Runtime .....	24
II.7.5.Linux Kernel .....	24
II.8.Les composants principaux d'une application .....	25
II.8.1.Activités .....	25
II.8.2.Services .....	25
II.8.3.Broadcast receivers .....	25
II.8.4.Content providers .....	25
II.9.Cycle de vie d'une activité.....	26
II.10.Développement d'une application Android .....	27
II.11.Différents équipements utilisant Android .....	28
II.12.Les outils que fournit le SDK.....	31
II.13.Conclusion .....	34

## **Chapitre III : Analyse et conception**

III.1.Introduction .....	36
III.2.Les Jeux de réflexion.....	37
III.2.1.L'intelligence implémentée dans les algorithmes .....	38
III.3.Présentation de l'UML .....	39
III.3.1.Modélisation avec l'UML.....	40
III.3.2.La démarche de modélisation avec l'UML .....	40
III.4.Analyse.....	41
III.4.1.Identification des besoins.....	41
III.4.2.Identification des acteurs .....	41
III.4.3.Spécification des tâches .....	41
III.4.4.Spécification des scénarios .....	42
III.4.5.Les cas d'utilisation.....	42
III.4.5.1.Description des cas d'utilisation.....	42
III.5.Conception.....	44
III.5.1.Les diagrammes des cas d'utilisation .....	44
III.5.2.Les diagramme de séquences .....	46
III.5.3.Les diagramme d'activités .....	48
III.5.4.Les diagramme de classe .....	51

III.6.Conclusion.....	53
-----------------------	----

## Chapitre IV : Réalisation

IV.1.Introduction .....	55
IV.2.Environment de travail .....	56
IV.2.1.Environment matériel .....	56
IV.2.2.Environment cible .....	56
IV.2.3.Environment logiciel .....	56
IV.2.3.1.Système d'exploitation .....	56
IV.2.3.2. Langage de programmation.....	56
IV.2.3.3.Eclipse .....	57
IV.2.3.4. IDE Eclipse .....	57
IV.3.Prise en main de l'environnement Android.....	58
IV.3.1.Software Development Kit (SDK).....	59
IV.3.2. Le plugin ADT pour Eclipse.....	60
IV.3.3.Android.....	61
IV.4.Mode de fonctionnement de l'application.....	62
IV.4.1.Le menu principal de l'application.....	62
IV.4.2.Interface difficulté.....	63
IV.4.3.Interface de jeu.....	64
IV.4.4.Interface de description .....	65
IV.5. Conclusion.....	66
Conclusion Générale.....	67
Annexe .....	68
Bibliographie .....	69
Netographie .....	70

## Liste des figures :

### Chapitre I : Les systèmes d'exploitation mobiles

Figure I.1: Smartphone BlackBerry. ....	6
Figure I.2: Écran d'accueil de Nokia (dernière version de Symbian).....	7
Figure I.3: Smartphone iPhone 5.....	8
Figure I.4: Smartphone Samsung. ....	9
Figure I.5: Performances des Systèmes d'Exploitation mobiles [N1]. ....	10
Figure I.6: Parts de marché des Systèmes d'Exploitation mobiles [N2]. ....	10

### Chapitre II : Présentation d'Android

Figure II.1: Le personnage Bugdroid. ....	15
Figure II.2: Evolution des versions d'Android. ....	15
Figure II.3: Architecture d'Android [N3].....	19
Figure II.4: Architecture d'Android « Kernel linux » [4]. ....	19
Figure II.5: Architecture d'Android « Bibliothèques C/C++ » [4]. ....	20
Figure II.6: Architecture d'Android « Android runtime » [4]. ....	20
Figure II.7: Architecture d'Android « Application framework » [4].....	21
Figure II.8: Architecture d'Android « Application » [4]. ....	21
Figure II.9: Architecture de ces médiathèques.....	23
Figure II.10: Transformation d'un fichier .java à un fichier .dex. ....	24
Figure II.11: Cycle de vie d'une activité. ....	26
Figure II.12: Une application sur Android.....	28
Figure II.13: Samsung Galaxy S3 .....	30
Figure II.14: Sony Xperia. ....	30
Figure II.15: HTC Legen. ....	30
Figure II.16: Galaxy Nexus. ....	30
Figure II.17: Interface du simulateur Android. ....	32

### Chapitre III : Analyse et conception

Figure III.1: Exemple de grille de départ du jeu sudoku. ....	38
Figure III.2 : La démarche adoptée pour la modélisation. ....	40
Figure III.3: Cas d'utilisation « Accéder à un nouveau jeu en spécifiant la difficulté du jeu ». ....	43
Figure III.4: Cas d'utilisation « Accéder à un jeu non terminé».....	43
Figure III.5: Cas d'utilisation « Accéder à un jeu non terminé».....	43
Figure III.6: Diagramme de cas d'utilisation le plus général.....	44
Figure III.7: Diagramme de cas d'utilisation générale. ....	45
Figure III.8: Diagramme de cas d'utilisation du jeu en lui-même. ....	45

Figure III.9: Diagramme de séquence de cas d'utilisation « Accéder à un nouveau jeu en spécifiant la difficulté du jeu ».....	46
Figure III.10: Diagramme de séquence de cas d'utilisation « Accéder à un jeu non terminé».....	47
Figure III.11: Diagramme de séquence de cas d'utilisation « Accéder à la description du jeu».....	47
Figure III.12: Diagramme de séquence de cas d'utilisation « Validation d'une grille ».....	48
Figure III.13: Diagramme d'activité de cas d'utilisation « Accéder à un nouveau jeu en spécifiant la difficulté du jeu ».....	49
Figure III.14: Diagramme d'activité de cas d'utilisation « Accéder à un jeu non terminé » ..	50
Figure III.15: Diagramme d'activité de cas d'utilisation « Accéder à la description du jeu ».	50
Figure III.16: Diagramme d'activité de cas d'utilisation « Validation d'une grille ».....	51
Figure III.17: Diagramme de classe « les deux classes de base ».....	52
Figure III.18: Diagramme de classe « Motif de conception composite pour les boîtes ».....	52
Figure III.19: Diagramme de classe « affichage des variables et fonction principale ».....	52

## Chapitre IV : Réalisation

Figure IV.1: Logo Eclipse et site web.....	57
Figure IV.2: plate-forme d'éclipse.....	58
Figure IV.3: Portail des développeurs Android.....	59
Figure IV.4: Interface d'installation du SDK Android.....	60
Figure IV.5: Interface d'installation ADT.....	61
Figure IV.6: Menu principal.....	62
Figure IV.7: La liste des difficultés.....	63
Figure IV.8: Interface de jeu (la grille).....	64
Figure IV.9: Interface de description.....	65

## Liste des tableaux :

### Chapitre II : Présentation d'Android

Tableau II.1: Fonctionnalités d'Android.....	18
--	----

### Chapitre III : Analyse et conception

Tableau III.1: Spécification des tâches.....	41
Tableau III.2: Spécification des scénarios.....	42

# Introduction Générale

Les jeux d'entraînement cérébral comme tout domaine n'ont pas été épargnés par l'avènement des Nouvelles technologies.

D'autre part, l'utilisation de la téléphonie mobile s'est bien amplifiée et son développement s'est vu accroître dans le monde entier, en 2010, la moitié des habitants ont accès à cette technologie. Les Smartphones basés sur les systèmes d'exploitation propriétaire et libres (iOS, Android, Symbian...) s'intègrent dans notre paysage et dans nos usages d'une manière croissante pour devenir un accessoire qui nous ouvre une porte vers de nouveaux services.

L'Android est une nouvelle plateforme en code source ouverte. De plus, selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sûre et ouverte. Grâce à l'ouverture du code source et des APIs, les développeurs obtiennent la permission d'intégrer, d'agrandir et de remplacer les composants existants. Les utilisateurs peuvent adapter les applications à leur besoin. C'est pour cela qu'on a choisi la plateforme Android pour faire une étude approfondie et pour développer notre application.

Le but de notre travail est de développer une application de type jeu « Sudoku » sous Android pour les Smartphones.

Le Sudoku est un jeu introduit dans le monde, il y a une trentaine d'années, et qui a connu récemment un grand succès. Un certain nombre de jeux de réflexion basés sur la présence et la répétition (ou non) de chiffres dans une grille étaient déjà répandus, au dix-neuvième siècle. Les règles de ces jeux ne correspondaient cependant pas à celle du Sudoku.

Cet engouement est tel que des jeux de Sudoku sont présents dans les quotidiens, sur les chaînes de télévision, que des concours nationaux et internationaux sont organisés. Un grand nombre d'implémentations du jeu sont présentes sur média informatisé, que ce soit sous forme de programme indépendant (sur ordinateurs, consoles de jeux et même téléphones portables) ou présenté sur le web.

Pour mener à bien notre travail, nous avons opté pour une démarche qui s'étale sur quatre chapitres :

- ❖ Le premier chapitre s'intitule « les systèmes d'exploitation mobiles », Ce chapitre a pour objectif de voir sommairement les différents systèmes d'exploitation mobiles.
- ❖ Le deuxième chapitre s'intitule « état de l'art d'Android », Ce chapitre est une introduction à la plate-forme, aux outils et à la configuration de notre environnement de travail.

- ❖ Le troisième chapitre sous le nom « analyse et conception », est consacré à l'analyse et la conception de notre application, afin de réaliser l'application mobile « sudoku » sous Android, pour cela on a opté pour le langage UML étant le mieux adapté pour les applications mobiles.
- ❖ Le quatrième et le dernier chapitre « Réalisation et implémentation », comporte quant à lui la présentation de l'environnement dont lequel notre application a été réalisée, les outils utilisés et quelques interfaces de notre application.

Enfin, nous clôturerons ce modeste mémoire par une conclusion générale sur le travail accompli.

Chapitre I :

Les systèmes d'exploitation mobiles.

## I.1. Introduction

Les systèmes d'exploitation pour téléphones mobiles dite systèmes embarqués intéressent beaucoup les fournisseurs, les développeurs et les éditeurs d'application pour téléphones portables qui peuvent donc déployer des services pour des sociétés basés sur la technologie du téléphone mobile.

Actuellement, il existe plusieurs Systèmes d'Exploitation pour les mobiles comme Windows Mobile, iPhoneOS, Symbian, BlackBerry qui sont les systèmes propriétaires. De plus, il y a les plateformes libres au code source ouvert comme Moblin.org, Ubuntu MID Edition, Android, etc...

Chaque type de SE est souvent approprié à quelques modèles concrets de téléphones. Cependant, chaque type de Système a des avantages et des limitations. Il est donc très difficile de choisir la plateforme répondant à l'objectif d'une société ou d'une personne.

## I.2. les différents systèmes d'exploitation mobiles

### I.2.1. BlackBerry

BlackBerry a été créé par Research In Motion (RIM), une société Canadienne. Il a été développé pour le businessman car le mode de compression utilisé réduit le poids du message, ce qui facilite la synchronisation de ses courriers électroniques avec le serveur de messagerie électronique. Cette synchronisation s'effectue via le réseau de téléphonie mobile sur lequel l'appareil est connecté. Ainsi, envoyer un email apparaît à la simplicité d'envoyer un SMS ou un MMS. De plus, le fournisseur offre aux abonnés plusieurs services tels que : [1]

- Des fonctions push de messagerie et d'accès mobile pratiques.
- La technologie BlackBerry Connect permet aux appareils qui marchent sur les autres plateformes de se connecter avec BlackBerry Enterprise Server et BlackBerry Internet Server pour bénéficier des différents services.
- De plus Son système de messagerie électronique est meilleur car il a su développer la technologie de la pièce jointe, cette dernière peut avoir différentes extensions tel que : .zip, .html, .doc, .dot, .PDF, etc....

Tout cela fait de BlackBerry un “ Standard d'or ” pour la messagerie électronique, et plus récemment permet une synchronisation complète avec les messageries d'entreprise telles que Microsoft Exchange ou IBM Lotus Domino, d'où sa popularité croissante dans le monde, qui se voit au nombre croissant d'applications tierces développées même si leur nombre reste largement inférieurs par rapport aux applications tierces des autres systèmes.



**Figure I.1:** Smartphone BlackBerry.

## I.2.2. Symbian OS

SymbianOS est très populaire (particulièrement en Europe), c'est le leader mondial du système d'exploitation pour Smartphones, pourtant il n'existe presque pas aux Etats-Unis car il y a peu d'appareils sous Symbian qui soutiennent la CDMA (Code division multiple access) qui est la technologie principale de transmission de la voix là-bas.

Symbian est extrêmement puissant, il offre un OS de soutien au niveau de la plupart des fonctionnalités se trouvant dans Windows Mobile et BlackBerry. Il offre une plateforme flexible, ce qui veut dire que les constructeurs de téléphones mobiles peuvent facilement y ajouter leurs technologies et infrastructures. De plus, il est soutenu par les grandes manufactures de l'industrie mobile comme Sony Ericsson, Motorola, et Nokia. [2]

Ce dernier étant constructeur numéro un des mobiles dans le monde et principal actionnaire de Symbian, cela garantit la position du Symbian dans le marché des systèmes d'exploitation pour les Smartphones. Néanmoins depuis quelques années le part de marche de Symbian diminue à cause de concurrence avec d'autres plateformes.

En janvier 2008, Nokia a annoncé des plans pour transformer Symbian en une plateforme en code source ouverte, espérant ainsi devenir un compétiteur de l'Android dans l'avenir, mais est revenue sur sa décision par la suite.



**Figure I.2:** Écran d'accueil de Nokia (dernière version de Symbian).

# Chapitre I : Les systèmes d'exploitation mobiles

## I.2.3. iPhoneOS (iOS)

iOS, anciennement iPhone OS, est le système d'exploitation mobile développé par Apple pour l'iPhone, l'iPod touch et l'iPad. Il est dérivé de Mac OS X dont il partage les fondations, le système d'exploitation occupe moins d'un demi-gigaoctet (Go) de la capacité mémoire totale de l'appareil, ce qui est plutôt un avantage étant donné que les iPhones ne disposent pas d'emplacement de carte SD. [1]

Le kit de développement iPhone (SDK) est disponible pour Mac OS X, et propose les outils nécessaires à la création d'une application pouvant tourner sous iOS. Si son téléchargement et son utilisation sont gratuits, la publication de telles applications requiert d'adhérer au programme des développeurs Apple, pour la somme de 99 \$ par an. Il n'en demeure pas moins que cette offre peut s'avérer intéressante pour bon nombre de développeurs, étant donnée la taille du marché créé par iOS.

En effet, Apple a annoncé, lors d'un événement musical le 9 septembre, avoir vendu 50 millions d'iPhone et d'iPods Touch. À titre informatif, les 40 millions d'appareils sous iOS n'avaient été dépassés que trois mois plus tôt, le 8 juin 2009. De surcroît, le portail App Store, dédié à l'exposition de toutes les applications développées pour ce système d'exploitation, est souvent présenté comme un modèle économique couronné de succès : avec un catalogue de 300 000 applications, qui ont fait l'objet de 10 milliards de téléchargements, l'App Store s'est imposé en 18 mois comme une référence parmi les kiosques d'applications mobiles.



Figure I.3: Smartphone iPhone 5.

## I.2.4. Windows Phone

Windows Phone est développé par la corporation Microsoft, ce qui fait qu'il ait la capacité de fonctionnement des logiciels sur « Windows ».

Compatible avec tous les logiciels de « Windows », il soutient beaucoup de types d'audio, vidéo, ainsi que la capacité de télécharger, de jouer les chansons et de regarder le TV en ligne. Grâce à ces avantages coté loisirs il amasse chaque année de plus en plus de fans.

La messagerie électronique du Windows Phone n'est pas aussi bonne que BlackBerry mais elle à la perspective du succès, Il reste tout de même moins sécurisée que les autres systèmes d'exploitation mobiles.

Coté applications tierces, il en a plus que les Symbian et BlackBerry. Il utilise la même collection de l'APIs que Windows, il est facile de développer des applications pour le développeur de Microsoft et de les exécuter.

Dans sa nouvelle version, Windows Phone a amélioré la capacité de connexion, et de compatibilité avec Office. Il a intégré plusieurs applications pour la communication comme Windows Live, hotMail, Messenger, etc.

Aujourd'hui, il existe dans le monde plusieurs appareils qui tournent sous Windows Mobile comme quelques modèles de Motorola, Samsung, T-Mobile, etc... mais Nokia qui est le leader de la téléphonie ne l'utilise pas.

En conclusion générale, Windows Phone tient son succès de la popularité de Microsoft.



**Figure I.4: Smartphone Samsung.**

## I.3. Comparaison entre les différents Systèmes d'exploitation mobiles

Ici nous proposons une comparaison entre les différents systèmes d'exploitation mobiles au niveau de leurs performances en premier lieu, puis de leur popularité et c'en examinant les part de marché de chacun.

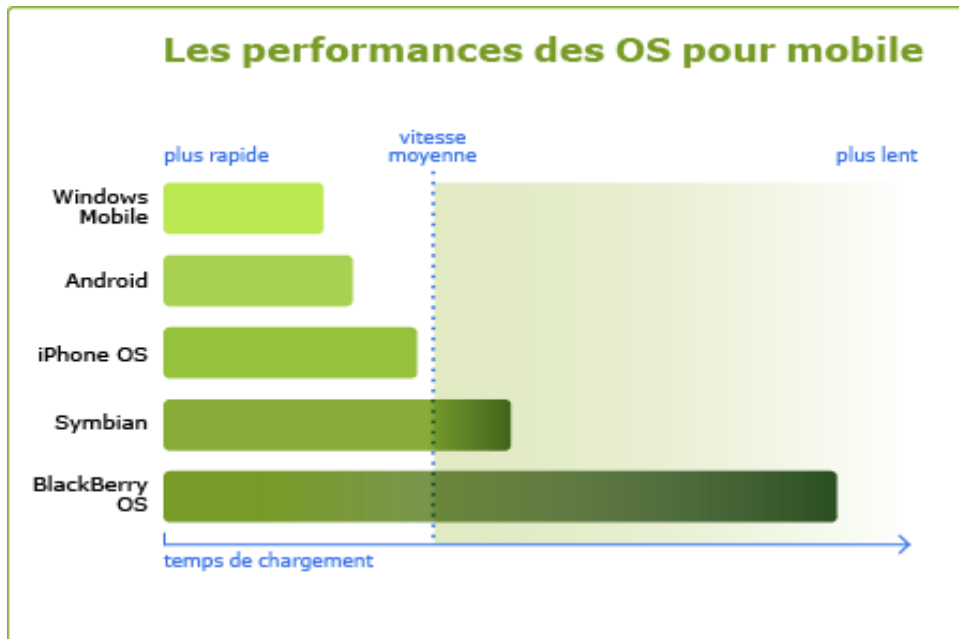


Figure I.5: Performances des Systèmes d'Exploitation mobiles [N1].

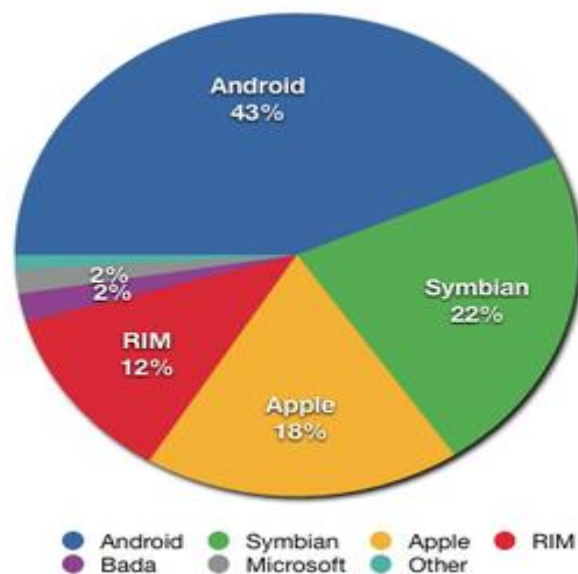


Figure I.6: Parts de marché des Systèmes d'Exploitation mobiles [N2].

## **I.4. Conclusion**

Dans ce premier chapitre nous avons introduit une vue générale sur les systèmes d'exploitation mobiles, mais vu que nous allons développer notre application sous Android, dans le prochain chapitre nous allons présenter d'une manière détaillée le système d'exploitation Android.

Chapitre II :

Présentation d'Android.

### II.1.Introduction

Android est une excellente opportunité pour appréhender le développement d'applications mobiles ambitieuses.

Rappelons les points clés d'Android en tant que plate-forme :

- ❖ elle est innovante car toutes les dernières technologies de téléphonie y sont intégrées : écran tactile, accéléromètre, GPS, appareil photo numérique etc. ;
- ❖ elle est accessible car en tant que développeur vous n'avez pas à acheter de matériel spécifique (si vous voulez aller plus loin que l'utilisation d'un émulateur, un téléphone Android pour effectuer vos tests vous sera toutefois nécessaire), ni à connaître un langage peu utilisé ou spécifique : le développement sur la plate-forme Android est en effet réalisé en langage Java, un des langages de programmation les plus répandus ;
- ❖ elle est ouverte parce que la plate-forme Android est fournie sous licence open source, permettant à tous les développeurs – et constructeurs – de consulter les sources et d'effectuer les modifications qu'ils souhaitent.

Ce chapitre est une introduction à la plate-forme, aux outils et à la configuration de notre environnement de travail.

### II.2.Description

Android est un système d'exploitation pour Smartphones et tablettes tactiles conçu par Android. D'autres types d'appareils possédant ce système d'exploitation existent, par exemple des téléviseurs, des radio-réveils ou des autoradios et même des voitures. Il a été développé par une petite startup qui fut achetée en 2007 par Google qui poursuit activement son développement avec l'Open Handset Alliance, Android est distribué sous licence open source depuis 2008. Ce système est assez nouveau auprès des programmeurs.

Selon Google qui est un majeur distributeur, Android est une plateforme puissante, moderne, sécurisée et ouverte. Basé sur le kernel Linux2.6 (noyau Linux 2.6) et utilisant la plateforme java pour ses applications. Il est entièrement gratuit et sa plateforme très flexible ce qui permet aux développeurs d'intégrer, d'agrandir et de remplacer les composants existants et d'adapter les applications aux besoins du client ou les remplacer entièrement, l'utilisateur peut donc personnaliser facilement son appareil.

De plus, il n'y a pas de distinction entre les applications natives et les applications qui sont développées par les développeurs, toutes sont disponibles sur l'Android Market(maintenant appelé Google Play Store).

En termes d'applications, Android a intégré plusieurs services de Google pour accéder rapidement aux services d'internet comme Gmail, You Tube, Google Talk, Google Calendar et Google Maps.

Android offre des avantages non négligeables, que ce soit pour les développeurs, ou pour les utilisateurs qui bénéficient grâce aux dernières versions de nombreuses fonctionnalités et d'un confort d'utilisation de qualité.

Android est : [3]

- **Open source :**  
Le contrat de licence pour Android respecte les principes de l'open source, c'est-à-dire que c'est possible de télécharger à tout moment les sources et les modifier. Android utilise des bibliothèques open source puissantes, comme par exemple SQLite pour les bases de données et OpenGL pour la gestion d'images 2D et 3D.
- **Gratuit (ou presque) :**  
Android est gratuit, autant pour les utilisateurs que pour les constructeurs. La publication d'applications sur le Google Play<sup>1</sup>, coûte la modique somme de 25\$. Ces 25\$ permettent de publier des applications à l'infini, contrairement à iPhone où il faut verser une somme de 100\$ par an.
- **Facile à développer :**  
Toutes les API mises à disposition facilitent et accélèrent grandement le travail. Ces APIs sont très complètes et très faciles d'accès. De manière un peu caricaturale, on peut dire qu'il est possible d'envoyer un SMS en seulement deux lignes de code.

---

<sup>1</sup>Google Play est un site de contenus numériques proposé par Google sur lequel vous pouvez trouver vos titres, vos livres, vos films, vos applications et vos jeux Android préférés.

## Chapitre II : Présentation d'Android

---

- **Facile à vendre :**  
Le Play Store (anciennement Android Market) est une plateforme immense et très visitée ; c'est donc une mine d'opportunités pour quiconque possède une idée originale ou utile.
- **Flexible :**  
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, différents processeurs... On trouve même des fours à micro-ondes qui fonctionnent à l'aide d'Android.  
Non seulement c'est une immense chance d'avoir autant d'opportunités, mais en plus 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 Google Play).
- **Ingénieux :**  
L'architecture d'Android est inspirée par les applications composites, et encourage par ailleurs leur développement. Ces applications se trouvent essentiellement sur internet et leur principe est la possibilité de combiner plusieurs composants totalement différents pour obtenir un résultat surpuissant. Par exemple, si on combine l'appareil photo avec le GPS, on peut poster les coordonnées GPS des photos prises.

### II.3.Open Handset Alliance

L'Open Handset Alliance (abrégé OHA) est un consortium Regroupant de grands constructeurs et développeurs de logiciels dont le but est de développer des normes ouvertes pour les appareils de téléphonie mobile.

Le consortium a été créé le 5 novembre 2007 à l'initiative de Google qui a su fédérer autour de lui 34 compagnies. On note ainsi la présence d'opérateurs comme NTT DoCoMo, Sprint Nextel, Telecom Italia ou Bouygues Telecom. Des équipementiers tels que Samsung Electronics, Motorola Mobility ou LG Electronics. Des semi-conducteurs comme Intel ou Nvidia. Et enfin des incontournables de l'Internet dont eBay. On remarquera quelques grands absents dont Nokia.

### II.4.Bugdroïde

Le personnage nommé Bugdroid est le petit robot vert utilisé par Google pour présenter Android. Ce personnage est sous licence « creativecommons by (3.0) » et peut donc être utilisé librement.

Le site Engadget annonce que Bugdroid, le logo d'Android, serait en fait un personnage d'un jeu des années 1990 sur Atari : Gauntlet: The ThirdEncounter.



**Figure II.1:** Le personnage Bugdroid.

### II.5. Historique des versions d'Android



**Figure II.2:** Evolution des versions d'Android.

Android a débuté avec la sortie de la version 1.0 « Apple pie » en septembre 2008. Android a connu plusieurs mises à jour depuis sa première version. Ces mises à jour servent généralement à corriger des bugs et à ajouter de nouvelles fonctionnalités. Dans l'ensemble, chaque version est développée sous un nom de code basé sur des desserts. Ces noms de codes suivent une logique alphabétique, en voici quelques-unes :

**Android 1.5 « Cupcake »** a apporté de nombreuses améliorations. La première grande étape fut une mise à jour du noyau vers la version 2.6.27 qui a rendu le système plus stable et corrigé le manque d'API et rendit le système plus utilisable.

**Android 1.6 « Donut », Android2.0 et 2.1 « Eclair »** ont apporté d'importantes améliorations respectivement sur les fonctionnalités et sur l'interface graphique du système.

## Chapitre II : Présentation d'Android

---

**Android 2.2 « Froyo »** a fortement mis l'accent sur la synergie avec Internet. L'envoi d'applications et de liens instantanés depuis un ordinateur est désormais possible. Aussi, Google annonce-t-elle que le navigateur chrome intégré à Android 2.2 est le navigateur mobile le plus rapide au monde grâce à l'intégration du moteur JavaScript V8.

**Android 2.3 « Ginerbread »** Dernière version dédiée uniquement aux smartphones. Cette version est parfois utilisée sur de petites tablettes.

Gingerbread était un système très réussi, qui devint à l'époque le système mobile le plus populaire.

**Android 3.0 « Honeycomb »** est spécialement étudié pour les tablettes tactiles. Les premiers modèles devraient être annoncés au CES 2011.

On y apprend quelques nouveautés comme la prise en charge de la vidéo-conférence via Gtalk, la nouvelle interface Gmail ou encore le lecteur de livre électronique Google.

La refonte graphique de l'interface utilisateur est assez réussie, plus d'informations devraient suivre dont sûrement des éclaircissements sur l'intégration ou non de l'interface de cette version d'Android sur les futurs smartphones.

**Android 4.0 « Ice Cream Sandwich »** devrait arriver très vite (mi 2011) pour rajouter encore plus de fonctionnalités aux terminaux. Pour le développement, ces nouvelles versions d'Android devraient proposer de nouveaux composants permettant de réaliser des applications avec une ergonomie plus adaptée aux tablettes tactiles.

**Android 3.0** et **Android 4.0** devraient apporter plus d'outils aux constructeurs leur permettant de proposer des tablettes tactiles, qui seront capables de rivaliser (surtout au niveau de l'ergonomie) avec Ipad.

**Android 4.1 « Jelly Bean »** Il ajoute un système de notification améliorée, la reconnaissance vocale sans connexion internet, et le « Project Butter » qui augmente la fluidité d'Android.

**Android 4.4 « KitKat »** Consommation en ressource moins élevée nécessitant moins de RAM, nouvelles icônes plus soignées, la barre du bas et celle de statut deviennent transparentes sur certains menus et changent de couleur en fonction du contenu affiché.

### II.6.Fonctionnalités d'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 :

## Chapitre II : Présentation d'Android

Version	Nom	API Level	Fonctionnalités
Android 1.0	Apple Pie (tarte aux pommes)	1	Fonctionnalités de base (téléphonie, SMS, réseaux de données, accès au répertoire, ...etc)
Android 1.1	Banana Split	2	<ul style="list-style-type: none"> <li>• corrections de bogues (alarme, Gmail...)</li> <li>• Amélioration des fonctionnalités MMS</li> <li>• Introduction des applications payantes sur l'Android Market</li> </ul>
Android 1.5	Cupcake (petit gâteau)	3	<ul style="list-style-type: none"> <li>• Nouveau clavier avec auto complétion</li> <li>• Support Bluetooth</li> <li>• Enregistrement de vidéos</li> <li>• Widgets</li> <li>• Animations améliorées</li> </ul>
Android 1.6	Donut(Beignet)	4	<ul style="list-style-type: none"> <li>• Recherche vocale améliorée</li> <li>• Indicateur d'utilisation de la batterie</li> <li>• Apparition de fonctionnalités pour les réseaux privés virtuels (VPN)</li> </ul>
Android 2.0 ; 2.0.1 ; 2.1	Eclair (sorte de gâteau à la crème)	5 ; 6 ; 7	<ul style="list-style-type: none"> <li>• Interface graphique améliorée</li> <li>• Gestion des contacts changée</li> <li>• Support d'HTML 5</li> <li>• Support Bluetooth 2.1</li> <li>• Clavier virtuel amélioré</li> <li>• Refonte de l'installation et des mises à jour du kit de développement</li> <li>• Réduction de la consommation de la batterie</li> <li>• Quelques améliorations graphiques comme l'écran de verrouillage</li> <li>• Fonds d'écran animés</li> <li>• Nouveaux effets 3D (notamment sur la gestion des photos)</li> </ul>
Android 2.2	Froyo(yaourt glacé)	8	<ul style="list-style-type: none"> <li>• Nouveaux Widgets</li> <li>• Gestion des points d'accès au réseau</li> <li>• Support multilingue</li> </ul>
Android 2.3	Gingerbread(pain d'épice)	9	<ul style="list-style-type: none"> <li>• Amélioration d'interface graphique</li> <li>• Téléphonie IP</li> <li>• Nouveau clavier virtuel</li> <li>• Support des grands écrans à résolutions extra-larges</li> <li>• Amélioration de la gestion de</li> </ul>

## Chapitre II : Présentation d'Android

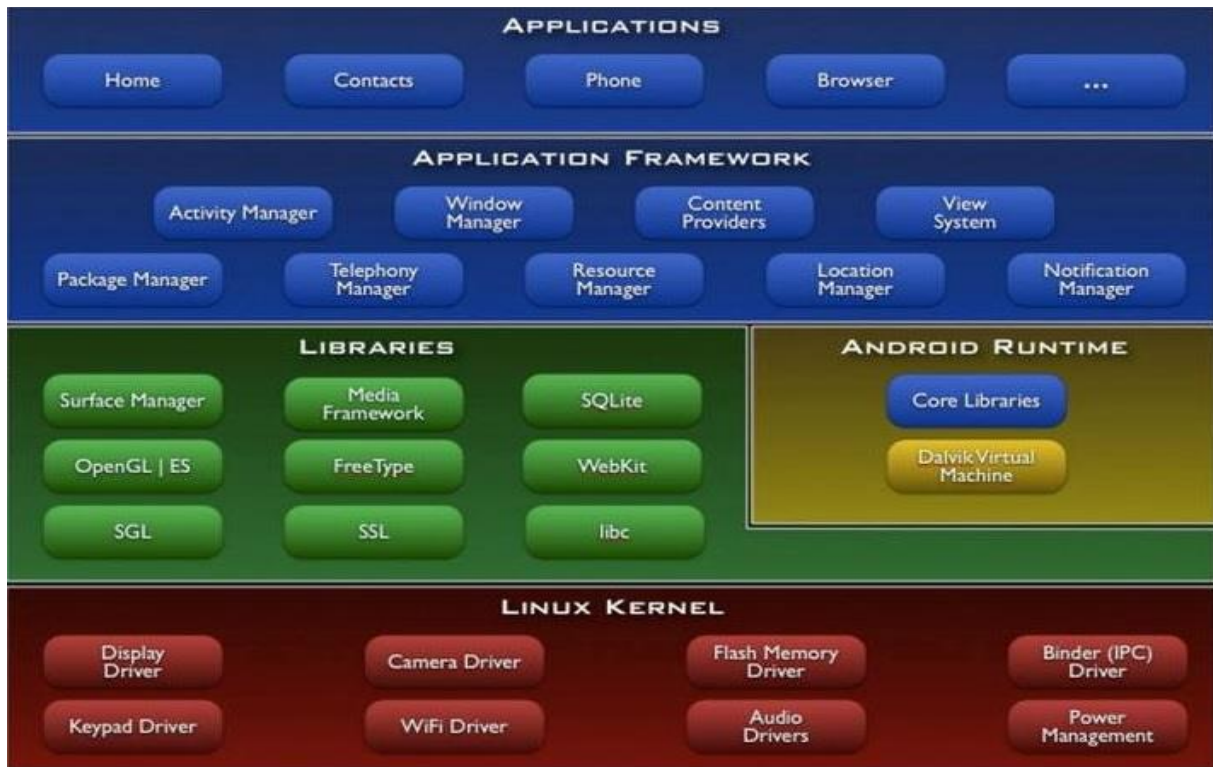
			<p>l'alimentation et du contrôle des applications</p> <ul style="list-style-type: none"><li>• Améliorations audio, graphiques et des commandes pour le développement des jeux</li></ul>
Android 3.0	Honeycomb(rayon de miel)	11	<ul style="list-style-type: none"><li>• Optimisé pour les tablettes et équipement à écran large</li><li>• Multitâche et système de notification amélioré</li></ul>
Android 4.0	Icecream sandwich (Sandwich à la crème glacée)	12	<ul style="list-style-type: none"><li>• Nouvelle interface graphique</li><li>• Amélioration de la sécurité</li><li>• Beaucoup de raccourcis (Appareil photo, accès sdcard, ...etc)</li></ul>
Android 4.1	Jelly Bean (dragée)	16	<ul style="list-style-type: none"><li>• Interface utilisateur plus fluide</li><li>• Amélioration de l'accessibilité</li><li>• Notifications extensibles</li><li>• Recherche vocale améliorée</li><li>• Application appareil photo améliorée</li></ul>

**Tableau II.1:** Fonctionnalités d'Android.

### II.7.Architecture Android

Le diagramme suivant illustre les composants principaux du système d'exploitation Android. Chaque section sera décrite dans ce qui suit :

## Chapitre II : Présentation d'Android



**Figure II.3:** Architecture d'Android [N3].

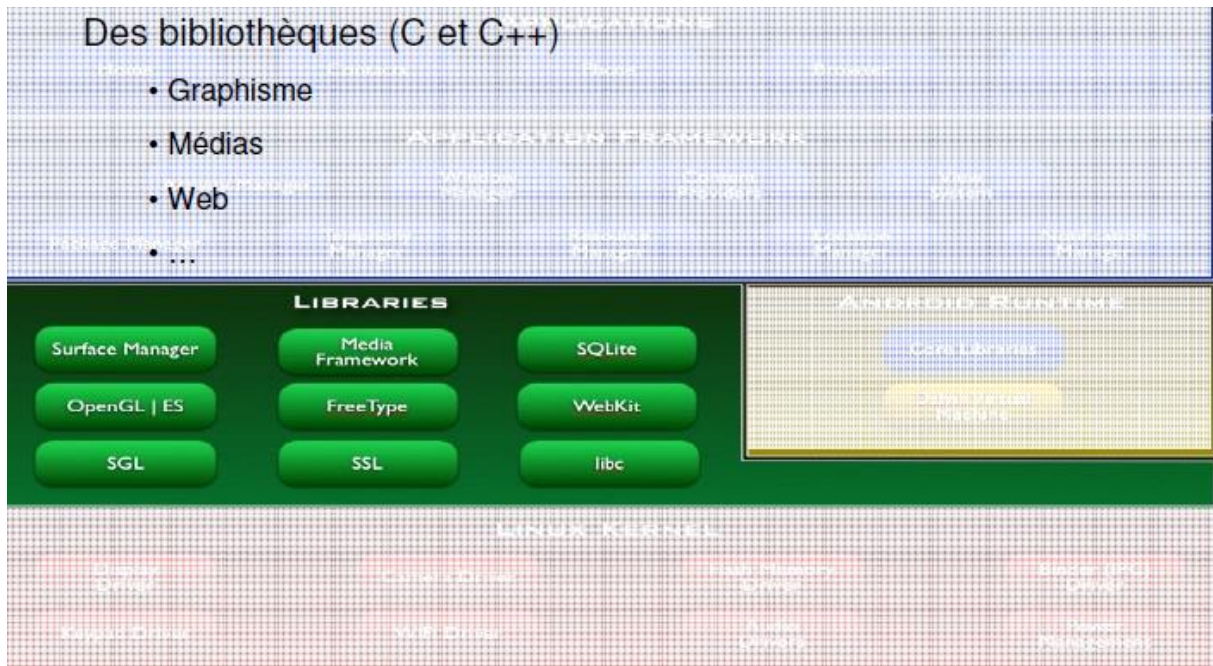
Android est basé sur un kernel linux 2.6.xx.



**Figure II.4:** Architecture d'Android « Kernel linux » [4].

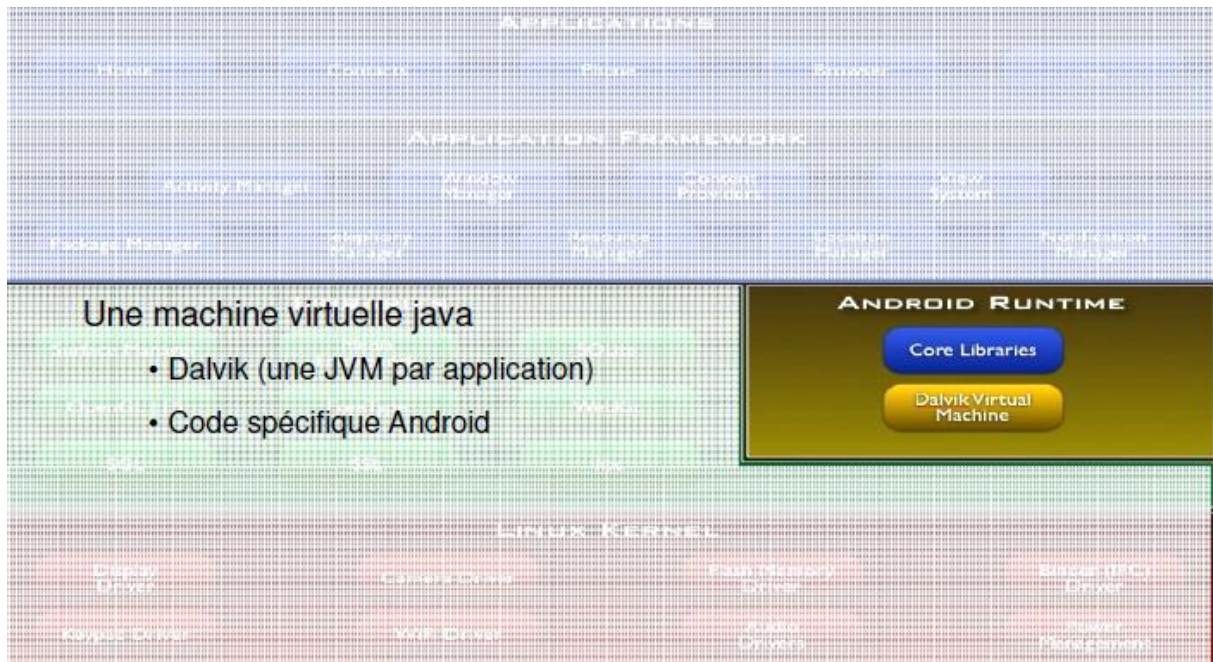
Au-dessus de cette couche, on retrouve les librairies C/C++ utilisées par un certain nombre de composants du système Android.

## Chapitre II : Présentation d'Android



**Figure II.5:** Architecture d'Android « Bibliothèques C/C++ » [4].

Au-dessus des bibliothèques, on retrouve l'Android Runtime. Cette couche contient les bibliothèques cœurs du Framework ainsi que la machine virtuelle exécutant les applications.



**Figure II.6:** Architecture d'Android « Android runtime » [4].

Au-dessus de la couche "Android Runtime" et des bibliothèques cœurs, on retrouve le Framework permettant au développeur de créer des applications.

## Chapitre II : Présentation d'Android



**Figure II.7:** Architecture d'Android « Application framework » [4].

Enfin au-dessus du Framework, il y a les applications.



**Figure II.8:** Architecture d'Android « Application » [4].

### II.7.1.Applications

Android est fourni avec un ensemble d'applications dont un client email, une application SMS, un calendrier, un service de cartographie, un navigateur... toutes écrites en JAVA.

### II.7.2.Framework de développement

En fournissant une plateforme de développement ouverte, le système Android offre aux développeurs la possibilité de créer des applications extrêmement riches et innovantes. Les développeurs sont libres de profiter du matériel périphérique et informations sur la localisation d'accès, exécuter des services d'arrière-plan, définir des alarmes, ajouter des notifications à la barre d'état, etc.

Les développeurs ont un accès complet au même framework API utilisé par les applications de base. L'architecture d'application est conçue pour simplifier la réutilisation des composants; n'importe quelle application peut publier ses capacités et n'importe quelle autre application peut alors faire usage de ces capacités (soumis à des contraintes de sécurité appliquées par le framework). Ce même mécanisme permet aux composants d'être remplacés par l'utilisateur.

Toutes les applications sous-jacentes forment un ensemble de services et de systèmes, y compris:

- Un jeu extensible de vues qui peuvent être utilisées pour construire une application.
- Des fournisseurs de contenu qui permettent aux applications d'accéder aux données des autres applications (telles que les Contacts), ou de partager leurs propres données.
- Un gestionnaire de ressources.
- Un gestionnaire de notification qui permet à toutes les demandes d'afficher des alertes personnalisées dans la barre d'état.
- Un gestionnaire d'activité qui gère le cycle de vie des applications et propose une navigation commune.

### II.7.3.Bibliothèques

Android dispose d'un ensemble de bibliothèques C / C++ utilisées par les différents composants du système Android. Elles sont offertes aux développeurs à travers le framework Android. En voici quelques-unes:

Système de bibliothèque C : une mise en œuvre dérivée de BSD de la bibliothèque C standard du système (libc), destinés aux systèmes embarqués basés sur Linux.

Comme cela a été dit précédemment, Android ne supporte pas la glibc, donc les ingénieurs d'Android ont développé une bibliothèque C (libc) nommé Bionlibc . Elle est optimisée pour les appareils mobiles et a été développée spécialement pour Android.

Les ingénieurs d'Android ont décidé de développer une libc propre à la plateforme Android car ils avaient besoin d'une libc légère (la libc sera chargée dans chaque processus) et rapide (les appareils mobiles ne disposent pas de CPU puissant).

## Chapitre II : Présentation d'Android

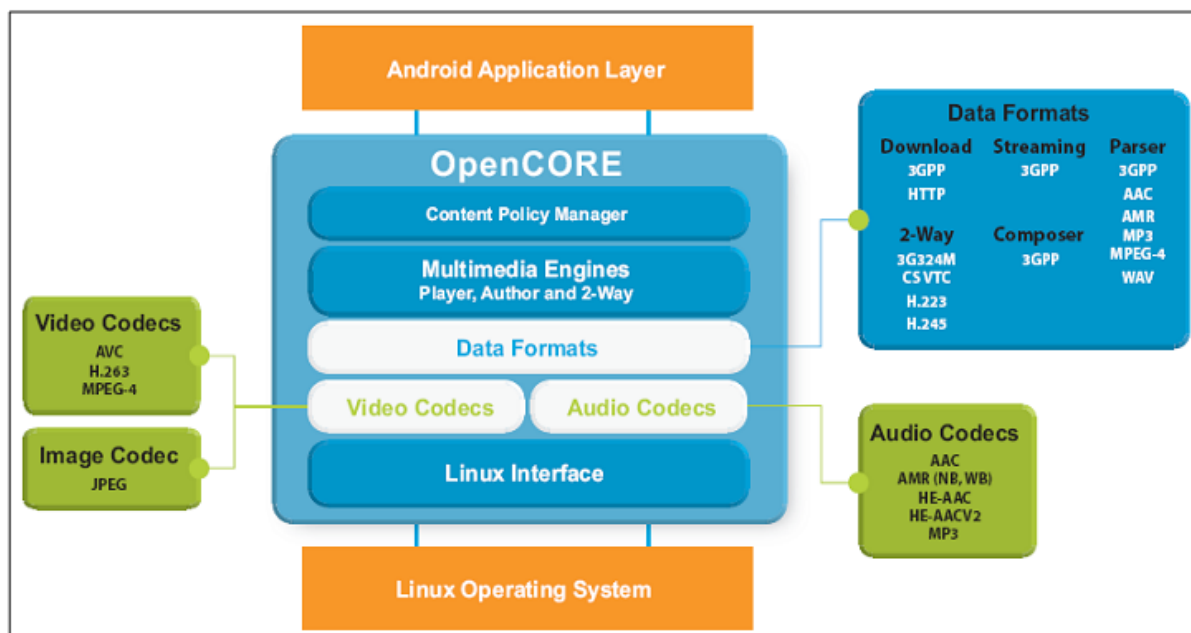
La Bionilibc a été écrit pour supporter les CPU ARM, bien que le support x86 soit présent. Il n'y pas de support pour les autres architectures CPU telles que PowerPC ou MIPS. Néanmoins, pour le marché des appareils mobiles, seulement l'architecture ARM est importante.

Cette libc est sous licence BSD. Elle reprend une grande partie du code des glibc issue d'OpenBSD, FreeBSD et NetBSD.

Ces caractéristiques importantes :

- Elle pèse environ 200Ko, soit la moitié de la glibc L'implémentation des pthreads (POSIX thread) a été complètement réécrite pour supporter les threads de la machine virtuelle Dalvik. De ce fait, la Bionilibc ne supporte pas les threads POSIX .
- Les exceptions C++ et les "wide char" ne sont pas supportés.
- Médiathèques : basée sur PacketVideo de OpenCore; les librairies permettant la lecture et l'enregistrement audio et vidéo, ainsi que la gestion des fichiers image, y compris MPEG4, H.264, MP3, AAC, AMR, JPG et PNG.

Le schéma ci-dessous décrit tous les éléments de l'architecture de ces médiathèques:



**Figure II.9:** Architecture de ces médiathèques.

- Surface Manager : gère l'accès au sous-système d'affichage et de façon transparente.
- LibWebCore : le navigateur web présent dans Android est basé sur le moteur de rendu sous licence BSD WebKit.
- WebKit : est moteur de rendu, qui fournit une "fondation" sur laquelle on peut développer un navigateur web. Il a été originellement dérivé par Apple du moteur de rendu KHTML pour être utilisé par le navigateur web Safari et maintenant il est

## Chapitre II : Présentation d'Android

développé par KDE project, Apple, Nokia, Google et d'autres. WebKit est composé de deux bibliothèques : WebCore et JavaScriptCore qui sont disponibles sous licence GPL.

WebKit supporte le CSS, Javascript, DOM, AJAX. La dernière version a obtenu 100% au test Acid 3. La version de WebKit présent dans Android a été légèrement modifiée pour s'adapter aux appareils mobiles. Ainsi, le moteur de rendu basé sur WebKit présent dans Android supporte l'affichage sur une colonne.

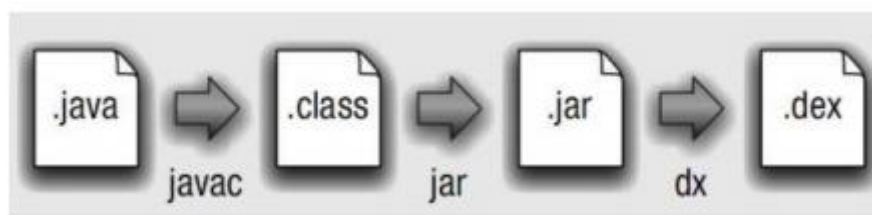
- SGL : le moteur graphique 2D.
- Bibliothèques 3D : une implémentation basée sur OpenGL ES 1.0 API; les bibliothèques utilisent l'accélération 3D matérielle (si disponible).
- FreeType : bitmap et vectoriel de rendu de police.
- SQLite : un moteur de base de données relationnelles puissant et léger, disponible pour toutes les applications.

### II.7.4.Android Runtime

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 sa propre instance de la machine virtuelle Dalvik. Dalvik a été écrit pour que le dispositif puisse faire tourner plusieurs machines virtuelles de manière efficace. La machine virtuelle Dalvik 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.



**Figure II.10:** Transformation d'un fichier .java à un fichier .dex.

### II.7.5.Linux Kernel

Android est basé sur un kernel linux 2.6 mais ce n'est pas linux. Il ne possède pas de système de fenêtrage natif (X window system). La glibc n'étant pas supportée, Android utilise une libc customisée appelée Bioniclibc.

## Chapitre II : Présentation d'Android

---

Enfin, Android utilise un kernel avec différents patches pour la gestion de l'alimentation, le partage mémoire, etc. permettant une meilleure gestion de ces caractéristiques pour les appareils mobiles.

Android n'est pas linux mais il est basé sur un kernel linux. Pourquoi sur un kernel linux ?

Le kernel linux a un système de gestion mémoire et de processus reconnu pour sa stabilité et ses performances.

Le model de sécurité utilisé par linux, basé sur un système de permission, est connu pour être robuste et performant. Il n'a pas changé depuis les années 70.

- Le kernel 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 kernel linux est entièrement open source et il y a une communauté de développeurs qui l'améliorent et rajoutent des drivers.

C'est pour les points cités ci-dessus que l'équipe en charge du noyau a décidé d'utiliser un kernel linux.

### II.8. Les composants principaux d'une application

Les applications Android sont composées de 4 types de composants :

#### II.8.1. Activités

Une Activité représente un écran de l'application. Une application peut avoir une ou plusieurs activités (par exemple pour une application de messagerie on pourrait avoir une Activité pour la liste des contacts et une autre pour l'éditeur de texte). Chaque Activité est implémentée sous la forme d'une classe qui hérite de la classe Activity.

#### II.8.2. Services

Les services n'ont pas d'interface graphique et tournent en tâche de fond. Il est possible de s'inscrire à un service et de communiquer avec celui-ci en utilisant l'API Android.

#### II.8.3. Broadcast receivers

Il se contente d'écouter et de réagir aux annonces broadcast (par exemple changement de fuseau horaire, appel entrant...).

#### II.8.4. Content providers

Il permet de partager une partie des données d'une application avec d'autres applications.



## Chapitre II : Présentation d'Android

---

### **Paused (en pause) :**

L'activité est partiellement visible à l'écran.

C'est le cas lors de la réception d'un SMS et qu'une fenêtre semi-transparente se pose devant l'activité pour afficher le contenu du message et permettre d'y répondre par exemple.

Ce n'est pas sur cette activité qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application sus-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser de l'application qui l'obstrue, puis il pourra à nouveau interagir avec.

### **Stopped (Arrêtée) :**

L'activité n'est pas visible à l'écran.

Si une activité est complètement masquée par une autre activité, il est arrêté. Il conserve tous les états membres et de l'information, cependant il n'est plus visible pour l'utilisateur que sa fenêtre est cachée et il sera souvent tués par le système lorsque la mémoire est nécessaire ailleurs.

### **Dead (Mort) :**

Cette activité a terminé ou il n'a jamais été démarré.

Si une activité est en pause ou arrêtée, le système peut chuter l'activité de la mémoire, soit par lui demandant de se terminer, ou tout simplement tuer le processus. Quand il est affiché de nouveau à l'utilisateur, il doit être redémarré et restauré à son état antérieur.

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

## **II.10.Développement d'une application Android**

Voici quelques étapes principales dans le processus de développement d'une application sur Android :

- Faire la conception de la base de données.
- Créer des classes pour représenter les données physiques (couche Mapping) et pour définir des actions comme : supprimer, ajouter, modifier des données.
- Dessiner des interfaces en les fichiers XML ou en codage :
  - ✓ Les vues (View) : Text, Edit, List, Image, Web, Map, etc.
  - ✓ Les arrangements (layout) : Frame, Linear, Relative, Table, Absolute.
- Choisir des arrangements (layout) : Les layouts sont les ressources qui indiquent les interfaces des activités. On utilise les fichiers XML pour exprimer les interfaces. Mais il existe d'autre technique pour dessiner l'interface. Dans cette technique, on programme directement les composants graphiques en utilisant le codage.

## Chapitre II : Présentation d'Android

- Organiser des ressources: les constantes globales (string.xml), les icônes, les images, etc.
- Créer et mettre à jour le fichier de configuration : AndroidManifest.xml. AndroidManifest.xml (configuration de l'application) est utilisé pour stocker les dispositions (settings) globales comme les permissions de l'application, les activités, les filtres de l'intention.



**Figure II.12:** Une application sur Android.

- Créer des activités (Créer les classes pour exécuter les fonctions avec la base de données (ajouter, supprimer, modifier, mettre à jour, etc.) :
  - ✓ Chaque activité peut correspondre avec un écran ou une fonction de cette application.
  - ✓ Il existe quelques activités qui s'occupent des méthodes pour communiquer avec la base de données (Couche Mapping).
  - ✓ Les Intents sont utilisées pour orienter des activités (CALL, ACTION\_MAIN, ACTION\_VIEW, etc).

### II.11. Différents équipements utilisant Android

#### ❖ Les Smartphones :

Le premier mobile commercialisé sous Android est le HTC Dream/G1 produit par la firme Taïwanaise HTC, lancé aux États-Unis le 22 octobre 2008.

En France, le premier mobile sous Android fut commercialisé par Orange avec le HTC Dream depuis le 12 mars 2009. Le HTC Magic est arrivé début mai 2009 chez SFR, suivi par le Samsung Galaxy chez Bouygues. Le HTC Hero, bénéficie de Sense, un habillage particulier de l'interface. Son successeur : le HTC Legend est commercialisé

## Chapitre II : Présentation d'Android

---

sur le marché européen en avril 2010 avec la version 2.1 d'Android et, comme le Hero, il bénéficie de Sense.

En novembre 2009, Motorola a lancé aux États-Unis le Droid, le premier téléphone muni de la version 2.0 d'Android. L'appareil qui a trouvé 250 000 acheteurs une semaine après son lancement fut lancé au Canada début février 2010.

Le 5 janvier 2010 Google annonce le Nexus One, téléphone conçu par la firme de Mountain View et sous-traité par HTC. Doté de caractéristiques alors assez impressionnantes (écran AMOLED de 3,7 pouces, processeur de 1 GHz, 512 Mo de mémoire RAM et Android 2.1), il devait également avoir l'avantage de recevoir directement ses mises à jour de Google. Il fut en effet le premier Smartphone à bénéficier d'Android 2.2 Froyo en juin 2010. Cependant ses ventes n'ont pas été exceptionnelles.

En décembre 2010, Samsung fabrique le Nexus S pensé par Google sous Android 2.3, et le mois de sa sortie celui-ci est envoyé dans l'espace à plus de 32 000 mètres d'altitude.

Le 18 octobre 2011, Samsung et Google dévoilent le Samsung Galaxy Nexus, premier Smartphone sous Android 4 "Ice Cream Sandwich". Celui-ci intègre le déverrouillage par reconnaissance faciale, l'utilisation de boutons virtuels et un système de reconnaissance vocale avancé.

## Chapitre II : Présentation d'Android

---



**Figure II.13: Samsung Galaxy S3.**



**Figure II.14: Sony Xperia.**



**Figure II.15: HTC Legend.**



**Figure II.16: Galaxy Nexus.**

## Chapitre II : Présentation d'Android

---

### ❖ **Les Tblettes :**

En septembre 2010, Samsung présente à l'IFA de Berlin le Samsung Galaxy Tab, tournant sous Android 2.2 (FroYo). Archos avec sa génération 7 de tablettes internet introduit Android. Dans la même lignée, les tablettes Archos de la génération intègrent Android 2.2 (FroYo).

Motorola a présenté au CES 2011 la Xoom, première tablette bénéficiant de la nouvelle version du système mobile de Google, Honeycomb (Android 3.0). Elle y a reçu le prix de la meilleure innovation. S'ensuivent de nombreuses tablettes sous Android Honeycomb proposées par un très large choix de constructeurs, dont Samsung, Acer, ASUS, Toshiba et Sony.

Android a été porté sur d'autres appareils comme la HP TouchPad, le portage a été réalisé début 2012 avec la version 9 de CyanogenMod, basé sur Ice Cream Sandwich, et fin 2011 avec la version 7 de CyanogenMod basé sur Gingerbread.

### ❖ **Les Télévisions :**

Le 5 avril 2010, la première télévision sous Android est dévoilée. Celle-ci est développée par l'entreprise suédoise People of Lava et se nomme Scandinavia. Elle possède les applications Facebook, YouTube, Google Maps et Twitter, possède un navigateur Web ainsi qu'un client de messagerie électronique.

### ❖ **Audio radios :**

La société française Parrot SA a dévoilé au CES 2011 le premier autoradio tournant sous Android : la Parrot Asteroid. Cet autoradio offre notamment un adaptateur GPS, des ports USB et une connectivité Bluetooth pour contrôler la musique de son téléphone mobile.

### ❖ **Netbook :**

La version 4.1 (Jelly Bean) devrait pouvoir être intégré aux miniportables, comme Chrome OS avec une possibilité de dual-boot entre ce dernier et Android.

### ❖ **Console de jeux vidéo :**

DEA commercialisé depuis septembre 2012, sous le nom de MyPlay, une console de jeux vidéo portable sous Android. Archos par contre commercialise une console nommée GamePad.

## II.12. Les outils que fournit le SDK

Afin de faciliter le processus de réalisation d'applications pour les développeurs, le SDK Android fournit des outils pour les aider, les voici :

## Chapitre II : Présentation d'Android

### Emulateur :

Emulator en anglais, il permet de lancer sur la machine du développeur un terminal virtuel représentant à l'écran un téléphone embarquant Android. C'est bien évidemment un outil indispensable pour le développement mobile.

A chaque version d'Android est associée une version de l'émulateur, permettant au développeur de voir exactement à quoi ressemblera son application sur un matériel réel sans utiliser un matériel physique ce qui est très pratique quand on ne dispose pas d'un Smartphone ou d'une tablette sur place. Il est entièrement intégré sous Eclipse et contient plusieurs images en fonction des différentes versions d'Android.

Rappelons cependant que l'émulateur ne propose pas toutes les fonctionnalités d'un vrai téléphone. Il ne permet par exemple pas d'émuler la gestion du Bluetooth.



Figure II.17: Interface du simulateur Android.

### L'ADB (Android Debug Bridg) :

ADB est un outil disponible en ligne de commande polyvalent. Il permet de communiquer avec une instance de l'émulateur directement ou avec un téléphone physique à l'aide d'un câble USB standard.

## Chapitre II : Présentation d'Android

---

Il donne la possibilité d'installer ou de désinstaller des applications, d'envoyer ou de récupérer des fichiers et de se connecter en ligne de commande sur l'appareil. Il est aussi utilisé par le SDK pour publier et déboguer des applications.

### **II.13.Conclusion**

Dans ce chapitre, nous avons fait une étude détaillée sur l'Android tout en présentant un bref historique, les fonctionnalités que nous pouvons trouver sur ce système d'exploitation et l'architecture d'Android, à savoir les principaux composants du système.

Chapitre III :

Analyse et conception.

### **III.1.Introduction**

La conception de toute solution informatique est d'une grande importance et elle doit être traitée avec rigueur et précision, car elle constitue la base du système à développer. Avant de s'engager dans la conception, il est impératif de passer par la phase d'analyse qui permet d'identifier les différents acteurs qui interagissent avec le système ainsi que leurs besoins. Puis on passe à la conception qui, en s'appuyant sur les résultats de la phase d'analyse, donnera la description détaillée du système cible et des objectifs à atteindre.

Pour ce faire, ma démarche va s'appuyer sur le langage UML, qui permet la conception et développement de l'application.

### III.2. Les Jeux de réflexion

Le cerveau est comme un muscle, pour le maintenir régulièrement en forme il faut l'exercer et le solliciter. Pour cette raison on a réalisé une application facilement téléchargeable destinée aux fans de jeux de réflexion ayant un smartphone. Parmi tous les jeux de réflexion qui existent, on a choisi le célèbre jeu Sudoku.

Les jeux de réflexion se distinguent par l'utilisation de mécanismes abstraits, où le hasard a une part nulle (ou limitée). Un jeu de réflexion peut être :

- un casse-tête ou jeu solitaire où il faut résoudre une énigme, séparer des objets, reconstituer un objet, résoudre des équations...
- un jeu de société : combinatoire abstrait, jeu de lettres...
- un jeu de simulation ou un jeu de guerre, qui s'appuie plus ou moins fidèlement sur un thème particulier (simulation de combat militaire, conquête d'un territoire, développement d'une civilisation...).
- un jeu vidéo de réflexion qui peut être de type solitaire (résolution de problème), guerrier, ...

Dans notre cas on s'intéresse aux jeux solitaires « Casse-tête ».

Un casse-tête est un jeu qui se joue seul. Il consiste par exemple, à partir d'une situation initiale donnée ou aléatoire à aboutir à une situation particulière en suivant un certain nombre de règles.

Il existe beaucoup de jeu casse-tête, citons quelques-uns :

- ✓ Rubik's cube
- ✓ Sudoku
- ✓ Casse-têtes à Bardes
- ✓ Les Dominos
- ✓ Casse-tête de déplacements
- ✓ Puzzle multi-pyramidal
- ✓ Casse-tête numériques et logiques
- ✓ Dédales et labyrinthes

#### ❖ **Sudoku**

Sudoku (abréviation d'une phrase japonaise signifiant "les nombres doivent être seuls") est un jeu introduit dans le monde occidental, il y a une trentaine d'année, et qui a connu récemment un grand succès. Une recherche sur le mot « Sudoku » dans un moteur de recherche bien connu retourne des liens vers plus de 61500000 pages web.

A la fin des années 1990, un néo-zélandais remarque le jeu, développe un programme pour produire automatiquement des problèmes (grilles) et les propose à la presse britannique, friande de mots croisés et autres jeux de réflexion. C'est à partir de cette introduction que

## Chapitre III : Analyse et conception

l'engouement que nous connaissons actuellement est né. Cet engouement est tel que des jeux de Sudoku sont présents dans les quotidiens, sur les chaînes de télévision, que des concours nationaux et internationaux sont organisés. Un grand nombre d'implémentations du jeu sont présentes sur média informatisé, que ce soit sous forme de programme indépendant (sur ordinateurs, consoles de jeux et même téléphones portables) ou présenté sur le web.

### ❖ Principe

Notre application offre aux utilisateurs un jeu logique-basé de puzzle qui est le célèbre jeu « Sudoku », c'est l'objectif d'une grille (9 × 9) avec les chiffres (1 à 9) pour se compléter mutuellement de sorte que chaque chiffre dans chaque colonne, chaque ligne et dans chaque bloc (3 × 3) se produit qu'une seule fois. Le point de départ est une grille dans laquelle plusieurs numéros sont déjà donnés.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Figure III.1: Exemple de grille de départ du jeu sudoku.

### III.2.1.L'intelligence implémentée dans les algorithmes

Sudoku est conçu pour être résolu par des joueurs humains avec un crayon et du papier, mais avec des algorithmes bien définis, il est possible de résoudre le temps dans presque réel par ordinateur ou même un smartphone.

Voici quelque algorithme de résolution de sudoku:

#### ❖ Backtracking (Le retour sur trace) :

Algorithmes de traçage sont adaptés pour résoudre le Sudoku qui parcourt toutes les solutions possibles pour le sudoku donné. Si les solutions sont assignées ne conduit pas à des solutions de Sudoku, l'algorithme jeter les solutions et restauration aux solutions originales et jugé à nouveau et où le nom de retours en arrière.

### ❖ **Exact cover (Couverture exacte) :**

Sudoku peut être décrite comme un exemple de la couverture exacte problème. Cela permet à la fois d'une élégante description du problème et une solution efficace en utilisant un algorithme de retour en arrière. Tandis que la couverture exacte ne garantit pas efficaces temps de solution pour les grands réseaux, les implémentations de Sudoku en utilisant des algorithmes de couverture exacte, comme Danse Liens, résoudre généralement 9x9 grilles de Sudoku avec un temps de calcul minimale de l'ordre de quelques secondes.

### ❖ **Brute force algorithm (Brute force algorithm) :**

Certains amateurs ont développé des programmes informatiques qui permettent de résoudre les puzzles de Sudoku en utilisant un algorithme de force brute. En utilisant ce dernier peut être une méthode pratique pour résoudre des énigmes si le code est bien conçu.

Les avantages de cette méthode sont les suivantes:

- ✓ une solution est garantie (tant que le casse-tête est valide)
- ✓ de temps à résoudre est la plupart du temps rien à voir avec le degré de difficulté

L'inconvénient de cette méthode est qu'elle peut être relativement lente par rapport aux méthodes de résolution d'ordinateur modélisées après les méthodes déductives.

### ❖ **Stochastiques méthodes de recherche / d'optimisation :**

Sudoku peut être résolu en utilisant stochastiques (aléatoires)-recherche en fonction des méthodes. Un exemple de cela est:

- ✓ l'attribution aléatoire des nombres aux cellules vides dans la grille
- ✓ calculer le nombre d'erreurs
- ✓ "Mélange" ces chiffres insérés dans le réseau jusqu'à ce que le nombre d'erreurs soient réduit à zéro

Algorithmes d'optimisation stochastique fondée sont connus pour être très rapide, mais ils ne sont peut-être pas aussi rapidement que certaines techniques basés sur la logique. Contrairement à ce dernier toutefois, des algorithmes d'optimisation ne nécessitent pas nécessairement des problèmes d'être logique-résoluble, en leur donnant la possibilité de résoudre un large éventail de cas problème.

## III.3.Présentation de l'UML

UML (Unified Modelling Language) est un langage de modélisation objet qui décrit la Mise en œuvre de la notation développée en réponse à l'appel lancé par omg (Object management groupe) dans le but de définir une notation standard par modélisation des Applications construites à l'aide des objets. [5]

UML possède 9 diagrammes de modélisation, répartis sur trois axes du niveau conceptuel :

- ✓ **Fonctionnel**
- ✓ **Structural**
- ✓ **Temporel**

### III.3.1. Modélisation avec l'UML

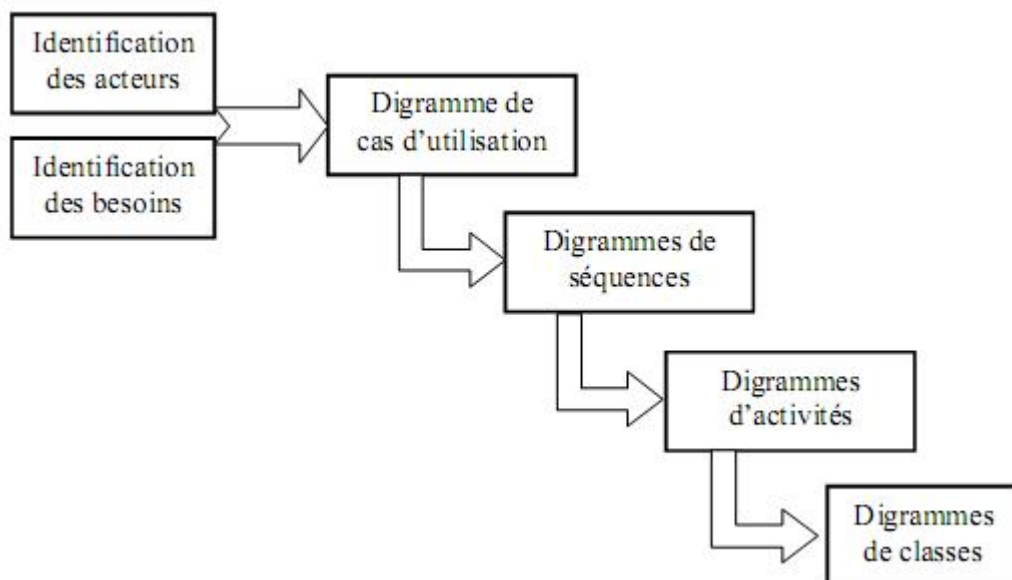
UML permet de représenter des modèles, mais il ne définit pas de processus d'élaboration de modèles. Les auteurs d'UML conseillent tout de même une démarche pour favoriser la réussite d'un projet, cette démarche doit être : [6]

- ✓ **Une démarche itératif et incrémentale** : Pour comprendre et représenter un système complexe, pour analyser par étapes, pour favoriser le prototypage et pour réduire et maîtriser l'inconnu.
- ✓ **Une démarche guidée par les besoins des utilisateurs** : Tout est basé sur le besoin des utilisateurs du système, le but du développement lui-même est de répondre à leur besoin. Chaque étape sera affinée et validée en fonction des besoins des utilisateurs.

**Une démarche centrée sur l'architecture logicielle** : c'est la clé de voute de succès d'un développement, les choix stratégiques définiront la qualité du logiciel.

### III.3.2. La démarche de modélisation avec l'UML

La démarche de modélisation choisie pour concevoir notre application peut être représentée graphiquement comme suite :



**Figure III.2** : La démarche adoptée pour la modélisation.

## III.4. Analyse

### III.4.1. Identification des besoins

La liste des fonctionnalités présentes sur l'application est la suivante :

- ✓ Commencer un nouveau jeu (nouvelle grille).
- ✓ Compléter une grille donnée par le programme.
- ✓ Valider une grille qu'il aura complétée.
- ✓ Continuer une partie déjà commencé (grille non complète).
- ✓ Choix de la difficulté du jeu.
- ✓ Consulter la description du jeu (comment jouer).
- ✓ Quitter l'application.

### III.4.2. Identification des acteurs

**Définition :** Un acteur représente un ensemble de rôles joués par des entités externes (utilisateur humain, dispositif matériel ou autre système) qui interagissent directement avec le système étudié. [7]

Dans le cas de notre application, nous avons un acteur qui interagit avec le système qui est :

- **User :** Cet acteur doit lancer l'application, choisir une activité à faire dans le menu principal, commencer un nouveau jeu ou continuer.

### III.4.3. Spécification des tâches

**Définition :** Une tâche est l'ensemble des différentes fonctions qui veut être accédés pas un acteur bien spécifique.

L'acteur définis précédemment effectue un certain nombre de tâches, ces tâches sont résumées dans le tableau ci-dessous :

Acteur	Tâches
Utilisateur (User)	T1 : Lancer l'application. T2 : Accéder à un nouveau jeu. T3 : Choix de la difficulté du jeu. T4 : Compléter une grille donnée par le programme. T5 : Valider une grille complétée. T6 : Accéder à un jeu non terminé. T7 : Accéder à la description du jeu. T8 : Quitter l'application.

**Tableau III.1:** Spécification des tâches.

### III.4.4. Spécification des scénarios

**Définition :** C'est une instance (occurrence) de cas d'utilisation, chaque fois qu'une instance d'un acteur déclenche un cas d'utilisation, un scénario est créé. Ce scénario suivra un chemin particulier dans le cas d'utilisation. [7]

**Note :**

Durant notre étude les scénarios seront symboliser par **si** (**i** : représente le numéro scénario). Le tableau ci-dessous récapitule les différents scénarios créés par les cas d'utilisation cités.

Acteurs	Tâches	Scénarios
Utilisateur (user)	T1 : Lancer l'application.	S1 : Cliquer sur l'icône de l'application.
	T2 : Accéder à un nouveau jeu.	S2 : Cliquer sur le bouton « New Game » dans l'interface d'accueil.
	T3 : Choix de la difficulté du jeu.	S3 : Sélectionner la difficulté du jeu parmi les choix
	T4 : Compléter une grille donnée par le programme.	S4 : Remplir les cases vide de la grille avec les nombre.
	T5 : Valider une grille.	S5 : Si la grille est complète alors elle valide.
	T6 : Accéder à un jeu non terminé.	S6 : Cliquer sur le bouton « continue » dans le menu principal.
	T7 : Accéder à la description du jeu.	S7 : Cliquer sur le bouton « about » dans l'interface d'accueil.
	T8 : Quitter l'application.	S8 : Cliquer sur le bouton « exit » dans l'interface d'accueil.

**Tableau III.2:** Spécification des scénarios.

### III.4.5. Les cas d'utilisation

**Définition :** Un cas d'utilisation (**user case**) représente un ensemble de séquence d'actions qui sont réalisées par le système et qui produisent un résultat observable intéressant pour un acteur particulier. Un cas d'utilisation modélise un service rendu par le système. [7]

#### III.4.5.1. Description des cas d'utilisation

Nous procéderons aux descriptions des cas d'utilisation dans notre système :

Cas d'utilisation : Accéder à un nouveau jeu en spécifiant la difficulté du jeu « New game ».

Acteur : Utilisateur

Description :

- 1- L'utilisateur lance l'application en cliquant sur l'icône.
- 2- Le système affiche l'écran d'accueil.
- 3- L'utilisateur choisit le bouton « nouvelle partie » dans le menu principal.
- 4- Le système affiche l'écran des difficultés.
- 5- L'utilisateur sélectionne la difficulté qu'il veut.
- 6- Le système affiche l'interface du jeu (la grille).
- 7- L'utilisateur complète et valide la grille.

**Figure III.3:** Cas d'utilisation « Accéder à un nouveau jeu en spécifiant la difficulté du jeu ».

Cas d'utilisation : Accéder à un jeu non terminé « Continue ».

Acteur : Utilisateur

Description :

- 1- L'utilisateur lance l'application en cliquant sur l'icône.
- 2- Le système affiche la page d'accueil.
- 3- L'utilisateur choisit le bouton « Continue » dans le menu principal.
- 4- Le système affiche l'interface du jeu.
- 5- L'utilisateur complète et valide la grille.

**Figure III.4:** Cas d'utilisation « Accéder à un jeu non terminé ».

Cas d'utilisation : Accéder à la description du jeu « About »

Acteur : Utilisateur

Description :

- 1- L'utilisateur lance l'application en cliquant sur l'icône.
- 2- Le système affiche la page d'accueil.
- 3- L'utilisateur choisit le bouton « About » dans le menu principal.
- 4- Le système affiche l'interface de description du jeu.

**Figure III.5:** Cas d'utilisation « Accéder à un jeu non terminé ».

### III.5.Conception

Le processus de conception de notre projet se caractérise par un seul niveau : le niveau applicatif.

Le niveau applicatif s'appuie essentiellement sur quelques diagrammes de l'extension du langage de modélisation UML, A cet effet nous avons adopté la démarche suivante:

- ✓ Après l'identification des différents acteurs ainsi que les cas d'utilisation qui sont mis en œuvre par ces acteurs, le diagramme de cas d'utilisation est élaboré.
- ✓ Chaque cas d'utilisation se traduit par un ou plusieurs scénarios. Chaque scénario fait l'objet d'une description sous forme graphique à l'aide d'un diagramme de séquence et un digramme d'activité.
- ✓ Une identification des classes est fournie par la synthèse des diagrammes de séquence, ainsi le diagramme de classe sera élaboré.

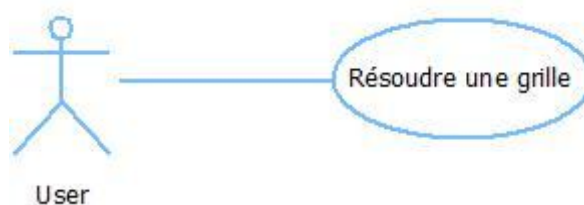
#### III.5.1.Les diagrammes des cas d'utilisation

Lors de la phase d'analyse nous avons pu identifier les acteurs ainsi que les cas d'utilisation associés à ces derniers. Ce qui nous donne l'opportunité d'élaborer le diagramme des cas d'utilisation. [8]

**Définition :** Les diagrammes de cas d'utilisation permettent de représenter un ensemble de cas d'utilisation, d'acteurs et leurs relations. Ils présentent la vue statique des cas d'utilisation d'un système et sont particulièrement importants dans l'organisation et la modélisation des comportements d'un système.

- ✓ **La relation d'inclusion (include) :** Elle indique que le cas d'utilisation source contient aussi le comportement décrit dans le cas d'utilisation destination. Cette relation permet de décomposer des comportements et de définir les comportements partageables entre plusieurs cas d'utilisations.
- ✓ **La relation d'extension (Extend) :** Elle indique que le cas d'utilisation source ajoute son comportement au cas d'utilisation destination. L'extension peut être soumise à des conditions.

En bleu sont représentés les états auxquels nous voulions arrivés dans nos scénarios.



**Figure III.6:** Diagramme de cas d'utilisation le plus général.

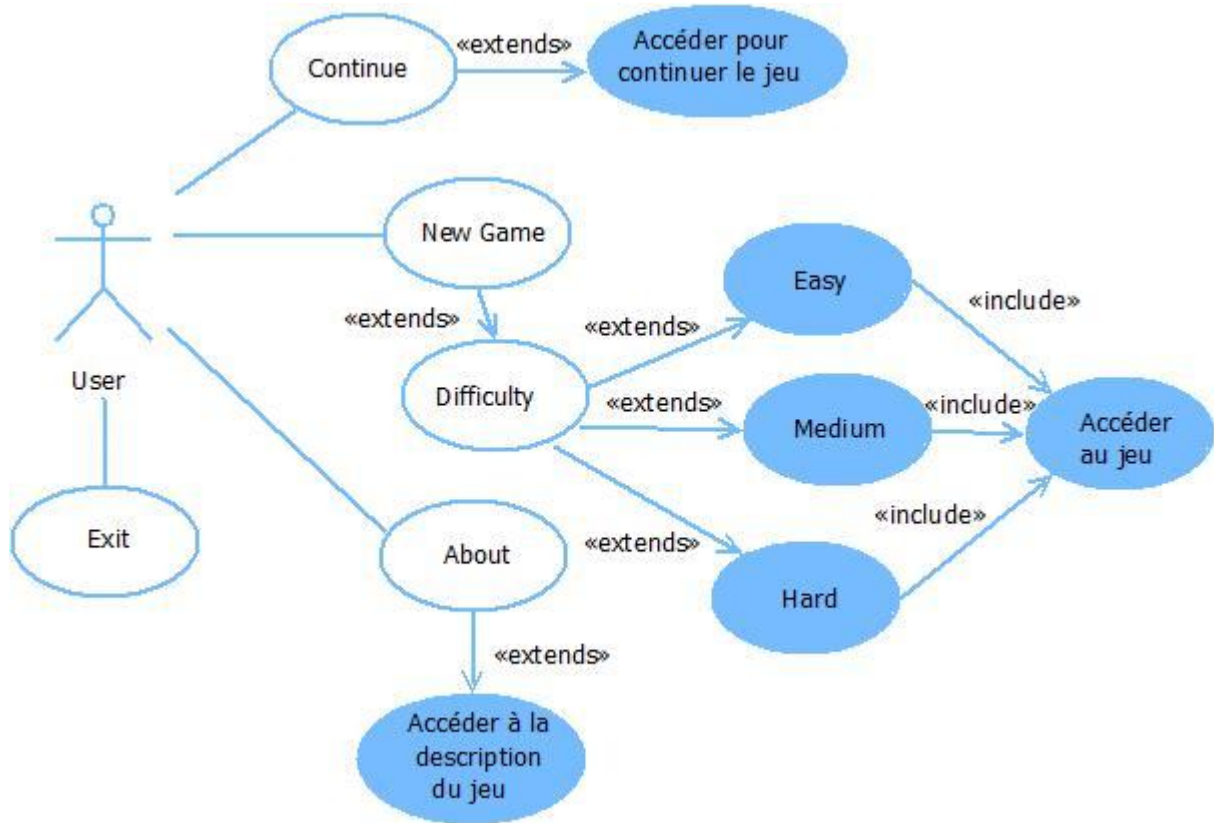


Figure III.7: Diagramme de cas d'utilisation générale.

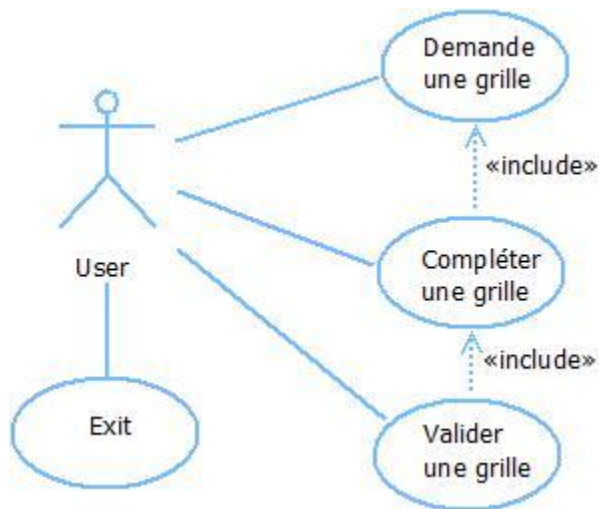


Figure III.8: Diagramme de cas d'utilisation du jeu en lui-même.

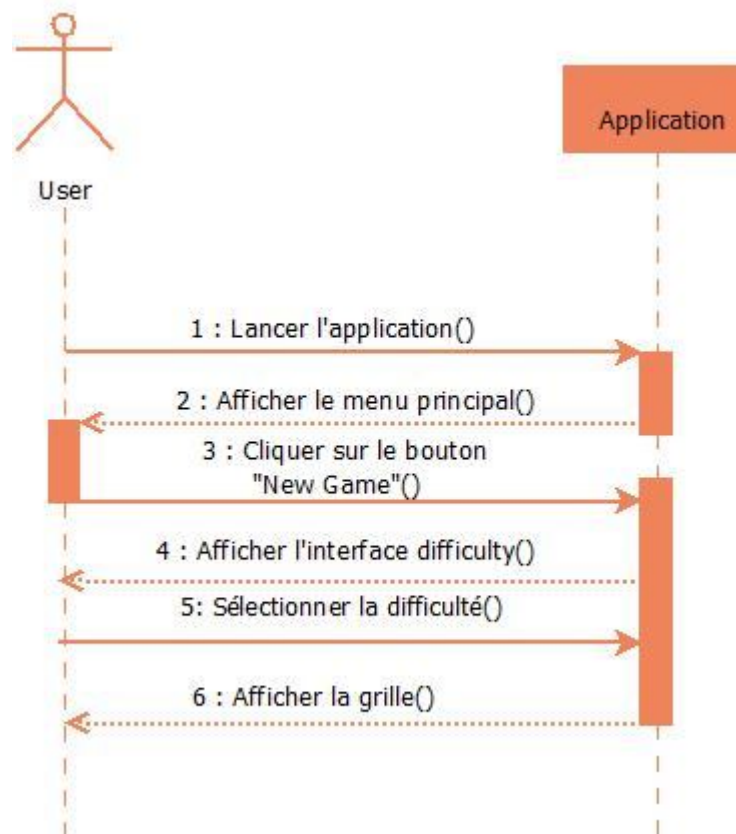
### III.5.2. Les diagramme de séquences

Un *diagramme de séquence* permet de spécifier les interactions qui existent entre un groupe d'objet selon un point de vue temporel, on y met l'accent sur la chronologie des envois de message. Le diagramme de séquence est toujours de haut vers le bas, il illustre l'ordre dans lequel les messages sont envoyés entre les objets. Il peut servir à illustrer un cas d'utilisation.

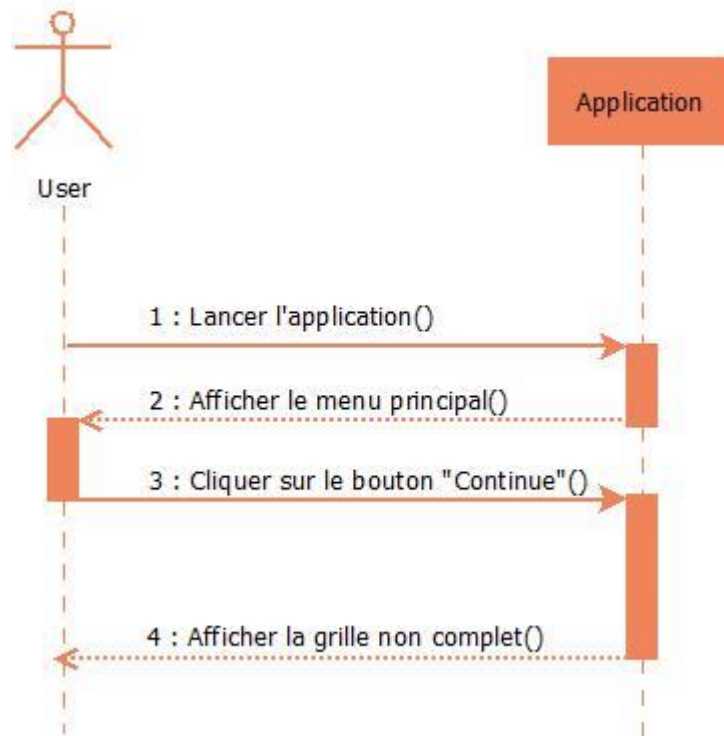
Les composants d'un diagramme de séquence sont les suivants :

**Les objets :** ils apparaissent dans la partie supérieure, ce qui facilite l'identification des classes qui participent à l'interaction.

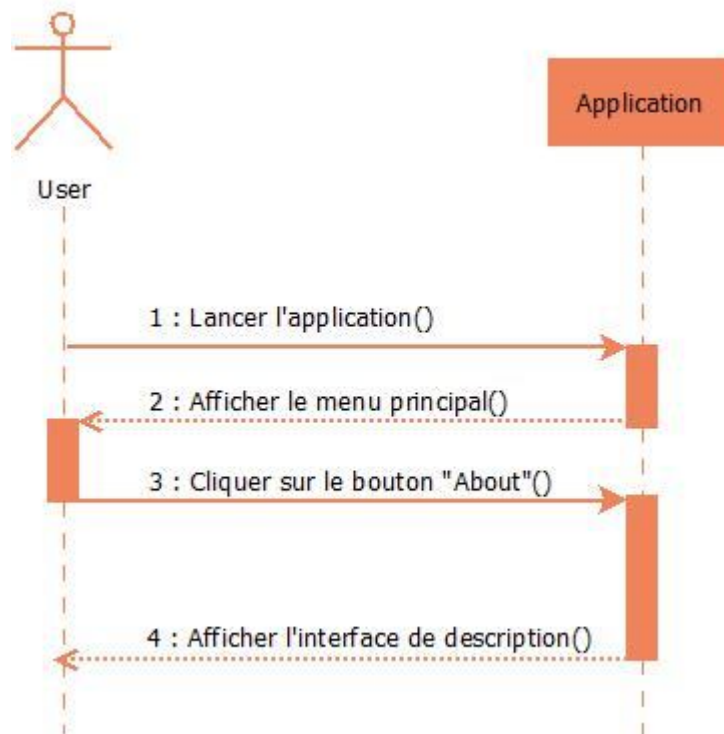
**Les messages :** ils sont représentés par des flèches directionnelles. Au-dessus des flèches directionnelles figurent un texte nous informant du message envoyé entre les objets.



**Figure III.9:** Diagramme de séquence de cas d'utilisation « Accéder à un nouveau jeu en spécifiant la difficulté du jeu ».



**Figure III.10:** Diagramme de séquence de cas d'utilisation « Accéder à un jeu non terminé ».



**Figure III.11:** Diagramme de séquence de cas d'utilisation « Accéder à la description du jeu ».

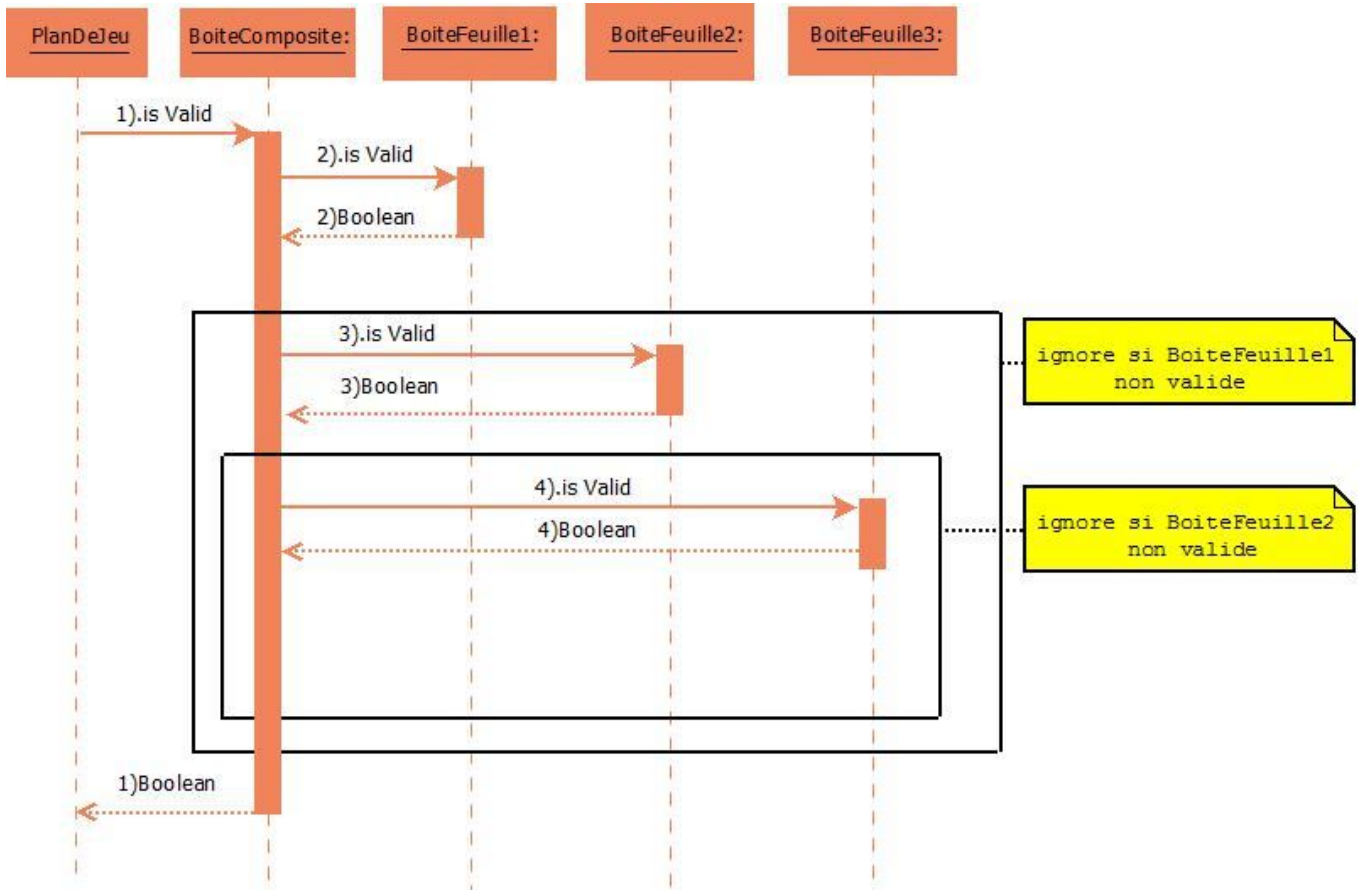
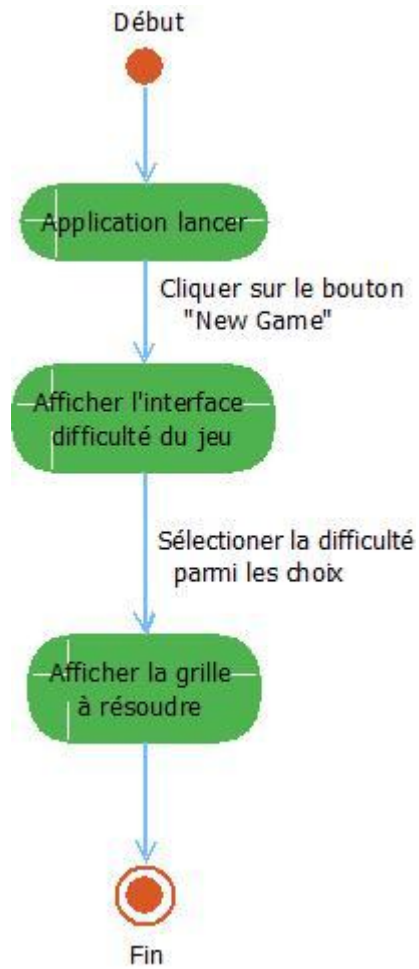


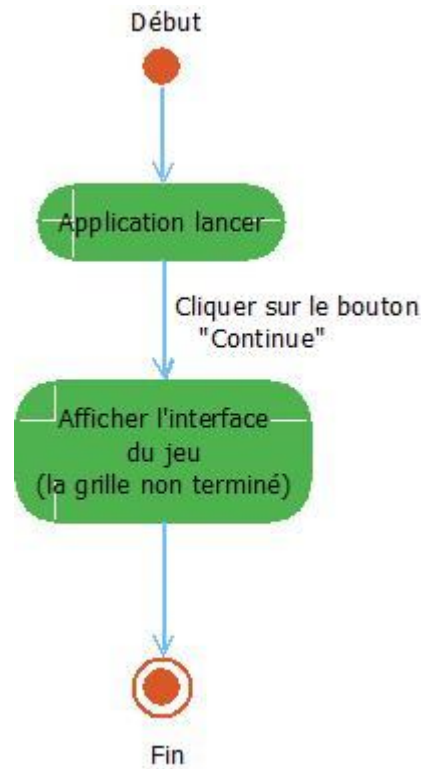
Figure III.12: Diagramme de séquence de cas d'utilisation « Validation d'une grille ».

### III.5.3. Les diagramme d'activités

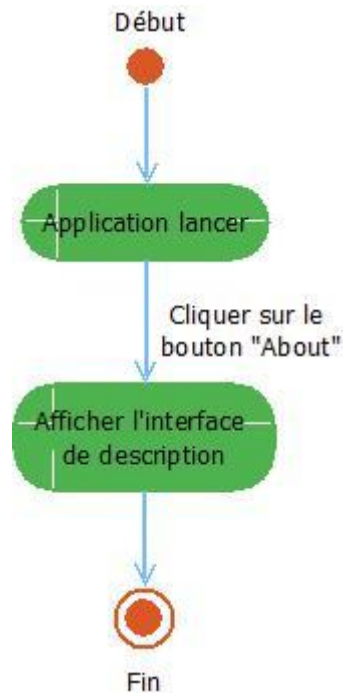
Un *diagramme d'activités* : apporte un point de vue complémentaire à l'aspect dynamique de la modélisation. Il offre un pouvoir d'expression très proche des langages de programmation objets. Il est donc bien adapté à la spécification détaillée des traitements en phase de réalisation. Un diagramme d'activités se concentre plutôt sur les activités entre les objets, c'est-à-dire, il met en évidence l'activité qui a lieu dans le temps, donc les opérations transmises entre les objets. [9]



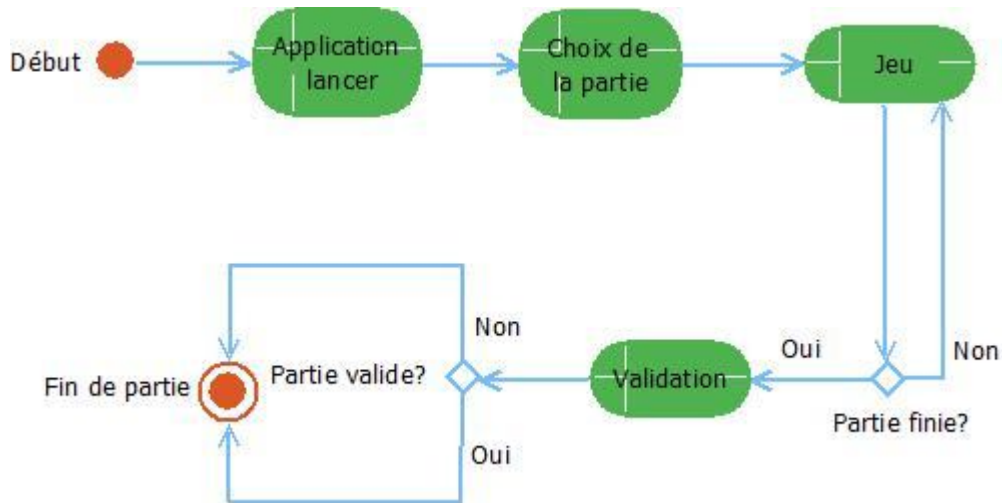
**Figure III.13:** Diagramme d'activité de cas d'utilisation « Accéder à un nouveau jeu en spécifiant la difficulté du jeu ».



**Figure III.14:** Diagramme d'activité de cas d'utilisation « Accéder à un jeu non terminé ».



**Figure III.15:** Diagramme d'activité de cas d'utilisation « Accéder à la description du jeu ».

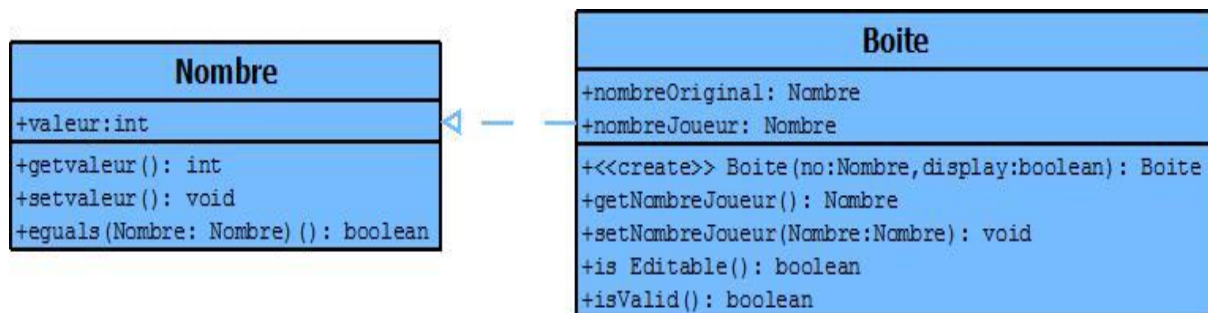


**Figure III.16:** Diagramme d'activité de cas d'utilisation « Validation d'une grille ».

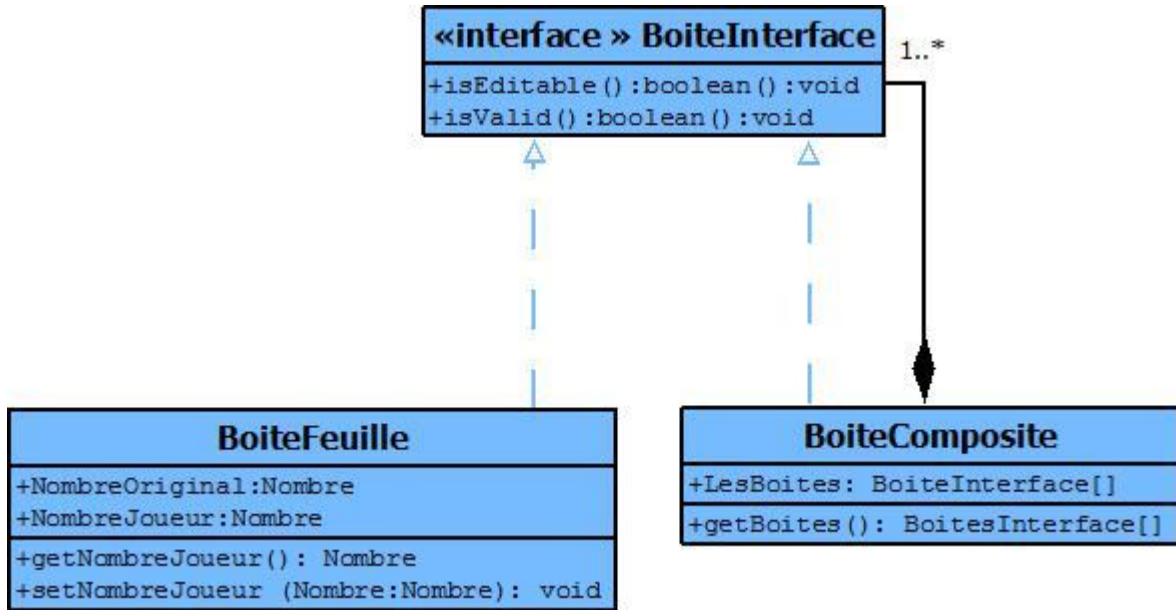
### III.5.4. Les diagramme de classe

Dans cette partie, nous allons passer à la modélisation de l'aspect statique de notre application, c'est-à-dire nous allons modéliser l'intérieur de notre système. Pour ce faire, nous allons utiliser les diagrammes de classes. [9]

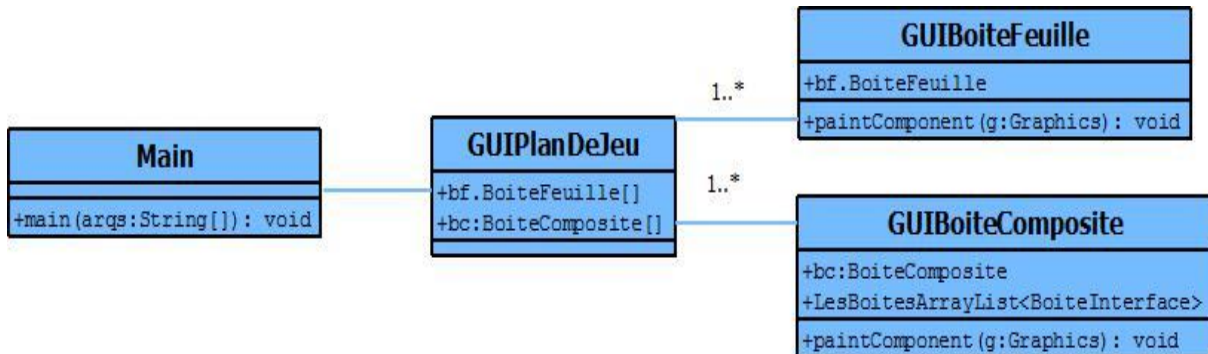
*Le digramme de classe* est considéré comme le plus important de la modélisation orientée objet. Alors que les diagrammes précédents (diagrammes de séquences et diagrammes d'activités) montrent le système du point de vue dynamique, le digramme de classe en montre la structure interne. Il contient principalement des classes reliées par des associations et chaque classe contient des attributs et des opérations.



**Figure III.17:** Diagramme de classe « les deux classes de base ».



**Figure III.18:** Diagramme de classe « Motif de conception composite pour les boites ».



**Figure III.19:** Diagramme de classe « affichage des variables et fonction principale ».

### **III.6.Conclusion**

La phase de conception sert à identifier les différents objets qui nous aident à certifier les fonctionnalités souhaitées. Tout en donnant une description complète de ces diagrammes, ce qui permet une meilleure compréhension du système.

Après ce chapitre, on passe au dernier chapitre dont on décrit le travail réalisé ainsi expliqué l'environnement logiciel.

Chapitre IV :

Réalisation.

### **IV.1.Introduction**

La tâche de réalisation ou d'implémentation est la phase finale de l'élaboration d'un système. Elle permet aux matériels et logiciels d'entrer en fonction pour passer de l'expression d'un besoin informatique à un système fonctionnel fiable. Cette phase consiste à traduire la conception exprimée à l'aide d'un formalisme en un code source écrit dans un langage donné. Pour ce faire on va montrer des aperçus d'écran qui décrivent les diverses fonctionnalités mise en place dans l'application.

### IV.2. Environnement de travail

#### IV.2.1. Environnement matériel

Dans notre projet, l'environnement de travail disponible pour le développement de l'application est le suivant :

- PC Portable : Intel Core i3, 2.40 GHZ, 4,00 GO de mémoire vive, Windows 7.

#### IV.2.2. Environnement cible

Notre application est destinée pour tous les Smartphones équipé d'un système Android 2.3.3 dite Gingerbread et plus.

#### IV.2.3. Environnement logiciel

Avant de pouvoir faire du développement, il faut nécessairement préparer l'environnement de travail. Dans notre cas, on a choisi de faire notre développement mobile sur la plateforme Eclipse. Pour la préparer, on a effectué différentes tâches qui sont :

- ✓ Installation Eclipse.
- ✓ Installation SDK Android.
- ✓ Intégration SDK Android sous Eclipse pour pouvoir créer des projets Android.

Après la préparation de logiciel de travail eclipse et pour pouvoir développer une application dans de bonnes conditions, il faut bien savoir choisir son environnement de développement selon les besoins, de ce fait, on a développé une application pour mobile en JAVA, qui va être compilé en fichier DEX par le Dalvik VM ensuite déployé sous une extension APK pour être installé sur un mobile fonctionnant avec le système Android, pour cela on a choisi Eclipse IDE qui nous permettra de réaliser ce travail.

##### IV.2.3.1. Système d'exploitation

[Notre application a été développée sous le système d'exploitation Windows 7 service pack1, en langage JAVA]. Elle peut être ainsi intégrée dans n'importe quel autre système d'exploitation supportant la machine virtuelle java

##### IV.2.3.2. Langage de programmation

Le langage java est un langage de programmation orienté objet mis au point par Sun Microsystems. Sa caractéristique principale est qu'il est indépendant de toute plateforme, il est possible d'exécuter des programmes java sur tous les environnements qui possèdent une « Java Virtual Machine » (JVM).

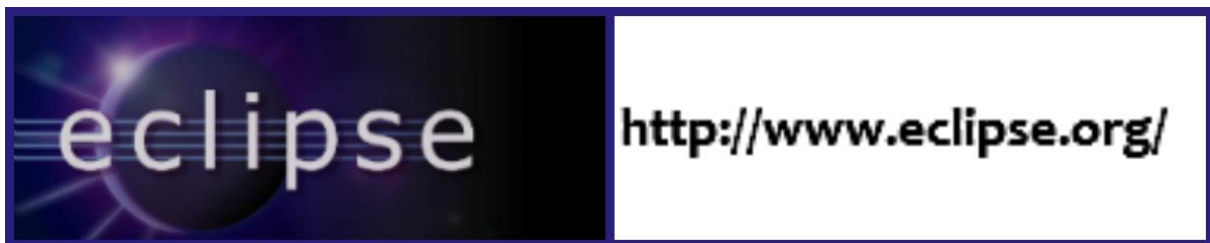
Ce concept est à la base du slogan de Sun pour java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). Sun fournit aussi gratuitement un ensemble d'outils et d'API pour permettre le développement de programmes avec ce langage, ce kit est nommé JDK (Java Development Kit).

Java est caractérisée aussi par la réutilisabilité de son code ainsi que la simplicité de sa mise en œuvre.

### IV.2.3.3.Eclipse

Eclipse est un IDE qui permet de programmer dans différents langages grâce à ses nombreux plug-ins et notamment le plug-in d'Android. Une interface spécifique permet de gérer des fichiers java et de compiler ses programmes. Les fichiers sont organisés selon une arborescence qui correspond aux paquetages java définis. L'analyse syntaxique permet de mettre en valeur les mots clés dans les fichiers java.

Eclipse dispose aussi d'un système d'auto complétion des fonctions, de détection des erreurs syntaxiques en temps réel sans oublier un système de débogage permettant d'exécuter ses programmes pas à pas.



**Figure IV.1:** Logo Eclipse et site web.

### IV.2.3.4. IDE Eclipse

L'IDE Eclipse est un environnement de développement permettant d'écrire, compiler, déboguer et déployer des programmes. Il est écrit en java. Par ailleurs il existe un grand nombre de modules pour étendre l'IDE Eclipse.

La spécificité d'Eclipse IDE vient du fait de son architecture totalement développée autour de la notion de plugin : toutes les fonctionnalités de cet atelier logiciel sont développées en tant que plug-in.

Pour le développement de mon application, on a choisi Eclipse KEPLER (Son interface principale est donnée dans la figure IV.1 ), avec le plugin ADT de Google.

## Chapitre IV : Réalisation

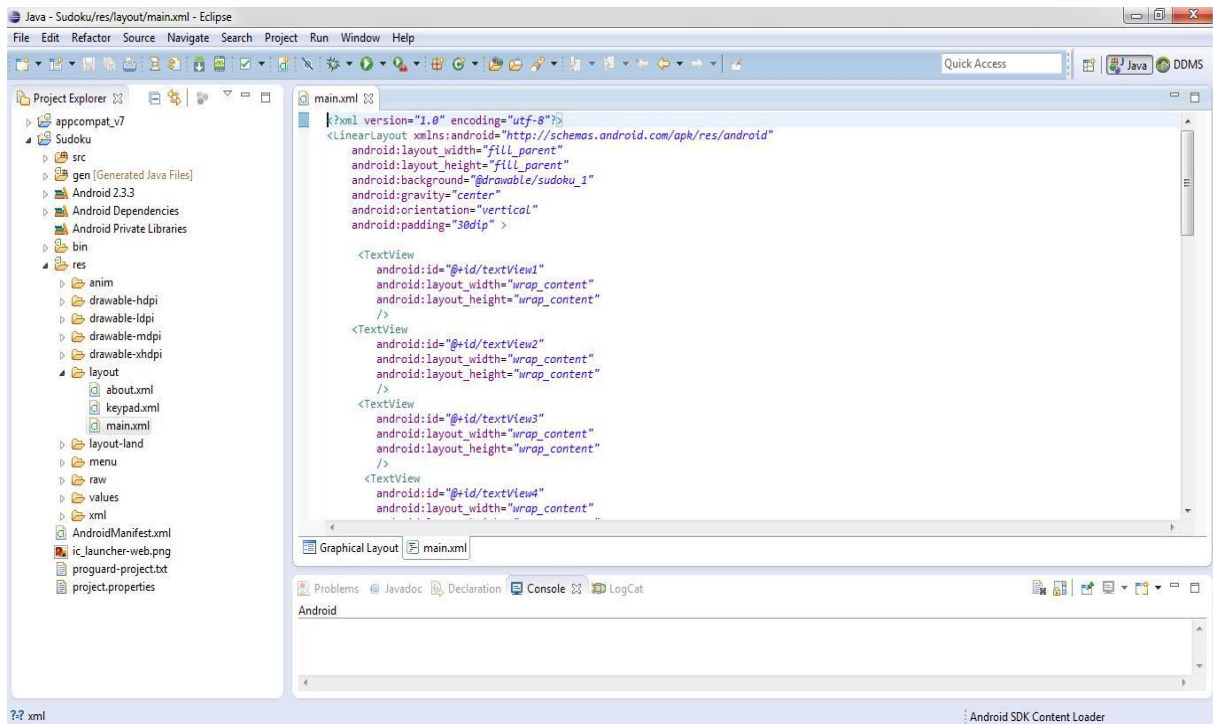


Figure IV.2: plate-forme d'éclipse.

### IV.3.Prise en main de l'environnement Android

La première étape de notre travail avec l'environnement Android a été d'appréhender le SDK, l'architecture et le développement d'une application ainsi que son déploiement sur un terminal embarquant Android.

Google a mis en place un grand nombre d'outils pour aider les développeurs Android.

#### ❖ Le portail des développeurs :

La première chose à visiter est le portail des développeurs Android, mis en place par Google. [N4].



**Figure IV.3:** Portail des développeurs Android.

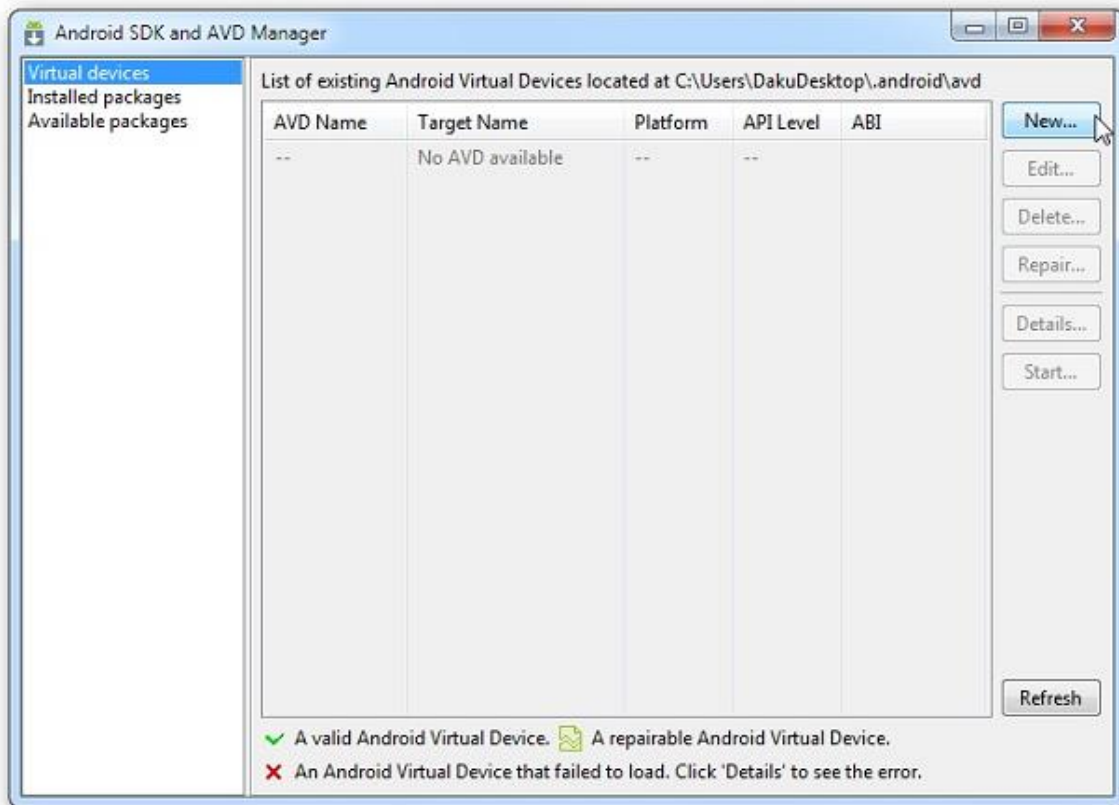
Très complet, ce site présente Android, explique comment installer et utiliser les différents outils (SDK, NDK etc.), propose un ensemble de tutoriels et articles concernant le développement d'applications Android, expose la référence de l'API Android ainsi que les actualités liées à Android.

Le tout est très bien fait et permet de rapidement être confortable vis-à-vis du développement sur Android.

### IV.3.1. Software Development Kit (SDK)

L'outil le plus important est le SDK Android. Facile à installer, il permet de télécharger tous les outils indispensables au développement d'applications. Un petit logiciel permet d'abord de télécharger les différentes versions du SDK (une version du SDK par version d'Android : 1.4, 1.5, 1.6, 2.0 etc.). Il permet également de télécharger les différentes versions des Google APIs (APIs pour intégrer des fonctionnalités liées aux services Google tels que Maps etc.) ou de la documentation JavaDoc. Son fonctionnement est similaire aux gestionnaires de paquets de Linux. [N5].

C'est un kit de développement basé sur le langage Java. Il s'agit des outils que Google fournit pour interagir avec Eclipse.



**Figure IV.4:** Interface d'installation du SDK Android.

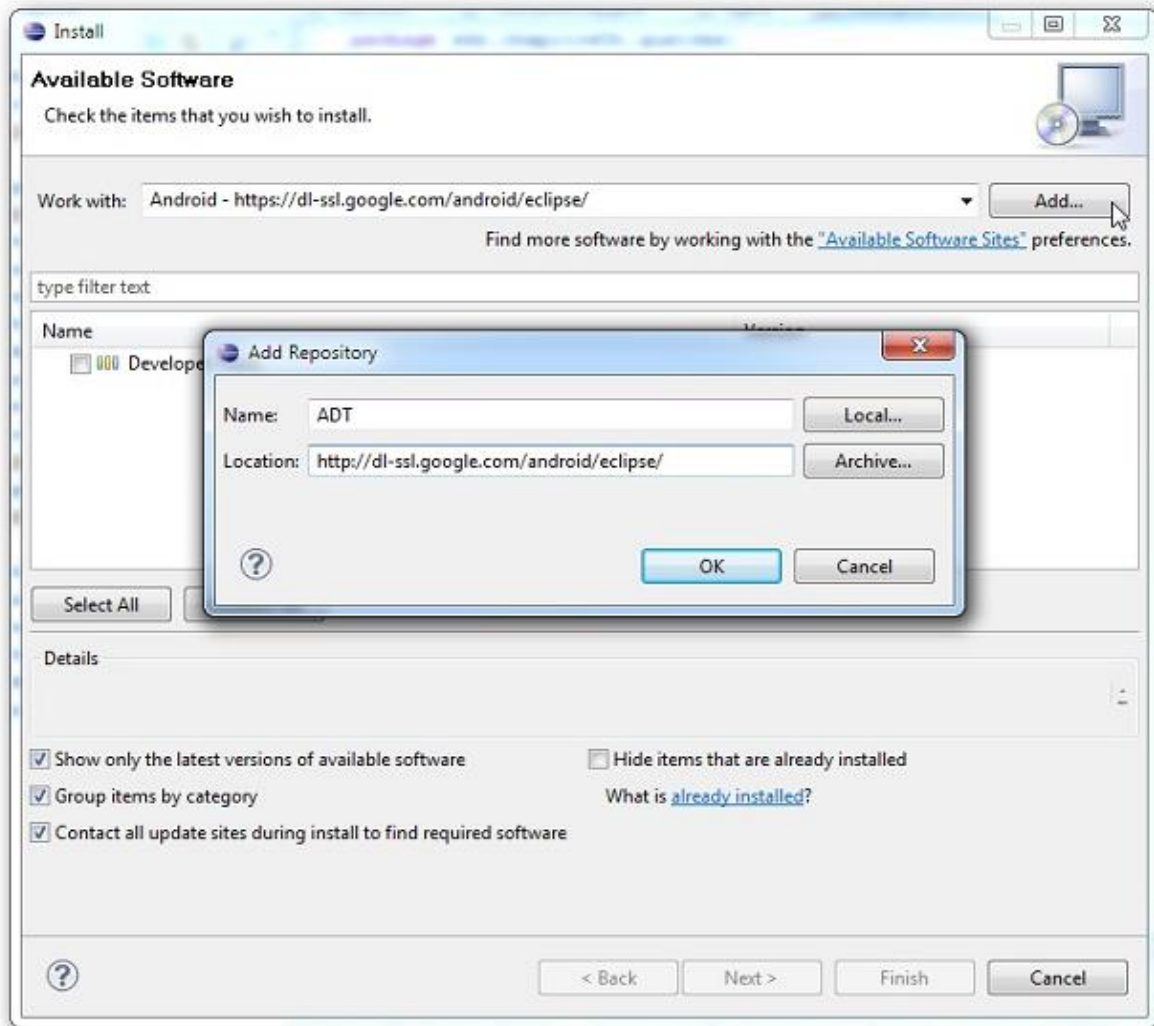
### IV.3.2. Le plugin ADT pour Eclipse

Eclipse est l'Environnement de Développement Intégré (ou IDE) le plus largement utilisé pour la programmation Java; très performant, il est de plus gratuit et open source. [N6]

Le langage privilégié pour le développement d'applications Android est justement Java. Google a donc tout naturellement conçu un plugin (Un plugin est un module qui complète un logiciel hôte pour lui apporter de nouvelles fonctionnalités) pour Eclipse.

Android Development Tools, ou ADT, est très complet et surtout très pratique : conception graphique d'interfaces utilisateur, debug distant sur un téléphone, gestion de l'architecture de fichiers d'une application etc.

Pour développer en Android, on a installé le plugin Android qui rajoutera à Eclipse les fonctionnalités spécialisées dans le développement sous Android.



**Figure IV.5:** Interface d'installation ADT.

### IV.3.3.Android

*Android* est un système d'exploitation open source pour smart phones, PDA et terminaux mobiles. Il s'agit également d'un langage de programmation basé sur le JAVA et sur le XML. Le JAVA permet d'interagir avec l'utilisateur en faisant la liaison entre l'interface graphique et la base de données. Quant au XML, il permet notamment de décrire les interfaces graphiques.

### IV.4.Mode de fonctionnement de l'application

#### IV.4.1.Le menu principal de l'application

En premier lieu l'utilisateur lance l'application en appuyant sur l'icône lui correspondant dans le menu du Smartphone, ce qui donne ceci :



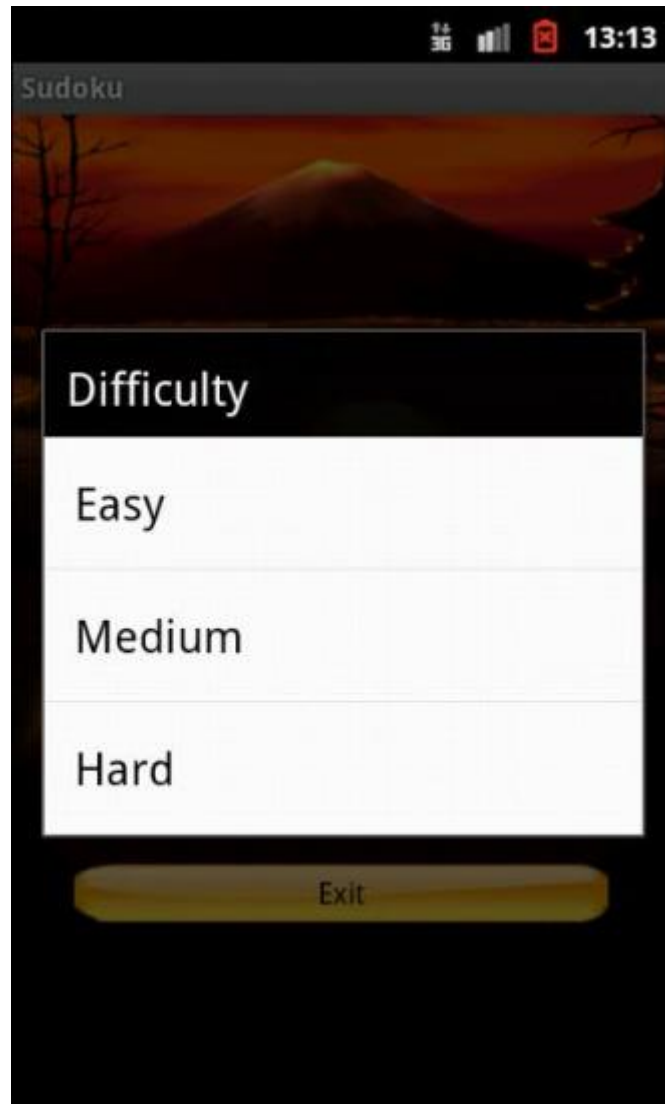
**Figure IV.6:** Menu principal.

Dispose de cinq boutons :

- 1 : continuer une partie.
- 2 : commencer un nouveau jeu.
- 3 : ouvre la description du jeu
- 4 : quitter l'application.

### IV.4.2. Interface difficulté

En appuyant sur le bouton « New Game » on aura ceci :

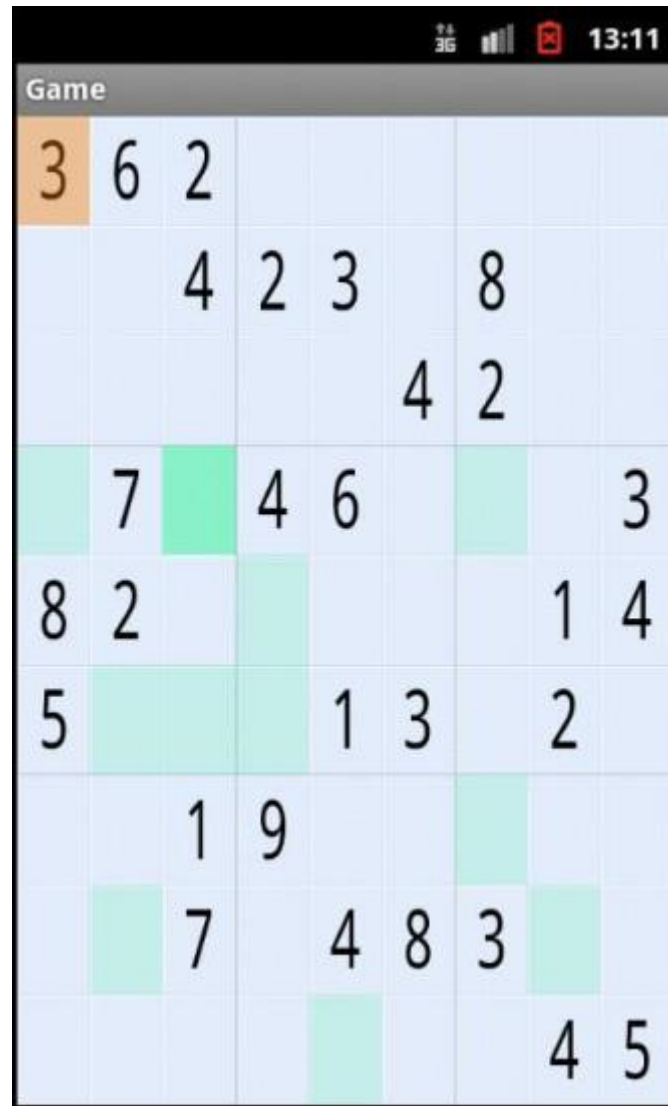


**Figure IV.7:** La liste des difficultés.

Pour choisir une difficulté il suffit de cliquer sur la difficulté désiré.

### IV.4.3. Interface de jeu

Après le choix de difficulté on aura ceci :



**Figure IV.8:** Interface de jeu (la grille).

Il suffit de remplir les cases vides avec les nombres correspondant pour valider le jeu.

### IV.4.4. Interface de description

En appuyant sur le bouton « about » on aura ceci :



**Figure IV.9:** Interface de description.

Interface explique la règle du jeu.

### **IV.5. Conclusion**

Dans ce dernier chapitre nous avons présenté une vue générale sur l'application que nous avons eu pour but de réaliser. Nous avons montré les différents rendus visuels que nous offrait l'application ainsi que ses nombreux composants et le rôle et les besoins de chacun.

## **Conclusion Générale**

Au cours de ce projet, nous avons développé une application mobile éducative permettant à son utilisateur de faire travailler son cerveau tout en se divertissant. La mobilité de notre application, du fait qu'elle soit directement téléchargeable sur un Smartphone fonctionnant sous système Android (version 2.3.3 et plus) permet à son utilisateur une grande liberté d'utilisation.

Cette réalisation nous a aussi permis d'enrichir notre savoir et de développer nos connaissances informatiques, notamment dans le domaine de la programmation et de la mobilité. En effet, l'application a exigé des connaissances du langage JAVA et des outils de développement indispensables à sa réalisation. La mise en œuvre de notre travail a exigé des connaissances très approfondies en la matière ainsi qu'une bonne maîtrise de la configuration d'éclipse et de l'environnement Android.

Par ailleurs ce projet de fin d'étude est une expérience humaine très enrichissante. Nous avons pu s'entraider et critiquer afin de mener à bien notre travail. Cela nous a appris que seul l'acharnement et le travail assidu pouvaient nous mener à trouver des solutions. Ce fut donc une épreuve qui nous servira de tremplin pour continuer sur notre lancée, Et ainsi accomplir de nouveaux projets pour le système Android et les autres systèmes d'exploitation mobiles, ainsi que dans d'autres domaines, car l'informatique est un monde tellement vaste qui ne demande qu'à être exploré.

Pour finir, nous espérons que notre application répondra aux besoins de futurs utilisateurs et que notre mémoire apportera un soutien considérable pour de prochains développeurs désireux de créer une application mobile pour le système d'exploitation Android.

## **Annexe : Manuel d'installation et de mise en marche.**

1. Installation de l'Eclipse [N7].
2. Installation du SDK Android [N8].
3. Installation du plugin ADT sous Eclipse [N9].
4. Création d'un Android Virtual Device (AVD) :

***La Configuration d'un nouveau AVD se fait en remplissant les champs suivants lors de sa création:***

- ✓ Name : Le nom à donner à votre émulateur (sans espace).
- ✓ Target : La version du SDK Android sur lequel l'émulateur doit fonctionner.
- ✓ SD Card: Configuration de la SD Card (Taille, etc.). Ce champ est facultatif.
- ✓ Skins : Choix du thème de l'émulateur. Des émulateurs préconfigurés se trouvent dans Built-in.

Hardware: Cette partie permet de rajouter le matériel et de le personnaliser.

## Bibliographie

[1]: Wiley & Sons, 2011, Next Generation Mobile Communications Ecosystem: Technology Management for Mobile Communications John Wiley & Sons, 25 févr. 2011.

[2]: Wiley & Sons, 2008, Developing Software for Symbian OS 2nd Edition: A Beginner's Guide to Creating Symbian OS v9 Smartphone Applications John Wiley & Sons, 28 feb. 2008.

[3]: Créez des applications pour Android (site du zéro).

[4] : Développement d'application pour Android, M.Dalmau- IUT  
de Bayonne- Pays Basque.

[5]: Alain Le Guennec, « génie logiciel et méthode formelle avec UML spécification et généralisation de testes »juin 2001.

[6]: Grady Booch, James Rambough et Ivar Jacobso « Le guide de l'utilisateur UML », Edition Eyrolles, 2001.

[7]: Jim Conallen, « Concevoir des application web avec l'UML » Edition Eyrolles, Octobre 2000.

[8]: " Analyse et Conception Orientées Objet avec UML et réalisation en C++ " Guide étudiant, Edition : Sun 1999, pdf.

[9]: Grady Booch et al, " Le Guide de l'utilisateur UML ", Edition : Eyrolles, 2003.

## Netographie

[N1] : <http://www.smartphonefrance.com/>

[N2] : <http://fr.wikipedia.org/>

[N3] : <http://developer.android.com/guide/basics/what-is-android.html>.

[N4]: Portail des développeurs Android [En ligne].

-<http://developer.android.com/>.

[N5]: Référence du SDK Android[En ligne].

-[http://developer.android.com/sdk/ndk/1.5\\_r1/index.htm](http://developer.android.com/sdk/ndk/1.5_r1/index.htm).

[N6] : ADT Plugin android.

-<http://developer.android.com/tools/sdk/eclipse-adt.html>.

[N7]: <http://www.eclipse.org/downloads/>

[N8]: <http://developer.android.com/index.html>

[N9]: <https://dl-ssl.google.com/android/eclipse/>

[N10]: <https://dl-ssl.google.com/android/eclipse/>.

[N11]: <http://www.java.com/fr/>.

[N9] : <http://www.android.com/>.

[N10] : <http://www.wikipedia.fr>.