

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

pour l'obtention du diplôme de Licence en Informatique

Thème

Développement d'une application mobile
Le jeu « smile »

Réalisé par :

- REMACI Zineb Yasmina
- GHITRI Salim

Présenté le 10 Juin 2014 devant la commission d'examination composée de :

- Mr M. BENAZZOUZ (Encadreur)
- Mr I.SMAHI (Examineur)
- Mr M.MESSABIHI (Examineur)

Année universitaire : 2013-2014

REMERCIEMENTS

Louange à Dieu qui nous a donné la force, le courage, et l'espoir nécessaire Pour accomplir ce travail et surmonter l'ensemble des difficultés.

Nous exprimons notre gratitude, remerciements à nos parents qui ont fait de leur mieux pour nous aider.

Nous tenons à remercier vivement :

- ✓ Notre encadreur Monsieur M.BENAZZOUZ pour ces conseils et son suivi durant la réalisation de notre projet.
- ✓ *Messieurs I. SMAHI et M. MESSABIHI* qui ont bien voulu faire partie du jury.

Nous remercions aussi les personnes qui nous ont aidés et encouragés le long de ce travail.

DEDICACE

Je dédie ce travail à:

Mes parents

Vous vous êtes dépensés pour moi sans compter.

En reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d'atteindre cette étape de ma vie.

Avec toute ma tendresse.

A mes sœurs.

A mes oncles, tantes, cousins et cousines.

Spécial dédicace à mes amies OMILLOU Horiya et MOKADEM Asma qui m'ont bien aidé le long de mon parcours universitaire.

A tous les membres de ma promotion.

A mes amis.

A tous mes professeurs.

REMACI Zineb Yasmina

DEDICACE

Je dédie ce travail à:

Mes parents

Vous vous êtes dépensés pour moi sans compter.

*En reconnaissance de tous les sacrifices consentis par
tous et chacun pour me permettre d'atteindre cette
étape de ma vie.*

Avec toute ma tendresse.

A mes sœurs.

A mes oncles, tantes, cousins et cousines.

A tous les membres de ma promotion.

A mes amis.

A tous mes professeurs.

GHUITRI Salim

Table des matières

Introduction générale.....	4
Chapitre I.....	5
I. Introduction	5
II. Application mobile	5
II. 1 Définition	5
Application natives	5
Application web	5
Application hybrides.....	6
III. Le système d'exploitation de téléphone mobile	7
III.1 Définition	7
III.2 Les différentes plateformes mobiles	7
III.2.1 iOS	7
III.2.2 Android.....	7
III.2.3 Windows Phone	7
III.2.4 BlackBerry	8
III.2.5 Symbian OS	8
IV. Quelques statistiques sur le marché.....	8
V. Conclusion	9
Chapitre II	10
I. Introduction	10
II. Pourquoi avons- nous choisie Android ?.....	10
III. Android et Google	10
IV. Les différentes versions d'Android	11
V. Avantages et inconvénients de l'Android	12
VI. Architecture du système Android	12
VII. Les outils et les données auxiliaires	13
1. L'environnement Eclipse	13
2. JDK5 ou JDK6	13

3. SDK	13
4. Android Development Tools ADT	13
5. Android Virtual Device Manager AVD	13
6. L'émulateur	14
VIII. Composantes d'une application Android	15
1. Les Activité	15
2. Les Services	15
3. Les fournisseurs de contenus Content Provider	15
4. Les Broadcast receivers	15
5. Les Intents	15
IX. Développement d'une application sur Android	15
X. Cycle de vie d'une activité	16
XI. Conclusion	17
Chapitre III	18
I. Introduction	18
II. Pourquoi le langage java ?	18
III. Langage XML	18
IV. Accès et gestion des bases de données locales	19
V. Les avantages du développement du jeu	19
VI. Description du jeu	19
VII. Réalisation	19
1) Méthodologie de conception UML	19
2) Menu de l'application	20
2.1 L'interface graphique	20
2.1.1 Les activités d'accueil	22
2.1.2 Les activités du jeu	27
2.2 Réalisation	28
2.2.1 Etapes de réalisation	28
A. Changement de couleur	28
B. Affichage des images	28
C. Changement du score	29

2.2.2 Réalisation des versions du jeu	29
2.2.2.1 Joueur vs joueur	29
2.2.2.1 Joueur vs Système	30
VIII. Risque et Contraintes de développement	35
IX. Conclusion.....	35
Conclusion générale	36
Références	37
Liste des figures	38
Liste des abréviations	39

Introduction générale:

« Le mobile est en train de changer le monde. » [1]

Aujourd'hui, le monde connaît une avancée considérable dans l'utilisation des appareils téléphonique portables grâce aux applications mobiles, ces dernières sont capables de satisfaire les besoins actuelles des utilisateurs avec de nombreuses fonctionnalités et en offrant plusieurs services.

Les téléphones mobiles peuvent remplacer et dans différentes situations, l'ordinateur portable. C'est pour quoi, les systèmes d'exploitation de ces derniers (SE) deviennent de plus en plus importants, particulièrement chez les fournisseurs, les développeurs et les éditeurs d'application pour les mobiles.

En effet, depuis l'apparition des Smartphones avec leurs systèmes d'exploitation, ces équipements se sont transformés en véritables terminaux de jeux très lucratifs, il est évident que le domaine va connaître, encore de beaux jours dans les années qui viennent.

Notre objectif, dans ce modeste travail, est l'étude des applications mobiles, leurs techniques et leurs outils de développement, dans le but de concevoir un jeu mobile, celui-ci sera installé sur des terminaux Android.

Notre étude peut ainsi être subdivisée en trois chapitres :

- Le premier chapitre présentera l'application mobile puis ses caractéristiques, les systèmes d'exploitation pour les téléphones mobiles et s'achèvera sur des statistiques du marché.
- Le deuxième chapitre aura pour tâche d'étudier l'art de l'Android et ses différentes versions et de définir l'environnement du travail ainsi que les outils correspondants.
- La réalisation et l'implémentation de notre application fera l'objet du troisième chapitre dans lequel nous illustrerons les différentes parties de l'application.
- Enfin, nous terminerons ce document par une conclusion générale.

I. Introduction :

Les applications mobiles prennent de plus en plus d'espace dans l'utilisation de nos terminaux mobiles, elles sont conçues pour des plateformes mobiles et utilisées pour des services de l'information, médias sociaux, jeux etc....

Dans ce chapitre nous aborderons la notion de l'application mobile et ses caractéristiques, ses types, la notion de système exploitation mobile et pour finir nous verrons des statistiques sur le marché.

II. Application mobile :

II.1 Définition:

Une application mobile est un logiciel applicatif téléchargeable de façon gratuite ou payante, et exécutable à partir du système d'exploitation du téléphone mobile, un téléphone intelligent ou une tablette numérique. [2]

On peut distinguer plusieurs solutions pour créer ces applications mobiles : les applications natives, les applications web et les applications hybrides.

• Applications natives :

Une application native est une application mobile qui est développée spécifiquement pour un des systèmes d'exploitation utilisés par les smartphones et les tablettes (iOS, Android, Windows Phone etc.). Elle est conçue avec un langage spécifique à son système d'exploitation et ne peut être distribuée que par l'intermédiaire des plateformes d'applications qui contrôlent sa nature et ses contenus.

• Applications Web :

Une web application mobile est une application développée en HTML accessible et exécutable par le biais d'un navigateur Internet pour téléphone mobile.

Elle utilise le navigateur du smartphone et ne nécessite pas forcément de télécharger l'application. Elle est normalement accessible par tous les smartphones quelque soit leur marque et leur système d'exploitation.

La web application mobile « complète » les applications natives qui sont développées spécifiquement pour un système d'exploitation et qui doivent être téléchargées et installées par les mobinautes.

• Applications hybrides :

L'application hybride est une application pour mobiles qui combine des éléments HTML5 sous forme de web application mobile et des éléments d'une application native permettant l'utilisation des fonctionnalités natives des Smartphones et d'être distribuée

en tant qu'application sur les plateformes d'applications (App Store, Android Market, etc.).

III. Le système d'exploitation du téléphone mobile :

Comme l'ordinateur, le téléphone mobile fonctionne grâce à son système d'exploitation.

III.1 Définition:

Le système d'exploitation mobile est un ensemble de programmes, responsables de la liaison entre les ressources matérielles de l'appareil et ses applications logicielles. Il assure le démarrage et le fonctionnement du téléphone mobile. [3]

III.2 Les différentes plateformes mobiles :

Actuellement, il existe plusieurs systèmes d'exploitation grâce à la concurrence, comme Windows Mobile, Palm OS, Symbian, BlackBerry, iOS qui sont des systèmes propriétaires. De plus, il y a les plateformes libres ou code source ouvert comme Moblin.org, Ubuntu MID Edition, Android, etc.

III.2.1 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 OS X dont il partage les fondations (le kernel hybride XNU basé sur le micro-noyau Mach, les services Unix et Cocoa, etc.). iOS comporte quatre couches d'abstraction, similaires à celles de Mac OS X : une couche « Core OS », une couche « Core Services », une couche « Media » et une couche « Cocoa ». [4]

III.2.2 Android :

Android est développé par l'Open Handset Alliance. Il a été annoncé en 2007 et en 2008, il est devenu une plateforme en code source ouverte.

L'Android est basé sur le noyau Linux et utilise la plateforme Java pour ses applications.

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

III.2.3 Windows Phone :

Windows a été lancé en novembre 2010 par **Microsoft**, il succède Windows Mobile en étant plus orienté vers un grand public. Il est basé sur un noyau Windows CE.

Cet OS mobile équipe uniquement des smartphones, c'est le futur Windows 8 qui est annoncé comme pouvant équiper des tablettes à sa sortie.

Le développement d'applications pour Windows Phone 7 est possible via deux langages au choix : C# et VB.NET. Windows Phone 7 supporte deux plateformes de développement, que l'on peut mixer dans la même application, et toutes les deux sont bâties sur le .NET Compact Framework.

Microsoft propose ses Windows Phone Developer Tools pour le développement des applications. Ce package, gratuit, comprend :

Visual Studio 2010 Express, la version légère casino online et gratuite de Visual Studio 2010. Un émulateur Windows Phone Expression Blend, pour la création des écrans.

III.2.4 BlackBerry :

Du fabricant canadien RIM (Research In Motion) , fut un précurseur sur le marché des PDA et smartphones.

Les possibilités de développement sur BlackBerry :

- Le SDK Java, et le plugin Eclipse associé, les BlackBerrys disposant d'une JVM spécifique.
- Le SDK pour Adobe AIR, afin de développer des applications Flash en Action Script.

III.2.5 Symbian OS :

Nokia créa Symbian OS en 1998 en compagnie de Panasonic, Psion, Ericsson et Motorola. Nokia fut ensuite le principal utilisateur de Symbian pendant de nombreuses années pour équiper ses téléphones mobiles et smartphone, et racheta tous les droits du consortium **Symbian Ltd** en 2008.

Aujourd'hui, la plateforme pour téléphones mobiles Symbian succède à **Symbian OS et Nokia Series 60**, en unifiant ces deux composantes système.

Auparavant, Symbian OS nécessitait une surcouche pour présenter une IHM aux utilisateurs. Le framework Series 60 de Nokia était alors un de ceux couramment utilisés dans ce but, en compagnie d'UIQ et Java ME. Le développement, réputé difficile, était réalisé en C. Il peut aujourd'hui être fait principalement avec le Framework **QT**, toujours en **C**, via les IDE **QT Creator** ou **Carbide**. La plateforme Symbian, en perte de vitesse, a été cédée à Accenture en 2011 suite au partenariat de Nokia et Microsoft autour de Windows Phone.

IV. Statistiques sur le marché :

Selon le rapport du cabinet américain Gartner¹, publié sur les téléchargements d'applications mobiles dans le monde : le marché de l'application mobile représente 25 milliard de dollars comme chiffre d'affaire en 2013, et il estime qu'en 2014 il correspondrait à 58 milliard de dollar.

En moyenne, 102 milliards d'applications téléchargées en 2013 contre 64 milliards en 2012, le nombre de ces téléchargements pourrait atteindre 139 milliards en 2014, et près de 269 milliards en 2017.

Android domine le marché des plateformes, en 2013 il dépasse les 80%, Apple chute à 12,1% et Microsoft à 3,6%, BlackBerry toujours plus bas en occupant seulement 1.7%. [5]

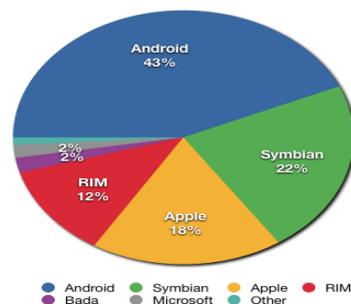


Figure I.1 Part du marché des smartphone par systèmes d'exploitation.

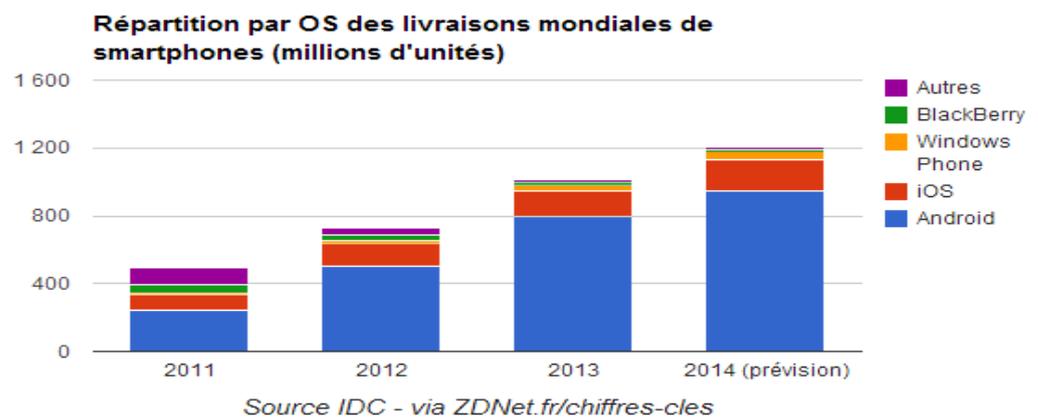


Figure I.2 livraison mondiales de smartphones

¹ **Gartner Inc.** fondée en 1979, est une entreprise américaine de conseil et de recherche dans le domaine des techniques avancées dont le siège social est situé à Stamford, Connecticut¹

Le Jeu dans le mobile est toujours la première industrie culturelle mondiale, il génère 12 milliards de dollars dans le monde en 2013. De plus ont compte 500 millions de joueurs sur téléphones/tablette.

*Angry Birds plus de 1 milliards de téléchargement sur mobiles.

V. Conclusion :

L'application mobile réalisée dans une technologie n'est pas nécessairement compatible avec toutes les plateformes (c'est d'ailleurs impossible).

L'application est réalisée, aussi, à partir de différents langages de programmation car il existe des environnements très hétérogènes au niveau des plateformes (les OS mobiles) installées sur les téléphones portables.

I. Introduction :

Il existe plusieurs types de système d'exploitation pour le téléphone mobile, chaque système a ses avantages et ses limites. Il est donc très difficile de choisir la plateforme répondant bien à l'objectif d'une société ou d'une personne.

Dans ce chapitre nous allons présenter la raison du choix d'Android, nous aurons aussi à présenter ce système et les outils pour créer une application dans la pratique.

II. Pourquoi avons-nous choisie Android ?

Android est un système d'exploitation puissant et moderne, qui se caractérise par la simplicité et la flexibilité; cela signifie que le système est développé avec un simple langage java, et il s'adapte à beaucoup de structures différentes.

De plus, Android est open source; donc il offre aux développeurs la possibilité d'améliorer les applications.

Le noyau Linux lui fourni une grande mémoire, la gestion de processus, le modèle de sécurité, le soutien des bibliothèques partagé...etc.

Le SDK de l'Android offre complètement les APIs, avec un accès facile pour développer l'application.

- Une API, ou « interface de programmation » en français, est un ensemble de règles à suivre pour pouvoir dialoguer avec d'autres applications. Dans le cas de Google API, il permet en particulier de communiquer avec Google Maps. [6]

III. Android et Google :

L'histoire a commencé en Octobre 2003 dans l'État de Californie. Android est à l'origine d'une start-up fondée par Andrew Rubin, Chris White, Nick Sears et Rich Miner puis racheté par Google en 2005. Le nom Android venant lui-même d'« androïde » qui désigne un robot construit à l'image d'un être humain. [7]

Au début de Novembre 2007, Google a créé l'Open Handset Alliance OHA qui regroupait 35 entreprises. Cette alliance annonce la création de son propre système mobile open source, son objectif est de développer des normes ouvertes pour les appareils mobiles.

La première version du SDK Android 1.0 sort en 2008 avec le premier téléphone sous Android HTC Dream et la dernière version kitkat 4.4.3 en 2014.

En outre, il est prévu que le système d'exploitation Android peut être utilisé sur les téléviseurs et les ordinateurs personnels très bientôt. [8]

IV. Les différentes versions :

Android a évolué très rapidement. Voici quelques versions d'Android qui portent le nom gâteaux (américain) :

Android 1.0 (23 septembre 2008).

Android 1.1 (9 février 2009)

Android 1.5 (Cupcake) (30 avril 2009).

Android 1.6 (Donut) (15 septembre 2009).

Android 2.0 (Eclair) (26 octobre 2009).

Android 2.0.1 (Eclair) (3 décembre 2009).

Android 2.1 (Eclair) (12 janvier 2010).

Android 2.2 (Froyo) (20 mai 2010).

Android 2.3 (Gingerbread) (6 décembre 2010).

Android 3.0 (Honeycomb) (26 janvier 2011).

Android 4.0 (Ice Cream Sandwich) (19 octobre 2011).

Android 4.1 (Jelly Bean) (27 juin 2012).

Android 4.3 (Jelly Bean) (24 juil. 2013).

Android 4.4 (KitKat) (31 octobre 2013).

Android 4.4.3 (KitKat) (14 avril 2014).

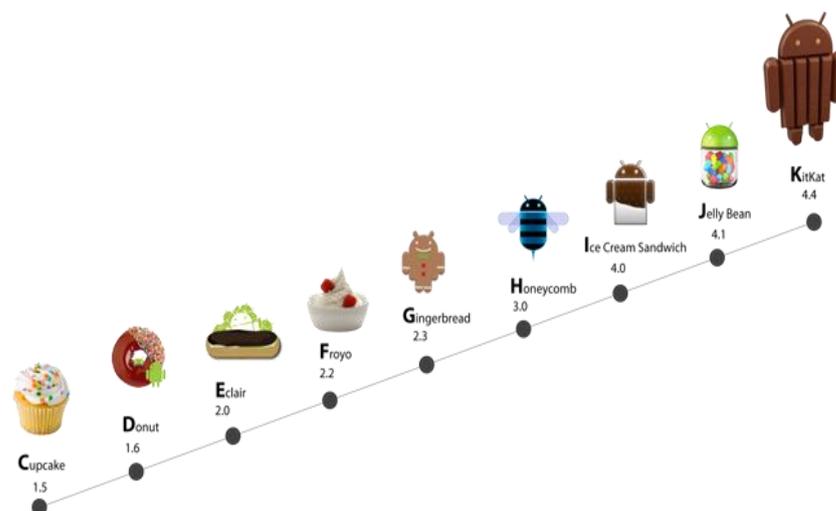


Figure II.1 Les versions d'Android.

V. Avantages et inconvénients d'Android :

1. Les Avantages:

- OS Kernel Robuste.
- Bibliothèque innovante.
- Facilités de développement.
- Exécution rapide.

2. Inconvénients :

Parmi les inconvénients majeurs chez Android :

- Les considérations des performances qui nécessitent des périphériques assez puissants et rapides.
- La difficulté d'intégration pour les vendeurs et sa forte dépendance de Google.

VI. Architecture du système Android :

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

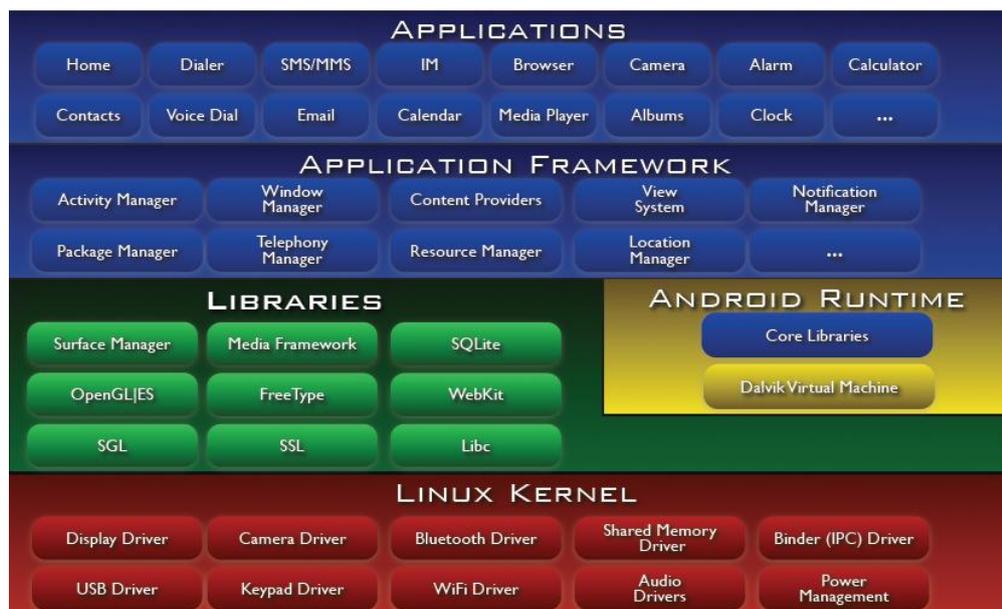


Figure II.2 Les composants principaux de système d'exploitation Android

- Android est basé sur un kernel linux 2.6. Alors, il y a plusieurs avantages comme la grande mémoire, la gestion du processus, le modèle de sécurité et le soutien de bibliothèque partagé.
- les librairies C/C++ utilisées par un certain nombre de composants du système Android.

- l'Android Runtime : Cette couche contient les bibliothèques cœurs du Framework ainsi que la machine virtuelle exécutant les applications.
- le Framework permettant au développeur de créer des applications.
- les applications : Ce sont les applications qui marchent sous la plateforme Android comme le réveil, la calculatrice, le calendrier, la caméra, les contacts, etc. Toutes les applications sont développées par Java.

VII. Les outils et données auxiliaires :

Pour développer une application sur Android, nous utiliserons :

1. L'environnement Eclipse :



Eclipse est l'Environnement de Développement Intégré (ou IDE), il trouve son origine au sein de la société IBM, qui a décidé en 2001 de mettre à disposition de la communauté Open Source l'ébauche d'une plate-forme de développement ouverte, entièrement écrite en Java, capable d'intégrer des extensions adaptées à diverses activités (débugage, modélisation, interfaces graphiques...).

2. JDK5 ou JDK6 :

Java Development Kit bibliothèques logicielles de base pour langage Java plus les outils pour compiler et déboguer le code.

3. SDK :

Le SDK est l'outil le plus important, un kit de développement est une collection d'APIs facile à installer qui fournit les procédures disponibles et les fonctions pour faciliter le processus du développement d'une application sur Android.

4. Android Development Tools ADT :

Permet la conception graphique d'interfaces utilisateur, debug distant sur un téléphone, gestion de l'architecture de fichiers d'une application, il permet d'exporter les projets au format APK.

5. Android Virtual Device Manager AVD:

Permet de créer des configurations de l'émulateur, configure la partie graphique mode tablette/téléphone, taille de l'écran, etc...

Configure la partie matérielle (émulé) accessible, configure l'image de l'émulateur à utiliser.

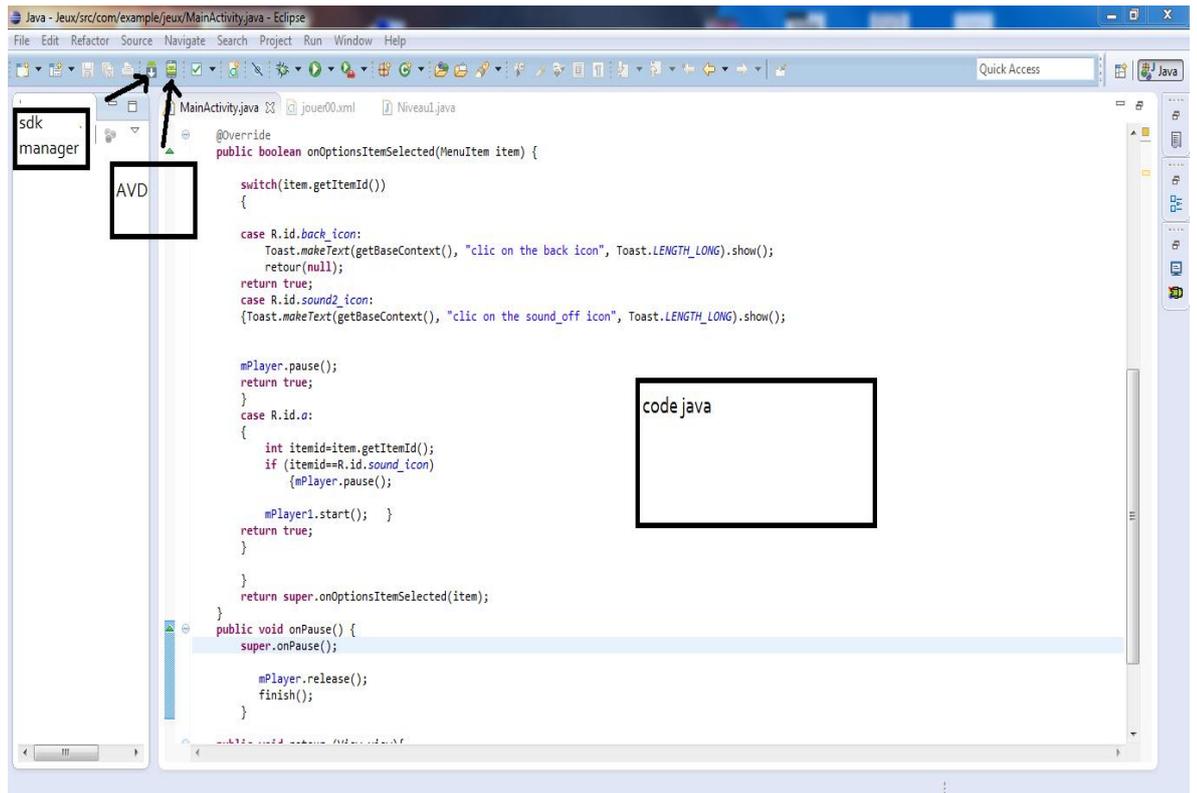


Figure II.3 Environnement Eclipse, ADV et SDK

6. L'émulateur :

L'émulateur est un appareil mobile virtuel qui fonctionne sur l'ordinateur, il permet aux développeurs de concevoir, tester et évaluer les applications Android sans l'aide d'un appareil physique.

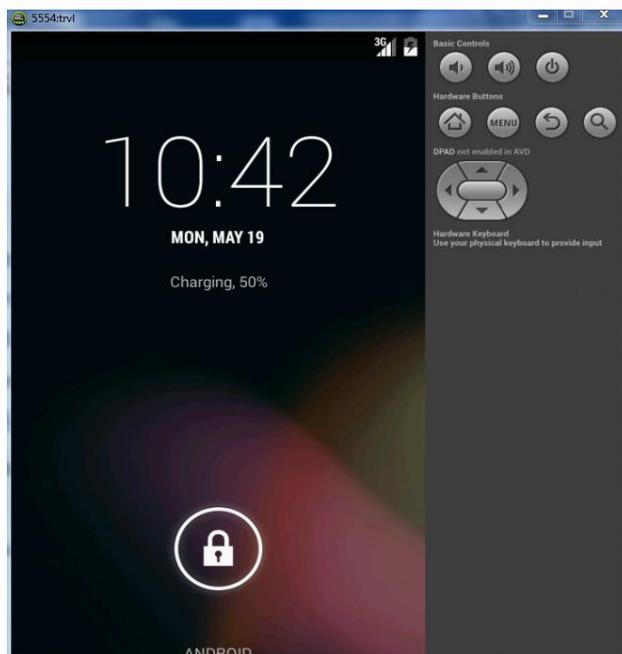


Figure II.4 Emulateur Android

VIII. Composantes d'une application Android : [9]

1. Les Activités :

Une activité est une composante principale de l'application Android. Elle représente l'implémentation et les interactions des interfaces.

Plusieurs choix se proposent pour mettre en place l'interface visuelle : utiliser un fichier XML pour manipuler l'interface, créer les éléments de l'interface à l'intérieur du code.

2. Les Services

Un service est une sorte d'activité, mais ne possédant pas d'interface visuelle. Celle-ci est donc lancée en fond, et permet par exemple de jouer de la musique d'effectuer une vérification des mails toutes les x minutes...

3. Les fournisseurs de contenus « ContentProvider » :

Comme l'indique le nom, un ContentProvider permet de partager les données d'une application peuvent être stockées dans une base SQLITE, dans des fichiers ou sur web.

Le but étant de permettre à d'autres applications de requêter ces données.

4. Les Broadcast receivers :

Un broadcast receiver est une composante qui réagit à un évènement système (par exemple batterie faible, Le téléphone reçoit un SMS...).

Les broadcast receivers ne possèdent pas d'interface utilisateur et doivent effectuer une tâche légère.

5. Les Intents :

Les composantes Android communiquent via des messages système que l'on appelle intent. Ils sont émis par le terminal pour prévenir les différentes applications du déclenchement d'évènements.

IX. Développement d'une application sur Android :

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

- Création du projet.
- Dessiner des interfaces en fichiers XML ou en codage : les vues (View) : Text, Edit, List, Image, Web, Map, etc.
- 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 (code

java). Dans cette technique, on programme directement les composants graphiques en utilisant le codage.

- Organiser des ressources: les constantes globales (string.xml), les icônes, les images, etc.
- Création des activités:
 1. chaque activité peut correspondre à un écran ou une fonction de cette application.
 2. Les Intents sont utilisées pour orienter des activités (CALL, ACTION_MAIN, ACTION_VIEW, etc).
- Créer et mettre à jour le fichier de configuration : 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.

X. Cycle de vie d'une activité (Activity Lifecycle):

Pour développer d'une application sur Android, on doit comprendre le cycle de vie d'une activité.

- L'état **Active/courant** (Running) : l'activité se trouve au premier plan, et reçoit les interactions utilisateurs. Si l'appareil a besoin des ressources, l'activité se trouvant en bas de la back est tuée.
- L'état **Paused** (il est en pause) :L'activité est visible mais l'utilisateur ne peut pas interagir avec. La seule différence avec l'état précédent est la non-réception des événements utilisateurs.
- L'état **Stopped** : L'activité n'est pas visible mais toujours en cours d'exécution. Toutes les informations relatives à son exécution sont conservées en mémoire.
- L'état **Dead** : l'activité est tuée, elle n'est plus en cours d'exécution, et disparaît de la back stack.
- **Back stack** : toutes les activités sont stockées dans une liste que l'on appelle généralement back stack.

Il existe trois boucles principales:

Lors de la création d'activité, la méthode **onCreate** est appelée .Cet appel est suivi par la méthode **onStart** afin de signifier le lancement effective de l'application. Puis la méthode **onResume** est appelée afin d'exécuter tous les traitements nécessaires au fonctionnement de l'activité. Ces traitement devront

I. Introduction :

Ce chapitre est basé sur la réalisation et la mise en œuvre de notre application. Nous allons utiliser l'environnement Eclipse en langage Java, et le langage XML pour implémenter la partie graphique. Enfin nous mettrons en évidence les principales interfaces et activités de notre application.

II. Pour quoi le langage java ?

Java, langage orienté objet, développé par SUN(1991). Le but de Java à l'époque était de constituer un langage de programmation pouvant s'intégrer dans les appareils électroménagers afin de pouvoir les contrôler, de les rendre interactifs, et surtout de permettre une communication entre les appareils. [10]

Dans le domaine mobile il y a plusieurs systèmes d'exploitation utilisés dans le téléphone portable, afin d'adopter notre application aux différents systèmes mobiles on a choisi le langage java, car ce dernier a été utilisé dans la majorité des systèmes mobiles et adapter à n'importe quelle système d'exploitation.

La particularité principale de Java est que les logiciels écrits dans ce langage sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'UNIX, Windows, Mac OS ou GNU/Linux, avec peu ou pas de modifications. C'est la plate-forme qui garantit la portabilité des applications développées en Java.

III. Langage XML :

L'Extensible Markup Language (XML « langage de balisage extensible » en français) est un langage informatique de balisage générique qui dérive du SGML. Cette syntaxe est dite « extensible » car elle permet de définir différents espaces de noms, c'est-à-dire des langages avec chacun leur vocabulaire et leur grammaire, comme XHTML, XSLT, RSS, SVG... Elle est reconnaissable par son usage des chevrons (<>) encadrant les balises. L'objectif initial est de faciliter l'échange automatisé de contenus complexes (arbres, texte riche...) entre systèmes d'informations hétérogènes (interopérabilité). [11]

IV. Accès et gestion des bases de données locales SQLite:

SQLite est une base de données très appréciée car elle fournit une interface SQL, tout en offrant une empreinte mémoire très réduite et une rapidité de traitement satisfaisante. En outre, elle appartient au domaine public et tout le monde peut donc l'utiliser. De

nombreuses sociétés (Adobe, Apple, Google, Sun, Symbian) et plusieurs projets open-source (Mozilla, PHP, Python) fournissent désormais des produits intégrant SQLite. SQLite étant intégré au moteur d'exécution d'Android, toute application peut créer des bases de données SQLite. Ce SGBD disposant d'une interface SQL, son utilisation est assez simple pour quiconque a une expérience avec d'autres SGBDR. Cependant, son API native n'est pas JDBC, qui, d'ailleurs, serait trop lourd pour les terminaux limités en mémoire comme les téléphones. Par conséquent, les programmeurs Android doivent apprendre une nouvelle API. [12]

V. Les avantages du développement du jeu :

Les développeurs de jeux ne sont plus obligés de s'adapter aux conditions posées par les opérateurs, et les utilisateurs n'ont pas besoin de se connecter aux sites internet de distributeurs à la recherche de jeux optimisés pour leur appareil particulier ou système d'exploitation.

VI. Description du jeu :

Smile ☺ est un jeu de réflexion exigeant l'intelligence, il se pratique à deux ou avec un seul joueur contre la machine.

Le jeu se déroule sur une simple grille dont la forme diffère selon le niveau.

- Le principe du jeu :

Le jeu consiste à former le plus grand nombre de carré sur la grille. Les bordures de chaque carré sont des boutons blancs et cliquables une seule fois. Lors du clic, la couleur du bouton est changée, si le carré est complet le joueur gagne un point.

VII. Réalisation:

1) Méthodologie de conception UML :

UML (Unified Modeling Language, que l'on peut traduire par "langage de modélisation unifié) est une notation permettant de modéliser un problème de façon standard. Ce langage est né de la fusion de plusieurs méthodes existant auparavant, et devenu désormais la référence en termes de modélisation objet. [13]

Parmi les diagrammes UML qu'on a utilisé lors de la conception de notre application nous avons :

- Le diagramme de cas d'utilisation

Le diagramme de cas d'utilisation nous montre l'interaction entre l'utilisateur et l'application.

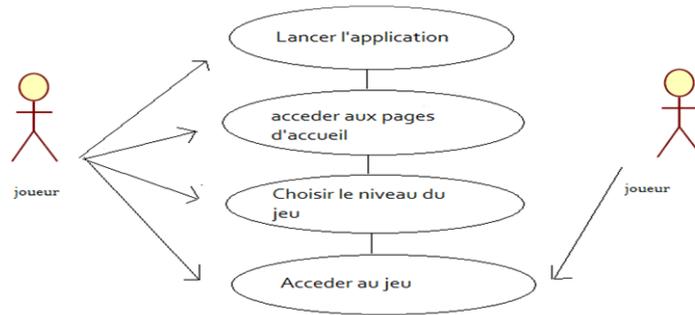


Figure III.1 digramme de cas d'utilisation.

- Le diagramme de classe

Diagramme de classe pour présenter les classes et les interfaces d'un système ainsi que les différentes relations entre celles-ci.

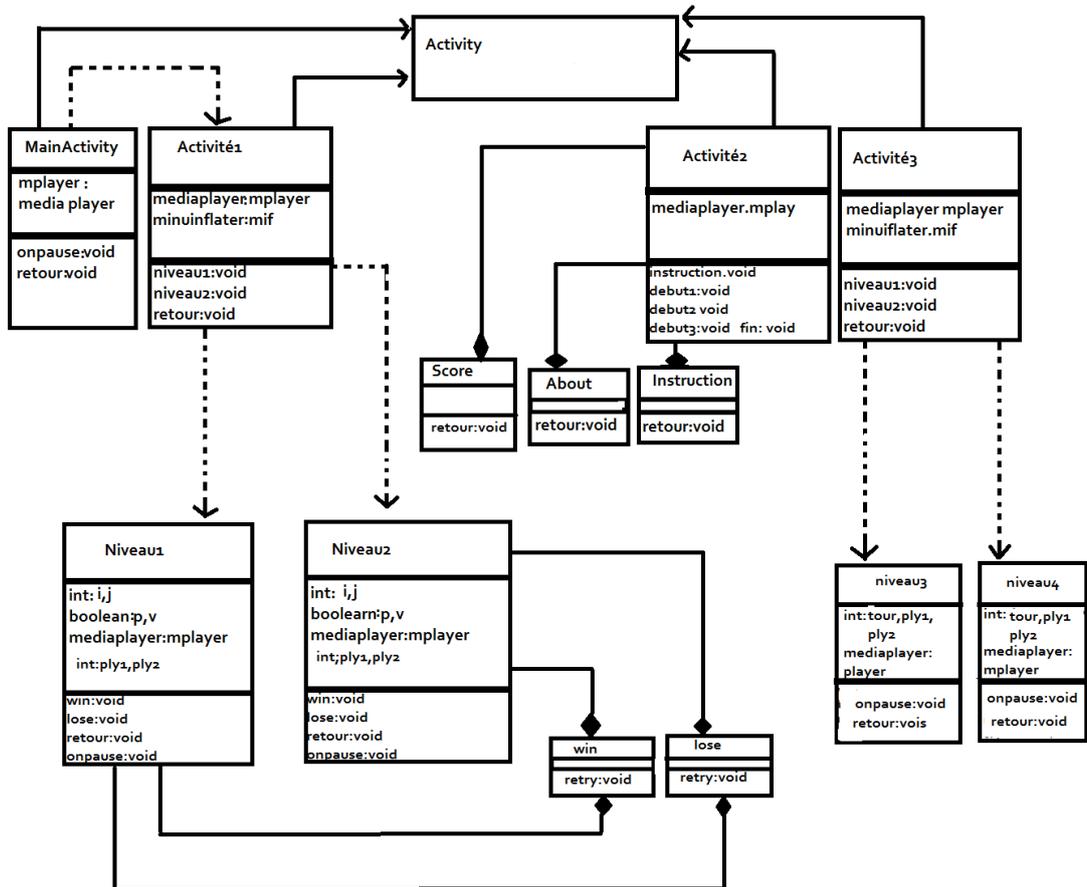


Figure III.2 Le diagramme de classe

2) Menu de l'application:

2.1 L'interface graphique :

On trouve des activités d'accueil et d'autres activités de jeu, à chaque activité on a implémenté du son, action bar.

- **Action bar** : existe depuis Android 3.0, cet élément est une structure importante d'une application Android. Elle se place en haut de l'écran de l'application et persiste quelque soit l'écran. L'implémentation d'une Action bar s'effectue dans le dossier **menu**.

L'objectif d'une action bar est de :

- rendre les actions les plus importantes d'une application facilement accessibles.
- permettre la navigation plus facilement dans l'application.

Les composantes de l'action bar :

- L'icône pour jouer le son.
- l'icône pour stopper le son.
- navigation entre activité : il est utilisé pour naviguer entre les différentes activités d'application.



Figure III.3 Action bar.

- **Le Son** : l'objectif est de pouvoir jouer un son en arrière plan (fond). Pour réaliser cette tâche il faut créer un nouveau dossier dans le répertoire **res**, nommer **raw** et copier les sons dans ce dossier.

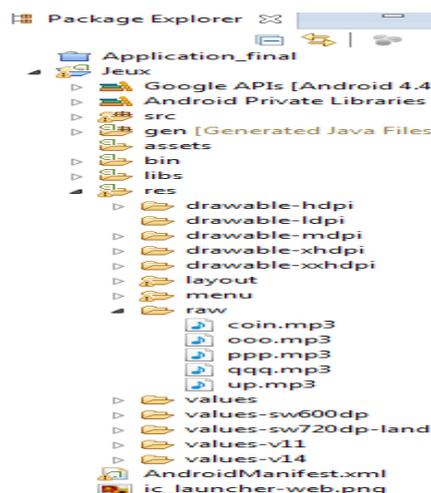


Figure III.4 Dossier raw avec des sons

2.1.1 Les activités d'accueil :

On trouve dans la première activité un bouton cliquable, sous forme d'image (background) pour atteindre la deuxième activité.

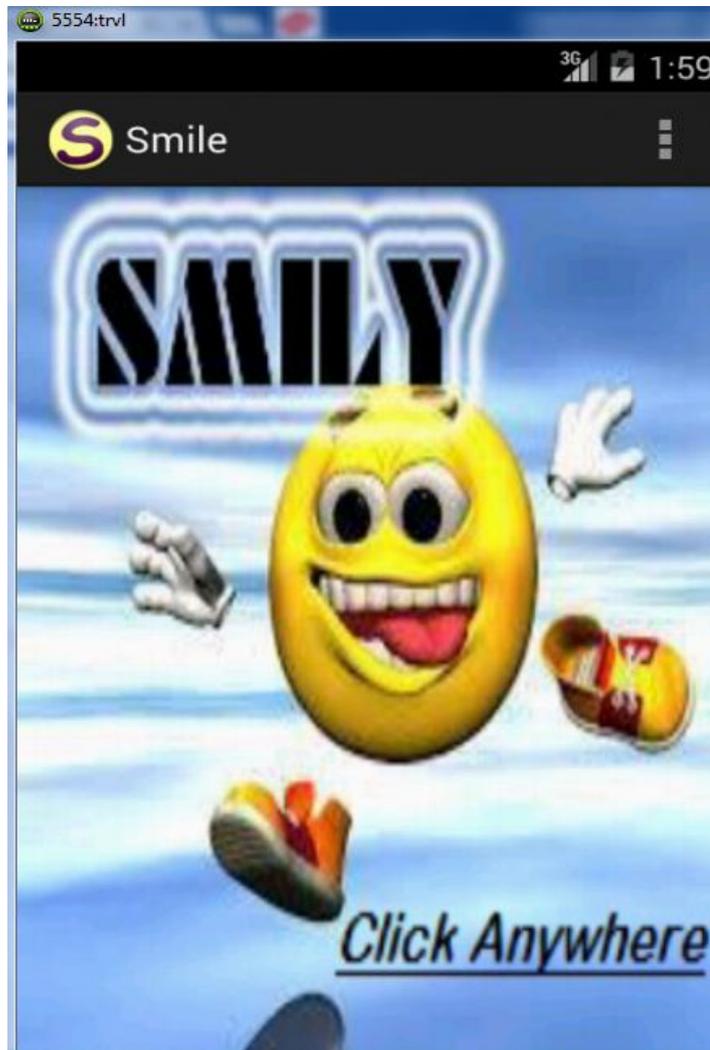


Figure III.5 Activité d'accueil

```
<Button
    android:id="@+id/button1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_alignParentLeft="true"
    android:layout_alignParentTop="true"

    android:background="@drawable/smile"
    android:onClick="start"/>
```

Figure III.6 Création d'interface graphique avec XML

La figure ci-dessus décrit comment générer l'interface graphique de cette activité.

<Button/> permet de créer un widget « bouton », qui s'agit d'un TextView cliquable. La propriété "android:id" rajoute automatiquement dans le fichier "R" l'identifiant du widget. La propriété "android:background" est utilisée pour afficher un fond. "android:onClick" précise le nom de la méthode a appelé (start), pour ouvrir une activité (deuxième activité d'accueil).

La deuxième activité d'accueil affiche 6 boutons :

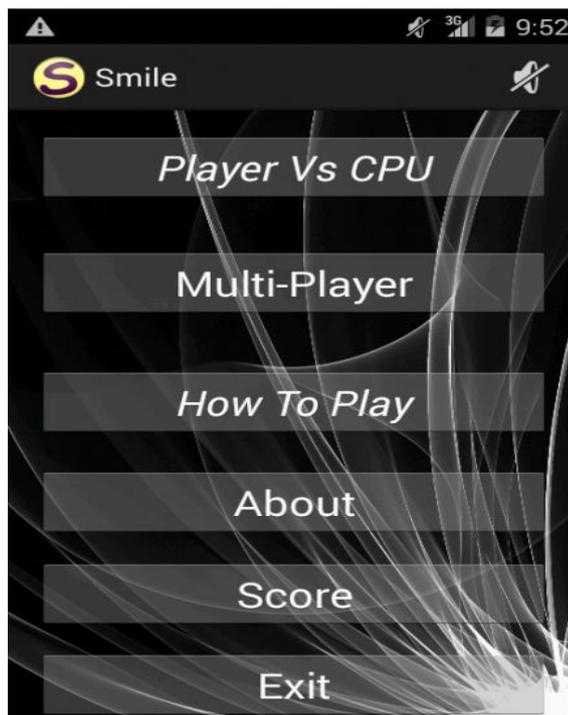


Figure III.7 La deuxième activité d'accueil

- Bouton Player vs Cpu : conduit vers l'activité de choix.
- Bouton Multi-Player : conduit vers l'activité de choix.

Cette dernière contient deux boutons du choix du niveau (selon la dimension de grille) pour accéder directement au jeu.

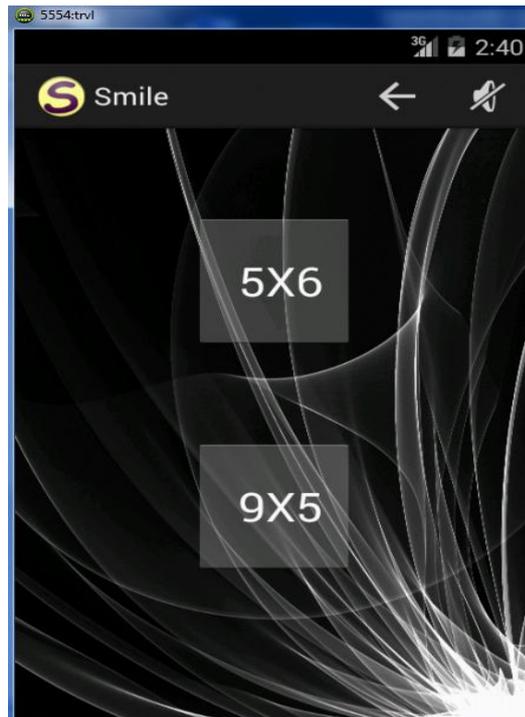


Figure III.8 La troisième activité d'accueil

- Bouton How To Play : conduit vers une activité d'instruction. Cette activité décrit les étapes du jeu.

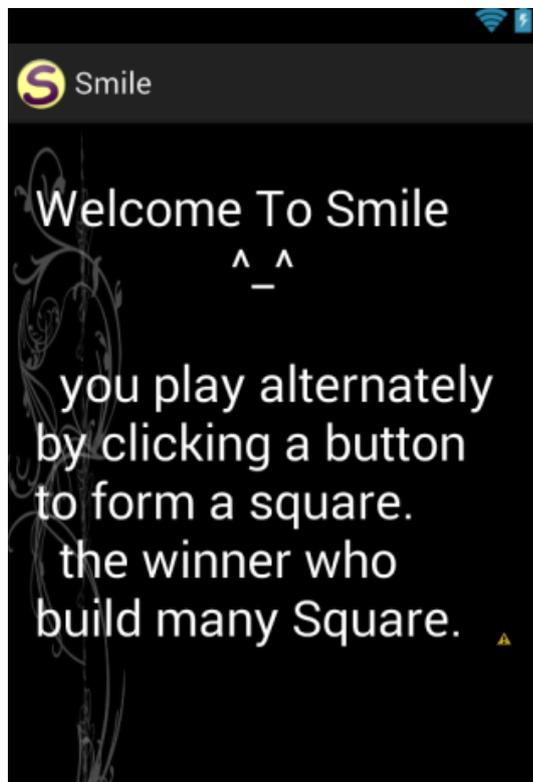


Figure III.8 Activité d'instruction

- Bouton About : conduit vers l'activité About. Cette activité décrit les développeurs du jeu.



Figure III.9 Activité A-propos

- Bouton Score : conduit vers l'activité Score. Cette interface permet l'affichage du nom et du score du joueur à l'aide d'une base de données SQLite.

Principe : pour créer et ouvrir une base de données, la meilleure solution consiste à créer une sous classe de SQLiteOpenHelper. Cette classe enveloppe tout ce qui est nécessaire à la création et à la mise à jour d'une base, selon nos spécifications et les besoins de notre application.

- onCreate() : pour passer l'objet SQLiteDatabase qui sera rempli avec les tables et les données initiales (nom et score).
- onUpgrade() : pour passer un objet SQLiteDatabase ainsi que l'ancien et le nouveau numéro de version afin de pouvoir convertir au mieux une base de données d'un schéma vers un autre.

L'écriture du nom est effectuée par le biais du joueur. Le score et le nom sont enregistrés dans la table SQLite à la fin du jeu, puis affichées dans l'interface d'activité Score.

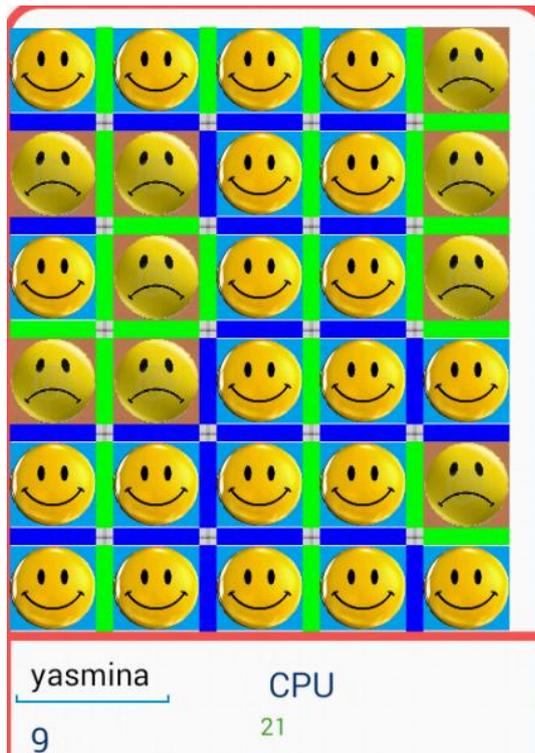


Figure III.10 L'écriture du nom de joueur, plus le score finale

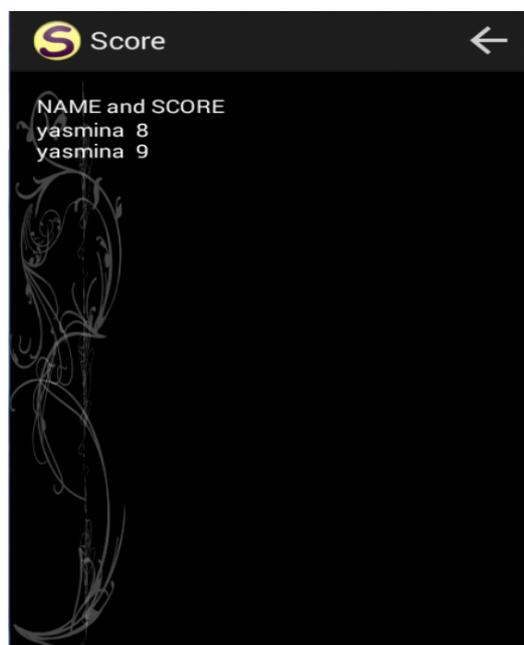


Figure III.11 L'activité score avec le nom et le score

- Bouton Exit : pour quitter le jeu.

2.1.2 Les activités du jeu :

Ces activités ressemblent à une grille. Elles contiennent des boutons cliquables, et à l'intérieure de chaque carré il existe des vues (imageView) vides. A la fin, l'interface comporte des widgets (text view, EditText) pour afficher le score.

Activité 1 : une grille de 5*6 :

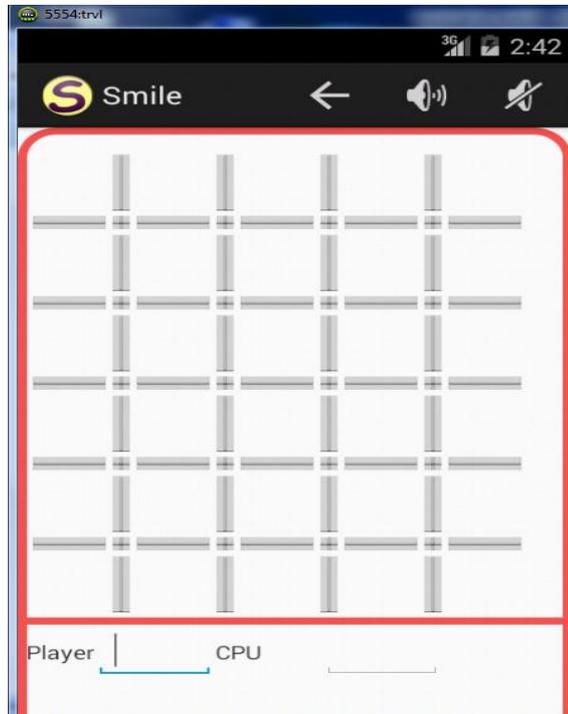


Figure III.12 Grille de jeu 5*6

Activité 2 : une grille de 5x9 illustrés comme suite :

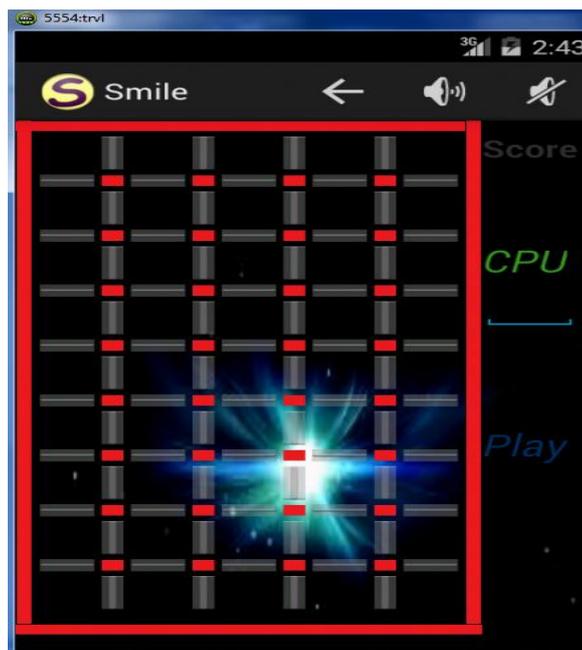


Figure III.13 grille de jeu 5*9

2.2 Réalisation :

L'interface est réalisée via l'écriture d'un code java de l'application, cette dernière contient 11 classes. Chacune a une relation avec les activités précédentes par la composante intention (intent).

2.2.1 Les étapes de la réalisation :

Le déroulement du jeu est basé sur 3 étapes :

A. **Changement de couleur :** En cliquant sur un bouton sa couleur change.

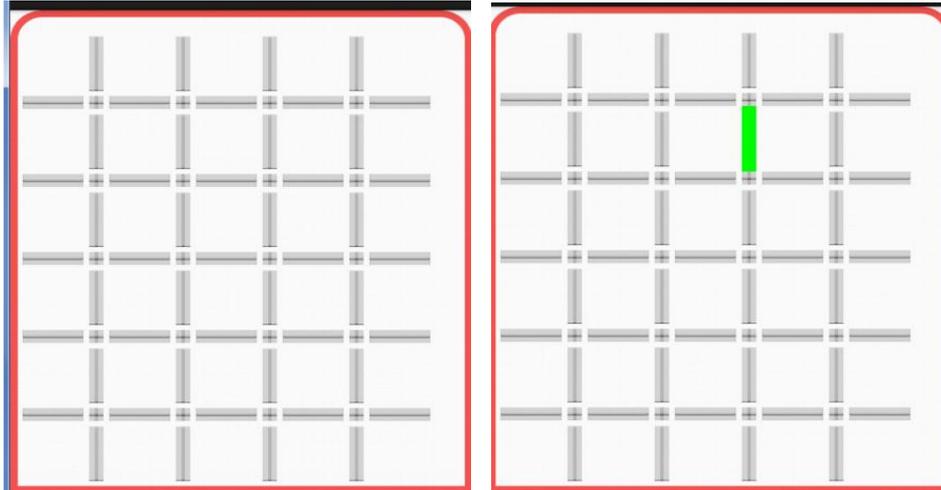


Figure III.14 Première étape changement de couleur

B. **L'affichage des images :** Si le bouton jouer complète un carré (ou deux), une image est apparu dans la surface de ce carré.

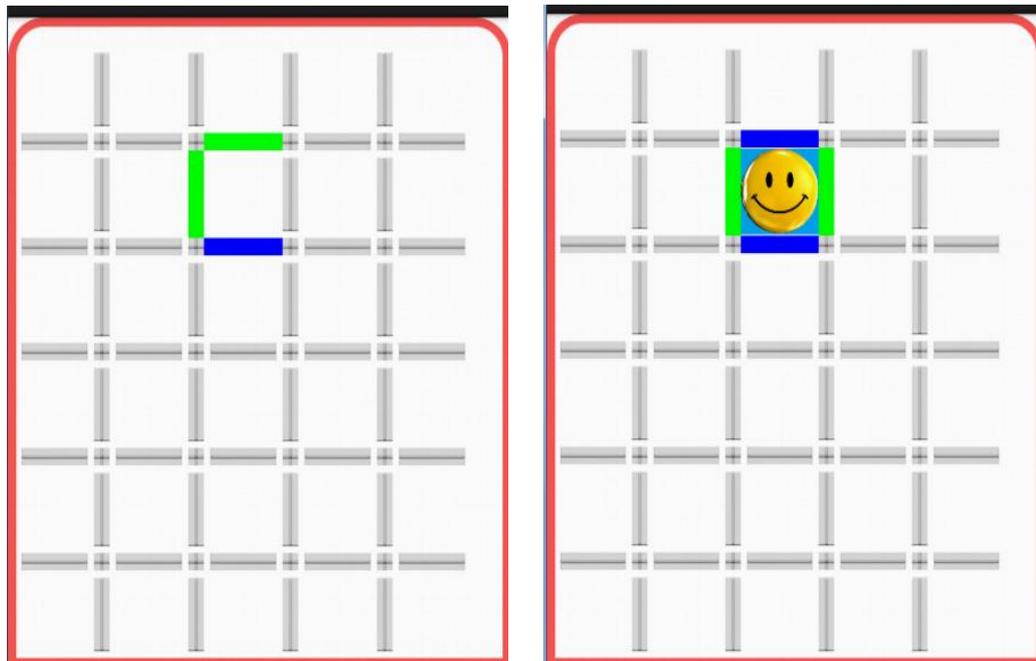


Figure III.15 Deuxième étape affichage d'image

C. **Changement du score** : Après l'affichage d'image, un des 2 compteurs sera incrémenté puis affiché.



Figure III.16 Troisième étape incrémentation de score

2.2.2 Réalisation des versions du jeu :

Nous avons deux versions; joueurs vs joueurs et joueurs vs machine implémenté comme suit :

2.2.2.1 Joueur vs joueur :

La classe java de ce modèle comporte une seule méthode (Test) qui gère le déroulement du jeu en basant sur les étapes précédentes :

- Le changement de couleur s'effectue lors du clic, selon le tour du joueur (soit vert ou bleu).
- L'image affichée se diffère selon le joueur.
- L'incrémentation du score s'effectue lors de l'affichage de l'image.

```

public void Test(View view ){
    switch (view.getId()) {
        case (R.id.btnver1):if (btnver[1]==false){
            btnver[1]=true;
            Button monButton=(Button)findViewById(R.id.btnver1);
            if (tour==1){monButton.setBackgroundColor(Color.GREEN);}
            else {monButton.setBackgroundColor(Color.BLUE);}

            if (btnhor[1]==true){
                if (tour==1) {ply1++;}
                else{ply2++;}
                ImageView image=(ImageView) findViewById(R.id.textView1);
                if(tour==1) image.setImageResource(R.drawable.o);
                else image.setImageResource(R.drawable.x);
            }
            if (btnhor[2]==true && btnver[2]==true){
                if (tour==1) {ply1++;}
                else{ply2++;}
                ImageView image=(ImageView) findViewById(R.id.textView2);
                if(tour==1) image.setImageResource(R.drawable.o);
                else image.setImageResource(R.drawable.x);
            }
            if (tour==1) tour=2;
            else tour=1;
        }
        break;
    }
}

```

Figure III.17 La méthode test.

2.2.2.2 Joueur vs Système :

Ce modèle consiste à remplacer le rôle du système ; cela signifie que tous les événements se déroulent automatiquement d'une manière intelligente comme suit :

La classe java est composée de 4 méthodes principales :

- La méthode Test : gère le jeu (pour le joueur) par les étapes décrites précédemment.
 - la méthode tester : joue le rôle du système, pour cela il faut suivre les règles des priorités suivantes :
- 1) un bouton non cliqué qui peut compléter deux carrés en même temps est prioritaire, sa couleur change automatiquement (sans clic), deux images sont affichés en même temps, et le score est incrémenté (2 fois) puis faire appel à la méthode Test, sinon passer à 2.

```

public void tester() {
    boolean P=false;
    if (btnver[6]==false && P==false){
        if (btnver[5]==true && btnhor[2]==true && btnhor[7]==true && btnhor[3]==true && btnhor[8]==true && btnver[7]==true ){
            btnver[6]=true;

            Button monButton=(Button)findViewById(R.id.btnver6);
            monButton.setBackgroundColor(Color.BLUE);
            ply2++;
            ImageView image=(ImageView) findViewById(R.id.textView7);
            image.setImageResource(R.drawable.x);
            ply2++;
            ImageView image1=(ImageView) findViewById(R.id.textView8);
            image1.setImageResource(R.drawable.x);
            P=true;
        }
    }
}

```

Figure III.18 Méthode tester, le code source du première cas.

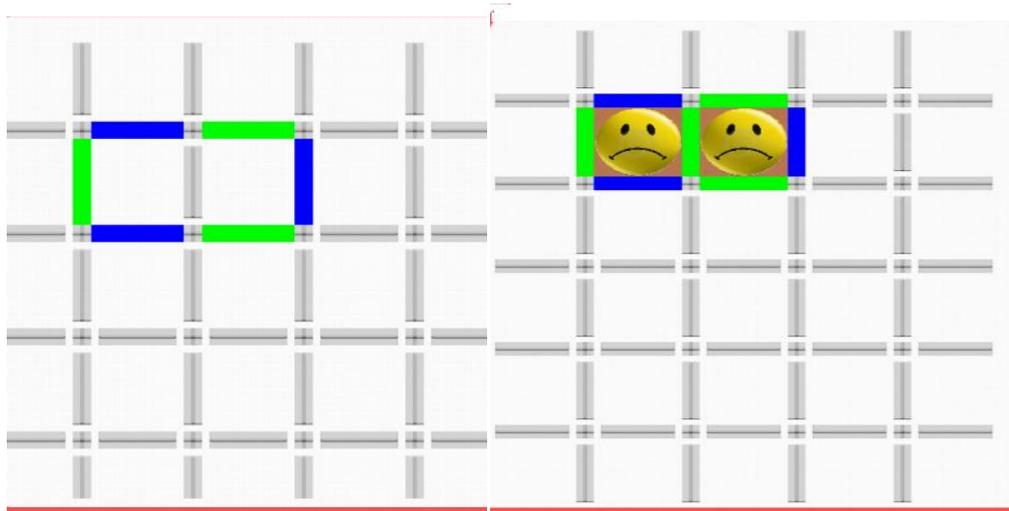


Figure III.19 Resultat

- 2) Si on a la possibilité de compléter quelques carrés, on parcourt la grille en comptant le nombre des bordures blanches pour chaque bouton non cliqué .la priorité est donné au bouton qui a le plus grand nombre, sa couleur est changé automatiquement et l'image s'affiche à l'intérieure du carré, le score de la machine est incrémenté puis faire appel à la méthode Test, sinon passer a 3.

```

//
if (P==false){
if(btnver[1]==false){
int m=0;
if (btnhor[1]==true){
if (btnver[2]==false) m++;
if (btnhor[2]==false) m++;
tabv2[i]=(ImageView)findViewById(R.id.textView1);
}
if (btnhor[2]==true && btnver[2]==true){
if (btnhor[1]==false) m++;
tabv2[i]=(ImageView)findViewById(R.id.textView2);
}

tabv1[i]=(Button)findViewById(R.id.btnver1);
tabv0[i]=m;}
i++;

```

Figure III.20 le code source du deuxième cas

Explication du code : ce code décrit le calcul du nombre des bordures du bouton (dans cet exemple le bouton verticale 1), le nombre est stocké dans le premier tableau (il représente la priorité), l'image est déposé dans le deuxième tableau, le bouton dans un troisième.

```

while (i<=41 && m<3){
    i++;
    if (tabv0[i]>=m){m=tabv0[i];x=i;}
}
while (j<=37 && m<3){
    j++;
    if (tabh0[j]>=m){m=tabh0[j];y=j;}
}
if (m!=0) {
    if (y==0){
        btnver[x]=true;
        tabv1[x].setBackgroundColor(Color.BLUE);
        tabv2[x].setImageResource(R.drawable.x);
        ply2++;
        P=true;
    }
    else {
        btnhor[y]=true;
        tabh1[y].setBackgroundColor(Color.BLUE);
        tabh2[y].setImageResource(R.drawable.x);
        ply2++;
        P=true;
    }
}
}

```

Figure III.21 le code source du deuxième cas, test finale

Ce code permet de comparer la priorité de tous les boutons, en parcourant le tableau de priorité (premier tableau).

Faire un changement de couleur pour le bouton prioritaire, et un affichage d'image en utilisant le deuxième et troisième tableau.

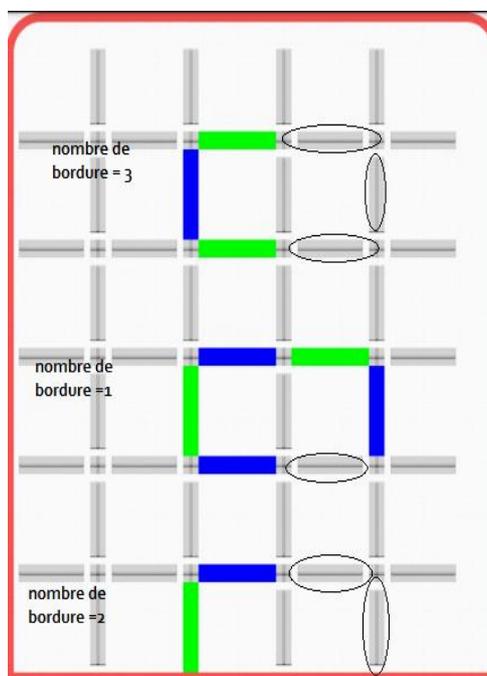


Figure III.22 Le nombre des bordures

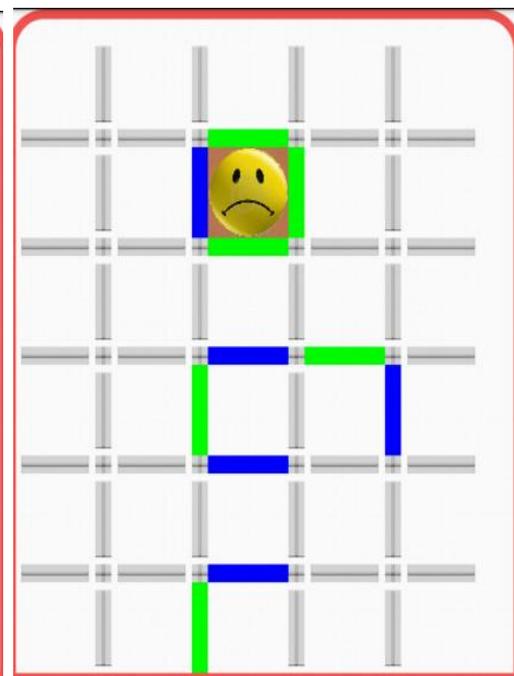


Figure III.23 Changement de couleur et affichage d'image

- 1) Au début du jeu, nous n'avons aucune possibilité de compléter des carrés, on applique le même principe en comptant le nombre des bordures blanches pour chaque bouton non cliqué. La priorité est donnée pour le bouton qui a le plus grand nombre, sa couleur est changée automatiquement, puis faire appel à la méthode Test.

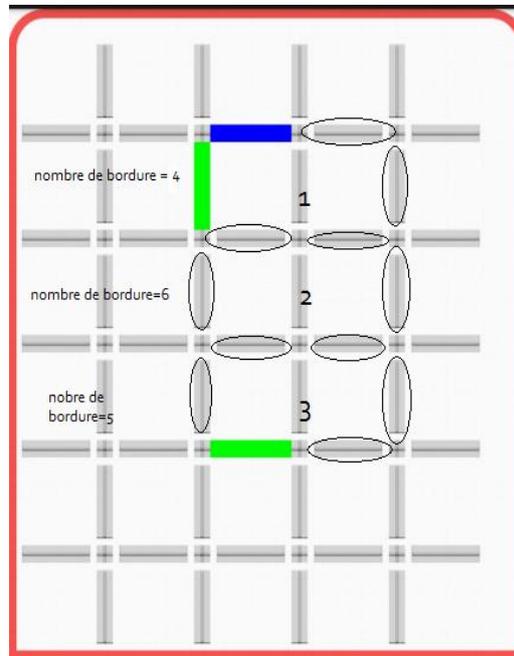


Figure III.24 Le bouton 2 est le prioritaire

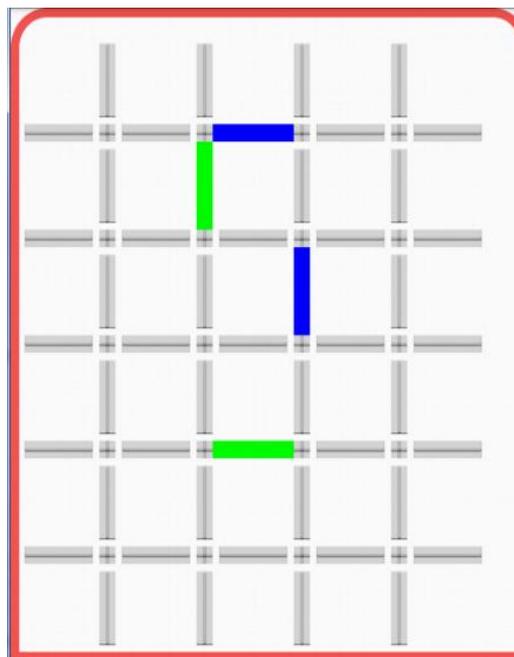


Figure III.25 Le résultat

- la méthode win : pour accéder automatiquement à une activité dans le cas où le joueur gagne en comparant le score. On utilise la composante intent.



Figure III.26 L'interface d'activité win

- la méthode lose : pour accéder automatiquement à une activité dans le cas où le joueur perd en utilisant la composante intent.

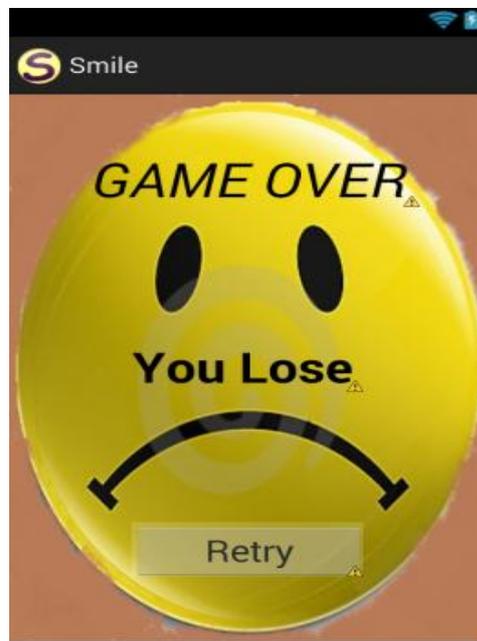


Figure III.27 L'interface d'activité lose

```

boolean v=false;
for(int i=1;i<25;i++){
    if (btnver[i]==false ) v=true; }
for(int i=1;i<26;i++){
    if (btnhor[i]==false ) v=true; }
    if(v==false){if (ply1<ply2) lose();
        else if (ply1>ply2) win();}
}

public void win(){
    Intent ouvrirf=new Intent(Niveau1.this,WinActivity.class);

    startActivity(ouvrirf);finish();
}
public void lose(){
    Intent ouvrirf=new Intent(Niveau1.this,LouseActivity.class);

    startActivity(ouvrirf);finish();
}
}

```

Figure III.28 Code source de la méthode win et lose

Explication de code : vérifier si tous les boutons sont cliqués, puis tester si le score du joueur est supérieur par rapport au score de la machine alors faire un appel à la méthode win sinon un appel à la méthode lose.

VIII. Contraintes de développement :

- manque des connaissances sur la programmation mobile.
- Temps de réalisation de PFE.

IX. Conclusion :

Dans ce chapitre nous avons proposé un exemple d'application pour illustré notre travail. Nous avons réalisé un petit jeu réflexif puis présenté les différentes interfaces d'application.

Conclusion générale :

Ce travail nous a permis d'avoir une idée plus claire et plus approfondie des applications mobile. Nous avons étudié d'une façon générale les systèmes d'exploitation pour les mobiles, qui se différencient selon le langage de développement, l'interface, l'existence de portabilité.

Puis, nous avons étudié dans le détail le système d'exploitation Android, qui domine le marché des smartphones grâce à sa puissance, la simplicité de développement et l'ouverture du code. De plus nous avons présenté l'environnement de travail et les outils de développement de ce système.

Le dernier volet de notre travail est de développer une application « jeu Smile » sous Android, elle est implémentée en utilisant langage java sous l'environnement Eclipse.

Ce projet a fait l'objet d'une expérience intéressante, qui nous a permis d'apprendre et d'améliorer nos connaissances et nos compétences dans le domaine de la programmation.

Cependant l'amélioration de notre application reste envisageable telle que l'installation d'un wifi direct qui permettra une interconnectivité entre les différents joueurs ainsi qu'un perfectionnement de la résolution de l'écran pour plus de compatibilité avec les divers supports existants.

Références:

- [1]: E.Benedict, "Mobile is eating the world, présentation au Book Expo America", mai 2013.
- [2]: P.Froicois, "Etude sur les besoins de compétence dans le développement d'applications mobiles", Octobre 2013.
- [3] :http://www.maison dugsm.com/a/encyclopedie/definition/4/systeme_d_exploitation.html.
- [4] :[http://fr.wikipedia.org/wiki/IOS_\(Apple\)](http://fr.wikipedia.org/wiki/IOS_(Apple)).
- [5] :<http://www.zdnet.fr/actualites/chiffres-cles-les-os-pour-smartphones-39790245.htm>
- [6] : E .Frediric, "Créer des applications pour Android",2012.
- [7] :<http://fr.wikipedia.org/wiki/Android>.
- [8] :<http://www.planeteinformatics.com/android/lhistoire-du-systeme-dexploitation-mobile-android/>.
- [9] :N.BENBOURAHLA,"Android 4 les fondamentaux fe développement d'application java",2012 .
- [10] :<http://abaqueg.free.fr/JAVA.html>.
- [11] : <http://fr.wikipedia.org/wiki/XML>.
- [12] : M. Mark, "L'art de développement Android" , 2008.
- [13] : M.MOSTEFA "PLACEMENT DES TACHES REPETITIVES SUR UNE ARCHITECTURE REGULIERE EMBARQUEE", thèse d'ingénieria, 2009.

Liste des figures

Figure I.1 Part du marché des smartphone par systèmes d'exploitation.....	8
Figure I.2 livraison mondiales de smartphones.....	8
Figure II.1 Les versions d'Android.....	11
Figure II.2 Les composants principaux de système d'exploitation Android.	12
Figure II.3 Environnement Eclipse, ADV et SDK.....	14
Figure II.4 Emulateur Android.....	14
Figure II.5 Le cycle de vie d'une activité.	17
Figure III.1 Le digramme de cas d'utilisation.	24
Figure III.2 Le diagramme de classe.	24
Figure III.3 Action bar.....	21
Figure III.4 Dossier raw avec des sons.	21
Figure III.5 Activité d'accueil.	22
Figure III.6 Création d'interface graphique avec XML.....	22
Figure III.7 La deuxième activité d'accueil.....	23
Figure III.8 Activité d'instruction.....	24
Figure III.8 La troisième activité d'accueil.	24
Figure III.9 Activité A-propos.	25
Figure III.10 L'écriture du nom de joueur plus le score finale.....	26
Figure III.11 L'activité score avec le nom et le score.	26
Figure III.12 Grille de jeu 5*6.....	27
Figure III.13 grille de jeu 5*9.....	27
Figure III.14 Première étape changement de couleur.....	28
Figure III.15 Deuxième étape affichage d'image.....	28
Figure III.16 Troisième étape incrémentation de score.	29
Figure III.17 La méthode test	30
Figure III.18 Méthode tester, le code source du première cas.....	30
Figure III.19 Resultat.....	31
Figure III.20 le code source du deuxième cas.....	31
Figure III.21 le code source du deuxième cas, test finale	Erreur ! Signet non défini.
Figure III.22 Le nombre des bordures	32
Figure III.23 Changement de couleur et affichage d'image	32

Figure III.24 Le bouton 2 est le prioritaire	33
Figure III.25 Le résultat.	33
Figure III.26 L'interface d'activité win.	34
Figure III.27 L'interface d'activité lose.....	34
Figure III.28 Code source de la méthode win et lose	35

Liste des abreviation :

SE	S ystème d' E xploitation.
OS	O perating S ystem.
IOS	I phone O perating S ystem.
RMI	R earch I n M otion limited .
OHA	O pen H andset A lliance.
IDE	E nvironnement de D éveloppement I ntégré, I ntegrated D evelopment E nvironment.
API	A pplication P rogramming I nterface.
SDK	S oftware D evelopment K it .
JDK	J ava D evelopment K it.
ADV	A ndroid V irtual D evice .
ADT	A ndroid D evelopment T ool.
XML	E xtensible M arkup L anguage.
UML	U nified M odeling L anguage

Résumé :

La disponibilité croissante des mobiles présente un avantage pour les usagers comme pour les développeurs des applications, le développement des applications mobiles est se positionne actuellement comme une activité majeure du génie logiciel.

Le secret des appareils mobiles repose sur les systèmes d'exploitation qu'ils utilisent, ils fournissent des caractéristiques favorables à un développement réussi des jeux sur mobiles.

Notre objectif est de développer un jeu installé sur Android en utilisant l'environnement Eclipse et Java comme langage.

Abstract:

The increasing availability of mobile has an advantage for users as for developers of applications; the development of mobile applications is currently positioned as a major activity in software engineering.

The secret of mobile devices based on the operating systems, they provide support for a successful development of mobile games features.

Our goal is to develop a game installed on Android using Eclipse and Java as a language.

ملخص :

تزايد توفر المحمول لديه ميزة للمستخدمين و لمطوري التطبيقات، يتم وضع تطوير التطبيقات النقالة حاليا كنشاط رئيسي في هندسة البرمجيات.

سر الأجهزة النقالة هو أساس أنظمة التشغيل التي يستخدمونها، فهي توفر ميزات الدعم

لنجاح تنمية الألعاب المحمول. هدفنا هو تطوير لعبة مثبتة على Android باستخدام

Eclipse وJava كلغة.

