



IUT ROBERT
SCHUMAN

OBSERVATOIRE
ASTRONOMIQUE DE
STRASBOURG

Rapport de stage

Application Android communicant avec Aladin sur le
principe d'une télécommande intelligente

Auteur :

Valentin EHKIRCH

Année :

2013 – 2014



IUT ROBERT SCHUMAN

Département Informatique

72 route du Rhin

67400 Illkirch-Graffenstaden



OBSERVATOIRE
ASTRONOMIQUE DE
STRASBOURG

11 rue de l'Université

67000 Strasbourg

Rapport de stage

Application Android communicant avec Aladin sur le
principe d'une télécommande intelligente

Auteur :

Valentin EHKIRCH

Responsables :

M. Laurent MICHEL

M. André SCHAAFF

Remerciements

Tout d'abord, je tiens à remercier l'Observatoire Astronomique de Strasbourg et l'ensemble de ses membres pour leur accueil et pour m'avoir permis de travailler dans de si bonnes conditions.

Plus particulièrement, je tiens à remercier André SCHAAFF, ingénieur de recherche à l'Observatoire et tuteur de mon stage, pour m'avoir proposé ce stage. Je le remercie également pour son aide et son soutien durant l'intégralité de mon stage.

Enfin, je souhaite remercier mon second tuteur, Laurent MICHEL, qui m'a permis d'avancer de par ses conseils et son implication au sein du projet.

Table des matières

Introduction	5
1 Présentation du stage	7
1.1 L'Observatoire Astronomique de Strasbourg	7
1.1.1 Présentation générale	7
1.1.2 Les équipes des recherches	8
1.2 Le CDS	9
1.2.1 Présentation générale	9
1.2.2 Les différents services du CDS	10
1.2.3 International Virtual Observatory Alliance	11
1.3 Sujet de stage	12
1.3.1 A l'origine du projet	12
1.3.2 Énoncé détaillé du sujet	13
2 Déroulement du stage	15
2.1 Mise en place de l'environnement de travail	15
2.1.1 Les grandes lignes du développement Android	15
2.1.2 Les outils de développement	16
2.1.3 Autres ressources utilisées	17
2.2 Les débuts : première approche et résultats	18
2.2.1 Premières versions de l'application	18
2.2.2 Les différents éléments de l'interface graphique	20
2.3 L'avancée du projet	20
2.3.1 Les améliorations graphiques	21
2.3.2 Les améliorations techniques	22
3 Résultat final et bilan	25
3.1 L'application finale : Arches Walker	25
3.1.1 Aperçu de l'application	25

3.1.2	Les améliorations futures	26
3.1.3	Difficultés rencontrées et solutions proposées	27
3.2	Bilan du stage	28
	Conclusion	29
	Références	31
	Glossaire	32
	Annexes	33

Introduction

Dans le cadre de mon DUT Informatique, j'ai dû réaliser un stage de 10 semaines en entreprise. Recherchant un stage enrichissant et intéressant, j'ai été très satisfait lorsque André SCHAAFF a répondu à ma demande en me proposant un stage basé sur du développement Android. Durant ma deuxième année de DUT, j'ai déjà été amené à faire de la programmation Android, je savais donc que c'était un domaine qui me plaisait. De plus, cela m'a permis d'avoir un stage différent et quasi unique.

Le stage que j'ai effectué consiste en la réalisation d'une application Android pour tablettes fonctionnant comme une télécommande intelligente avec Aladin, un atlas interactif du ciel codé en Java qui sera actif sur un ordinateur standard. Aladin étant plus un logiciel à destination des spécialistes, l'idée est de faire découvrir Aladin — et plus généralement l'astronomie — au grand public de façon simple et intuitive via une tablette tactile.

Cette application est réalisée dans le cadre du projet ARCHES.

Une preuve de concept nommée *SkyTouch* montrant la faisabilité du projet, notamment en terme de communication, a déjà été réalisée par un ancien stagiaire.

Afin de présenter le travail que j'ai réalisé, j'organiserai mon propos en trois parties. Premièrement, je présenterai l'Observatoire de Strasbourg ainsi que mon sujet de stage plus en détail. Cela aura pour but de présenter le contexte dans lequel j'ai travaillé mais également de mieux comprendre comment ce sujet de stage est né. Je parlerai ensuite de la mise en œuvre et des différents outils et moyens utilisés à la réalisation du projet, l'intérêt étant ici d'avoir un aperçu plus technique du stage. Enfin, je présenterai mon travail et le résultat auquel je suis arrivé à la fin de mon stage.

1 Présentation du stage

1.1 L'Observatoire Astronomique de Strasbourg

1.1.1 Présentation générale

L'Observatoire Astronomique de Strasbourg est un Observatoire des Sciences de l'Univers ainsi qu'une Unité Mixte de Recherche de l'Université de Strasbourg et du CNRS. Il est actuellement dirigé par Hervé WOZNIAK.

Historiquement, Strasbourg a accueilli trois Observatoires Astronomiques. Le premier a vu le jour en 1673 grâce à l'astronome Julius REICHLET et se situait sur les tours d'enceinte de la ville. Le second à quant à lui été construit en 1828 sur le site de l'Académie de Strasbourg. Ce n'est qu'en 1881 que l'Observatoire Astronomique de Strasbourg rejoint son emplacement actuel. Celui-ci fait partie, avec le jardin botanique voisin, des actions mises en œuvre par l'empereur Guillaume I^{er} pour faire de Strasbourg une vitrine suite à la guerre de 1870.

Actuellement, l'Observatoire Astronomique de Strasbourg est composé de trois bâtiments dont le principal est celui de la Grande Coupole (cf. FIGURE 1). Ce dernier contient la 3^e plus grande lunette astronomique de France. Outre ses propres bâtiments, l'Observatoire héberge également le Planétarium de Strasbourg dont il a eu la responsabilité de 1986 à 2008. Celui-ci fait désormais partie du Jardin des Sciences de l'Université de Strasbourg.

L'Observatoire Astronomique de Strasbourg est constitué de trois équipes de recherches — Galaxies, Hautes énergies et le CDS — et de deux services d'observation de l'Institut National des Sciences de l'Univers — SSC-XMM et le CDS.

Si à l'origine, l'Observatoire avait pour quasi unique vocation l'observation de comètes, de météorites et d'étoiles variables, ses missions sont bien plus diversifiées à l'heure actuelle. En effet, l'Observatoire Astronomique de Strasbourg a pour mission de contribuer au progrès de la connaissance par l'acquisition de données d'observation, le développement de moyens appropriés ou encore l'élaboration des outils théoriques nécessaires. De plus, l'Observatoire est également chargé d'assurer la diffusion des connaissances, d'assurer la formation des étudiants et des personnels de recherche, de prendre part à des activités de coopération internationale, et bien plus encore.

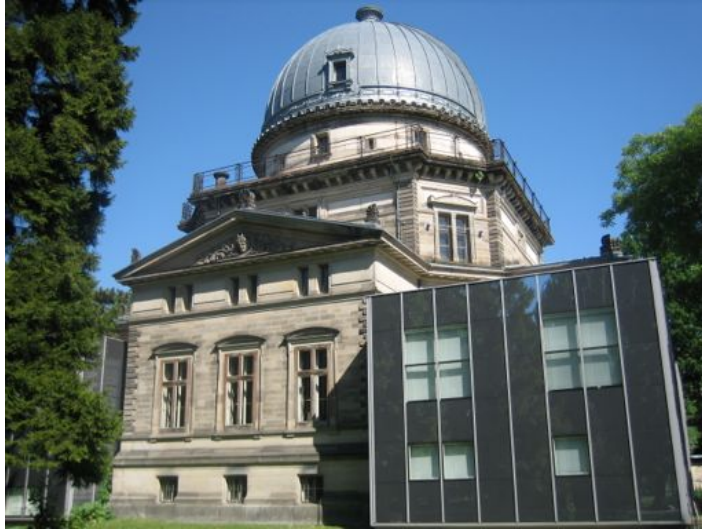


FIGURE 1 – Bâtiment de la Grande Coupole

1.1.2 Les équipes des recherches

Galaxies

Le champ d'action de l'équipe Galaxies regroupe tout ce qui concerne la formation des galaxies ainsi que l'étude et le recensement des populations stellaires qui composent ces galaxies. Plus particulièrement, cette équipe va s'intéresser à notre propre galaxie, la Voie Lactée, ainsi qu'à ses voisines du Groupe Local. Outre les recherches sur les populations stellaires, l'équipe Galaxies s'intéresse également à la dynamique gravitationnelle qui régit les étoiles et les matériaux à l'intérieur des galaxies. Le but de cela est de réunir suffisamment d'informations afin de pouvoir établir l'histoire précise de l'évolution d'un système stellaire et de reconstituer les événements clés dans la vie d'une galaxie.

En plus de cela, l'équipe Galaxies mène des recherches sur les amas stellaires — composants fondamentaux d'une galaxie — afin de mieux comprendre la formation de celles-ci.

Enfin, l'équipe s'implique également dans des missions satellitaires telles que Gaia, mission lancée en décembre 2013 par l'Agence Spatiale Européenne ayant pour but de fournir des relevés sur la Voie Lactée.

Hautes énergies

L'équipe Hautes énergies s'intéresse aux sources émettrices de rayon X, aux objets compacts — étoiles à neutrons par exemple — et aux noyaux actifs des galaxies. Elle est impliquée dans le Survey Science Center d'XMM (SSC-XMM), un consortium international de laboratoires sélectionnés par l'ESA qui est en charge de fournir des

catalogues complets d'objets observés par le satellite XMM-Newton.

L'équipe Hautes énergies participe également à des projets communautaires tels que le projet européen ARCHES, en collaboration avec le CDS.

CDS

Le Centre de Données astronomiques de Strasbourg est à la fois un service d'observation et une équipe de recherche. Le CDS a entre autre développé des services d'accès aux données astronomiques — Simbad, VizieR — et de visualisation de ce même données — Aladin — qui sont utilisés par l'ensemble de la communauté internationale. C'est aussi l'un des acteurs majeurs du développement de l'International Virtual Observatory Alliance (IVOA). Depuis 2008, le CDS a été labellisé « Très Grande Infrastructure de Recherche », ce qui le place au même niveau que des infrastructures européennes comme l'European Southern Observatory.

1.2 Le CDS

Ayant effectué mon stage au sein de ce service, je vais maintenant le présenter plus en détail.

1.2.1 Présentation générale

Créé en 1972, le CDS est d'abord connu sous le nom de Centre de Données Stellaires avant de devenir le Centre de Données astronomiques de Strasbourg en 1983. Hébergeant entre autres la base de données de référence pour l'identification d'objets astronomiques, ces services — Simbad, VizieR et Aladin — sont largement utilisés par la communauté astronomique internationale.



FIGURE 2 – Logo du CDS

Ses missions sont nombreuses et comprennent entre autres :

- le rassemblement des informations utiles concernant les objets astronomiques,

- la mise à jour de ces données,
- la distribution de ces données à la communauté internationale,
- la mise en place de recherche à propos de ces données.

Le CDS emploie actuellement 33 personnes réparties de la façon suivante : 8 chercheurs/astronomes-adjoints, 1 chercheur postdoctoral, 11 informaticiens, 10 documentalistes et 3 administratifs.

1.2.2 Les différents services du CDS

Simbad

Simbad est la base de données de référence mondiale en ce qui concerne la nomenclature et la bibliographie d'objets astronomiques. Possédant l'équivalent de plus de 40 ans de données, Simbad permet aux astronomes de trouver facilement des informations sur plus de 7 millions d'objets grâce à plus de 290 000 références bibliographiques. Simbad dispose également d'un résolveur de noms permettant de retrouver n'importe quel objet désigné par l'un des 18 millions d'identifiants que la base contient. Simbad a reçu en moyenne 284 000 requêtes par jour pour l'année 2011, soit plus de 3 requêtes par secondes.



FIGURE 3 – Logo de Simbad

VizieR

VizieR est une base de données regroupant des catalogues d'objets astronomiques. Ces catalogues sont constitués de données relevées durant des missions d'observation et sont ajoutés à VizieR par les documentalistes du CDS. A l'heure actuelle, la base est constituée de plus de 12 000 catalogues.

Interrogeable sur de nombreux critères (longueur d'ondes, nom de la mission, etc.), VizieR permet de rassembler et d'homogénéiser les données astronomiques afin de pouvoir les comparer et les exporter.

Entre 2011 et 2012, le nombre de requêtes faites sur la base de données ont été en moyenne de 370 000 par jour, avec des pics à plus de 2 millions.



FIGURE 4 – Logo de VizieR

Aladin

Aladin est un atlas interactif du ciel permettant de visualiser et de comparer des images du ciel. Il a été entièrement développé en Java par Pierre FERNIQUE, Thomas BOCH et François BONNAREL.

Aladin utilise des images provenant d’observatoires au sol ou spatiaux et génère des cartes du ciel en trois dimensions (technologie HEALPix). Ces images peuvent être issues de la base de données inclus à Aladin, d’autres bases telles que la NASA Extragalactical Database, ou encore provenir de l’utilisateur lui-même.

A l’heure actuelle, Aladin est disponible sous quatre formes [4] : une application téléchargeable, une applet Java, une version JavaScript et un simple previewer en ligne.



FIGURE 5 – Logo d’Aladin

1.2.3 International Virtual Observatory Alliance

L’international Virtual Observatory Alliance est une organisation internationale regroupant les projets d’Observatoire Virtuel¹ plus de 20 pays (France, Allemagne,

1. Virtual Observatory ou VO : projet dont le but est de créer un réseau transparent similaire à Internet pour les données astronomiques

Chine, États-Unis, Brésil, Australie, . . .). Le but de l'IVOA est de mettre en commun au niveau mondial les travaux astronomiques et d'établir des standards ainsi que des stratégies à adopter. Cela permet de faciliter les échanges internationaux au même titre que la collaboration entre les différents pays.



FIGURE 6 – Logo de l'IVOA

Le CDS participe activement à ce projet par la réalisation des trois services cités auparavant, mais aussi par son implication au sein des groupes de travail de l'IVOA. En effet, plusieurs de ces groupes de travail qui composent l'IVOA sont présidés ou vice-présidés par des membres du CDS.

1.3 Sujet de stage

1.3.1 A l'origine du projet

Archés

ARCHES — Astronomical Resource Cross-matching for High Energy Studies — est un projet européen basé sur les observations en rayon X du satellite XMM-Newton. Le projet ARCHES suit plusieurs buts, le principal étant l'amélioration du catalogue XMM-Newton, grâce notamment à la création de *spectral distributions* multi-longueur d'ondes pour les objets présents dans le catalogue. L'intérêt de cela étant de diffuser ensuite l'ensemble des éléments, à la communauté scientifique mais aussi à des fins éducatives, afin de faire avancer la recherche.



FIGURE 7 – Logo du projet ARCHES

C'est dans cette optique éducative que s'inscrivent mon stage et l'application An-

droid que j'ai réalisée. Ce projet — nommée Arches Walker — a pour but la distribution de l'application créée afin d'encourager le grand public, et surtout les étudiants, à s'intéresser à la science. Afin de plaire au plus grand nombre, l'application se doit d'être attirante et facile à prendre en main. De plus, comme le but est de distribuer l'application au sein des institutions du projet ARCHES, l'application doit être traduite en quatre langues (anglais, français, allemand et espagnol).

SkyTouch

Réalisé en 2013 par Maxime HECKEL, un stagiaire de l'ENSIIE de Strasbourg, SkyTouch est une application Android permettant de contrôler à distance Aladin. Cette application a servi de preuve de concept, permettant ainsi de s'assurer de la faisabilité du projet. L'idée a donc été de reprendre le principe de SkyTouch, i.e. le contrôle d'Aladin à distance, et de l'adapter au projet ARCHES (cf. Annexe A).

1.3.2 Énoncé détaillé du sujet

L'ensemble du projet Arches Walker se compose de deux parties :

- une partie **ordinateur** sur lequel est affiché Aladin,
- une partie **tablette** servant de télécommande pour contrôler Aladin à distance.

Étant prévu pour un stage de plus de 10 semaines, la réalisation de l'application dans son intégralité n'était pas possible. Mon stage consistait donc en la réalisation de la partie graphique tablette et, suivant l'avancement du projet, de la mise en place de la communication entre l'ordinateur et la tablette.

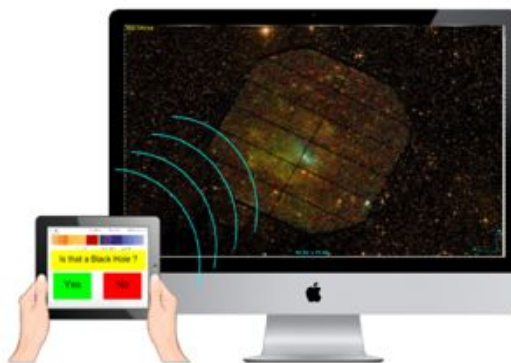


FIGURE 8 – Principe de fonctionnement de l'application Arches Walker

L'application Android propose à l'utilisateur deux scénarios différents. Le premier est un mode « découverte » : l'utilisateur choisit une classe d'objets (galaxies, étoiles, etc.) et choisit une instance de cette classe (Voie Lactée, Andromède,

etc.) parmi une liste proposée. En plus de cela, l'utilisateur peut, via des boutons, accéder à une description de l'objet qu'il a choisi, le visualiser dans d'autres longueurs d'ondes ou encore obtenir des grandeurs qui caractérisent l'objet. Tous ces éléments sont affichés sur l'écran de l'ordinateur ; il convient donc à l'application d'émettre les bons messages au bon moment afin de répondre aux différentes actions de l'utilisateur.

Le second scénario dispensé par l'application est un quiz. L'application choisit aléatoirement une classe d'objets et en affiche une instance sur Aladin, puis l'utilisateur doit trouver à quelle classe l'instance affichée appartient. Pour cela, il peut s'aider en affichant des grandeurs caractéristiques ou en demandant à voir une autre instance de la même classe. Une fois la bonne réponse trouvée, l'application passe en mode découverte sur la classe d'objets en question. L'utilisateur peut alors en apprendre plus sur cette classe ou bien continuer avec une nouvelle question.

2 D eroulement du stage

2.1 Mise en place de l'environnement de travail

2.1.1 Les grandes lignes du d veloppement Android

Afin de rendre plus clair mes propos, j'ai d cid  de pr senter les points importants du d veloppement Android avant de rentrer dans les d tails plus techniques de mon stage. De fa on simplifi e, d velopper une application Android consiste   d velopper en Java et   y ajouter des API/classes sp cifiques.

Activit s & Fragments

L'Activit  est l' l ment de base de l'application Android, toute application  tant constitu e d'au moins une Activit . Plus concr tement, chaque partie distincte d'une application correspond   une Activit , comme les diff rentes pages d'un site internet par exemple.

Au sein d'une Activit , les Fragments constituent une nouvelle division apparue   partir de la version 3.0 d'Android. Un Fragment correspond   une section — aussi bien en terme de design qu'en terme de fonctionnalit s — d'une Activit . Une Activit  peut comporter plusieurs Fragments diff rents et ces derniers ne sont pas li s   une unique activit . Toutefois, le Fragment ne peut pas exister de lui-m me, i.e. si une Activit  est d truite, il en est de m me pour les Fragments qu'elle contient (cf. Annexe C).

De plus en plus, le d veloppeur Android va chercher   privil gier les Fragments au profit des Activit s. En effet, les Fragments sont plus l gers, plus faciles   manipuler et   contr ler. On se tourne donc de plus en plus vers des applications Android comportant peu d'Activit s mais beaucoup de Fragments.

Le dossier *res*

Le dossier *res* va, entre autre, contenir tout ce qui concerne l'aspect graphique d'une application. Ce dossier est organis  en plusieurs sous-dossiers — *drawable*, *values*, *layouts*, . . . — regroupant tous les fichiers d'un m me type. Ainsi, toutes les ic nes et images de l'application vont se situer dans le dossier *drawable* tandis que l'architecture des diff rentes Activit s et Fragments seront dans le dossier *layout*. En plus de regrouper les diff rents fichiers, le dossier *res* a une autre utilit  tr s importante : il permet de « personnaliser » le rendu selon de nombreux crit res.

Ce dossier est donc particuli rement utile quant   la gestion des diff rentes langues et des diff rentes tailles d' cran. L'id e est simplement de rajouter un suffixe aux sous-

dossiers pour les rendre plus sélectifs (cf. Annexe B). Par exemple, en stockant la traduction anglaise des textes dans le dossier *values-en*, l'application va automatiquement utiliser le texte anglais plutôt que le français si la tablette est en anglais. Le fonctionnement est identique pour les tailles d'écran, ce qui permet de définir différentes tailles d'images et de polices par exemple. Cela facilite ainsi grandement la mise en place d'une application *responsive*, i.e. qui s'adapte correctement à toutes les tailles d'écrans.

2.1.2 Les outils de développement

Android Studio

Android Studio est un IDE² développé par Google dédié au développement Android. Bien qu'encore en version bêta, j'ai décidé d'utiliser Android Studio plutôt qu'un autre IDE plus classique comme Eclipse durant mon stage.



FIGURE 9 – Logo d'Android Studio

Durant les premiers jours de stage, M. SCHAAFF m'a parlé d'Android Studio comme une éventuelle solution alternative à Eclipse. Déjà disponible depuis plus d'un an, le logiciel n'était pas assez avancé pour une utilisation réelle lors de la réalisation du projet SkyTouch. J'ai donc décidé de le télécharger pour voir ce qu'il en était à l'heure actuelle.

Côté installation, celle d'Android Studio s'est avérée être beaucoup plus simple et plus rapide que celle d'Eclipse. En effet, étant dédié au développement Android, Android Studio inclut directement les outils nécessaires — SDK Android, Android Developer Tool, ... — tandis qu'il faut les ajouter manuellement dans Eclipse. Côté pratique, Android Studio est plus facile à prendre en main et plus efficace selon moi. De plus, Android Studio semble disposer de plus de fonctionnalités qui facilitent le codage. J'ai donc rapidement fait mon choix et poursuivi mon stage en utilisant Android Studio.

2. Integrated Development Environment : logiciel regroupant un ensemble d'outils facilitant la programmation

ADB shell

Android Debug Bridge est un outil en ligne de commande inclus dans le SDK Android permettant de communiquer et de gérer les appareils Android connectés à l'ordinateur.

Dans le cadre de mon stage, j'ai utilisé la commande *shell* d'ADB qui permet de contrôler et d'avoir accès aux données internes du téléphone depuis l'ordinateur. Ainsi, il est possible de modifier les caractéristiques techniques de l'appareil utilisé (cf. Annexe D), notamment concernant l'affichage. Même en n'ayant pas accès à d'autres tablettes Android directement sur mon poste de travail, j'ai ainsi pu simuler différentes tailles et différentes résolutions d'écrans grâce à cet outil, tout cela dans le but de rendre l'application adaptée à un maximum de tablettes.

Tablette Nexus 10

Afin de pouvoir tester mon application sur un appareil réel et pas uniquement sur émulateur, l'Observatoire m'a fourni une tablette tactile Google Nexus 10 ayant les caractéristiques suivantes :

- Version d'Android : 4.4.2 (KitKat)
- Taille d'écran : 2560x1600 px
- Densité - résolution : 300 ppp

Grâce à M. SCHAAFF, j'ai également pu avoir un aperçu de l'application sur une tablette Nexus 7, ce qui a été utile au moment il a fallu s'occuper de l'aspect graphique et le rendre homogène sur des tailles de tablettes différentes.

2.1.3 Autres ressources utilisées

Codes Sources

Ayant déjà réalisé un projet Android durant ma deuxième année de DUT, j'ai récupéré le code de ce projet afin de pouvoir, si nécessaire, le réutiliser ou au moins m'en inspirer. Cela s'est révélé utile notamment lorsqu'il a fallu mettre en place la récupération des données stockées dans un fichier externe. J'ai ainsi eu l'occasion de gagner du temps sur certains points car j'ai eu accès rapidement à une solution fonctionnelle. En plus de cela, j'ai également eu accès au code source de SkyTouch, application dans laquelle la partie communication tablette-ordinateur fonctionne correctement. Malheureusement, le code de la partie qui m'intéressait a été codé en JavaScript, je n'ai pas vraiment pu l'utiliser. En effet, la partie communication d'Arches Walker est basée sur du Java via l'utilisation de Servlets avec Tomcat.

Tomcat

Tomcat est un conteneur web de Servlets Java développé par la fondation Apache. Il comporte également un serveur HTTP et va être utilisé comme serveur pour gérer la communication entre Aladin et la tablette tactile.

Un Servlet est une classe Java permettant de créer ou de modifier des données dynamiquement au sein d'un serveur HTTP. Ces données sont généralement présentées au format HTML, mais peuvent aussi l'être au format JSON comme ce sera le cas ici. Les Servlets sont exécutés côté serveur par un conteneur, ici Tomcat, ce qui leur permet de persister tout au long des requêtes clients.

Internet

Datant de plusieurs années maintenant, Android a attiré un grand nombre de personnes depuis sa création en 2005. La communauté autour du développement Android est donc très présente sur Internet, ce qui a été un atout durant mon stage. J'ai ainsi pu trouver rapidement des solutions à mes problèmes en me basant sur les solutions proposées aux personnes ayant rencontré les mêmes difficultés. De même, il était assez facile de trouver des sites proposant codes sources et bibliothèques répondant aux besoins que j'avais, ce qui m'a permis de gagner du temps.

Enfin, le site dédié au développement Android [1] a été une source d'information d'une grande aide tout au long du stage. Très bien réalisé et très complet, ce site contient notamment toutes les informations utiles sur les classes et sur les outils de développement ainsi que de nombreux renseignements quant au design général d'une application Android.

2.2 Les débuts : première approche et résultats

2.2.1 Premières versions de l'application

Une fois l'environnement de travail mis en place, le projet a commencé. Il a fallu dès le début réfléchir sur l'aspect général de l'application. A ce stade, il ne s'agissait pas encore de rentrer dans les détails graphiques mais de définir le cheminement de l'utilisateur dans l'application.

Après un temps de réflexion, j'ai commencé mon travail en me basant sur l'idée suivante. Au départ, l'utilisateur se situe face à un menu permettant de choisir le mode de jeu : mode *découverte* ou mode *quiz*. S'il se dirige vers le mode découverte, il doit une classe d'objets ainsi qu'une instance de cette classe. A partir de là, un

nouveau Fragment (cf. Annexe C & partie 2.1.1) se lance, l'instance est affichée et l'utilisateur peut interagir avec Aladin. Si au contraire, l'utilisateur choisit le quiz, il doit deviner à quelle classe appartient une instance affichée à l'écran avant de découvrir cette instance plus en détail. L'idée était d'utiliser la même interface pour le mode découverte et la partie découverte du quiz.

Après avoir fait part de mon idée à mes tuteurs, il en est ressorti que j'avais laissé de côté un aspect majeur de l'application : l'aspect « télécommande ». Suite aux conseils de M. SCHAFF et M. MICHEL, j'ai donc repensé l'application en gardant en tête que chaque mode devait être affiché sur une seule fenêtre. Il a donc fallu regrouper les éléments afin de les disposer différemment pour avoir accès à toutes les fonctionnalités.

Une fois fixé sur la hiérarchie générale du projet, il a donc fallu réfléchir plus précisément au positionnement des éléments dans les différentes Activités de l'application. La principale difficulté rencontrée ici se situait au niveau du mode Découverte. En effet, il a fallu regrouper un grand nombre de fonctionnalités — choix de la classe d'objets et d'une instance de cette classe, différents boutons commandants Aladin, ... — sur l'écran de la tablette. La partie quiz a posé moins de problèmes, il a suffi d'adapter l'interface du mode découverte pour correspondre aux critères du quiz. C'est à partir de ce moment-là que l'application a vraiment pris forme et s'est visuellement approchée du résultat final.



FIGURE 10 – 2^e version de l'application Arches Walker

2.2.2 Les différents éléments de l'interface graphique

Pour arriver aux résultats ci-dessus et pour répondre aux problèmes posés par la présence d'un grand nombre d'éléments sur un seul écran, j'ai dû me poser la question de l'organisation de l'interface de l'application. Après avoir réfléchi à plusieurs façons différentes d'organiser cette interface, j'ai choisi de m'orienter vers une division en 2 parties :

- la partie de gauche, sous forme de liste, permet de choisir parmi les différentes classes d'objets celle qui nous intéresse,
- la partie droite, centrée autour des instances, va permettre à l'utilisateur de choisir une instance de la classe choisie et d'interagir avec Aladin. C'est cette partie qui regroupe les éléments nécessaires au bon fonctionnement de l'application. On y trouve ainsi les boutons permettant d'afficher toutes les informations utiles — description de la classe et de l'instance, SED, grandeurs caractéristiques à propos de l'instance choisies. De même, un TouchPad a été ajouté afin de pouvoir se déplacer et zoomer sur Aladin, comme c'était le cas dans l'application SkyTouch. Enfin, cette partie comporte également une barre permettant de modifier la longueur d'onde dans laquelle l'instance est affichée comme prévu dans le cahier des charges.

Le système de liste a été simple à mettre en place car c'est une disposition proposée nativement par Android lors de la création de nouvelles Activités. Je n'ai donc pas eu à implémenter par moi-même la partie concernant la communication entre la liste et le panneau de droite. Un fond d'écran provisoire a été ajouté afin d'avoir un aperçu plus fidèle du résultat final. Toujours dans le but d'approcher le résultat final, j'ai également créé de fausses classes et instances afin de ne pas présenter une application sans aucune donnée.

2.3 L'avancée du projet

Au fil du temps, de nombreuses remarques et suggestions de la part de mes tuteurs m'ont poussé à améliorer continuellement l'application que j'ai réalisée. Peu de temps avant la fin du stage, nous avons décidé d'un commun accord de revoir les priorités quant à l'application. Ainsi, plutôt que de se concentrer principalement sur l'aspect graphique de toute l'application, M. SCHAAFF, M. MICHEL et moi-même avons décidé de laisser de côté la partie quiz au profit de la partie découverte. Plutôt que de présenter une application avec plusieurs fonctionnalités mais non abouti, nous avons préféré un résultat final moins diversifié mais pleinement fonctionnel.

2.3.1 Les améliorations graphiques

Bien que déjà bien avancé, l'interface a malgré tout subi plusieurs changements afin de la rendre plus attrayante et plus simple à comprendre pour le grand public. En effet, l'interface n'était pas défini de manière optimale : par exemple, le bouton activant la description de la classe était situé avec la partie *instance* et inaccessible sans choisir d'instance par exemple. J'ai donc modifié l'interface au fur et à mesure en suivant les recommandations de mes tuteurs de stage.

Fond d'écran de l'application

Le fond d'écran représente une partie importante dans une application car il va orienter les choix des couleurs et du style des autres éléments de l'application (boutons, textes, ...). J'ai décidé dès le départ d'utiliser sur un fond composé de couleurs sombres, mais je n'avais pas d'idées précises en tête. M. MICHEL a donc proposé d'utiliser des images d'Aladin, plus particulièrement celles issues relevés du satellite XMM-Newton comme le projet ARCHES est ciblé autour de ces relevés. De plus, utiliser un tel fond permet de montrer aux utilisateurs de vraies images astronomiques, différentes de celles que le grand public à l'habitude de voir car très peu modifiées. J'ai donc suivi cette idée et trouvé plusieurs images pouvant potentiellement être utilisées (cf. Annexe E). Il a toutefois fallu les adapter un peu pour pouvoir les intégrer à l'application, notamment en matière de luminosité. Parmi les différents choix, nous sommes rapidement tombés d'accord sur une image qui a été mise en place définitivement.

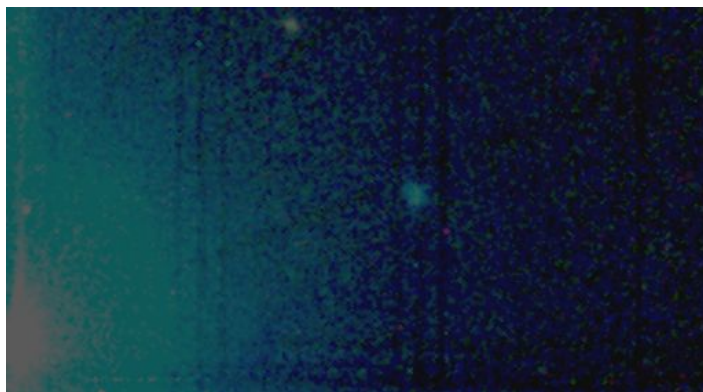


FIGURE 11 – Fond d'écran retenu

Boutons : dispositions et styles

Grâce à la mise en place du fond d'écran, il a été possible de définir une gamme de couleur pour les différents boutons de l'application. Cela a malgré tout pris un certain temps pour trouver un visuel qui plaisait à tous et qui était adapté à l'application.

La disposition des boutons a également dû être revue puisqu'elle n'était pas pratique. Ainsi, le bouton concernant la description de la classe a été déplacée pour être plus cohérent. Cette action a eu pour effet de libérer de la place et donc de rendre l'application plus attractive et plus claire. Cette disposition a encore changé avant d'arriver au résultat final.

Suite à la demande de mon tuteur, les différents traits séparant les parties ont disparu, toujours dans un but de clarté. De même, le texte des boutons a disparu, l'idée étant d'interpeller l'utilisateur via les icônes pour l'inciter à appuyer sur les boutons.

Cover Flow

Cover Flow est une interface utilisateur servant à naviguer à travers une bibliothèque de données. Ce système a été créé par deux développeurs puis racheté par Apple qui l'a utilisé dans son logiciel iTunes entre autre (cf. FIGURE 12). Avec mes tuteurs, nous avons décidé de reprendre ce système pour l'affichage et la sélection des différentes instances. Cela permet de limiter le nombre d'actions que l'utilisateur doit faire en sélectionnant automatiquement l'instance se trouvant au milieu (se déplacer uniquement contre se déplacer et cliquer avant). Après plusieurs tentatives vaines, j'ai trouvé sur Internet une librairie qui répondait à mes attentes, ce qui m'a permis de mettre cela en place rapidement.



FIGURE 12 – Aperçu du système *Cover Flow* sur iTunes

2.3.2 Les améliorations techniques

En plus de l'aspect graphique, il a fallu mettre en place tout le code permettant à l'application de fonctionner correctement. Cette partie, bien qu'invisible pour l'utilisateur, représente un aspect important de l'application.

Aspects techniques du mode de jeu

Pour que l'application soit fonctionnelle, il a été nécessaire de rendre tous les éléments de l'interface actifs. La première étape consistait à adapter le Cover Flow permettant de sélectionner les instances. Comme Aladin doit afficher en temps réel l'instance sélectionnée, j'ai modifié le système de sélection de base. J'ai donc mis en place un système qui attend un temps prédéfini avant de valider la sélection. Ainsi, cela laisse le temps à Aladin de se déplacer sur l'instance en question et évite de surcharger le serveur de messages dès que l'utilisateur change d'instance.

Concernant les différents boutons, il a fallu mettre en place l'envoi de message permettant de contrôler Aladin. J'ai donc dû attendre la mise en place du serveur avant de réellement pouvoir tester le fonctionnement des boutons. Outre ces points, j'ai également du rendre le *TouchPad*³ fonctionnel, i.e. gérer le déplacement et le zoom. C'est d'ailleurs cette dernière partie qui a été la plus contraignante.

Mise en place et connexion au serveur Tomcat

Les données de jeu étant stockées sur l'ordinateur et non sur la tablette, il a fallu mettre en place un serveur Tomcat auquel la tablette se connecte par la suite pour récupérer ces données. Grâce à l'aide de Laurent MICHEL qui s'est occupé de la partie serveur, la mise en place de Tomcat a été rapide. Une fois cela fait, il a fallu se connecter à ce serveur via Wi-Fi. Cette partie a nécessité que j'effectue de nombreuses recherches, à la fois pour se connecter mais aussi pour récupérer l'adresse IP du serveur depuis la tablette. En effet, cette adresse IP est nécessaire puisque la tablette va envoyer des messages au serveur. C'est d'ailleurs cette partie qui semblait problématique au départ. Le développement Android étant toutefois très répandu sur Internet, j'ai rapidement trouvé ce que je cherchais et cela s'est avéré plutôt simple à mettre en place.

```
/**
 * Get the remote IP.
 * The property dhcp1.serverAdresse is an Integer and must be formatted into a String ("x.x.x.x")
 */
public void getServerIp() {
    DhcpInfo dhcp1 = wn.getDhcpInfo();
    int ip = Integer.valueOf(dhcp1.serverAdresse);
    serverIp = intToStrIp(ip);
}

/**
 * Convert IP from int to string ("x.x.x.x")
 * @param ip
 */
public static String intToStrIp(int ip) {
    String ipString = String.format(
        "%d.%d.%d.%d",
        (ip & 0xff),
        (ip >> 8 & 0xff),
        (ip >> 16 & 0xff),
        (ip >> 24 & 0xff));
    return ipString;
}
```

FIGURE 13 – Récupération de l'IP du serveur depuis la tablette

3. TouchPad : zone tactile permettant de déplacer l'image ou de zoomer sur Aladin

Génération dynamique des données

Une fois le serveur mis en place, il a été possible de récupérer les données stockées sur celui-ci. L'ensemble des informations nécessaires est stocké dans des fichiers JSON (cf. Annexe F) qu'il faut traiter au niveau de la tablette. Pour récupérer ces fichiers, il suffit de se connecter au serveur via une simple url pour obtenir le code de la page en question. La partie importante a donc été la mise en place cet url nécessitant entre autre l'adresse IP du serveur et le nom des dossiers à parcourir.

Une fois les fichiers récupérés, il fallait simplement les parcourir à l'aide d'objets *JSONArray* et *JSONObject* pour initialiser les données sur la tablette. Les images utilisées comme icônes m'ont posé quelques problèmes et m'ont obligé à modifier les classes que j'avais créées pour gérer les différents d'objets astronomiques. En effet, je ne pouvais pas les manipuler directement comme des images propres à l'application, il a donc fallu que je les convertisse au bon format pour pouvoir les utiliser simplement.

3 Résultat final et bilan

3.1 L'application finale : Arches Walker

3.1.1 Aperçu de l'application

Après 10 semaines de stage, l'application que j'ai réalisée dans le cadre du projet ARCHES est convaincante. Même si la partie quiz a été mise de côté, l'application est malgré tout utilisable en tant que telle grâce à une partie découverte fonctionnelle et remplissant la majorité des critères demandés.



FIGURE 14 – Aperçu du résultat final de l'application

L'application comprend actuellement les fonctionnalités suivantes :

- une liste permettant de choisir une classe d'objets astronomiques parmi plusieurs proposées,
- une liste de type *Cover Flow*⁴ permettant de choisir une instance parmi toutes les instances de la classe choisie auparavant,
- 3 boutons permettant d'afficher au choix sur l'écran de l'ordinateur une description de la classe choisie, un SED ou des valeurs caractérisant l'instance choisie,
- un TouchPad permettant de zoomer ou de déplacer la vue courante sur Aladin,
- une barre permettant de changer la longueur d'onde dans laquelle est affichée l'instance courante.

4. Cover Flow : interface servant à naviguer dans une bibliothèque de données

En plus de cela, une description de la classe d'objets est automatiquement affichée lorsqu'une classe est choisie.

L'utilisation de l'application nécessite la mise en place d'un réseau ad-hoc utilisant le Wi-Fi, un serveur Tomcat contenant les données à charger par l'application Arches Walker ainsi qu'une application serveur gérant Aladin et les fenêtres d'affichage sur l'ordinateur (partie réalisée par mon tuteur Laurent MICHEL, cf. Annexe G).

3.1.2 Les améliorations futures

Bien que pouvant déjà être utilisée à l'heure actuelle, l'application nécessite encore du travail pour remplir toutes les fonctionnalités demandées.

Parmi les ajouts à faire, la partie la plus importante concerne le mode quiz. Comme l'application s'est limitée au mode découverte que vers la fin de mon stage, j'ai également eu l'occasion de travailler sur ce mode de jeu. Ainsi, la génération aléatoire de la solution ainsi que l'interface sont déjà existantes. Il ne reste donc plus qu'à l'adapter pour avoir un style homogène et l'intégrer à l'application. Pour ce faire, il est également nécessaire de prévoir un menu de sélection, déjà existant lui aussi, ainsi qu'un moyen de naviguer entre ces deux modes.

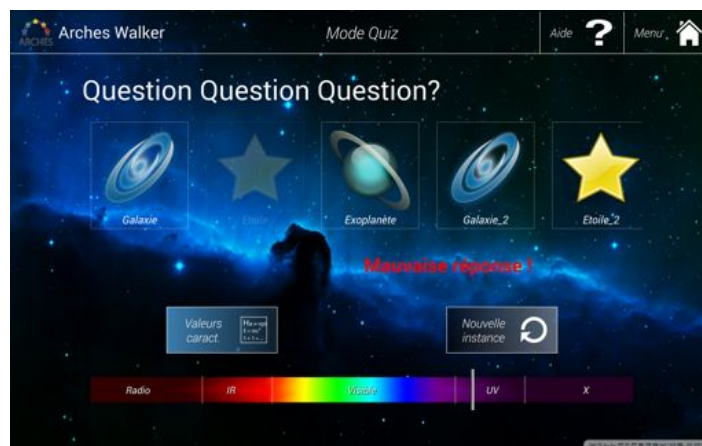


FIGURE 15 – Etat du mode quiz à l'heure actuelle

Une autre amélioration importante consiste en l'ajout de nouvelles données pour l'application. En effet, le nombre d'objets présents à l'heure actuelle est très limité. Malgré tout, de par son implémentation, l'ajout de nouvelles données est très facile à mettre en place : il suffit de modifier le fichier nommé *summary.json* et de fournir les images et autres données nécessaires (description, SED et grandeurs).

Enfin, un réel apport pour la distribution de l'application serait d'automatiser la mise en place du réseau Wi-Fi et de la partie serveur nécessaire au bon fonctionnement de l'application. En effet, tout cela doit être fait manuellement à l'heure actuelle, ce qui n'est pas très pratique pour un utilisateur lambda. Cela pourrait se présenter sous la forme d'un fichier — script ou exécutable — à lancer sur l'ordinateur par exemple.

3.1.3 Difficultés rencontrées et solutions proposées

Durant toute la durée de mon stage, j'ai dû faire faces à plusieurs difficultés. Ces difficultés ont nécessité temps et réflexion afin de trouver des solutions simples et efficaces.

Le premier point problématique est apparu dès le début du stage. En effet, bien que le sujet de stage est très clair quant à son fonctionnement général, il n'y a pas de contraintes à proprement parler concernant l'interface. Il ne fallait donc pas se précipiter afin de ne pas partir dès le départ sur quelque chose d'irréalisable. J'ai donc dû réfléchir plus en détail à l'interface que j'allais mettre en place. N'ayant jamais fait cela auparavant, je ne savais pas réellement si les premières ébauches que je réalisais étaient convenables. Il s'est avéré que la première version de l'application ne répondait pas de façon satisfaisante aux critères demandés. Grâce aux conseils de mes tuteurs, j'ai réalisé une deuxième version plus réfléchie de l'interface. Cette deuxième idée est plus réussie que la première. En effet, malgré de nombreux changements qui sont intervenus par la suite, la deuxième version contenait dès le départ des éléments actuellement dans la version finale de l'application.

Concernant la partie graphique, un autre aspect n'a pas été des plus simples à gérer : rendre le design de l'application *responsive*. Cela n'a toutefois pas été une surprise pour moi car j'ai déjà été confronté à ce problème lors du projet Android que j'ai réalisé dans le cadre de mon DUT Informatique. A l'époque, le problème avait été traité en partie seulement, ce qui était impossible dans un cadre professionnel. N'ayant pas pensé à réfléchir à cela dès le début, j'ai par la suite dû modifier de nombreuses parties de mon code afin de le rendre adaptatif aux différentes tablettes.

Bien qu'Android prévoit un système de fichier prévu pour simplifier la mise en place d'un design adaptatif, il a tout de même fallu que je fasse de nombreux tests avant d'arriver à un résultat convenable. En effet, il a été nécessaire que je fournisse des images de tailles différentes adaptées à toutes les tailles d'écrans ciblées. De façon

identique, j'ai dû définir des tailles de polices et de marges différentes selon les types de tablettes.

Pour effectuer mes tests et détecter les éventuels problèmes, j'ai dû utiliser l'outil ADB shell car je ne disposais que d'une seule tablette à ma disposition. Cela m'a été très utile pour avoir un aperçu du rendu de l'application sur des tablettes plus petites. Je me suis cependant rendu compte grâce à M. SCHAAFF qui a testé l'application sur sa propre tablette — une tablette Google Nexus 7 — que l'aperçu via ADB shell et le rendu réel était parfois différent. Il a donc fallu réajuster certains points. Grâce à cela et à l'aide fourni sur le site dédié à Android [1], j'ai ainsi réussi à mettre en place une interface qui s'adapte correctement à la plupart des tablettes du marché.

3.2 Bilan du stage

Après 10 semaines de stage, je pense avoir atteint l'objectif fixé initialement, même si beaucoup de choses peuvent encore être améliorées. En effet, l'application fonctionne correctement et peut être utilisée à l'heure actuelle (sous réserve d'avoir mis en place la partie serveur).

Durant mon stage, j'ai eu l'occasion de voir mon travail présenté à la communauté internationale [3]. M.SCHAAFF a ainsi présenté l'application que j'ai réalisé lors d'une conférence de l'IVOA (cf. Annexe H). Il en est ressorti des retours positifs de la part de l'audience. L'application semble donc intéresser le public, ce qui était l'objectif principal.

Dans l'ensemble, je suis très satisfait de mon stage et de l'application que j'ai réalisé au sein de l'Observatoire Astronomique de Strasbourg . Ce stage a été une expérience professionnelle enrichissante et intéressante, aussi bien au niveau du sujet qu'au niveau de l'environnement de travail.

Grâce à ce stage, j'ai pu acquérir de nombreuses connaissances liées au développement Android. Cela m'a également permis de travailler en autonomie sur un projet, et cela dès le début. Ce dernier point a été un réel apport car je me suis rendu compte qu'il est primordial de ne pas se précipiter pour mener à bien un projet.

J'ai également eu la chance de travailler avec des personnes qui m'ont aidé lorsque j'en ai eu besoin et qui m'ont guidé tout au long du stage.

Conclusion

L'objectif initial du stage étant la réalisation de la partie graphique d'une application Android, j'ai été amené à faire de nombreuses recherches à ce sujet dans un premier temps. Par la suite, le fait de redéfinir les priorités du sujet avec mes tuteurs m'a permis de diversifier mon travail et donc les connaissances que j'ai acquises. Grâce aux outils que m'a fourni l'Observatoire Astronomique de Strasbourg, j'ai pu tester en temps réel le rendu de l'application et faire les changements nécessaires pour répondre aux critères définis. Cela m'a également permis de présenter mon travail à mes tuteurs qui ont ainsi pu me donner conseils et recommandations plus qu'utile. Bien qu'ayant rempli les objectifs qui m'étaient fixés, il reste encore de nombreuses améliorations possibles afin de rendre l'application plus intéressante et plus facile à prendre en main pour l'utilisateur.

Avec le recul et l'expérience que j'ai acquis durant mon stage, je m'aperçois aujourd'hui des erreurs que j'ai commises. Je me suis rendu compte de l'importance d'avoir une réflexion préalable au développement pour éviter de perdre du temps. C'est principalement sur ce point que je changerai ma façon de faire si ce stage était à refaire.

Enfin, ce stage représente un apport professionnel et personnel important, notamment de par l'autonomie dont j'ai dû faire preuve. Je me suis également aperçu de l'importance de l'informatique et des informaticiens dans la science et la recherche. Ainsi, même dans un domaine de recherche tel que l'astronomie, il est important de se rendre compte de la place de l'apport de l'informatique.

Références

- [1] Google - Android. *Android Developers*. 2014. URL : <http://developer.android.com/index.html>.
- [2] Stack Exchange network. *Stack Overflow*. 2014. URL : <http://stackoverflow.com/>.
- [3] André SCHAAFF. *IVOA Madrid Education Arches*. 2014. URL : <http://wiki.ivoa.net/internal/IVOA/InterOpMay2014Edu/IVOA-Madrid-Education-Arches.pdf>.
- [4] Observatoire Astronomique de STRASBOURG. *Aladin Sky Atlas*. 2014. URL : <http://aladin.u-strasbg.fr/>.

Glossaire

Ad-hoc : réseau Wi-Fi local, défini directement entre plusieurs appareils sans autres infrastructures (à l'inverse de internet).

Adresse IP : numéro d'identification de 4 nombres (entre 0 et 255) adressé à tout appareil connecté à un réseau.

Android : système d'exploitation utilisé sur la plupart des smartphones et tablettes (hors produits Apple).

API : Application Programming Interface, ensemble de classes fournissant les outils nécessaires au programmeur.

Applet Java : logiciel Java qui s'exécute dans la fenêtre d'un navigateur web.

Beta : version d'un logiciel lorsque celui-ci est encore en cours de développement, juste avant la sortie de la première version.

ESA : European Space Agency, ou Agence spatiale européenne.

Groupe Locale : ensemble d'une quarantaine de galaxies auquel appartient la Voie Lactée.

IDE : Integrated Development Environment, logiciel regroupant des outils permettant de faciliter la programmation (éditeur de texte, compilateur, ...).

JavaScript : langage de programmation de scripts utilisé principalement dans les pages web interactives.

JSON : format de données textuelles regroupant les données sous formes de listes et d'ensembles (symboles [] et {}).

Responsive : se dit d'un design qui s'adapte et reste identique sur différentes tailles d'écrans ou de fenêtres.

SDK Android : Software Development Kit, kit de développement contenant des nombreux outils facilitant le développement Android.

Serveur HTTP : serveur utilisant des requêtes HTTP (HyperText Transfer Protocol), protocole développé pour le web.

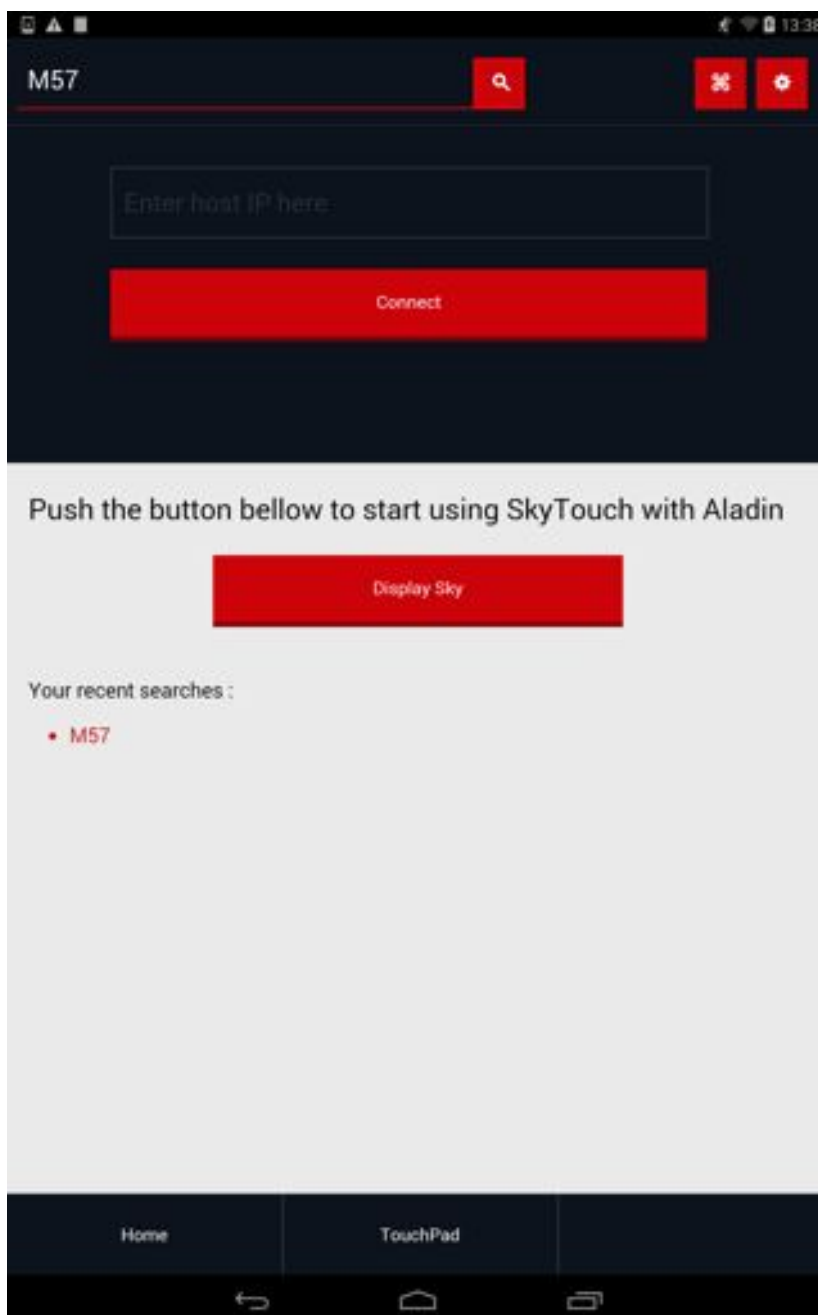
URL : Uniform Resource Locator, chaînes de caractères utiles pour adresser des ressources sur internet (adresse web <http://www.google.fr> par exemple).

XMM-Newton : satellite développé par l'ESA et lancé en 1999 dont le but est l'observation en rayons X.

Annexes

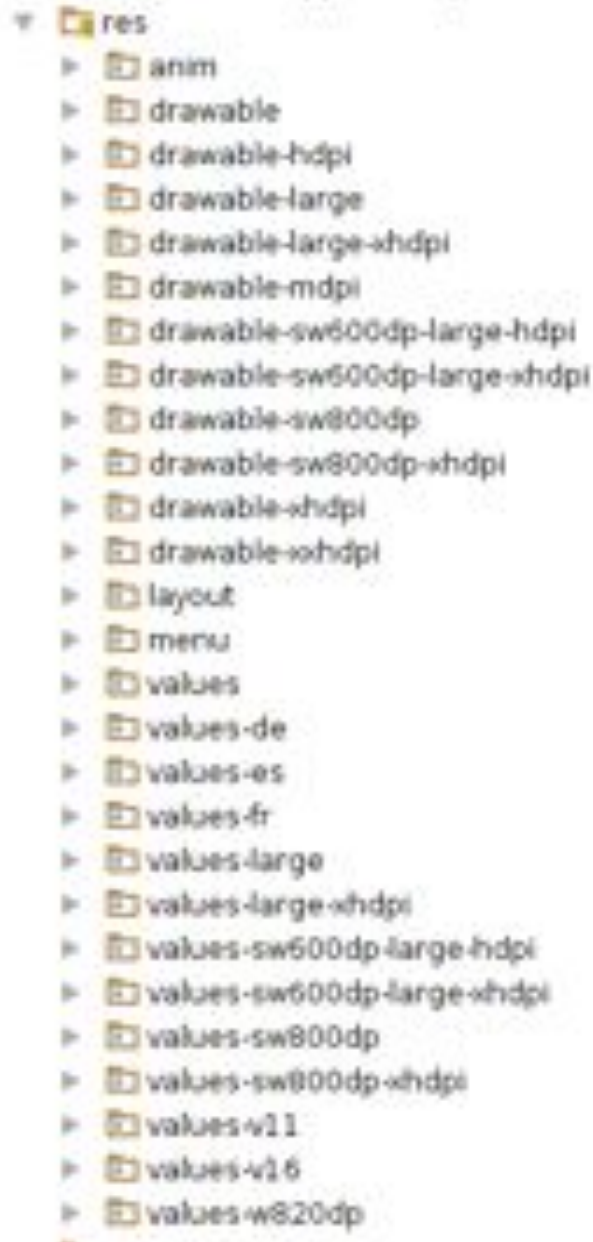
Annexe A

Aperçu de l'application *SkyTouch*



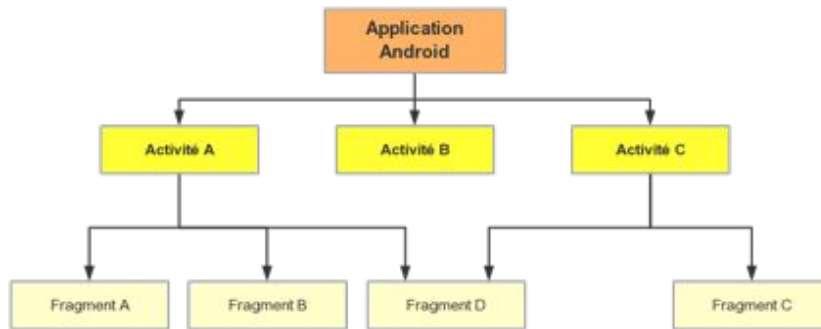
Annexe B

Le dossier *res* et ses sous-dossiers



Annexe C

Exemple d'organisation d'une application Android



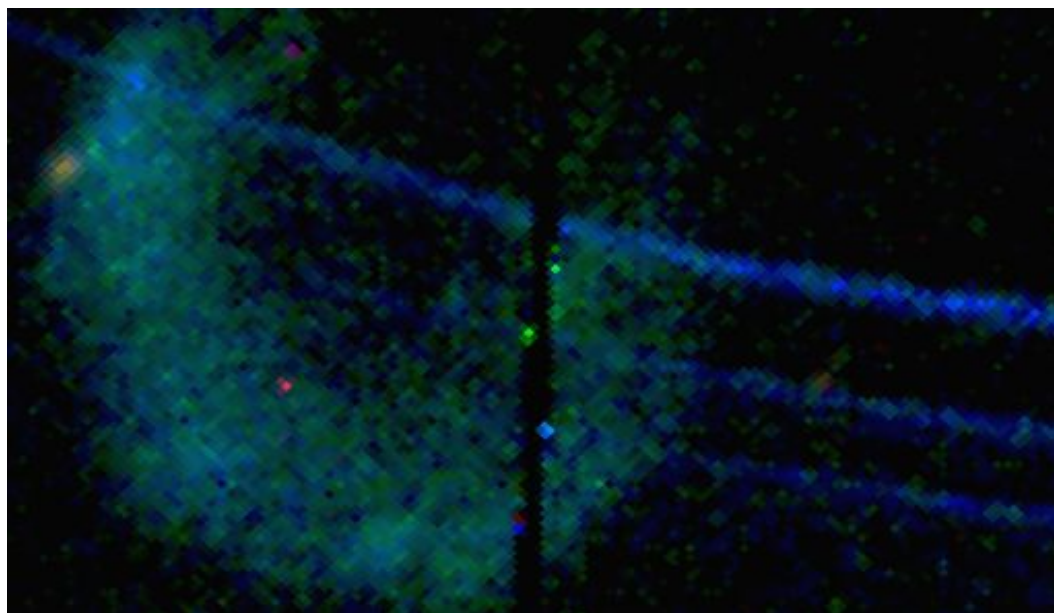
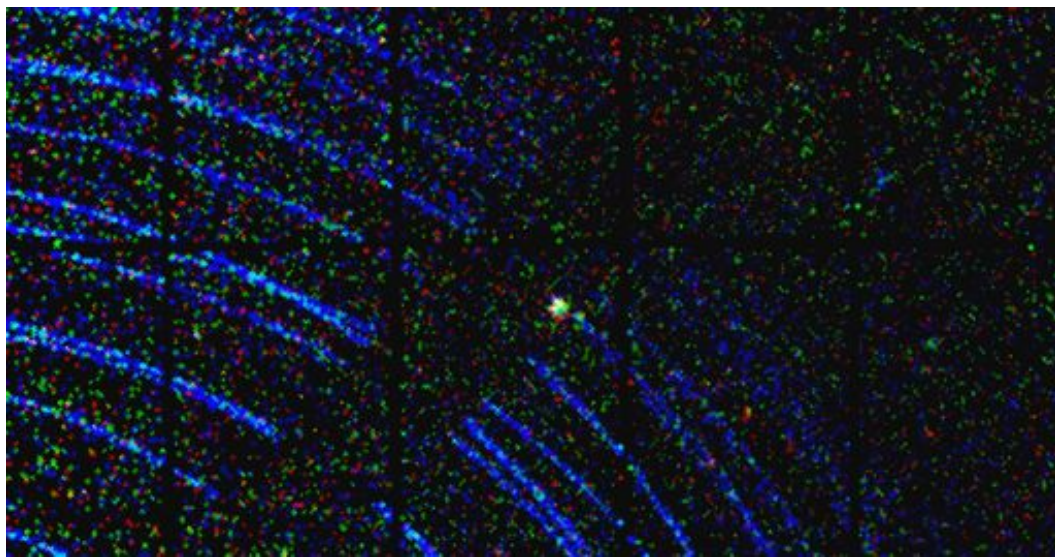
Annexe D

Exemple d'utilisation d'ADB shell

```
ehkirch@cds-stage-ms3:~$  
ehkirch@cds-stage-ms3:~$ adb shell  
shell@manta:/ $ wm size 1280x800  
shell@manta:/ $ wm density 160  
shell@manta:/ $  
shell@manta:/ $  
shell@manta:/ $ wm size reset  
shell@manta:/ $ wm density reset
```

Annexe E

Autres fonds d'écran retenus au départ



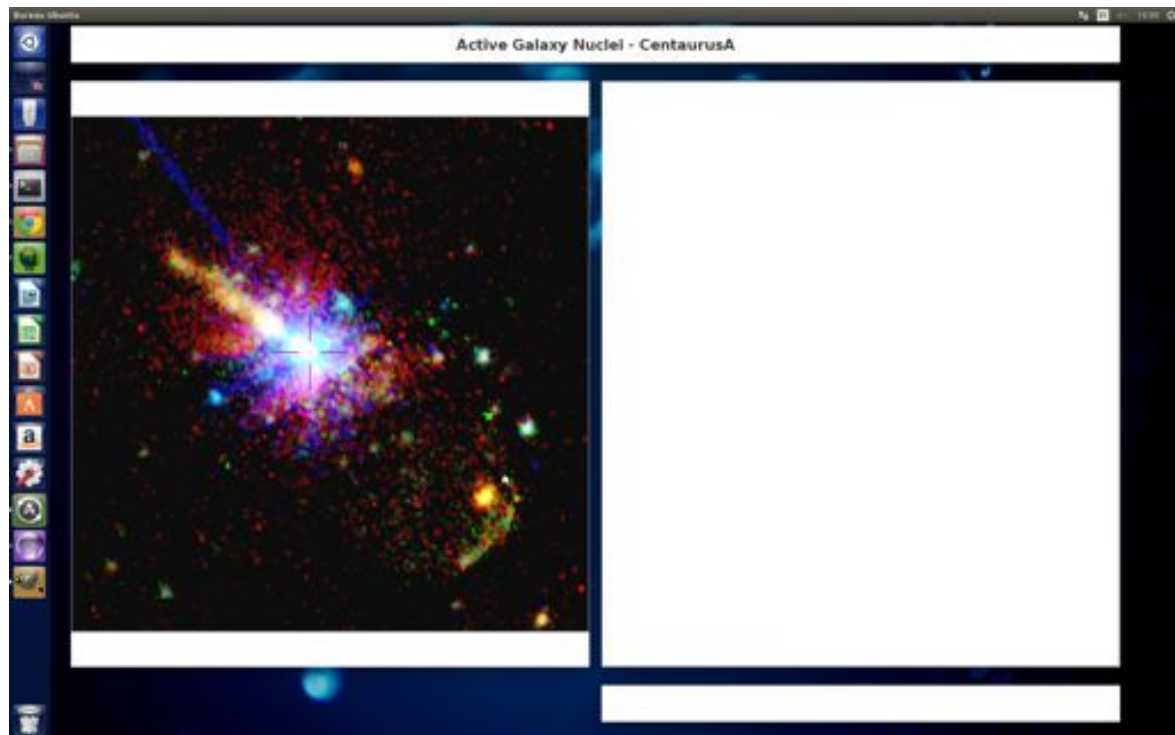
Annexe F

Aperçu du fichier *summary.json* contenant les données chargées

```
{
  "classes": [
    {
      "name": "Galaxy Cluster",
      "dirname": "Galaxy_Cluster",
      "objects": [
        {
          "name": "Abell426",
          "dirname": "Abell426",
          "image": "image.png"
        }
      ]
    },
    {
      "name": "Active Galaxy Nuclei",
      "dirname": "Active_Galaxy_Nuclei",
      "objects": [
        {
          "name": "CentaurusA",
          "dirname": "CentaurusA",
          "image": "image.png"
        }
      ]
    }
  ]
}
```


Annexe G

Fenêtres affichés par l'application sur l'écran de l'ordinateur




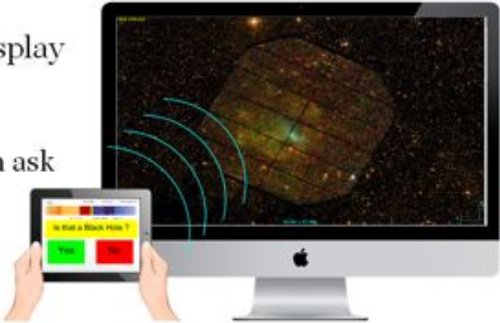
Annexe H

Slides de la présentation de l'application lors d'une conférence de l'IVOA




Making the data corpus available through a demo booth

- A wide screen controlled from a mobile device
- The user/player can display objects in various wavelengths, he/she can ask for *extra data*



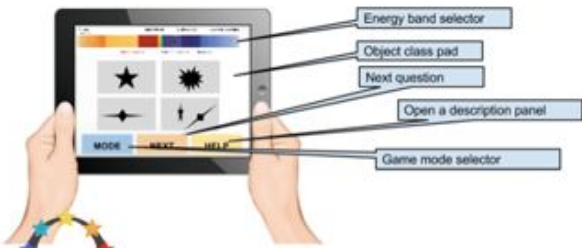


IVOA Madrid, 19-23 May 2014

5



Extra data



IVOA Madrid, 19-23 May 2014

6



Arches Walker

- The mobile device communicates with the display station with a had hoc Wifi
 - The display station hosts a local database with the whole data corpus, it can work without network
 - It runs a Java application including the Aladin sky atlas and all viewer (images, html....)
 - This application is commanded by Java Servlets
-
- Work in collaboration with students
 - Android development
 - Data corpus gathering



Will be delivered as a kit for interested partners

IVOA Madrid, 19-23 May 2014

7



Quiz

Quiz



IVOA Madrid, 19-23 May 2014

8

Arches Walker Discovery

Galaxy

Learn more

Star

Class_2

Class_3

Class_4

Andromeda

Bode's I

NGC 2418

UGC 3

More about NGC 2418

Characteristic values

Display RGB

Color

Discovery

Arches Walker Discovery

Galaxy

Learn more

Star

Class_2

Class_3

Class_4

Andromeda

Bode's I

NGC 2418

UGC 3

More about NGC 2418

Characteristic values

Display RGB

Color

Discovery

IVOA Madrid, 19-23 May 2014

9



Remarks

- The same concept can be easily reused for other domains of the astronomy
- A small investment (a desktop + screen and/or projector + an internet tablet)
- Probably more fun than just a touchscreen, larger field of view !!



