

Rapport de Stage

Année Universitaire 2011-2012

SkyObjects

Développement d'un Name Resolver sous Android et prototypage en HTML5

Etudiant : Romain Houpin S4-OS

Maître de stage : André Schaaff

Parrain de stage : Samuel Cruz-Lara

Centre de Données Astronomiques de Strasbourg
Observatoire Astronomique de Strasbourg
11, rue de l'université
67000 Strasbourg

IUT Nancy Charlemagne
Université de Lorraine
2 ter Boulevard Charlemagne
54052 Nancy Cedex



Rapport de Stage

Année Universitaire 2011-2012

SkyObjects

Développement d'un Name Resolver sous Android et prototypage en HTML5

Etudiant : Romain Houpin S4-OS

Maître de stage : André Schaaff

Parrain de stage : Samuel Cruz-Lara

Centre de Données Astronomiques de Strasbourg
Observatoire Astronomique de Strasbourg
11, rue de l'université
67000 Strasbourg

IUT Nancy Charlemagne
Université de Lorraine
2 ter Boulevard Charlemagne
54052 Nancy Cedex



Remerciements :

Je remercie :

- André SCHAAFF, ingénieur de recherche au CDS de m'avoir pris au sein de l'équipe durant ce stage et de m'avoir aussi bien encadré, que ce soit pour la mission qu'il m'avait confiée ou pour le rapport et la soutenance de stage.
- Gilles LANDAIS, travaillant au CDS, pour m'avoir accueilli les premiers jours en attendant le retour de M. André Schaaff
- Anaïs OBERTO, Patricia VANNIER, Thomas BOCH, Laurent MICHEL, Gilles LANDAIS tous les cinq travaillant à l'observatoire et qui m'ont fait découvrir tout comme M. André SCHAAFF, leur travail et les différents services de l'observatoire et du CDS.
- Samuel CRUZ-LARA, enseignant de l'IUT, qui fut le lien entre l'IUT et l'entreprise en étant mon parrain de stage, m'a aidé lors de la rédaction du rapport et qui m'a rendu visite au CDS.
- Julien SCHEFFMAN, stagiaire venant d'un autre IUT avec qui j'ai réalisé une partie de la conception de l'application et la réalisation du code en HTML5.
- Toutes les personnes travaillant à l'observatoire pour leur bon accueil, leur sympathie et les cafés des vendredis matins.
- Bernard MANGEOL, enseignant à l'IUT, qui connaissant mes loisirs pour avoir déjà beaucoup discuté avec lui, m'a recommandé de prendre contact avec M. SCHAAF pour mon stage.

Table des matières

Remerciements :	2
Introduction	5
Chapitre 1 :	6
Présentation de l'entreprise	6
I / L'Observatoire Astronomique de Strasbourg	6
1.1) Présentation générale	6
1.2) Les équipes de recherche	7
II / Le CDS	9
2.1) Présentation :	9
2.2) Les services du CDS :	10
III/ L'observatoire Virtuel et l'IVOA	15
3.1) Présentation.....	15
3.2) Le CDS au sein de l'IVOA	16
Chapitre 2 :	17
L'Existant	17
I / Sesame	17
II / Les résolveurs	19
III / Le « Name Resolver » Indien	20
Chapitre 3 :	23
Le Stage	23
I / Le sujet du stage	23
1.1) La demande	23
1.2) Le cahier des charges de SkyObjects	24
II / Les premiers-pas	25
2.1) Découverte des locaux et du personnel	25
2.2) Découverte des outils	26
III / L'application Android : SkyObjects	28
3.1) La base de données	28
3.2) L'interface utilisateur	33

3.3) Le moteur de recherche.....	36
3.4) Le moteur de mise à jour.....	37
3.5) Les notes	38
3.6) Les logs	38
IV / L' « IVOA Interop »	39
4.1) Présentation.....	39
4.2) Corrections logicielles.....	39
4.3) Améliorations logicielles	39
4.4) Améliorations graphiques	40
V / Le prototype en HTML5.....	40
5.1) La base de données	41
5.2) L'interface utilisateur.....	42
5.3) PhoneGap.....	42
VI/ Comparaison	42
Chapitre 4 :	45
Perspectives.....	45
I / Le futur de SkyObjects	45
1.1) Le pointage d'objet	45
1.2) L'ajout de catalogues	46
1.3) Le mode éducation.....	46
1.4) Publication	47
II / Mon futur à l'Observatoire.....	47
Conclusion.....	49
Références bibliographiques	51
Annexes	52
A.1) Diagramme de Gant	53
A.2) MCD.....	54
A.3) Schéma du moteur de recherche	55
A.4) Captures d'écrans de SkyObjects.....	56

Introduction

Dans le cadre de la 2^{ème} année du DUT informatique à l'IUT Nancy-Charlemagne, je devais effectuer un stage de 10 semaines. Je voulais profiter de ce stage pour appliquer mes connaissances mais surtout en apprendre de nouvelles. De plus, j'espérais être pris dans un lieu très intéressant et enrichissant pour moi. Habitué à discuter avec M. Mangeol, il savait que l'une de mes passions était l'astronomie. Vers la mi-janvier, j'ai eu l'occasion d'avoir une conversation très intéressante avec lui. En effet, à la fin d'un cours, il a désiré me parler de l'Observatoire astronomique de Strasbourg et plus précisément de prendre contact avec M. André Schaaff, un ami à lui, qui proposait en général des stages à l'IUT. Dès le lendemain après-midi, j'ai pris contact avec M. André Schaaff et dans la soirée, un rendez-vous avec lui était déjà convenu. Ce rendez-vous fut très enrichissant pour moi (j'ai pu notamment visiter l'observatoire) et se conclut parfaitement bien. Toujours lors de cette entretien qui a duré au final plus de 2H30, le sujet exact du stage fut établi : « Etudes et prototypages pour des services Astronomiques ». Derrière ce sujet assez flou au premier abord, se cachait en réalité le développement d'une application Android ainsi que la réalisation d'un prototype de cette application en HTML5 afin de les comparer sur différents critères. Le fait d'être pris était déjà pour moi une excellente nouvelle car j'allais travailler dans un lieu qui collait parfaitement à ma passion de l'astronomie mais en plus, le sujet du stage, concernant la programmation sur Android, correspondait parfaitement à l'apprentissage que je voulais faire lors des vacances après le stage. C'est donc avec un grand bonheur que j'ai quitté l'entretien et l'observatoire. Désormais que le stage est terminé, nous allons voir de manière thématique le déroulement de mon stage.

Tout d'abord, dans une première partie, je vais présenter l'Observatoire astronomique de Strasbourg en insistant sur le centre de données de Strasbourg (CDS). Ensuite, dans un deuxième temps, je ferai une étude de l'existant, autrement dit de tout ce qui existe déjà et qui me fut utile dans le cadre de mon stage. Ensuite, nous verrons le déroulement même du stage de manière thématique et chronologique. Enfin, dans une dernière partie, nous verrons les perspectives de ce stage en parlant du futur de ce que j'ai développé mais également de mon propre futur.

Commençons donc tout de suite avec le chapitre un et la présentation de l'entreprise.

Chapitre 1 :

Présentation de l'entreprise

I / L'Observatoire Astronomique de Strasbourg

1.1) Présentation générale

L'Observatoire astronomique de Strasbourg est un Observatoire des Sciences de l'Univers (OSU). Il est une Unité Mixte de Recherche (UMR 7550) entre l'Université de Strasbourg (l'observatoire est d'ailleurs situé au cœur de l'université) et le CNRS (Centre National de la Recherche Scientifique).

L'observatoire a été construit en 1881 et possède en son sein la 3^{ème} plus grande lunette astronomique de France de par son diamètre. L'observatoire accueille également le planétarium qui depuis 2008 n'est plus sous sa responsabilité. On peut trouver également à l'Observatoire des ouvrages anciens et un patrimoine d'anciens instruments.

Pour réaliser ses missions, l'observatoire possède deux équipes de recherche que nous présenterons plus tard et deux Services d'Observation de l'Institut National des Sciences de l'Univers (INSU) : le Survey Science Centre d'XMM-Newton (SSC-XMM) et le Centre de Données astronomiques de Strasbourg (CDS) dans lequel j'ai travaillé.

L'observatoire est au cœur du dispositif national mis en œuvre par l'INSU et doit assurer plusieurs missions comme l'acquisition de données d'observation, l'élaboration d'outils théoriques ainsi que le développement et l'exploitation de moyens appropriés. En parallèle



de ses missions et pour remplir ses conditions d'observatoire, il doit fournir des services liés à son activité, assurer la formation des étudiants et des personnels de recherche, assurer la diffusion des connaissances et il est chargé d'activités de coopération internationale.

1.2) Les équipes de recherche

L'observatoire possède principalement deux équipes de recherche en plus du CDS (qui lui cumule le service d'observation avec la recherche). Nous allons donc voir dans cette partie les deux équipes de recherche à part entière qui sont « l'équipe Hautes Energies » et « l'équipe Galaxies ».

L'équipe Galaxies :

L'équipe « Galaxies » étudie de manière précise comment se passe la naissance des galaxies et comment elles évoluent dans le temps avec leurs populations stellaires. L'équipe étudie également notre galaxie. Pour mener à bien cela, elle utilise les populations stellaires connues, la dynamique des étoiles et la matière noire (catégorie de matière hypothétique jusqu'à présent non détectée). Elle est également très utile et impliquée dans les préparatifs de la mission satellitaire astrométrique Gaia de l'Agence Spatiale Européenne, prévue d'être lancée en 2012. L'équipe est également impliquée dans le relevé RAVE : un grand relevé cinématique.

De manière plus précise, l'équipe étudie les caractéristiques et les propriétés physiques de notre galaxie ainsi que les galaxies voisines. La population stellaire de ces galaxies est analysée, après avoir été observée, de manière à pouvoir étudier la dynamique gravitationnelle qui régit les mouvements du gaz et des étoiles. Ces études permettent ensuite de lier l'histoire avec l'information ce qui permet de connaître les moments clés de la vie des galaxies. Bien évidemment, il ne s'agit là que d'une explication très brève de ce que fait l'équipe dans ce domaine.

L'équipe possède également un autre domaine de recherche qui se porte sur les amas stellaires (concentration locale d'étoiles d'origine commune et liées entre elles par la gravitation). Ce domaine de recherche consiste, entre autre, à construire numériquement des amas en fusionnant toutes les connaissances sur la physique des étoiles et sur l'évolution gravitationnelle des amas. Cela est très utile pour comprendre la formation des galaxies car, pour faire simple, on pourrait comparer la galaxie à une maison en briques et dans ce cas, les amas stellaires seraient tout simplement les briques.

L'équipe possède également d'autres thèmes de recherche visant dans tous les cas à en savoir plus, de près ou de loin, sur les galaxies.

L'équipe Hautes Énergies :

L'équipe « Hautes Énergies » s'intéresse, entre autre, aux sources galactiques et extragalactiques émettrices en rayons X mais également aux objets compacts (étoiles à neutrons, naines blanches, etc.) et aux noyaux actifs de galaxies. L'équipe participe très activement depuis 1996 au SCC (Survey Science Center) qui est un consortium international de laboratoires sélectionnés par l'ESA et labellisés par l'INSU comme Service d'Observation. Le SCC-XMM est en charge de fournir des catalogues de la manière la plus complète possible de sources X.

Pour parvenir à cela, les chercheurs de l'équipe Hautes Energies utilisent de nombreux instruments, tous très performants et capables d'observer les rayons X. On trouve donc les satellites Chandra et XMM-Newton, ainsi que des télescopes au sol. Ces instruments ne sont pas français mais ils sont mis à disposition des astronomes français. Il est important de noter que de tels instruments coûtent extrêmement cher et qu'il est donc indispensable pour toute la communauté astronomique de mutualiser les instruments.

Coté scientifique pur, l'équipe s'occupe des problèmes d'astrophysiques de population comme la caractérisation de la densité ou encore les échelles de hauteurs des sources X ainsi de l'évolution des sources X qui peuvent provenir par exemple de couronnes stellaires. Enfin, elle s'occupe également de la physique des disques d'accrétion et des jets.

Pour résumer, l'équipe s'occupe de tout ce qui touche aux rayons X c'est-à-dire aux Hautes Energies et de l'évolution de ses sources à rayons X.

II / Le CDS

2.1) Présentation :

Le Centre de Données astronomiques de Strasbourg (CDS) a été créé en 1972. La mission principale du CDS est de gérer un ensemble de données astronomiques qu'il faut recueillir, unifier et homogénéiser, conserver et enfin distribuer de manière à ce que la communauté scientifique internationale puisse en bénéficier.

L'équipe du CDS est composée d'une trentaine de personnes environ réparties de manière équivalente entre trois métiers : astronomes, documentalistes et bien évidemment informaticiens.

Le CDS possède ses propres infrastructures, il possède de nombreux postes de travail (sous Ubuntu dans la plupart des cas) ainsi que d'une salle de serveurs, ...

Il est également à noter que le CDS est une référence mondiale dans le domaine astronomique et que depuis fin 2009, son utilisation est en augmentation continue. Grâce à sa notoriété mais surtout sa très grande base de données pour le référencement d'objets astronomique, le CDS fait partie des TGIR (les Très Grandes Infrastructures de Recherche). C'est le Ministère de l'Enseignement Supérieur et de la Recherche qui définit la liste des TGIR : infrastructures reconnues comme les plus utiles à la communauté scientifique. Ces TGIR bénéficient d'un financement complémentaire et du point de vue du CDS, cela permet

notamment de financer quelques CDD et d'acheter de l'équipement.

On peut enfin noter que le CDS est dans une situation un peu paradoxale car les TGIR sont en général des infrastructures de taille beaucoup plus importante (des dizaines et des centaines de fois le budget du CDS) mais le ministère de la recherche a reconnu l'utilité du centre de données en l'intégrant à cette liste.

2.2) Les services du CDS :

Le CDS possède de nombreux services, mais, de manière générale, trois grands services représentent le CDS : Simbad, Vizier et Aladin. Ils ont chacun leur propre utilité mais sont complémentaires entre eux.

Simbad :

Simbad est une base de données regroupant des informations sur un nombre remarquable d'objets astronomiques (un objet astronomique peut être une galaxie, une étoile, un trou noir, un nuage de gaz, etc.). Cette base



de données fut créée il y a près de 40 ans et contient aujourd'hui sept millions d'objets. Pour chaque objet, Simbad est capable de fournir de nombreuses caractéristiques physiques comme la position de l'objet, sa magnitude, sa parallaxe, etc. Ces informations proviennent toutes de catalogues recensant et décrivant les objets astronomiques. Afin de faciliter les choses, Simbad permet de rechercher un objet dans sa base et affiche les informations qu'il connaît de manière rangée via une interface web agrémentée parfois d'une petite photo représentant l'objet (voir figure 1). La recherche d'un objet sur Simbad peut se faire de manière multiple. Par exemple, on peut la faire en donnant un des noms de l'objet recherché ou sa position ou encore faire une recherche par critères, etc.

SIMBAD basic query result

Object query : Andromeda C.D.S. - SIMBAD4 rel 1.194 - 2012.06.07CEST21:17:02

Available data : [Basic data](#) • [Identifiers](#) • [Plot & images](#) • [Bibliography](#) • [Measurements](#) • [External archives](#) • [Notes](#) • [Annotations](#)

Basic data :
M 31 -- LINER-type Active Galaxy Nucleus query around with radius arcmin

Other object types: LIN () , G (LEDA, 2MASX, MCG, UGC, UZC, Z, [M98c]) , AGN ([VV2000c], [VV2003c], [VV98c]) , Rad (2C, DA, [DGW65]) , IR (IRAS, IRC, RAFGL) , GiC (GIN) , GiG (K79) , QSO ([VV2006]) , X (XSS)

ICRS coord. (ep=J2000) : 00 42 44.330 +41 16 07.50 (Infrared) [~ ~ ~] B [2006AJ...131.1163S](#)

FK5 coord. (ep=J2000 eq=2000) : 00 42 44.330 +41 16 07.50 (Infrared) [~ ~ ~] B [2006AJ...131.1163S](#)

FK4 coord. (ep=B1950 eq=1950) : 00 40 00.09 +40 59 41.7 (Infrared) [~ ~ ~] B [2006AJ...131.1163S](#)

Gal coord. (ep=J2000) : 121.1743 -21.5733 (Infrared) [~ ~ ~] B [2006AJ...131.1163S](#)

Radial velocity / Redshift / cz : V (km/s) -301 [7] / z (~) -0.001004 [0.000023] / cz -300.99 [6.90] (~) D [2002LEDA.....0P](#)

Morphological type: Sb D ~

Angular size (arcmin): 44.673 25.062 45 (~) (IR) C [2006AJ...131.1163S](#)

Fluxes (6) :
 U 4.86 [0.03] D [2007ApJS..173..185G](#)
 B 4.36 [0.02] D [2007ApJS..173..185G](#)
 V 3.44 [0.03] D [2007ApJS..173..185G](#)
 J 2.094 [0.016] C [2006AJ...131.1163S](#)
 H 1.283 [0.017] C [2006AJ...131.1163S](#)
 K 0.984 [0.017] C [2006AJ...131.1163S](#)

essential notes : • See GALEX UV data in [GALEX data](#)

Figure 1 : Capture d'écran du service Simbad affichant les détails de l'objet M31 (Andromède)

Il est également important de préciser que Simbad est capable de fournir tous les identifiants d'un objet. En effet, un même objet astronomique, en fonction des études dont il a fait l'objet et des pays, peut avoir différents noms. Par exemple, la galaxie "ANDROMEDA" (Andromède en français) est également connue sous le nom de "M31" ou encore "IRAS F00400+4059". Andromède possède en tout 28 identifiants mais il peut y en avoir beaucoup plus (la figure 2 démontre cet exemple). Ainsi, c'est plus de dix-sept millions d'identifiants qui sont stockés dans Simbad sachant que chaque identifiant peut être relié à un objet. De manière à faciliter les choses pour relier les identifiants entre eux, Simbad mémorise pour chaque objet un "oid". Ainsi, chaque identifiant renvoie vers un oid unique et pour un oid donné (donc un objet donné), Simbad peut retrouver tous les identifiants et ces derniers peuvent parfois se compter en dizaines.

Identifiants (28) :

M 31	K79 1C	NAME AND NEBULA	Z 0040.0+4100
2C 56	LEDA 2557	NGC 224	[DGW65] 4
DA 21	2MASX J00424433+4116074	RAPGL 104	[M98e] 004000.1+405943
GIN 801	MCG+07-02-016	UGC 454	[VV2000e] J004244.3+411610
IRAS F00400+4059	NAME ANDROMEDA NEBULA	UZC J004244.3+411608	[VV2003e] J004244.3+411610
IRAS 00400+4059	NAME ANDROMEDA	XSS J00425+4102	[VV2006] J004244.3+411610
IRC +40013	NAME ANDROMEDA Galaxy	Z 535-17	[VV98c] J004245.1+411622

Figure 2 : Capture d'écran de l'affichage des identifiants via Simbad de M31

De part son efficacité et son nombre important d'objets référencés de manière très complète, Simbad sont devenue la base de données de référence pour la nomenclature et la bibliographie des objets astronomiques. Aujourd'hui, Simbad traite des centaines de milliers de requêtes par jour venant de très nombreux pays et a su devenir une référence mondiale sans équivalent.

VizieR :

Nous avons vu plus tôt que les objets référencés dans Simbad provenaient de catalogues astronomiques. Ces catalogues sont très nombreux (plus de 10000) et sont en général constitués de nombreuses tables (ces tables regroupent des informations sur de nombreux objets). Toutes ces tables et catalogues ont été stockés par le CDS (plus précisément par les documentalistes) dans ce qui s'appelle aujourd'hui VizieR. VizieR permet donc d'accéder aux objets astronomiques via des catalogues de manière homogène alors que les données des catalogues sont à la base hétérogènes. On ne recherche plus un objet en particulier mais des catalogues contenant un certain objet (cf. figure 3). VizieR est également un outil puissant qui permet donc de consulter les tables des catalogues mais aussi de les croiser entre elles tout en respectant, si la personne le souhaite, des critères précis (recherche par mission, par longueur d'ondes, par date ...). Après ces calculs, VizieR affiche les résultats mais est également capable de les exporter dans de multiples formats (XML, CSV, TSV, etc.).



VizieR est donc un outil très puissant pour la recherche sur des critères.

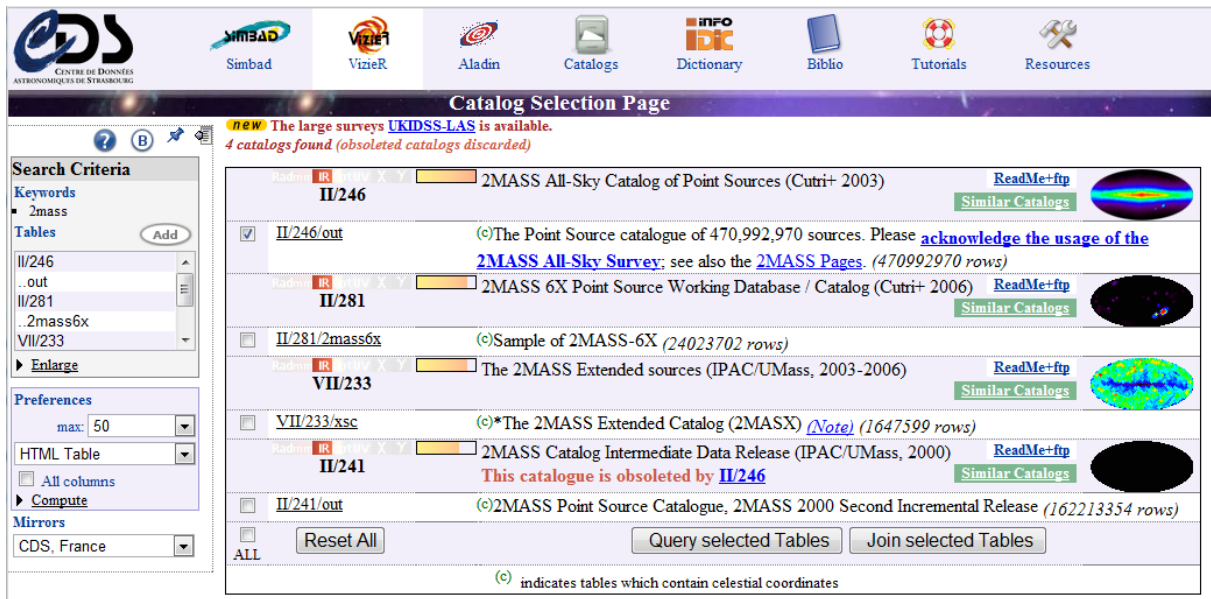


Figure 3 : Capture d'écran du service VizieR après avoir recherché le catalogue 2MASS

Aladin :

Aladin est le dernier né parmi les 3 services (1997) et est extrêmement puissant. Il permet de visualiser un atlas du ciel de manière très interactive. De par sa nature, ce service est encore en développement par plusieurs personnes du CDS (Simbad et VizieR évolue également encore mais de manière moins intense).



Avant de poursuivre, il est important de savoir qu'il y a en réalité deux Aladin. En effet, on trouve la base de données Aladin qui est une base contenant une quantité très importante d'images, et, on trouve Aladin, un logiciel (disponible également en applet) écrit en Java. Ce dernier s'appuie sur la base de données Aladin ainsi que d'autres pour former un atlas du ciel (cf. figure 4). Cet atlas, comme nous le disions précédemment, se veut très interactif. Ainsi, il est possible de se positionner n'importe où dans le ciel, de zoomer, d'y superposer des objets ou d'anciennes images pour voir l'évolution dans le temps des objets

astronomiques (pour voir notamment s'ils se déplacent) et de multiples autres options comme la possibilité de sélectionner une zone du ciel pour y connaître les objets présents et des informations dessus, la comparaison d'images dans différentes longueurs d'ondes, etc. Aladin utilise pour produire ses sphères la technologie HEALPIX développée par la NASA (cf. figure 5). Cette technologie (disponible sous la forme d'une API) permet entre-autre de représenter une sphère grâce à un ensemble d'images en évitant le plus possible les déformations (contrairement à une sphère classique où les déformations sont grandes au niveau des pôles). Cette technologie est également utilisée dans une application Android dénommée SkySurveys, développée par le CDS par un ancien stagiaire venant de l'IUT. Cette application, permet en utilisant Healpix et OpenGL d'afficher l'atlas du ciel sur les téléphones Android.

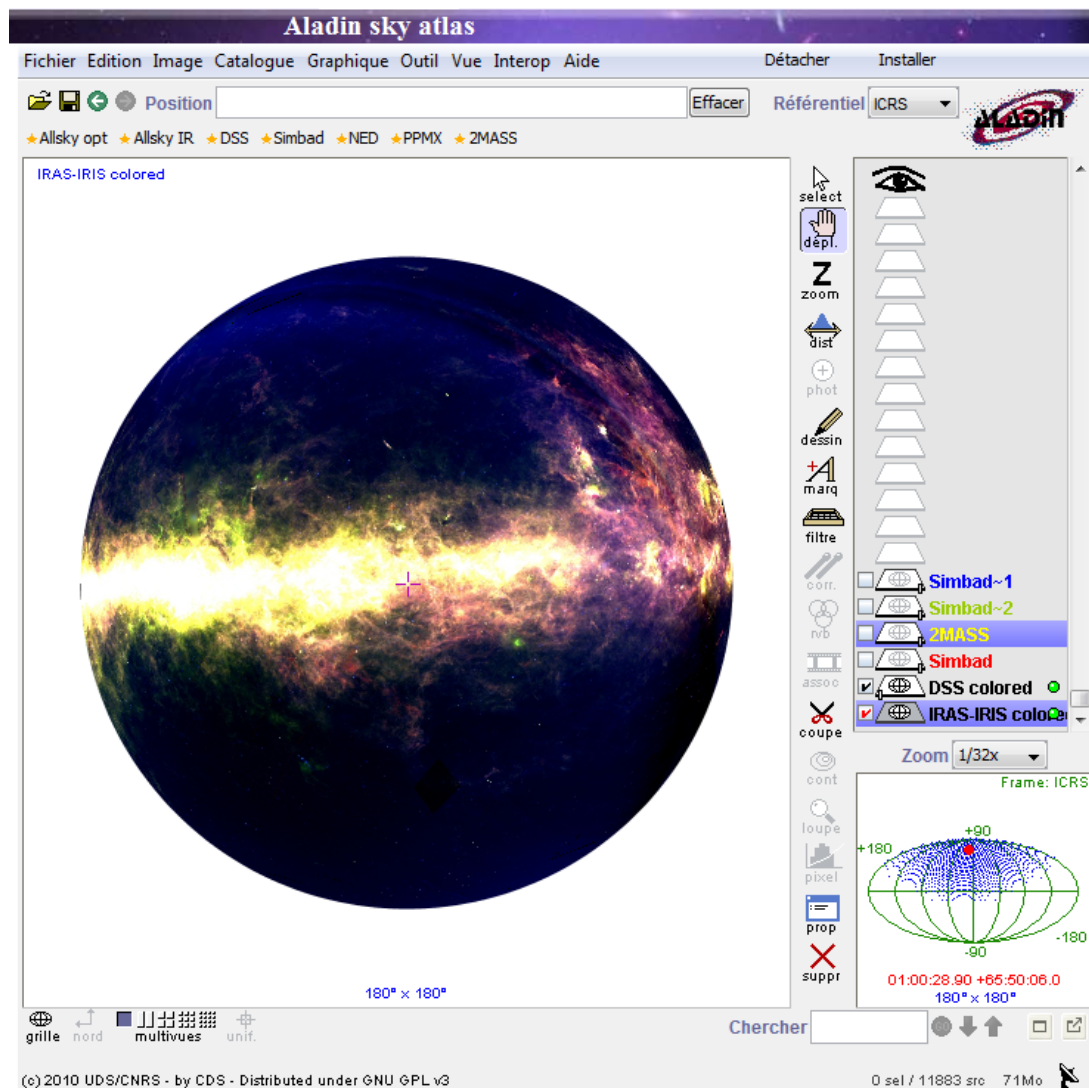


Figure 4 : Capture d'écran du logiciel Aladin affichant dans ce cas l'atlas du ciel en infrarouge

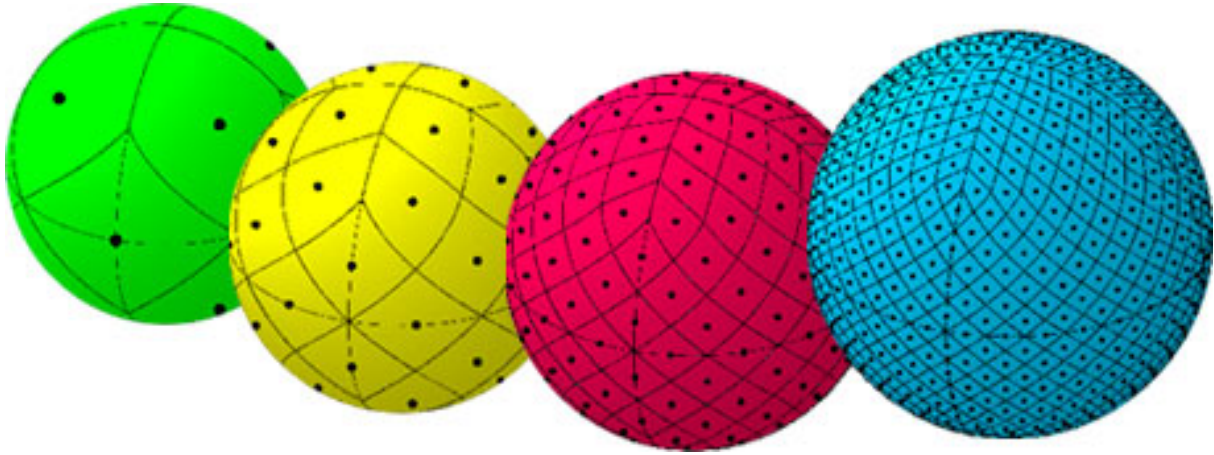


Figure 5 : Représentation de sphères HEALPix

En bref, Aladin est un service très puissant qui contrairement aux deux précédents, a une approche plus imagée des objets astronomiques.

III/ L'observatoire Virtuel et l'IVOA

3.1) Présentation

Un observatoire virtuel est un ensemble de données distribuées et accessibles de partout avec des logiciels puissants ayant un fort potentiel en calculs. A l'inverse, un vrai observatoire dispose, lui, de ses instruments (télescopes, radiotélescopes, etc.) qui permettent de réaliser des observations ou des relevés du ciel à un instant donné et dans une direction donnée (suivant la localisation de l'instrument). L'accès à de multiples sources de données nécessite la définition de standards d'accès à ces données. C'est pour cette raison qu'est né l'IVOA.

L'IVOA (International Virtual Observatory Alliance) est une organisation internationale ayant pour objectif de définir des standards pour faciliter les échanges internationaux et les collaborations afin de fournir des services (utilisant des archives astronomiques) utilisables par toute la communauté mondiale sans adaptation nécessaire. Cette organisation crée donc un contexte d'Interopérabilité homogénéisant l'accès aux données et l'utilisation des outils et des services en général afin de rendre leurs données

compatibles.

L'IVOA fut créé en juin 2002 et compte désormais de nombreux membres comme euro-VO, Aus-VO, China-VO, VO France, etc. La volonté de standardiser les données est une opération complexe mais nécessaire pour pouvoir avancer de nos jours en Astronomie. De nombreux projets sont en cours et l'ESA ainsi que l'ESO participent de manière active à cette internationalisation et standardisation des échanges. L'IVOA compte désormais plus de 20 pays ou organisations membres.

Enfin, l'IVOA organise deux fois par an des réunions d'interopérabilité. Ces réunions durent une semaine et accueillent toutes les équipes de travail. Durant cette semaine, les différents participants se rencontrent pour discuter des différents projets à travers de très nombreux exposés. La dernière réunion a eu lieu durant le stage (du 21 au 25 mai 2012) à Urbana Champaign aux Etats-Unis.

Une fois que les standards sont validés par l'IVOA, il tient à la responsabilité des différents acteurs de les respecter que ça soit au niveau de ses outils ou de ses données.

3.2) Le CDS au sein de l'IVOA

Le CDS participe à l'Action Concertée Observatoire Virtuel France, autrement dit de l'observatoire virtuel Français et est depuis plus de 40 ans, reconnu comme impliqué dans les collaborations internationales. Les astronomes du monde entier connaissent le CDS et il joue donc un rôle majeur dans le cadre de l'IVOA. Ainsi, de manière très active, le CDS fait partie de la plupart des groupes de travail de l'IVOA (un groupe de travail est constitué de plusieurs partenaires qui discutent des standards à mettre en place).





De plus, un grand rôle est donné au CDS car il est présent dans de nombreux groupes de travail, mais en plus de cela, il en dirige ou codirige certains. On trouve ainsi au CDS M. André Schaaff pour le groupe de travail "Grid and Web Services", Sébastien Derrière pour le groupe de travail sémantique chargé de la mise en place des Unified Column Descriptors (autrement dit une manière unique de décrire les différentes colonnes des tables de données astronomique (contenu dans VizierR par exemple)) et enfin François Ochsenbein pour le groupe de travail VOTable.

Chapitre 2 :

L'Existant

I / Sesame

Sesame est un autre service développé, hébergé et maintenu à jour par l'équipe du CDS, il s'agit d'un « Name Resolver », (en Français, un résolveur de noms). Ce service est capable d'interroger plusieurs bases de données (qui sont appelées les résolveurs) à partir d'un nom d'objet astronomique (en dehors des objets du système solaire) et affiche ensuite les paramètres fondamentaux (type d'objet, J2000 position). Il permet également via l'interface web d'accéder à des liens renvoyant vers l'objet pour chaque base de données. Sesame utilise trois bases de données : Simbad, NED et VizieR. La figure 6 nous montre un exemple de fonctionnement de Sesame avec une recherche portée sur M31



*The **Sesame** Service queries several databases from the name of an astronomical object (outside the Solar System bodies), and displays some fundamental parameters (type of object, J2000 position). The databases queried are [Simbad](#), [NED](#), and [VizieR](#). (For more explanations, see the [Documentation](#)).*

Target	OType	J2000 Position	Refs	Resolver
m 31	LIN	00:42:44.32 +41:16:07.5	6921	M 31 [Simbad]
	G	00:42:44.35 +41:16:08.6 ± 0.080"	3389	MESSIER 031 [NED]
	VizieR (local): No table found for: m 31			[VizieR]

Simbad first

Ned first

All Resolvers

Ignore cache

XML output

Enter the name of the astronomical object

Alternatively enter the name of the file containing object names, one per line (lines starting by # are comments)

Aucun fichier choisi

Figure 6 : Capture d'écran de Sesame après la recherche d'un identifiant (M 31)

Le service Sesame ne se contente pas de fournir aussi peu d'informations. En effet, la recherche d'objets sur Sesame est également possible directement via une URL. Sesame, en retour, renvoie un fichier XML (cf. figure 7) contenant les informations de l'objet pour les trois bases de données. Grâce à ces recherches directement par URL il est ainsi possible de faire très rapidement (via par exemple des scripts ou des logiciels tiers) de très nombreuses requêtes sur Sesame. Nous verrons plus tard que ce service, très utilisé à travers le monde, me sera également très utile dans le cadre de mon stage.

```

▼<Sesame xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="http://vizier.u-strasbg.fr/xml/sesame_4x.xsd">
  ▼<Target option="-SNVA">
    <name>M31</name>
    <!-- Q11143745 #1 -->
    ▼<Resolver name="S=Simbad (via url)">
      <oid>1575544</oid>
      <oname>M 31</oname>
      <otype>LIN</otype>
      <jpos>00:42:44.32 +41:16:07.5</jpos>
      <jradeg>10.6847083</jradeg>
      <jdedeg>41.26875</jdedeg>
      ▼<z>
        <v>-0.001004</v>
        <e>0.000023</e>
        <q>D</q>
        <r>2002LEDA.....0P</r>
      </z>
      <MType>Sb</MType>
      <nrefs>6921</nrefs>
    </Resolver>
    ▼<Resolver name="N=NED">
      <otype>G</otype>
      <jpos>00:42:44.35 +41:16:08.6</jpos>
      <jradeg>010.6847929</jradeg>
      <jdedeg>+41.269065</jdedeg>
      <refPos>2010ApJS..189...37E</refPos>
      <errRAmas>80</errRAmas>
      <errDEmas>80</errDEmas>
      ▼<Vel>
        <v>-300.09229</v>
        <e>3.89730</e>
        <r>1991RC3.9.C...0000d</r>
      </Vel>
      <MType>SA(s)b</MType>
      <oname>MESSIER 031</oname>
      <nrefs>3389</nrefs>
    </Resolver>
    ▼<Resolver name="V=VizieR (local)">
      <INFO>Zero (0) answers</INFO>
    </Resolver>
  </Target>
</Sesame>
<!-- - =====Done (2012-Jun-07,13:56:49z)===== -->

```

Figure 7 : Fichier XML fournit par Sesame lors de la recherche de l'objet M31 via l'URL.

II / Les résolveurs

Que ce soit dans le cadre du stage ou dans le fonctionnement de Sesame, trois résolveurs sont utilisés. Un résolveur est en fait une base de données contenant des informations précises sur des objets astronomiques. Les trois résolveurs sont : Simbad, VizieR et NED. Concernant les deux premiers, nous avons déjà vu leur fonctionnement car appartenant au CDS. Quant au troisième, il s'agit d'un service créé et maintenu par la NASA (NED : NASA Extragalactic Database). Sur sa forme, il s'agit du même service que Simbad, c'est à dire une base de données qui contient une grande quantité d'objets avec des informations sur chacun. Mais, il existe une différence notable entre les deux. Simbad possède un nombre d'objets conséquents mais se concentre surtout sur les informations fournies pour chaque objet. A l'inverse, NED essaye de référencer une quantité d'objets plus importante mais possède moins d'informations pour chacun d'entre eux.



Coté chiffres, NED contient aujourd'hui plus de 170 millions d'objets référencés ainsi que 182 millions d'identifiants. Coté service, il est possible tout comme Simbad de lancer des recherches sur un objet en particulier (via son nom, sa position, etc.). L'affichage qui suit cette recherche affiche, de la même manière que Simbad, les différentes informations sur l'objet ainsi que les autres dénominations de cet objet comme le montre la figure 8.

Ces trois résolveurs réunis ensemble forment une unité très puissante. En effet, avec NED qui connaît de très nombreux objets, Simbad qui est capable de donner beaucoup d'informations et VizieR qui se charge de référencer les tables où se trouve l'objet, il est facilement possible de faire des recherches. Comme nous le disions précédemment, c'est Sesame qui se charge, entre autre, d'unifier ce trio de bases de données.

NED results for object MESSIER 031

1 objects found in NED. [Skyplot\(first 100\)](#)

SOURCE LIST																		
Row No.	Object Name	EquJ2000.0 RA	DEC	Object Type	Velocity/Redshift km/s	z	Mag./ Filter	Separ. arcmin	Refs	Notes	Phot	Posn	Vel/z	Diam	Assoc	Images	Spectra	Row No.
1	MESSIER 031	00h42m44.3s	+41d16m09s	G	-300	-0.001001	4.36	...	3389	18	142	8	26	7	2	Retrieve	Retrieve	1

Detailed information for each object

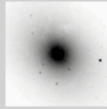
Object No. 1 - MESSIER 031

INDEX for MESSIER 031

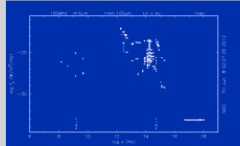
Essential Data (jump to sub-section of this query report):

- [Essential Note](#)
- [Cross-IDs](#)
- [Coordinates](#)
- [Basic Data](#)
- [Quantities Derived from Redshift](#)
- [Redshift-Independent Distances](#)
- [Quick-Look Photometry and Luminosities](#)
- [NEW Quick-Look Angular and Physical Sizes](#)
- [Classifications](#)
- [Foreground Galactic Extinction](#)
- [External Services](#)

Detailed Data (NED queries):



[Images](#)



142 Photometric data point(s) and SED

[Spectra](#)

[Redshift-Independent Distances](#)

[3389 Reference\(s\)](#)

[8 Position data point\(s\)](#)

[26 Redshift data point\(s\)](#)

[7 Diameter data point\(s\)](#)

[18 Note\(s\)](#)

[2 Association\(s\)](#)

[UGC data](#)

[RC3 data](#)

ESSENTIAL NOTE for MESSIER 031 ([Back to INDEX](#))

N/A

Figure 8 : Capture d'écran du résolveur NED affichant les informations de M31.

III / Le « Name Resolver » Indien

L'observatoire virtuel Indien possède une application Android basée sur les résolveurs NED et Simbad et permet de rechercher un objet en donnant son nom dans le champ de recherche. L'interface utilisateur est minimaliste et se compose d'un champ de saisie de l'objet (exemples : M31, M51, etc.), d'un bouton de validation et d'une partie permettant l'affichage de quelques informations concernant l'objet (cf. figure 9). Cette application reste cependant très simpliste et pose de nombreux problèmes. Néanmoins, cette application permet de voir les bases d'un résolveur de noms et permet de comprendre tous ce qu'il doit être fait afin de le rendre vraiment efficace.



Figure 9 : Capture d'écran de l'application Android du VO indien

Comme nous l'avons vu précédemment, cette application est loin d'être parfaite et possède de nombreux problèmes. Voici donc une liste rapide des problèmes les plus gênants que nous avons remarqués :

- Un seul résolveur est affiché à la fois. Changer de résolveur oblige la personne à relancer une recherche.
- L'application est dépendante d'Internet. En effet, sans réseau internet, l'application n'est pas capable de fonctionner et celle-ci peut même parfois planter.
- La recherche est faite au caractère près. Il n'existe pas de système d'aide ou d'auto-complétion. La personne désirant faire une recherche doit taper à la lettre prête le nom de l'objet.
- Très peu d'informations sont affichées par objet, ce qui est dommage surtout lorsque l'on connaît la capacité du résolveur Simbad.
- Il n'est pas possible de stocker des objets en mémoire. Même si vous recherchez souvent un objet, il est obligatoire de refaire des requêtes en ligne sur les serveurs.

- Ce type d'application est très gourmand pour les résolveurs car à la moindre recherche, des requêtes sont faites dessus et une utilisation de cette application par de nombreux utilisateurs en plus des services habituels pourrait facilement surcharger les serveurs ou en tout cas les ralentir.

En conclusion, si un service de ce genre devait être recréé, il serait indispensable de prendre en compte tous les défauts de cette application indienne.

Chapitre 3 :

Le Stage

I / Le sujet du stage

1.1) La demande

Avant de décrire en détails ce que m'a demandé de faire le CDS et plus particulièrement M. André Schaaff, il est important de comprendre le fonctionnement du CDS et sa façon de procéder ces derniers temps.

Ouverture et modernisation

Le CDS a toujours cherché à fournir des services très efficaces et pouvant fonctionner sur tous les supports. De plus, le CDS souhaite également que ses services soient disponibles où que l'on se trouve. Ces dernières années, l'informatique a grandement évolué et nous avons vu naître les smartphones, autrement dit, des téléphones multifonctions permettant notamment l'installation d'applications et l'utilisation d'internet que l'on soit n'importe où. Par la suite, les tablettes ont également fait leur apparition. Le CDS a donc voulu suivre cette modernisation et s'ouvrir à toutes ces nouvelles plateformes en y proposant ses services. C'est donc ainsi qu'est né le sujet de mon stage que nous allons découvrir sans plus attendre.

Objectifs du stage

Lors de mon entretien avec M. André Schaaff au mois de Janvier, le sujet du stage et donc l'objectif de ce stage était défini comme suit : « Etudes et prototypage pour des services Astronomiques », mais au fil des mois et jusqu'au premier jour du stage, le sujet s'est affiné et l'objectif durant ce stage est devenu triple. En effet, le premier objectif fut au final de réaliser un résolveur de noms, tout comme le fait Sesame, mais pour les appareils mobiles tournant sur Android (un autre stagiaire s'occupant de la plateforme iOS). Le deuxième objectif fut quant à lui de réaliser un prototype de cette application sous HTML5 afin de voir les possibilités actuelles de HTML5 et afin de voir s'il était possible de fournir un seul code dans ce langage. Le but étant de réaliser une application universelle, autrement dit, compatible sur n'importe quel terminal mobile, qu'il soit sous Android, iOS ou tout autre système d'exploitation sans avoir besoin de toucher au code natif de chacun de ces systèmes. Enfin, une fois tous cela réalisé, il reste encore un troisième objectif, qui est d'étudier les deux versions fournies (Android/HTML5), afin d'en ressortir un comparatif sur de nombreux paramètres tels que le temps de développement, la rentabilité, l'efficacité, etc.

1.2) Le cahier des charges de SkyObjects

Le résolveur de noms pour Android (et iOS), autrement dit, SkyObjects, a eu un cahier des charges très évolutif durant toute la durée du stage. Ainsi, l'application s'est vue rajouter des fonctionnalités au fur et à mesure de son développement mais je peux désormais fournir un listing de tout ce que devait faire l'application à réaliser. Avant cela, il est important de dire que cette application se voulait être plus performante que celle créée par le VO Indien en corrigeant notamment toutes ses erreurs et en ajoutant de nombreuses fonctionnalités. Ainsi, l'application Indienne se positionnait plus pour nous comme un modèle qu'il ne fallait surtout pas reproduire, bien au contraire.

L'application doit donc avoir:

- une navigabilité via des onglets.
- un accueil avec notamment la possibilité d'y afficher l'image du jour.
- un onglet dédié à la recherche permettant l'auto-complétion.
- une gestion de favoris avec un onglet dédié.

- une possibilité de configuration notamment pour réguler les accès internet pour éviter les connexions et téléchargements trop nombreux pour les personnes qui pourraient être surtaxées a cause de ces connexions.
- la possibilité de rédiger une note pour chaque objet dans les favoris.
- Une ensemble de vues contenant toutes les informations d'un objet pour chaque résolveur (une vue par résolveur), sa note s'il est dans les favoris et ses identifiants. Le passage d'une vue à une autre se faisant de manière simple grâce au glissement du doigt sur l'écran.
- Une base de données interne stockant une multitude d'objets (les plus couramment recherchés), les notes et les favoris. Le stockage d'objet dans une BDD interne permet l'utilisation de l'application même lorsque la personne n'est pas connectée à internet, de minimiser les connexions internet et les requêtes sur les différents résolveurs.

Voilà donc pour tout ce qui concerne les points importants à développer et sur lesquels nous allons donc revenir. Il est également important de savoir que l'application utilise les trois résolveurs étudiés auparavant soit Simbad (CDS), VizieR (CDS) et NED (NASA).

Pour la suite de ce chapitre, nous allons voir de manière thématifiée et quasiment chronologique le déroulement du stage en y définissant les points de conception et de réalisation les plus importants et en soulevant également les différents problèmes que j'ai rencontrés. Mais avant de passer à la conception et réalisation même de l'application, je trouve important de revenir sur mes premiers pas au CDS.

II / Les premiers-pas

2.1) Découverte des locaux et du personnel

Lors du premier jour de stage et de mon arrivée à l'observatoire, je n'étais pas perdu car M. Schaaff m'avait déjà fait visiter l'observatoire lors de mon entretien avec lui durant le mois de janvier. En revanche, je ne connaissais pas encore la pièce où j'allais passer mes 10 semaines. C'est donc Gilles Landais (personne que j'avais également vue lors de l'entretien et

qui m'a accueilli le premier jour car M. André Schaaff était en vacances) qui m'a conduit à mon bureau. Ce bureau, placé à l'entrée même du CDS et appelé l'aquarium car vitré sur deux cotés, est très bien équipé. En effet, celui-ci possède trois postes de travail avec trois machines sous divers environnements (un iMac en libre service, un PC sous Windows 7 pour Alexandre et un ordinateur sous Ubuntu pour moi). De plus, on avait également accès à un iPad et une tablette Archos sous Android pour les tests des applications. Tous les postes sont bien sûr reliés à Internet, très rapide d'ailleurs grâce à la fibre optique présente à l'observatoire. Toujours lors de ce premier jour, j'ai fait la connaissance de Julien, le stagiaire développant la même application que moi mais sous iOS et avec qui j'ai discuté, durant tout le long du stage, de la conception de l'application. En plus des machines du bureau, j'ai également durant tout le stage utilisé mon MacBook Air en tant que complément pour la partie Android et en tant que machine principale pour le développement en HTML5.

Du côté du personnel, j'en ai découvert une partie grâce aux réunions que nous avons eues lors des deux premières semaines et qui avaient pour but de nous présenter le CDS, ses différents services ainsi que le travail que chacun devait réaliser. Pour le reste du personnel du bâtiment où je me trouvais, je l'ai découvert au fil des journées. Enfin, grâce au café des vendredis matin, j'ai pu découvrir l'ensemble du personnel de l'observatoire. Quoiqu'il en soit, le personnel de l'observatoire fut très sympathique avec moi ce qui m'a permis une bonne adaptation.

2.2) Découverte des outils

Android et Eclipse

Le stage allait être pour moi l'occasion de toucher au système Android (depuis que je suis passé au Smartphone, j'ai toujours possédé un iPhone) et de programmer dessus (je m'étais quand même entraîné à développer sur Android les semaines précédentes le stage grâce à l'émulateur afin de connaître les bases au moment d'arriver au stage). M. Schaaff avait tout préparé avant mon arrivée et ainsi, il y avait déjà le SDK d'Android installé sur Eclipse (Eclipse lui ne m'a pas posé de problème pour l'avoir déjà utilisé à de multiples reprises). La prise en main du SDK d'Android et des nouveautés présentes par ce fait dans Eclipse fut extrêmement rapide grâce à mon entraînement préalable. Connaissant déjà JAVA, Android ne

m'a pas paru difficile à apprendre car le seul point qui m'avait un peu troublé au départ était d'utiliser du XML pour réaliser les interfaces graphiques.

HTML5/JavaScript

De la même façon que pour Android, cela allait être la première fois que j'allais utiliser des balises propres à HTML5 mais surtout que j'allais programmer du JavaScript. Heureusement pour moi, grâce à internet et aux nombreux tutoriels qui y voient le jour ainsi qu'à mes connaissances acquises à l'IUT, l'apprentissage fut très rapide. Pour la réalisation de l'application en HTML5, nous avons utilisé un outil appelé Coda2 (disponible sous mac) qui permet d'éditer du code à deux et en même temps. Un outil très utile vu que j'ai réalisé l'application en HTML5 en collaboration avec Julien.

PhoneGap

PhoneGap est un outil que nous avons testé dans le cadre du stage pour étudier ses performances. PhoneGap est une sorte d'API capable de convertir à la volée lors du lancement de l'application du code source en HTML5 vers le langage désiré (Java et XML s'il s'agit d'Android). Prenons un cas concret pour mieux comprendre. Imaginons que nous souhaitons rendre notre application en HTML5 disponible sous la forme d'un APK sous Android, et bien il suffit juste de créer une nouvelle application Android avec eclipse, d'importer le .jar de PhoneGap dans le dossier lib. Ensuite, concernant l'application en elle-même, il faut modifier l'activité principale de l'application en lui demandant d'exécuter PhoneGap et fournir les fichiers html dans un dossier www dans le dossier assets. C'est absolument tout ce qu'il y a faire, l'application peut être générée et sera fonctionnelle. De plus, PhoneGap est compatible Android mais aussi iOS, Windows Phone, BlackBerry, etc. On n'a donc plus qu'un seul et unique code pour générer des applications pour les différentes plateformes. J'ai donc utilisé avec Julien cet outil durant le stage et nous en reparlerons donc plus tard dans le rapport.

SVN et WIKI

Durant toute la durée du stage, j'ai eu accès au serveur SVN du CDS afin de pouvoir stocker toutes les sources des applications. De plus, un wiki sous mon nom était à ma

disposition. Grâce à lui, j'ai pu rédiger chaque jour et décrire ce que j'avais fait, signaler de multiples choses, donner des informations, déclarer les bugs, rendre disponible des versions de test, etc. Ce wiki était très pratique car pour tout ce que j'ai fait, il y a eu une trace écrite via ce wiki.

III / L'application Android : SkyObjects

Dans cette partie, nous allons voir les éléments importants dans les phases de conception et de réalisation de l'application. Nous allons voir que l'ensemble de l'application repose sur une base de données interne à l'application et nous allons donc pour cette raison, commencer par parler de celle-ci.

3.1) La base de données

Nous l'avons vu, l'application doit être un « Name resolver » et doit être ainsi capable de recueillir des informations sur des objets astronomiques. Les bases de données qui seront ainsi utilisées sont celle de Simbad, VizieR et NED d'où leurs présentations dans l'existant. Pour unifier et simplifier la recherche, l'application utilisera des requêtes URL de Sesame afin de recueillir les informations pour les trois bases de données en ne faisant qu'une seule requête. Cependant, faire de multiples recherches via cette application pourrait (si l'application est lancée depuis de nombreux terminaux Android en même temps) surcharger Sesame ainsi que les trois bases de données. De plus, lancer de multiples requêtes via une l'application signifie également l'utilisation de sa connexion internet et de l'utilisation de son forfait DATA.

Pour palier à ces deux problèmes, nous avons prit la décision d'intégrer une base de données interne à l'application. Cette base de données contient ainsi de nombreux objets, tous très souvent recherchés et permet donc de limiter les recherches en ligne car très souvent, la base de donnée interne possèdera l'objet recherché.

Structure

Nous avons choisi de faire la base de données en SQLite car ce type de base est gratuit, rapide et efficace et que ce type de base est directement implémenté dans Android sans avoir besoin de librairie externe. La base de données à construire avait pour rôle de stocker quatre types d'éléments : les objets astronomiques, les identifiants des objets, les notes pouvant être faites sur les objets et les favoris. Pour stocker ces éléments, nous avons créé quatre tables.

La table « Objects » :

La table « Objects » est celle permettant de stocker tous les objets astronomiques qu'ils proviennent de Simbad, NED ou VizieR. Ainsi, dans cette table, on trouve un champ « mainid » qui contient le nom principal de l'objet (un même objet dans le ciel peut avoir un nom principal différent sur les trois résolveurs). On trouve ensuite le champ « resolver » qui permet de connaître d'où provient l'objet stocké, c'est-à-dire de quel résolveur. Il y a donc trois possibilités pour ce champ : NED, Simbad ou VizieR.

Ainsi, un même objet astronomique peut être représenté trois fois dans la table « Objects » car pouvant être référencé sur les trois résolveurs. Les champs « mainid » et « resolver » forment la clé primaire de la table « Objects ». Nous venons de le voir, un même objet peut être référencé jusqu'à trois fois dans la table tout en possédant un « mainid » différent. Cela nous avait alors posé un gros problème puisqu'il nous était impossible de récupérer les trois lignes de la table correspondant à un même objet astronomique. Après de multiples discussions autour de ce problème, nous avons décidé de rajouter tout simplement un entier « idfromobject » dans la table permettant de relier les trois résolveurs. Ainsi, les trois lignes de la table « Objects » représentant un même objet astronomique possèdent tous les mêmes entiers « idfromobject ». Enfin, la table contient également une multitude de champs dédiés aux caractéristiques même de l'objet. Ainsi, on peut trouver un champ pour sa position, un champ pour sa vitesse, etc. Tous les champs de la table hormis le champ « idfromobject » sont de type « TEXT ».

La table « identifiens »

La table « identifiens » permet de retenir tous les identifiants (qu'il soit principal ou non) pour tous les objets présents dans la table « Objects ». On y trouve un champ « alias » contenant le nom des identifiants, un champ « mainid » (clé secondaire) qui permet de relier un identifiant à un objet de la table « Objects » et enfin la clé primaire « formataliases », qui est identique au champ « alias », mais auquel on a supprimé tous les espaces de la chaîne de caractère. Ce champ sans espace nous sera très pratique lors de la recherche d'objet dans la base via le moteur de recherche. Enfin, on peut dire que pour chaque objet de la table « Objects », il y a au moins un identifiant qu'il lui est associé (relation 1, n).

La table « Notes »

La table « Notes » permet de retenir toutes les notes qui ont été rédigées via l'application elle-même. Ainsi, on trouve un champ « content » qui contient le texte, soit la note en elle même, et un champ « idfromobject » (clé principale et secondaire) qui permet de relier la note à un objet astronomique. Un objet de la table « Objects » peut avoir une seule et unique note (relation 0,1).

La table « Bookmarks »

La table « Bookmarks » contient deux champs déjà décrits précédemment car apparaissant dans les autres tables : « idfromobject » (clé principale et secondaire) et « mainid ». « idfromobject » permet de connaître les objets astronomiques en favoris et le champ « mainid » leurs noms pour pouvoir les afficher. Il est vrai qu'un champ « bookmarks » de type booléen aurait pu être directement ajouté à la table « Objects » pour savoir s'il est en favoris ou non, mais nous avons décidé de séparer les favoris pour une question de rapidité. En effet, si les favoris étaient intégrés à la table « Objects », cela aurait nécessité à l'application de parcourir l'intégralité de la table « Objects » en vérifiant le booléen « bookmarks » pour pouvoir faire un listing des favoris. Or cette table contient plusieurs milliers de lignes et cette opération, bien que restant rapide, aurait tout de même pris plus de temps que de lister le contenu de la table « Bookmarks ». La rapidité a donc été le point déterminant de la séparation des favoris dans une table à part.

La génération

Nous voulions que notre base de données contienne des objets très souvent utilisés dans le cadre des recherches. M. André Schaaff a donc pris la décision d'intégrer le catalogue « NGC » (catalogue astronomique, répertoriant de très nombreux objets astronomiques et correspondant en général à une mission réalisée) dans son intégralité. M. André Schaaff nous a alors fourni un fichier au format XML contenant toutes les informations pour chaque résolveur et pour chaque objet du catalogue. Ce fichier, devait être chargé via Internet au premier lancement de l'application puis parsé et transformé afin de créer la base de données. Malheureusement, nous nous sommes rendu compte que ce processus durait trop longtemps (plusieurs heures avec le catalogue NGC complet). Il était donc logique que nous ne puissions pas demander à l'utilisateur de l'application d'attendre plusieurs heures avant de commencer à utiliser l'application.

Pour palier à ce problème, nous avons alors décidé de générer la base de données auparavant et de l'intégrer à l'application, faisant ainsi grandir sa taille. Pour la générer, j'ai donc réalisé un petit programme en Java permettant de donner un fichier XML en entrée et de récupérer la base de données en sortie. De manière brève sur son fonctionnement, ce programme utilise la librairie KXML afin de parser le fichier XML. Une fois parsé, le programme contient une liste d'objets astronomiques. Chaque élément de cette liste est lui même une liste contenant une « HashMap ». Une HashMap contient les informations de l'objet pour un résolveur. La liste contenant ces HashMap en possède au maximum trois puisque l'objet est décrit par les trois résolveurs au maximum.

Une fois que cette architecture a été créée grâce au XML, chaque objet astronomique se voit attribuer un identifiant (pour le champ « idfromobject ») puis est inséré dans la base de données (celle-ci ayant bien évidemment été créée préalablement), en suivant la hiérarchie de celle-ci (« Objects » et « identifiens »). Le programme trie donc toutes les informations afin que toutes les insertions dans la base de données soient correctes. La génération de la base de données avec ce programme dure 2 minutes environ.

Le contenu

Au niveau du contenu, la base de données, on l'a dit, possède le catalogue NGC. Cela représente ainsi près de 7 800 objets. Etant donné que chaque objet peut être défini jusqu'à trois fois, on obtient au final plus de 23 000 descriptions d'objets et près de 170 000 identifiants. La base de données avec ce contenu atteint quasiment les 17 Mo.

Pour qu'elle soit intégrée à l'application, sous Android, il suffit de glisser le fichier de la BDD dans le dossier nommé « Assets ». Ce dossier dans une application Android permet de fournir des fichiers externes à l'application. Mais, pour utiliser la BDD, celle-ci devait être copiée dans la mémoire du téléphone car le dossier « Assets » ne permet pas l'écriture puisqu'il appartient au « Bundle » même de l'application. Cette copie dans la mémoire du téléphone était donc indispensable et c'est donc pour cela que nous avons mis en place « le premier chargement ». Le « premier chargement » a lieu lors du premier démarrage de l'application et affiche une page de chargement pour faire patienter l'utilisateur pendant que l'application s'occupe de copier la BDD dans la mémoire du téléphone. Grâce à cette solution et après avoir testé sur le Samsung Galaxy S2 de M. André Schaaff, je pensais que tous les problèmes pour la base de données étaient réglés car le premier chargement durait moins de cinq secondes et que la BDD était accessible en lecture mais aussi en écriture. Mais, malheureusement, j'avais tort. En effet, lors du test sur la tablette Archos qui était à ma disposition, je me suis rendu compte que l'application ne démarrait plus mais « plantait » tout bonnement au démarrage. Après de multiples recherches, j'ai trouvé d'où venait le problème sur Internet. En effet, je me suis rendu compte que pour les versions d'Android inférieures à la 2.3, celle-ci ne supportait pas dans son dossier « Assets » des fichiers de plus de 1Mo. La base de données avec ses 17Mo posait donc un problème pour ces versions d'Android. Après réflexion, j'ai trouvé une solution qui pour moi me paraissait la meilleure pour que l'application reste très réactive lors du premier lancement. En effet, j'ai pris la décision de découper le fichier de la BDD en dix-sept plus petits, tous inférieurs à 1Mo. Ainsi, tous ces petits fichiers allaient être à nouveau compatibles avec toutes les versions d'Android (chose que nous voulions absolument). Bien évidemment, nous ne pouvions pas laisser le reste de l'application tel quel et la copie du premier lancement a été modifiée en conséquence. En effet, désormais, lors du premier démarrage de l'application, les 17 fichiers du dossier « Assets » sont copiés dans la mémoire du téléphone l'un après l'autre mais dans un seul et

unique fichier afin de recréer dans le téléphone la BDD complète avec ses 17Mo. En modifiant ainsi l'application, celle-ci fut donc rendue compatible avec toutes les versions 2.0 ou supérieures d'Android, soit quasiment l'intégralité des terminaux Android à ce jour.

En conclusion de cette partie, nous pouvons dire que même si la BDD se voulait peu complexe à la base, sa génération et son intégration dans l'application a révélé des problèmes qui ont pris chacun un certain temps à résoudre...

3.2) L'interface utilisateur

De manière générale pour commencer, l'interface utilisateur se voulait d'être compatible dans deux langues différentes : le français lorsque le téléphone est dans cette langue et bien évidemment l'anglais lorsque le téléphone n'est pas en français. Pour parvenir à cela, j'ai donc réalisé un fichier langue pour chacune des deux langues et je les ai intégrés à l'application. Android permet très facilement de gérer le multi langages, il est donc inutile d'y entrer plus en détails.

Autre élément de l'interface utilisateur, l'orientation. Nous avons constaté que sur tablettes, contrairement aux smartphones, il était préférable d'afficher les interfaces au format paysage. Ainsi, grâce à des calculs au démarrage de l'application permettant de calculer la diagonale d'écran en pouce, nous avons décidé de bloquer l'interface utilisateur en mode vertical pour les smartphones et en mode paysage pour les tablettes. Nous avons placé la frontière entre smartphones et tablettes à 5,7 pouces. Ainsi, le dernier terminal en taille pouvant encore être classé dans les smartphones est le Samsung Galaxy Note.

Voilà donc pour ce qui concernait les interfaces graphiques de manière générale. Passons désormais aux différents écrans qui constituent l'application (les captures des différents écrans sont disponibles dans les annexes de ce rapport).

L'onglet « Accueil »

Il s'agit bien évidemment du premier écran que l'on voit en arrivant sur l'application. Celui-ci permet notamment de voir que l'application fonctionne via des onglets. Cet écran contient une image de fond et permet également l'affichage de l'image du jour au centre de

l'écran. Cette image du jour est une photo d'un des objets de la BDD interne. Chaque jour et si l'utilisateur l'autorise via les options, un objet est choisi de manière aléatoire et sa photo est téléchargée via Internet.

L'onglet « Recherche »

Il s'agit du deuxième onglet disponible et c'est grâce à celui-ci que des recherches d'objets sont possibles. Pour ce faire, il suffit de rentrer le nom de l'objet à rechercher dans le champ dédié. Une liste permet d'afficher les résultats de la recherche et un bouton de suppression permet d'effacer la recherche en cours.

L'onglet « Favoris »

Cet onglet permet d'afficher sous la forme d'une liste les différents noms d'objets qui sont enregistrés en temps que favoris.

L'onglet « Préférences »

Cet onglet permet de multiples configurations de l'application. Il permet également d'avoir plus d'informations sur l'application (auteurs, copyright, etc.), de réinitialiser l'application (remet notamment la BDD comme à l'origine), de reporter les erreurs (voir 3.6 de ce chapitre) et de limiter les connexions internet en activant ou désactivant les mises à jour ou encore en activant et désactivant l'image du jour.

Le panel d'informations.

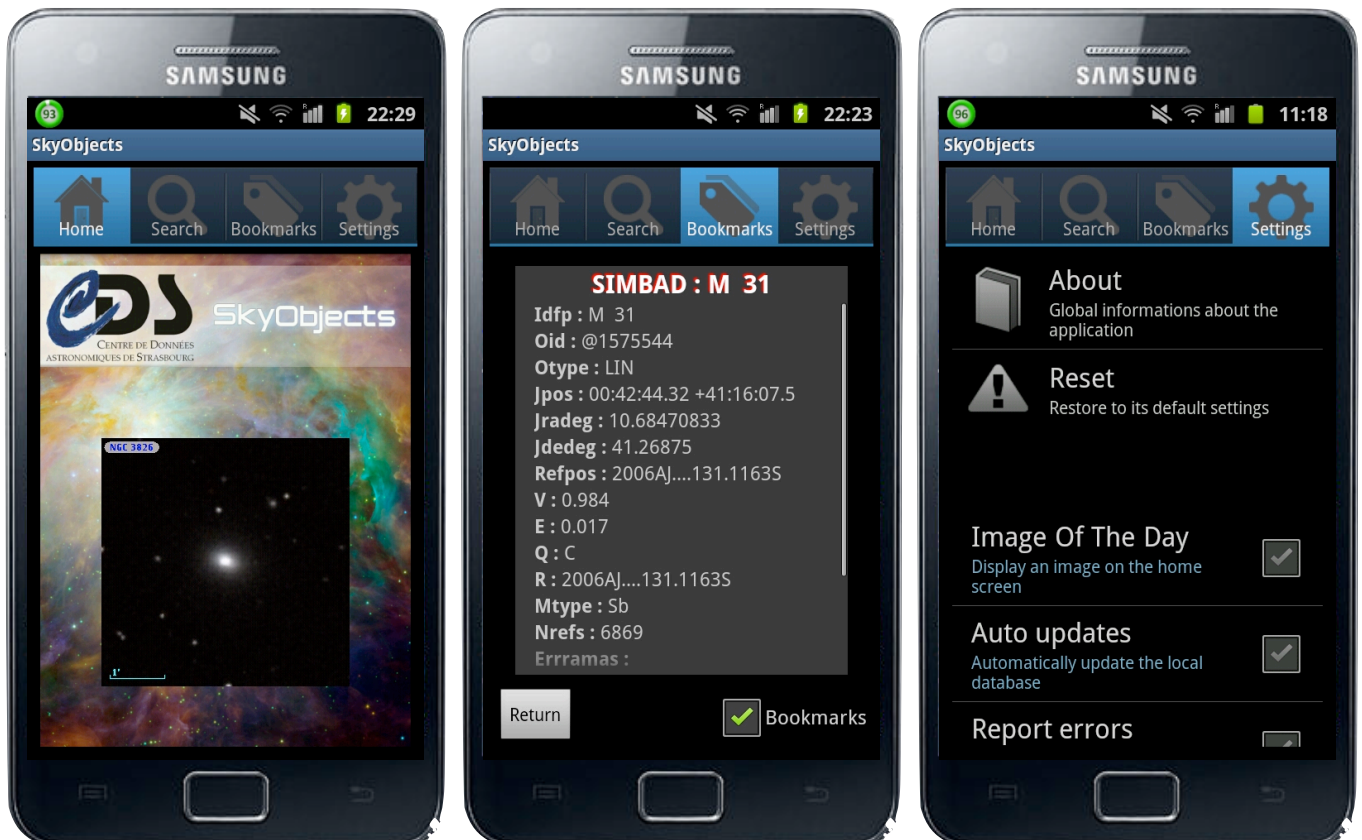
Ce panel d'informations est en fait un ensemble de vues. Ce panel s'affiche lorsque l'on demande des détails sur un objet (lorsqu'on le sélectionne dans la liste des résultats de recherche ou dans les favoris). Ce panel est constitué de cinq vues et pour passer d'une vue à une autre, il suffit tout simplement de glisser son doigt sur l'écran. Les trois premières permettent d'afficher les informations de l'objet en fonction des trois résolveurs, la quatrième

permet d'afficher tous les identifiants de l'objet. Quant à la dernière, elle n'est accessible que si l'objet est en favoris et permet de rédiger une note sur l'objet.

Quelle que soit la vue affichée sur ce panel, deux éléments sont toujours accessibles : le bouton retour et la checkBox favoris. Le bouton retour permet de revenir à la recherche ou à la liste des favoris, la checkBox permet quant à elle d'ajouter l'objet en favoris ou de le retirer. Lors de l'ajout d'un objet en favoris, si les informations de celui-ci proviennent d'une requête URL, alors l'intégralité de cet objet est ajouté à la BDD internes, ceci afin d'éviter de faire des requêtes URL pour des objets qui sont en favoris et donc supposés être utilisés souvent.

Des images sont plus parlantes que du texte lorsqu'il s'agit d'interfaces graphiques, différentes captures d'écrans sont donc disponibles en pages annexes ainsi que ci-dessous.

Figure 10 : captures d'écrans de SkyObjects



3.3) Le moteur de recherche

Avec la base de données, le moteur de recherche représente le deuxième gros morceau de SkyObjects. En effet, le moteur de recherche se devait d'être le plus efficace possible et le plus rapide possible. Après d'innombrables discussions autour de ce sujet que ce soit avec M. André Schaaff ou avec Julien, mon collègue de travail, nous sommes arrivés à mettre au point un moteur de recherche très efficace. Nous allons donc voir dès à présent son fonctionnement.

Tout d'abord, il est important de savoir que le moteur de recherche se lance dès qu'il y a un changement dans le champ de recherche. Lorsque c'est le cas, le moteur de recherche commence par récupérer ce qui est écrit dans le champ de recherche et supprime en interne tous les caractères vides. Deux suites sont possibles : soit il reste moins de deux caractères et le moteur s'arrête, soit il y a plus de deux caractères restants et la recherche commence réellement. Dans ce deuxième cas, le moteur va tout d'abord faire une recherche au caractère près dans la base de données interne sur le champ « formataliases » (c'est pourquoi nous avons stocké tous les identifiants avec et sans espaces). Deux cas sont à nouveau possibles : soit la recherche est concluante et le nom de l'objet ainsi que l'option « Afficher plus de résultats » s'affichent dans la liste des résultats, soit le moteur continue la recherche. L'option « Afficher plus de résultats » permet de continuer la recherche. En effet, si la personne cherchait M31, celle-ci a commencé à écrire M3. M3 est un objet qui existe dans la BDD interne et est donc affiché en tant que résultat. L'option « Afficher plus de résultats » permet donc d'afficher plus de noms commençant par M3 et permettra donc ainsi l'affichage de M31. Lorsque la recherche continue car aucun objet n'a été trouvé, le moteur fait alors une recherche approchée. C'est à dire qu'il va rechercher tous les objets dans la BDD interne qui commencent par ce que l'utilisateur a écrit. On obtient alors trois cas :

- Il y a plus de dix résultats et l'application affiche «Trop de résultats ».
- Il y a entre un et dix résultats et ils s'affichent dans la liste.
- Il n'y a pas de résultat et le moteur continue sa recherche.

Dans le troisième cas, le moteur de recherche continue la recherche en abandonnant la BDD interne et interroge par URL le service Sesame. Ce service, dans tous les cas, retourne un fichier XML que l'application récupère et parse. Après le « parsing », il y deux cas : soit Sesame a retourné un fichier XML contenant les informations de l'objet voulu et le nom de l'objet recherché est affiché dans la liste des résultats (avec une indication montrant qu'il a été trouvé en ligne), soit le fichier XML ne contient aucune information et l'utilisateur est informé qu'il n'y a aucun résultat pour sa recherche (cf. schéma du moteur en annexes).

Pour rester simple dans ce rapport, je ne rentrerais pas plus en détails sur le fonctionnement de ce moteur, mais il permet, de part son fonctionnement d'être efficace et rapide. Dans le meilleur des cas, lorsque l'objet est trouvé dans la BDD interne au caractère près, la recherche dure moins de 0,3 seconde sur le Samsung Galaxy S2 (varie selon les terminaux Android et lorsque le moteur va jusqu'à la partie de requête par URL, le temps de recherche varie selon la connexion).

3.4) Le moteur de mise à jour

Nous l'avons vu, l'application possède une base de données en interne contenant des objets astronomiques et il faut savoir que ces objets sont aujourd'hui toujours étudiés. Certains de ces objets sont donc parfois mis à jour avec des informations qui changent ou qui sont supplémentaires. Lors du changement des informations d'un objet du catalogue NGC sur l'un des résolveurs, il nous fallait trouver un moyen de le savoir et de mettre à jour la base de données interne de l'application. C'est pour cette raison qu'est né le moteur de mise à jour qui se lance de manière automatique au démarrage de l'application et dont voici le fonctionnement.

Le moteur commence toujours par vérifier s'il s'est déjà activé le jour même, car si tel est le cas, cela signifie que les mises à jour des informations auraient déjà été faites. En revanche, si c'est la première fois de la journée que le moteur de mise à jour est lancé, alors il continue son processus en déterminant la date de dernière mise à jour. Ensuite, pour chaque jour séparant la dernière mise à jour du jour actuel, le moteur va faire une requête URL sur les serveurs de l'Observatoire, en envoyant en paramètre le code du jour qu'il souhaite mettre à jour. Par exemple, si nous sommes le 18 juin 2012 et que l'application n'a pas été lancée depuis le 14 juin, l'application va faire successivement des requêtes URL en passant en paramètre la date (sous forme jjmmaaaa) : 15062012, 16062012, 17062012 et 18062012. Pour chaque requête URL, les serveurs de l'Observatoire renvoient un fichier XML contenant le différentiel des données par rapport à la veille du jour passé en paramètre. Ce différentiel est lu par l'application et met à jour les données dans la BDD interne qui ont changées. Ainsi, les informations dans la base de données internes sont toujours actualisées si l'utilisateur accepte les mises à jours automatiques dans l'onglet préférence.

3.5) Les notes

Nous l'avons déjà vu, les notes ne sont accessibles que lorsqu'un objet est en favoris. Lors du chargement des informations de l'objet, la note, si il y en a une de rédiger pour l'objet, est affichée. Si tel n'est pas le cas, un message indique que l'on peut rédiger une note sur l'objet. Lors de la rédaction d'une note, celle-ci est sauvegardée de manière automatique. Pour ce faire, il a fallu trouver tous les moyens qui permettent de quitter la rédaction des notes afin d'établir une liste d'éléments à surveiller. Ainsi, une sauvegarde de la note est faite lorsque l'utilisateur est sur la vue des notes et que le clavier disparaît de l'écran, lorsque que l'utilisateur change d'onglet, lorsque l'utilisateur change de vue sur le panel d'informations, etc. De plus, il a fallu prévoir le cas où l'application est arrêtée de manière brutale. Pour palier à cette possibilité, la note en cours de rédaction est également sauvegardée toutes les dix secondes. En bref, on peut considérer qu'il est impossible de perdre plus de dix secondes de rédaction de notes.

3.6) Les logs

Tout au long de la programmation, lorsqu'il y avait suffisamment de nouveautés, je sortais des nouveaux APK afin que M. André Schaaff puisse tester sur son Samsung Galaxy S2. Lors de ses tests de l'application, il arrivait que l'application plante pour telle ou telle raison. Mais, l'origine du problème n'était pas toujours identifiée par M. André Schaaff et nous n'arrivions pas toujours à reproduire l'erreur. Je ne savais donc pas ce qu'il fallait revoir. Pour cette raison, nous avons mis en place un système de logs. Ce système permet au démarrage de l'application de vérifier les logs du terminal Android afin de voir s'il y a eu une Java Exception de levée provenant de SkyObjects. Si tel est le cas, alors des informations sur le terminal Android (nom, version, etc.) sont envoyées sur les serveurs du CDS avec bien évidemment la Java Exception en elle-même.

Ainsi, lors des tests, si l'application venait à se fermer, il ne suffisait plus qu'à relancer l'application afin d'envoyer les logs aux serveurs du CDS. Grâce à ces logs que je récupérais depuis le serveur, je pouvais identifier le ou les problèmes et les corriger. Cette option d'envoi de logs est toujours possible en l'activant depuis l'onglet de configuration. Cette option s'appelle « Reporter les erreurs au CDS ».

IV / L' « IVOA Interop »

4.1) Présentation

Une IVOA Interop a lieu tous les deux ans et dure une semaine. Elle permet de rassembler les différents membres de l'IVOA afin de discuter entre eux et de montrer le travail effectué (à travers de nombreuses conférences) depuis la dernière fois. L'IVOA Interop change de lieu à chaque fois en se déplaçant « d'un observatoire virtuel membre à un autre ». Il y a deux ans, une IVOA Interop a eu lieu à Strasbourg. Quant à la dernière, elle a eu lieu du 21 au 25 mai 2012, c'est-à-dire durant le stage, à Urbana-Champaign aux Etats-Unis (à 150 km de Chicago).

M. André Schaaff a participé à cette IVOA Interop. Lors de son meeting, SkyObjects a été présenté devant la communauté présente. Nous avons alors eu l'opportunité de participer à la création d'un projet présenté lors d'un événement international et regroupant un très grand nombre de professionnels reconnus.

Nous avons appris que l'application allait être montrée à la communauté une semaine auparavant et la semaine avant le meeting fut donc très chargée car Julien et moi désirions que l'application soit la plus aboutie possible pour cette présentation importante.

4.2) Corrections logicielles

Durant la préparation de l'application pour la présentation à Urbana Champaign, il a d'abord fallu corriger tous les bugs qui restaient. J'ai donc utilisé l'application de toutes les manières possible afin de faire remonter tous les logs d'erreurs possibles. J'ai ensuite étudié tous les logs remontés et j'ai corrigé toutes ces erreurs. La correction a nécessité une journée complète de travail.

4.3) Améliorations logicielles

Deuxième étape que nous avons trouvée importante était de faire des améliorations logicielles. Plusieurs améliorations avaient été faites parmi lesquelles on trouve une

modification du moteur de recherche (afin de le rendre comme expliqué précédemment dans le rapport) ou encore l'ajout de la possibilité de faire des notes. De plus, comme les notes n'étaient pas prévues auparavant, la BDD a été modifiée et devait être régénérée. Nous avons passé deux jours sur ces améliorations au travail et nous avons également utilisé en partie les deux jours de notre week-end.

4.4) Améliorations graphiques

La dernière étape, qui fut la plus visuelle, a été de faire des améliorations graphiques. On peut dire que l'essentiel des graphismes a été fait à ce moment là : fond pour l'accueil, icônes sur les onglets, icônes sur les préférences, logo définitif pour l'application, etc. C'est donc en l'espace d'un seul jour que l'application s'est vue complètement remodelée graphiquement. Les interfaces entre la version Android et iOS sont devenues très proches à partir de ce moment là.

Le dernier jour avant la présentation a servi de test et a permis à M. André Schaaff de terminer la préparation de son meeting avec une version de SkyObjects prête pour l'évènement.

Tous nos efforts furent récompensés le jour du meeting par un mail de M. André Schaaff nous remerciant des efforts fournis durant la semaine.

Nous arrivons désormais à la fin de ce qui me semblait important de parler en ce qui concerne SkyObjects, la programmation de SkyObjects sous Android cache encore de nombreux éléments mais je pense avoir fait le tour de l'essentiel. Nous allons donc maintenant passer au prototype en HTML5.

V / Le prototype en HTML5

Dans cette partie, nous allons parler du développement du prototype développé en HTML5 mais nous insisterons plus sur les problèmes et les compatibilités car la conception

en elle même ne change pas de la version Android. Il est à noter que ce prototype est réalisé de manière à tourner avant tout sur les smartphones et tablettes.

5.1) La base de données

La base de données fut le premier problème que nous avons eu. En effet, il nous fallait reproduire la base de données interne sur ce service en ligne. Nous ne voulions pas passer par l'utilisation des cookies ou du cache car même si cela nous permet de stocker des informations, celles-ci ne sont pas définitives et sont régulièrement effacées. Il fallait donc que ce soit permanent et individuel pour chaque utilisateur. Autrement dit, il nous fallait un accès au disque dur de la personne utilisant l'application en ligne.

Après plusieurs recherches pour trouver la meilleure solution, nous avons fini par opter pour un script JavaScript (sqliteHTML5.js). Ce script nous permet de créer, d'ouvrir et de remplir une base de données stockée sur le disque dur de l'utilisateur. Ainsi, chaque utilisateur dispose de sa propre base de données (utile pour les favoris et les notes notamment) stockée sur son disque dur.

Cette solution fut pour nous très utile mais cela apporte également des contraintes. En effet, par défaut, les navigateurs internet ne permettent pas de stocker plus de 5mo sur le disque dur de la personne sans autorisation. Cela est bien évidemment une sécurité mais cela oblige la personne utilisatrice du service à autoriser l'utilisation de 20mo pour la base de données. Autre problème beaucoup plus embêtant, la compatibilité. En effet, seuls les navigateurs basés sur le Web Kit sont capables d'exécuter ces scripts JavaScript. Parmi les navigateurs internet les plus connus, seul Safari et Google-Chrome sont compatibles. Point positif tout de même, nous désirons avant tout que notre application fonctionne sur les Smartphones et tablettes. Or, les systèmes sous iOS ont Safari comme navigateur et Android un navigateur basé sur le moteur de Google-Chrome. Ainsi, sur les terminaux iOS et Android, l'application reste compatible.

5.2) L'interface utilisateur

L'interface utilisateur pour le prototype en HTML5 restera très simpliste, nous avons réalisé une feuille de style en CSS3. Cette interface permet tout de même de garder une navigation par onglets et gardant les mêmes fonctionnalités que SkyObjects. Aucun souci majeur durant la réalisation de l'interface. Point positif, cela nous a permis d'apprendre le CSS3, qui est vu de manière très rapide à l'IUT lors des cours de PHP.

5.3) PhoneGap

Nous l'avons vu lors de la découverte des outils, PhoneGap est l'outil que nous avons utilisé pour faire les conversions. Mais, PhoneGap c'est révélé moins utile que prévu lorsque nous avons tenté de faire les conversions. Hormis le fait qu'il permet d'avoir un APK, l'application ressemble au final à une simple WebView sans la barre URL. Rien n'est convertit en code natif que ce soit sur Android, iOS, ou autres. En revanche, PhoneGap permet de transformer en code natif la base de données ainsi que la gestion des capteurs.

En conclusion, PhoneGap nous a été beaucoup moins utile que prévu car même s'il permet d'avoir une application à installer, il se retrouve inutile du fait que la base de données est prise en compte par les navigateurs par défaut d'Android et iOS et que nous n'utilisons pas les capteurs pour ce prototype. En revanche, si nous désirerions pousser un peu plus le prototype en réalisant un meilleur design et en utilisant les capteurs, alors PhoneGap deviendrait rapidement un outil indispensable.

VI/ Comparaison

Dans cette partie, nous allons comparer le développement de SkyObjects en natif sur Android/iOS et le prototype de cette application en HTML5 sur différents critères afin de voir les différences entre la version en code natif (Android/iOS) et le prototype en HTML5. De manière à faciliter cette comparaison, nous allons étudier le tableau comparatif ci-dessous.

	SkyObjects	Prototype HTML5
Temps de conception	Identique car il s'agit de la même conception	
Temps de réalisation	3 semaines	1 semaine
Compatibilités	Android/iOS	Android/iOS/Safari/Chrome
Base de données	Complète et très bien gérée	Complète, quelques contraintes et problèmes de compatibilité
Langages à connaître	Java et Objective-C	HTML5 et JavaScript
Puissance et Api	Excellent	Mauvais

Un élément important dans cette comparaison et qui marque une première rupture est le temps de réalisation. Il faut d'abord savoir que le temps de réalisation pour SkyObjects n'est basé que sur l'équivalent du prototype en HTML5 afin que la comparaison soit la plus fiable possible. Ce point de comparaison montre que le développement en HTML5 a pris beaucoup moins de temps que le développement équivalent dans SkyObjects. Plusieurs raisons sont en cause, la réalisation pour SkyObjects a été faite deux fois : une fois pour Android pour ma part et une fois pour la version iOS. Ensuite, lorsque nous avons fait la réalisation en HTML5, le fonctionnement étant le même, nous avons peut être été plus rapides. Cependant, étant donné que nous avons une seule version pour les deux plateformes, il est normal que le temps de réalisation par rapport à SkyObjects soit divisé par deux. Dans notre cas, il a même été divisé par trois. Sur ce point, HTML5 est donc largement en tête pour le développement d'une application de ce type.

Passons maintenant à un autre point qui permet de réellement les comparer et qui va nous permettre de faire un bilan de la comparaison entre les deux. En effet, il s'agit du point qui est d'un côté en faveur d'HTML5 et en même temps en sa défaveur. Il s'agit de la compatibilité, en effet, au premier abord, HTML5 semble mieux ce qui en fait un bon point pour lui mais c'est aussi ce qui en fait un mauvais point car la compatibilité d'HTML5 de nos jours est encore très limitée. Pour SkyObjects, la compatibilité est limitée à iOS et Android mais cela est voulu alors que pour le prototype en HTML5, il s'agit vraiment d'une limitation. De plus, HTML5 ne permet pas l'accès complet aux capteurs du téléphone et à toute sa puissance (ligne Puissance et API) ce qui bride le terminal Android à des fonctions plus simple.

Dans ce rapport, je ne détaillerai pas tous les autres points mais ce qui a pu ressortir de mes discussions avec M. André Schaaff et Julien, c'est que HTML5 est une très bonne chose. Cependant, en raison de ses nombreux problèmes de compatibilité actuellement dus à sa conception et sa mise en place et au fait qu'il n'a pas accès à toutes les composantes du téléphone, il est préférable de programmer les applications directement en natif même si cela demande plus de temps et d'argent. En revanche, cette tendance devrait s'inverser dans les années voire les mois à venir, car la compatibilité d'HTML5 va devenir complète et grâce à son association avec le CSS3, les possibilités qui seront données aux applications seront immenses (à condition que tout les capteurs du terminal Android soit également prit en charge).

En conclusion, je pense que nous sommes dans la période de transition où l'HTML5 va devenir plus pratique pour développer des applications plutôt que de les développer en code natif. De plus, les connaissances à avoir auront besoin d'être moins grandes puisqu'il suffira de connaître l'HTML5 et JavaScript. Il ne sera plus nécessaire d'apprendre des langages multiples afin de développer sur les différentes plateformes.

Ce bilan que je viens de faire est presque celui de mon rapport, nous sommes arrivés à l'aboutissement de ce stage tout en ayant développé un nouveau service pour le CDS. Mais, avant de réellement conclure, il est intéressant de parler des perspectives.

Chapitre 4 :

Perspectives

I / Le futur de SkyObjects

L'application dans l'état actuel pourrait être déposée sur l'Apple Store ou sur Google Play, mais, SkyObjects n'est pas considérée comme terminée car de nouvelles fonctionnalités pourraient être ajoutées. Nous allons voir brièvement quels sont ces ajouts futurs de SkyObjects car même s'ils ne sont pas encore implémentés dans SkyObjects, nous y avons déjà réfléchi et fait quelques éléments de conception.

1.1) Le pointage d'objet

L'une des fonctionnalités les plus importantes à ajouter est sans conteste le pointage d'objet. Déjà afficher comme « fonctionnalité à venir » depuis la description des objets, cette fonctionnalité permettra de pointer de manière assistée les objets recherchés dans SkyObjects. Cette fonction utilisera alors le gyroscope et l'accéléromètre du Smartphone afin d'indiquer à la personne dans quel sens se déplacer afin de pointer l'objet. L'utilisateur, en suivant les instructions à l'écran finira par pointer exactement l'objet. Si l'objet n'est pas visible à l'oeil nu, il ne restera plus à la personne qu'à pointer son télescope dans la direction indiquée par son terminal Android. Du côté de la conception, c'est un algorithme qui, selon les données du gyroscope et de l'accéléromètre, calcule la position pointée et la compare à la position de

l'objet à pointer. En fonction de ces calculs, une flèche apparaît à l'écran pointant la direction vers laquelle doit être déplacé le terminal Android.

En bref, cette fonctionnalité est très intéressante en ajoutant une fonctionnalité attractive à l'application.

1.2) L'ajout de catalogues

Nous l'avons vu, le catalogue des objets NGC est ajouté par défaut dans la base de données. Une fonctionnalité à ajouter serait la gestion de catalogue et la possibilité d'en ajouter. Ainsi, il sera possible d'ajouter via le menu de configuration les n objets de tel ou tel catalogue. Lorsqu'un nouveau catalogue serait ajouté, c'est en fait une nouvelle base de données qui serait téléchargée (contenant le catalogue) et celle-ci serait fusionnée avec la base de données déjà présente dans le téléphone ou la tablette.

Cette fonctionnalité serait donc utile pour les astronomes qui par exemple utilisent plus un catalogue qu'un autre et, le téléchargement de ce catalogue permettrait pour cette personne de ne plus faire de requêtes sur les serveurs en ligne mais posséderait toutes les informations en local et n'aurait donc plus besoin d'internet.

Pour ce qui est du moteur de mise à jour, il serait légèrement modifié afin de mettre à jour tous les catalogues. Il y aurait donc une URL de mise-à-jours pour chaque catalogue et en fonction des catalogues téléchargés dans le terminal, l'application se chargerait d'appeler les URL correspondantes.

1.3) Le mode éducation

Le mode éducation est la dernière fonctionnalité à rajouter dont nous allons parler. Cette fonctionnalité permettrait depuis le menu de configuration, d'activer ce mode. L'intérêt de cette fonctionnalité serait de changer les différentes vues de l'application afin de les rendre moins professionnelles. Ainsi, les informations les plus complexes ne seraient plus visibles, une photo de l'objet y serait placée à la place, etc. Autre modification, étant donné que les non-professionnels ne connaissent pas les noms scientifiques des objets (exemple : M31 pour Andromède), la recherche serait également modifiée afin d'être compatible avec les noms courants. Ce nom courant est stocké dans le champ lng de la base de données. Ainsi, si

l'application est en français, il sera possible de rechercher M31 en tapant « Andromède » dans le champ de recherche.

Le mode éducation est la fonctionnalité qui à le moins été discutée pour le moment, ainsi, seul le champ dans la base de données a déjà été ajouté mais pour ce qui est du reste, quasiment tout reste à définir.

1.4) Publication

La publication sur l'Apple Store et sur Google Play reste bien sûr quelque chose d'indispensable pour SkyObjects mais pour le moment et dans le cadre du stage, il n'y est toujours pas d'où son classement dans le futur de l'application. La publication de SkyObjects devra être bien réfléchiée pour ne pas avoir de mauvaises notations. En effet, des personnes ayant peu de connaissances en Astronomie pourraient télécharger l'application et ne comprendraient pas forcément les informations affichées. En conséquence, ces personnes posteraient de mauvaises notes à l'application... La mise en ligne viendra donc plus tard une fois que l'application sera prête et que la page à afficher sur le store sera rédigée et très bien réfléchiée.

II / Mon futur à l'Observatoire

Grâce à ma rigueur et à mon travail appliqué au sein du CDS, M. Schaaff a décidé de prolonger mon travail au sein de l'équipe pour presque deux mois supplémentaires. Du 18 juin au 10 août, je serai donc embauché en CDD en tant qu'AI (Assistant Ingénieur). Durant la première semaine de ces 8 supplémentaires, je finaliserai l'application SkyObjects en y corrigeant tous les problèmes restants et j'ajouterai la fonctionnalité du pointage des objets. De plus, je m'occuperai également de sa mise en ligne sur les stores officiels.

Par la suite, durant les 7 semaines suivantes, deux situations sont possibles : soit j'arrive à comprendre rapidement le code de l'application Android SkySurveys (atlas du ciel sur Android avec l'utilisation de OpenGL et des sphères HEALPIX) et je continue son développement, soit je reprendrai un travail en HTML5 toujours basé sur le service Aladin et

donc l'atlas du ciel. Dans tous les cas, je suis extrêmement satisfait de pouvoir continuer à travailler au sein du CDS.

Conclusion

Nous avons réalisé dans ce rapport une présentation de l'Observatoire astronomique de Strasbourg, des équipes de recherche qui le composent et du fonctionnement de chacune. Il a d'autre part été présenté les différents services existants et avec lesquels nous nous sommes familiarisés au cours des premières semaines de stage, que ce soit les services propres au CDS (Simbad, VizieR, Aladin et Sesame) ou externes mais jugés utiles au développement de notre projet (Ned, Eclipse, etc.). Puis nous avons abordé le sujet et les objectifs précis de notre travail ainsi que les différentes étapes qui se sont avérées nécessaires pour son aboutissement : BDD, moteur de recherche, interface utilisateur, moteur de mise à jour, prototype en HTML5, etc.

A ce jour, le prototype en HTML5 reste très simpliste dans sa réalisation mais nous a permis d'apprendre suffisamment de choses pour en faire une étude et le comparer à l'application Android. SkyObjects, quant à lui, est dans une version qui pourrait être déposée sur le Google Play. Les améliorations possibles et l'avenir de ces applications a fait l'objet d'une réflexion.

Le stage est désormais terminé et celui-ci fut pour moi très riche en apports. J'ai réussi à remplir les objectifs qui m'étaient donnés en réalisant notamment une application Android se basant sur le modèle du service Sesame. Même si ce développement ne fut pas de tout repos, j'ai toujours fini par trouver les solutions aux différents problèmes, et l'aboutissement du travail représente pour moi une grande satisfaction.

Le stage fut riche en apport car il a fallu que je réalise un projet plus conséquent que ce que l'on peut faire à l'IUT. De plus, mon stage n'a pas été basé que sur de la programmation mais était constitué également d'études, notamment pour les confrontations SkyObjects/prototype HTML5. L'apport de connaissances fut très important pour moi car je ne connaissais ni Android, ni HTML5 et ni JavaScript. De plus, j'ai depuis longtemps une passion pour l'astronomie et faire ce stage en ces lieux a donc été pour moi un grand bonheur et cela m'a même permis d'acquérir de nouvelles connaissances dans le domaine. Ensuite, une autre satisfaction est le fait d'avoir travaillé dans une équipe qui est reconnue internationalement. La satisfaction fut même encore plus grande lorsque M. André Schaaff

nous a dit qu'il allait montrer notre travail lors de l'IVOA Interop. Enfin, travailler en ces lieux m'a permis de faire connaissance avec de nouvelles personnes. Pour ne citer qu'eux, il y a Julien (mon collègue stagiaire de travail) et M. André Schaaff avec qui je me suis toujours bien entendu et qui m'a très bien encadré durant ce stage, et bien évidemment tout le reste de l'équipe du CDS.

Références bibliographiques

- **tutomobile.com** : Site français contenant de nombreux tutoriels sur le développement Android et iOS
- **developer.android.com** Documentation officielle d'Android et contient notamment l'API Java d'Android et des tutoriaux.
- **stackoverflow.com** Forum anglophone dédié à la programmation, son avantage incontestable est d'être très actif.
- **forum.frandroid.com** Forum francophone dédié à Android, avec une partie consacrée au développement d'applications.
- **Programmation Android. De la conception au déploiement avec le SDK Google Android 2.** Damien Guignard, Julien Chable, Emmanuel Robles. Eyrolles édition.
- **dev.w3.org/html5/html-author/**
- **alsacreations.com**
- **ivoa.net/internal/IVOA/InterOpMay2012Applications/IVOA2012-Urbana-Application-CDS-AS.pdf** : Présentation de l'application SkyObjects fait par M. André Schaaff à Urbana Champaign lors de l' IVOA Interop.

Annexes

Table des annexes :

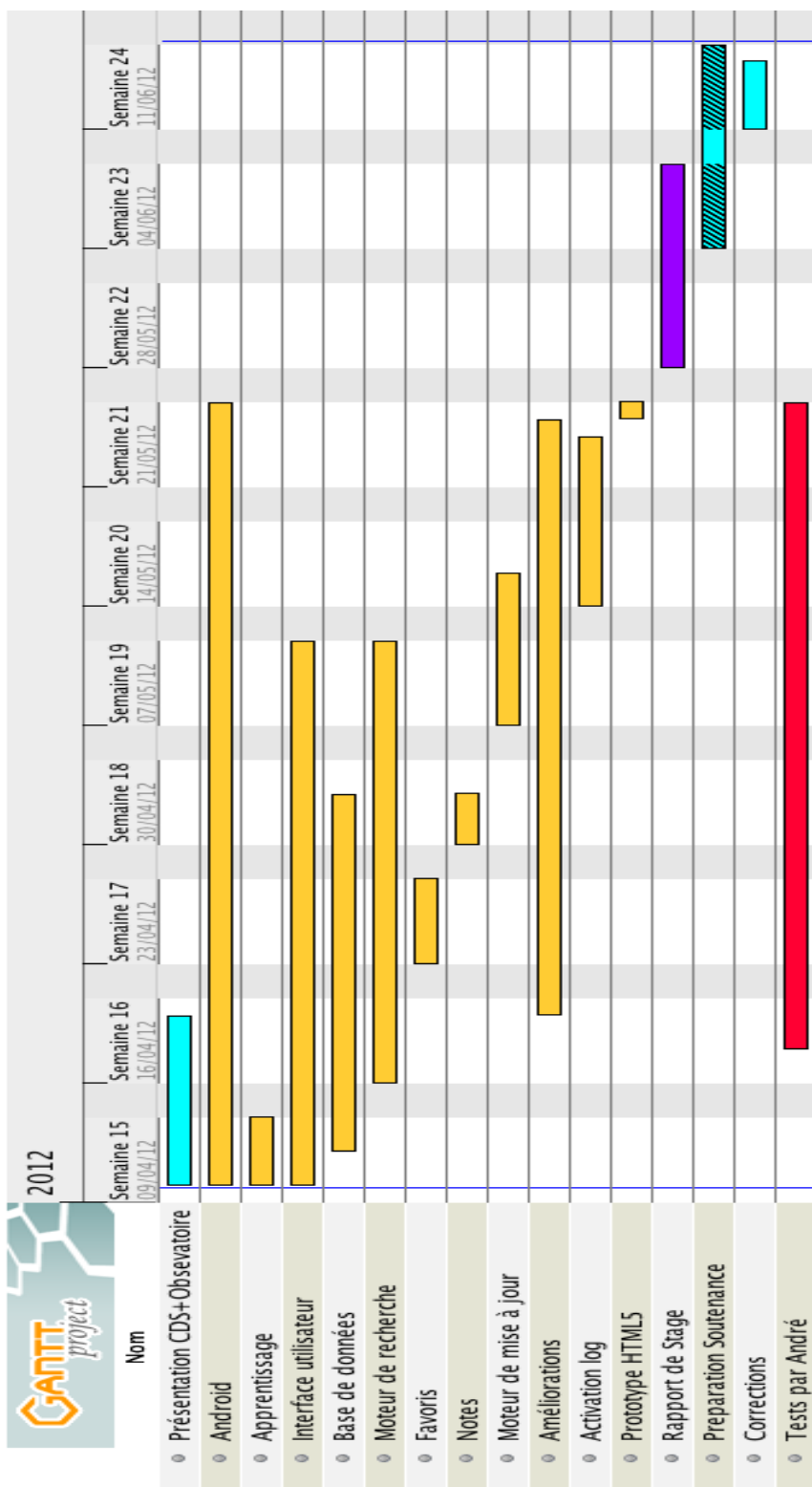
A.1) Diagramme de Gant

A.2) MCD

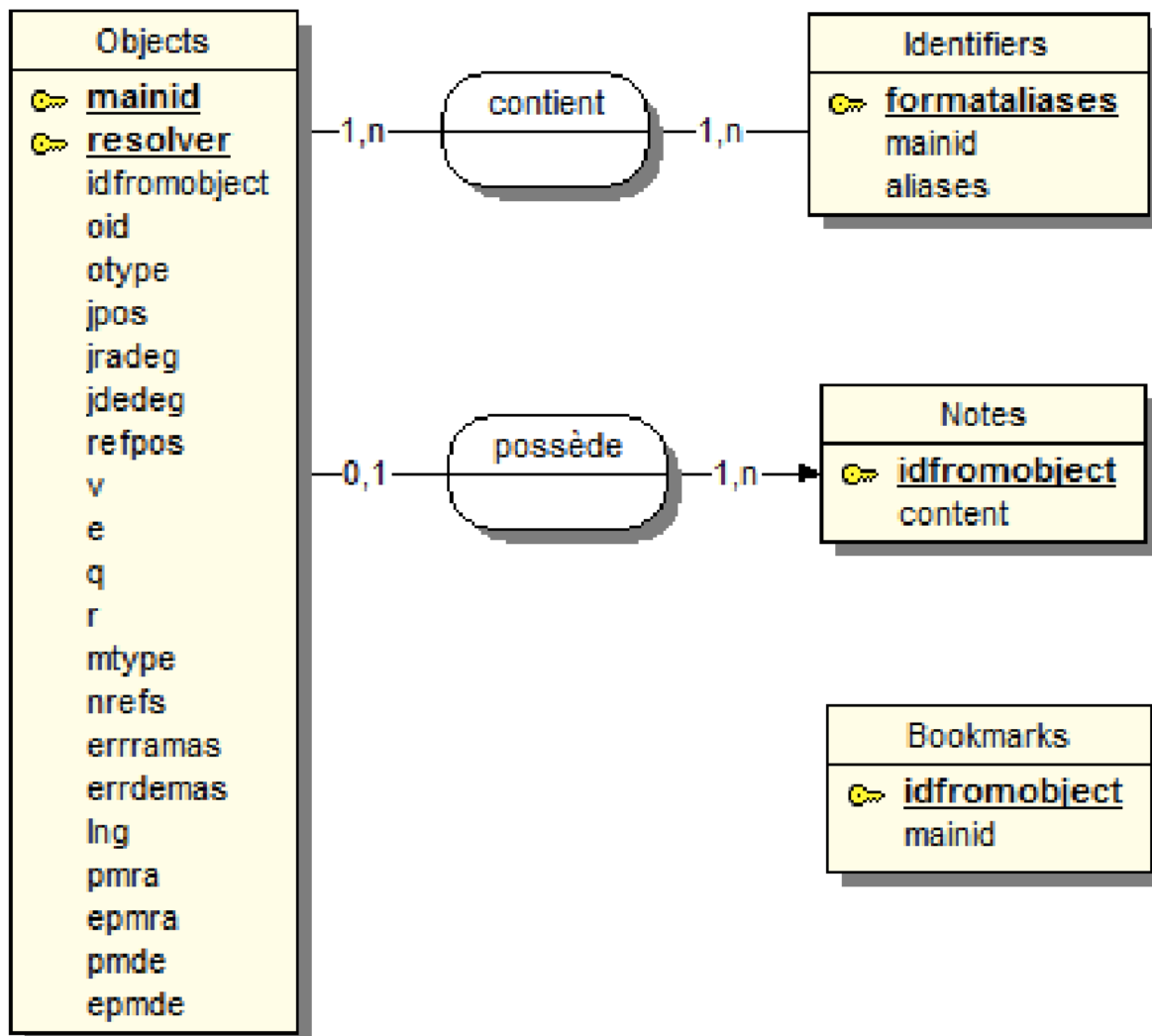
A.3) Schéma du moteur de recherche

A.4) Captures d'écrans de SkyObjects

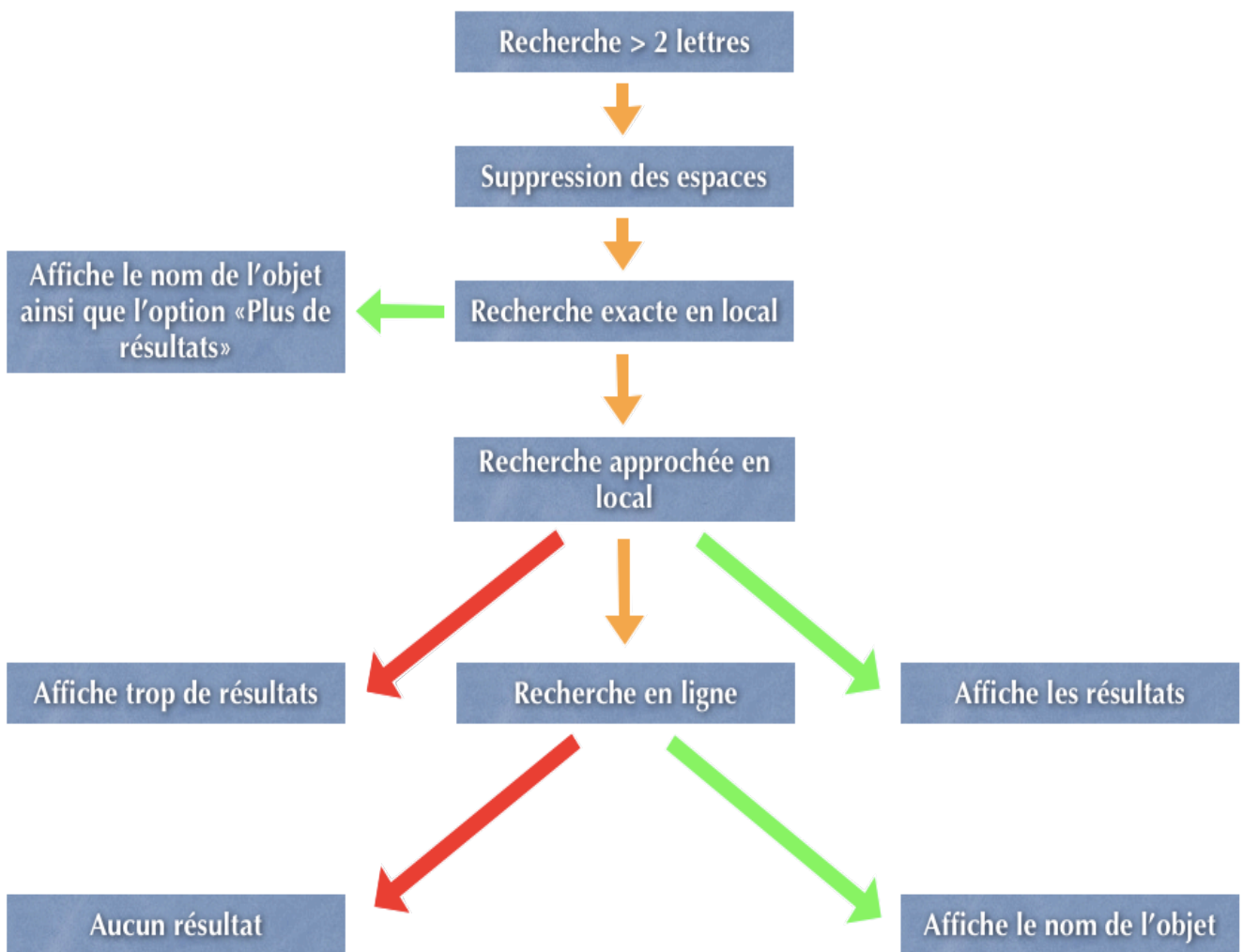
A.1) Diagramme de Gant



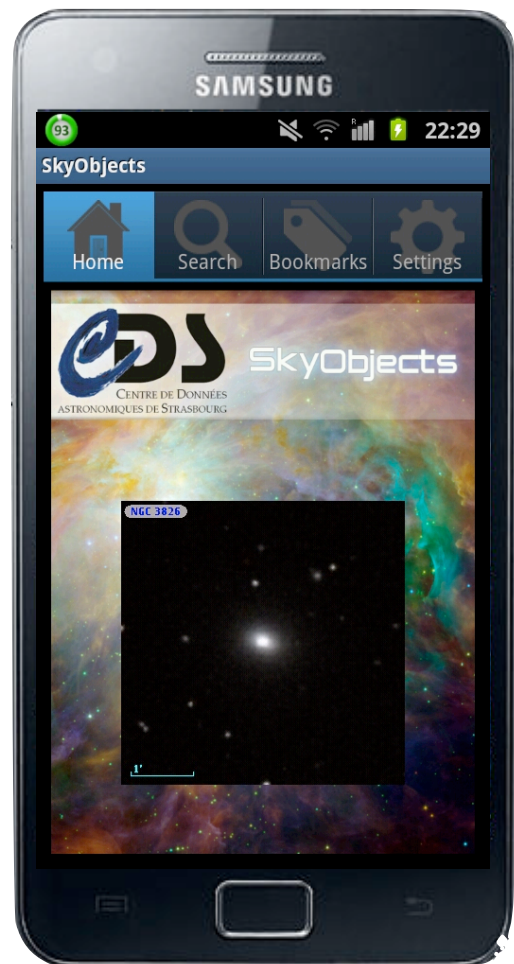
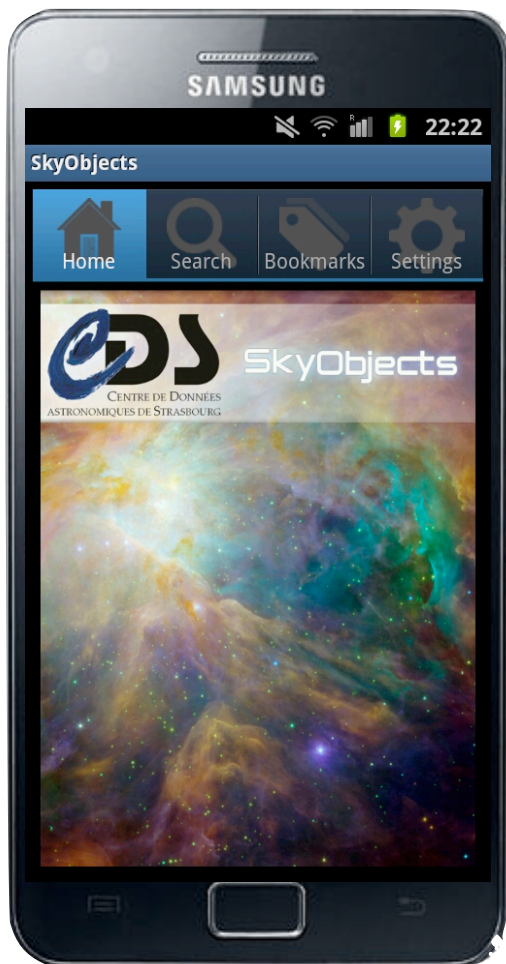
A.2) MCD

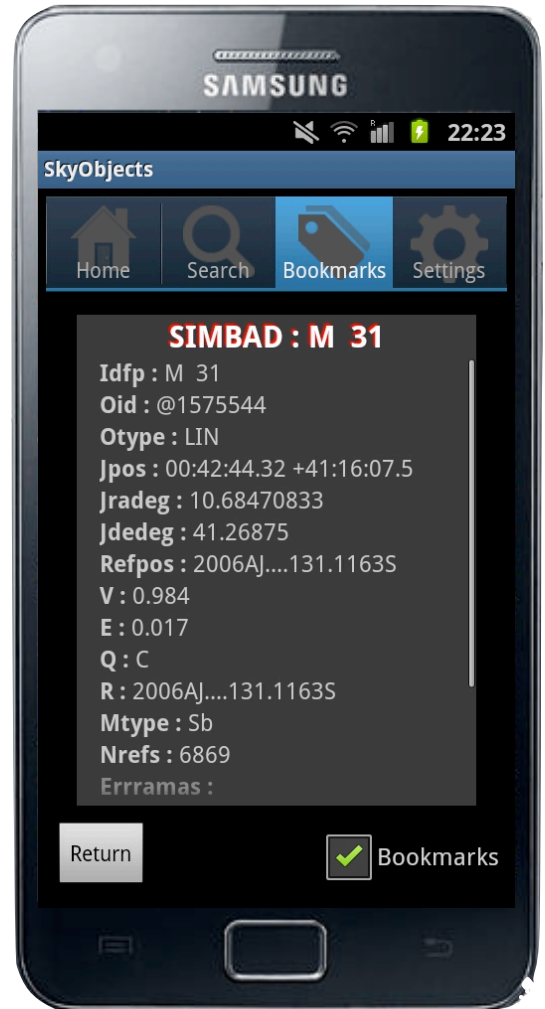
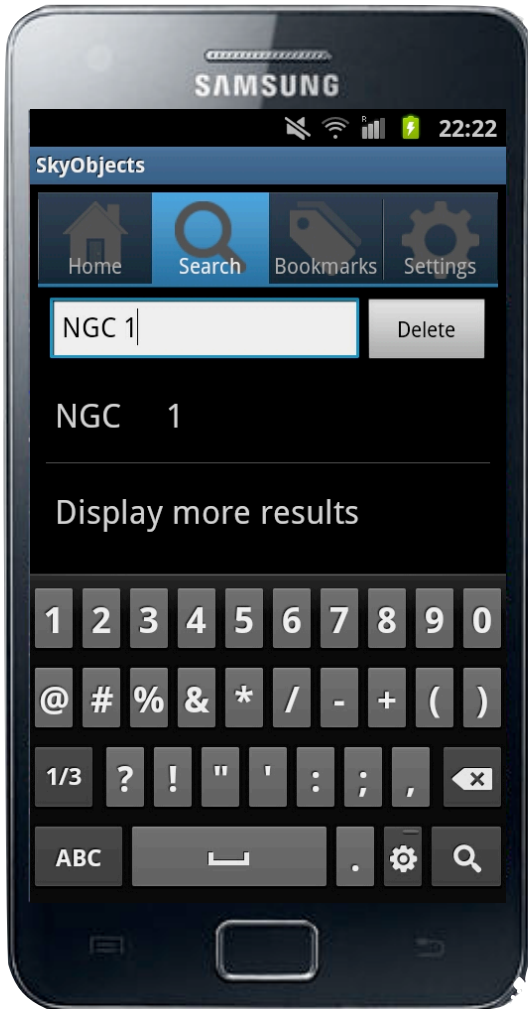


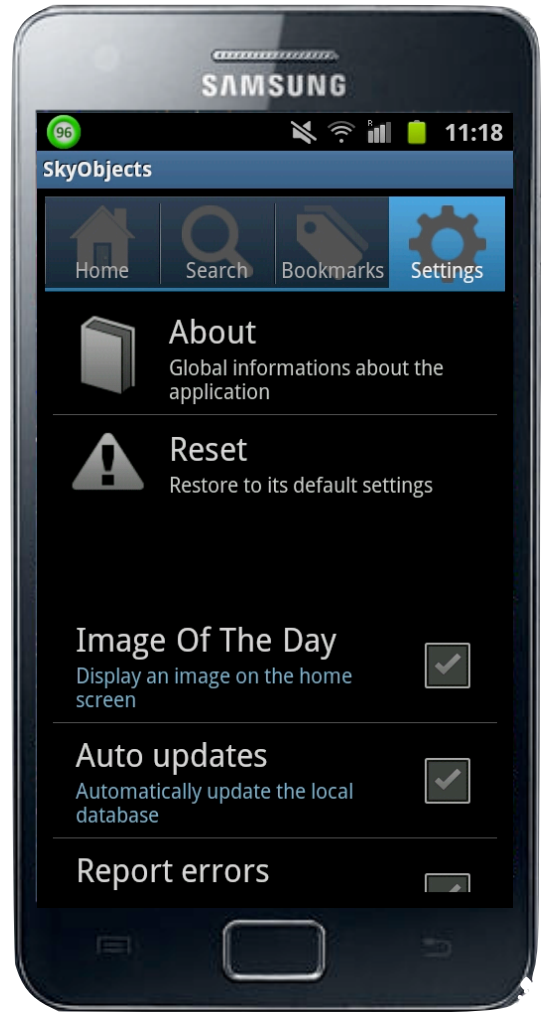
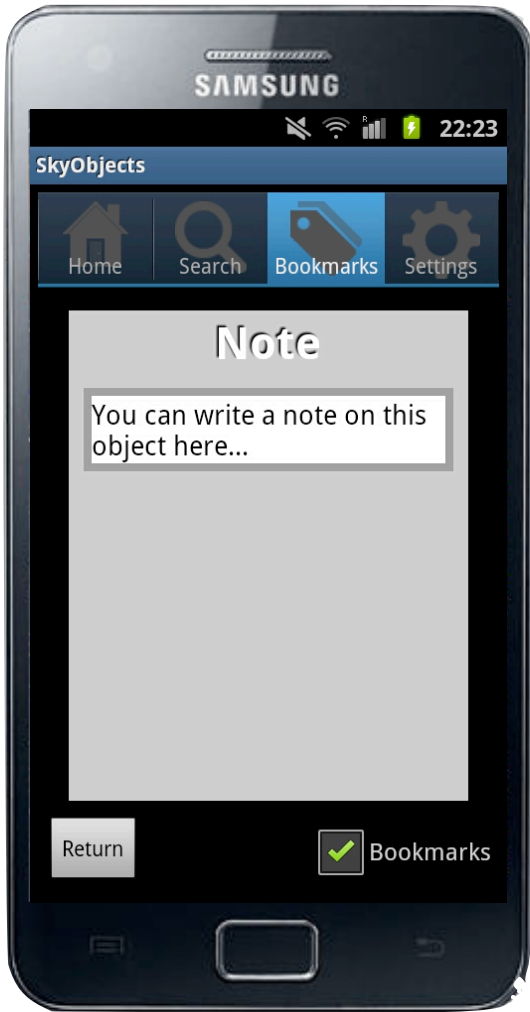
A.3) Schéma du moteur de recherche



A.4) Captures d'écrans de SkyObjects







FICHE RAPPORT DESTINEE A LA BIBLIOTHEQUE

RAPPORT CONFIDENTIEL ET NE DEVANT PAS FIGURER A LA BIBLIOTHEQUE :

Non

NOM ET PRENOM DE L'ETUDIANT : HOUPIN Romain

DUT : INFORMATIQUE

S4 OS

TITRE DU RAPPORT : SkyObjects, développement d'un Name Resolver sous Android et prototypage en HTML5

Nom de l'Entreprise : Centre de données astronomique de Strasbourg

Adresse :

Observatoire Astronomique de Strasbourg
11, rue de l'université
67000 Strasbourg

Type d'activité (domaines couverts par l'entreprise) : Enseignement et recherche

Nom du parrain (enseignant IUT) : Samuel Cruz-Lara

Mots-clés (sujets traités) : Android, HTML5, Application, JavaScript, PhoneGap, Astronomie

Résumé

L'objectif du stage était de développer une application native sous Android et d'en faire un prototype en HTML5 afin de comparer par la suite les deux versions. Ce rapport présente donc l'entreprise avec son existant puis le développement de ces applications et enfin, parle de la comparaison entre le développement natif et en HTML5.