

# Nouvelles interfaces au CDS : retour sur quelques développements

Journée nouvelles interfaces OV France  
26 novembre 2012

# Plan

1. Version mobile du portail CDS

2. Aladin «HTML5»

3. Widgets modulaires

En production  
En développement  
Premiers tests

# I. Version mobile du portail CDS

- 3 options :
  - applications mobiles (AppStore, Android Store, etc)
    - + visibilité (app stores), meilleures performances, look'n'feel natif
    - - chronophage, besoin de beaucoup de ressources (un développement par plateforme)
  - site web dédié
    - + rapide à mettre en oeuvre
    - - 2 URLs, détection des clients mobile pour redirection pas d'accès aisé aux fonctions natives des devices (GPS, caméra)
  - site web avec *responsive design*

# jqTouch

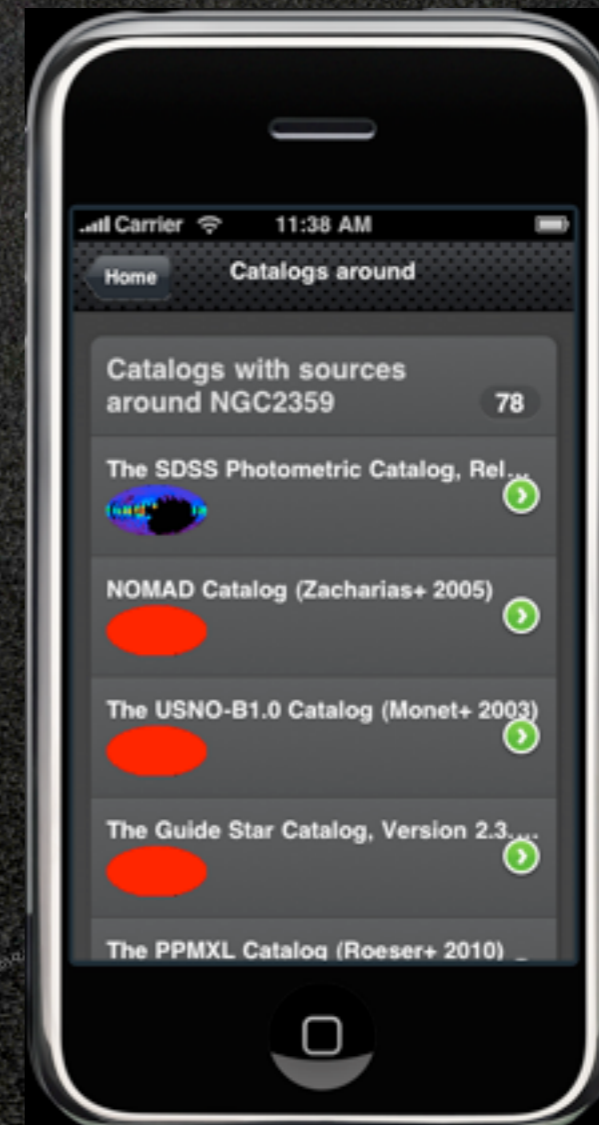
- <http://jqtouch.com/>
- plugin jQuery pour le développement Web mobile
- Fournit un ensemble de composants :
  - listes, boutons, formulaires
  - animations Webkit
  - détections des événements *touch* : *swipe* (balayage), *tap*
- adapté aux développements web pour *smartphones*
- optimisé pour navigateurs basés sur Webkit (iOS/Android)

# Alternative possible : jQuery mobile

- <http://jquerymobile.com/>
- Framework pour développement Web sur *smartphone* **et** *tablettes*
- Plus lourd que jqTouch, mais plus complet et plus général
- Autres frameworks :  
<http://www.markus-falk.com/mobile-frameworks-comparison-chart/>

# Démo portail mobile

- <http://cdsportal.u-strasbg.fr/mobile/>



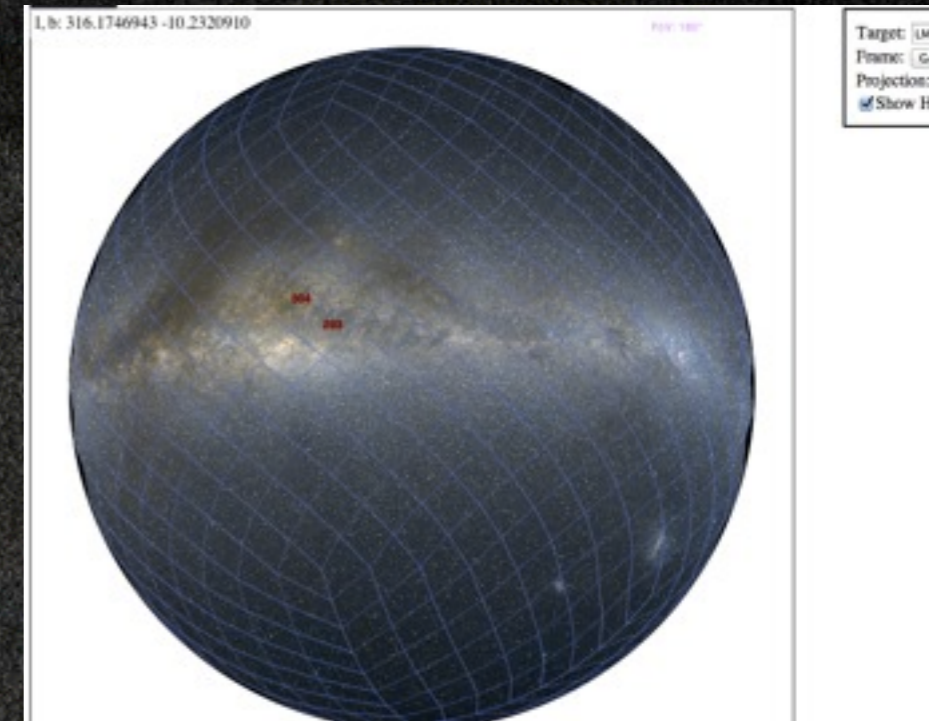
# 2. Aladin «HTML5»

- **Objectif :**

- développer une version légère d'Aladin, tournant dans le navigateur

- **Motivations :**

- perte de vitesse de la technologie *applet Java* et des plugins browser en général (Flash, Silverlight, ...)
- l'amélioration des moteurs Javascript et l'introduction de nouvelles technologies (HTML5, canvas) permettent le développement de véritables applications s'exécutant nativement dans le navigateur
- ouverture vers le monde des tablettes



# Rappel sur HTML5

- Ensemble de nouveaux standards pour le Web
  - nouvelles APIs Javascript
  - nouveaux éléments HTML
  - CSS3
- Introduction à HTML5 : <http://slides.html5rocks.com>
- Tester le support HTML5 de son browser : <http://html5test.com/>
- Support des fonctions HTML5 par navigateur : <http://caniuse.com/>



# Canvas 2D vs. WebGL

## ● Canvas 2D



- Développement classique : drawLine, drawImage, ...
- Supporté par la quasi totalité des navigateurs récents (y compris mobiles)



- dédié à la 2D  
--> 3D : gestion manuelle
- performances très variables selon environnement et navigateur

## ● WebGL



- Excellentes performances (sous-ensemble d'OpenGL)
- Accélération matérielle



- Une autre façon de développer : scene, render, objets, camera
- Pas de support sur IE
- Pas de support sur iOS
- Pas de support sur machines anciennes ou aux drivers datés

# Three.js

- <https://github.com/mrdoob/three.js/>
- Librairie Javascript pour la création d'animations 3D
- abstraction du renderer : WebGL ou canvas
- mais abstraction incomplète :
  - le rendu *canvas 2D* des particules est buggé
- Simplifie grandement le développement WebGL
- Tutorial three.js :  
<http://aerotwist.com/tutorials/getting-started-with-three-js/>

# Aladin HTML5, développements en cours

- Prototype reposant sur canvas 2D
  - Gestion « à la main » des projections, zooms, déplacement
- Difficultés :
  - Conversion librairie HEALPix Java en Javascript
  - Conversion librairies projection et conversion de coordonnées
- Challenges :
  - performances raisonnables sur le maximum d'environnements  
( <http://www.html5rocks.com/en/tutorials/canvas/performance/> )
  - gestion du cache des images à afficher

# 3. Widgets modulaires

- « **iGoogle pour l'astronomie** »
  - Précurseur : ASCOT - AStronomical COllaborative Toolkit  
Andrew Connolly, ADASS 2010  
[http://adass2010.cfa.harvard.edu/ADASS2010/incl/presentations/O05\\_2.pdf](http://adass2010.cfa.harvard.edu/ADASS2010/incl/presentations/O05_2.pdf)
- Proposer un ensemble de widgets s'exécutant dans le navigateur
  - partageables
  - inter-connectables
  - intégrables dans une page Web
- Chaque widget répond à un besoin de base :  
**chercher** un service, **interroger** un service, **visualiser** des données, etc
- Un utilisateur peut créer son propre espace de travail à partir des widgets de son choix

# Standards pour les widgets

- OpenSocial : standard de fait
  - Développé par Google et MySpace pour contrer la plateforme Facebook
  - Soutenu par de nombreux acteurs :
    - réseaux sociaux : Orkut, LinkedIn, Viadeo
    - services : IBM, Oracle, Cisco Systems
- W3C widgets
  - A l'état de draft

# Prototype

- Utilise le *container* Apache Rave
  - Intègre Apache Shindig (*container* OpenSocial)
  - Intègre Apache Wookie (*container* W3C widgets)