du web au mobile

quelles différences ?

Qui suis-je ?

- Hy-tsoung Chang
- Ingénieur R&D et Software, touche-à-tout
- Depuis 2004.
- Startups Internet : Exalead, Criteo...
- Dernier poste, Edition de logiciels Desktop & Mobile, Dashlane.

Pourquoi cette présentation

- Internet : Boum économique et technologique
  - Essor depuis : 1995-2000
    - 1ère bulle : 2000 ; 2ème bulle : 2010.
- Smartphones : arrivée d'un nouveau boom.
  - iPhone en 2007 ; SDK Développeurs 2008.
- Nouvelles technologies, usages, et Business.

Web vs Mobile

- Evoquer :
- Les technologie(s)
- Des métriques de projet
- Des Aspects Produit

Agenda

- Présentation en 4 points:
  - Programmation
  - Mise en production
  - Conception produit, Design UI & UX
  - Marketing & Business

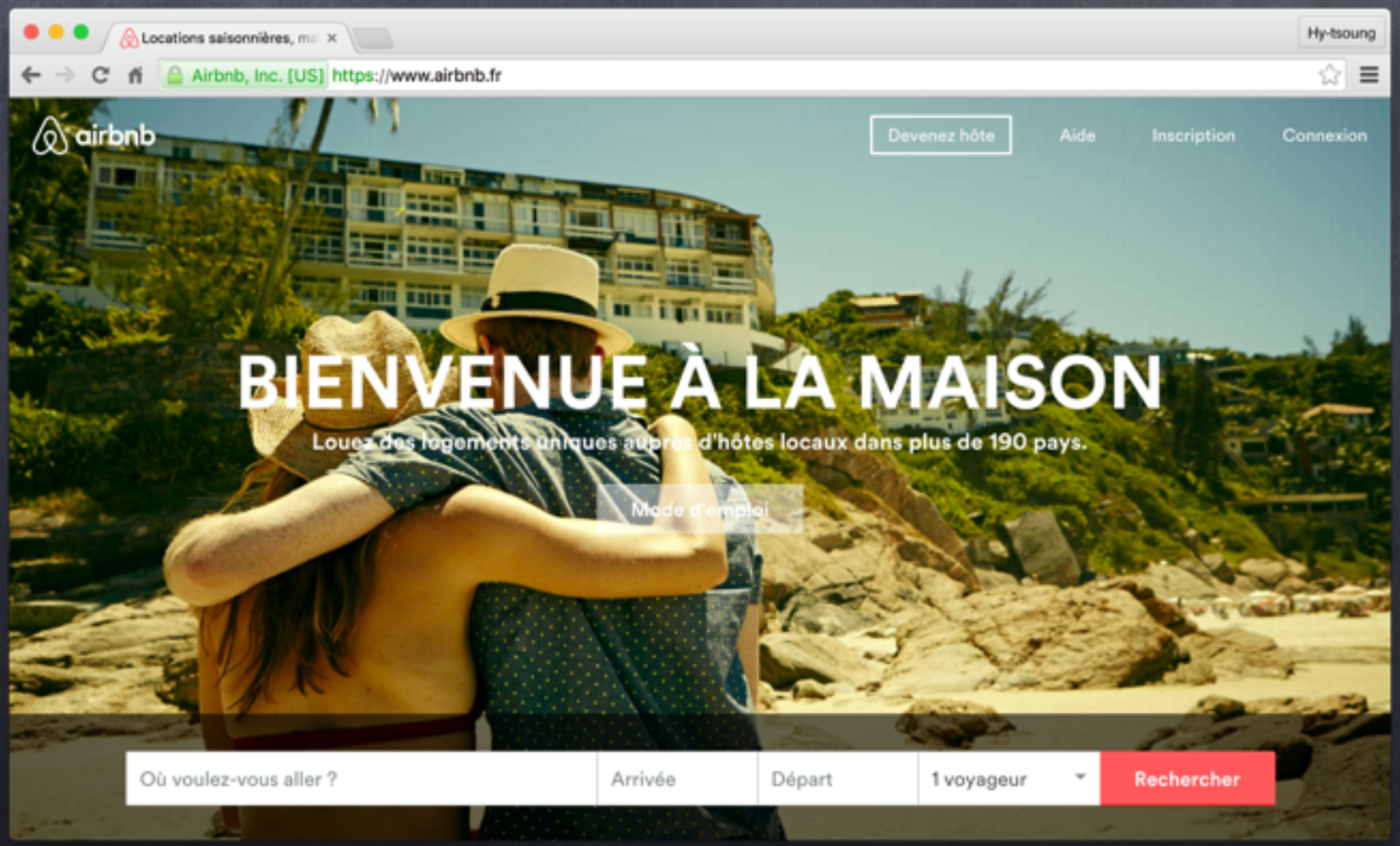
Cadre

- Objectifs du Développement mobile:
  - faire une application iOS ou Android (*)

(*) ReactNative, PhoneGap ou autres hybrides : pas directement abordés...

Cadre

- Objectifs du Développement web:
  - faire un site ou une application web

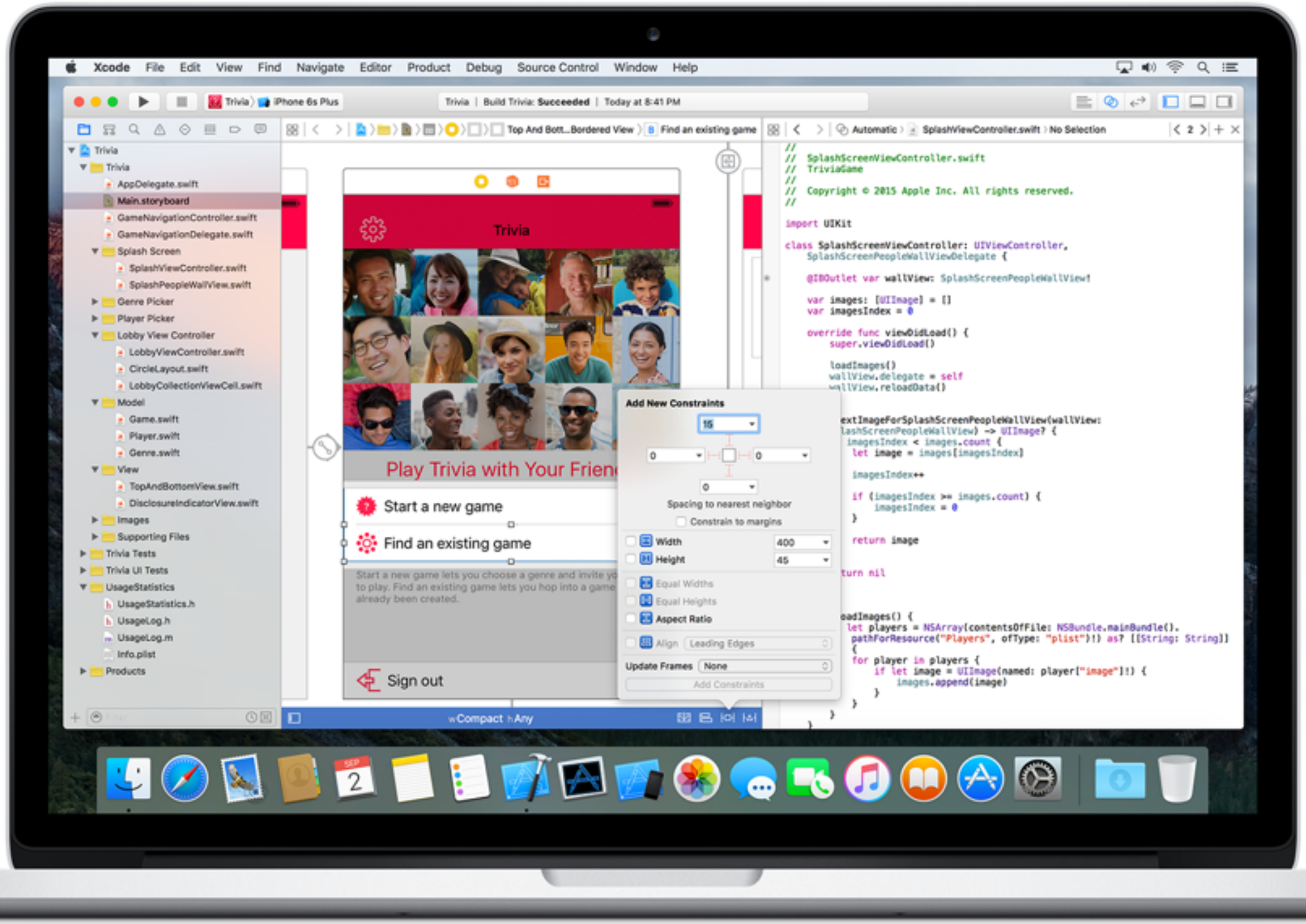


Programmation

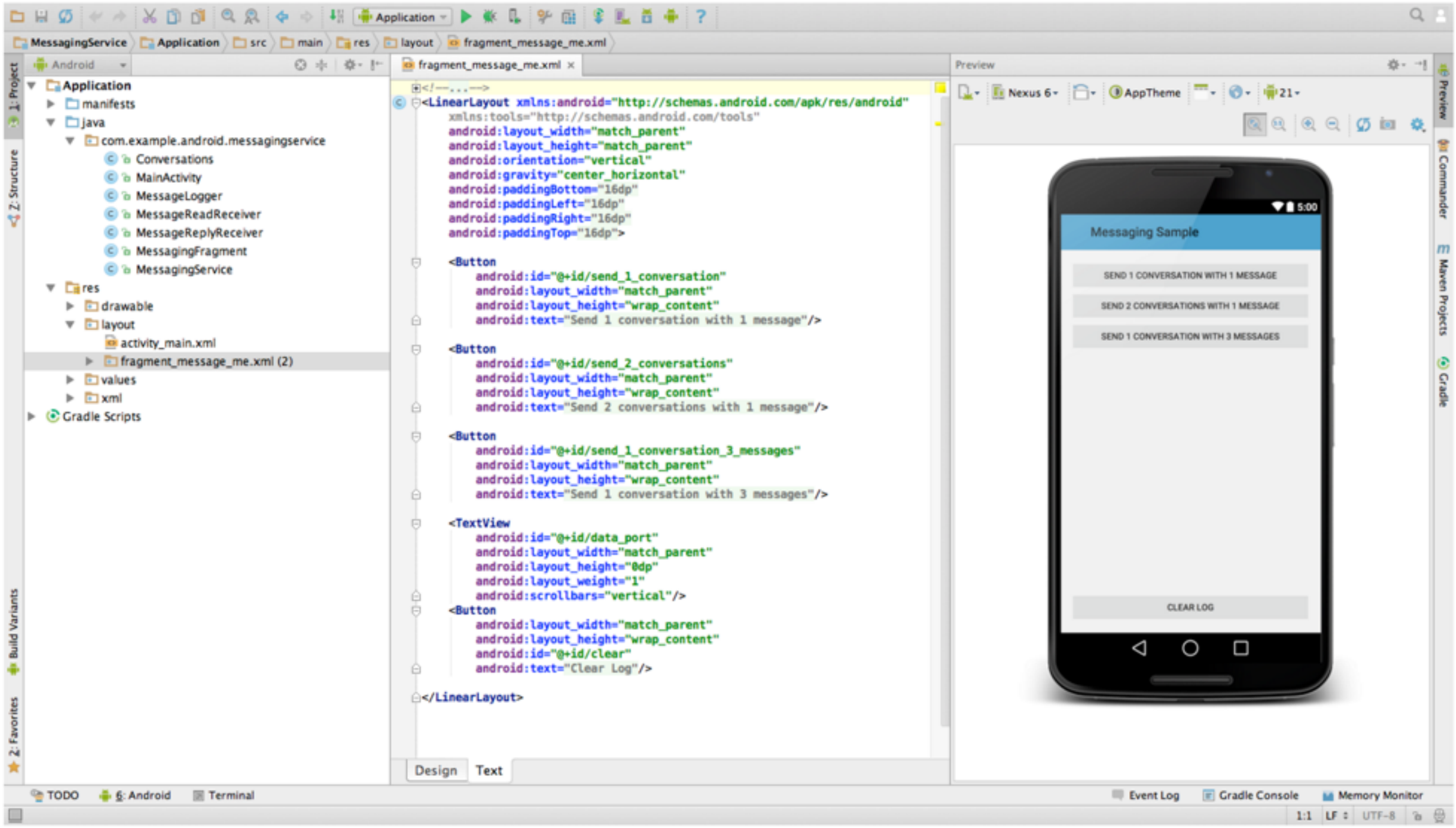
Outils de développement

- Mobile
  - IDE dédiés : Xcode ou AndroidStudio.
    - Autocompletion avec correction de syntaxe, aide intégrée, debugger
    - Templates, refactoring, traceurs. etc.
- Web
  - Editeur de texte avancé.

iOS : Xcode (Mac)

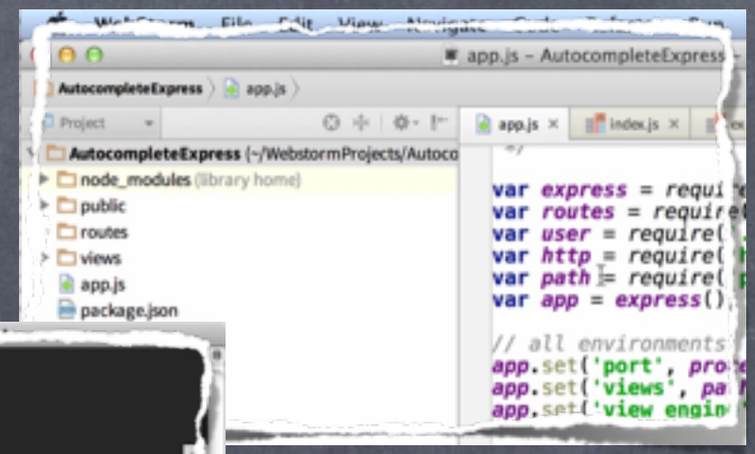


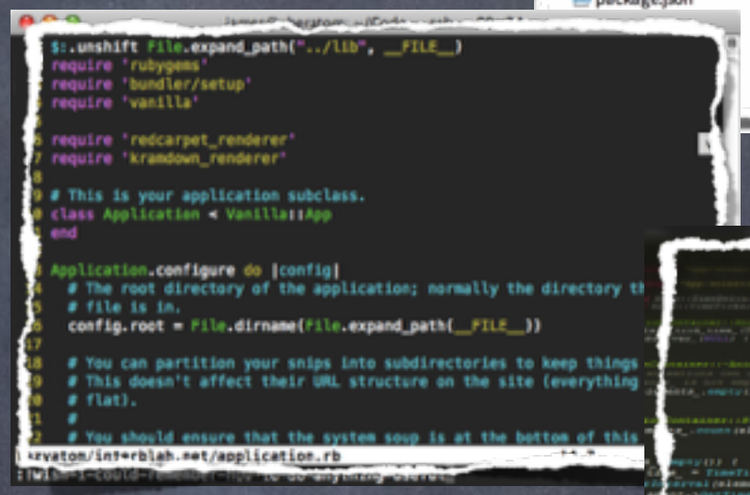
Android : AndroidStudio

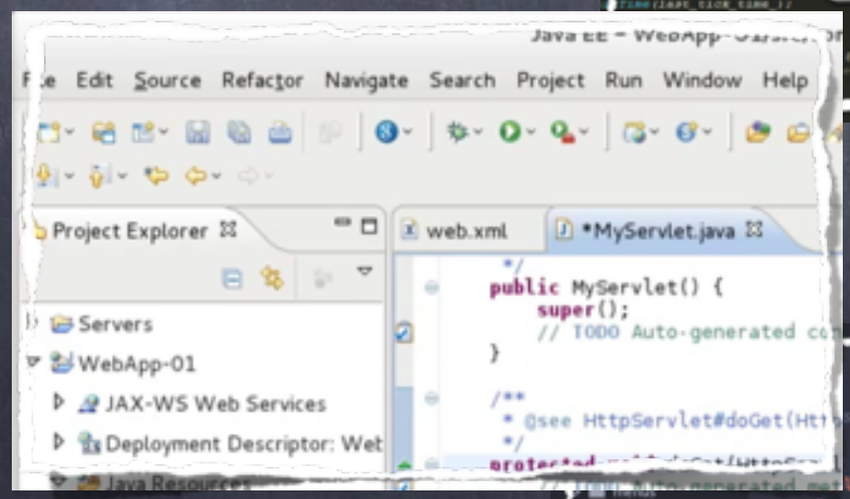


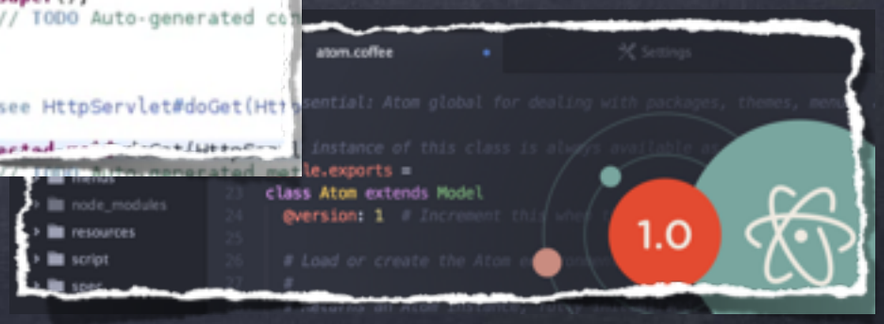
Développement web

- L'environnement dépend des gens... !
- Sublime, Atom, VI, Eclipse, WebStorm (JetBrains), ... etc.









Coûts de licences ?

- Mobile
  - iOS : 99 USD ; 70% revenus des ventes ; avoir un Mac.
  - Android : 25 USD ; 70% revenus des ventes
- Web
  - Open source principalement.
  - Coûts des serveurs, et hébergement

Elément de base pour construire un "écran"

- Mobile
  - iOS: code objective-c ou swift + xib/storyboard
  - Android: code java + .xml
- Web
  - Code html + css + javascript

Hello World

- example version mobile iOS (objective-c)

```
@interface HelloWorld : UIView
@property UILabel *myLabel;
@end

@implementation
- (id)initWithFrame:(CGRect)frame {
    self = [super initWithFrame:frame];
    if (self) {
        _myLabel.text = @«Hello World!»;
        [self addSubview:_myLabel];
    }
    return self;
}
@end
```

Hello World

- example version web

```
<html>
  <body>
    <h>Hello World!</h>
  </body>
</html>
```

Outils WYSIWYG

- Mobile :
  - iOS : éditer le storyboard via l'interface Xcode builder + drag&drop vers le .m/.swift
  - Android : éditer le .R avec interpréteur wysiwig/xml
- Web :
  - pas d'outil WYSIWYG « approuvé » par les développeurs

Principaux Languages

- Mobile:
  - Objective-c, Swift, Java-android.
  - langage orienté objet, SDK officiels et très structurés.
  - Paradigmes imposés
- Web:
  - Html, Javascript, Php, ...
  - langages de scripting, sans compilation, non-typé.
  - mais de plus en plus de structuration dans le javascript et Php 6.

SDK et frameworks

- Mobile:
  - SDK définis par Apple ou Google Android. Ils sont la référence.
- Web:
  - De nombreux frameworks open-source. Adoption par la communauté au fur et à mesure

Les acteurs de l'Ecosystème et conséquences

- Acteurs du Mobile : Google et Apple.
  - Répercussions : Une sortie majeure d'OS tous les ans. Beaucoup de nouvelles choses en général. Nécessité de les intégrer. Il faut suivre.
- Acteurs du Web: les développeurs, la communauté web.
  - Répercussions : Nouveaux frameworks tous les jours. Beaucoup de nouvelles choses en général. Il ne faut pas suivre...
  - (propositions d'améliorations)

Comment on développe et teste ?

- Mobile :
  - simulateur, émulateur ou vrai device
  - coder, compiler, déployer : ordre de grandeur de qq ~min.
- Web :
  - navigateur web
  - coder, rafraîchir la page : ~sec

Tenir compte des différentes « plateformes »

- Mobile :
  - iOS : ~3 iOS, ~10 devices. Relativement maîtrisé.
  - Android : beaucoup plus de disparités !
    - Obsolescence matérielle rapide et mises à jour annuelles.
- Web :
  - Facteurs : hardware, navigateurs et OS
  - Délai admissible avant mise à niveau obligatoire : ~ 99. ans.
- Fragmentation des plateformes : Web > Android > iOS

Beta-tester

- Mobile:
  - Android: Beta Channel du Playstore, assez facile.
  - iOS: TestFlight, HockeyApp ou autre, pas facile quelquefois à cause de limitations Apple (nombre de users, devices enregistrés, versions à signer)
- Web:
  - Serveur sous contrôle interne
  - site de test sous port dédiés, loadbalancing, re-routage, beta privée...

Mise en production

Mise en production

- iOS : soumission AppStore et attente validation ; ~2 semaines.
- Android : soumission Android ; 30 min à quelques heures.
- Web : déployer le code sur le(s) serveur(s) ; immédiat

Déploiement progressif ?

- Android :
  - staged rollout possible aussi via Beta channel. Le Rollback correspond à une nouvelle mise à jour.
- iOS :
  - pas de staged rollout. Pas de rollback à moins d'enlever l'application de l'AppStore. Ça passe ou ça casse... Note : procédure d' « expedite review ».
- Web :
  - Staged rollout. Rollback possible.

Obtenir des feedbacks

- Mobile :
  - Crashreports
  - feedbacks utilisateurs
    - logs maison
- Web :
  - server logs
- => Actionabilité : Web > Mobile

Quels problèmes de scalabilité ?

- Mobile :
  - le problème se pose moins, car consommation ressources matérielles essentiellement sur devices, donc calibrage en amont.
- Web :
  - si trop de trafic : ajouter des nouveaux serveurs et re-architecturer.

Agenda

- Présentation en 4 points:
  - Programmation
  - Mise en production
  - Conception produit, Design UI & UX
  - Marketing & Business

Conception Produit

Utilisateur avec une connection réseau ou non ?

- Mobile : prévoir l'expérience offline, et/ou un réseau lent.
  - Ex: téléchargement au préalable et/ou en arrière plan.
  - Fonctionnalités accessibles offline
- Web : n'a pas de sens...

Communication multi-modale

- Interagir davantage avec l'utilisateur :
  - Mobile : push notifications, badges, swipes, accéléromètre, vibreur...
  - Web : notifications (via les WebSockets).

Accès à des données personnelles de l'utilisateur

- Mobile : interconnectivités possible
  - Carnet d'adresses, photos, localisation GPS, stockage local sur device
- Web : outils en émergence
  - Géolocalisation, WebStorage.

Design UI & UX

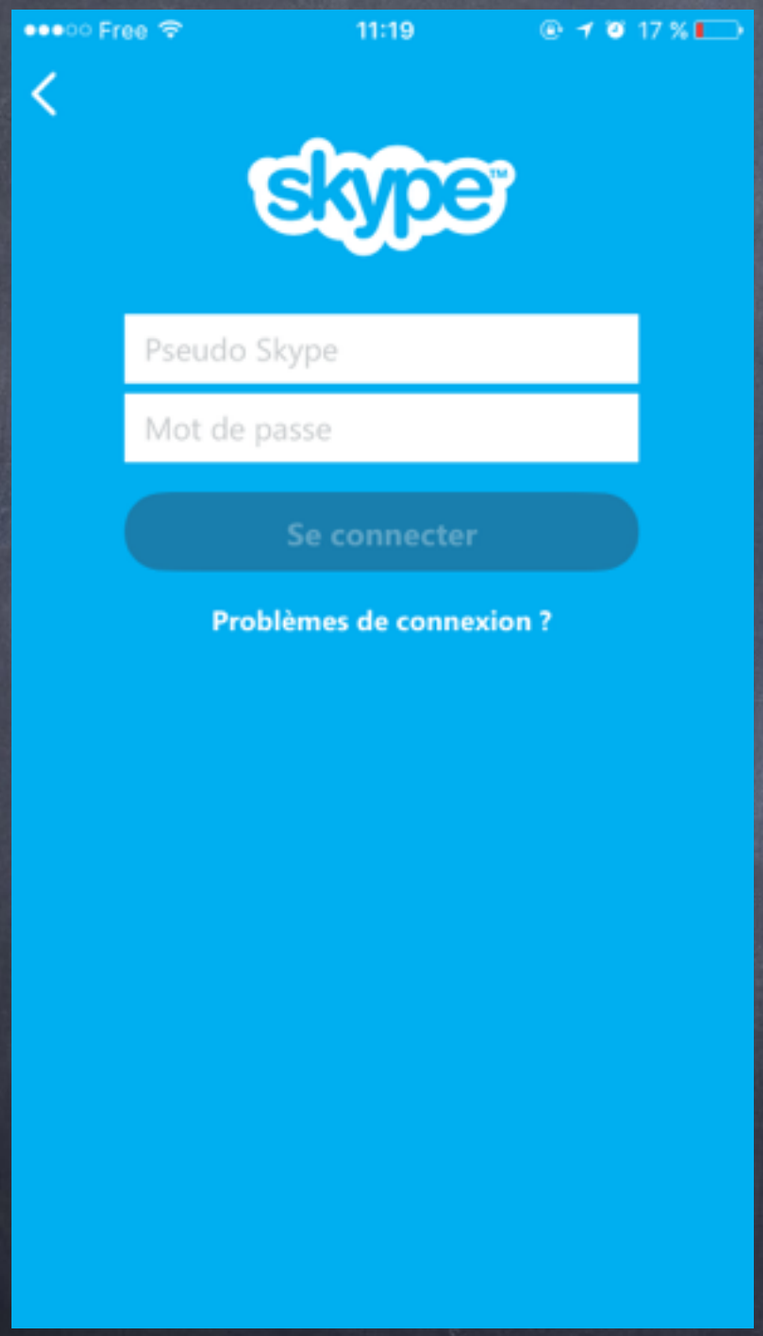
Design et identité graphique

- Mobile :
  - Adopter sa propre identité SANS s'éloigner trop des guidelines de Apple ou Android
  - Dur dilemme entre identité de la marque tout en collant à l'esprit de la plateforme
- Web :
  - Liberté, tant que l'ergonomie est là.
  - Site webs indépendants plus prescripteurs de tendance.

Gestion de l'espace

- Mobile : le défi des smartphones
  - peu d'espace
  - débats sur la conception d'une navigation ergonomique à travers l'app.
    - sous iOS pendant longtemps : coder des interfaces dynamiques était compliqué.
- Web : connaissances plus matures
  - gabarits assez standards
  - des designs en évolution régulière

Example du clavier virtuel



Appliquer un même design partout...

- Mobile : anticiper différentes tailles et résolutions.
  - retina ou non, smartphone/tablette, landscape/portrait, rotation
  - iOS : des lacunes au niveau SDK, en amélioration...
- Web : éventuellement envisager différentes tailles d'écran
  - interface « responsive »
  - taille de fenêtre variable de façon continue
- Note : standards et attentes plus élevées en général sous iOS.

Marketing Produit

Définition du positionnement produit

- Pour un même produit / marque,...
  - les utilisateurs n'ont pas forcément les mêmes attentes en allant sur le site web ou sur l'application mobile.
- ex: Interactivité ?
- ex: Contenu institutionnel ? Infos pratiques ?

Stickiness

- Mobile : quelques challenges
  - L'utilisateur doit d'abord télécharger l'app
  - L'utilisateur doit lancer l'application (mono-screen)
- Web :
  - Les utilisateurs ont le réflexe de la recherche Internet.
  - Navigateur multi-onglets, et ordinateur multi-fenêtres.

Leviers pour augmenter sa visibilité

- Mobile : Soigner sa visibilité sur les stores AppStore ou Google PlayStore
  - ses relations avec Apple et Android afin d'être dans l' « Editor's Choice »
  - Avoir de très bonnes reviews et une bonne note.
- Web : Soigner son référencement Google
  - liens entrants
  - mots-clés

Moyens de monétisation éventuels

- Mobile :
  - Applications à prix généralement bas (par rapport à application desktop) ou in-app purchases
- Web :
  - Souscription à des offres premium
  - Publicités (plus répandues que sur mobile)

Conclusion

Web vs Mobile

- Défi actuel : Création d'un nouvel usage à mi-chemin entre une application personnelle et l'univers Internet

Coûts de Développement

- Développement Mobile : beaucoup plus coûteux aujourd'hui
- Repartir de zéro alors qu'il existe un pôle de ressources et compétences matures en Web

Autres initiatives

- PhoneGap, Xamarin... : webview ou compilation interprétée.
- ReactNative : se base sur un language javascript commun et utilise les SDK natifs en-dessous.

Mobile : Bataille d'influence

- Acteurs économiques et technologiques avec un poids importants :
  - Android <- Google (!)
  - Apple
- Et des développeurs sous la dépendance de Google/Apple.

Le Web s'adapte, voire contre-attaque

- ReactNative <- Facebook (web)
- Responsive UI pour faire des sites web qui passent sur tout support mobile
- De plus en plus d'application web desktop avec une utilisation offline. (Google Docs, etc)
- Evolution ambitieuse du javascript : node.js, typescript, etc.
- Nouveautés Html 5 : animations, webstorage...

Merci

Sondage...

Annexes

Hello World

- version mobile Android

```
public class HelloWorld extends Activity {

   @Override public void onCreate(Bundle savedState) {
      super.onCreate(savedState);
      RelativeLayout layout = new RelativeLayout(this);
      TextView text = new TextView(this);
      text.setText(«Hello World!»);
      relativeLayout.addView(text);
      setContentView(layout)
   }
}
```

Comparaison avec une application desktop

- Application Desktop <-> Application mobile
  - Le mobile est une application classique avec des spécificités en plus
    - stockage sur machine locale, programme local
- Application Desktop <-> Application web
  - le web a commencé à faire des applications très récemment
  - fonctionnalités locales mais stockage online (?)

Mise en prod de correctifs

- Mobile: devoir envoyer une nouvelle version et passer par la soumission, validation, release publique.
- Web: déployer le correctif, fin.

Examples de controverses

- Mobile :
  - Menu principal rétractables à gauche
  - Fenêtres qui se « placent » les unes sur les autres.
- Web :
  - Barre de menu toujours accessible
  - Pages web sur un même template en général.

Interactions supplémentaires sur le mobile

- Interagir autrement avec l'utilisateur :
  - Pinch, swipe, 3D touch, secouer l'appareil, le retourner, faire vibrer, sonner, etc.
- Web: (souris)

Indicateurs clés

- En général :
  - Nombre d'utilisateurs uniques récurrents (par semaine, par mois), nombre d'inscrits.
- Web :
  - trafic
- Mobile :
  - Nombre de téléchargements