

L'accessibilité des sites Web et des applications pour les mobiles

Livre Blanc – Août 2016

Coordonné par **Olivier NOURRY**

Remerciements

Ce livre blanc reprend des contributions faites le 4 décembre 2015 à la Cité des Sciences à Paris lors du 22^{ème} Séminaire du Groupe de Travail AccessiWeb (GTA). Ce séminaire a été réalisé avec le soutien d'Access42, Bla Bla Car, la Cité des Sciences, Smile et V-Technologies.



A propos de BrailleNet

L'association BrailleNet est une association loi 1901 à but non lucratif. Elle œuvre depuis 1997 pour l'accessibilité numérique en faveur de toutes les personnes qui, du fait d'un handicap, ont des difficultés pour utiliser les services et outils numériques.

BrailleNet a développé des ressources et services pour la mise en œuvre effective de l'accessibilité numérique des sites web (création du référentiel AccessiWeb; organisation de formations à l'accessibilité numérique à destination des contributeurs, rédacteurs et développeurs; labellisation de sites web). Elle coordonne et initie des projets de recherche en collaboration avec ses partenaires scientifiques (INRIA, Université Pierre et Marie Curie, l'Institut de la Vision, l'INSERM) et avec des organismes privés.

BrailleNet organise chaque année des séminaires, conférences, ainsi que le Forum Européen de l'Accessibilité Numérique à la Cité des Sciences et de l'Industrie. Elle propose des services en faveur des personnes handicapées, tels que la Bibliothèque Numérique Francophone Accessible qui offre plus de 30 000 ouvrages dans différents formats adaptés.

En 2003 l'association BrailleNet décide de créer le Groupe de Travail AccessiWeb afin de développer l'expertise dans le domaine des technologies de l'accessibilité numérique (techniques, méthodologies...). Ce groupe contribue notamment à augmenter le nombre de professionnels du Web capables de réaliser des services numériques accessibles.

Conformément aux statuts et aux missions de l'association, l'activité du Groupe de Travail AccessiWeb est sans but lucratif. Il est avant tout basé sur le volontariat. Ses travaux ne concernent pas les aspects commerciaux du marché de l'accessibilité numérique.

Coordinateur

Olivier NOURRY (Smile)

Éditeurs

Alex BERNIER & Katie DURAND (BrailleNet)

Table des matières

Préface / Alex BERNIER	3
Introduction / Olivier NOURRY	4
La prise en compte des interfaces mobiles/tactiles par le RGAA 3.0 / Jean-Pierre VILLAIN	7
Concevoir, développer et tester des applications mobiles accessibles pour iOS et Android : présentation des guides d'accompagnement de la DINSIC / Alex BERNIER, Jérôme BOTINEAU	11
L'accessibilité des applications mobiles chez Orange : retours d'expérience de l'équipe EASE / Olivier DUCRUIX	14
Comportement des mobinautes : le point de vue d'un ergonomiste / Romain ROUYER	17
Atelier interactif : tests d'accessibilité sur mobile / Alex Bernier, Jérôme BOTINEAU, Olivier NOURRY	23
Glossaire	30

Préface



Alex BERNIER, Directeur de l'association BrailleNet

Alex BERNIER est ingénieur en informatique de l'Institut National des Sciences Appliquées (INSA) de Rennes. Il a travaillé sur divers projets liés aux livres et aux bibliothèques numériques. Il est notamment responsable de la Bibliothèque Numérique Francophone Accessible (BNFA) et d'un projet de recherche et développement visant à améliorer l'accessibilité des documents scientifiques et techniques pour les personnes déficientes visuelles.

Ce livre blanc rend compte d'une sélection des interventions réalisées lors du séminaire du Groupe de Travail AccessiWeb en décembre 2015. Ce séminaire a la particularité de s'éloigner des technologies du Web, du moins des langages HTML, JavaScript, CSS et ARIA, pour traiter la question de l'accessibilité des applications mobiles "natives", c'est-à-dire développées en utilisant les langages et API propres à chaque environnement mobile (Android et iOS).

Le sujet de l'accessibilité sur mobiles étant émergent, on constatera à la lecture des articles que contrairement à l'accessibilité du Web telle qu'on la connaît aujourd'hui, nous ne pouvons que très peu nous appuyer sur des standards existants.

La plupart des standards qui définissent aujourd'hui l'accessibilité du Web ont été pensés en ayant à l'esprit un utilisateur se servant d'un PC pour consulter des contenus. Cette vision est aujourd'hui dépassée :

- Environ dix ans après l'arrivée massive des smartphones sur le marché, le PC n'est plus l'outil dominant pour accéder à Internet ;
- Un site Web ne permet plus seulement mettre à disposition à un utilisateur du texte, des images et des vidéos, mais permet de proposer des applications complexes ;
- Avec le Web 2.0, l'utilisateur non seulement consulte, mais est devenu producteur de contenus.

Nous vivons un moment où les standards de l'accessibilité ont un pas de retard sur les technologies du Web. Il y a cinq ans, les problèmes brûlants étaient liés aux applications Web : ARIA a été créé pour les résoudre et y est parvenu avec succès dans une large mesure. Aujourd'hui, le pas à réaliser concerne l'accessibilité sur les mobiles, avec des enjeux techniques, ergonomiques (il faut prendre en compte un nouveau mode d'interaction, le "tactile"), et organisationnels, parce qu'il faut réussir à établir des standards dans un environnement technique extrêmement instable et susceptible d'évoluer rapidement.

Grâce à une mise en pratique lors d'un atelier dont nous rendons compte ici, les différences de restitution d'une application identique sur différents environnements ont été évaluées. Nous pouvons alors mesurer le chemin restant à parcourir pour "normaliser" le développement d'applications mobiles accessibles et leur restitution. Ce livre blanc constitue ainsi le premier chapitre d'une histoire qui en comptera sans doute quelques autres sur ce sujet durant les prochaines années...

Introduction



Olivier NOURRY, Consultant et Formateur Accessibilité - Smile

Professionnel du Web depuis 2000, consultant et formateur en accessibilité du Web depuis 2006. Riche d'une expérience en développement, en gestion de projet et en assistance MOE et MOA, Olivier accompagne les organisations dans la mise en œuvre de l'accessibilité au sein de leurs projets de création ou refonte de sites Web. Il forme développeurs, responsables de projet et rédacteurs à l'application des recommandations du RGAA et des bonnes pratiques.

En 2014, le nombre d'accès au Web avec un appareil mobile a dépassé ceux effectués avec un ordinateur de bureau.

Les raisons de cette évolution sont multiples, comme on le détaillera plus loin. Elle change fondamentalement la façon dont nous devons concevoir les sites et applications Web, afin de les rendre utilisables dans toutes les situations de consultation.

Pour les personnes handicapées, de nouvelles possibilités s'ouvrent, mais de nouvelles difficultés également.

L'objet du 22ème Séminaire Technique du GTA, dont est issu le présent Livre Blanc, est d'introduire la thématique de l'accessibilité mobile, d'en expliquer les enjeux et principaux défis, et de présenter des outils et méthodes permettant de créer des contenus plus accessibles avec un appareil mobile.

Les raisons du succès

Comparativement aux ordinateurs traditionnels, les appareils mobiles présentent de multiples avantages :

- Polyvalents : un smartphone moderne peut tout faire : naviguer sur le Web, rédiger, enregistrer du son, des images, des vidéos, consulter des contenus multimédias, jouer, s'orienter, communiquer par texte ou vidéo, et même... téléphoner !
- Plus ergonomiques : l'interface tactile proposée par la majorité des appareils s'avère plus intuitive que le diptyque clavier et souris. De plus la relative petitesse des écrans force les concepteurs à simplifier les interfaces, pour ne garder que l'essentiel.
- Toujours à disposition : de petite taille, légers, ils peuvent se glisser dans une poche ou un sac. Nombre d'entre eux peuvent être tenus d'une seule main, leur design en permettant l'usage avec le pouce de la même main
- Toujours prêts : un appui sur un bouton permet de sortir l'appareil de son mode veille, le rendant immédiatement prêt à l'emploi. Cette réactivité le fait préférer à l'ordinateur traditionnel même pour des usages sédentaires, pour des tâches simples comme consulter son courriel, ou une page Web.
- Toujours connectés : hormis quelques rares « zones blanches », il est possible d'accéder à Internet, via le réseau mobile, pratiquement partout sur le territoire

- Peu coûteux : même si les appareils les plus prestigieux dépassent les 1000€, les smartphones et tablettes sont comparativement moins chers qu'un PC portable. Il est possible d'acquérir un smartphone complet et polyvalent, sans abonnement, pour quelques dizaines d'euros. La concurrence entre opérateurs mobiles permet d'accéder à du matériel haut de gamme dont la charge est étalée dans le temps via des formules d'abonnement. Il est également possible de piocher dans un catalogue très fourni d'applications, nombre d'entre elles étant gratuites, financées par la publicité.

Côté infrastructure, il faut noter aussi que la construction d'un réseau mobile est nettement moins complexe et coûteuse que celle d'un réseau filaire, ce qui intéresse les pays en voie de développement, et les régions isolées.

Pour les personnes handicapées ?

Les personnes en situation de handicap y trouvent les mêmes avantages que le reste de la population, les bénéfices de la portabilité et du coût étant encore plus accentués. Par ailleurs, les systèmes d'exploitation, de conception plus récente que leurs équivalents sur ordinateur, intègrent de nombreuses fonctions d'accessibilité : lecteurs d'écran, commande vocale, adaptation des tailles et couleurs de caractères, gestion des contacteurs, etc.

Ajoutons à cela d'autres capacités telles que la détection et l'interprétation des mouvements, la géolocalisation, et la disponibilité de nombreuses applications dédiées au handicap... Un smartphone ou une tablette peut devenir un outil d'assistance à part entière. Ils sont d'ailleurs considérés comme tels dans certains pays, où leur coût est pris en charge par les systèmes de couverture sociale.

À situation nouvelle, problèmes nouveaux

Cette mutation technologique a été incroyablement rapide : entre la sortie du premier iPhone, initiateur d'une nouvelle lignée d'équipements, et le tournant de 2014, 5 ans se sont écoulés. Et sur cette période, les modèles ont énormément évolué : tailles d'écran, capacités, performances, interfaces... Chaque année, les modèles les plus avancés technologiquement sont surclassés par la génération suivante, et deviennent le tout-venant, avant d'être totalement obsolètes. La fréquence de renouvellement des smartphones est évaluée à 18 mois, alors que celle des ordinateurs est de 3 à 4 ans. Et encore, d'un PC au suivant, les fonctionnalités disponibles évoluent peu ; seul le rapport performances/coût change significativement.

L'écosystème n'est pas en reste : tant pour iOS qu'Android, les deux systèmes dominants, une version majeure est proposée chaque année, tandis que les systèmes d'exploitation plus traditionnels mettent plusieurs années à être renouvelés. Suivant cette logique d'évolution permanente, les applications sont régulièrement mises à jour, parfois en continu, de manière transparente pour l'utilisateur.

Dans ce contexte perpétuellement mouvant, tant au niveau des technologies, des usages, que des besoins, il est difficile de mettre au point des standards et des pratiques de développement stables. Difficulté accrue par les différences marquées entre les systèmes d'exploitation, en particulier au niveau des lecteurs d'écran intégrés par défaut ; au point que, pour les applications natives, les techniques de mise en accessibilité diffèrent notablement d'une plateforme à l'autre, pour chacune de ses versions majeures, et même d'un constructeur à l'autre.

Fait significatif : il n'existe pas, à ce jour, de normes d'accessibilité directement applicables aux équipements mobiles. Plus révélateur encore : les standards existants, relatifs à l'accessibilité des contenus Web, n'ont pas anticipé l'importance prise par les interactions tactiles dans ce nouveau paysage, et n'en font tout simplement pas mention.

C'est dans ce contexte paradoxal – besoins immédiats et importants, mais peu de recul et de connaissances formalisées – que les institutions et les professionnels tentent d'apporter des réponses aux besoins d'accessibilité.

Le 22ème Séminaire du Groupe de Travail AccessiWeb, consacré à l'accessibilité mobile, a permis de présenter les travaux initiés par l'État Français pour outiller les concepteurs et développeurs désireux de rendre leurs productions accessibles sur terminaux mobiles et tactiles. Des retours d'expérience sur la mise en place de méthodes, d'outils, de pratiques de développement, à l'échelle d'une entité ou d'un projet, ont illustré les challenges rencontrés, et les solutions apportées. Une étude basée sur l'observation des usages mobiles a permis de faire le lien entre ergonomie et accessibilité. Enfin, un atelier interactif a permis d'initier les participants aux tests d'accessibilité sur mobile, en particulier au travers d'un lecteur d'écran.

Ce Livre Blanc est la synthèse d'une sélection de ces différentes présentations. Bonne lecture !

La prise en compte des interfaces mobiles/tactiles par le RGAA 3.0



Jean-Pierre Villain - Access42

Jean Pierre Villain est en charge du Pôle R&D de la société Access42, dont il est co-fondateur. Auparavant, il avait travaillé en tant que Senior Accessibility Expert chez TecAccess, aux États-Unis, où il était en charge de rédiger une version opérationnelle des normes internationales pour ses clients et a été fondateur de Qelios. Principal rédacteur du référentiel AccessiWeb coordonné par l'association BrailleNet depuis la version 1.1, Jean-Pierre est également l'artisan du référentiel technique du [RGAA 3.0](#). Il a été formateur des Experts AccessiWeb en Évaluation depuis 2007 pour le compte de l'association BrailleNet. En dehors des référentiels, il voue un amour passionné à JavaScript et CSS.3

Retour vers le futur

Les premiers travaux sur l'accessibilité des plateformes mobiles ont débuté à l'aube de 2005, alors même que WCAG 2.0 peinait à entamer sa dernière ligne droite. Le document qui en résulta (mobile OK en 2007) a beaucoup souffert de devoir s'appliquer sur une technologie de rupture à une époque où personne ne savait vraiment quelle forme prendrait le terminal mobile du futur. La question du protocole (avec feu le [protocole WAP](#)) et celle des modes d'interaction (stylet, clavier virtuel ou physique, interaction tactile...), par exemple, n'étaient pas encore clairement définies, alors que le langage HTML entamait sa mue vers HTML5.

Il faudra attendre 2010 pour que l'accessibilité des contenus web sur mobile sorte de l'ornière poussée par l'adoption massive de la technologie mobile Apple par les utilisateurs, faute de concurrence, et les premiers travaux de WCAG sur ce sujet délicat.

Deux mondes en un

L'accessibilité sur mobile désigne en réalité deux domaines distincts, qui utilisent la même plateforme, mais ont recourt à des technologies différentes. L'un recourt aux langages de développement logiciel tels qu'[Objective C](#) pour les plateformes iOS ou Java pour les plateformes Android. L'autre utilise HTML5 et sa galaxie d'[API](#) destinées à offrir un socle technique efficace, rapide à développer et à maintenir. Cet article n'abordera que les technologies Web, le [RGAA](#) ne couvrant que ce type de contenus.

L'universalité de la diversité

Battant en brèche la pierre philosophale du [W3C](#), le monde du mobile n'a rien d'universel, bien au contraire. Si le nombre de plateformes est mesuré, leurs caractéristiques intrinsèques (par exemple la résolution), en revanche, compliquent énormément le paysage technologique. Si le [Responsive Web Design \(RWD\)](#) permet des adaptations raisonnables, pour l'accessibilité c'est une autre paire de manches. Le monde des interfaces mobiles basées sur le web est instable et très évolutif alors que la prise en charge des « standards du web » commence tout juste à se stabiliser et que le « tactile » ([touch event](#)) n'est pas encore normalisé du côté du W3C. À ces difficultés vient s'ajouter le rythme

important des mises à jour, notamment de la partie navigateur, sur laquelle on constate régulièrement des régressions.

Il existe également une grande différence par rapport aux plateformes de bureau sur lesquelles existent des technologies indépendantes du système d'exploitation. Sur les plateformes mobiles, les technologies d'assistance sont un composant intégré au système d'exploitation lui-même. Sur la plateforme iOS, il s'agit de VoiceOver, sur Android de Talkback. Ces technologies d'assistance sont particulièrement optimisées pour le système hôte et pour les applications logicielles. Cela crée une situation nouvelle où une application web fonctionnelle et accessible sur un ordinateur de bureau peut poser des difficultés lorsqu'on y accède via une interface mobile, y compris si cette dernière provient du même éditeur. C'est le cas, par exemple, de VoiceOver qui va réagir de manière différente sur OS X et sur iOS et pour iOS de manière différente entre un iPhone et un iPad.

Enfin, WCAG lui-même est en retard et ne propose actuellement aucune directive spécifique aux mobiles et certaines spécifications du côté des langages, comme l'[API Touch Events](#), sont encore en cours de normalisation.

L'ensemble de ces caractéristiques mises bout à bout dessine un espace de travail particulièrement hétérogène sur laquelle l'accessibilité s'apparente essentiellement à une chasse aux bugs perpétuelle et sur laquelle la notion de conformité, censée définir un socle stable n'a pour le moment pas beaucoup de sens.

RGAA 3 et l'accessibilité des applications mobiles basées sur le Web

C'est la raison pour laquelle le RGAA 3 n'intègre pas directement les sites et applications mobiles dans sa liste de critères.

En revanche, il propose une ébauche de référentiel mobile basé sur l'état de l'art et les premiers travaux exploratoires de WCAG, notamment.

Cette ébauche de référentiel devrait être intégrée à RGAA à terme et sera réévaluée en vue de son intégration à chaque mise à jour du RGAA.

Ce référentiel se présente donc comme un complément au RGAA, auquel il ajoute la liste des critères spécifiques aux plateformes mobiles pour les sites et applications basés sur le Web.

Les sources

Ce travail est basé sur un ensemble de sources issues de l'état de l'art et des travaux exploratoires de [WCAG 2.0](#), les principales sont :

- Mobile Accessibility : How WCAG 2.0 and Other [W3C/WAI](#) Guidelines Apply to Mobile¹, le travail exploratoire de WCAG ;
- BBC Standards and Guidelines for Mobile Accessibility², les recommandations produites par la BBC et qui font autorité ;
- Mobile Web Best Practices 1.0³, la première recommandation du W3C.

¹ <https://www.w3.org/TR/mobile-accessibility-mapping/>

² <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/mobile>

D'autres sources et références ont été utilisées et sont détaillées dans le document lui-même à des fins documentaires.

Le référentiel mobile/tactile

Le référentiel adopte la même structure que le [RGAA](#) : une liste de critères associés à des tests, un glossaire, des cas particuliers et des notes techniques si nécessaire.

Il est constitué de 9 critères et de 4 recommandations. Les recommandations n'ont pas été implémentées en tant que critères, car elles peuvent poser des problèmes si le site est utilisé à la fois sur un ordinateur de bureau et sur une plateforme mobile. Il s'agit donc de recommandations qui améliorent l'expérience utilisateur, mais sont vraiment spécifiques aux plateformes mobiles.

Liaison avec les documents de références et des niveaux WCAG

Chaque critère est lié à une des références principales citées précédemment.

Lorsque la référence est issue de la note du W3C et que cette dernière établit une correspondance vers un ou des critères WCAG existants, la référence aux critères est indiquée et le niveau est déduit selon la règle du RGAA : le niveau déduit est égal au plus bas niveau des critères liés. Par exemple, si un critère RGAA fait référence à deux critères WCAG, l'un de niveau A et l'autre de niveau AAA, le niveau déduit sera A

Lorsque la note du W3C n'indique pas de liaison avec un critère WCAG ou lorsque la recommandation n'est pas issue de la note du W3C, le niveau est déduit à partir de la note « Comprendre les niveaux de conformité ».

À noter qu'un des critères : « 14.9 [A] Les éléments interactifs adjacents, déclenchant la même action, doivent être groupés en un seul élément, cette règle est-elle respectée ? » a été introduit à titre provisoire, cette problématique devant être intégrée dans la prochaine mise à jour du RGAA.

Le glossaire

Le glossaire contient les définitions adoptées pour définir les concepts propres aux plateformes mobiles/tactiles par exemple les « interactions gestuelles complexes », la notion de « taille suffisante », etc.

Les cas particuliers

Un seul cas particulier a été considéré lorsqu'une interaction gestuelle impliquant un mouvement, comme une rotation, n'a pas d'équivalent dans un autre mode d'interaction non gestuelle.

Conclusion

Le référentiel mobile/tactile est une première étape vers l'intégration dans le RGAA de la prise en compte des sites et applications mobiles, basés sur le Web, diffusés en contexte mobile.

Ce travail a permis de mieux comprendre les besoins en matière d'accessibilité mobile, les interactions en jeu et les solutions à apporter.

³³ <https://www.w3.org/TR/mobile-bp/>

Il sera sans doute remanié ou complété au fur et à mesure des retours du terrain, de l'évolution et de la stabilisation des plateformes mobiles.

Néanmoins, il est utilisable dès maintenant lorsque la base de référence est enrichie de contextes mobiles et peut permettre d'évaluer et d'améliorer notablement l'accessibilité des sites et applications basés sur le Web en contexte mobile.

Il est disponible sur le dépôt du RGAA à l'adresse suivante : <https://github.com/DISIC/referentiel-mobile-tactile>, le public est encouragé à le tester et faire des retours si nécessaire.

Concevoir, développer et tester des applications mobiles accessibles pour iOS et Android : présentation des guides d'accompagnement de la DINSIC



Alex BERNIER, Directeur de l'association BrailleNet

Alex BERNIER est ingénieur en informatique de l'Institut National des Sciences Appliquées (INSA) de Rennes. Il a travaillé sur divers projets liés aux livres et aux bibliothèques numériques. Il est notamment responsable de la Bibliothèque Numérique Francophone Accessible (BNFA) et d'un projet de recherche et développement visant à améliorer l'accessibilité des documents scientifiques et techniques pour les personnes déficientes visuelles.



Jérôme BOTINEAU, Développeur front-end - V-Technologies

Jérôme Botineau est ingénieur d'études et développement diplômé de l'ESEO d'Angers.

Passionné par les technologies web front-end et l'accessibilité numérique, il a participé à de nombreux projets autour du RGAA3.

Il est notamment en charge avec son équipe des études sur l'accessibilité des bibliothèques JavaScript et des applications mobiles.

Introduction

Bien que les applications mobiles n'entrent pas, au sens strict, encore dans le champ direct d'application du RGAA, la Direction interministérielle du numérique et du système d'information et de communication de l'Etat (DINSIC) a souhaité demander la réalisation de travaux pour étudier la question de leur accessibilité. Les résultats de ces travaux sont un ensemble de guides, sont mis à disposition en tant que ressources d'accompagnement, c'est-à-dire que les documents proposés n'ont pas de caractère normatif.

Pour un projet d'application mobile, tout comme pour un projet de site Web, la question de l'accessibilité doit être prise en compte tout au long des travaux, de la conception aux tests, en passant bien sûr par le développement. Les guides suivants ont ainsi été produits :

- Guide de conception d'applications mobiles accessibles (https://github.com/DISIC/guide-mobile_app_conception)
- Deux guides pour le développement :
 1. Dans les environnements "natifs", Android et iOS (https://github.com/DISIC/guide-mobile_app_dev_natif)
 2. Dans des environnements hybrides : avec les *frameworks* OnsenUI et Ionic (https://github.com/DISIC/guide-mobile_app_dev_hybride)
- Guide d'audit d'applications mobiles (https://github.com/DISIC/guide-mobile_app_audit)

Concevoir une application mobile accessible

Ce guide décrit les recommandations qui vont avoir un impact principalement sur les spécifications fonctionnelles et sur le design de l'application.

À titre d'exemple, citons :

- Ne pas véhiculer une information uniquement par la couleur ;
- Utiliser des contrastes suffisants ;
- Donner aux zones sensibles une taille et des marges suffisantes ;
- Prévoir de décrire les éléments d'interface ;
- Donner des moyens d'interactions alternatifs aux gestes complexes : un geste simple ou un autre moyen d'interaction (reconnaissance vocale par exemple).

Développer une application mobile accessible

Pour le développement d'applications mobiles, deux situations sont à distinguer :

1. L'application est développée en utilisant l'environnement "natif" du système cible, c'est-à-dire les API mises à disposition dans les systèmes Android ou iOS;
2. L'application est développée grâce à un framework, c'est-à-dire un ensemble d'outils dont le rôle est de faciliter le développement sur plusieurs plateformes en rendant le code de l'application indépendant grâce à la définition d'une API commune qui pourra être utilisée de la même façon quelle que soit l'environnement ciblé. L'interface de l'application est alors définie en utilisant des langages comme HTML, JavaScript et CSS.

La première solution a l'avantage de permettre de créer des applications "proches" du système, mais a l'inconvénient de nécessiter de maintenir un code différent pour chaque environnement ciblé (Android et iOS par exemple). La seconde permet, au contraire, d'avoir un code commun aux deux environnements, mais l'accès au système est plus difficile, voire parfois impossible, car il est nécessaire de passer par l'intermédiaire du framework.

Pour le développement en "environnement natif", deux systèmes ont été étudiés : Android et iOS. En général, les technologies et API de ces deux systèmes diffèrent fortement : il en va de même en ce qui concerne la manière de prendre en charge l'accessibilité, ce qui a nécessité la rédaction de documents distincts.

Dans chaque document, sont décrites les différentes techniques qui permettent de coder une application accessible, illustrées par des exemples de code. L'accessibilité des éléments d'interface (widgets) courants a été testée : menus, listes, tableaux, formulaires, etc.

Enfin, sont décrites des techniques avancées pour faire en sorte que l'application ait un comportement spécifique qui s'adapte aux technologies d'assistance : détection de la présence d'un lecteur d'écran, récupération de notifications transmises par le lecteur d'écran, etc.

Pour le développement hybride, deux frameworks ont été étudiés : Ionic et OnsenUI. Les deux sont basés sur une "brique" commune nommée Cordova, qui permet de communiquer avec le système. L'interface est programmée en HTML, JavaScript et CSS, en appliquant aux éléments HTML des classes définies par chaque framework. Une évaluation de l'accessibilité des éléments d'interfaces

(widgets) courants a été menée, aboutissant à la conclusion que, pour les deux frameworks, il n'est actuellement pas possible de réaliser une application accessible : des pistes de corrections ont été proposées, qui devront être intégrées dans les frameworks pour en améliorer la prise en charge de l'accessibilité.

Tester l'accessibilité d'une application mobile

Ce dernier guide propose un ensemble de techniques permettant d'auditer une application afin d'en vérifier l'accessibilité. Ces techniques reposent largement sur l'usage de lecteurs d'écran pour vérifier la restitution correcte de l'application avec la synthèse vocale, mais pas uniquement : le guide propose des moyens de vérifier les contrastes, la taille des zones cliquables, ainsi que la possibilité d'utiliser l'application sans le tactile en simulant un clavier (les personnes en situation de handicap moteur utilisent en effet des "contacteurs", dispositifs reliés en Bluetooth qui permettent de piloter le focus grâce à un système d'interrupteur, sans avoir à toucher l'écran).

Conclusion

Les guides réalisés montrent qu'il est possible et relativement simple dans la plupart des cas, de créer des applications mobiles accessibles. Ces guides sont amenés à évoluer avec les environnements : iOS et Android font l'objet de mises à jour fréquentes et il peut y avoir des changements importants dans la manière de prendre en charge l'accessibilité d'une version d'un système à l'autre. Il est ainsi crucial de tester l'accessibilité de ses applications, sur autant de configurations que possible. Développer "en natif" est aujourd'hui le seul moyen de créer une application accessible, mais cette constatation pourra être amenée à évoluer également : il n'y a en effet pas d'impossibilité technique majeure à ce que les frameworks permettant de créer des applications hybrides prennent en charge l'accessibilité, la difficulté étant principalement de sensibiliser les développeurs des frameworks concernés afin qu'ils mettent en œuvre de bonnes pratiques.

L'accessibilité des applications mobiles chez Orange : retours d'expérience de l'équipe EASE



Olivier DUCRUIX, directeur de projet en accessibilité numérique, Groupe Orange

Olivier DUCRUIX est ingénieur informaticien diplômé de l'INSA Lyon (1988). Malvoyant en raison d'une maladie de la rétine (Stargardt) depuis sa plus tendre enfance, Olivier est cadre supérieur au sein d'Orange depuis 1988 ou pendant les 17 premières années de sa carrière, il a successivement exercé les fonctions d'ingénieur analyste, ingénieur réseaux, chef de projet informatique, directeur de département, architecte technique. En 2005 il propose et obtient la création d'une équipe dédiée à l'accessibilité numérique au sein de la DSI du groupe Orange. Directeur du centre de compétences en ergonomie et accessibilité jusqu'en 2015, il exerce depuis un an une mission transverse de directeur de projet en accessibilité numérique.

Introduction

Cet article présente la démarche que le groupe Orange a adoptée pour produire des applications mobiles accessibles. Les aspects organisationnels, méthodologiques et techniques seront abordés et les points clés pour réussir dans cette démarche, dans le contexte compliqué d'une grande entreprise, seront mis en évidence.

Les acteurs de l'accessibilité chez Orange

Un premier facteur de succès réside dans l'implication collective au sein de l'entreprise. Ainsi, depuis plus de 10 ans, plusieurs services s'engagent au quotidien pour une meilleure accessibilité :

- La Mission Insertion Handicap au sein de la DRH
- La Direction Accessibilité du Groupe au sein du Technocentre
- Orange Labs en matière d'innovation technologique
- EASE (Ergonomics and Accessibility Solutions for Everyone) centre de compétences portant l'accessibilité numérique au sein de la DSI du groupe.

Les deux actions de base pour produire des applications mobiles accessibles

D'abord sensibiliser les décideurs et un maximum d'acteurs dans les projets.

Pour ce faire, un [dashboard](#) semestriel sur l'accessibilité des applications mobiles a été élaboré. Pour construire ce tableau de bord, un audit utilisateur de 20 applications mobiles prioritaires est réalisé. Pour chacune entre 3 et 7 parcours clients sont audités.

Les résultats sont présentés tous les 6 mois au Comité de Direction du Technocentre. Les directeurs sont ainsi impliqués, et aident à relayer le message vers les responsables de produits et autres acteurs dans les projets.

En second lieu, sont fourni « sur un plateau aux chefs de projets et concepteurs/développeurs, » les informations nécessaires pour qu'ils s'approprient l'accessibilité. Un maximum d'efforts de synthèse et de simplification de la part de l'EASE facilite la prise en compte par les projets !

En particulier des recommandations techniques pour chaque technologie iOS et Android sont fournies. Elles s'appuient sur une liste de critères d'accessibilité calqués sur les [WCAG 2.0](#)

- Elles sont cohérentes avec celles d'Apple et Google
- L'approche par critères facilite l'intégration de l'accessibilité comme une exigence, l'appropriation des tests par les projets et la restitution des audits.

14 critères ont été définis pour chaque technologie :

- Utilisation des couleurs
- Alternatives textuelles
- Utilisation des composants standards
- Titres et entêtes
- Etat des éléments
- Ordre de lecture
- Etendue des zones de clic
- Eléments fantômes
- Contrôle de contenu
- Scroll horizontal
- Taille de texte
- Formulaires
- Changement de contenu
- Langue (spécifique iOS)
- Navigation au focus (spécifique Android)

Accompagner les projets

Aujourd'hui, pour une bonne prise en compte de l'accessibilité, il est encore nécessaire d'accompagner les acteurs au sein des projets afin qu'ils montent en compétence sur le sujet, tout au long du cycle de conception/développement/qualification du produit.

Ci-après les bonnes pratiques essentielles pour un accompagnement efficace :

- Prendre en compte l'accessibilité dès la phase de conception et de maquettage :
 - Contrastes, taille de textes, [wording](#) (y compris celui des alternatives textuelles)
- Privilégier l'utilisation des composants standards
- Sensibiliser et former les développeurs pour qu'ils comprennent l'intérêt et l'importance de développer accessible.
 - Une heure de sensibilisation comprenant :
 - présentation de l'accessibilité
 - options disponibles sur mobiles
 - mise en situation – découverte de Talkback/VoiceOver

- Une heure de formation (par techno), limitée à présenter les options du [SDK](#) iOS/Android.

Il n'est pas nécessaire d'effectuer des travaux pratiques, la mise en situation et la présentation des options du SDK suffisent pour qu'ils intègrent l'essentiel !

- Ne jamais oublier que se rendre disponible et s'adapter aux contraintes du projet sont également des points essentiels pour que l'accessibilité soit réellement intégrée au mieux au sein du projet.

Le coût de l'accessibilité

Si l'accessibilité est prise en compte dès le départ, le surcoût qu'elle génère sur le projet est très faible, il se limite à celui de la formation plus quelques corrections souvent mineures, et l'expérience montre qu'il se situe entre 1 et 2 jours en moyenne.

A titre d'exemple, même sur l'application Ma Livebox qui comporte un écran très complexe (écran de planification du WIFI), le coût de mise en accessibilité s'est chiffré à 3 jours.

Retours d'expériences – principaux points techniques

Android et iOS proposent tout ce qu'il faut pour rendre leurs applications accessibles. Ainsi, il existe des attributs qui permettent de donner des informations supplémentaires aux outils d'accessibilité.

En pratique, certaines remarques d'accessibilité reviennent systématiquement. C'est le cas du statut des éléments : par exemple, un système d'onglets ne restitue pas toujours le type d'élément, la position et l'état au lecteur d'écran. Une bonne utilisation de l'attribut « alternative textuelle » évite ce problème : « onglet - 1 sur 2 – titre de l'onglet – activé ».

D'autres problèmes sont plus inhérents au mobile : ce qu'on appelle « éléments fantôme » consiste en une superposition de vues. Cela a pour conséquence de bloquer la navigation d'un utilisateur de lecteur d'écran.

En conclusion, maximiser l'utilisation des composants standards et respecter les règles de développement des constructeurs permet de garantir une application sans « gros » problème d'accessibilité, tout du moins facilement corrigé avec les options à disposition.

Comportement des mobinautes : le point de vue d'un ergonomiste



Romain ROUYER, UX Designer – Smile Digital

Ergonome, spécialisé dans les interfaces homme-machine, Romain met quotidiennement en place des méthodologies de conception centrée utilisateur au sein de l'agence Digitale du groupe Smile. Il sensibilise ses collègues et clients au développement de sites web accessibles. Son bagage technique lui permet d'en comprendre les rouages pour faciliter son discours face aux développeurs. Romain est également à l'initiative d'un laboratoire ouvert dédié aux Sciences Cognitives à La Paillasse, un hackerspace.

Particularités d'un utilisateur mobile

En France, 61% de la population possède un mobile et passe en moyenne 1 heure quotidienne sur son smartphone. 40% d'entre eux regardent leur smartphone au réveil.

L'usage d'un mobile est très particulier, car peu souvent dans des situations idéales :

- L'utilisateur est occupé et continuellement distrait par son environnement
- Le contexte d'usage n'a pas de limite : dans le métro, à la maison, en marchant, à la plage, ...
- La variété technologique des smartphones est très hétérogène (taille, qualité, performance, système d'exploitation)
- Un objet technologique ultra personnel et personnalisable, chaque utilisation est unique
- L'aisance et la capacité d'apprentissage varient en fonction des générations, classes sociales et milieux culturels

Difficultés et freins en situation de mobilité

La lisibilité

Notre œil à un angle de vision limité et l'écran d'un mobile est très petit. Le mobinaute doit donc adopter une posture particulière, souvent néfaste pour la colonne vertébrale.

Son usage est universel, en intérieur, dans le noir, mais aussi en plein soleil rendant la lecture des informations et la distinction des couleurs parfois très difficile.

La taille des boutons

Pour des personnes valides, nous sommes sur une utilisation exclusivement tactile. Contrairement à une souris, notre pouce est large et la main masque la zone de contact. La taille des boutons doit être suffisamment importante pour permettre leur utilisation.



La norme [ISO 9241-9](#) indique que la taille des boutons doit correspondre, au minimum, à la largeur maximum de la dernière phalange de l'index de 95% de la population mâle, soit environ 22mm. Par convention la taille minimale est fixée à 44 pixels.

La taille peut également être adaptée en fonction des persona. Par exemple, pour une application de GPS en mode piéton, l'utilisateur est en mouvement, impliquant une attention et des gestes moins précis, et potentiellement une visibilité dégradée.

Le nombre d'erreurs est également important lorsque les boutons sont trop près les uns des autres.

Zones d'accessibilité

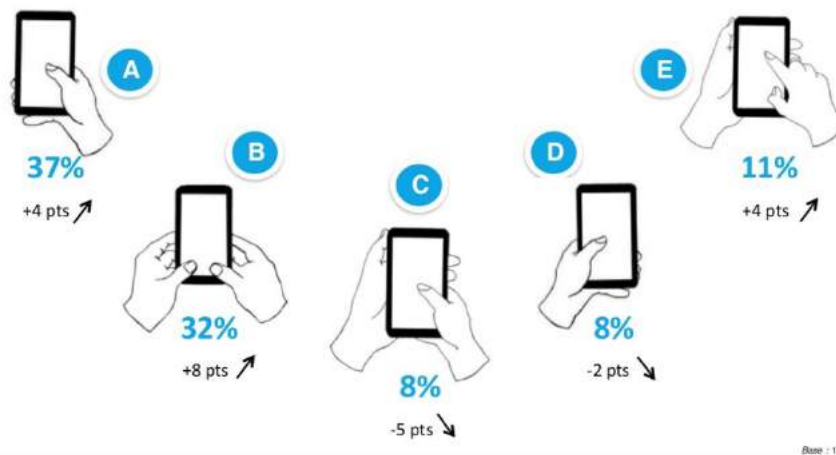
Quelques études permettent d'identifier des zones d'utilisations privilégiées et varient beaucoup en fonction de l'utilisateur et de son contexte.



Parmi ces utilisations, nous distinguons :

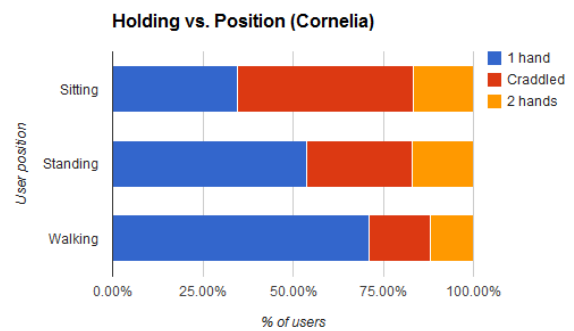
- La posture des mains : une ou deux mains, gaucher ou droitier, avec souvent l'utilisation d'un pouce et un index. La posture évolue beaucoup
- La taille de l'écran : un écran plus large sera plutôt maintenu avec les deux mains
- L'orientation de l'écran : généralement utilisé en mode portrait, une vidéo un jeu ou certaines applications seront par contre orienté en paysage
- La présence d'un étui ou coque de protection peu réduire la zone d'accessibilité

5 principales manières d'utiliser son smartphone



La posture est également un bon référentiel, nous distinguons un mobinaute :

- Assis, un maintien stable, généralement à deux mains (ex : au bureau ou à la maison)
- Debout, une utilisation plutôt à une main, la seconde pouvant servir à sécuriser le maintien du smartphone ou améliorer notre précision (ex : métro)
- Marchant, utilisation et maintien presque exclusivement à une main, la zone d'accessibilité est fortement dégradée (ex : piéton)



Multitâches

L'interruption d'une tâche et la perte d'attention sont très fréquentes en situation de mobilité, de quelques secondes à un abandon total, en fonction de nombreux paramètres.

- Notre environnement et posture.
- La réception d'une notification (ex : alterner SMS et jeux dans le métro)
- Un chargement long. Au bout de 5s d'attente, un site consulté sur mobile perd 74% de ses visiteurs, souvent liée à une connexion lente ou à des performances faibles
- La perte de connexion réseau

Les tâches les plus fréquentes d'un mobinaute :

1. Accéder à ses e-mails
2. Échanger des SMS / téléphoner
3. Aller sur internet
4. Aller sur les réseaux sociaux
5. Recevoir des actualités
6. Jouer
7. Écouter de la musique
8. Lire
9. Regarder des vidéos / films
10. Trouver son chemin
11. Lampe de poche

Quelques bonnes pratiques

L'utilisateur mobile est impatient, son environnement très diversifié et la qualité du réseau aléatoire. Des bonnes pratiques sont à respecter

- Ajuster la taille du texte à la distance de lecture
- Dé-corréler la zone visible et la zone interactive des boutons
- Il n'y a pas qu'une modalité d'interaction (doigt, main, latéralité)
- Ne pas forcer le mode paysage
- Être vigilant au maintien d'un bon contraste
- Proposer des feedback multimodaux (son, vibration, ...)
- Penser une interface personnalisable et simple

Boite à outil d'un ergonomiste

Nous employons quasi exclusivement des méthodes permettant l'étude des utilisateurs, émanant d'une discipline scientifique pluridisciplinaire appelée Sciences Cognitives :

- Audit, benchmark
- Focus group, questionnaire, interview, tri par carte
- *Persona, Customer Journey Map*
- Tests utilisateurs

Pour éviter certains biais et enrichir nos études, d'innombrables outils de *tracking* sont à disposition, les principaux étant :

- Matériel : Occulométrie, caméra, capteurs

- Logiciels : JavaScript, OS Mobiles



Les baromètres d'observations sont également très utiles et permettent de dégager des tendances générales à plus grande échelle. Ce document s'appuie sur 3 études portant sur les usages des mobinautes (observations) :

- 2013 : Steven Hooper a observé 1000 personnes dans les aéroports américains
- Juin 2014 & juillet 2015 : « Smartphone Street Observer », l'agence Personae User Lab a observé 1000 personnes dans les rues de Nantes
- 2015 : Cornelia Laros, travaux de recensement et synthèse de ces études

L'intelligence ambiante (capteurs, *Big Data*, *Deep learning*) et les interfaces multimodales permettent aujourd'hui de nous aider à concevoir des interfaces mobiles de plus en plus efficaces, abordables et personnalisées ayant la capacité de s'adapter en fonction de l'usage et de l'environnement de l'utilisateur.

Quelques ressources

Études

- http://www.insee.fr/fr/themes/document.asp?ref_id=ip1452#inter4
- <http://www.zdnet.fr/actualites/les-francais-toujours-plus-equipes-en-smartphones-et-tablettes-39807091.htm>
- <http://www2.deloitte.com/fr/fr/pages/presse/2014/usages-mobiles-2014.html>
- <http://fr.slideshare.net/dmolsenwvu/developing-a-progressive-mobile-strategy>
- <http://www.criteo.com/resources/mobile-commerce-q1-2015/>

- <http://research.microsoft.com/pubs/75812/parhi-mobileHCI06.pdf>

Usages

- <http://www.agence-csv.com/blog/etude-comportement-mobinautes/>
- <http://fr.slideshare.net/agencedagobert/etat-des-lieux-du-mobile-en-france-mai-2012-13015076>
- <http://www.mediametrie.fr/internet/solutions/l-audience-de-l-internet-mobile.php?id=93>
- <http://www.audiencelemag.com/?article=70>
- <http://insights.mobify.com/understanding-mobile-user-experience-the-3-modes-of-mobile-usage/>
- <http://www.informationweek.com/mobile/google-lays-out-its-mobile-user-experience-strategy/d/d-id/1053921>
- <http://www.smashingmagazine.com/2013/02/25/there-is-no-mobile-internet/>
- <http://alistapart.com/article/organizing-mobile/>

Interactions & accessibilité tactile

- <http://www.uxmatters.com/mt/archives/2013/02/how-do-users-really-hold-mobile-devices.php>
- <http://fr.slideshare.net/shoobe01/40min-how-peopleholdtouchconveys-30874671>
- <http://realites-paralleles.com/tag/mobile-devices/>
- <http://www.lukew.com/ff/entry.asp?1085>
- <http://sebastien-gabriel.com/designers-guide-to-dpi/>
- <http://www.pompage.net/traduction/ecrans-tactiles-mort-aux-mythes>
- <https://msdn.microsoft.com/en-us/library/bb158589.aspx>
- <http://www.uxmatters.com/mt/archives/2013/11/design-for-fingers-and-thumbs-instead-of-touch.php>
- <http://www.uxmatters.com/mt/archives/2013/03/common-misconceptions-about-touch.php>
- <http://germanforblack.com/post/39814923293/phone-in-hand>

Atelier interactif : tests d'accessibilité sur mobile



Alex BERNIER, Directeur de l'association BrailleNet

Alex BERNIER est ingénieur en informatique de l'Institut National des Sciences Appliquées (INSA) de Rennes. Il a travaillé sur divers projets liés aux livres et aux bibliothèques numériques. Il est notamment responsable de la Bibliothèque Numérique Francophone Accessible (BNFA) et d'un projet de recherche et développement visant à améliorer l'accessibilité des documents scientifiques et techniques pour les personnes déficientes visuelles.



Jérôme BOTINEAU, Développeur front-end - V-Technologies

Jérôme Botineau est ingénieur d'études et développement diplômé de l'ESEO d'Angers.

Passionné par les technologies web front-end et l'accessibilité numérique, il a participé à de nombreux projets autour du RGAA3. Il est notamment en charge avec son équipe des études sur l'accessibilité des bibliothèques JavaScript et des applications mobiles.



Olivier NOURRY, Consultant et Formateur Accessibilité - Smile

Professionnel du Web depuis 2000, consultant et formateur en accessibilité du Web depuis 2006. Riche d'une expérience en développement, en gestion de projet et en assistance MOE et MOA, Olivier accompagne les organisations dans la mise en œuvre de l'accessibilité au sein de leurs projets de création ou refonte de sites Web. Il forme développeurs, responsables de projet et rédacteurs à l'application des recommandations du RGAA et des bonnes pratiques.

Préambule

Les standards d'accessibilité et les méthodes de tests associées sont souvent très adaptés aux ordinateurs et à leur environnement logiciel. Mais on manque à la fois d'outils et de techniques de tests orientés vers les terminaux mobiles, en particulier pour les applications natives.

Il existe pourtant un moyen simple de détecter les principaux problèmes d'accessibilité : il suffit de se mettre en situation d'usage. En particulier, utiliser un lecteur d'écran est généralement très révélateur des défauts de conception qui rendent l'application difficile à utiliser de manière non visuelle.

L'un des obstacles à cette approche est la grande diversité des terminaux. Chaque plateforme a ses propres spécificités. Pour ne parler que des deux plus répandues, iOS et Android, on constate que chaque version du système introduit une part de variabilité. Qui plus est, dans le cas d'Android, certains constructeurs proposent une version personnalisée du système d'exploitation, voire du lecteur d'écran intégré. La variété de dimension et de qualité des écrans peut également avoir des effets différenciants sur certains aspects de l'accessibilité.

Objectifs

Au travers de cet atelier nous avons voulu mettre en exergue ces deux constats : d'une part, il est relativement rapide et facile d'obtenir un premier ressenti du niveau d'accessibilité d'une application native, en utilisant les technologies d'assistance. D'autre part, on ne peut pas se contenter de tester sur une ou deux configurations, et sur des simulateurs. Il est nécessaire de tester sur un large éventail d'appareils réels, en panachant les versions de systèmes d'exploitation, les constructeurs et les modèles.

Mise en place

Conditions matérielles

Une semaine avant le séminaire, nous avons demandé à chaque participant d'apporter au moins un smartphone ou une tablette exploitant iOS ou Android, sans imposer de version ni de modèle. Avec environ 40 participants, nous avons ainsi obtenu un échantillon assez varié, avec environ trois quarts de modèles sous Android et un quart sous iOS. Pratiquement toutes les versions d'Android, de la 4.0 à la 6.0, étaient représentées. Les principaux constructeurs Android (Samsung, Sony, LG, Google) étaient représentés, avec différents modèles.

Détail pratique important : il a été demandé aux participants d'apporter des casques audio afin d'éviter la cacophonie lors des tests de lecture vocale.

Enfin, pour permettre de mesurer les tailles de zones activables, nous avons également demandé d'apporter une règle graduée.

Choix des tests réalisés

Les tests ont été sélectionnés parmi ceux décrits dans l'extension « Tactile/Mobile » du RGAA 3.

Nous sommes partis du postulat que certains participants n'auraient jamais utilisé les fonctions d'accessibilité de leurs appareils. Nous avons donc choisi des tests à la fois simples à expliquer, à effectuer, et permettant d'obtenir un premier avis significatif sur l'accessibilité de l'application.

Nous avons fait tester les éléments suivants :

- Taille des zones activables
- Modification de la taille des caractères
- Utilisation des fonctionnalités de base d'un lecteur d'écran

Optionnellement, les participants pouvaient tester les effets du changement d'orientation (passage de l'orientation « paysage » à l'orientation « portrait », et vice-versa).

Préparation aux tests

Avant de commencer les tests proprement dits, nous avons pris le temps de décrire les techniques préconisées pour chacun d'entre eux.

Mesure de la taille des zones activables

Le [critère n° 14.1 de l'extension mobile/tactile du RGAA 3](#)⁴ demande de vérifier que la taille des zones sensibles (activables au toucher) est au minimum de 9 mm de côté. Cette dimension doit être vérifiée sur l'élément tel qu'il est affiché, physiquement, sur l'écran. Elle peut donc potentiellement varier en fonction de la taille et de la résolution de l'écran.

La façon la plus simple de visualiser les zones sensibles est d'activer le lecteur d'écran (voir le paragraphe « Navigation avec lecteur d'écran » ci-après). La taille du rectangle affiché est celle de la zone active en cours de consultation. Il suffit alors de mesurer physiquement l'élément, à l'aide d'une règle graduée.

La véritable difficulté pratique de ce test, c'est qu'il faut disposer d'un large éventail d'appareils, pour couvrir un maximum de situations utilisateurs. Il n'est en effet pas possible, à partir d'une mesure sur un appareil donné, d'extrapoler la taille d'affichage sur un autre appareil. De ce point de vue, le principe de cet atelier est donc d'autant plus intéressant que l'on dispose d'un grand nombre de participants ; chaque participant apportant son appareil, on obtient davantage de données lorsque l'on augmente leur nombre.

Modification de la taille des caractères

L'objectif de ce test est de vérifier que les personnes nécessitant une taille de caractères différente du réglage par défaut sont en mesure d'utiliser les applications.

Pour ce test, on demandait aux participants d'utiliser les applications selon un scénario prédéterminé, avec les réglages de taille de caractères par défaut. Puis, de modifier cette taille (agrandissement ou rétrécissement), et de dérouler le même scénario, en observant les différences dans l'affichage ou le comportement.

Ce réglage s'effectue de la manière suivante :

- Sous iOS : Réglages > Général > Accessibilité > Police plus grande, ou Réglages > Luminosité et affichage > Taille du texte ;
- Sous Android : Paramètres > Accessibilité > Grands caractères (réglage simple), ou Paramètres > Affichage > Taille de la police (réglage entre « petite » et « très grande »)

Navigation avec lecteur d'écran

Le lecteur d'écran s'active et se paramètre via le menu Réglages > Général > Accessibilité > VoiceOver (sous iOS), ou Paramètres > Accessibilité > TalkBack (sous Android).

Une attention particulière a été portée à cette partie de l'atelier. Les lecteurs d'écran introduisent une manière spécifique de naviguer dans une interface tactile. En lecture visuelle, le fait de toucher une zone active est assimilé à un appui sur le bouton de souris, avec les conséquences qui y sont associées. En lecture non visuelle, ceci conduirait à déclencher involontairement des événements, l'utilisateur ne sachant pas où le doigt a été posé. L'activation du lecteur d'écran modifie donc le modèle d'interaction : il permet de donner le focus à chaque élément de l'interface, et d'en restituer

⁴ <https://github.com/DISIC/referentiel-mobile-tactile/blob/master/referentiel-mobile-tactile-liste-criteres.md#141-chaque-zone-sensible-a-t-elle-une-taille-suffisante>

la fonction et le contenu textuel, sans l'activer. Pour déclencher l'action associée (équivalence du clic), l'utilisateur va taper deux fois, rapidement (« double tap »), n'importe où sur l'écran.

Deux méthodes permettent de prendre le focus sur un élément :

- soit l'exploration au toucher : l'utilisateur passe le doigt sur l'écran et découvre son contenu pendant le déplacement,
- soit la navigation par balayage (« swipe ») : l'utilisateur fait passer le curseur successivement sur chaque élément, dans l'ordre défini par l'interface, en balayant l'écran selon un geste prédéfini

La navigation par balayage se fait de manière identique sur iOS et Android : un geste horizontal, vers la droite pour « suivant », vers la gauche pour « précédent ». En revanche, le scrolling (passage d'un écran à un autre) se fait différemment selon la plateforme : balayage avec trois doigts pour iOS, et seulement deux pour Android. Ce geste est nécessaire dans les applications occupant plusieurs écrans, car en navigation par balayage, seuls les éléments présents à l'écran sont atteignables.

Nous avons volontairement limité les gestes utilisés à ces deux-là, de manière à réduire la durée d'apprentissage au strict minimum. Afin de permettre aux participants de se familiariser avec ce mode d'interaction, nous les avons incités à explorer les menus de réglages des paramètres d'accessibilité. Au bout de quelques minutes, la plupart des participants maîtrisaient les deux gestes nécessaires aux tests.

Nous avons également encouragé les participants à naviguer sans regarder l'écran, de manière à se rapprocher de la situation utilisateur réelle.

Choix des applications testées

Pour choisir les applications utilisées pour les tests, nous avons établi les critères suivants :

- Disponible en version gratuite sur iOS et Android
- Destinée au grand public
- Comportant au minimum des contenus, des éléments de navigation, des éléments visuels
- Ne nécessitant pas d'inscription ou d'enregistrement
- Présentant une interface et un contenu identiques pour tous les utilisateurs non enregistrés, pour une plateforme donnée

Compte tenu de la durée de la session (1 heure 30 au total) nous avons fixé le nombre d'applications à 3. Les applications retenues étaient :

- Météo France
- Le Monde
- Télé Loisirs

Il a été demandé aux participants d'installer ces applications sur leur terminal avant le séminaire, pour éviter les éventuelles difficultés de connexion ; et peu de temps avant (idéalement, la veille), pour éviter également d'avoir des versions différentes de l'application parmi les participants.

Déroulement des tests

Afin de coordonner les tests, et collecter les observations de manière constructive, nous avons demandé aux participants d'effectuer tous les tests sur une application avant de passer à la suivante. Les observations étaient exprimées spontanément, et exposées oralement aux autres participants.

Les scénarios de tests ont volontairement été simplifiés, afin de garantir que tous les participants aient la possibilité de tester les 3 applications ciblées. Les participants les plus avancés avaient la possibilité d'explorer d'autres pistes.

Les scénarios :

- Atteindre l'icône de lancement de l'application
- Lancer l'application
- Explorer les contenus de l'écran d'accueil de l'application
- Ouvrir un autre écran, et explorer son contenu

Cette combinaison a permis aux participants de maîtriser progressivement les fonctions de base (circuler dans l'interface, activer un contrôle d'interface), et de laisser un certain degré d'initiative. Le but était de permettre aux participants les plus avancés d'avoir plus d'opportunités de découvrir des problèmes d'accessibilité (approche heuristique).

Résultats obtenus

Sur les applications

Les participants ont pu observer différents problèmes d'accessibilité, sur chacune des 3 applications testées, voire sur les interfaces du système d'exploitation :

1. Zones sensibles trop petites, et/ou trop rapprochées
2. Textes agrandis illisibles (masqués par des éléments d'interface, ou recouverts par d'autres textes). A noter : sur deux des applications pour iOS, les réglages de police système étaient désactivés, remplacés par un paramétrage propre à l'application. Ce qui implique pour l'utilisateur de devoir découvrir ce mécanisme, et nécessite donc de pouvoir lire l'interface avec la taille « normale » de caractères ; ce qui ne conviendra pas pour certains des utilisateurs ayant besoin de ce réglage.
3. Éléments graphiques sans alternative, ou avec une alternative non pertinente (exemple : « 10 » pour le pictogramme signalant un programme télévisé déconseillé aux moins de 10 ans)
4. Informations faisant partie d'un même ensemble, mais restituées séparément (exemple : date, lieu et température sur l'application météo)
5. Informations dépendant de la position à l'écran (exemple : températures positionnées sur les cartes régionales : seule l'information de température est restituée, pas le lieu correspondant à l'emplacement)
6. Piège du focus : le focus reste bloqué sur un écran publicitaire affiché au démarrage, sans information ni contrôle pertinent permettant de régler le problème
7. Différences entre la structure visuelle affichée, et la structure logique de l'interface (sous Android, les applications utilisent une notion de « vues », comparables à des calques, ou couches, dans les logiciels de graphisme). Parfois ces vues sont assemblées de façon non cohérente avec l'apparence visuelle qui en résulte, ce qui modifie l'ordre de circulation du focus

8. Écrans sans titre, ou avec un titre non pertinent
9. Une application ne prend pas en compte les changements d'orientation, et impose le mode portrait.

Sur les objectifs de l'atelier

L'atelier poursuivait deux objectifs :

1. Montrer la nécessité de tests physiques ;
2. Montrer qu'un premier niveau de tests, rapide à mettre en place, permet de détecter bon nombre de problèmes d'accessibilité.

Sur le premier point : la diversité des terminaux était suffisante pour mettre en lumière des écarts entre différentes situations de tests.

Ainsi, certains terminaux sous Android ont des boutons de navigation mécaniques, d'autres sensitifs, et d'autres encore, virtuels (ils sont affichés sur l'écran et activables comme des éléments de l'interface). Ce dernier cas a pour effet de rajouter une « vue » à l'interface, qui dans certains cas peut interférer avec la navigation par balayage. D'autres constructeurs ont personnalisé leur version d'Android, rajoutant par exemple un bouton « égaliseur » (contrôle fin du volume audio), sans les étiqueter correctement. Enfin, le constructeur Samsung a développé une variante de TalkBack, disponible sur sa gamme Galaxy S6. Tous ces éléments concourent à une variabilité des résultats des tests, même sur des scénarios aussi simples que ceux de cet atelier.

Les appareils du constructeur Apple font logiquement preuve d'une plus grande homogénéité. Il est cependant notoire que les différentes versions d'iOS et de VoiceOver présentent des différences de comportement relatives à l'accessibilité ; avec parfois des régressions au fil des versions. Il a aussi été observé des variations entre iPad et iPhone pour une même version d'iOS et de VoiceOver. Par ailleurs, certains appareils récents ont des capacités nouvelles (par exemple : l'interface tactile sensible à la pression exercée, dénommée 3D Touch) qui peuvent introduire des différences dans l'utilisation d'une même version d'iOS sur deux terminaux distincts. Cette variabilité n'a cependant pas été observée sur l'échantillon (une dizaine de testeurs) sur les scénarios joués lors de l'atelier.

Sur le second point : il a suffi de quelques minutes à l'ensemble des participants pour acquérir les gestes de base du lecteur d'écran, et distinguer ce qui relevait d'un problème réel d'accessibilité, d'une difficulté à utiliser l'outil⁵. Ceci est à moduler par le fait qu'il s'agissait d'un public de professionnels du Web ; nous estimons cependant que la cible de ce type d'actions (développeurs, testeurs, responsables produits) est parfaitement en mesure d'obtenir le même résultat.

Conclusion

L'atelier a parfaitement rempli ses objectifs, en démontrant les deux faits suivants :

- Une campagne de tests d'accessibilité d'applications mobiles requiert un large panel de terminaux « physiques », dans différentes versions d'OS, et avec différents modèles ;
- Un grand nombre de résultats exploitables peuvent être obtenus avec une formation très minime à l'utilisation d'un lecteur d'écran sur mobile.

⁵ Il a été ponctuellement nécessaire d'utiliser des gestes plus avancés (comme « aller en fin de liste ») pour consulter les applications. Ceci peut être davantage considéré comme un problème d'ergonomie que d'accessibilité (l'action est possible, mais requiert une maîtrise supérieure de l'outil).

Le dispositif décrit ici peut être reproduit à des fins de sensibilisation, en phase de tests initiaux, ou encore lors de formations à l'accessibilité numérique. La charge d'organisation est minimale, ne nécessitant qu'une information préalable (pour que les participants téléchargent les applications en amont), une mini-formation en début de session, et un accompagnement des participants en cours d'atelier, pour répondre aux questions, noter les observations, et débloquer en cas de difficulté.

Il est important de rappeler, toutefois, qu'il ne peut se substituer à une campagne de tests organisée et systématique, avec des utilisateurs en situation de handicap et des experts en accessibilité numérique. Une telle campagne est en effet préconisée pour détecter tous les problèmes rencontrés par les utilisateurs, en fonction des modalités d'accès, et sur l'ensemble du périmètre de l'application.

Glossaire

API	Un <i>Application Programming Interface</i> (interface de programmation) est un ensemble normalisé de classes, de méthodes ou de fonctions qui sert d'interface par laquelle un logiciel offre des services à d'autres logiciels.
API Touch Events	La spécification <i>API Touch Events</i> définit un ensemble d'événements qui représentent un ou plusieurs points de contact avec une surface tactile, et les changements de ces points par rapport à la surface et des éléments DOM affichés.
Big Data	Des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données ou de gestion de l'information. Les perspectives du traitement des <i>big data</i> comprennent de nouvelles possibilités d'exploration de l'information, de connaissance et d'évaluation, d'analyse tendancielle et prospective et de gestion des risques.
Customer Journey Map Dashboard	La cartographie du parcours client Tableau de bord
Deep learning	Le <i>deep learning</i> (apprentissage profond) est un ensemble de méthodes d'apprentissage automatique tentant de modéliser avec un haut niveau d'abstraction des données grâce à des architectures articulées de différentes transformations non linéaires.
Framework	En programmation informatique, un <i>framework</i> est un ensemble cohérent de composants logiciels structurels, qui sert à créer les fondations ainsi que les grandes lignes de tout ou d'une partie d'un logiciel (architecture).
ISO 9241-9	Exigences ergonomiques pour travail de bureau avec terminaux à écrans de visualisation (TEV) -- Partie 9: Exigences relatives aux dispositifs d'entrée autres que les claviers. Cette norme a été révisité par ISO 9241-400:2007 : Ergonomie de l'interaction homme-système -- Partie 400: Principes et exigences pour les dispositifs d'entrée physiques.
Java	Le langage Java est un langage de programmation orienté objet. Les logiciels écrits dans ce langage doivent être très facilement portables sur plusieurs systèmes d'exploitation avec peu ou pas de modifications.
Objective C	L'Objective-C est un langage de programmation orienté objet réflexif. Aujourd'hui, il est principalement utilisé dans les systèmes d'exploitation d'Apple : Mac OS X et son dérivé iOS.
Persona	Les <i>personas</i> sont des personnes fictives utilisées dans le développement de logiciels informatiques. Il s'agit d'archétypes d'utilisateurs possibles de l'application développée auxquels les concepteurs pourront se référer lors de la conception de l'interface.
Protocole WAP	Le protocole WAP (en anglais : Wireless Application Protocol) est un protocole de communication qui permet d'accéder à Internet à partir d'un appareil de transmission sans fil, comme un téléphone mobile ou un assistant personnel.

Responsive web design	Un site web responsive, ou adaptatif en français, est un site web dont la conception vise, grâce à différents principes et techniques, à offrir une expérience de consultation confortable sur une large gamme d'appareils (moniteurs d'ordinateur, smartphones, tablettes, TV, etc.).
RGAA 3.0	<i>Référentiel Général d'Accessibilité des Administrations</i> , version 3. Référentiel officiel de l'État français, repris d'AccessiWeb HTML5/ARIA, qui répertorie thème par thème l'ensemble de critères de vérification du respect des bonnes pratiques d'accessibilité.
SDK	<i>Software development kit</i> ou kit de développement logiciel
Touch event	Les événements tactiles
Tracking	Le <i>tracking</i> permet d'analyser des données objectives révélant le comportement de l'utilisateur face à des problèmes d'ergonomie
WAI	L'Initiative sur l'Accessibilité du web ou Web Accessibility Initiative (WAI) fut lancée en avril 1997 par le World Wide Web Consortium (W3C). La principale mission de la WAI est de proposer des solutions techniques pour rendre le World Wide Web accessible aux personnes handicapées et d'une manière plus générale à toute personne sans nécessiter de prérequis particulier.
W3C	Le World Wide Web Consortium est un organisme de normalisation à but non lucratif, fondé en octobre 1994 chargé de promouvoir la compatibilité des technologies du World Wide Web (HTML5, HTML, XHTML, XML, CSS, etc.).
WCAG	Les Règles pour l'accessibilité des contenus Web (<i>Web Content Accessibility Guidelines</i>) sont un standard du W3C proposant un large éventail de recommandations pour rendre les contenus Web plus accessibles et s'adressant à tous les distributeurs de contenu sur le Web. Après les WCAG1.0 publiées en 1999, les WCAG2.0 sont le standard officiel depuis le 11 décembre 2008.
Wording	Le choix des termes employés (le champ lexical)