

 A BOOK APART

Les livres de ceux qui font le web

N°.

13

Scott Jehl

DESIGN WEB RESPONSIVE ET RESPONSABLE

PRÉFACE DE Ethan Marcotte

EYROLLES



Le responsive design a considérablement amélioré le design web de ces dernières années, mais compte tenu de l'évolution et de la diversité du Web mondial, il faut désormais aller plus loin et concevoir des sites responsables en veillant à leur utilisabilité, leur accessibilité, leur durabilité et leur performance.

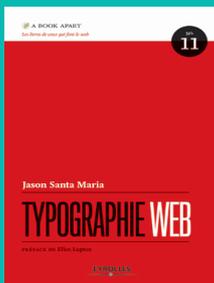
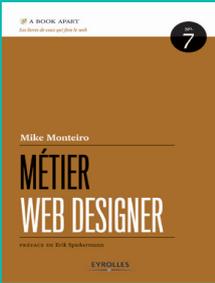
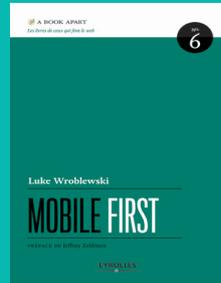
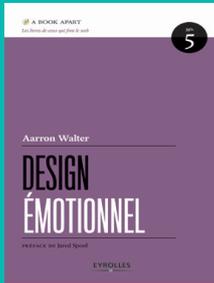
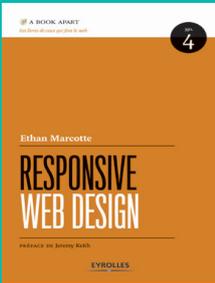
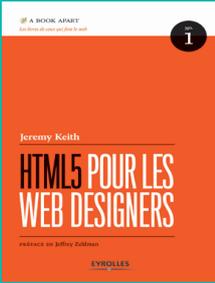
Scott Jehl nous montre ici comment porter un regard critique sur nos créations en développant pour de nouveaux contextes et de nouvelles fonctionnalités, pour des réseaux rapides comme des réseaux plus lents et pour un public véritablement global. Il décrit comment proposer le bon contenu à différentes plateformes et comment optimiser les performances d'un site. Lisez cet ouvrage et concevez des sites et des applications utilisables pour les années à venir !

Au sommaire **Design responsable** * Concevoir pour l'utilisabilité * Concevoir pour le tactile * Concevoir pour l'accessibilité * **Détection durable** * Détection d'appareil : l'évolution d'un palliatif * Nous avons le contrôle * Détecter des fonctionnalités avec JavaScript * Testing responsable dans la suite * **Optimiser les ressources** * Requêtes, requêtes, requêtes ! * Familiarisez-vous avec vos outils de développement * Établir un budget de performance * Moins de requêtes * Préparer les fichiers pour leur transfert * **Transmission responsable** * Transmettre du HTML * Transmettre du CSS * Transmettre des images * Abandonner le pixel * Transmettre des polices de caractères * Transmettre du JavaScript * Faire la synthèse *



A BOOK APART

Les livres de ceux qui font le web



Scott Jehl

**DESIGN WEB
RESPONSIVE
ET RESPONSABLE**

EYROLLES

The logo for Eyrolles, featuring the word "EYROLLES" in a bold, sans-serif font. Below the text is a horizontal line with a small teal circle centered under the letter "O".

ÉDITIONS EYROLLES
61, bld Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

Traduction autorisée de l'ouvrage en langue anglaise intitulé *Responsible responsive design* de Scott Jehl (ISBN : 978-1-937557-1-7-1), publié par A Book Apart LLC

Adapté de l'anglais par Charles Robert

© 2014 Scott Jehl pour l'édition en langue anglaise
© Groupe Eyrolles, 2015, pour la présente édition, ISBN : 978-2-212-14214-3

Dans la même collection

HTML5 pour les web designers - n°1, Jeremy Keith, 2010, 96 pages.
CSS3 pour les web designers - n°2, Dan Cederholm, 2011, 128 pages.
Stratégie de contenu web - n°3, Erin Kissane, 2011, 96 pages.
Responsive web design - n°4, Ethan Marcotte, 2011, 160 pages.
Design émotionnel - n°5, Aaron Walter, 2012, 118 pages.
Mobile first - n°6, Luke Wroblewski, 2012, 144 pages.
Métier web designer - n°7, Mike Monteiro, 2012, 156 pages.
Stratégie de contenu mobile - n°8, Karen McGrane, 2013, 164 pages.
La phase de recherche en web design - n°9, Erika Hall, 2015, 176 pages.
Sass pour les web designers - n°10, Dan Cederholm, 2015, 96 pages.
Typographie web - n°11, Jason Santa Maria, 2015, 160 pages.
Web designer cherche client idéal - n°12, Mike Monteiro, 2015, 152 pages.

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans autorisation de l'éditeur ou du Centre Français d'Exploitation du Droit de Copie, 20, rue des Grands Augustins, 75006 Paris.

TABLE DES MATIÈRES

7		<i>Introduction</i>
22		CHAPITRE 1 Design responsable
55		CHAPITRE 2 Détection durable
99		CHAPITRE 3 Optimiser les ressources
121		CHAPITRE 4 Transmission responsable
191		<i>Conclusion</i>
192		<i>Remerciements</i>
194		<i>Ressources</i>
197		<i>Références</i>
202		<i>Index</i>

AVANT-PROPOS

ON N'A JAMAIS MANQUÉ DE métaphores pour décrire le Web. Au début, c'était notre nouvelle presse à imprimer ; au fil du temps, c'est devenu notre terrain de jeu, puis notre marché global. Aujourd'hui, c'est nos albums photo, nos journaux intimes, nos blogs de voyage, nos instants partagés, nos vidéos, nos GIF, etc., etc. Et avec l'explosion des appareils portables offrant un accès permanent au Web, plus de gens y accèdent aujourd'hui qu'à aucun autre moment de sa courte existence. On peut convenir que le Web est bien plus que la somme de ses parties : d'un tas de câbles échangeant des paquets de données, il est devenu le lieu où nous publions, vendons, communiquons, travaillons et jouons.

Mais voilà le problème : le Web est bien plus fragile que nous ne voulions l'admettre. Il est truffé d'incertitudes - une connexion peut être coupée, la latence d'un réseau peut être trop élevée - qui font que des pans entiers de nos designs risquent de ne jamais parvenir à nos utilisateurs. Bien sûr, il est tentant de voir cela comme un problème temporaire, un problème qui se résorbera progressivement à mesure que les appareils et les réseaux s'amélioreront. Mais entre l'infrastructure vieillissante des économies développées et la popularité croissante des appareils mobiles, plus économiques et moins énergivores sur les marchés émergents, on a l'impression d'assister à l'avènement d'une nouvelle norme pour le Web - un support auquel on accède depuis les quatre coins de la planète, mais qui est également beaucoup plus lent que ce qu'on avait imaginé.

Cela pourrait nous angoisser, mais l'histoire ne s'arrête pas là.

Lorsque les « mobiles » sont entrés dans l'équation, nous avons eu une opportunité : au lieu de créer plusieurs sites séparés pour différents types d'appareils, nous nous sommes rendu compte que nous pouvions utiliser des mises en page flexibles et des *media queries* pour concevoir des sites *responsive* (réactifs), des mises en page qui peuvent potentiellement s'adapter à une infinité d'écrans de tailles différentes.

Et aujourd'hui, nous avons une autre opportunité : celle de nous assurer que nos mises en page ne soient pas simplement adaptatives, mais viables à long terme - qu'elles soient conçues

pour offrir un contenu attrayant et des interfaces riches, non seulement pour les appareils les plus récents et les bandes passantes les plus larges, mais pour tous les écrans du monde.

Par chance, Scott Jehl est là pour nous montrer la voie à suivre.

J'ai eu le plaisir de travailler avec Scott sur plusieurs projets de responsive design, et je n'ai jamais vu de web designer ayant autant de considération et de respect pour la fragilité du Web que lui. Et dans ce petit livre, Scott partagera cette expertise avec vous, cher lecteur, en vous apprenant à concevoir des interfaces agiles et légères qui sauront s'adapter à la volatilité du Web.

Ces dernières années, à force de pratiquer le responsive design, nous avons appris à nous défaire de notre besoin de contrôler la largeur et la hauteur de nos mises en page. Aujourd'hui, Scott Jehl nous montre l'étape suivante : construire des designs responsive de façon responsable, faire en sorte qu'ils soient non seulement adaptés à des écrans de différentes tailles, mais à la forme mouvante d'un Web universel qui ne fait pas de différence.

À vous de jouer.

Ethan Marcotte

INTRODUCTION

DÉBUT 2012, ma femme et moi avons loué un appartement à Siem Reap, au Cambodge. Elle était venue faire du bénévolat dans un hôpital pour enfants ; quant à moi, je continuais mon travail de web designer à distance avec mes collègues de Filament Group aux États-Unis. J'ai ainsi travaillé pendant plusieurs mois au gré de nos déplacements, alors que nous traversions certaines des régions les plus démunies du monde - le Laos, l'Indonésie, le Sri Lanka et le Népal. Chaque étape du parcours m'a donné l'occasion d'utiliser le Web dans les mêmes conditions, souvent limitées, que les gens qui vivaient là. Mes préjugés de designer et ma patience d'utilisateur ont volé en éclats.

Vous avez probablement déjà lu quelque part que les services mobiles étaient le principal mode d'accès à Internet dans certains pays en voie de développement, et mes observations personnelles le confirment. Des vitrines remplies d'appareils mobiles dont j'ignorais l'existence inondaient les marchés (et m'ont permis de remplir mon sac d'appareils à tester). Non seulement tout le monde semblait avoir un téléphone relié à Internet, mais j'ai été surpris de constater que les gens se servaient fréquemment des réseaux cellulaires pour connecter d'autres appareils au Web. Le moyen le plus courant pour connecter un ordinateur portable à Internet était d'acheter une carte SIM prépayée et une clé USB. C'est donc ce que j'ai fait.

Utiliser le Web de cette façon a mis ma patience à rude épreuve. Combien d'heures ai-je perdu à basculer entre plusieurs onglets en cours de chargement et à cliquer sur « Rafraîchir » dans l'espoir d'arriver enfin à ouvrir une application Web, grignotant petit à petit le quota de données limité de ma carte prépayée. En fervent défenseur de bonnes pratiques comme l'amélioration progressive (*progressive enhancement*) et le responsive design, je me plaisais parfois à penser que si ces sites avaient été « bien conçus », ces problèmes n'existeraient pas. Mais la vérité, c'est que beaucoup de ces bonnes pratiques ne fonctionnent pas aussi bien que prévu. Malheureusement, il m'est apparu que la promesse simple d'un accès universel au Web était loin de se concrétiser.

Je ne suis pas le premier à m'en rendre compte. Un article paru dans *Wired* en 2014 décrit l'expérience de cadres de Facebook voulant utiliser leur propre service au cours d'une visite au Nigeria, où plus de 30 % des utilisateurs d'Internet sont inscrits sur Facebook (<http://bkaprt.com/rrd/0-01/>) :

Nous avons lancé l'appli, et puis nous avons attendu... et attendu... Ça a pris un temps fou. Même des fonctionnalités simples, comme télécharger des photos - des choses que font la plupart des utilisateurs de Facebook - ne fonctionnaient tout simplement pas. Ça a été un coup dur pour nous. Nous avons développé une application pour des utilisateurs comme nous. Mais nous étions l'exception, pas la règle.

Nous autres développeurs web sommes des gens exceptionnels. Notre travail demande des réseaux rapides et fiables pour transmettre d'énormes quantités de données, et nous avons accès aux appareils les plus récents et les plus puissants. Mais si la plupart d'entre nous travaillent dans des conditions relativement idéales, nous ne pouvons pas nous contenter de concevoir pour des utilisateurs comme nous ; nous ne pouvons pas oublier que pour la plus grande partie du monde, le Web ne fonctionne pas comme ça.

Vous vous dites peut-être, « mais ce n'est pas le public que je vise », et vous avez peut-être raison. Mais songez au fait que la majorité du trafic web mondial proviendra cette année d'appareils bon marché sur les marchés émergents (<http://bkaprt.com/rrd/0-02/>). Même dans certaines régions les plus développées, les connexions mobiles sont souvent lentes, intermittentes et instables, et les forfaits de données sont de plus en plus chers et limités. Une rapide recherche sur Twitter confirmera que le réseau cellulaire de Londres est toujours notoirement mauvais, et là où j'habite en Floride, j'ai rarement l'occasion d'avoir mieux qu'une connexion EDGE.

Bon nombre de nos voisins, de nos utilisateurs, de nos clients ne disposent pas d'un accès fiable et efficace au Web. En tant que Web designers, nous sommes bien placés pour améliorer cette situation. Si je mentionne les clients, c'est parce que promouvoir un meilleur accès n'est pas seulement une question

d'altruisme, c'est également un moyen d'étendre la portée de nos services et de les rendre plus robustes pour tous ceux qui y accèdent.

Ce livre traite d'accessibilité : élargir l'accès aux services que nous produisons sans compromettre les fonctionnalités qui font avancer le Web. La diversité est une caractéristique essentielle du Web, pas un bug. Nous devons nous efforcer de rendre notre contenu et nos services accessibles à tous les appareils qui en sont capables. Si cela semble difficile, eh bien, c'est parce que ça l'est parfois. Mais j'ai l'intention de vous convaincre que c'est possible, et que ça en vaut grandement la peine.

Commençons par un petit récapitulatif de ce que font nos utilisateurs.

Notre Web se diversifie

Les chiffres en attestent. Apple a vendu plus d'appareils iOS en 2011 que d'ordinateurs au cours des 28 années précédentes (<http://bkaprt.com/rrd/0-03/>). En 2013, l'usage mondial de données mobiles a crû de 81 % (<http://bkaprt.com/rrd/0-04/>). En janvier 2014, 58 % des Américains possédaient un smartphone et 42 % possédaient une tablette, quatre ans après le lancement de l'iPad (<http://bkaprt.com/rrd/0-05/>). La vitesse de cette croissance est époustouflante, mais il ne s'agit pas que des mobiles.

Nos appareils représentent une palette de plus en plus large de facteurs de forme, de fonctionnalités, de contraintes environnementales et d'utilisations (**FIG 0.1**). La variabilité de la taille des écrans à elle seule est stupéfiante - voyez plutôt ce graphi-



FIG 0.1 : Un échantillon des divers écrans que nous devons prendre en charge aujourd'hui

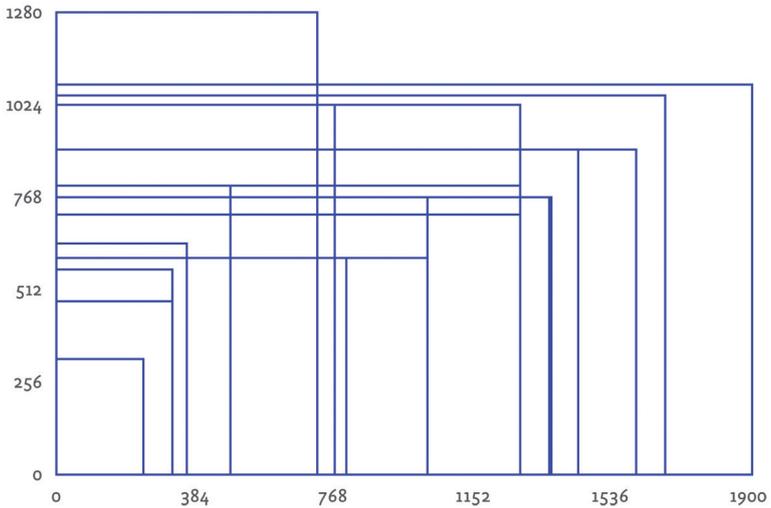


FIG 0.2 : Les différentes tailles d'écran des vingt appareils les plus répandus (<http://bkaprt.com/rrd/o-06/>)

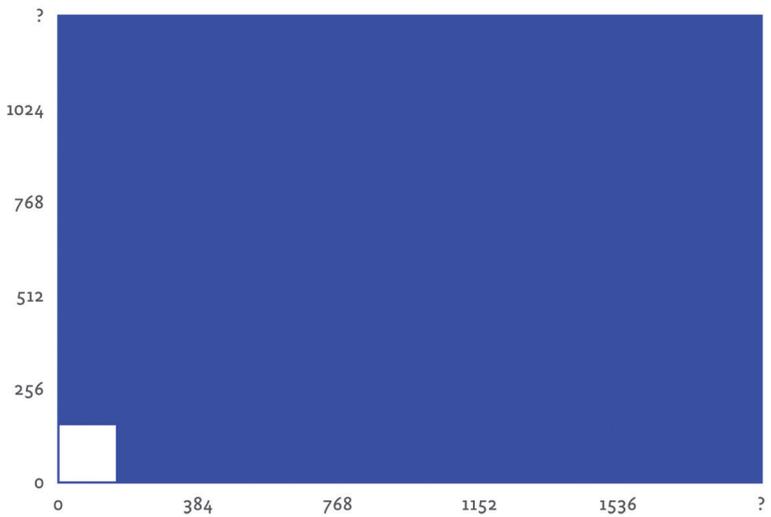


FIG 0.3 : Fragmentation de la taille des viewports sur le Web, redessinée à partir d'un tweet de Cennydd Bowles (<http://bkaprt.com/rrd/o-07/>)

que illustrant les dimensions des vingt écrans les plus répandus au début de l'année 2013 (FIG 0.2).

Les dimensions de l'écran ne sont pas un indicateur de la résolution de l'affichage, qui peut être plus élevée que la définition standard ; elles ne permettent pas non plus de prédire la taille de la fenêtre du navigateur, ou *viewport*, qui diffère souvent de la taille de l'écran. Comme le fait remarquer le designer Cennydd Bowles, étant donnée la variabilité quasi infinie du viewport du navigateur, les dimensions que nous devons prendre en charge sont bien plus nombreuses que ce que le classement des écrans ne laisse entendre (FIG 0.3).

Alors ça, c'est de la fragmentation ! Heureusement, nous avons plus ou moins résolu le problème de la conception d'un design qui s'adapte de manière fluide aux viewports de toutes tailles.

Responsive design : un point de départ responsable

Plutôt que de créer des designs déconnectés, conçus chacun pour un appareil ou un navigateur particulier, nous devrions les traiter comme les facettes d'une même expérience.

Ethan Marcotte, « Responsive web design », *A List Apart*

En 2010, Ethan Marcotte a inventé le terme « responsive web design » pour décrire une approche du web design combinant des grilles fluides, des images fluides et des media queries CSS3 afin de produire des sites web qui réagissent comme par magie à leur environnement (FIG 0.4).

Si le titre de ce livre vous a amené à croire que le responsive web design n'était pas responsable, permettez-moi de clarifier tout de suite. Un design responsive est un design responsable. Fin. Merci de votre attention.

Bon, restons sérieux.

Avec cette combinaison ingénieuse de plusieurs technologies standard du Web, Marcotte nous a offert un moyen durable de développer des mises en page visuelles capables de s'adapter à tous les appareils. Mais il serait le premier à insister sur le fait qu'une mise en page responsive n'est que l'une des nombreuses variables à prendre en compte lorsque nous concevons des sites

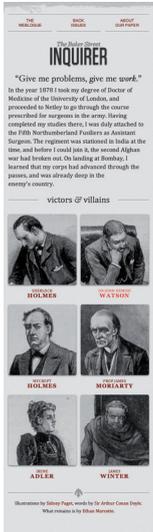


FIG 0.4 : Exemple donné par Ethan Marcotte d'une mise en page responsive dans son article paru sur *A List Apart* (<http://bkaprt.com/rrd/o-08/>)

et des applications multi-appareils. La mise en page n'est que le départ. Nous devons voir plus loin que la fenêtre du navigateur et envisager des moyens de prendre en charge la myriade de fonctionnalités qu'offrent différents appareils, déterminer comment préserver l'accessibilité des interfaces les plus complexes et comment transmettre nos ressources de manière optimale.

Comme l'écrit Trent Walton dans son essai « Device Agnostic » : « De même que les voitures sont conçues pour rouler par une chaleur extrême comme sur des routes verglacées, les sites web doivent être conçus pour faire face à la réalité de la variabilité inhérente au Web » (<http://bkaprt.com/rrd/o-09/>). Heureusement, être responsive du point de vue de la mise en page ne nous empêche pas d'être également responsive du point de vue des performances, de l'interactivité ou de l'accessibilité.

Responsive et responsable

Pour tenir notre promesse d'un Web accessible au plus grand nombre, plaisant à utiliser et durable, nous devons combiner le

responsive design avec quelques principes de design responsable. Un projet de design responsive et responsable doit prendre en compte les facteurs suivants à parts égales :

- **Utilisabilité** : la façon dont l'interface utilisateur d'un site web est présentée à l'utilisateur, et dont elle répond à différentes interactions et conditions de navigation.
- **Accès** : la possibilité pour les utilisateurs de tous les appareils, navigateurs et technologies d'accessibilité d'accéder aux fonctionnalités et au contenu de votre site et de les comprendre.
- **Durabilité** : la capacité pour les technologies d'un site web ou d'une application de fonctionner sur les appareils qui existent aujourd'hui et de continuer à être utilisables et accessibles par tous les utilisateurs, appareils et navigateurs à l'avenir.
- **Performances** : la vitesse à laquelle les fonctionnalités et le contenu d'un site sont transmis à l'utilisateur et l'efficacité avec laquelle ils fonctionnent au sein de l'interface utilisateur.

C'est plutôt complet, non ? Avant que je ne renomme ce livre *Bienvenue sur Internet par Scott*, examinons de plus près les défis que nous rencontrons pour répondre de manière responsable aux besoins de nos utilisateurs.

Concevoir pour une meilleure utilisabilité : capteurs, mécanismes de saisie et interactivité

Il est révolu le temps où les sites web que l'on concevait devaient seulement pouvoir être utilisés à l'aide d'une souris (si ce temps a jamais existé). Aujourd'hui, nous devons prendre en compte les interfaces tactiles, les claviers, les stylets, etc. que nous rencontrons sur un mélange d'appareils mobiles, de tablettes et d'ordinateurs portables. Les appareils les plus populaires du moment sont nombreux à prendre en charge l'interactivité tactile. Par exemple, le système d'exploitation Windows 8 supporte les interactions tactiles sur les ordinateurs portables comme sur les tablettes (FIG 0.5). Microsoft Kinect suit les mouvements des mains et des bras en temps réel (FIG 0.6). En réponse à ces nouveaux mécanismes de saisie, nous ne pouvons pas nous appuyer uniquement sur les traditionnelles interactions de la



FIG 0.5 : Le système d'exploitation Windows 8 fonctionne sur des appareils qui prennent en charge plusieurs modes de saisie, de l'écran tactile à la souris en passant par le clavier. Photographie de Kārlis Dambrāns (<http://bkaprt.com/rrd/0-10/>).



FIG 0.6 : Le Kinect de Microsoft suit les mouvements du corps entier, ce qui laisse peut-être entrevoir de futurs modèles d'interaction pour le Web. Photographie de Scott et Elaine van der Chijs (<http://bkaprt.com/rrd/0-11/>).

souris, comme le survol des liens ; nos interfaces doivent être prêtes à prendre en charge divers mécanismes de saisie au sein de notre univers multi-appareil.

Il existe souvent une disparité entre la puissance des applications natives et les API limitées que nous voyons sur le Web, et c'est un véritable obstacle à la conception d'applications web. Par chance, de plus en plus de navigateurs ont maintenant accès aux fonctionnalités natives du système d'exploitation, telles que le positionnement GPS, les contacts, le calendrier, les notifications, les systèmes de fichiers et la caméra. Ces interfaces standardisées nous permettent de communiquer avec les fonctions locales de l'appareil sans utiliser de plug-ins comme Flash ou Java, qui de toutes façons fonctionnent rarement sur certains appareils. En plus des données des API locales, de plus en plus de navigateurs sont capables d'accéder aux informations des capteurs de l'appareil, tels que le capteur de proximité, le GPS, l'accéléromètre, le niveau de batterie et même l'éclairage ambiant. Avec chaque nouvelle fonctionnalité, la plateforme web fait une nouvelle percée.

Concevoir pour l'accès : technologies d'accessibilité et continuité entre les appareils

Comme les technologies d'accessibilité sont de plus en plus souvent installées par défaut sur nos appareils, nous devons nous assurer que nos sites gardent un sens lorsqu'on les parcourt dans un contexte non visuel. Tous les ordinateurs Apple et les appareils iOS incluent désormais par défaut le logiciel de lecture d'écran VoiceOver, qui est intégré au navigateur et offre un système de navigation tactile tout en lisant la page à voix haute. Son système tactile rotatif « multitouch » permet par exemple de naviguer le Web en parcourant les titres et les liens, ce qui nous donne d'autant plus de raisons d'être vigilants quant aux balises que nous utilisons pour communiquer notre contenu (**FIG 0.7**).

Les technologies d'accessibilité ne sont pas uniquement destinées aux personnes handicapées ; la voix et l'audio peuvent être le mode d'interaction préféré et le plus sûr pour tous les utilisateurs dans certains contextes. Le lecteur d'écran le plus utilisé est probablement Siri d'Apple, qui est pratique pour les



FIG 0.7 : Rotor de VoiceOver sur l'iPhone (<http://bkaprt.com/rrd/0-12/>)

personnes qui ne peuvent temporairement pas regarder leur écran (au volant, par exemple), ou qui préfèrent l'aisance de l'interaction vocale à la saisie manuelle. Alors que les applications web continuent à faire des percées dans nos systèmes d'exploitation natifs, nous pouvons nous attendre à ce que les logiciels de ce genre soient de plus en plus répandus.

En plus d'offrir une expérience utilisable dans les contextes isolés, nous devons garder à l'esprit que les gens passent de plus en plus fréquemment d'un appareil à l'autre et s'attendent à pouvoir accéder à votre contenu en toutes circonstances. Une étude publiée par Google en 2012 intitulée *The Multi-Screen World* a révélé que beaucoup de gens utilisaient plusieurs appareils au cours d'une même journée, et bien souvent pour accomplir une même tâche (FIG 0.8). L'étude a notamment révélé que 65 % des acheteurs qui ajoutaient un article à leur panier d'achats sur un appareil mobile concluaient la transaction plus tard sur un ordinateur portable. Il est possible que la personne soit interrompue

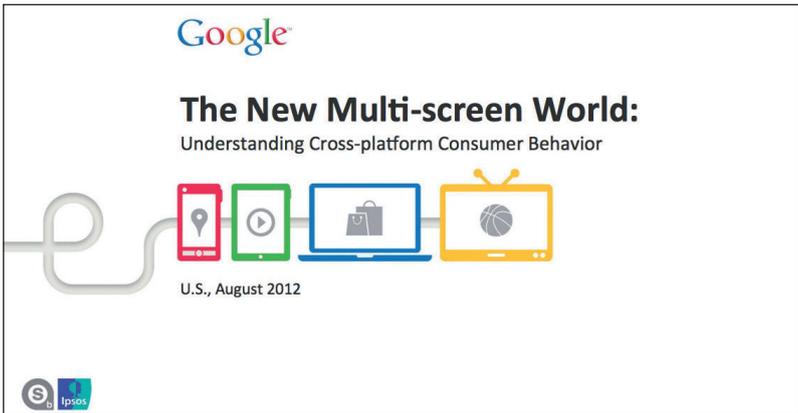


FIG 0.8 : L'étude publiée par Google en 2012 intitulée The New Multi-Screen World (<http://bkaprt.com/rrd/0-13/>)

par un coup de fil ou préfère terminer la procédure de paiement sur un appareil pourvu d'un clavier. Quelle qu'en soit la raison, nous devons accueillir nos utilisateurs où qu'ils se trouvent.

Navigateurs : la tendance est au vintage

Alors que les navigateurs modernes tels que Google Chrome, Firefox et même Internet Explorer ne cessent de sortir de nouvelles fonctionnalités, beaucoup d'appareils dans la nature et dans les magasins utilisent un navigateur qui n'est plus en cours de développement. Par exemple, la version 2 du système d'exploitation Android est toujours très répandue dans le monde alors qu'il en est aujourd'hui à la version 5, et elle inclut un navigateur intégré qui n'a pas été mis à jour depuis 2011 (<http://bkaprt.com/rrd/0-14/>) ! Les développeurs chevronnés (c'est-à-dire vieux) comme moi se souviendront peut-être d'une situation similaire avec l'interminable fin de règne d'Internet Explorer 6. Malheureusement, à terme, le support des navigateurs finit toujours par être délaissé au profit d'une nouvelle plateforme ou des nouvelles priorités de l'entreprise, abandonnant ainsi tous les utilisateurs existants.

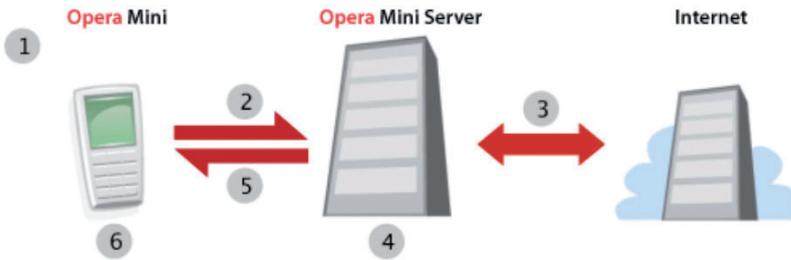


FIG 0.9 : Un graphique d'Opera illustrant comment le navigateur Opera Mini accède au Web (<http://bkaprt.com/rrd/o-15/>)

Ces contraintes des navigateurs constituent un défi, mais sont également une chance. Les utilisateurs recherchent souvent d'autres navigateurs pour leurs plateformes, dont certains offrent des fonctionnalités et des arguments de vente inhabituels. Par exemple, des millions de gens qui préfèrent que les pages web se chargent rapidement et consomment moins de données sur leur forfait (dingue, non ?) choisissent un navigateur comme Opera Mini, qui télécharge tout le contenu Web par l'intermédiaire de serveurs proxy qui optimisent la taille de téléchargement de chaque page (**FIG 0.9**). Les navigateurs qui fonctionnent ainsi prennent peu de JavaScript en charge, voire pas du tout ; ironiquement, des pratiques bénéfiques pour des navigateurs plus anciens, telles qu'un code HTML fonctionnel, sont également d'une aide précieuse pour les millions de personnes qui utilisent ces nouveaux navigateurs par proxy !

Optimiser les performances : réseaux, poids et incidence sur les performances

Les contraintes posées par les réseaux mobiles sont de plus en plus nuancées et difficiles à cerner, bien qu'elles tendent globalement à s'améliorer. Pour le moment, les connexions aux réseaux mobiles sont intermittentes et lentes, même dans les pays développés. Pour garantir de bonnes performances, nous devons revoir notre façon de transmettre notre contenu, réduire le poids et le volume de nos ressources et résoudre les dysfonctionnements qui empêchent d'y accéder.

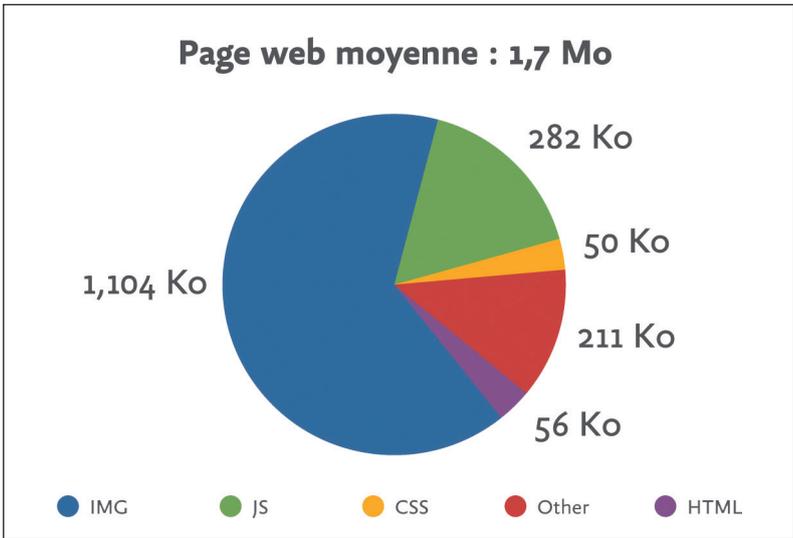


FIG 0.10 : Poids moyen des sites Web, avril 2014 (<http://bkaprt.com/rrd/o-17/>)

Toute portion de code inutilisée que nous transmettons est une perte de temps et d'argent pour nos utilisateurs, et nous avons une grande marge de manœuvre pour nous améliorer. Dans un billet publié en avril 2013 et intitulé « What are Responsible Websites Made Of ? », Guy Podjarny a analysé la taille des fichiers transmis par 500 sites web responsive et a constaté que 86 % d'entre eux transféraient des ressources similaires à tous les appareils (<http://bkaprt.com/rrd/o-16/>). Plutôt que de servir des images optimisées, par exemple, les sites fournissaient des images pour grand écran (comptant ainsi sur le navigateur pour les redimensionner à la taille de l'écran) ainsi que pléthore de CSS, de JavaScript et d'autres fonctionnalités qui n'étaient nécessaires que dans certains contextes.

Bien sûr, le problème du poids du Web n'est pas l'exclusivité du responsive design. Il prolifère depuis longtemps sur le Web de bureau ; en avril 2014, un site web pesait en moyenne 1,7 Mo (**FIG 0.10**).

Des sites web lourds et mal optimisés entraînent des temps de chargement plus longs pour les utilisateurs. Une étude réalisée

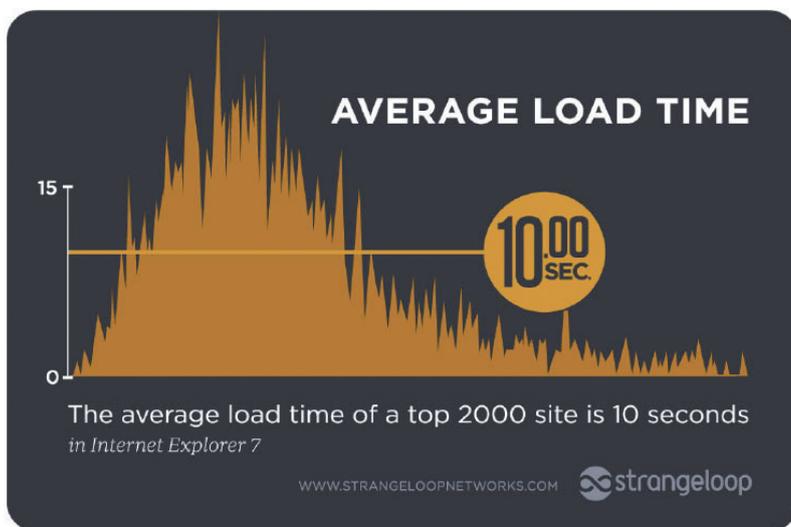


FIG 0.11 : La taille de transfert d'un site affecte sa vitesse de chargement (<http://bkaprt.com/rrd/o-19/>)

en 2012 par StrangeLoop sur les sites du Top 2 000 d'Alexa a établi que le temps de chargement moyen d'un site était de six à dix secondes sous Internet Explorer 7 avec une connexion wi-fi, alors ne parlons même pas de ce que ce serait sur une connexion mobile (**FIG 0.11**) ! De mauvaises performances ont une incidence directe sur les utilisateurs, et par conséquent sur l'activité commerciale. En 2012, Walmart a constaté que pour chaque seconde de temps de chargement en moins, le taux de conversion augmentait de 2 % ; ses recettes augmentaient d'un pour cent pour chaque diminution de 100 millisecondes (<http://bkaprt.com/rrd/o-18/>).

Par ailleurs, alors que la vitesse et la fiabilité des réseaux cellulaires limitent toujours l'usage de données mobiles, le coût des données elles-mêmes est devenu prohibitif. En 2014, si vous achetiez un iPhone dans un Apple Store aux États-Unis, le forfait le moins cher de Verizon à 60 \$ par mois comprenait seulement 250 Mo de données. Avec une taille de page moyenne à

1,7 Mo, vous comprenez mieux pourquoi votre quota mensuel s'épuise aussi vite !

Comme un usage responsable du réseau est vital pour les performances, ce livre consacre plusieurs chapitres à la minimisation des données transférées, mais également à l'optimisation de la transmission de notre code afin que nos sites soient utilisables aussi rapidement que possible.

Accepter l'imprévisible

Le Web a toujours été un support de design hostile. Alors que l'usage d'appareils multiples est en hausse, les scénarios tels qu'une faible bande passante, un écran de petite taille, une orientation imprévisible de l'écran ou une navigation non visuelle ne sont plus des exceptions ; ce sont des contextes quotidiens. Il n'est plus possible de concevoir selon un ensemble de conditions inflexibles, car les interfaces que nous concevons et développons sont de plus en plus utilisées dans des circonstances que nous ne pouvons ni prévoir ni contrôler.

Pour répondre aux besoins de tous nos utilisateurs sur le Web aujourd'hui, nous devons rendre nos designs responsive dans le moindre détail et préparer notre code pour parer à toute éventualité. Et pour cela, nous devons prendre en compte à la fois les modes d'utilisation actuels et potentiels. Nous devons créer des sites responsive qui mettent l'accent sur les performances, l'accessibilité, l'utilisabilité et la durabilité, mais il n'est pas simple de parvenir à ces objectifs. Dans ce livre, nous explorerons les défis auxquels nous devons faire face pour implémenter un design responsive. En suivant des pratiques et des modèles responsables, nous pouvons éliminer bon nombre de problèmes d'accessibilité et de performance avant qu'ils ne se présentent, pour offrir une expérience adaptée et optimisée quelles que soient les fonctionnalités et les contraintes du navigateur.

Notre objectif est de créer des expériences plaisantes et inclusives (rien que ça), alors poursuivons et familiarisons-nous avec quelques manières d'aborder les défis que nous devons relever.

DESIGN RESPONSABLE

Mon amour du responsive design repose sur l'idée que mon site web vous accueillera où que vous soyez - sur un portable, un ordinateur de bureau ou tout autre appareil.

Trent Walton, « Fit To Scale » (<http://bkaprt.com/rrd/1-01/>)

LE RESPONSIVE DESIGN, avec ses principes fondamentaux (grilles fluides, images fluides et media queries), constitue une avancée considérable vers un Web plus universel qui transcende les différences entre appareils. Cependant, il repose sur des fonctionnalités qui sont susceptibles de ne pas marcher comme prévu - voire pas du tout. Nos sites doivent réagir à toutes sortes d'imprévus, qu'il s'agisse du comportement de nos utilisateurs, des conditions du réseau ou de scénarios de compatibilité uniques.

Dans ce chapitre, nous allons explorer deux principes responsables : l'utilisabilité et l'accessibilité. Nous aborderons des considérations d'ordre général avant d'étudier des exemples de code concrets et conçus pour durer que vous pourrez implémenter dès maintenant. Pour commencer, parlons de design.

CONCEVOIR POUR L'UTILISABILITÉ

Lorsque nous pensons à l'utilisabilité d'un design responsive, nous cherchons un moyen de présenter le contenu et les fonctionnalités du design sur différents appareils et tailles d'écran. Les composants de l'interface doivent-ils faire place au contenu lorsque la surface de l'écran est réduite ? Ces composants fonctionnent-ils de manière intuitive en réponse aux différents modes de saisie ? Le contenu et la hiérarchie sont-ils clairement balisés ? La longueur des lignes permet-elle une lecture agréable sur les écrans de toutes tailles ?

Passez rapidement au navigateur

Ne parlons plus de « concevoir dans le navigateur », mais de « décider dans le navigateur ».

Dan Mall, The Pastry Box Project (<http://bkaprt.com/rrd/1-02/>)

Chez Filament Group, nous démarrons la plupart de nos projets sous Adobe Illustrator, avec lequel nous itérons des concepts de design visuel généraux. Nous essayons ensuite de passer au code aussi vite que possible. À ce stade, nous visons à concevoir le strict nombre de variations nécessaire pour communiquer un plan pour la mise en page et l'interactivité sur différents view-ports - de simples suggestions pour différents types d'appareils. Nous ne cherchons pas encore à tirer parti des différents mécanismes de saisie et des fonctionnalités de chaque navigateur, ni à établir quel appareil recevra quelle variante de la mise en page. L'objectif est de passer au navigateur aussi vite que possible pour prendre des décisions de design et d'interaction en contexte, ce qui se traduira par des recommandations plus informées pour nos clients.

Trouvez vos points de rupture

Les points de rupture sont les dimensions auxquelles nous passons d'une mise en page fluide à une autre à l'aide de media queries. Voici deux exemples :

```

/* premier point de rupture */
@media (min-width: 520px){
  ...styles pour les viewports de 520 px de largeur et
  plus
}
/* deuxième point de rupture */
@media (min-width: 735px){
  ...styles pour les viewports de 735 px de largeur et
  plus
}

```

S'il est tentant de choisir des points de rupture tôt dans le processus de design, basés par exemple sur les dimensions d'appareils courants que nous voulons prendre en charge, la vérité, c'est qu'il ne faut pas les choisir. Nous devons plutôt les trouver en nous laissant guider par notre contenu.

Commencez d'abord par une fenêtre de petite taille, puis élargissez-la jusqu'à ce que ça ait l'air moche. C'est le moment d'ajouter un point de rupture !

Stephen Hay, <http://bkaprt.com/rrd/1-03/>

Le design et le contenu de la page doivent modeler et influencer les points de rupture. Comme Hay le fait remarquer, le moyen de plus simple de trouver des points de rupture est simplement de redimensionner la fenêtre du navigateur jusqu'à ce que le contenu devienne bizarre (c'est le terme technique) à utiliser ou à lire – et *presto*, un point de rupture !

En plus de votre instinct, vous pouvez vous fier à des recommandations un peu plus objectives. D'après le livre de Richard Rutter écrit en hommage à Robert Bringhurst, *The Elements of Typographic Style Applied to the Web* (<http://bkaprt.com/rrd/1-05/>), la mesure (nombre de caractères par ligne dans une colonne de texte) optimale pour une lecture immersive est généralement donnée entre 45 et 75 caractères, espaces compris (**FIG 1.1**). Lorsque vous élargissez votre fenêtre, faites attention au moment où une colonne de texte approche cette plage : c'est probablement là que vous devrez ajuster votre mise en page.

2.1.2 Choose a comfortable measure

“Anything from 45 to 75 characters is widely regarded as a satisfactory length of line for a single-column page set in a serifed text face in a text size. The 66-character line (counting both letters and spaces) is widely regarded as ideal. For multiple column work, a better average is 40 to 50 characters.”

FIG 1.1 : Dans cet exemple, la mesure est de soixante-dix caractères par ligne, ce qui permet une lecture confortable (<http://bkaprt.com/rrd/1-04/>).

Lorsque vous travaillerez sur des projets de responsive design complexes, vous vous apercevrez que ces points de rupture se situent souvent à différentes tailles pour différentes portions d'une mise en page, et que certains sont plus importants que d'autres.

Les points de rupture majeurs marquent des changements importants, consistant généralement à ajouter des colonnes ou à modifier radicalement la présentation de plusieurs éléments ; les points de rupture mineurs impliquent généralement de légers ajustements du design (comme de changer l'attribut `font-size` d'un élément pour éviter le retour à la ligne) qui exploitent les espaces restants entre les points de rupture majeurs. Généralement, les points de rupture majeurs sont décidés tôt dans le développement, tandis que les points de rupture mineurs sont ajoutés à titre de touche finale. Moins nous utiliserons de points de rupture et plus le design sera simple à entretenir.

Intéressons-nous à un exemple. Sur le site web du *Boston Globe*, la mise en page comporte deux ou trois points de rupture majeurs, mais les composants plus complexes varient plus fréquemment. L'encadré d'en-tête du site comporte quatre points de rupture majeurs, plus quelques ajustements mineurs pour éviter les retours à la ligne (**FIG 1.2**).



Majeur

Premier point de rupture : les options de navigation et de recherche se déroulent d'un clic.



Majeur

Deuxième point de rupture : le logo passe à gauche pour partager la largeur avec les icônes de navigation.



Majeur

Troisième point de rupture : le logo repasse au centre, la boîte de recherche est visible à tout moment.



Majeur

Quatrième point de rupture : la boîte de recherche passe à droite du logo, la barre de navigation s'élargit.



Mineur

Point de rupture final : la boîte de recherche s'élargit, la liste des petites annonces est visible à tout moment dans le coin supérieur gauche.

FIG 1.2 : Points de rupture majeurs et mineurs de l'en-tête du Boston Globe

Concevez de façon modulaire

Comme dans l'exemple d'en-tête qui précède, je trouve utile de compiler les différentes variations de chaque composant séparément ; ainsi, je peux tester son utilisabilité et documenter ses variations en un seul endroit. Le développeur Dave Rupert, de Paravel, explore ce concept dans un billet intitulé « Responsive Deliverables » (<http://bkaprt.com/rrd/1-06/>). Rupert écrit : « Des livrables responsive doivent être des systèmes pleinement fonctionnels du style Twitter Bootstrap (<http://bkaprt.com/rrd/1-07/>),

taillés sur mesure pour les besoins de vos clients. » En d'autres termes, nous devons construire et documenter nos composants du plus petit au plus grand, comme les pièces d'un puzzle qui s'emboîtent parfaitement.

Même contenu, moins de bruit

Vous êtes parvenu à trouver des points de rupture horizontaux pour des viewports de différentes tailles. Comment allez-vous faire tenir tout ce contenu sur les plus petits écrans sans que tout soit confus ? Le responsive design essuie parfois des critiques à cause de sites qui essaient d'éviter les situations confuses en dissimulant une partie de leur contenu aux utilisateurs - les empêchant d'accéder à un contenu qui était suffisamment important pour qu'on l'inclue à la base. Si c'est un contenu utile, il est utile pour tout le monde. Comme l'indique Luke Wroblewski dans *Mobile first*, plutôt que de dissimuler le contenu qui n'est pas pratique à afficher, mieux vaut réorganiser le design de sorte à préserver l'utilisabilité sur les plus petits écrans.

Par chance, nous disposons de nombreux modèles de design qui permettent de s'adapter aux contraintes des petits écrans de façon intéressante, intuitive et responsable.

Dévoilement progressif

L'un de ces modèles est le dévoilement progressif (*progressive disclosure*), qui consiste à afficher le contenu à la demande. En clair, il est acceptable de masquer le contenu du moment que vous laissez à vos utilisateurs un moyen d'y accéder. L'idée du dévoilement progressif est simple : masquez des portions du contenu, mais offrez une interface intuitive pour que vos utilisateurs puissent l'afficher s'ils le souhaitent (FIG 1.3).

Le dévoilement progressif est le plus souvent un simple jeu d'afficher-masquer comme dans l'exemple précédent, mais nous avons bien d'autres moyens pour basculer visuellement d'un contenu à un autre. Par exemple, cette annonce immobilière effectue une rotation 3D pour révéler des informations supplémentaires sur la propriété, comme son adresse et son emplacement sur une carte (FIG 1.4). Pour les navigateurs qui ne prennent pas en charge les animations 3D en CSS, les utilisateurs peuvent ouvrir la carte sans transition animée, tandis

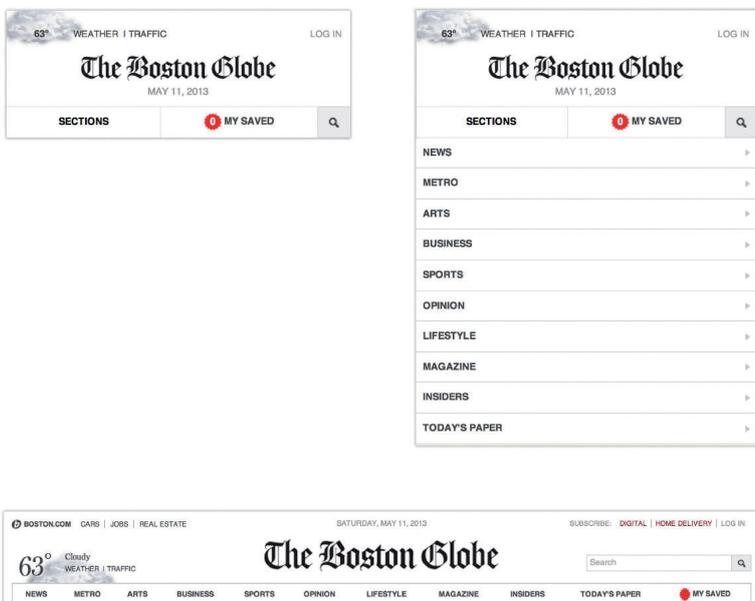


FIG 1.3 : L'interface de navigation du Boston Globe utilise un système de dévoilement progressif sur les petits écrans.

que les navigateurs les plus basiques affichent la carte en toutes circonstances, sous les informations du bien.

La « mise en page hors canvas », terme inventé par Luke Wroblewski dans son article « Off-Canvas Multi-Device Layouts », décrit une autre approche notable pour réduire la complexité sur les petits écrans (<http://bkaprt.com/rrd/1-08/>). Wroblewski décrit plusieurs modèles de design consistant à placer les composants non prioritaires de l'interface en dehors de l'écran jusqu'à ce que l'utilisateur les appelle en touchant une icône ou autre élément similaire ; le contenu qui se trouvait en dehors de l'écran vient alors se superposer ou déplacer le contenu principal (FIG 1.5). Cette approche « à la demande » est de plus en plus répandue sur les petits écrans.

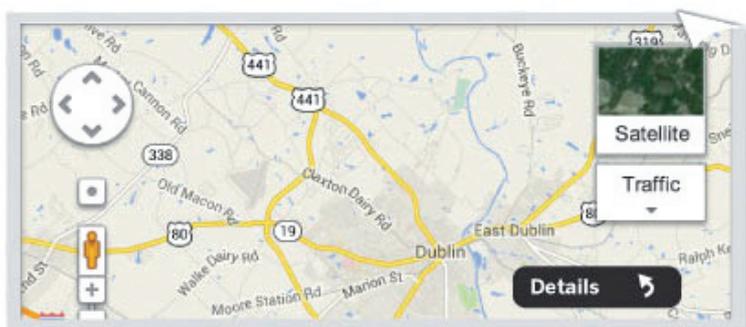
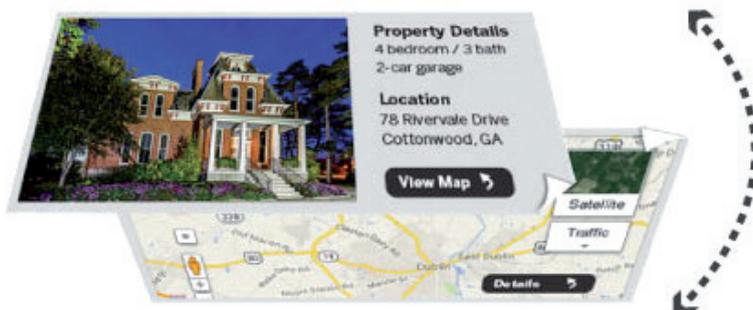
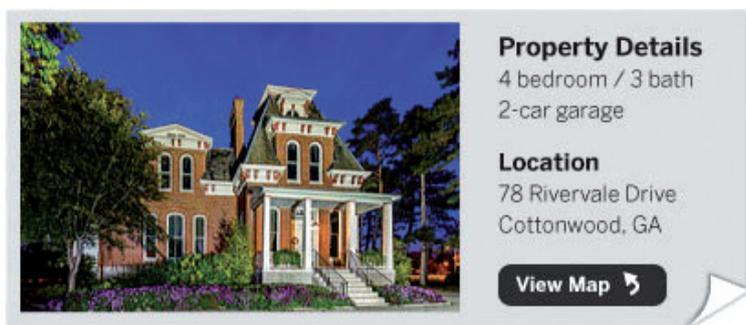


FIG 1.4 : Dévoilement progressif du contenu à l'aide d'une rotation 3D

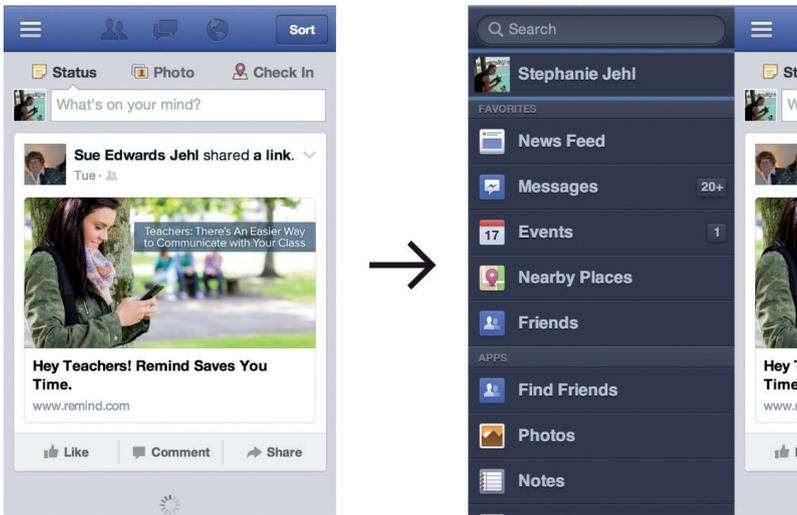


FIG 1.5 : Lorsque vous touchez l'icône du menu, la barre de navigation hors canvas de Facebook apparaît sur la gauche de l'écran.

Tableaux responsive

Les tableaux sont l'un des types de contenu les plus difficiles à présenter sur petit écran. Il est souvent essentiel que l'utilisateur voie les légendes des lignes et des colonnes associées à une cellule du tableau, mais le nombre de lignes et de colonnes que nous pouvons afficher est limité (**FIG 1.6**).

Chez Filament, nous avons beaucoup expérimenté et nous avons découvert deux modèles qui marchaient suffisamment bien pour être inclus dans le framework jQuery Mobile. Le premier modèle, Reflow (<http://bkaprt.com/rrd/1-09/>), reformate le tableau d'une vue multicolonne à une vue en liste ; chaque cellule devient sa propre ligne, avec la légende de la ligne à sa gauche (**FIG 1.7**).

Pour cela, Reflow utilise CSS pour donner à chaque cellule du tableau l'attribut `display: block`, créant ainsi une nouvelle ligne, et JavaScript pour récupérer les légendes de chaque colonne du tableau et les insérer dans les cellules qui serviront de légende (tout en masquant les légendes supplémentaires pour les lecteurs d'écran). Reflow convient bien aux tableaux simples

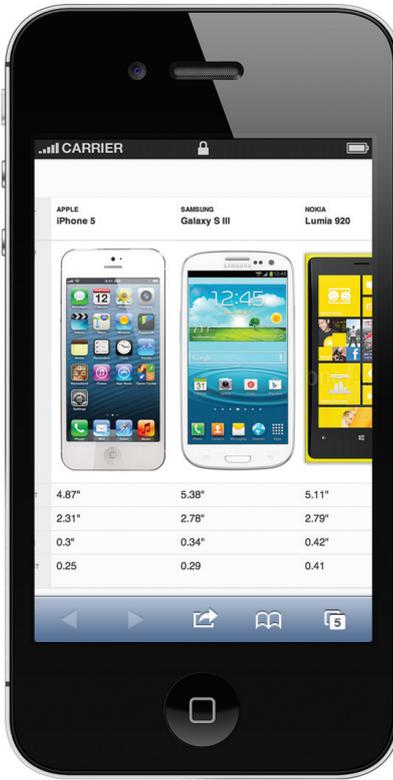


FIG 1.6 : Les tableaux de grande taille peuvent poser des problèmes d'utilisabilité sur les petits écrans.

Rank	1			
Movie Title	Citizen Kane			
Year	1941			
Rating	100%			
Reviews	74			
Rank	2			
Movie Title	Casablanca			
Year	1942			
Rating	97%			

Rank	Movie Title	Year	Rating	Reviews
1	Citizen Kane	1941	100%	74
2	Casablanca	1942	97%	64
3	The Godfather	1972	97%	87
4	Gone with the Wind	1939	96%	87
5	Lawrence of Arabia	1962	94%	87
6	Dr. Strangelove Or How I Learned to Stop Worrying and Love the Bomb	1964	92%	74
7	The Graduate	1967	91%	122
8	The Wizard of Oz	1939	90%	72
9	Singin' in the Rain	1952	89%	85
10	Inception	2010	84%	78

FIG 1.7 : Exemple du modèle de tableau Reflow dans le framework jQuery Mobile, avec le même tableau présenté dans deux largeurs différentes

MODEL	APPLE iPhone 5	Samsung Galaxy S III	Nokia Lumia 920	HTC One X
HEIGHT	4.87"	5.38"	5.3"	
WIDTH	2.31"	2.78"	2.75"	
DEPTH	0.3"	0.34"	0.37"	
WEIGHT (LBS.)	0.25	0.29	0.29	

MODEL	APPLE iPhone 5	SAMBSUNG Galaxy S III	NOKIA Lumia 920	HTC One X
HEIGHT	4.87"	5.38"	5.11"	5.3"
WIDTH	2.31"	2.78"	2.79"	2.75"
DEPTH	0.3"	0.34"	0.42"	0.37"
WEIGHT (LBS.)	0.25	0.29	0.41	0.29

FIG 1.8 : Exemple du modèle de tableau Column Toggle dans le framework jQuery Mobile, avec le même tableau présenté dans deux largeurs différentes

qui agissent comme des listes formatées, mais sa présentation sur petit écran est peu pratique lorsque vous avez besoin de comparer des données sur plusieurs lignes.

Le modèle Column Toggle (<http://bkaprt.com/rrd/1-10/>) vient répondre à ce problème. Il fonctionne en affichant sélectivement les colonnes du tableau dans les limites de l'espace horizontal disponible. S'il n'y a pas assez de place, CSS masque les données des colonnes, mais un menu permet à l'utilisateur d'afficher tout de même les colonnes, et il est alors possible de faire défiler le tableau horizontalement (**FIG 1.8**).

Ce ne sont que deux des nombreux modèles possibles pour présenter le contenu de vos tableaux de manière responsable. Pour plus d'exemples, consultez le projet Responsive Patterns de Brad Frost (<http://bkaprt.com/rrd/1-11/>). Vous y trouverez toutes sortes de solutions, des composants de navigation horizontaux qui s'escamotent dans des menus lorsque l'espace est limité aux grilles CSS Flexbox pour les mises en page complexes.