



Ecole Nationale Supérieure
des Télécommunications

Université Pierre et Marie Curie Paris VI

Nouvelles techniques d'interaction pour environnements Web collaboratifs sur dispositifs mobiles

Etudiant : Mathias Baglioni
Master d'Informatique 2^{ème} année
Intelligence Artificielle et Décision
UPMC Paris VI

Encadrants : Eric Lecolinet, Gilles Bailly
GET / ENST / INFRES
46, rue Barrault
75013 Paris
01 45 81 78 87

Date du stage : 15 avril 2008 au 15 septembre 2008

Remerciements

Je tiens tout d'abord à remercier Eric Lecolinet pour m'avoir permis de réaliser ce stage et m'avoir fait confiance tout au long du projet. Je le remercie également pour son aide et son suivi tout au long du stage. Je tiens à remercier Gilles Bailly pour son investissement et son aide précieuse. Ainsi que sa passion pour les glaces qu'il aura partagée avec tout le monde. Je remercie Karim-Pierre Maalej pour son écoute, sa collaboration et pour sa science. Je remercie Anne Roudaut pour ces conseils et idées, mais aussi pour ses petits moutons et les fou-rires que nous avons eus. Je remercie Sylvain Malacria d'être la seule personne à ne porter que des Nike Dunk assorties à ses tee-shirts. Je tiens à remercier Hayette Sousou pour sa bonne humeur quotidienne. Et pour finir je tiens à remercier tous mes collègues stagiaires pour la bonne ambiance qui a régné tout au long de ce stage. Merci donc à Rémi de Gouvion Saint-Cyr et Olivier Hugues pour leur sympathie et leur écoute. Et enfin merci à Florent Duchiron pour sa science et pour m'avoir supporté avec humour.

I. Sommaire	
REMERCIEMENTS.....	2
I. INTRODUCTION.....	5
1. CONTEXTE.....	5
2. SUJET.....	6
3. OBJECTIFS.....	6
II. PRESENTATION DE L'ORGANISME D'ACCUEIL.....	7
1. PRESENTATION DE L'ÉCOLE.....	7
2. L'ÉQUIPE IHM.....	8
III. ETAT DE L'ART.....	8
1. LES WIKIS.....	8
a) <i>Le moteur de wiki</i>	9
b) <i>XWiki</i>	10
2. LES ENJEUX DE LA MOBILITE.....	10
c) <i>Contraintes</i>	11
d) <i>Quelques solutions</i>	11
3. L'IPHONE.....	12
e) <i>Les gestes</i>	12
f) <i>Safari</i>	14
g) <i>SDK</i>	14
4. TECHNOLOGIES WEB.....	15
a) <i>Contexte</i>	15
b) <i>JavaScript et Canvas</i>	15
5. TECHNIQUES DE MENUS INNOVANTES.....	17
IV. ANALYSE DES BESOINS.....	20
1. QUESTIONNAIRE.....	20
2. FONCTIONNALITES.....	21
c) <i>La navigation</i>	21
d) <i>Le tag</i>	22
e) <i>L'annotation</i>	22
V. CONCEPTION.....	23
1. CONCEPTION DES MENUS.....	23
a) <i>Occupation spatiale</i>	24
b) <i>Interactions</i>	25
2. LES FONCTIONNALITES.....	27
VI. IMPLEMENTATION.....	28
1. PREMIER PROTOTYPE.....	28
a) <i>L'application native</i>	28
b) <i>Le prototype</i>	29
c) <i>Exemple de dessin avec <canvas></i>	30
2. LA BOITE A OUTILS.....	30
a) <i>Implémentation générale</i>	31
b) <i>Le module de reconnaissance de Gestes</i>	32

VII.	GESTION DE PROJET	32
VIII.	CONCLUSION.....	34
	1. BILAN TECHNIQUE.....	34
	2. BILAN PERSONNEL.....	35
	BIBLIOGRAPHIE.....	37
IX.	ANNEXE 1	38
X.	ANNEXE 2	43
XI.	ANNEXE 3	44

I. Introduction

Etudiant en Master Informatique à l'Université Pierre et Marie Curie, j'ai été amené dans le cadre de cette formation à réaliser un stage de fin d'étude de 5 mois dans la continuité du parcours « Interaction et Coopération homme machine » choisi au cours de cette année. Ce stage s'étalant sur la période du 15 avril au 15 septembre 2008 m'a permis de mettre à profit les connaissances acquises au cours de l'année.

1. Contexte

Le laboratoire de recherche où s'est déroulé mon stage est l'ENST (Ecole Nationale Supérieure des Télécommunications), dans le groupe de recherche IC2 (Information, Cognition, Complexity) du département INFRES (informatique et réseaux).

Mon stage s'inscrit dans le cadre du projet XWiki Concerto (Xwiki, 2007-2008). XWiki Concerto est un projet de recherche labellisé RNTL en 2006 et co-financé par l'ANR. XWiki Concerto réunit XPertNet, l'INRIA (équipes ECOO et ATLAS), l'ENST, Mandriva et l'EISTI. Les activités de recherche menées dans le cadre de ce projet permettront au moteur open-source XWiki de supporter le travail collaboratif pair-à-pair en situation de mobilité. Dans ce projet l'ENST s'occupe de la partie mobilité. Nous devons donc assurer le passage de l'application XWiki sur dispositif mobile (Figure 1).



FIGURE 1: ADAPTATION DE XWIKI SUR MOBILE

Mes responsables de stages Eric Lecolinet et Gilles Bailly font tout deux partie du groupe IC2. Eric Lecolinet est maître de conférences et est spécialisé dans le domaine de l'IHM (Interactions Homme-Machine). Gilles Bailly est doctorant en 3^{ème} année dans le même domaine.

J'ai également travaillé en collaboration avec Karim-Pierre Maalej, employé par l'ENST en CDD pour ce projet. L'application développée se divise en deux parties complémentaires : une partie native iPhone, et une application Web associée.

2. Sujet

Le sujet de ce stage intitulé '*Nouvelles techniques d'interaction pour environnements Web collaboratifs sur dispositifs mobiles*' se focalise sur l'adaptation de la plateforme XWiki sur iPhone et plus particulièrement sur le développement de nouvelles techniques de menus dans un environnement Web afin d'en faciliter l'utilisation sur dispositifs mobiles.

3. Objectifs

Nous pouvons donc identifier les étapes de travail suivantes :

- **Réaliser un état de l'art des appareils mobiles utilisés, l'iPhone, des technologies Web et des techniques de menus existantes** afin de bien choisir d'une part le langage de développement (compatibilité avec l'iPhone, XWiki et nos techniques); et d'autre part choisir les techniques de menus que nous allons implémenter dans notre application.
- **Réaliser un questionnaire d'analyse des besoins** afin d'essayer de répondre aux besoins des utilisateurs futurs de l'application.
- **Définir les fonctionnalités à adapter sur dispositif mobile**
- **Concevoir et réaliser un prototype de technique de menus** adapté à l'application XWiki, afin de tester la faisabilité du projet.
- **Concevoir et développer une technique de menus** qui sera intégrée à la plateforme XWiki sur mobile.

II. Présentation de l'organisme d'accueil

1. Présentation de l'école

L'École Nationale Supérieure des Télécommunications (ENST) ou Télécom Paris est une grande école d'ingénieurs publique française, généraliste dans le domaine des Nouvelles Technologies de l'Information et de la Communication (NTIC). Appelée autrefois École Supérieure de Télégraphie, elle a été membre fondateur de ParisTech et fait aujourd'hui partie de la Conférence des Grandes écoles. Elle fait partie du GET (Groupe des écoles des Télécommunications) aux côtés de l'ENST Bretagne, Télécom INT, INT Management et Télécom Lille 1. Le GET constitue un organisme de recherche dans le domaine des technologies de l'information, de leur économie et de leurs usages. L'école est habilitée à délivrer le diplôme d'ingénieur (Bac +5) généraliste, divers MSc) du GET et du groupe ParisTech. Télécom Paris est également un partenaire actif -lié par convention- de plusieurs cursus universitaires de préparation de diplôme du Master. Les élèves étudiants ont ainsi la possibilité de suivre la formation d'un de ces Master, parallèlement à leur 3e année de formation d'ingénieur. Enfin, l'ENST est habilitée à délivrer des doctorats. Son école doctorale, commune avec l'université Paris 6, porte le nom d'École Doctorale d'Informatique, de Télécommunications et d'Électronique (EDITE) de Paris.

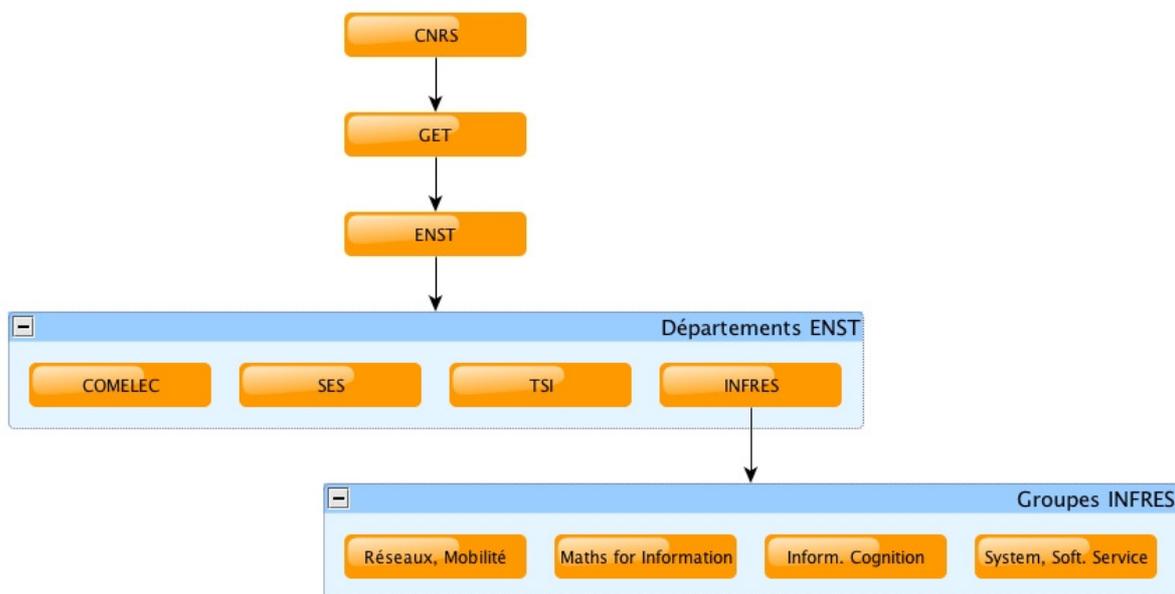


FIGURE 2 : ORGANISATION STRUCTURELLE

2. L'équipe IHM

Éric Lecolinet, spécialisé dans les IHMs, travaille au sein du groupe de recherche IC2 en étroite collaboration avec Yves Guiard, chercheur CNRS en psychologie expérimentale. Il encadre actuellement 4 doctorants :

- Gilles Bailly, Doctorant 3^{ème} année, travaille sur la modélisation de nouveaux menus pour des applications informatiques, et plus particulièrement sur l'élaboration de nouveaux Marking menus (Glossaire 4).
- Tahir Muhammad, Doctorant 3^{ème} année, travaille sur la proposition de nouvelles techniques d'interactions 3D bi-manuelles.
- Anne Roudaut, Doctorante 2^{ème} année, travaille sur la recherche de nouvelles techniques de visualisations et d'interactions sur dispositifs mobiles (Doctorat en partenariat industriel avec Alcatel)
- Sylvain Malacria, Doctorant 1^{er} année, travaille sur la recherche de nouveau cadre d'utilisation du stylo Anoto et papier augmenté dans le cadre du projet ENEIDE.

Karim-Pierre Maalej est lui engagé par l'ENST dans le cadre du projet XWiki Concerto.

III. Etat de l'art

Cet état de l'art se propose d'étudier le contexte dans lequel nous avons mené nos travaux. Il se divise en trois parties. Une première partie nous permettra d'étudier d'une part XWiki : le moteur de wiki centre du projet, d'autre part les enjeux liés à la mobilité et à l'utilisation d'appareils mobiles. Ensuite dans une deuxième partie nous parlerons le dispositif mobile utilisé dans ce projet, l'iPhone, et les technologies web. Enfin la dernière partie nous permettra de présenter les techniques de menus innovantes qui ont inspiré notre application.

1. Les wikis

Un wiki est un système de gestion de contenu de site web. Les pages web sont librement modifiables par tous les visiteurs qui sont autorisés. *On utilise les wikis pour faciliter l'écriture collaborative de documents avec un minimum de contraintes. Le wiki a été inventé en 1995 par Ward Cunningham, pour une section d'un site sur la programmation informatique qu'il a appelée WikiWikiWeb. Le mot « wiki » vient du redoublement hawaïen « wikiwiki », qui signifie « rapide ». Au milieu des années 2000, les wikis ont atteint un bon*

niveau de maturité ; ils sont depuis lors associés au Web 2.0. Créé en 2001, Wikipédia (figure 3) est devenu le site web écrit avec un wiki le plus visité.

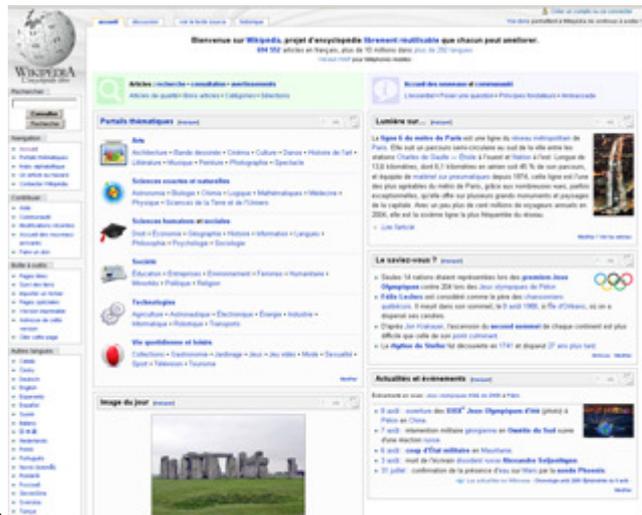


FIGURE 3 : WIKIPEDIA

a) Le moteur de wiki

Il est cependant nécessaire de faire la distinction entre le wiki, le site internet, et le « moteur de wiki » qui est derrière. En effet la plupart des utilisateurs ne connaissent que la partie « immergée de l'iceberg », c'est-à-dire le résultat sous forme de page web du wiki. Alors que ce « moteur de wiki » est ce qui permet de générer la page internet et d'en gérer le contenu. Comme on peut le voir sur la figure 4 le moteur de wiki est au centre du wiki.

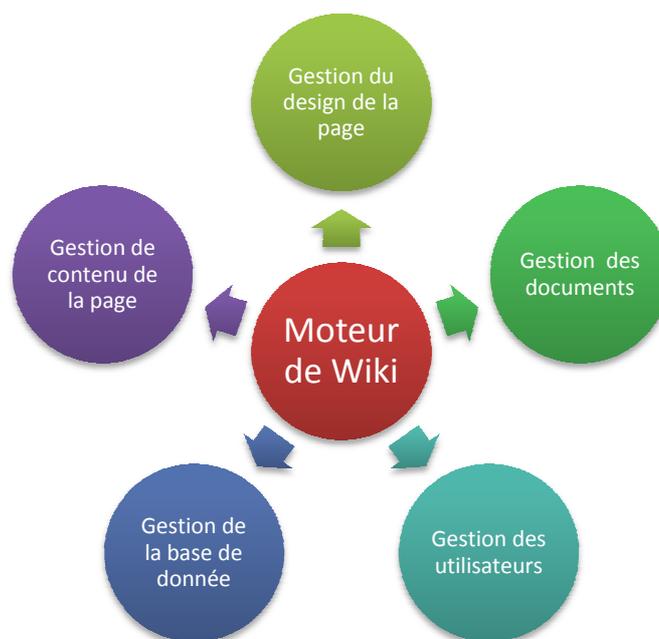


FIGURE 4: MOTEUR DE WIKI AU CENTRE DU WIKI

On trouve un grand nombre de moteurs de wiki dans le monde du web. Parmi les plus utilisés on retrouve Médiawiki, TikiWiki, Xwiki, etc. Le langage utilisé diffère selon le moteur. Par exemple Médiawiki et TikiWiki sont réalisés en PHP alors que XWiki est réalisé en Java. Le domaine d'utilisation diffère aussi selon le moteur de wiki : TikiWiki, facile d'utilisation, sera utilisé pour la réalisation de wikis dans un cadre plus « grand public », et XWiki plus complexe, mais surtout plus complet au niveau des fonctionnalités, sera utilisé dans le domaine de l'entreprise. C'est ce dernier qui nous intéresse dans le cadre du projet XWiki Concerto (figure 5).



FIGURE 5 : XWIKI CONCERTO

b) XWiki

Xwiki est un moteur de wiki OpenSource, réalisé en Java. Il intègre les fonctionnalités de base des wikis comme la gestion de contenu, la gestion de version, la gestion de droits, ou encore la recherche textuel. Il possède d'autre part beaucoup de fonctions avancées présentes dans peu de wikis, comme par exemple les fonctions qui sont le plus liées au projet comme la gestion de la personnalisation de l'interface (skin, CSS), une API de programmation en Velocity.

2. Les enjeux de la mobilité

L'utilisation de petits appareils en situation de mobilité soulève des enjeux qu'il est important de présenter. Aujourd'hui un grand nombre de dispositifs mobiles tels que les PDAs ou SmartPhones (figure 6) disposent d'écrans tactiles avec lesquels on peut interagir avec un stylet ou avec les doigts.



FIGURE 6 : PDA , SMARTPHONE ET IPHONE

c) Contraintes

Cependant ces deux techniques d'interaction avec les mobiles apportent des contraintes. L'utilisation du stylet oblige l'utilisateur à se servir de ses deux mains pour interagir avec son appareil, et demande une grande attention (surtout si l'utilisateur se déplace). Ce n'est donc pas la meilleure façon d'interagir avec son mobile. L'utilisation au pouce semble résoudre le problème d'utilisation des deux mains, cependant il engendre aussi des contraintes : la perte de précision du pointage, l'occlusion de l'écran et la difficulté pour atteindre les bords de l'écran sur certains mobiles. De plus la plupart des interfaces sur mobiles sont pensées pour une utilisation au stylet.

d) Quelques solutions

Plusieurs techniques ont été proposées pour répondre à ces problèmes. Par exemple la technique « Shift » (Vogel, et al., 2007) propose de déplacer le contenu occulté par le doigt dans une partie visible de l'écran à côté du doigt comme on peut le voir sur la Figure 7.

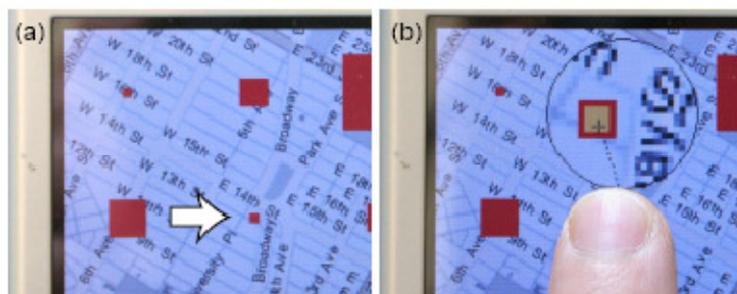


FIGURE 7 : SHIFT TECHNIQUE

D'autres recherches ont permis de mettre en valeur l'importance de la taille des cibles (Parhi, et al., 2006) avec lesquelles l'utilisateur doit interagir sur mobile, la difficulté d'atteindre certaines zones de l'écran ou de reproduire certains gestes (Karlson, et al., 2006).

Toutes ces contraintes ont été prises en compte tout au long de la réalisation de notre projet, notamment dans le choix de l'utilisation de l'iPhone. En effet comme nous allons le voir, il répond à certaines de ces contraintes.

3. L'iPhone

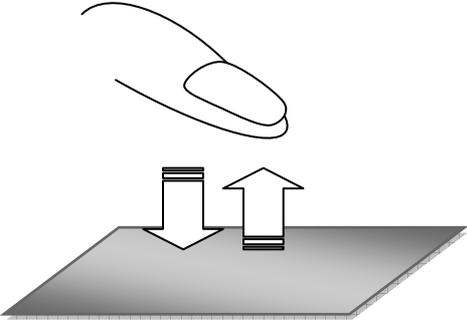
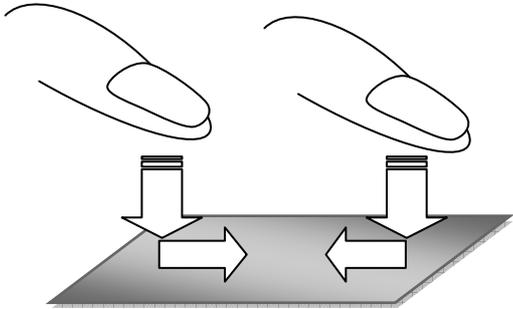
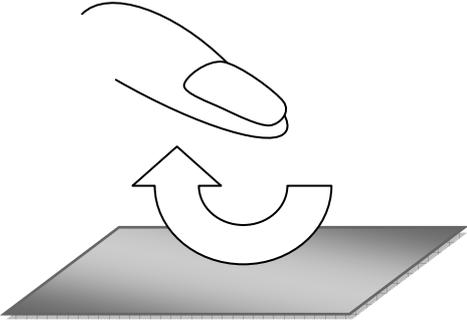
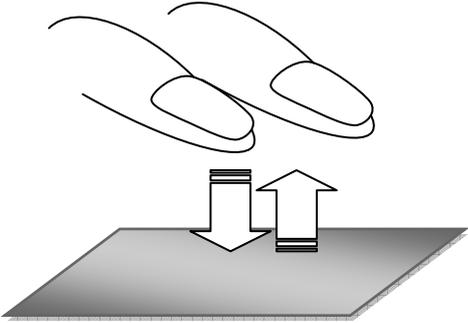
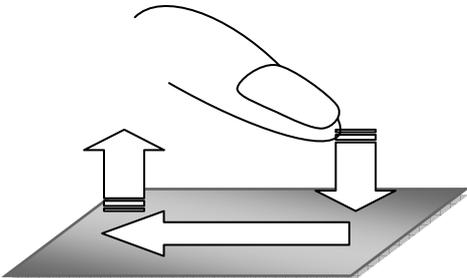
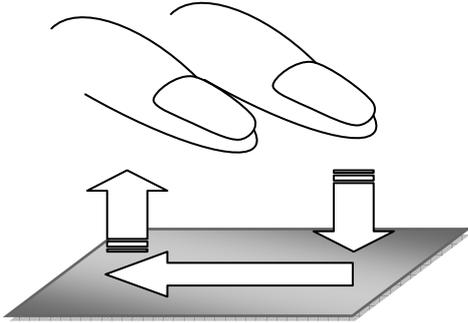
L'iPhone (Figure 8) (Apple Inc., 2008) fait partie des téléphones nouvelle génération (smartphones) incorporant un grand nombre de fonctionnalités. Muni d'un écran multi-toucs, de capteurs de lumière et de distance et d'un accéléromètre, il permet une grande variété d'interactions. De plus il répond à certains problèmes soulevés dans la partie précédente. L'iPhone est en effet orienté pour une utilisation au doigt, et ne fonctionne pas avec un stylet. On notera aussi qu'il ne possède pas de bords autour de l'écran ce qui facilite les interactions à ces endroits.



FIGURE 8 : IPHONE 3G

e) Les gestes

Comme nous l'avons dit précédemment il possède un écran multi-touch permettant une grande variété de gestes et d'interactions réalisables. Plusieurs gestes sont reconnus par l'iPhone : le « tap », le « swipe », le « drag », le « pinch » et le « double tap ». Ils peuvent se regrouper en deux catégories, selon qu'ils utilisent un ou plusieurs doigts. Le tableau suivant propose une classification selon ces critères.

Un doigt	Plusieurs doigts
<p>Tap : on « tape » l'écran avec le doigt. Le doigt ne doit pas rester posé sur l'écran.</p> 	<p>Pinch : On pose deux doigts sur l'écran et on les rapproche ou on les écarte.</p> 
<p>Swipe : on « caresse » l'écran avec le doigt. C'est un mouvement rapide.</p> 	<p>Double tap : comme le tap mais avec deux doigts simultanément.</p> 
<p>Drag : on pose le doigt sur l'écran et on le déplace dessus avant de le relever.</p> 	<p>Drag à deux doigts : comme le drag avec un doigt.</p> 

Pour notre étude nous nous concentrerons sur les gestes à un doigt qui correspondent à la problématique de l'utilisation en situation de mobilité. Nous allons voir maintenant un deuxième élément important de l'iPhone, Safari.

f) Safari

Safari est à l'origine un navigateur web prévu pour les ordinateurs. Il est réalisé autour du moteur de rendu Webkit. Cependant sur l'iPhone ce navigateur a dû être adapté. Il permet contrairement au navigateur web présent sur les autres dispositifs mobiles d'afficher une page web telle que nous avons l'habitude de la voir sur un ordinateur, bien qu'il apporte quelques changements au niveau de l'interaction qui diffèrent de sa version pour ordinateur.

Les événements

En effet l'interaction d'entrée étant différente (écran tactile/souris clavier) les événements clavier ou souris habituels n'ont pas une correspondance directe. Déplacer le pointeur de la souris sur l'écran ne correspond pas à déplacer le doigt sur l'écran. C'est pourquoi dans la première version de Safari sur iPhone ce type d'événements n'est pas géré au niveau de la page Web. Les applications web nécessitant ce type d'interactions sont donc mises à l'écart. C'est un point très important dans le cadre de notre projet puisque nous devons réaliser une application web permettant ces interactions. Nous verrons dans la partie conception comment nous avons contourné ce problème en créant un navigateur web simplifié pour iPhone.

Evolutions

Cependant, au cours du stage (juillet 2008), Apple a apporté des évolutions à son navigateur, notamment concernant la gestion des différents gestes grâce à une API JavaScript. Cela nous a permis de tester directement notre application sous Safari, tout en nous obligeant à revoir l'implémentation de notre application.

Limitations

Safari apporte aussi quelques limitations au niveau des technologies utilisables pour réaliser des applications web. Il n'intègre pas de « plugins » permettant la lecture d'applications J2E, Flash, Flex, Silverlight, etc, ce qui, comme nous le verrons dans la partie III.4 nous a orienté vers l'utilisation de JavaScript.

g) SDK

Afin de pouvoir développer sur iPhone, Apple fournit un SDK utilisable sous Mac seulement, qui ne permet pas de tester directement sur l'iPhone sans avoir une licence. Or jusqu'à la sortie de l'iPhone 3G et du firmware 2.0 pour iPhone et iPod Touch, il était impossible de se procurer ce type de licence. Nous avons donc dû nous contenter du simulateur fourni avec le SDK et d'un iPod Touch « jailbreaké » (i.e. permettant d'y installer des applications).

4. Technologies Web

Bien qu'il existe aujourd'hui un grand nombre de technologies Web permettant de réaliser des applications avancées, il nous fallait rester cohérent avec d'une part la technologie utilisée pour XWiki et d'autre part celle compatible avec l'iPhone.

a) Contexte

Ces technologies demandent en générale un module supplémentaire permettant de les interpréter. Pour des technologies comme Flash, Flex, AJAX (JavaScript) l'interpréteur se situe au niveau client, c'est-à-dire que c'est le navigateur de l'utilisateur qui doit avoir la capacité de les exécuter, soit par l'ajout de plugins (ex : flash), soit parce que le moteur web peut les interpréter (JavaScript). En revanche les technologies comme PHP étant interprétées au niveau serveur, le client ne doit rien posséder pour qu'elles s'exécutent.

Or, comme nous l'avons vu plus haut le moteur de rendu Web de l'iPhone, WebKit possède, un interpréteur JavaScript mais ne possède pas encore de plugins permettant la lecture et l'exécution d'applications Flash, Flex, Silverlight, etc. Nous avons donc choisi JavaScript comme langage compatible avec l'iPhone. Il permet aussi une intégration plus aisée au moteur XWiki. Nous allons voir de plus près ce langage.

b) JavaScript et Canvas

JavaScript est un langage de programmation de scripts essentiellement utilisé pour la réalisation de pages web dynamiques. Basé sur la norme ECMA-262, et sur le langage Java, il simplifie la programmation orientée client. Créé en 1995 par Brendan Eich, il s'est développé au cours des années qui ont suivies, adoptant de nouvelles fonctionnalités parallèlement à l'évolution de la puissance des ordinateurs. Délaissé quelque temps pour les applications Web riches comme Flash, il a retrouvé sa place grâce à l'apparition de bibliothèques graphiques et d'animations telles qu'AJAX (figure 9), Mootools (mad4milk, 2008), DOJO (The Dojo Foundation, 2008), etc. Il est aussi maintenant possible de réaliser des animations complexes sur la page web et de gérer un accès asynchrone avec une base de données. On ne doit plus recharger toute la page pour pouvoir accéder à de nouvelles informations ou pour pouvoir déplacer du contenu.

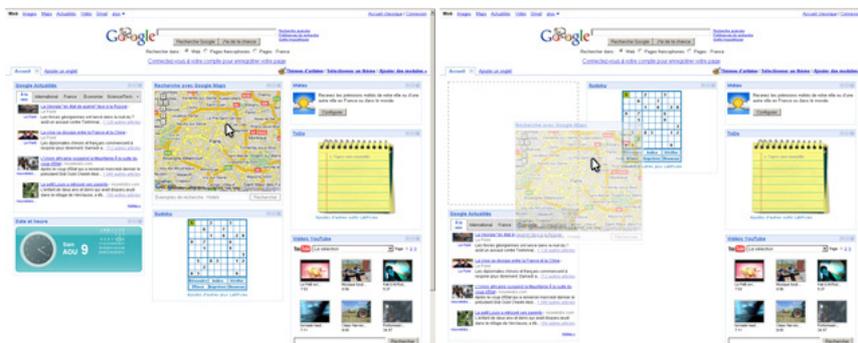


FIGURE 9 : IGOOGLE REALISEE EN AJAX PERMET DE DEPLACER LES ELEMENTS DE LA PAGE

De plus JavaScript permet, lorsqu'il est couplé avec la balise (objet) HTML <canvas>, de réaliser des dessins et des animations complexes (figure 11). Comme exemple nous trouvons les « Widgets », utilisés par Apple dans son « Dashboard » (figure 10), réalisés avec la balise <canvas>.



FIGURE 10 : DASHBOARD APPLE



FIGURE 11 : ANIMATION REALISEE AVEC LA BALISE CANVAS

JavaScript est donc un langage approprié pour développer des techniques de menus innovantes.

5. Techniques de menus innovantes

Avec la plupart des logiciels, nous sommes amenés constamment à utiliser des menus pour sélectionner les commandes. Ces menus peuvent être attachés à la « barre de menus » (généralement positionnée en haut des fenêtres) ou contextuels (ils sont alors généralement activés par un clic droit au dessus d'un objet d'intérêt). Les menus linéaires (figure 12) organisent les commandes verticalement de haut en bas. Lorsque le nombre de commandes est trop important, ils peuvent avoir une structure hiérarchique et sont alors composés de sous-menus cascades.



FIGURE 12: MENU LINEAIRE

Il existe de nombreux menus alternatifs dans le domaine de la recherche en IHM pour améliorer les performances (rapidité, précision, apprentissage, mémorisation, satisfaction) des traditionnels menus linéaires. Les plus célèbres sont certainement les Marking menus (Figure 13).

Les Marking menus proposent d'augmenter les menus classiques par des gestes experts. On trouve de manière générale plusieurs modes d'accès aux commandes d'un menu : un mode novice où l'utilisateur visualise le menu déroulé afin de choisir la commande, et un mode expert où l'utilisateur accède directement à la commande sans prévisualisation. Avec les menus linéaires le mode expert correspond à l'utilisation des raccourcis clavier associés aux commandes. Avec les Marking menus le mode expert est représenté par des gestes permettant d'accéder aux commandes. Les Marking menus disposés de manière circulaire permettent en moyenne d'activer plus rapidement les commandes et favorisent une meilleure mémorisation de leur utilisation. En effet les gestes à réaliser correspondent à l'emplacement de la commande dans le menu. Comme on peut le voir sur la figure 13 la commande copier se situe vers le haut du menu. Pour l'activer en mode expert on réalise un geste allant de bas en haut. Ces menus peuvent également être hiérarchiques.

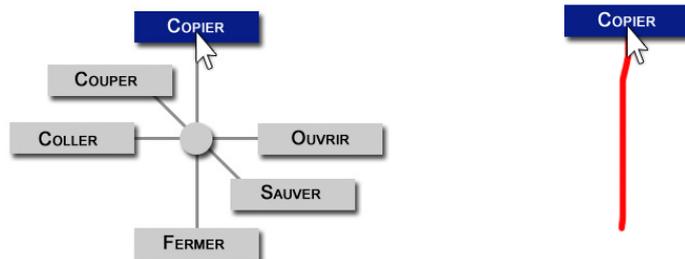


FIGURE 13: MARKING MENU MODE NOVICE ET EXPERT

Ils comportent cependant quelques inconvénients : Le nombre de commandes par niveau est limité (au dessus de 8 l'interaction devient problématique) et l'espace utilisé lors du parcours de sous-menus devient vite grand. D'autres types de Marking menus tentent de répondre à ces problèmes. Par exemple le Flower Menu (figure 14) (Bailly, et al., 2008) propose d'ajouter des gestes courbes afin d'augmenter le nombre de commandes disponibles à chaque niveau.

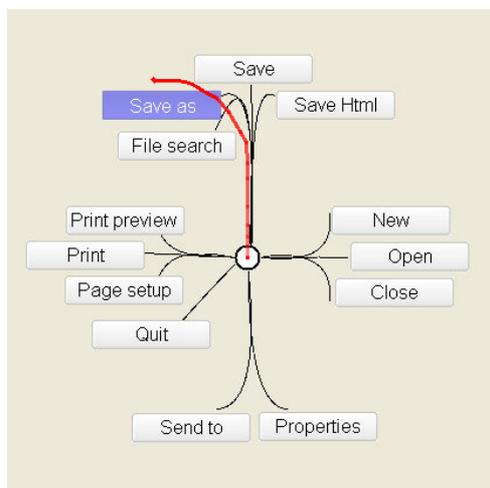


FIGURE 14: FLOWER MENU

Cependant, les terminaux et usages mobiles posent des problèmes spécifiques et notamment du fait de la petite taille de l'écran. Le Leaf Menu (figure 15) (Bailly, et al., 2008), qui n'est pas un Marking Menu mais en reprend certaines fonctionnalités a été développé en ce sens. Il propose en effet d'augmenter un menu linéaire classique par des gestes afin de pouvoir disposer d'un mode expert sur mobile.

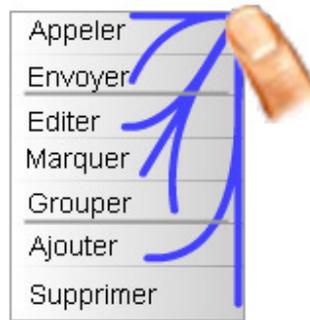


FIGURE 15: LEAF MENU

D'autres menus, comme les Arch Menu et Thumb Menu, ont été proposés pour répondre à l'utilisation des menus sur mobiles avec le pouce (occultation, surface d'appui importante) (Huot, et al., 2007).

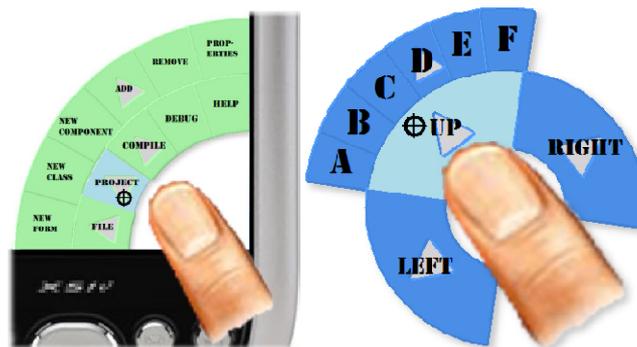


FIGURE 16 : ARCH ET THUMB MENU

Enfin « les Control menus (figure 17) sont des variantes des traditionnels Marking menus qui permettent à la fois de sélectionner et de contrôler des opérations en un seul geste » (Lecolinet, et al., 2002), c'est-à-dire sans relâcher la souris. Cette technique permet une interaction continue qui facilite, entre autre la manipulation de fonctions continues (comme par exemple le zoom) sur dispositifs mobiles.

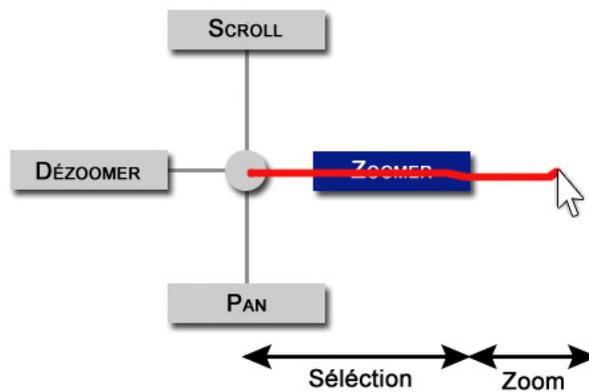


FIGURE 17 : CONTROLE MENU

Ces travaux de recherche nous ont inspirés pour concevoir deux nouvelles techniques optimisés pour dispositifs mobiles : le DockMenu et le CornerMenu qui seront détaillés à la section V.

IV. Analyse des Besoins

Dans le cadre de notre projet il était important de bien définir les besoins des futurs utilisateurs. C'est pourquoi nous avons utilisé une conception centrée utilisateur, grâce à la méthode du questionnaire.

1. Questionnaire

Un questionnaire (Annexe) a donc été réalisé afin d'essayer de répondre aux demandes des utilisateurs. Ce questionnaire se divise en plusieurs parties :

- Contexte d'évolution de l'utilisateur
- Utilisation des mobiles
- Utilisation des wikis
- Utilisation des wikis sur mobile

La première partie permet de situer l'utilisateur, son âge, son domaine de travail et son affinité avec les technologies en jeu dans notre projet. Ensuite dans la deuxième partie nous tentons de voir quelle utilisation l'utilisateur fait de ses terminaux mobiles : si c'est simplement pour téléphoner ou s'il les utilise aussi pour consulter ses mails et naviguer sur internet. Ensuite nous cherchons à savoir quelle utilisation il pourrait avoir des wikis, s'il les utilise pour le travail, simplement pour la consultation, ou pour de la gestion. Enfin, nous

demandons à l'utilisateur quelles fonctionnalités il souhaiterait voir adaptées des wikis sur ordinateur vers leur adaptation sur mobile.

Malheureusement nous n'avons eu qu'un nombre limité de réponses (sans doute à cause de la période estivale). Malgré tout nous avons pu obtenir des réponses intéressantes venant d'autres partenaires du projet Xwiki concerto lesquels nous ont fait parvenir les résultats d'un questionnaire réalisé en 2007 sur l'aspect général du projet. On a pu y trouver certaines réponses intéressantes concernant notamment la mobilité.

Finalement, grâce à ces deux questionnaires, nous avons pu cibler les fonctionnalités importantes à implémenter dans notre application.

2. Fonctionnalités

Choisir les fonctionnalités adéquates pour l'utilisation d'un wiki sur mobile n'est pas aisé. Cependant nous avons réussi grâce à l'étude des besoins et des technologies, à choisir trois fonctionnalités principales : la navigation, le tag, les annotations. Nous allons maintenant voir ces trois fonctionnalités en détails.

c) La navigation

Cette fonctionnalité permet d'accéder aux différentes pages d'un site web. Bien qu'elle paraisse évidente lorsqu'il s'agit de « naviguer » sur un site web, elle l'est beaucoup moins lorsque l'on doit « naviguer » sur un mobile. En effet, le contenu réservé à la navigation d'une page web comme celle d'un wiki prend beaucoup de place, comme on peut le voir sur la figure suivante en rouge.

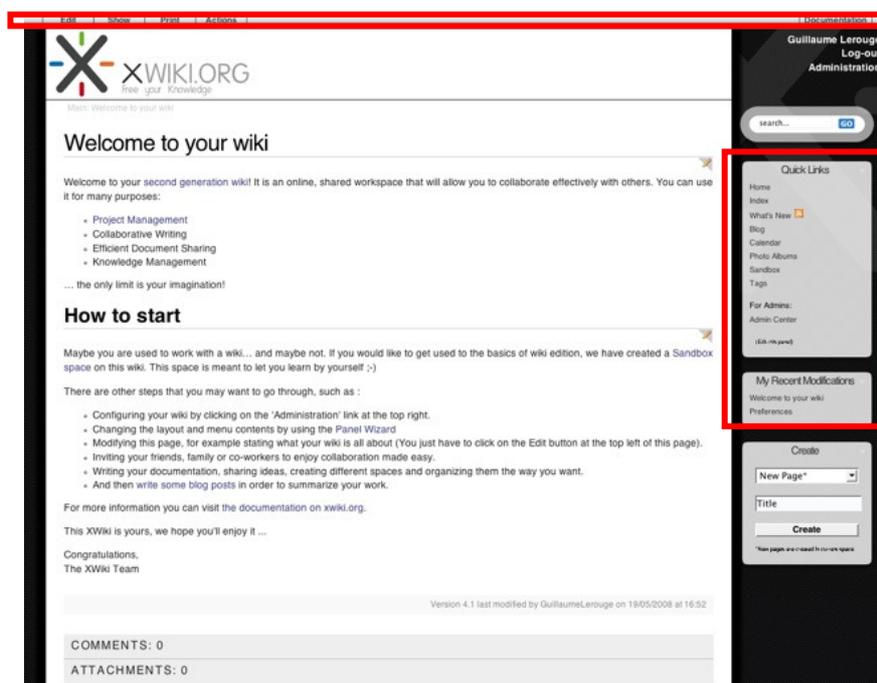


FIGURE 18 : LOCALISATION DES MENUS DE NAVIGATION

Or, sur dispositif mobile, il devient impossible de garder une telle disposition. Etant donné la petite taille de l'écran, les commandes seraient inaccessibles. Cependant, nous ne pouvons pas négliger cette fonctionnalité. Il faut donc trouver un autre moyen d'accéder à ces commandes.

d) Le tag

Le « tag » consiste à apposer sur une page web un ou plusieurs mots qui définissent le contenu de la page. Ainsi, il devient aisé de retrouver une page en fonction du contenu, simplement en recherchant le mot qui la définit. Cette fonctionnalité est très utilisée dans les sites web proposant du contenu non textuel comme les photos ou les vidéos, comme on peut le voir en rouge sur la figure du site YouTube.

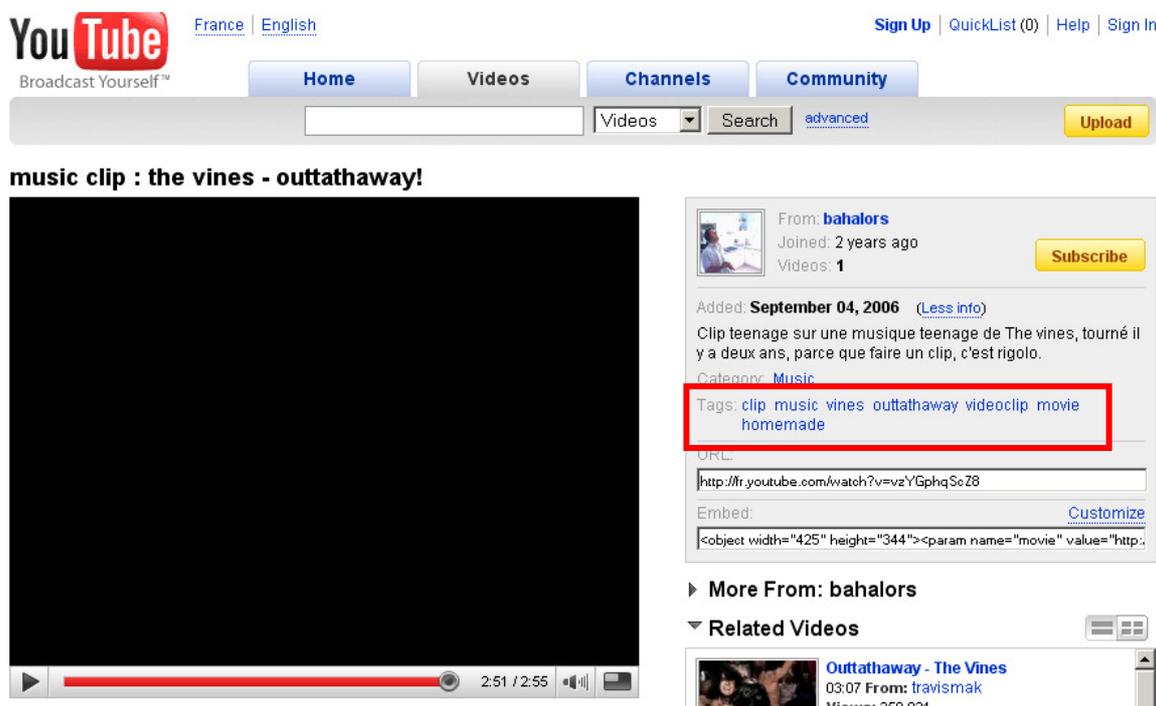


FIGURE 19 : TAG DE VIDEO YOUTUBE

Dans le cadre de XWiki concerto on trouve sur la page du wiki un espace réservé pour permettre le tag de la page. Là encore, si l'on se place dans le contexte de la mobilité rajouter sur chaque page un cadre réservé prend de la place pour le contenu textuel et rend difficile l'interaction.

e) L'annotation

L'annotation est une fonctionnalité peu répandue mais qui fait aussi l'originalité XWiki. Cette fonctionnalité permet de laisser une annotation sur une partie d'une page web. Elle se rapproche de la fonctionnalité de commentaire qui consiste à écrire un commentaire sur une page web en rapport avec son contenu à une place réservée (figure 21), mais s'en différencie en permettant d'écrire l'annotation à l'endroit où l'on veut sur la page. On

retrouve un concept similaire dans le site Flickr permettant de laisser une annotation sur une photo (figure 20). Ainsi, l'annotation a une plus forte valeur contextuelle que le commentaire.

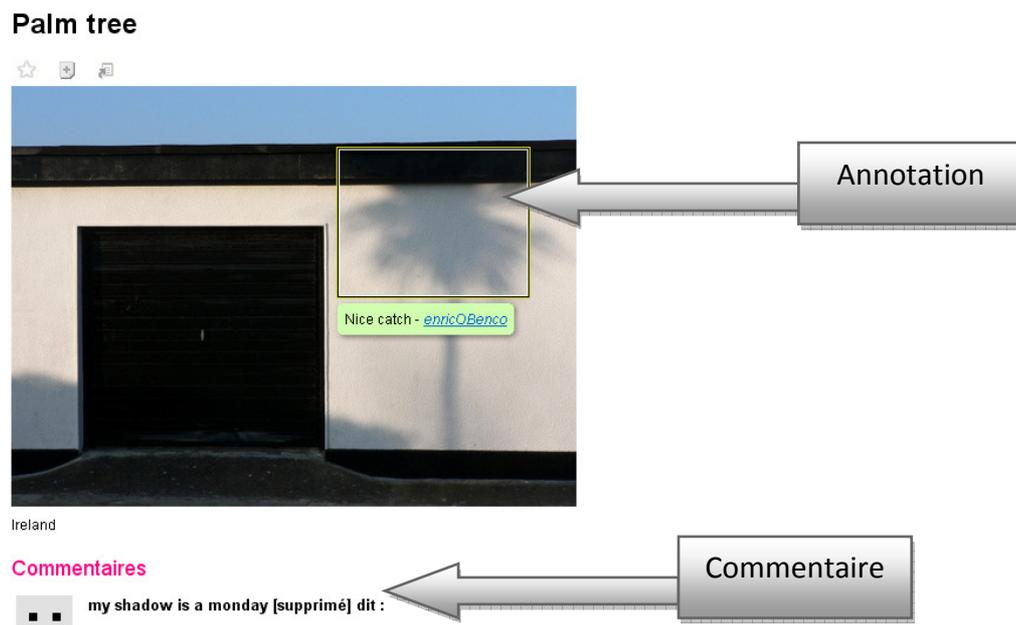


FIGURE 20 : ANNOTATION ET COMMENTAIRE

Nous avons donc vu les trois fonctionnalités que nous souhaitons garder, tout en soulevant le problème de leur intégration à la page Web. Nous allons donc expliquer dans la partie suivante comment nous avons intégré ces fonctionnalités.

V. Conception

Une fois les fonctionnalités définies il faut trouver un moyen de les intégrer en respectant les contraintes de mobilité et d'utilisation sur iPhone. Nous ne pouvons pas les intégrer directement à la page web comme nous l'avons vu dans la partie précédente. Nous avons donc choisi de réaliser des techniques de menus regroupant ces fonctionnalités. Ainsi, comme nous allons le voir, nous préservons l'espace nécessaire à l'affichage de la page web et nous optimisons leur utilisation.

Notons que la fonctionnalité d'annotation est nommée « post-it » dans notre application.

1. Conception des menus

Le Corner Menu et le Dock Menu regroupent donc les mêmes fonctionnalités, vues précédemment. Ces deux menus tentent de répondre à deux critères essentiels : l'espace occupé et la facilité d'interaction, chacun favorisant l'un des deux critères.

a) Occupation spatiale

Le Corner Menu placé dans les coins de l'écran (figure 21) occupe peu d'espace et préserve la lecture de la page web alors que le Dock Menu (figure 22) occupe plus de place et donc donne une visibilité plus faible.



FIGURE 21 : LOCALISATION DU CORNER MENU



FIGURE 22 : LOCALISATION DU DOCK MENU

Ces deux menus ont la particularité de pouvoir être placés à des endroits différents en fonction de l'utilisateur (gaucher/droitier) et de ses préférences.

b) Interactions

Afin de rendre l'interaction avec ces menus plus aisée nous les avons dotés d'un mode novice et d'un mode expert et de concepts issus des control menus (cf. III.5). Ainsi nous avons deux moyens d'accéder aux commandes (fonctionnalités). Soit on fait apparaître le menu avec un « tap » (cf. III.3) et on « tap » sur la commande choisie pour l'activer, soit on réalise un geste expert pour activer la commande. Il convient donc de définir les gestes associés aux commandes.

Les Gestes

Les gestes utilisés pour nos menus sont inspirés des gestes que l'on trouve dans les Marking menus et dans le leafMenu (cf. III.5). Ainsi, comme on peut le voir sur la figure 23, il y a une relation entre la position de l'item dans le menu et le geste qui lui est associé. Dans cette figure nous retrouvons les gestes vus dans les Marking Menus.

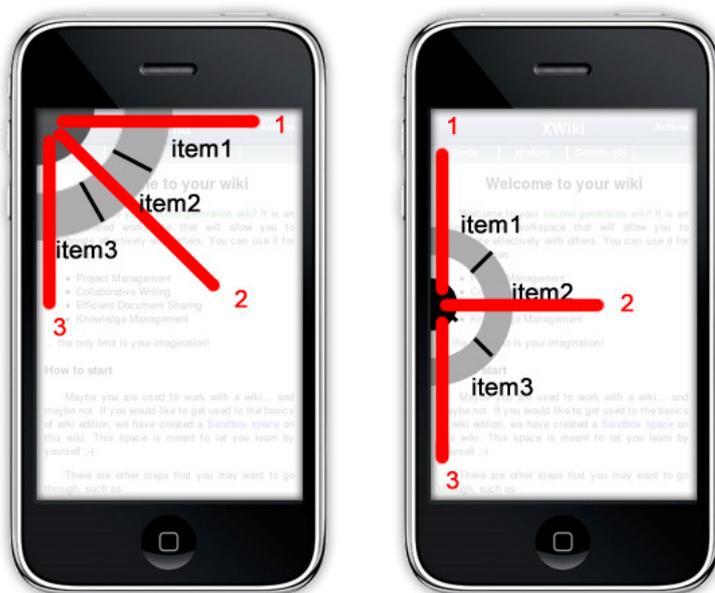


FIGURE 23 : GESTES ASSOCIES AU CORNERMENU ET AU DOCKMENU

Dans le cas où l'application nécessiterait davantage de commandes, des gestes courbes comme dans le Leaf Menu sont utilisés pour le Corner Menu (figure 24), mais pas pour le Dock Menu (figure 24). C'est là que le Dock Menu apporte une plus grande facilité d'interaction que le Corner Menu.

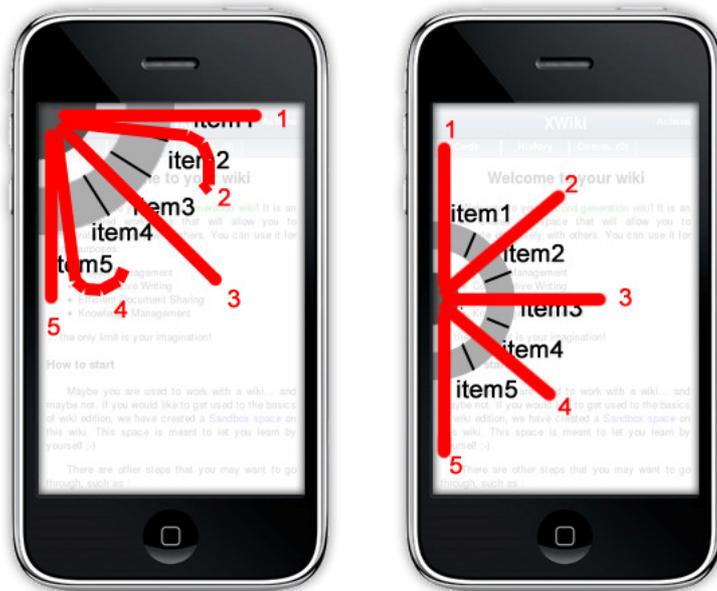


FIGURE 24 : GESTES COURBES SUR LE CORNERMENU

Les control menus

Nous avons montré plus haut que les fonctionnalités « tag » et « post-it » étaient contextuelle. Or nos menus ne le sont pas. Ils sont placés à un endroit précis de l'écran et n'apparaissent que si on les active dans cette zone. Nous avons donc implémenté ces fonctionnalités comme dans les control menus (cf.III.5) : c'est-à-dire en prolongeant le geste réalisé pour activer la commande afin de déplacer le « tag » ou le « post-it » sur la page, comme on peut le voir sur la figure 26 avec l'exemple du « post-it ».



FIGURE 25 : SELECTION ET DEPLACEMENT DU "POST-IT"

2. Les fonctionnalités

Nous avons donc vu comment intégrer ses trois fonctionnalités à la page web. Maintenant nous allons définir ce qu'il est possible de faire avec. Nous insisterons surtout sur la fonctionnalité d'annotation, « post-it ».

La navigation reprend le principe des menus de navigation vus plus haut. Nous accédons par notre menu aux différentes pages du site. De plus, l'utilisateur aura la possibilité de rajouter des « bookmark » (pages favorites) par l'application native (cf. I.3).

Le tag permettra donc de « tagger » une page ainsi que de « tagger » une annotation.

Enfin le « post-it » permet de laisser une annotation sur la page. Nous voulions également que l'utilisateur ait la possibilité en sélectionnant un « post-it » présent sur la page, de le déplacer (figure 26), de l'éditer (figure 27), ou de le supprimer (figure 28) et enfin de choisir les droits du « post-it » : public, privé, groupe (figure 29). (C'est-à-dire de décider qui pourra voir ce « post-it » en consultant la page où il se trouve).

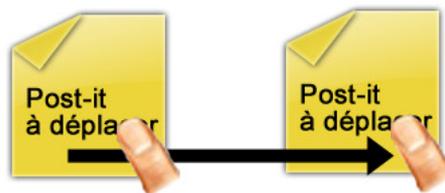


FIGURE 26 : DRAG AND DROP



FIGURE 27 : ÉDITION DU TEXTE



FIGURE 28 : BOUTON DE SUPPRESSION



FIGURE 29 : SELECTION DES DROITS DU POST-IT

VI. Implémentation

Nous allons dans cette partie présenter les principaux modules de l'implémentation des techniques développées. Nous décrirons d'abord le premier prototype qui a permis entre autre de tester la faisabilité de notre conception. Ensuite nous verrons comment ces menus ont été implémentés.

1. Premier prototype

Nous avons d'abord réalisé un premier prototype afin de se familiariser avec le contexte technique et matériel du projet. N'ayant pas la possibilité de tester directement sur l'iPhone par le biais du SDK fourni par Apple, nous avons dû trouver une solution alternative. C'est pourquoi nous avons « Jailbreaké » (cf. état de l'art) un ipod touch afin de pouvoir développer une application native permettant de lire des pages HTML et, surtout, d'envoyer les événements d'interactions (tap, drag, swipe) à la page Web. Comme nous l'avons vu dans la partie III, nous ne pouvons pas utiliser Safari comme plateforme de test. Ensuite nous avons pu réaliser un premier menu de test.

a) L'application native

Nous avons réalisé une application en Objective C capable d'être exécutée sur le firmware 1.x de l'iPhone (ou iPod Touch). Cette application comprend trois « vues » : la vue globale qui sert de fenêtre à l'application, la vue qui gère les interactions et la vue qui gère l'affichage de la page web. Ces trois vues sont imbriquées les unes dans les autres.

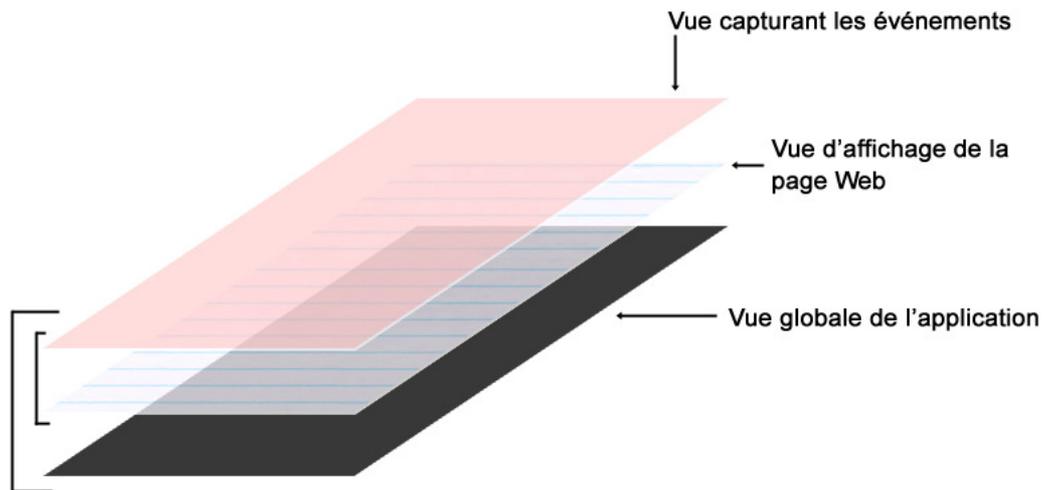


FIGURE 30 : DISPOSITION DES VUES EN COUCHE

Avec ce modèle nous faisons le lien entre les gestes réalisés par l'utilisateur et la page Web. En effet, les interactions récupérées par la vue de gestion des événements sont renvoyées à la page Web et permettent l'exécution de fonctions JavaScript.

b) Le prototype

Nous avons réalisé un prototype de Marking menu circulaire (figure 31) permettant de réaliser des commandes basiques : défilement de la page et changement de la couleur du texte de la page.

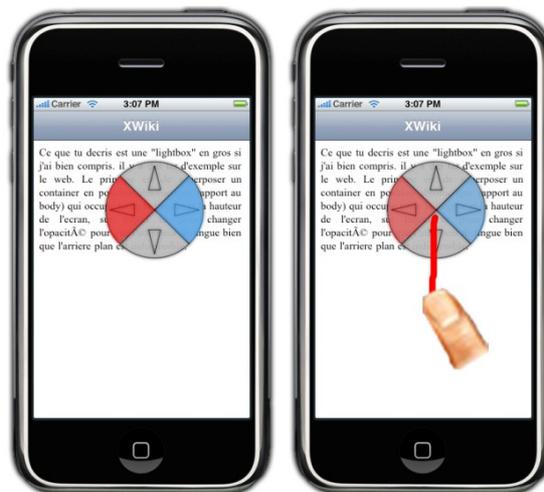


FIGURE 31 : MARKING MENU CIRCULAIRE

c) Exemple de dessin avec <canvas>

Nous allons voir comment on accède en JavaScript au canvas et comment on peut dessiner un motif simple.

Définition d'une balise canvas dans la page HTML.

```
<canvas id="canvas" width="300" height="300"></canvas>
```

Récupération de l'objet canvas .

```
var canvas = document.getElementById("canvas");
```

Récupération du contexte graphique 2d du canvas.

```
var ctx = canvas.getContext("2d");
```

Définition du style avec lequel on trace le premier rectangle.

```
ctx.fillStyle = "rgb(200,0,0)";  
ctx.fillRect (10, 10, 55, 50);
```

Définition du le style avec lequel on trace le second rectangle

```
ctx.fillStyle = "rgba(0, 0, 200, 0.5)";  
ctx.fillRect (30, 30, 55, 50);
```

Ci-dessous le résultat :



FIGURE 32:CANVAS

Nous avons pu ainsi tester si nos concepts de menus étaient réalisables d'un point de vue technique et avec un niveau de performances acceptable. Nous allons maintenant voir leur implémentation de plus près.

2. La boîte à outils

Afin de développer ces deux techniques nous avons d'abord développé une « boîte à outil » (cf VI.1). Il ne s'agit pas ici de faire un descriptif exhaustif du code développé pour la

réalisation des menus mais plutôt de présenter l'intérêt de l'implémentation que nous avons choisie.

a) Implémentation générale

Nous sommes partis du constat qu'il n'existe pas de librairie ou « toolkit » permettant de réaliser des menus « avancés » en utilisant le couple JavaScript/<canvas>. Nous avons alors choisi d'orienter notre implémentation pour répondre à ce manque. C'est pourquoi, plutôt que de ne développer que les menus choisis, nous avons développé une « boîte à outils » permettant de réaliser des techniques d'interaction nouvelles appropriées la réalisation de l'objectif final. Elle est dotée de plusieurs « modules » (figure 34):

- Un module Classe : définissant une classe abstraite de menu, ainsi qu'une classe de définitions de propriétés qui serviront à créer les différents menus.
- Un module Evénements : permettant de gérer les événements de « touch » de l'écran.
- Un module HTML : permettant de gérer le contenu HTML (création, suppression d'élément HTML).
- Un module Auxiliaire : regroupant quelques fonctions utilitaires (gestion d'effacement des <canvas>).
- Un module Calcul : regroupant les différentes fonctions de calcul de distance, et d'angles
- Un module Gestures : permettant de reconnaître les gestes réalisé par l'utilisateur.

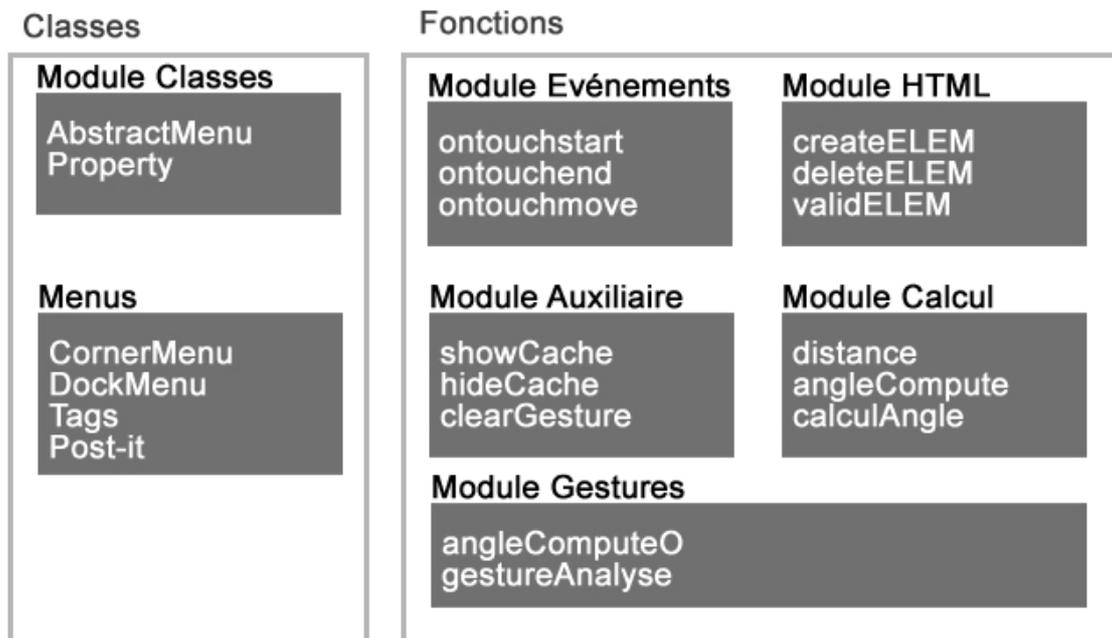


FIGURE 33 : SCHEMA GENERAL

Le diagramme de classes de l'application est en Annexe 2.

b) Le module de reconnaissance de Gestes

Ce module est très important, car c'est lui qui permet de savoir quel item est sélectionné, si le menu doit être activé et quel geste l'utilisateur a réalisé. Il possède plusieurs fonctions de calcul d'angle, de distance et une fonction d'analyse de geste. Nous allons surtout regarder la fonction de reconnaissance des gestes.

Comme ces menus utilisaient les mêmes gestes que le LeafMenu, j'ai regardé l'algorithme de reconnaissance qui lui était associé. Ecrit en C#, il n'était pas directement utilisable dans notre cas. J'ai donc décidé, après avoir discuté avec Gilles Bailly et Anne Roudaut des éléments importants à la réalisation de cet algorithme, de développer un nouvel algorithme optimisé pour nos techniques et le langage utilisé (JavaScript). Durant les recherches sur le Leaf Menu il avait été montré que l'angle de départ du tracé (les 2 tiers du tracé) et celui fait par la fin du tracé suffisaient à déterminer le geste réalisé. Il fallait donc récupérer la liste des points, définir deux vecteurs et calculer leurs angles respectifs. Notre algorithme fonctionne par élimination comme on peut le voir sur la figure 34. Il procède par passage successifs jusqu'à ce qu'il ne reste qu'un ou zéro geste reconnu.

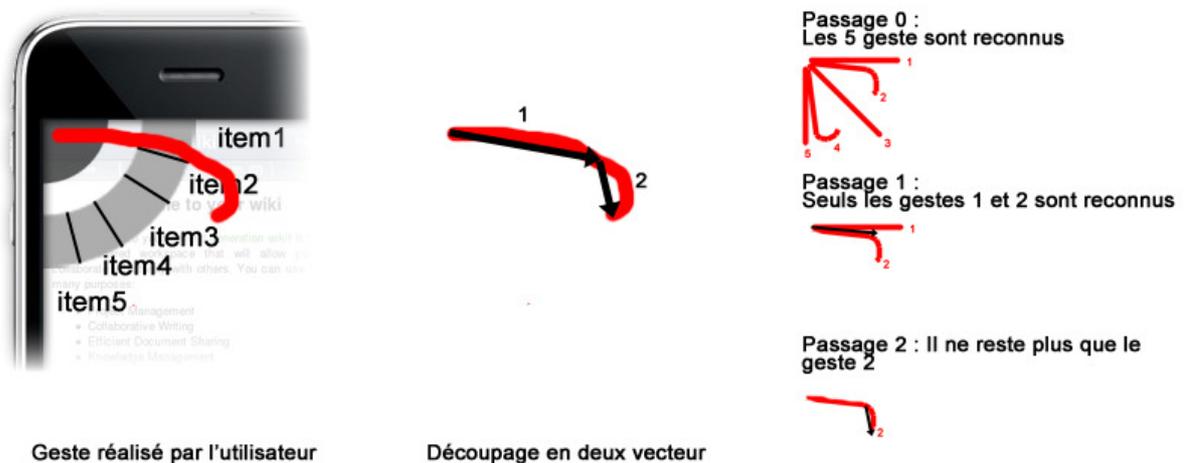


FIGURE 34 : ANALYSE DE GESTES

Code de l'algorithme en Annexe 3.

VII. Gestion de Projet

Ce stage est un peu particulier pour moi puisque venant d'une formation professionnelle j'ai réalisé un stage dans la recherche. J'ai donc découvert les méthodes de travail inhérentes à cet environnement. J'ai en effet commencé ce stage par une étude bibliographique permettant de situer le contexte de notre projet, d'un point de vue technique et

technologique. J'ai ensuite après deux mois de stage réalisé un premier planning prévisionnel (figure 35) afin d'organiser le déroulement du projet. Cependant plusieurs évènements comme la sortie du firmware 2.0 de l'iPhone ou l'avancement de la date de rendu du rapport XWiki concerto m'ont obligé à recentrer mes priorités et à adapter mon planning (figure 36). J'ai donc dû repenser le développement de l'application finale pour qu'elle rentre dans le cadre de l'iPhone 2.0. Et j'ai du mettre au deuxième plan l'implémentation de l'application pour rédiger le rapport XWiki concerto.

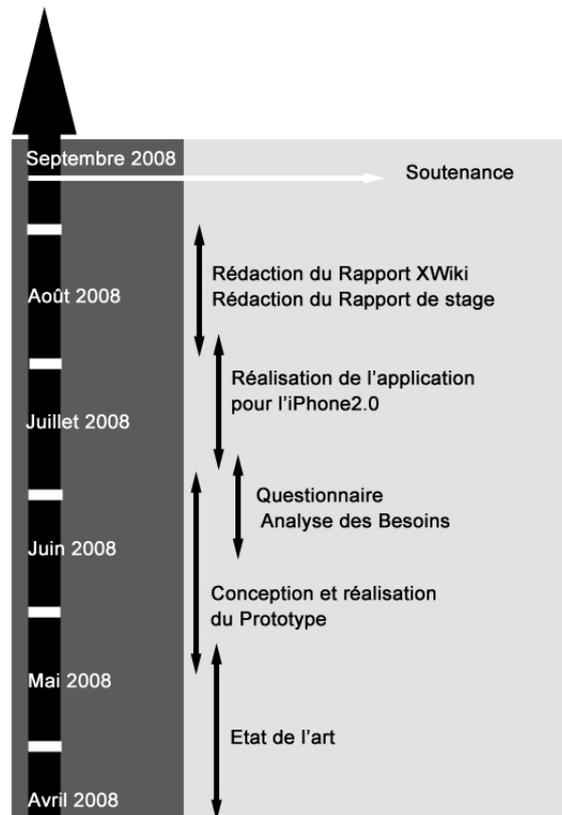


FIGURE 35 : PLAN DE TRAVAIL PREVISIONNEL

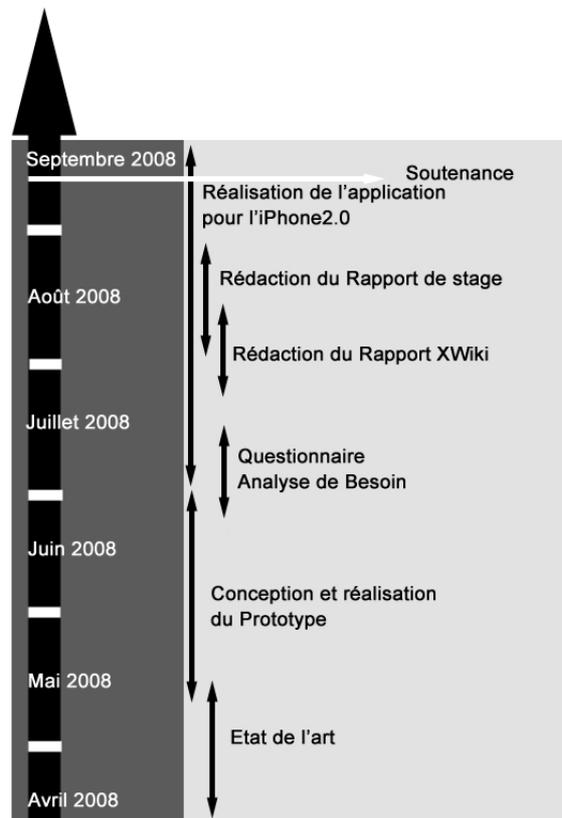


FIGURE 36 : PLAN DE TRAVAIL REVISE

VIII. Conclusion

1. Bilan technique

Afin de répondre aux attentes du projet XWiki Concerto et notamment aux contraintes liées au thème de la mobilité nous avons donc conçu et développé une application Web riche permettant de profiter pleinement des interactions directes avec les pages Web de XWiki sur iPhone. De plus l'utilisation de JavaScript laisse ouverte l'adaptation de manière aisée sur d'autres plateformes mobiles pouvant lire des pages Web. Enfin notre boîte à outils laisse la possibilité de continuer le développement de nouvelles fonctionnalités qui pourraient apparaître dans l'avenir parallèlement à l'évolution de Xwiki. La figure 37 résume la démarche de travail.

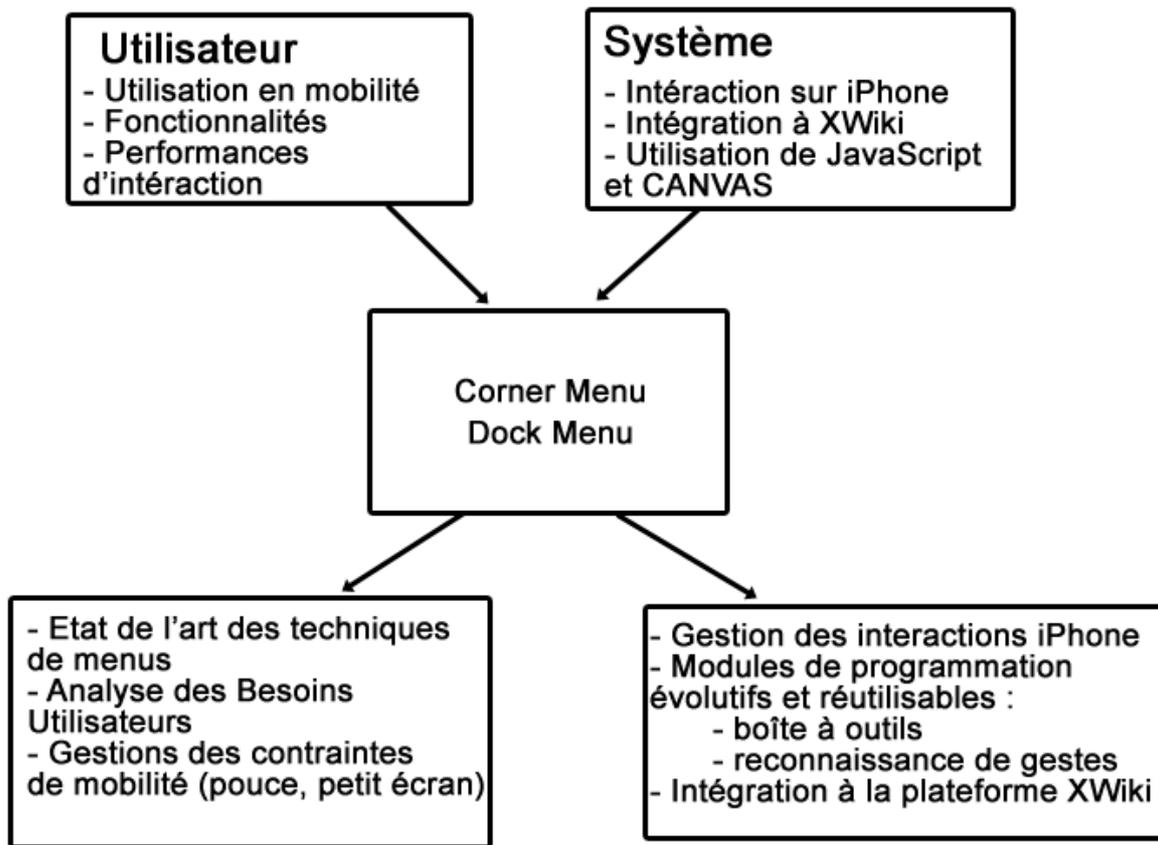


FIGURE 37 : DEMARCHE DE TRAVAIL

Nous prévoyons de réaliser des tests et des évaluations afin de valider notre application.

2. Bilan personnel

Ce stage m'a permis de mettre à profit les compétences acquises au cours de ma formation à l'Université Pierre et Marie Curie, notamment en ce qui concerne les enseignements propres au parcours ICHM au niveau interaction et réalisation d'interface web, ainsi qu'à certaines compétences plus générales du master IAD comme la reconnaissance de geste.

J'ai pu perfectionner mes connaissances des langages de programmations, en particulier JavaScript et des nombreuses possibilités qu'il offre. J'ai pu aussi découvrir et apprendre le langage Objective C, grâce à mes connaissances en C++ et C, acquises au cours de mes études.

J'ai pu apprendre à travailler en collaboration étroite avec d'autres personnes, ceci me permettant de développer ma communication et mon expérience du travail en équipe tout

en laissant place à une grande autonomie qui m'a permis d'acquérir des compétences d'organisation et de gestion de mon temps de travail.

J'ai eu la chance de pouvoir travailler au sein d'une équipe sympathique et dynamique, faisant de ce stage une expérience enrichissante humainement, où j'ai pu rencontrer des gens disponibles, compétents et à l'écoute.

Enfin ce travail, complétant pleinement ma formation, m'a aussi permis de me familiariser avec le monde de la recherche et de confirmer mon désir de compléter cette formation par un Doctorat en IHM encadré par Eric Lecolinet.

Bibliographie

Apple Inc. 2008. *iPhone Human Interface Guidelines for Web Applications*. 2008.

Apple Inc. 2008. Apple - iPhone. *site Web Apple*. [En ligne] Juillet 2008. <http://www.apple.com/iphone/>.

Bailly, Gilles, et al. 2008. Menus Leaf : Enrichir les menus linéaires par des gestes. *IHM*. 2008.

Bailly, Gilles, Lecolinet, Eric and Niguay, Laurence. 2008. Flower Menus: A New Type of Marking Menus with Large Menu Breadth, Within Groups and Efficient Expert Mode Memorization. *Advanced Visual Interfaces*. 2008.

Huot, Stéphane et Lecolinet, Eric. 2007. ArchMenu et ThumbMenu: Contrôler son dispositif mobile « sur le pouce ». *IHM*. 2007.

Karlson, Amy K., Bederson, Benjamin B. et Contreras-Vidal, Jose L. 2006. Understanding Single-Handed Mobile Device Interaction. 2006.

Lecolinet, Eric et Stuart, Pook. 2002. Interfaces zoomables et "Control Menu". *LCN*. 2002, Vol. 3, 2.

mad4milk. 2008. Mootools. *site Mootools*. [En ligne] 2008. <http://mootools.net/>.

Parhi, Pekka, Karlson, Amy K. et Bederson, Benjamin B. 2006. Target Size Study for One-Handed Thumb Use on Small Touchscreen Devices. *MobileHCI'06*. 2006, 203-210.

The Dojo Foundation. 2008. DojoToolkit. [En ligne] 2008. <http://dojotoolkit.org/>.

Vogel, Daniel et Baudish, Patrick. 2007. Shift : A technique for Operating Pen-Based Interfaces Using Touch. *CHI'07*. 2007, 657-666.

Xwiki. 2007-2008. *XWiki concerto*. [En ligne] 2007-2008. <http://concerto.xwiki.com>.

IX. Annexe 1

Questionnaire Xwiki.

Afin d'améliorer les services XWiki sur appareils mobiles (iphone,smartphone) , nous vous demandons de bien vouloir répondre à un petit questionnaire. Nous pourrions ainsi aller vers une solution plus proche de l'utilisateur et des problèmes liés à la mobilité.

Domaine de travail

- Informatique
- Multimédia
- Social
- Santé
- Public
- Administration
- Education
- Recherche
- Autre :

Age

- inférieur à 25
- entre 25 et 50
- supérieur à 50

Combien de temps passez-vous en moyenne devant un ordinateur par jour?

- moins d'une heure
- entre 1 et 5 heures
- plus de 5 heures

Combien d'appareils mobiles avez-vous ? (iphone, pda, smartphone, téléphone standard...)

- 0
- 1
- 2
- 3
- plus de 3

Quels types de mobiles de la liste suivante possédez-vous ?

- iPhone
- Blackberry
- autres SmartPhones/PDAphones
- téléphones WAP
- autres téléphones
- Autre :

Votre mobile possède-t-il :

- un clavier physique complet? (une touche par lettre)
- un clavier physique réduit? (plusieurs lettres par touche)
- un clavier logique ? (affiché sur l'écran)
- un écran tactile ?
- un stylet ?

Qu'utilisez-vous pour interagir avec votre mobile ?

- le clavier
- le pouce
- l'index
- le stylet
- une seule main
- les deux mains
- commande vocale

Si vous avez l'habitude d'utiliser vos deux mains pour interagir aimeriez-vous pouvoir interagir d'une seule main ?

- oui
- non

Utilisez-vous des services Internet sur mobile ?

- oui
- non

Si oui lesquels ?

- Navigation Web
- Messagerie (e-mail)
- VOD (vidéo à la demande)
- Téléchargement de musique
- Autre :

Utilisez-vous ces services au moins ...

- une fois par jour ?
- une fois par semaine ?
- une fois par mois ?
- Autre :

Utilisez-vous des wikis sur ordinateur ?

- oui
- non

Si oui lesquels ?

- XWiki
- blueKiwi
- Dotclear
- TikiWiki
- TWiki
- MediaWiki
- Autre :

Utilisez-vous des wikis sur mobile ?

- oui
- non

Quels sont dans les fonctionnalités suivantes celles que les wikis sur mobile devraient conserver ? Consultation des pages du wiki? Notez-les. 1 étant la moins bonne note et 5 la meilleure

- 1
- 2
- 3
- 4
- 5

Edition des pages?

- 1
- 2
- 3
- 4
- 5

Commentaires ?

- 1
- 2
- 3
- 4
- 5

Transfert et partage de fichier sur le wiki?

- 1
- 2
- 3
- 4
- 5

Administration du wiki ?

- 1
- 2
- 3
- 4
- 5

Quels fonctionnalités avancées pensez-vous utile de conserver sur le wiki pour mobile ?

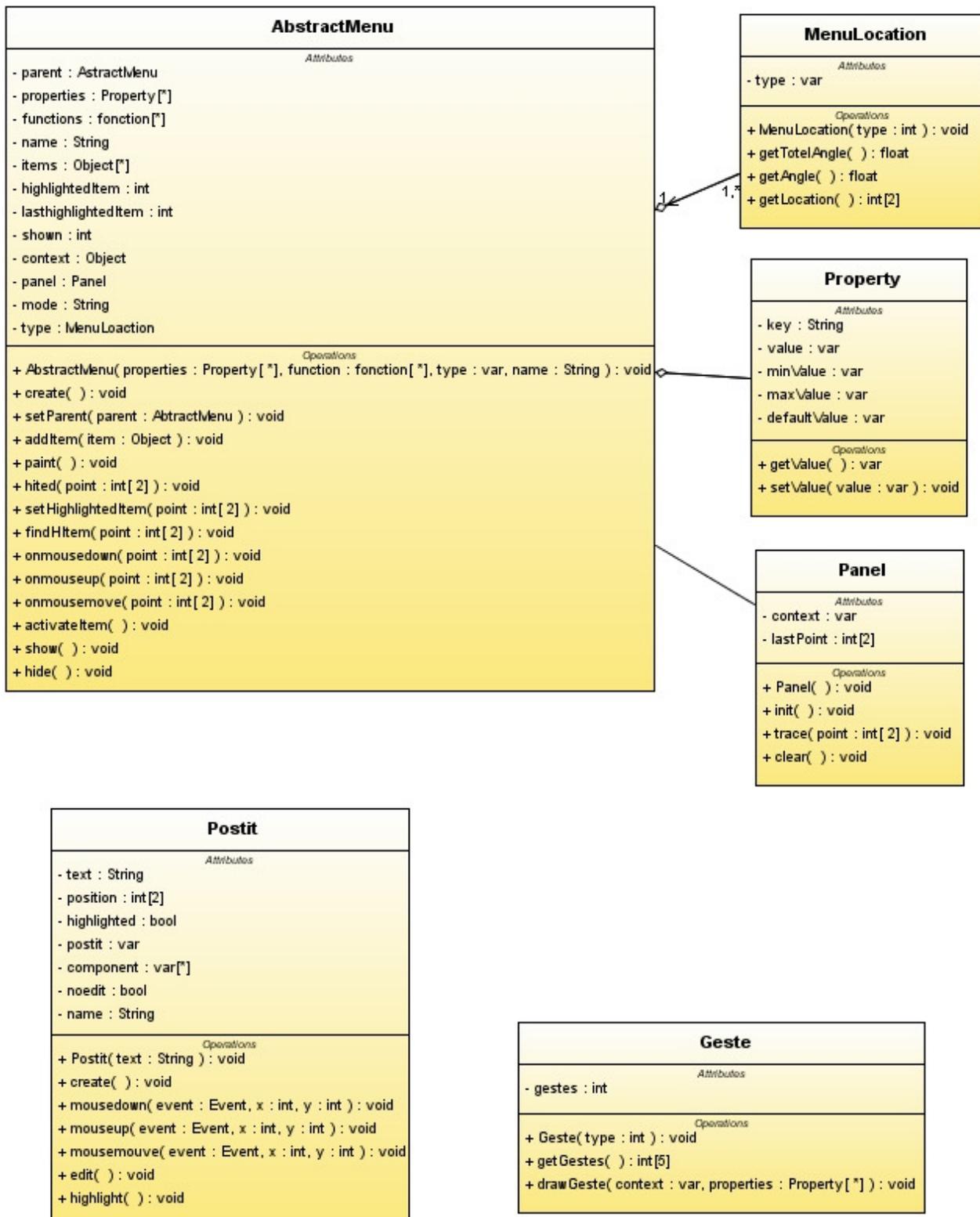
- Edition de Classe
- Edition d'Objets
- Edition de droits
- Aucune

Quels sont les fonctionnalités qui vous semblent les plus importantes sur wikis sur mobile ?

Avez-vous des suggestions pour les wikis sur mobile ?

X. Annexe 2

Diagramme UML de l'application.



XI. Annexe 3

```
function distance (x, y, x1, y1){
    var hypo = Math.sqrt(Math.pow(x-x1,2)+Math.pow(y-y1,2));
    return hypo;
}
```

```
function angleComputeO(x, y, x1, y1, orientation){
    var hypo = distance (x, y, x1, y1);
    var angle=Math.acos((x-x1)/hypo);
    var signe ;
    if (y >= y1 ) {
        signe = -1;
    }else{
        signe = 1;
    }
    return (angle-orientation)*signe;
}
```

```
function calculAngle(points){
    var listAngle = new Array();
    var l = points.length;
    var ld = Math.floor(l/3);
    listAngle.push(
        angleComputeO(points[ld][0],
            points[ld][1],
            points[0][0],
            points[0][1],
            Math.PI)
    );
    listAngle.push(angleComputeO(
        points[l-1][0],
        points[l-1][1],
        points[ld*2][0],
        points[ld*2][1],
        Math.PI));
    return listAngle;
}
```

```
function gestureAnalyse(listAngle,geste,angle,flag){
    //on élimine les gestes selon le premier angle
    if (flag) {
        geste[0] = listAngle[0] >= Math.PI / 8 ? 0 : geste[0];
        geste[1] = listAngle[0] >= Math.PI / 8 ? 0 : geste[1];
        geste[2] = listAngle[0] < Math.PI / 8 || listAngle[0] > 3*Math.PI / 8 ? 0 : geste[2];
        geste[3] = listAngle[0] <= 3* Math.PI / 8 ? 0 : geste[3];
        geste[4] = listAngle[0] <= 3 * Math.PI / 8 ? 0 : geste[4];
    }else{
        geste[0] = listAngle[angle] > Math.PI / 8 ? 0 : geste[0];
        geste[1] = listAngle[angle] <= Math.PI / 8 ? 0 : geste[1];
        geste[3] = listAngle[angle] >= 3*Math.PI/8 ? 0 : geste[3];
        geste[4] = listAngle[angle] < 3*Math.PI / 8 ? 0 : geste[4];
    }

    var count = 0;
    for(var v=0; v<geste.length; v++){
        if (geste[v] == 1) {
            count++;
        }
    }

    //s'il ne reste plus qu'un geste on le renvoie
    if (count < 2) {
        return geste;
    }
    else {
        //sinon on regarde le dernier angle.
        //en effet dans le pire cas on trouve trois gestes encore présents
        // mais le dernier angle permet de les différencier

        return gestureAnalyse(listAngle, geste, 1, false);
    }
}
```