



ADAPTATION SUR TABLETTE TACTILE D'UN LOGICIEL ÉDUCATIF POUR ENFANTS EN ÂGE PRÉSCOLAIRE

**MÉMOIRE REALISÉ EN VUE DE L'OBTENTION DE LA MAÎTRISE UNIVERSITAIRE EN
SCIENCES ET TECHNOLOGIES DE L'APPRENTISSAGE ET DE LA FORMATION**

PAR

Jasmina Travnjak

DIRECTRICE DE MÉMOIRE

Mireille Bétrancourt, FPSE-TECFA, Université de Genève

JURY

| | | |
|----------------------|--------------|----------------------------------|
| Christelle Bozelle | examinatrice | FPSE-TECFA, Université de Genève |
| Cyril Roiron | examineur | DIP, Etat de Genève |
| Nicolas Szilas | examineur | FPSE-TECFA, Université de Genève |
| Mireille Bétrancourt | directrice | FPSE-TECFA, Université de Genève |

Genève, juin 2014

RESUME

Cette recherche porte sur l'adaptation d'un logiciel éducatif destiné aux enfants non lecteurs vers une utilisation mobile et tactile. Le logiciel existant étant prévu pour une utilisation sur ordinateur fixe, nous avons pour objectifs de l'adapter à une nouvelle utilisation sur tablette tactile. Dans le cadre de ce mémoire, nous avons dans un premier temps identifié les contraintes à prendre en compte lors d'un développement de logiciel pour les plateformes mobiles, puis nous avons spécifié les différences d'interaction à intégrer afin d'adapter le logiciel à son nouvel dispositif : la tablette tactile. Le logiciel a ensuite été développé en tenant compte des recommandations retenues et a été testé auprès de 5 enfants afin de valider sa facilité d'utilisation. Les tests ont permis de constater que l'utilisation du logiciel était relativement intuitive, sauf lorsqu'il s'agit d'exécuter des actions de type drag'n'drop qui s'avèrent difficiles pour de si jeunes enfants.

Remerciements

Ces 3 années passées à travailler pour le Master MALTT, ce n'était pas tous les jours facile. Travailler à 100 % et étudier en parallèle, c'est un défi que je suis fière d'avoir relevé, mais je n'aurai pas réussi sans être si bien entourée !

Je tiens donc ici à remercier mes parents Mina et Mustafa et mes deux sœurs Mirela et Alma qui m'ont apporté leur soutien tout au long de ces 3 années.

Je tiens également à remercier mon copain ainsi que mes nombreux amis et amies pour leur compréhension pour mes absences à répétition à toute sorte d'évènements, mais surtout pour leurs encouragements et pensées positives quand j'en avais besoin.

Merci également à mes anciens et actuels collègues qui ont été à mon écoute les lundis matins alors que je revenais au travail après un weekend studieux et qui me servaient souvent de cobaye pour les tests, et de conseillers pour certains projets.

Mon spectre de connaissances s'est enrichi de manière indescriptible durant ces 3 ans. J'ai non seulement appris plein de choses nouvelles mais j'en ai aussi appris long sur moi-même. J'ai également rencontré des gens incroyables qui m'ont accompagné tout au long de cette aventure et avec qui je compte bien rester en contact même après le MALTT. Je tiens donc à remercier tous les gens qui ont fait du MALTT ce qu'il est : merci à ma directrice de mémoire, aux enseignants, aux assistants, aux collaborateurs et de manière générale à toute l'équipe TECFA ! Et un grand merci à tous mes camarades du MALTT, les « Maltesers » !



Table des matières

| | | |
|-------|--|----|
| 1 | Introduction..... | 7 |
| 1.1 | Education et supports technologiques..... | 7 |
| 1.2 | Contexte | 8 |
| 1.3 | Vers une nouvelle utilisation | 9 |
| 1.4 | Problématique..... | 10 |
| 1.4.1 | Utilisation actuelle du logiciel | 10 |
| 1.4.2 | Passage vers une utilisation mobile et une interaction tactile | 10 |
| 2 | Cadre théorique : Outils mobiles et tactiles en éducation..... | 12 |
| 2.1 | Le support mobile en tant qu'outil éducatif | 12 |
| 2.1.1 | Définition et usages..... | 12 |
| 2.1.2 | Avantages | 13 |
| 2.1.3 | Inconvénients | 14 |
| 2.1.4 | Le mobile et les enfants..... | 15 |
| 2.1.5 | Conclusion | 15 |
| 2.2 | La tablette tactile en tant qu'outil éducatif | 16 |
| 2.2.1 | Définition..... | 16 |
| 2.2.2 | Avantages | 17 |
| 2.2.3 | Inconvénients | 18 |
| 2.2.4 | Conclusion | 19 |
| 2.3 | IHM tactiles | 19 |
| 2.3.1 | Conception d'IHM tactiles | 19 |
| 2.3.2 | Conception d'IHM tactiles destinées aux enfants..... | 24 |
| 2.4 | Conclusion du cadre théorique | 25 |
| 3 | Conception | 28 |
| 3.1 | Présentation du logiciel existant | 28 |
| 3.2 | Recueil des fonctionnalités | 30 |
| 3.2.1 | Type d'exercices 1 : Guidé par la voix | 31 |
| 3.2.2 | Type d'exercices 2 : Libre, avec voix sur le passage de la souris..... | 32 |
| 3.3 | Objectifs retenus pour la refonte du logiciel..... | 33 |
| 3.3.1 | Aspect Multi-plateformes du logiciel | 33 |
| 3.3.2 | Aspect interaction tactile | 36 |
| 3.3.3 | Aspects de suivi de progrès | 36 |
| 3.3.4 | Aspect diffusion du contenu sur support mobile..... | 36 |

| | | |
|-------|--|----|
| 3.4 | Définition des priorités de développement | 37 |
| 3.5 | Développement | 38 |
| 3.5.1 | Environnement de développement | 38 |
| 3.5.2 | POC (Proof of Concept) | 39 |
| 3.6 | Fonctionnalités développées..... | 42 |
| 3.6.1 | Phase 1 : Adaptation du POC au monde mobile et tactile | 42 |
| 3.6.2 | Phase 2 : Réalisation du passage d'un niveau à un autre | 44 |
| 3.6.3 | Phase 3 : Réalisation d'un exercice de catégorie libre | 45 |
| 3.6.4 | Phase 4 : Réalisation d'un niveau de type guidé avec éléments sur la scène | 45 |
| 3.6.5 | Phase 5 : Amélioration de la diffusion du contenu | 46 |
| 3.6.6 | Phase 6 : Nouveau type d'exercice..... | 48 |
| 3.6.7 | Stockage local | 48 |
| 3.6.8 | Structure du code (cf. Annexe 2 : code source du logiciel) | 49 |
| 3.6.9 | Fin du développement | 50 |
| 4 | Tests d'utilisabilité..... | 53 |
| 4.1 | Objectifs..... | 53 |
| 4.2 | Public cible..... | 53 |
| 4.3 | Méthode | 53 |
| 4.4 | Résultats | 54 |
| 4.4.1 | Observations générales | 54 |
| 4.4.2 | Résultats détaillés..... | 55 |
| 4.5 | Discussion des résultats | 60 |
| 5 | Conclusion | 61 |
| 6 | Bibliographie et Webographie | 63 |
| 6.1 | Articles..... | 63 |
| 6.2 | Guides de développement pour les plateformes tactiles | 66 |
| 7 | Annexes | 67 |
| 7.1 | Annexe 1 : accès au logiciel en ligne | 67 |
| 7.2 | Annexe 2 : code source du logiciel | 67 |
| 7.3 | Annexe 3 : suivi des versions du logiciel sur GitHub | 67 |
| 7.4 | Annexe 3: Recueil de fonctionnalités sous Excel..... | 67 |
| 7.5 | Annexe 4 Ajout de données pour un nouvel exercice | 67 |

Table des figures

| | |
|--|----|
| Figure 1 : Exemple d'exercices du domaine "Animaux" | 8 |
| Figure 2 : Exemple de supports mobiles (Jaynes, 2013) | 12 |
| Figure 3 : Tablette tactile (source http://www.silicon.fr/)..... | 16 |
| Figure 4 : Gestes basiques selon Lao et al. (2009) | 20 |
| Figure 5 : Exemples d'utilisation d'une tablette (Wroblewski L., 2012)..... | 22 |
| Figure 6 : Zones atteignables sur une tablette (Wroblewski L., 2012)..... | 22 |
| Figure 7 : Exemple d'utilisation de tablette posée sur le bureau (Boujol, 2014)..... | 23 |
| Figure 8 : Zones atteignables sur un écran d'ordinateur (Nogier, 2003) | 23 |
| Figure 9 : Disposition des informations sur un ordinateur (Nogier, 2003) | 24 |
| Figure 10 : Ecran de navigation | 28 |
| Figure 11 : Navigation 2ème niveau..... | 28 |
| Figure 12 : Exemple Animaux, exercice 1 | 29 |
| Figure 13 : Exemple d'exercice de catégorie 1.3 | 31 |
| Figure 14 : Exemple d'exercice de catégorie 1.4 | 31 |
| Figure 15 : Exemple d'exercice de catégorie 1.5 | 32 |
| Figure 16 : Exemple d'exercice de catégorie 2.1 | 33 |
| Figure 17 : Support de l'élément Canvas..... | 34 |
| Figure 18 : Support de l'Audio grâce à l'API Web Audio en JavaScript | 35 |
| Figure 19 : Formats de son supportés | 35 |
| Figure 20 : Support du Web Storage API par les navigateurs | 37 |
| Figure 21 : Résultats du 1er POC | 40 |
| Figure 22 : Résultats du 2ème POC | 41 |
| Figure 23 : Résultats du 3ème POC | 42 |
| Figure 24 : Comparaison Animaux, exercice 1 | 42 |
| Figure 25 : Ecran de fin de niveau | 44 |
| Figure 26 : Exemple d'exercice de type libre..... | 45 |
| Figure 27 : Exemple d'exercices du domaine Objets..... | 46 |
| Figure 28 : Ecran de chargement..... | 46 |
| Figure 29 : Diagramme de chargement des médias..... | 47 |
| Figure 30 : Exemple d'un exercice des Adjectifs | 48 |
| Figure 31 : Structure du code | 49 |
| Figure 32 : Icône de lancement sur iPad | 50 |
| Figure 33 : Jeu en plein écran sur iPad | 51 |
| Figure 34 : Suivi de progrès | 51 |
| Figure 36 : Usability tests, Thomas, 2 ans | 54 |
| Figure 35 : Usability tests, Gaïa, 5 ans..... | 55 |
| Figure 37 : Menu de fin de niveau..... | 56 |
| Figure 38 : Icône de haut-parleur..... | 57 |
| Figure 39 : Exemple d'objets trop petits pour les enfants | 57 |
| Figure 40 : Lévana, 3 ans et demi devant un exercice du domaine Adjectif..... | 59 |

1 Introduction

1.1 Education et supports technologiques

Le monde de l'éducation est en évolution constante. Ces dernières décennies, le tableau noir a été remplacé par le tableau blanc puis dans certains établissements même par un tableau blanc interactif. Les salles de classe disposent de plus en plus d'au moins un ordinateur, à disposition pour un support à l'enseignement, avec des exercices plus ou moins ludiques, et des contenus pédagogiques plus ou moins adaptés.

L'utilisation de la technologie en tant que support éducatif est de plus en plus répandue et les enseignants tendent à faire la paix avec l'usage des technologies dans leurs classes. Mais, la technologie, à peine installée dans les écoles, a continué d'évoluer. Ces dernières années, les tablettes tactiles et autres smartphones commencent à remplacer dictionnaires, cahiers d'écriture, calculatrices, etc.

La technologie est en constante évolution, et son utilisation dans le monde éducatif aussi.

Avec le temps, ce qu'on peut remarquer c'est que la technologie est devenue de plus en plus mobile. En effet, il y a une dizaine d'années, lorsqu'on parlait d'avoir un ordinateur en classe, il s'agissait d'un ordinateur fixe, c'est-à-dire d'une tour et d'un grand écran, le tout relié à une prise électrique et éventuellement à une prise réseau. Les salles de classe disposaient d'un coin réservé à l'ordinateur et il fallait que l'élève se déplace dans ce coin spécifique s'il voulait avoir accès aux ressources informatisées. Ceci impliquait une certaine logistique autant au niveau de la place dans les salles de classe que dans la gestion de l'accès à La Machine.

Ces ordinateurs fixes ont fini par perdre en taille et en encombrement. Ils sont devenus de plus en plus petits, transportables puis portables avec l'évolution technologique. Cette évolution constante, rend les supports technologiques de plus en plus mobiles, que ce soit par la transportabilité du support que par les réseaux sans fil qui permettent des connexions internet sans prise à proximité.

L'ordinateur portable n'en est qu'un exemple. Parmi les supports mobiles on cite aussi les smartphones, puis les tablettes tactiles qui se sont fait une place auprès du grand public au milieu des années 2000 (Buxton, 2010).

Ces petits gadgets prennent de plus en plus de place dans notre vie. Ils sont bien plus transportables qu'un ordinateur portable. Ils sont aussi plus rapides et réactifs (temps d'allumage) et plus ludiques car ils mettent à disposition des applications de tout genre que l'on ne trouve pas toujours sur leurs grands frères, ordinateurs portables.

De ce fait, ces petits objets permettent des interactions plus spontanées, plus mobiles, et plus personnalisées. Rien d'étonnant donc à ce que leur utilisation dans le monde de l'éducation soit en pleine expansion.

Le fait d'intégrer ces nouveaux supports technologiques en tant qu'outil éducatif, nécessite de repenser les logiciels existants prévus pour un usage sur ordinateur fixe. En effet, les logiciels existants peuvent être adaptés à un usage plus mobile et à une interaction tactile sans pour autant remettre en question leur contenu pédagogique. Mais, il faut pour cela adapter le contenu existant pour sa

diffusion sur le nouvel outil. Bien entendu, cela nécessite un travail de réflexion et de refonte, travail qui sera justement étudié dans le présent document.

Ce mémoire a pour objectif d'étudier et de réaliser le passage d'un logiciel éducatif existant, prévu initialement pour un usage sur ordinateur fixe, vers un support mobile à interface tactile.

Pour ce changement, il n'est à priori pas nécessaire de repenser le contenu pédagogique du logiciel existant. Il faut cependant étudier la mise à disposition du contenu existant sur son nouveau support mobile, ainsi que les changements d'interaction dus à la mise à disposition du futur logiciel sur un support tactile où les doigts remplacent la souris.

1.2 Contexte

L'unité TECFA (Technologies de formation et d'apprentissage) de l'Université de Genève, en collaboration avec le Centre Romand d'implants cochléaires (HUG), a développé en 2007 un logiciel de support à l'apprentissage du français, dédié à aider les enfants sourds avec implant cochléaire. Ce développement faisait partie d'une recherche financée par la Fondation Hans Wilsdorf.

Le logiciel existant porte le nom « Les Aventures de Lili et Théo ». Il s'agit d'une suite d'exercices proposés sous forme de quizz imagés dans lesquels Lili et Théo sont mis en scène dans différents contextes. Les exercices sont initiés à l'aide des instructions d'une voix off qui raconte les aventures de Lili et Théo. Pour chaque domaine d'apprentissage, les deux aventuriers se trouvent dans une situation donnée, et l'enfant doit les aider à avancer dans leurs aventures en répondant correctement aux questions de la voix off.



Figure 1 : Exemple d'exercices du domaine "Animaux"

Par exemple, dans le domaine « Animaux » (Figure 1), la voix off va dire à l'enfant : « *Lili et Théo sont dans la forêt (à gauche ci-haut) ou à la montagne (à droite ci-haut) et rencontrent plein d'animaux, aide-les à les découvrir* ». Elle demande ensuite à l'enfant de choisir le bon animal parmi une liste d'images à disposition. La voix va dire par exemple « le chat » et l'enfant devra cliquer sur le chat, et ainsi de suite jusqu'à ce que tous les animaux soient choisis. Au fur et à mesure que l'enfant choisit un animal, un feedback va l'informer de son succès ou de son erreur et l'encourager à réessayer le cas échéant. L'animal choisi va apparaître dans le décor, dans le cas des animaux il s'agit d'une forêt. Une fois le niveau terminé, l'enfant est félicité et peut passer au quizz suivant ou changer de domaine en revenant dans le menu principal.

Le logiciel comporte plus de 100 exercices divisés en 12 domaines (animaux, habits, objets, parties du corps, etc.). Il permet de faire connaître de nouveaux mots de manière ludique, interactive et adaptée aux enfants non lecteurs en réhabilitation auditive.

Le logiciel avait été initialement créé pour une utilisation sur PC, c'est-à-dire un ordinateur avec un système d'exploitation Windows. Une première version du logiciel avec seulement quelques exercices a été développée à l'aide de l'outil Authorware dans le cadre de la réalisation d'un mémoire de Master (Bozelle, 2006). Par la suite, le logiciel a été complété dans le cadre d'un travail de doctorat qui est toujours en cours. Ce choix technique était tout à fait en accord avec les supports technologiques de l'époque et était alors parfaitement adapté à la demande.

Dans le cadre de son mémoire de Master, Bozelle (2006), a testé la première version du logiciel auprès du public cible et a eu des résultats qui montraient une amélioration des performances concernant les sons exercés, ce d'autant plus dans le cas d'une utilisation régulière du logiciel : *« En considérant les résultats obtenus au cours des comparaisons de moyennes [...], nous pouvons conclure [...] que le logiciel est utile et améliore les performances des utilisateurs »*. (Bozelle, 2006, p.65) En ce qui concerne l'évaluation de la version complète du logiciel, développée dans le cadre du doctorat de Bozelle, elle est actuellement en cours de finalisation.

Etant donné ces résultats concluants, l'unité TECFA souhaite aujourd'hui étendre son utilisation auprès d'enfants non-francophones âgés de 2 à 5 ans. En effet, si le logiciel existant aide les enfants sourds et non-lecteurs à appréhender les sons du langage parlé, il pourrait servir pour des enfants non-francophones de moins de 6 ans.

Son nouvel objectif sera donc de faciliter l'intégration d'enfants non-francophones et de leur donner des bases pour la compréhension de la langue française, base qui leur servira à l'apprentissage de la lecture et l'écriture. Le nouveau logiciel aura pour nom « Le Moulin à paroles ».

1.3 Vers une nouvelle utilisation

Le futur logiciel est destiné en priorité à un usage dans les écoles genevoises. En effet, on compte plus de 100 enfants de langue étrangère qui débutent leur scolarité à Genève chaque année (OFS, 2012), et qui se retrouvent immergés dans leur nouvelle langue, le français, dans les structures préscolaires, d'où le besoin d'un support à l'apprentissage de la langue française dans les écoles.

Au vu de l'évolution des supports technologiques et de leur omniprésence dans nos quotidiens, le nouveau logiciel devra être adapté à des supports mobiles ainsi qu'à une utilisation au travers d'interfaces tactiles. On passera donc d'un logiciel prévu pour une utilisation plutôt fixe dans une salle de classe, à un logiciel mobile de par son accessibilité par le Web et son adaptabilité sur différentes plateformes tactiles ou non.

Afin de réussir la refonte du logiciel existant, le futur « Moulin à Paroles », sera développé à l'aide des nouvelles fonctionnalités du langage HTML5. Ceci permettra de le rendre accessible depuis n'importe quel support ayant accès à Internet.

Le fait de développer le futur logiciel en utilisant les fonctionnalités de HTML5, le rendra plus mobile. De ce fait son utilisation, bien qu'en premier lieu prévue pour les écoles, pourra s'étendre à une utilisation à la maison ou en déplacement, grâce à un support connecté à Internet.

Une fois développé, le logiciel pourra être utilisé de manière autonome par l'enfant aidé de ses proches ou en jardin d'enfants, crèche ou école afin de maximiser l'apport d'un tel logiciel pour l'apprentissage du français à tout moment.

1.4 Problématique

L'évolution des supports technologiques peut vite rendre obsolètes les outils à disposition sur ces supports d'une version à l'autre. Lorsqu'il s'agit d'un contenu à but pédagogique, le contenu en soit n'est pas obsolète. Si l'on prend l'exemple d'un logiciel d'apprentissage de mathématiques ou de langues, en général son contenu n'évolue pas aussi vite que le support sur lequel il est diffusé. Il serait dommage de devoir toujours recommencer la conception pédagogique de ces outils à chaque changement de support, alors que l'on pourrait garder le contenu et l'adapter à son utilisation sur les nouveaux supports.

Un logiciel, qui était initialement prévu pour une utilisation sur un ordinateur Windows, peut donc être repensé de manière à être adapté vers une utilisation plus mobile et plus hétérogène, et adapté à l'évolution technologique. Pour faire cela, il faut toutefois réfléchir à la problématique du changement du support et du changement de l'interaction étant donné qu'on tend vers des supports mobiles à interaction tactile.

Le cas du logiciel de support à l'apprentissage de la langue française « Moulin à Paroles » est l'exemple sur lequel se basera cette recherche pour la refonte vers une utilisation mobile et adaptée à l'interaction tactile. Le contenu pédagogique du logiciel étant parfaitement adapté aux enfants de 2 à 5 ans que ce soit par les niveaux, les types ou les graphismes des exercices, il mérite d'être gardé et adapté à une utilisation sur de nouveaux supports.

1.4.1 Utilisation actuelle du logiciel

A l'état actuel, le logiciel est prévu pour une utilisation sur un ordinateur de type Windows. Ce choix était tout à fait pertinent en 2006 mais avec l'évolution constante des technologies, le logiciel est devenu contraignant dans son utilisation.

En effet, l'apprenant est limité dans l'utilisation du logiciel et par le lieu et par le support. Il ne peut l'utiliser que sur un support de type ordinateur, qui plus est, il faut que cet ordinateur puisse exécuter ce type de logiciels, destiné uniquement à des ordinateurs Windows.

Dans le cadre de l'utilisation dans les écoles, ceci implique d'avoir au moins un ordinateur dans la salle, avec tout l'aménagement de l'espace et de l'infrastructure que cela nécessite.

De plus, le logiciel ayant été conçu pour une utilisation avec une souris, les interactions ont été imaginées en fonction de cela, ce qui fait que, tel quel, le logiciel n'est pas adapté à un support tactile.

1.4.2 Passage vers une utilisation mobile et une interaction tactile

Le futur logiciel devra être repensé afin de rendre son utilisation plus mobile et plus flexible. De plus, il faudra adapter les interactions au tactile.

Afin de réaliser ce logiciel, il est nécessaire, dans un premier temps, de s'intéresser à l'apprentissage mobile, que ce soit par l'intermédiaire d'un support mobile ou par le fait que le contenu ou l'apprenant soit eux-mêmes mobiles.

On pourrait penser qu'un contenu déjà existant sur ordinateur peut simplement être retranscrit tel quel sur un support mobile, mais ce n'est pas suffisant. En effet, pour pouvoir délivrer un contenu sur un support mobile, il faut aussi tenir compte des capacités de celui-ci. On peut imaginer des problématiques telles que la bande passante limitée ou l'absence totale de connexion, sans parler de la multiplicité des supports mobiles ayant tous des caractéristiques différentes autant par leur navigateur que la taille de leurs écrans, etc. L'apprentissage mobile n'est donc pas une simple conversion du contenu existant, c'est une nouvelle façon de diffuser le contenu afin de cibler spécifiquement le support mobile.

Il faudra donc définir le support mobile, ses avantages et contraintes, afin de produire une base théorique pour la refonte du futur logiciel.

Dans un deuxième temps, il sera nécessaire de se focaliser sur les changements d'interaction qu'implique l'utilisation de ce nouveau logiciel sur les supports tactiles. En effet, il ne s'agira pas de simplement transposer l'interaction pensée sur ordinateur vers une interaction tactile car cette dernière obéit à une autre logique. Selon Bétrancourt (2012), l'interface tactile réintroduit la coordination œil-main. En effet, tout comme lorsqu'on dessine, ou l'on écrit, l'œil suit et contrôle ce que fait la main. Alors que dans une interaction avec un logiciel sur ordinateur, la main est sur la souris et les yeux sont sur l'écran, sur un support tactile, l'interaction est directe comme celle imaginée par Norman (Hutchins, Hollan, & Norman, 1985).

Afin de comprendre les impacts du passage vers un support mobile et l'adaptation au monde tactile, le mémoire essaiera tout d'abord d'apporter un cadre théorique sur l'apprentissage mobile ainsi que sur l'interaction homme-machine du point de vue tactile.

Une fois le cadre théorique en place, le mémoire se focalisera sur les choix techniques qui seront fait afin d'adapter au mieux le nouveau logiciel à son support innovant et le rendre mobile, de par son utilisation multiplateforme.

2 Cadre théorique : Outils mobiles et tactiles en éducation

2.1 Le support mobile en tant qu'outil éducatif



Figure 2 : Exemple de supports mobiles (Jaynes, 2013)

Le support mobile (Figure 2) est partout : téléphones, ordinateurs portables, smartphones, tablettes tactiles, liseuses, lecteurs mp3, consoles de jeux portables, et autres supports média se trouvent autant dans les poches qu'à la maison, au travail, et bien évidemment dans les écoles.

Dans cette partie du mémoire, nous traiteront des supports mobiles dans l'éducation en général avant de nous focaliser sur les tablettes de manière plus spécifique.

2.1.1 Définition et usages

Lorsqu'on utilise le terme « mobile » en se référant à l'apprentissage, plusieurs définitions émergent.

Ainsi selon Hashemi, Azizinezhad, Najafi, & Nesari (2011), l'apprentissage mobile peut être défini comme « une exploitation des technologies « de poche » omniprésentes, en combinaison avec les réseaux sans fil, pour faciliter, aider, améliorer et étendre la portée de l'enseignement et de l'apprentissage ». D'autres vont dans un sens similaire, comme Alexander (2004), cité par Hamdani (2013), qui définit l'apprentissage mobile comme l'utilisation de supports mobiles en tant que médiateur dans le processus d'apprentissage et d'enseignement.

Naismith, Sharples, Vavoula, & Lonsdale (2004), élargissent l'apprentissage mobile aux apprenants. Selon ces auteurs, il est possible d'apprendre de manière mobile sur un support fixe car c'est l'apprenant qui se déplace dans ce cas et a accès à une information dans un lieu différent. L'exemple typique d'un apprentissage mobile sur support fixe serait la borne informative dans les musées.

Une définition plus large de l'apprentissage mobile est proposée par Vavoula & Sharples (2002), cité par (Naismith et al., 2004) : « *l'apprentissage est mobile en termes d'espace, il a lieu au travail, à la maison et à des moments de loisirs ; l'apprentissage est mobile au travers différentes aires de la vie, il peut être lié à des demandes au travail, à des améliorations personnelles, ou aux loisirs ; il est mobile en fonction du temps, il se passe à des moments différents de la journée, le weekend ou les jours de la semaine.* » Dans un esprit similaire, Kukulka-Hulme & Traxler (2007), cité par Bidin & Ziden (2013), se concentre plus sur la mobilité de l'apprenant dans sa définition de l'apprentissage mobile.

En ce qui concerne l'usage des supports mobiles, il est similaire à l'usage de ses prédécesseurs, ordinateurs fixes. Selon Bétrancourt M. (2012), l'usage des supports technologiques peut se classer dans 4 catégories :

- Stockage et réutilisation : par l'accès à l'information sur le Web, le stockage et création et partage de productions, comme par exemples de wikis
- Visualisation : l'aspect le plus utilisé par les enseignants, qui consiste à consulter voire créer des vidéos, animations, etc.
- Traitement automatique : par les exercices, ou jeux éducatifs, qui permettent à l'élève de progresser à son rythme et à l'enseignant de se passer des corrections
- Communication et collaboration : de par les outils de création collaboratifs (wikis, etc.), de par la possibilité d'interagir à plusieurs (notamment le multi-touch des tablettes) : ces aspects encouragent la discussion entre élèves et la confrontation des opinions pour un meilleur apprentissage.

Les courants sont donc partagés entre l'apprenant mobile ou le support mobile, mais sont d'accord sur le côté omniprésent lié à l'apprentissage mobile et le fait que sa portée est étendue à des environnements divers. Il est donc possible d'apprendre n'importe où, et n'importe quand, que ce soit grâce au support mobile ou grâce à l'apprenant qui accède alors à un contenu mobile diffusé sur des supports fixes. En ce qui concerne les usages, le mobile reprend les usages déjà connus de ces prédécesseurs avec un aspect plus collaboratif.

2.1.2 Avantages

Selon (Naismith et al., 2004) le support mobile bouscule les pratiques de l'enseignement tout en étant compatible avec les pratiques issues des courants pédagogiques classiques.

En effet, on y retrouve :

- un aspect béhavioriste, le stimulus se trouve dans la présentation du problème, suivi par la réponse qui est la contribution de l'apprenant à la solution, alors que le feedback du système joue le renforcement
- le constructivisme, car l'apprenant est encouragé à participer et construire activement ses connaissances
- l'apprentissage situé, car le mobile est particulièrement bien adapté pour se placer dans un contexte donné
- les aspects collaboratifs via des discussions en ligne, le « informal and lifelong learning », car l'on peut apprendre n'importe où et n'importe quand sans même en avoir l'intention.

Hashemi et al. (2011), donne une liste non exhaustive des avantages des supports mobiles de type « handheld », c'est-à-dire : smartphones, PDA, consoles de jeux portables, tablettes, mini-notebooks, etc. Il cite le fait que comparé à un support fixe de type ordinateur, les apprenants peuvent interagir les uns avec les autres et avec l'enseignant au lieu d'être cachés derrière de grands écrans d'un ordinateur fixe. Il parle notamment du fait qu'un tel support est bien plus léger et donc plus facile à transporter, que les supports mobiles peuvent être utilisés n'importe où et à n'importe quel moment, et que ces supports suscitent de l'intérêt pour les apprenants, surtout les jeunes. L'avantage logistique qui peut concerner une salle de classe est aussi abordé par ces auteurs. En effet, il est plus facile d'aménager une classe avec un support mobile par élève plutôt qu'avec un ordinateur par élève, ordinateur qui nécessite un bureau suffisamment grand, des prises électriques à proximité etc.

Bidin & Ziden (2013) propose un classement en deux catégories des avantages du mobile en tant qu'outil éducatif, du point de vue d'utilisabilité et du point de vue fonctionnel. Dans la catégorie utilisabilité on retrouve le fait que le support soit léger et très portable, les apprenants sont plus à l'aise et plus libres, et le processus d'apprentissage devient flexible et peut avoir lieu à tout moment en tout lieu. Dans la catégorie des aspects fonctionnels, on retrouve l'accès à l'information à tout moment dans tout lieu ce qui rend l'apprentissage moins contraignant. De plus, le support fournit des informations de manière instantanée grâce à des applications tels que les dictionnaires, les calculatrices etc.

Druin (2009, p. 15, p.27), dit que, comparé à un apprentissage plutôt sédentaire sur PC, l'utilisation d'un support mobile est très motivant, et augmente l'engagement des enfants dans leur apprentissage. Elle cite notamment une étude faite en Chine, qui démontre que des enfants en scolarité élémentaire ont amélioré leur vocabulaire d'anglais et ont été beaucoup plus enthousiastes dans leur apprentissage au travers d'une plateforme mobile (Tan & Liu, 2004).

On conclut que les avantages principaux du support mobile se retrouvent dans sa facilité de transport, sa disponibilité lorsque l'apprenant le souhaite, et l'intérêt que celui-ci suscite auprès du jeune public. L'apprentissage devient aussi plus flexible. Cette flexibilité amène plus de contrôle sur son apprentissage à l'apprenant, ce qui renforce son envie d'apprendre et lui permet d'être au centre de son processus d'apprentissage (Watts, 1997 ; Selfe, 1999 ; Bidin & Ziden, 2013).

2.1.3 Inconvénients

Comme tout support éducatif, le mobile n'a pas que des avantages. En reprenant Hashemi et al., (2011), on peut citer la taille des écrans, qui est souvent relativement petite par rapport à un écran d'ordinateur, ce qui limite la façon dont l'information peut être affichée.

Cette limite va bien au-delà d'une simple mise à l'échelle d'un affichage prévu sur un écran plus grand. Il ne suffit pas de réduire la taille des menus et des icônes (Druin, 2009, p. 92), il faut aussi tenir compte du style d'interaction qui devient court et rapide contrairement à une interaction sur ordinateur. Cela implique de se poser des questions et de prendre des décisions difficiles sur ce qu'il faut garder ou non sur un écran de petite taille.

Parmi les autres inconvénients, cités par Hashemi et al., (2011), on retrouve :

- la durée des batteries qui doivent être régulièrement rechargées, surtout si l'on utilise du contenu multimédia.
- la bande passante limitée (ex. 3G), ce qui peut rendre difficile l'utilisation des logiciels avec beaucoup de graphiques, vidéos ou de sons.
- la fragilité du support, plus fragile qu'un ordinateur, et, du fait qu'il soit si facilement transportable, plus propice à des accidents.

Enfin, le marché du support mobile évolue rapidement. De nouvelles versions de téléphones, de tablettes, etc. sortent plusieurs fois dans l'année, ce qui fait que le support peut très vite être dépassé.

De ces inconvénients, il est important de retenir que le support est en évolution constante. De ce fait, si l'on souhaite adapter un logiciel vers un support mobile, il faut que celui-ci soit conçu de manière à pouvoir s'adapter au changement. De plus, il ne suffit pas de transposer un contenu prévu sur des ordinateurs à grands écrans et le réduire de manière à ce que son affichage soit proportionnel à la

taille de l'écran mobile. De même, qu'il ne suffit pas de transposer le contenu tel quel, alors que la bande passante est limitée voire inexistante. Le contenu doit être envoyé au support mobile de manière à gérer les problématiques de connexion.

Enfin, étant donnée la fragilité du support, celui-ci doit être géré par l'enseignant dans le cadre d'une utilisation en classe. Les enseignants devront donc s'occuper de le recharger régulièrement, et surveiller les élèves afin de limiter la casse.

2.1.4 Le mobile et les enfants

Druin (2009, p. xviii) définit les jeunes d'aujourd'hui comme des « iChildren ». Elle choisit ce terme car ils sont interactifs, indépendants et internationaux. En effet, les jeunes d'aujourd'hui, dans les pays développés du moins, utilisent des formes de technologie diverses et variées, allant de l'ordinateur au smartphone, en passant par les consoles de jeux. Ils sont indépendants dans leur utilisation de ces technologies, et ils sont connectés au monde qui les entoure via des blogs, des chaînes Youtube, etc.

Les enfants commencent à manipuler les nouvelles technologies et à comprendre les interfaces avec écran à partir de l'âge de 2 ans (Revelle & Strommen, 1990). Certains vont même jusqu'à classer les générations en deux catégories, les natifs et les migrants, « digital native and digital immigrant » (Prensky, 2001). Selon cette classification les enfants d'aujourd'hui sont des utilisateurs natifs et grandissent avec les technologies, ce qui les rendrait très à l'aise avec leur utilisation.

Toutefois, si l'on applique cette classification à un autre domaine, tel que par exemple l'écriture, on constate que l'on naît tous avec l'écriture mais qu'il faut tout de même apprendre à lire et à écrire. Il en va de même en ce qui concerne les technologies. En effet, les jeunes générations font un usage fréquent des outils technologiques, mais leur compétence réelle en est-elle plus élevée ? Selon Baron & Bruillard (2008), les jeunes ont une tendance à banaliser rapidement les outils, mais il faut se méfier de cette impression de familiarité, car plusieurs études suggèrent que le savoir-faire acquis est réel, mais « qu'en l'absence de l'enseignement la conceptualisation ainsi que le spectre d'utilisation sont souvent limités » (GIANNOULA & BARON, 2002 ; Fluckiger, 2008 ; Mediappro, 2006). Baron & Bruillard (2008) va jusqu'à suggérer que les « digital natives » sont en fait des « digital naïves » du fait qu'ils sont des proies faciles pour les incitations du marché. La preuve est que beaucoup de supports mobiles leurs sont directement destinés.

Druin (2009, p. 47) dresse une liste non-exhaustive des supports mobiles destinés aux enfants depuis les années 1970 à nos jours (45 supports différents). Cette liste contient autant de consoles de jeux vidéo, que de mini-ordinateurs destinés aux enfants, des tablettes avec stylos interactifs, en passant par le célèbre Tamagotchi, jusqu'aux supports plus évolués tels que les tablettes tactiles éducatives de VTech, fabricant de jouets pour enfants.

2.1.5 Conclusion

Au vue des définitions relatives à l'apprentissage mobile, qui tendent soit vers l'apprenant soit vers le support, il serait idéal de prendre en compte ces deux aspects.

Afin d'optimiser l'apprentissage mobile, il faut donc essayer de créer des contenus qui sont disponibles sur les supports mobiles et sur les supports fixes, afin que l'apprenant puisse y accéder à son gré et profiter pleinement de sa mobilité. De plus, afin d'avoir un apprentissage mobile de qualité, il faut tenir compte des paramètres de taille, portabilité et connectivité des technologies mobiles. Ces facteurs

doivent être centraux à tout effort de design selon Druin (2009, p. 31). Des applications qui ont été créées sans tenir compte de la différence entre les supports mobiles et les autres supports, peuvent affecter l'utilisabilité et empêcher l'utilisateur d'atteindre les buts d'apprentissage (Druin, 2009, p. 32).

Le facteur de l'évolution rapide des supports mobiles et aussi un élément clé dans le design de logiciels éducatifs destinés à ces supports. Les logiciels doivent tenir la route et pouvoir s'adapter à ces évolutions constantes, ils doivent aussi être utilisables sur des types de supports différents (mobiles tactiles ou ordinateurs fixes) pour maximiser leur apport.

En ce qui concerne les enfants en particulier, ils sont nés avec les technologies à portée de main et les utilisent de manière fréquente et spontanée. Leur enthousiasme dans l'utilisation de ces technologies (Druin, 2009, p. 15 ; Hashemi et al., 2011) les pousserait à s'engager plus dans leur apprentissage, ce qui pourrait à terme mener vers un apprentissage meilleur. Le choix d'un logiciel sur support mobile pour l'aide à l'apprentissage du français aux enfants de 2 à 5 ans prend son sens au vu des avantages de l'apprentissage sur support mobile. Celui-ci devra toutefois tenir compte des différents inconvénients du monde mobile et y répondre de manière adaptée.

Enfin, parmi les catégories d'usages définies par Bétrancourt M. (2012), le traitement automatique proposant des exercices et donnant à l'élève la possibilité de progresser à son rythme, est l'usage qui va nous intéresser dans la présente recherche.

2.2 La tablette tactile en tant qu'outil éducatif

2.2.1 Définition

Parmi les supports mobiles vus précédemment, on retrouve donc des outils allant d'ordinateurs portables aux consoles de jeux portables, en passant par les smartphones et tablettes tactiles.

Les supports tactiles n'en sont pas à leurs débuts dans l'histoire de la technologie. Ils existent depuis plusieurs décennies sous des formes diverses (ex. calculatrices, écrans tactiles, pavé tactile remplaçant la souris, etc.) (Buxton, 2010). Mais, ce n'est qu'à partir du milieu des années 2000 que le support tactile s'est généralisé auprès du grand public, à savoir avec l'utilisation croissante de smartphones, puis des tablettes tactiles, et depuis peu des ordinateurs hybrides (ex. Windows 8).



Figure 3 : Tablette tactile (source <http://www.silicon.fr/>)

Cette partie du mémoire traitera essentiellement des tablettes tactiles (Figure 3). A la différence des ordinateurs portables avec écran tactile ou pavé tactile, les tablettes ont comme seule interaction

possible leur écran. Nous allons donc nous concentrer sur ce type de support uniquement, ces derniers étant aussi des supports mobiles qui peuvent servir de médiateurs de contenus d'apprentissage.

La sortie du premier iPad en 2010 a provoqué un véritable engouement pour les tablettes tactiles. Avec sa connexion 3G et Wifi, sa portabilité et sa facilité d'utilisation, ce nouveau gadget était créé pour amener internet dans les mains des utilisateurs (Jobs, 2010 cité dans Meurant, 2010). La concurrence a vite fait de suivre la vague, et en 2011, on compte plus de 75 modèles de tablettes tactiles différentes. Les ventes de tablettes sont immenses, plus de 63 millions **vendues dans le monde**, pendant la seule année 2011 (Lachapelle-Bégin, 2012).

Le monde de l'éducation a suivi cet engouement : notamment au Québec où plus de 10'000 élèves utilisent des tablettes en classe au quotidien, et aussi aux Etats-Unis où ce chiffre dépasse les 4,5 millions (KARSENTI & FIEVEZ, 2013).

Mais il ne suffit pas d'équiper les écoles et les élèves de tablettes tactiles pour améliorer leur apprentissage, ainsi que le dit Bétrancourt, M. (2012) : « *Comme pour tout support numérique, c'est moins ce que l'outil fait qui importe que ce que l'on peut faire avec cet outil.* »

En faisant la part des avantages et inconvénients des tablettes, nous allons essayer d'identifier les recommandations utiles pour la présente recherche en vue de les appliquer dans le futur logiciel destiné à une utilisation non seulement mobile, mais aussi prévue pour des tablettes tactiles.

2.2.2 Avantages

Les avantages d'un support mobile se retrouvent évidemment dans les tablettes, il est donc inutile de les répéter ici.

En plus d'être un support mobile, la tablette tactile propose aussi une interaction simplifiée et plus intuitive comparé à un ordinateur. En effet, la tablette réintroduit une interaction qualifiée de directe par Norman, du fait qu'elle réintroduit la coordination œil-main puisque l'œil suit le doigt et non un curseur sur l'écran comme c'était le cas avec la souris. Cette interaction simplifiée permet de réduire les ressources mobilisées par l'utilisateur pour manipuler son interface (Bétrancourt, 2012). Lachapelle-Bégin (2012) rejoint Bétrancourt sur l'interaction simplifiée et précise que la tablette favorise les expériences d'apprentissage kinesthésiques, en prenant l'exemple d'une application de représentation de molécules en trois dimensions que les élèves peuvent manipuler avec les doigts. Selon Lachapelle-Bégin (2012), ce type de simulation peut être très stimulant pour les étudiants, notamment dans des matières dont le contenu est abstrait (physique, chimie, biologie, etc.). Cet aspect intuitif du tactile est un réel avantage dans une utilisation de la tablette en tant que support éducatif, puisque l'élève perdra moins de temps à s'approprier l'outil, il pourra ainsi se concentrer sur la tâche éducative. En outre, il devient plus facile pour l'enseignant d'organiser des activités puisqu'il faut moins d'explications et d'interventions sur l'utilisation de l'outil.

Par rapport à un autre support mobile comme l'ordinateur portable, la tablette offrirait une meilleure intégration avec les autres outils de classe : elle se range facilement dans le bureau, son format est proche de celui d'un cahier. De plus, du fait qu'elle peut être posée sur la table, son écran dérange peu aux interactions entre élève et enseignant. Le cas de l'ordinateur portable diminue déjà cette interposition, comme on l'a vu dans les avantages du support mobile en général, mais la position plutôt horizontale de la tablette accentue cette diminution (Bétrancourt, 2013 cité dans Boujol 2014).

L'écran tactile rend aussi le travail collaboratif autour d'une même tablette plus convivial : pas de souris ou de clavier à faire circuler (Lachapelle-Bégin, 2012 ; Bétrancourt, 2012). De plus l'écran étant multi-touch (permet plusieurs points de contact), on peut imaginer des applications qui acceptent des interactions simultanées, comme par exemple une application musicale qui permettrait de jouer au piano à 4 mains, difficilement faisable sur un ordinateur.

Murphy (2011), cite Melhuish & Falloon (2010) qui dit que la tablette (à l'époque il parlait directement du modèle Apple, l'iPad) offre toutes les fonctionnalités et connectivités d'un ordinateur portable avec la portabilité d'un smartphone. Il est vrai que les tablettes tactiles ont des fonctionnalités similaires à un ordinateur, on y retrouve des applications de traitement de texte, des applications de partage de documents (DropBox, Google Drive), des applications de communication vidéo (Skype, FaceTime), l'accès aux emails, etc.

Qui plus est, ces applications sont très faciles à installer, une fois qu'un compte a été configuré dans le store de la tablette (ex. Apple Store ou Play Store). Il n'y a pas d'étapes ni de messages de confirmation, comme c'est le cas sur les ordinateurs, et elles sont régulièrement mises à jour de manière invisible pour l'utilisateur. Plus important encore, les applications sont souvent gratuites ou peu chères. Si l'on prend en exemple un support mobile tel qu'un ordinateur portable, les catégories de prix sont très différentes. En effet on peut trouver des applications à 1 ou 2 francs (Apple Store, Google Play Store), comparé à une application pour PC ou MAC dont les prix d'entrée se situent à 50 francs (Manor, Genève).

Enfin, plusieurs études suggèrent une motivation accrue de la part des étudiants dans leur apprentissage au travers d'une tablette tactile. Cette motivation semble être perçue autant du point de vue des apprenants (Kinash, Brand, & Mathew, 2012; Karsenti & Fievez, 2013) que du point de vue des enseignants en observant leurs apprenants (Sachs & Bull, 2011).

Ces avantages de portabilité, de facilité de collaboration, de diminution d'interposition entre l'élève et l'enseignant et de coûts font de la tablette un outil idéal pour les salles de classe.

2.2.3 Inconvénients

En plus des inconvénients déjà cités concernant les supports mobiles, les interfaces tactiles présentent quelques inconvénients supplémentaires. Bétrancourt M. (2012) parle notamment du fait que l'on se serve des doigts comme pointeur, ce qui rend l'interaction moins précise. De plus, elle rappelle que la tablette réagit au moindre contact, même involontaire, ce qui peut poser problème.

De par son écran tactile, la tablette est plus sujette aux salissures, de plus son utilisation en pleine lumière n'est pas optimale (Bétrancourt, M., 2012).

Selon Bétrancourt (2012), l'interaction simplifiée de la tablette est à la fois avantage et inconvénient de cet outil, car elle ne permet pas la subtilité des applications PC. De ce fait, elle serait plutôt réservée à des usages ponctuels, en déplacement ou avec des enfants en bas âge. Cet avis s'oppose à l'idée un peu utopiste Melhuish & Falloon (2010) qui parlait d'un outil égal à un PC portable (ci-haut).

Enfin, une étude de Karsenti et Fievez (2013), a mis en évidence que certains manuels scolaires disponibles pour une utilisation sur iPad étaient parfois mal adaptés, ce qui pouvait décourager les élèves. Qu'il s'agisse d'un problème de compatibilité avec le type de tablette ou de besoin de

connexion à Internet en permanence, il s'agit d'inconvénients qui peuvent vite devenir frustrants et dont il faut tenir compte dans le développement d'un logiciel sur tablette.

2.2.4 Conclusion

L'accès spontané à la tablette qui ouvre des portes sur le World Wide Web et sur des milliers d'applications éducatives font de ce support un outil intéressant à exploiter en classe. Pour une utilisation optimale, il faut adapter les logiciels en vue d'une utilisation tactile et mobile, et faire en sorte que ceux-ci soient compatibles avec les différentes versions existantes des tablettes tactiles.

De plus, il faut tenir compte de son interaction simplifiée qui limite son utilisation dans des contextes plus poussés. Cette simplicité rend toutefois l'outil intéressant pour un usage ponctuel avec des jeunes enfants qui pourront s'approprier l'outil facilement et profiteront pleinement du contenu éducatif mis à disposition.

2.3 IHM tactiles

L'interaction homme-machine (IHM) définit les moyens mis en œuvre pour que l'homme puisse contrôler et communiquer avec une machine (Wikipedia, 2014).

Parmi les moyens à disposition pour communiquer avec un ordinateur, nous avons l'écran, la souris, le clavier, le pavé tactile, la caméra ou encore le microphone pour la reconnaissance vocale et les hauts parleurs pour les réponses de la machine. En ce qui concerne les tablettes tactiles, nous disposons toujours d'un écran, d'une caméra, d'un microphone et de haut-parleurs. Mais, la souris, ainsi que le clavier, laissent place à des interactions avec les doigts, qui remplacent la souris, et à un clavier tactile avec aide à la saisie, qui remplace le clavier standard.

Comme vu précédemment, les supports ayant une Interaction Homme-Machine (IHM) tactile, proposent une interaction simplifiée par rapport à un ordinateur standard. Cette simplicité est non seulement due à l'absence de périphériques externes (ex. clavier, souris) mais aussi au fait que les interactions se font directement avec les doigts. Ceci réintroduit la coordination œil-main, l'œil suit et contrôle ce que fait la main, ce qui n'est pas le cas lorsqu'on travaille avec une souris puisque la main est sur la souris alors que les yeux sont sur l'écran. L'interaction est donc directe comme celle imaginée par Norman (Hutchins et al., 1985).

Afin de concevoir un outil facilement utilisable sur un support tactile, il faut donc définir les gestes utilisés lors des interactions avec cet outil, gestes qui remplaceront la souris lorsqu'on reprend un logiciel existant sur un ordinateur.

Dans la suite de ce mémoire, nous allons donc nous centrer sur la conception d'IHM tactiles ainsi que sur les IHM tactiles spécifiquement destinées aux enfants afin de tirer les meilleures recommandations pour la conception d'un logiciel utilisable sur un support tactile.

2.3.1 Conception d'IHM tactiles

Comme cela a été dit à plusieurs reprises, les IHM tactiles ont pour principal outil d'interaction le doigt. Mais, les interactions faites avec les doigts doivent correspondre à ce que le support tactile attend (comprend). Il existe donc des recommandations et des classifications des gestes effectuels sur un écran tactile, par un ou plusieurs doigts (multi-touch).

Lao, Heng, Zhang, Ling, & Wang (2009) proposent un modèle d'interaction multi-touch et classifient les gestes faisables par les doigts en deux catégories principales : gestes simples et complexes. Ces derniers étant une combinaison de plusieurs gestes simples.

Les gestes simples sont au nombre de trois :



Figure 4 : Gestes basiques selon Lao et al. (2009)

- Appui-long (pressing), qui consiste à toucher l'interface et garder le doigt appuyé.
- Appui-court (tapping), qui consiste à toucher l'interface et enlever le doigt aussitôt
- Glissement (dragging), qui consiste à toucher l'interface tout en effectuant un mouvement

Ces trois gestes principaux (Figure 4), dits simples, se retrouvent aussi dans les Guides de développement de Apple pour leurs supports tactiles, ainsi que dans les guides de développement Android et Microsoft. On remarque toutefois une différence d'appellation de ces gestes selon la plateforme. Ainsi certains vont utiliser touch, d'autres tap pour définir le geste d'appui court, nous allons ici utiliser le terme « tap » pour l'appui-court, le terme « drag » pour le glissement et le terme « long press » pour l'appui-long.

En plus de ces 3 gestes simples, les tablettes tactiles multi-touch supportent des combinaisons de gestes simples qui deviennent alors des gestes plus complexes. Parmi les gestes dits complexes les plus utilisés, on retrouve le zoom, qui se fait en réalisant un pincement avec deux doigts qui s'écartent l'un de l'autre pour zoomer ou le contraire pour dé-zoomer. Un autre geste fréquemment utilisé et cité également dans les guides de développement est le double-tap, qui consiste à tapoter deux fois de suite l'écran de manière rapide. Il peut parfois faire office de zoom ou permettre de sélectionner un texte.

En faisant le tour des gestes faisables avec les doigts, on remarque certaines similitudes avec ceux de la souris. On peut ainsi assimiler le clic de la souris à un « tap » avec le doigt. Le « double tap » est compte à lui l'équivalent du double-clic. Le « long press » fait office de clic droit car il permet souvent d'activer un menu contextuel ou alors de faire des actions supplémentaires sur les objets touchés. Le drag a son équivalent bien sûr avec la souris et son drag and drop. Enfin, avec le doigt, il est possible aussi de faire du « scrolling » simplement en faisant le geste du drag directement sur l'écran ou sur une barre de défilement. Le défilement dans les interfaces tactiles est faisable dans les deux sens, horizontal et vertical, ce qui est un plus par rapport à la souris standard à roulette.

Les doigts couvrent donc la plupart des interactions que la souris proposait déjà. Néanmoins, il reste une interaction qui ne peut pas être gérée dans les interfaces tactiles : il s'agit du survol. Avec une souris, lorsqu'on déplace le curseur à l'écran, on n'effectue pas d'action tant qu'on n'a pas appuyé sur l'un des boutons, alors qu'avec les doigts, on se passe du curseur et on appuie directement à l'endroit qui nous intéresse. Il n'y a donc plus possibilité de faire apparaître des info-bulles, jouer des sons ou animer les menus au survol de la souris. Il est important de tenir compte de l'absence de cette interaction dans la conception d'interfaces tactiles : l'utilisation doit être intuitive et ne doit pas

nécessiter d'info-bulles contextuelles, ou alors elle doit être expliquée autrement. De plus il faut éviter les menus qui se déroulent au survol de la souris au risque de bloquer l'utilisateur (Leclerc, 2012).

Les interfaces tactiles étant multi-touch, certaines interactions permettent l'utilisation d'un ou plusieurs doigts simultanément. L'aspect multi-touch doit faire l'objet de quelques précautions, notamment chez les utilisateurs n'ayant pas l'habitude de ces interactions. En effet, selon une étude de Wobbrock, Morris, & Wilson (2009), qui consistait à étudier les interactions gestuelles d'un groupe de 20 individus n'ayant jamais eu à faire à des interfaces tactile, il serait recommandé de ne pas trop tenir compte du nombre de doigts utilisés lors d'une interaction. Les gens ne feraient pas attention au nombre de doigts utilisés de manière générale, sauf pour des activités très précises comme par exemple jouer du piano. Cependant, on peut déduire qu'à partir de 4 doigts utilisés, le geste est exécuté par la main plutôt que par un ensemble de doigts, ce qui nous permet de fixer une limite et ainsi distinguer un tapotement de doigts d'un tapotement de main qui peut être significatif. L'étude de Wobbrock datant de 2009, il est judicieux de tenir compte de l'évolution technologique de ces 5 dernières années. Il faut considérer notamment l'utilisation généralisée des smartphones qui sont désormais entièrement tactiles. Grâce à ces appareils, les gens ont pris l'habitude d'interagir avec un seul doigt sauf lorsque l'utilisation de plusieurs doigts est nécessaire. Les tablettes étant souvent similaires aux smartphones, cette habitude devrait être naturellement transmise dans l'utilisation de ces dernières.

Le doigt peut donc quasiment tout faire, qu'il soit seul ou en combinaison avec ses voisins, mais il amène aussi quelques complications. En effet, comparé au pointeur de la souris, le doigt a une taille nettement plus grande. Les composants touchables de l'interface doivent donc être de taille suffisante. Selon Leclerc (2012), il faut avoir des éléments touchables non-contigus d'au minimum 25px de haut et 38px de large, ce qui fait 5mm par 7,6mm. Dans le cas d'éléments contigus, la hauteur minimale passe à 42 pixels (8,4mm) et il faut prévoir un espace suffisant entre ces éléments pour ne pas pouvoir en toucher deux à la fois.

Ces recommandations sur la taille des éléments touchables se retrouvent aussi dans le livre « Usability of iPad apps and websites, 2nd edition » (Budiu & Nielsen, 2011). Les auteurs préconisent une taille d'au moins 1cm sur 1cm, ce qui est un peu plus grand que vu précédemment, mais qui a du sens si l'on tient en plus compte du fait que l'utilisateur peut éventuellement interagir avec plusieurs doigts comme l'a démontré l'étude de Wobbrock et al. (2009).

Budiu & Nielsen (2012), suggèrent de distinguer clairement les objets et menus touchables dans une application. Il faut donc définir une manière de démarquer tous les objets avec lesquels l'utilisateur peut interagir.

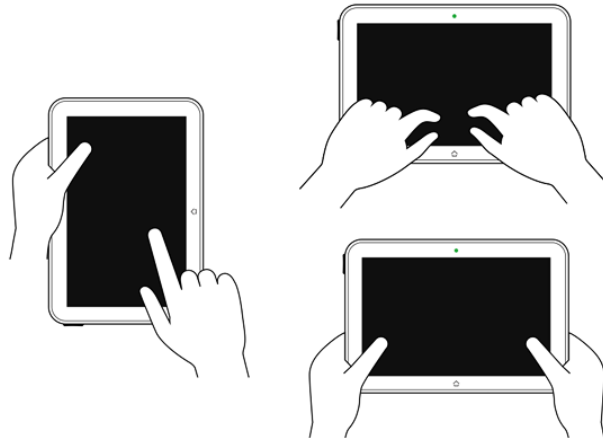


Figure 5 : Exemples d'utilisation d'une tablette (Wroblewski L., 2012)

De plus, concernant la disposition des éléments tactiles, il faut se rappeler que les supports tactiles sont mobiles et que leur utilisation peut se faire tout en les tenant dans les mains ou sur les genoux (Figure 5), ce qui fait que les éléments de l'interface doivent pouvoir être atteignables avec des mouvements réduits. Il est donc préconisé de disposer les éléments en fonction de cela. La zone la moins accessible d'une tablette se trouve sur la partie du haut de l'écran.

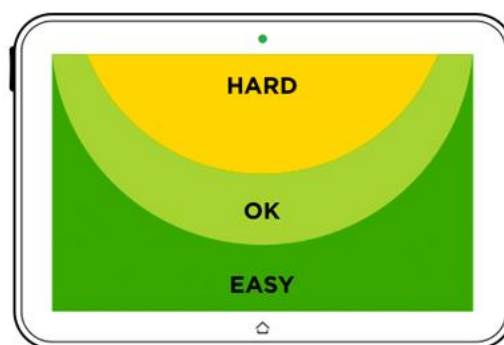


Figure 6 : Zones atteignables sur une tablette (Wroblewski L., 2012)

Sur la Figure 6, les zones les plus atteignables sont marquées en vert. Il est clairement visible que la partie du haut de la tablette est difficile à atteindre. Ces paramètres sont à prendre en compte lors d'une utilisation de la tablette, tout en la tenant dans les mains.



Figure 7 : Exemple d'utilisation de tablette posée sur le bureau (Boujol, 2014)

Il en est de même lorsqu'on pose la tablette sur un bureau (Figure 7) et que l'on souhaite interagir sans bouger le coude. La main reste plutôt en bas de l'écran, et dans le cas où il faut interagir avec la partie haute de l'écran, il faut bouger le bras ce qui peut entraîner une certaine fatigue si il s'agit d'un mouvement qui se fait souvent.

Selon le site www.ergonomie-interface.com, l'idéal est donc de placer les actions les plus utilisées dans la partie basse de l'écran, alors que les actions occasionnelles sont à positionner dans la partie haute. Les actions de type suppression ou quitter sont aussi à placer dans les parties hautes afin d'éviter un déclenchement accidentel qui peut être problématique. De plus, il faut tenir compte du fait que, lorsque l'utilisateur doit appuyer sur une zone de l'écran, sa main masque une partie de l'interface. Il faut donc disposer les éléments tactiles de manière à ce que les interactions puissent être effectuées sans nuire à la visibilité de l'écran. Ce même conseil est à prendre en compte pour l'emplacement des libellés pour certaines icônes. Sur les ordinateurs, les libellés sont souvent placés en dessous des icônes, or, sur un support tactile on risque de les cacher avec la main. Il est donc recommandé de mettre les libellés plutôt en-dessus des icônes quand c'est possible (Druin, 2009, p. 281).

Regroupement en zones

1. très visible, peu accessible
2. très visible
3. peu accessible
4. très accessible
5. la + visible, la + accessible
6. très accessible
7. peu accessible
8. peu visible, peu accessible
9. peu accessible

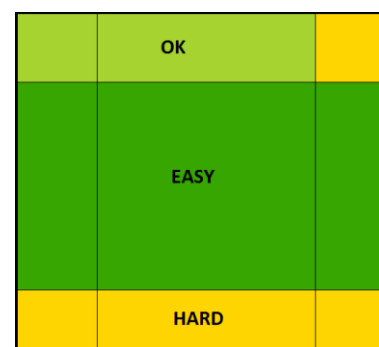
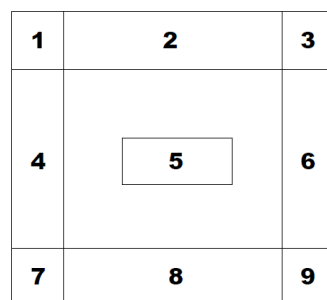


Figure 8 : Zones atteignables sur un écran d'ordinateur (Nogier, 2003)

Il est intéressant de comparer ces zones atteignables sur tablettes à celles habituellement proposées sur un ordinateur. Dans la Figure 8, reprise de Nogier 2003, les zones d'écran les plus visibles et atteignables sont présentées sur la gauche. Nous avons retranscrit ces zones dans le style de la Figure 6 qui présentait les zones atteignables sur tablette, afin d'avoir une comparaison visuelle. Nous

pouvons constater que la partie tout en bas de l'écran sur une tablette devient très intéressante alors que c'était le contraire dans le cas d'un écran d'ordinateur.

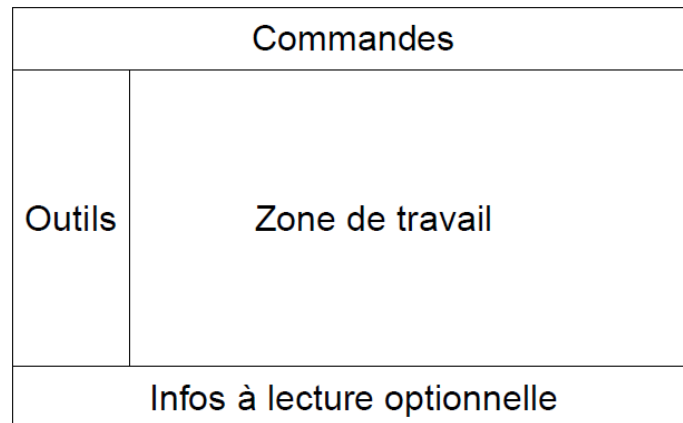


Figure 9 : Disposition des informations sur un ordinateur (Nogier, 2003)

Sur la Figure 9, on constate que les outils les plus utilisés dans les logiciels destinés aux ordinateurs se trouvaient sur la gauche. Or sur la tablette, on pourrait les imaginer dans toute la zone en vert comme vu sur la Figure 6, c'est-à-dire sur la gauche, la droite et le bas de l'écran.

En plus de ces recommandations spécifiques aux tablettes tactiles, il convient de suivre quelques recommandations usuelles dans les interactions homme-machine selon Budiu & Nielsen (2011). Afin de faciliter la navigation dans l'application, il faut aussi toujours prévoir un bouton qui permet de revenir en arrière, et prévoir des messages de confirmation lorsque les actions entraînent une interruption ou une suppression. De plus, lorsqu'une application demande un chargement préalable d'éléments multimédia, qui peuvent être volumineux et donc prendre du temps à charger selon la vitesse de connexion de l'utilisateur, il est recommandé de prévoir toujours une barre de progression qui informe réellement l'utilisateur du progrès d'avancement (p.83). Dans le cas où les chargements doivent prendre plus de 20 secondes, il faut penser à occuper l'utilisateur durant l'attente, afin d'éviter qu'il s'ennuie et qu'il finisse par abandonner l'utilisation de l'application.

2.3.2 Conception d'IHM tactiles destinées aux enfants

En ce qui concerne la conception d'IHM tactiles spécifiquement destinées aux enfants, peu d'études ont été trouvées dans notre recherche. Nous reprendrons dans un premier temps des recommandations destinées à des logiciels pour ordinateurs avant d'amener quelques éléments pour les logiciels destinés aux supports tactiles.

Selon un article de Michel, Sandoz-guermond, & Serna (2011), plusieurs études (Colombi 2008, Hourcade, 2008 ; Lueder & Rice, 2007 ; Markopoulos, Read, MacFarlane, & Hoysniemi, 2008) se sont déjà concentrés sur les critères à intégrer dans un logiciel ludo-éducatif, et ont notamment précisé l'importance du type d'interaction (souris, stylet, etc.) et de l'utilisation de sons dans les logiciels pour enfants. Ces études sont centrées sur les interfaces WIMP¹, cependant, certaines parties peuvent s'appliquer dans le monde tactile, notamment l'utilisation de sons, ainsi que le type d'interaction qui

¹ Windows, Icons, Menus, Pointing device

sera ici le doigt, similaire au stylet de par son interaction directe. La présence de sons et d'instructions auditives dans les logiciels destinés aux enfants est également recommandée par Talley, Lancy, & Lee (1997), cité par Gimbert & Cristol (2004). D'autres caractéristiques sont conseillées, tels que des tâches d'apprentissage ouvert, avec des routines animées et des possibilités d'interruption ainsi que des feedback rapides. Ces caractéristiques permettraient aux enfants d'entretenir leur intérêt pour le logiciel.

Druin (2009, p. 280) se base sur les résultats de ses différentes recherches et propose des recommandations clés en matière de conception de logiciels tactiles destinés aux enfants. Selon elle, il serait judicieux de :

- *Utiliser des instructions audio claires et fréquentes* : parfois même avec des instructions orales, les enfants hésitent à toucher l'écran, surtout lorsqu'ils sont novices, c'est pourquoi une invite audio claire et fréquente est recommandée pour expliquer les interactions.
- *Utiliser un langage spécifique et concret* : on évitera, par exemple, d'utiliser le mot « scroll » en préférant « toucher et bouger » qui est plus effectif auprès des enfants
- *Programmer les événements de manière à ce qu'ils réagissent au « long press » et non au « tap » (cf. 2.3.1)* : les enfants ont tendance à appuyer plus longuement sur l'objet interactif plutôt que de seulement l'effleurer. Si le logiciel ne répond pas, ils auront alors tendance à appuyer encore plus fort, ce qui peut provoquer une certaine frustration.
- *Utiliser le multi-touch avec précaution* : les enfants interagissent facilement avec un seul doigt, mais dès qu'il faut combiner l'utilisation de deux doigts (p. ex. zoom) l'utilisation s'avère plus difficile.
- *Eviter de mettre des boutons d'action trop près des bords de l'écran* : les enfants touchent les bords de la tablette lorsqu'ils la tiennent en main, ils peuvent donc activer des actions par accident, et ne pas comprendre ensuite ce qui s'est passé.
- *Prévoir un feedback immédiat, surtout en réponse au toucher* : s'il y a un délai de réponse les enfants vont penser que leur réponse est incorrecte et vont essayer autre chose. Il faudrait donc minimiser les problèmes de performance.
- *Utiliser des mises en évidence visuelles pour les éléments interactifs* afin de montrer à l'enfant quand les objets sont touchables ou non. Ceci rejoint les recommandations générales des IHM tactiles tels que décrites par Budiu & Nielsen (2011) (cf. 2.3.1).

2.4 Conclusion du cadre théorique

L'objectif de cette recherche était d'amener des éléments d'aide à la refonte d'un logiciel éducatif existant sur un support fixe vers un nouveau support innovant mobile et tactile. Afin d'aider dans cette refonte, nous avons besoin de découvrir de quelle manière le contenu existant devait être modifié afin de permettre sa diffusion sur d'autres supports. De plus, le support visé étant à interaction tactile, il était important de définir les changements d'interactions nécessaires pour une utilisation sans souris.

Dans un premier temps, nous nous sommes intéressés aux aspects du support mobile dans l'apprentissage ainsi qu'au cas spécifique de la tablette tactile.

Cette partie de la recherche nous a permis de comprendre qu'il est important qu'un logiciel éducatif puisse non seulement être disponible sur un support mobile, mais aussi sur un support fixe afin d'être complémentaire à la mobilité de l'apprenant. Celui-ci pourra alors l'utiliser sur des supports mobiles et fixes, en fonction du lieu où il se trouve et des moyens qu'il a à disposition.

De plus, nous avons vu que le contenu diffusé sur un support mobile devrait subir quelques modifications et tenir compte des éléments suivants :

- la bande passante réduite ou même inexistante, afin de prévoir des moyens de diffuser le contenu des conditions sans connexion.
- la taille réduite des écrans, pour adapter les contenus et leur disposition
- l'évolution rapide du support, qui nécessite des logiciels standardisés, à la pointe technologique et facilement adaptables au changement

Dans le cas particulier de la tablette tactile nous retenons, en plus de ce qui s'applique au mobile en général, que bien que son interaction simplifiée soit un avantage, elle empêcherait un travail sur des tâches trop poussées ou précises. Ceci la rendrait intéressante plutôt pour des usages ponctuels. Il serait donc préférable de prévoir des contenus sur lesquels on peut revenir de manière irrégulière.

Dans un deuxième temps, nous avons analysé les recommandations pour la conception d'IHM tactiles en vue de rendre le logiciel utilisable sur les supports tactiles qui ont comme seule interaction possible leur écran et les doigts de l'apprenant.

Cette partie de la recherche nous a amené les éléments suivants :

- Les objets avec lesquels l'apprenant doit interagir doivent être clairement reconnaissables
- La disposition des objets touchables doit répondre au schéma des zones atteignables sur tablette qui ne sont pas les mêmes que sur un écran d'ordinateur
- La taille des objets touchables devrait prendre en compte le changement de la taille du curseur : en effet les doigts humains sont plus épais que le pointeur de la souris.
- L'espacement entre les objets touchables contingents devrait également tenir compte de ce changement de taille du curseur
- Le survol n'existant plus dans l'interaction tactile, il faudrait changer les interactions qui s'en servaient afin de ne pas bloquer l'utilisateur
- La main pouvant cacher certaines parties de l'écran, il est recommandé de réfléchir à la disposition des légendes et des éléments nécessitant une visibilité constante sur l'écran

En ce qui concerne la conception d'IHM tactiles destinées spécifiquement aux enfants, il convient de tenir compte de quelques éléments supplémentaires :

- Les objets touchables ne devraient pas se trouver trop près des bords de l'écran
- Les instructions devraient de préférence être présentées sous forme audio et doivent être claires et concrètes
- Les objets touchables doivent répondre au « long press » et pas seulement au « tap » car l'enfant a tendance à appuyer plus longuement sur l'écran

De manière générale, nous retenons aussi quelques recommandations qui sont applicables à tout type de support :

- Laisser à l'utilisateur la possibilité de revenir sur l'écran précédent à tout moment
- Donner un feedback immédiat à l'utilisateur sur son action
- Demander confirmation lorsqu'une action engendre la fin du programme ou une suppression
- Informer l'utilisateur sur la progression du chargement s'il y en a

Ces différentes lectures, bien que peu nombreuses, étant donné l'aspect récent des supports visés, nous ont donné des éléments intéressants et importants pour notre futur travail de développement. Nous allons essayer d'appliquer ces conseils dans le développement du « Moulin à paroles ».

Dans la suite de ce mémoire, nous décrivons la phase de conception et de développement du futur logiciel. Puis, une fois le logiciel terminé, nous effectuerons des tests d'utilisabilité afin de mettre en confrontation les éléments retenus de notre recherche et l'utilisation réelle du futur logiciel par le public cible.

3 Conception

3.1 Présentation du logiciel existant

Afin de décrire les changements effectués dans la nouvelle version du logiciel, nous allons commencer par décrire l'existant de manière plus détaillée.

Le logiciel existant est de type "point & click", c'est-à-dire que la plupart des exercices sont basées sur un mécanisme où l'enfant doit pointer la souris sur un élément et cliquer dessus pour interagir avec celui-ci. Il y a toutefois quelques exercices qui sont de type drag'n'drop, mais le clic est prédominant, comme nous le verrons ci-après dans l'analyse fonctionnelle.



Figure 10 : Ecran de navigation

Le jeu démarre sur un écran de navigation (Figure 10) qui permet de choisir un domaine de connaissances (ex. Animaux, Habits, Objets, Phrases, Prépositions, etc.).

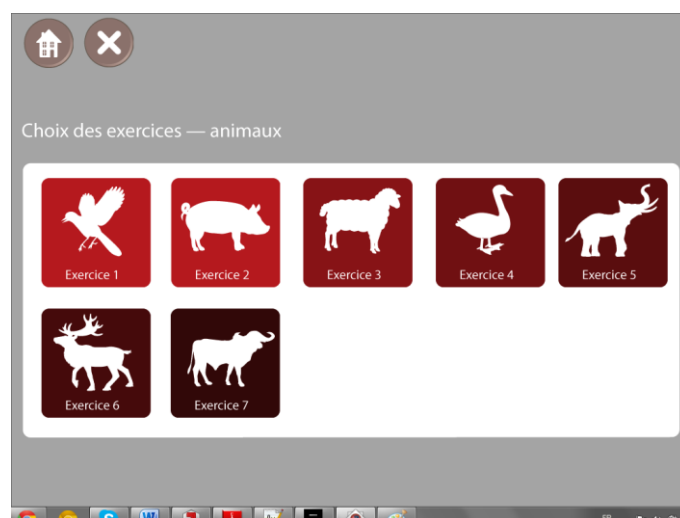


Figure 11 : Navigation 2ème niveau

Puis, pour chaque domaine, l'enfant peut choisir un exercice dans un écran de navigation de deuxième niveau (Figure 11). Dans chaque domaine il y a plusieurs exercices avec à chaque fois de nouveaux mots de vocabulaire. Les exercices sont ordonnés par ordre de difficulté.

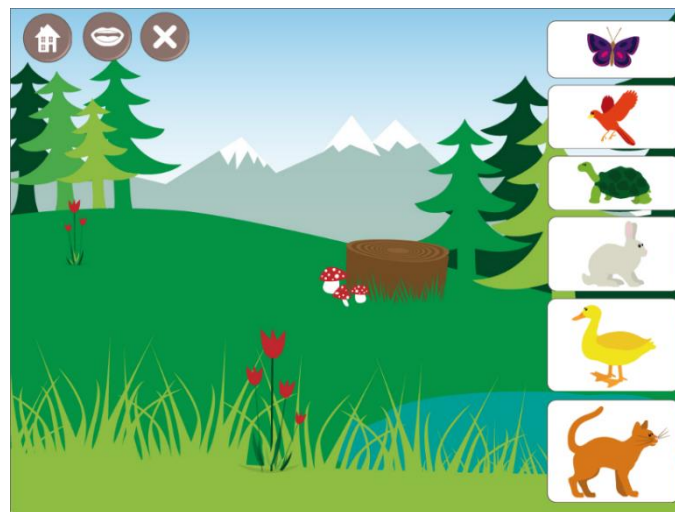


Figure 12 : Exemple Animaux, exercice 1

Une fois que le choix de l'exercice à l'intérieur du domaine est fait, l'enfant se trouve dans une scène donnée (Figure 12) et doit cliquer sur les éléments à disposition selon les instructions données par la voix off en début de chaque exercice.

Ci-dessus un exemple d'exercices sur les animaux (Figure 12). Nous y voyons un écran central que l'on va dénommer ci-après « la scène ». Dans l'exemple ci-dessus nous sommes à la campagne. Sur la droite on voit les éléments sur lesquels l'enfant doit cliquer, ces éléments seront dénommés ci-après « éléments interactifs ». Au début de chaque exercice, la voix off, explique à l'enfant la situation globale ainsi que ce qu'il doit faire. Par exemple dans le niveau ci-dessus, la voix dit *“Lili et Théo se baladent à la campagne et découvrent plein d'animaux qu'ils ne connaissent pas. Aide-les à les découvrir !”*.

La voix va ensuite donner des noms d'animaux, par exemple “Lapin” et l'enfant doit cliquer sur le lapin parmi les animaux présents sur la droite de la scène. S'il clique sur le bon animal, l'animal va apparaître sur la scène, et ainsi de suite jusqu'à ce que l'enfant ait cliqué sur tous les animaux. Si l'enfant se trompe, la voix l'en informe et prononce à nouveau le son de l'animal demandé afin que l'enfant puisse réessayer. Lorsque l'enfant sélectionne le bon animal, un feedback encourageant lui est donné, par exemple « *Bravo, tu as trouvé !* ».

Grâce aux icônes se trouvant en haut à gauche de la scène, l'enfant peut à tout moment, respectivement : retourner à l'écran de navigation principal, répéter le dernier mot prononcé par la voix off, ou quitter le jeu.

Dans d'autres niveaux, la voix ne dit pas à l'enfant sur quel élément il faut cliquer. L'enfant entend le mot correspondant à l'élément en passant la souris dessus et peut ensuite soit cliquer dessus pour le sélectionner soit le tirer sur la scène à l'aide du drag'n'drop.

Durant les périodes où la voix off parle à l'enfant, aucune action sur la scène n'est admise, même si l'enfant clique sur les éléments interactifs, rien ne se passe, ce n'est qu'une fois que la voix a fini de parler que les interactions sont acceptées. Il en va de même pour les icônes de navigation.

Nous allons décrire ci-dessous de manière plus détaillée les différents types d'exercices et leurs interactions spécifiques.

3.2 Recueil des fonctionnalités

Afin d'entamer le développement de la nouvelle version du logiciel, nous avons commencé par faire une analyse de l'existant en faisant un recueil des fonctionnalités existantes du logiciel. Pour faire cela, nous avons créé un fichier Excel dans lequel nous avons noté tous les exercices existants. Il a donc fallu parcourir une première fois l'ensemble du logiciel éducatif avec ces 100 exercices groupés en 12 domaines de connaissances.

Dans un premier temps, chaque exercice a été décrit en détail de manière à pouvoir revenir dessus directement dans le fichier Excel et s'en rappeler sans avoir à rejouer.

Dans un deuxième temps, nous avons revu notre liste d'exercices pour tenter de les regrouper en catégories. Pour faire ce regroupement nous avons tenu compte principalement de deux critères : la disposition des éléments interactifs sur la scène, et la manière dont l'enfant était guidé dans les exercices. Le fait que l'enfant était guidé ou non dans les exercices était important dans le cadre de cette recherche puisque, dans ce type d'exercices, la voix off était déclenchée au moment du passage de la souris sur les éléments. Or, nous avons vu que cette interaction disparaît lors d'un passage vers une utilisation tactile, nous devons donc distinguer ces exercices des autres afin de pouvoir par la suite adapter le développement. La division en catégories d'exercices s'est faite en deux temps.

En effet, nous avons d'abord identifié 8 types d'exercices possibles, puis après une analyse plus approfondie nous avons groupé ces types en 2 catégories principales avec chacune des sous-catégories. Les deux principales catégories se différencient par le fait que l'enfant est guidé ou non par les instructions de la voix off.

En effet, dans certains exercices l'enfant doit cliquer sur l'élément demandé par la voix, alors que dans d'autres, l'enfant peut choisir librement les éléments utilisés dans la situation donnée.

Dans cette deuxième catégorie, lorsque l'enfant passe la souris sur un élément, la voix off prononce le mot qui se trouve sur la carte et l'enfant décide ensuite s'il veut ou non utiliser cet élément dans l'exercice.

Ci-dessous un tableau récapitule l'ensemble des types d'exercices.

| Numéro catégorie | Description | Nombre d'exercices |
|------------------|--|--------------------|
| 1 | Guidé par la voix | 87 |
| 1.1 | Éléments interactifs sur la droite | 28 |
| 1.2 | Éléments interactifs à gauche et à droite | 3 |
| 1.3 | Éléments interactifs sur scène | 20 |
| 1.4 | Memory | 12 |
| 1.5 | Cartes à retourner | 20 |
| 2 | Libre, avec voix sur passage de la souris | 17 |
| 2.1 | Totalement libre | 8 |
| 2.2 | Choix cohérent avec la situation | 2 |
| 2.3 | Libre partiel | 7 |

3.2.1 Type d'exercices 1 : Guidé par la voix

Nous voyons dans le tableau ci-dessus qu'une grande majorité des exercices est de ce type. Les enfants sont guidés par la voix off, et doivent cliquer sur les éléments demandés. On voit que les éléments interactifs sont majoritairement uniquement sur la droite de la scène (catégorie 1.1) comme dans l'exemple ci-dessous (Figure 12). Un très petit nombre d'exercices reprend ce même fonctionnement mais avec des éléments à droite et à gauche de la scène.

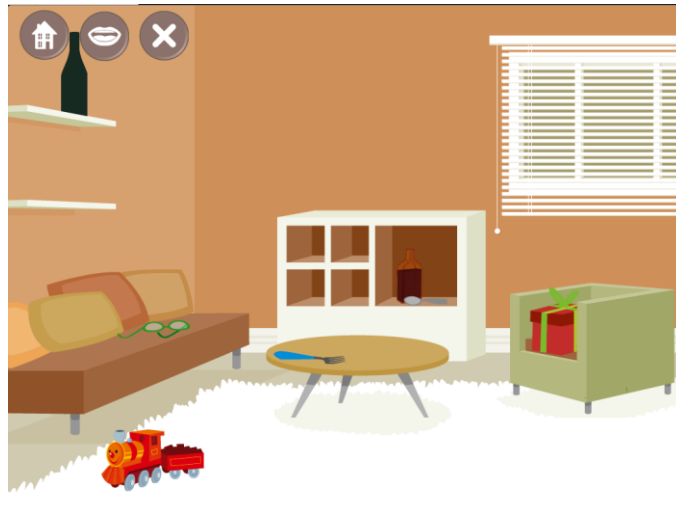


Figure 13 : Exemple d'exercice de catégorie 1.3

Il y a aussi des exercices dans lesquels les éléments interactifs sont dispersés sur la scène et l'enfant doit les trouver directement dans le décor (Figure 13). Dans cette catégorie (1.3), lorsque l'enfant clique sur l'élément se trouvant sur la scène, cet élément disparaît de l'écran à la différence des catégories précédentes, dans lesquelles les éléments cliqués disparaissaient de la droite pour réapparaître sur la scène directement. C'est le cas par exemple dans le domaine Objets (Figure 13), ou encore Prépositions.

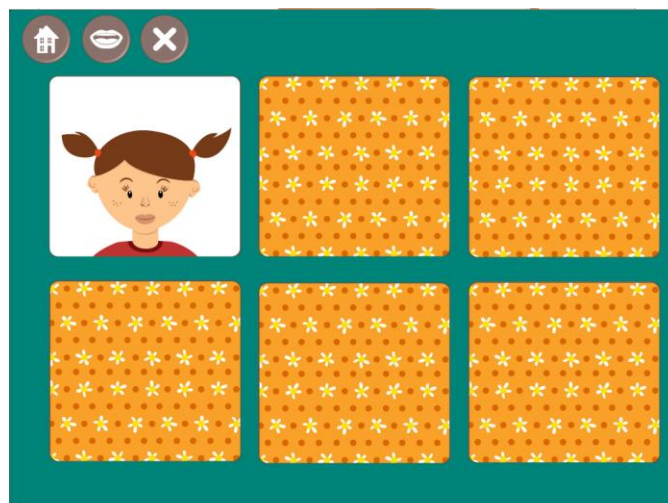


Figure 14 : Exemple d'exercice de catégorie 1.4

La catégorie 1.4 concerne un jeu de type « Memory » mais en version auditive (Figure 14). Cette fois, l'enfant n'est plus du tout dans une configuration de scène comme précédemment. Il arrive sur un écran avec 6 cartes retournées. La voix dit ensuite le nom d'un objet, par exemple « Belle » et la première carte correspondant à l'objet est automatiquement retournée. Ensuite, l'enfant peut passer la souris sur les autres cartes, et en passant la souris, il entend quel mot se cache derrière la carte retournée. Il faut trouver le même mot que celui de la carte déjà retournée en se basant sur ce qui est entendu. Les mots cachés ont une certaine consonance. Par exemple on trouvera dans le même niveau les mots balle, bol, belle.

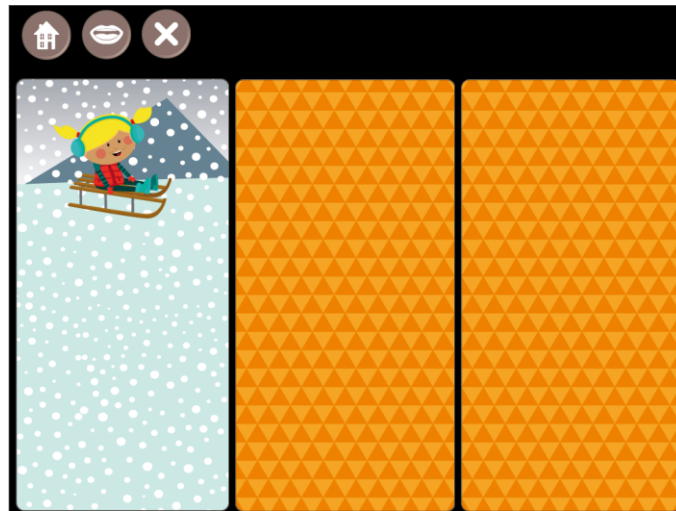


Figure 15 : Exemple d'exercice de catégorie 1.5

La catégorie 1.5 (Figure 15) est similaire à la 1.4 si ce n'est qu'on arrive sur un écran où il y a seulement 3 cartes. Ces trois cartes illustrent une phrase chacune. Au début de l'exercice, les 3 cartes apparaissent de dos. Puis une phrase est dite et la carte correspondante est retournée. Une deuxième phrase suit, puis une troisième et à chaque fois la carte correspondante est retournée. Les cartes sont ensuite à nouveau de dos, et la voix off demande à l'enfant de trouver la carte correspondant à l'une ou l'autre des phrases. Ce type d'exercice se diffère des autres par le fait que l'enfant doit se rappeler une situation plutôt que reconnaître un élément.

3.2.2 Type d'exercices 2 : Libre, avec voix sur le passage de la souris

Ce deuxième type d'exercice est clairement minoritaire. Il s'agit d'un mode où l'enfant est plus libre. En effet ce n'est plus la voix qui lui dicte sur quel élément il faut cliquer. Il doit lui-même écouter les différents éléments à disposition et en choisir un.



Figure 16 : Exemple d'exercice de catégorie 2.1

La catégorie 2.1 est totalement libre, c'est-à-dire que l'enfant choisit les éléments sur la droite ou gauche dans l'ordre souhaité, et il doit tous les sélectionner pour terminer le niveau (Figure 16). On retrouve cet exercice dans l'apprentissage du domaine Habits. L'enfant doit habiller Lili, il choisit les habits sur la gauche et sur la droite. Lorsqu'il passe la souris dessus il entend de quel habit il s'agit. Lorsqu'il clique dessus, Lili met l'habit sélectionné et la voix off dit à l'enfant, par exemple : « Lili a mis ses chaussettes ».

La catégorie 2.2 a un comportement similaire sauf qu'il faut choisir uniquement les éléments cohérents à la situation. Par exemple, s'il fait froid et qu'il faut habiller Lili, le choix d'un maillot de bain parmi les éléments à disposition est considéré comme faux.

La catégorie 2.3, est totalement libre de choix, comme dans la 2.1, sauf que pour terminer le niveau l'enfant doit choisir 3 éléments parmi les six proposés, alors que dans la catégorie 2.1 l'enfant devait sélectionner tous les éléments interactifs.

Le fichier Excel contenant les descriptions détaillées des catégories ainsi que la classification des exercices est disponible en annexe.

3.3 Objectifs retenus pour la refonte du logiciel

Une fois le recueil et l'analyse des fonctionnalités existantes terminés, nous avons fait la part des contraintes du projet et avons décidé de fixer des objectifs basés sur notre cadre théorique ainsi que sur le recueil de fonctionnalités.

3.3.1 Aspect multiplateforme du logiciel

Comme vu dans le cadre théorique, du fait de la mobilité de l'apprenant et du contenu d'apprentissage, il est important de concevoir un logiciel qui fonctionne sur différentes plateformes. Au vu de la multitude de versions de tablettes tactiles et d'ordinateurs, le choix de développement en HTML5 semble un allié idéal. En effet, il s'agit d'une technologie Web. Le web fonctionne sur n'importe quel système du moment que celui-ci dispose d'une connexion internet et d'un navigateur web. De plus, HTML5 étant le nouveau courant du web, il se profile comme un futur standard qui pourra permettre de garder notre logiciel fonctionnel même si le support sur lequel il est diffusé ne cesse d'évoluer.

Cependant, les navigateurs web, eux aussi, peuvent différer d'une plateforme à une autre, et nous allons ici nous concentrer sur 4 navigateurs principaux :

- Chrome (installé par défaut sur les plateformes avec un système d'exploitation Android)
- Safari (installé par défaut sur les plateformes avec un système d'exploitation Apple)
- Internet Explorer (installé par défaut sur les plateformes avec un système d'exploitation Windows)
- Mozilla FireFox (navigateur populaire tout public)

HTML5 étant encore en cours de validation, tous les navigateurs ne supportent pas forcément toutes les fonctionnalités de ce nouveau langage. Il faut donc choisir les composants de ce projet avec prudence et vérifier s'ils sont compatibles sur les supports mobiles utilisés et sur les différents navigateurs disponibles sur ces plateformes.

Pour ce faire, nous avons utilisé le site « [Can I use](#) », qui s'avère très utile pour déterminer à l'avance les possibilités des différents navigateurs du marché. Sur ce site, nous pouvons avoir un aperçu rapide de l'étendue de support par les navigateurs actuels de différents éléments introduits par HTML5, tel que par exemple l'élément Canvas, le support des sons, de vidéos etc. Nous allons ci-après détailler chacun des éléments nécessaires dans notre futur logiciel et vérifier leur fonctionnement sur les différents navigateurs.

3.3.1.1 Composant « Canvas » de HTML5

Etant donné que notre logiciel est composé principalement d'images, nous avons opté pour l'utilisation du composant Canvas de HTML5. En effet, ce composant permet une gestion d'éléments graphiques très performante et similaire aux possibilités de Flash. En effet, cet élément permet une gestion optimale d'images sur plusieurs couches (scène, éléments interactifs, boutons, etc.) avec des possibilités d'animations et de déplacement ainsi que de drag'n'drop.

De plus, l'élément Canvas prévoit un mécanisme d'échelonnage (scaling) des images afin que l'utilisation de notre futur logiciel soit optimale sur toutes les plateformes. En effet, étant donné que le logiciel devra pouvoir s'exécuter sur des tablettes aux formats variés, voir sur des smartphones, ainsi que sur des ordinateurs, il est important de prévoir un contenu à taille adaptable et optimisée pour toutes les plateformes.

| # Canvas (basic support) - Candidate Recommendation | | | | | | | | | | Usage stats: Global | |
|--|------|---------|--------|--------|-------|------------|------------|-----------------|--------------------|---------------------|--------|
| Method of generating fast, dynamic graphics using JavaScript | | | | | | | | | | Support: | 83.77% |
| | | | | | | | | | | Partial support: | 4.21% |
| | | | | | | | | | | Total: | 87.98% |
| Show all versions | IE | Firefox | Chrome | Safari | Opera | iOS Safari | Opera Mini | Android Browser | BlackBerry Browser | IE Mobile | |
| | | | | | | | | 2.1 | | | |
| | | | | | | | | 2.2 | | | |
| | | | | | | | | 2.3 | | | |
| | | | | | | | | 3.0 | | | |
| | | | | | | | | 3.2 | | | |
| | | | | | | | | 4.0-4.1 | | | |
| | | | | | | | | 4.2-4.3 | | | |
| | 8.0 | | 31.0 | | | | | 4.0 | | | |
| | 9.0 | | 32.0 | | | | | 4.1 | | | |
| | | | | | | | | 4.2-4.3 | | | |
| | 10.0 | 28.0 | 33.0 | 5.1 | | | | 7.0 | | | |
| | | | | | | | | 6.0-6.1 | | | |
| Current | 11.0 | 29.0 | 34.0 | 7.0 | 20.0 | 7.0 | 5.0-7.0 | 4.4 | 10.0 | 10.0 | |
| Near future | | 30.0 | 35.0 | | 21.0 | | | | | | |
| Farther future | | 31.0 | 36.0 | | 22.0 | | | | | | |
| 3 versions ahead | | 32.0 | 37.0 | | | | | | | | |

Figure 17 : Support de l'élément Canvas

Nous voyons dans la Figure 17 ci-dessous que l'élément Canvas est bien supporté par les différents navigateurs actuellement sur le marché, et notamment par ceux qui nous intéressent dans la présente recherche : Chrome, Safari, Internet Explorer et Firefox.

3.3.1.2 Les fonctionnalités audio de HTML5

Un autre défi du support multiplateforme sera la gestion du son. Ce défi se découpe en deux dimensions : la gestion du son par le Javascript grâce à l'API Web Audio de HTML5, et le support de différents formats de son (wav, mp3, etc.).

En effet, bien que les nouveautés HTML5 commencent à être supportées par la majorité des navigateurs modernes, il faut savoir que chaque navigateur a sa propre gestion de ces nouveautés. Ainsi, notamment pour le son, tous les formats ne sont pas supportés de la même façon par tous les navigateurs.

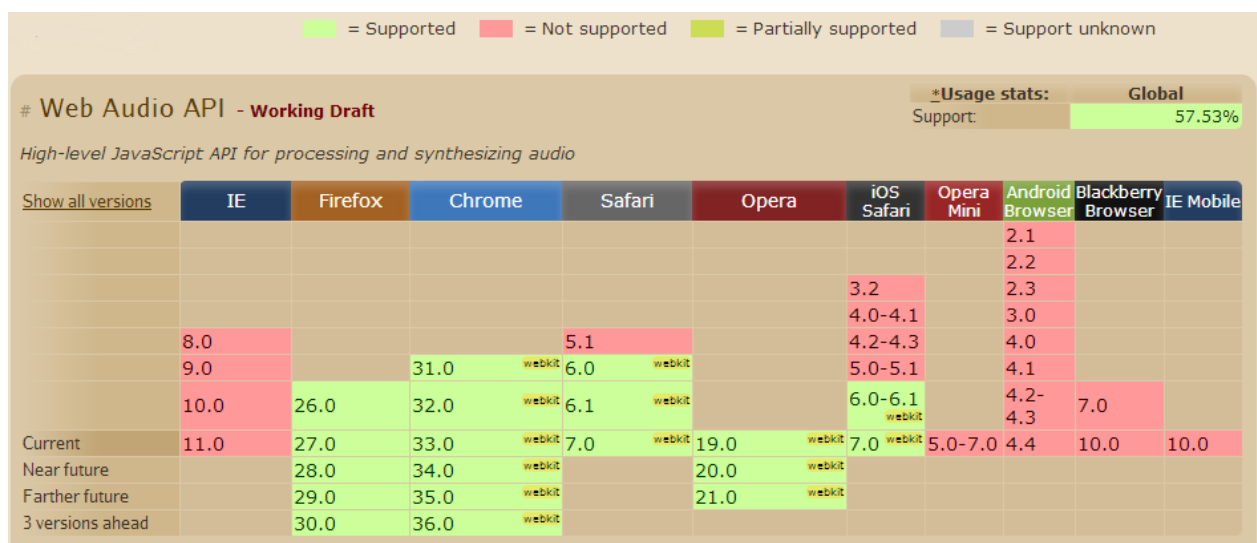


Figure 18 : Support de l'Audio grâce à l'API Web Audio en JavaScript

Nous pouvons voir dans la figure ci-dessous (Figure 18), que l'API Web Audio est gérée par Firefox ainsi que Chrome et Safari mais qu'il risque d'y avoir des problèmes avec Internet Explorer.

| Desktop Browser | Version | Codec Support |
|-------------------|---------|----------------------|
| Internet Explorer | 9.0+ | MP3, AAC |
| Chrome | 6.0+ | Ogg Vorbis, MP3, WAV |
| Firefox | 3.6+ | Ogg Vorbis, WAV |
| Safari | 5.0+ | MP3, AAC, WAV |
| Opera | 10.0+ | Ogg Vorbis, WAV |

Figure 19 : Formats de son supportés

En ce qui concerne le support des divers formats, nous pouvons voir dans la Figure 19 ci-dessus la diversité des formats supportés par chaque navigateur.

Dans le cadre de ce projet, les différents sons du jeu existant nous sont fournis. Ils sont actuellement tous au format WAV. Nous voyons ci-haut que ce format est bien géré par la plupart des navigateurs, sauf Internet Explorer.

Cela étant, le problème de l'encodage n'est pas un problème insurmontable en soi. En effet, il suffirait de ré-encoder nos sons actuels au format wav, pour les passer dans l'un ou l'autre des formats demandés par les différents navigateurs. Malheureusement, la transformation d'un format à un autre prend du temps étant donné le nombre élevé de sons à disposition dans « Le Moulin à Paroles ». Nous allons donc les laisser au format WAV dans le cadre de ce travail, ce qui fait que le logiciel ne sera pas fonctionnel dans Internet Explorer dans un premier temps.

3.3.2 Aspect interaction tactile

Nous avons vu dans le cadre théorique que du fait d'un passage vers l'interaction tactile, certaines parties du logiciel actuel doivent être modifiées.

Ainsi, nous allons entreprendre les modifications suivantes :

- Disposer les boutons dans les zones répondant au schéma des zones atteignables sur tablette qui ne sont pas les mêmes que sur un écran d'ordinateur
- rendre les éléments interactifs dans les exercices plus reconnaissables grâce à un effet d'ombre qui va les différencier de la scène
- Eloigner les éléments interactifs les uns des autres afin de prendre en compte la taille des doigts. Ceci va entraîner une réduction du nombre d'éléments disponibles sur la scène afin de gagner de la place.
- Eliminer les éléments qui sont trop petits pour une interaction tactile : certains exercices contiennent des petits éléments comme par exemple des fourchettes, couteaux, lunettes qui nécessitent un pointeur précis comme c'est le cas du pointeur de souris mais pas du doigt humain
- Repenser les niveaux dans lesquels les interactions nécessitent un survol de souris
- Refaire les icônes de manière à afficher leur légende au-dessus des images et non en-dessous pour éviter qu'elle ne soit cachée par la main

3.3.3 Aspects de suivi de progrès

Du fait que le futur logiciel est à but pédagogique, nous allons y ajouter des options de suivi de progrès afin de permettre à l'enfant ou à son parent/enseignant de voir les exercices réalisés ainsi que les résultats obtenus dans ces exercices. Ces indications seront gérées à l'aide d'un système d'étoiles comme c'est le cas dans de nombreux logiciels qui ont déjà fait leurs preuves sur les plateformes mobiles et fixes (Angry Birds, Trivial Pursuit, Puzzles animaux, etc.). Ce type d'indications permet non seulement de savoir où l'on est dans l'avancement des exercices, mais peut également inciter le joueur à vouloir se surpasser en recommençant le niveau jusqu'à obtention du meilleur score possible. De plus, s'agissant d'une symbolique très répandue dans les jeux et autres applications disponibles sur tablette, il permettra à notre futur logiciel de ressembler à ce qui se fait actuellement sur le marché mobile.

3.3.4 Aspect diffusion du contenu sur support mobile

Comme vu dans le cadre théorique, le fait que le logiciel soit prévu pour une utilisation mobile, nécessite de penser à la manière de diffuser le contenu sur ces supports en cas de vitesse de connexion réduite ou non existante.

Pour pallier à ce problème nous allons dans un premier temps essayer de répartir le téléchargement du contenu tout au long du jeu afin d'éviter des temps d'attentes supérieures à 20 secondes à l'enfant.

Web Storage - name/value pairs - Recommendation

Method of storing data locally like cookies, but for larger amounts of data (sessionStorage and localStorage, used to fall under HTML5).

Usage stats: Global
 Support: 89.65%
 Partial support: 0.08%
 Total: 89.73%

| Show all versions | IE | Firefox | Chrome | Safari | Opera | iOS Safari | Opera Mini | Android Browser | Blackberry Browser | IE Mobile |
|-------------------|------|---------|--------|--------|-------|------------|------------|-----------------|--------------------|-----------|
| | | | | | | | | 2.1 | | |
| | | | | | | | | 2.2 | | |
| | | | | | | 3.2 | | 2.3 | | |
| | | | | | | 4.0-4.1 | | 3.0 | | |
| | 8.0 | | 31.0 | | | 4.2-4.3 | | 4.0 | | |
| | 9.0 | | 32.0 | | | 5.0-5.1 | | 4.1 | | |
| | 10.0 | 28.0 | 33.0 | 5.1 | | 6.0-6.1 | | 4.2-4.3 | 7.0 | |
| Current | 11.0 | 29.0 | 34.0 | 7.0 | 20.0 | 7.0 | 5.0-7.0 | 4.4 | 10.0 | 10.0 |
| Near future | | 30.0 | 35.0 | | 21.0 | | | | | |
| Farther future | | 31.0 | 36.0 | | 22.0 | | | | | |
| 3 versions ahead | | 32.0 | 37.0 | | | | | | | |

Notes Known issues (2) Resources (6) Feedback Edit on GitHub

Figure 20 : Support du Web Storage API par les navigateurs

Dans un deuxième temps, nous souhaiterions stocker les différentes images et sons directement dans le navigateur de l'utilisateur à l'aide des nouvelles fonctionnalités HTML5, à savoir le Web Storage API. Ceci permettra à l'enfant d'exécuter le logiciel sans connexion internet, à partir du moment où il a joué au jeu au moins une fois et que celui-ci a été chargé dans son navigateur. Nous pouvons voir dans la Figure 20 : Support du Web Storage API par les navigateurs ci-dessus que cette fonctionnalité est supportée par tous les navigateurs ciblés.

3.4 Définition des priorités de développement

Afin de gérer au mieux le développement du projet nous avons décidé de fixer des priorités sur les fonctionnalités développées. Ceci nous a permis de fixer des objectifs à atteindre dans le temps restreint qui nous était imparti. En effet, dans le cadre de la réalisation du mémoire nous n'avons malheureusement pas le temps de développer le logiciel dans son ensemble.

Le logiciel initial a été développé de manière à respecter un ordre de difficultés, nous avons donc décidé de conserver ce paramètre dans la priorisation des développements. En effet, sachant qu'en plus le futur logiciel s'adresse à un public non-francophone il est important de commencer par des exercices faciles afin de permettre à l'enfant de commencer en douceur.

De plus, nous avons souhaité développer au moins un domaine d'exercices pour les deux types d'interaction principaux (guidée et libre avec voix au survol de la souris cf. 3.2.1 cf. 3.2.2). Bien que le deuxième type d'exercice soit minoritaire, nous souhaitons relever le défi de réflexion sur la suppression du survol de souris dans le cas d'une interaction tactile. Nous souhaitons également voir comment les enfants réagiraient à la nouvelle interaction de ces niveaux.

Enfin, la difficulté technique liée au développement a également été prise en compte dans la priorisation des fonctionnalités à développer.

En tenant compte de ces trois paramètres, nous avons décidé de commencer par développer des niveaux du domaine Animaux qui correspond au type d'exercice Guidé par la voix (cf. 3.2.1), puis

d'enchaîner sur le domaine Habits qui correspondait au type d'exercice libre avec voix au survol de la souris(cf. 3.2.2).

En fonction du tableau des catégories, nous avons pu tenir compte du nombre d'exercices proposés par chaque catégorie ce qui a permis de confirmer le début des développements par le domaine Animaux. En effet celui-ci contient des exercices de catégorie 1.1 (voir tableau 3.2). Ce type d'exercice est le plus nombreux de tout le logiciel (28 exercices de ce type). Il était donc important de commencer par mettre en place ce type d'exercices, afin de pouvoir par la suite ajouter les 28 autres.

Le deuxième domaine dans l'ordre des priorités était le domaine Habits, qui nous permettait de développer un mécanisme pour la gestion des exercices de type libre (catégorie 2.1, voir tableau 3.2).

Ensuite, nous nous sommes référés au nombre d'exercices de la catégorie 1.3 qui se trouve au nombre de 20, ce qui nous a permis de prioriser en troisième position le domaine Objet qui utilise les exercices de cette catégorie.

Le résumé des priorisations de développement peut être vu dans le tableau ci-dessous.

| Domaine | Difficulté pour l'enfant | Catégorie d'exercices | Nb d'exercices similaires | Priorité |
|---------|--------------------------|---|---------------------------|----------|
| Animaux | Facile | 1.1 Guidé par la voix, éléments interactifs sur la droite | 28 | 1 |
| Habits | Facile | 2.1 Libre, avec voix sur passage de la souris | 8 | 2 |
| Objets | Facile | 1.3 Guidé par la voix, éléments interactifs sur scène | 20 | 3 |

Pour la suite des développements, nous avons pris en compte les limitations techniques. Nous allons décrire dans les chapitres suivants les étapes détaillées du développement ainsi que les fonctionnalités développées.

3.5 Développement

3.5.1 Environnement de développement

3.5.1.1 IDE

Nous avons opté pour l'utilisation d'une IDE (Integrated Development Environment) gratuite permettant de développer des projets HTML5 avec un serveur local intégré. Il s'agit de l'IDE [NetBeans](#). Cette IDE est largement utilisée dans les milieux éducatifs ainsi que dans les entreprises. Elle est plus accessible qu'une IDE comme Eclipse. En effet, elle nécessite moins de configurations de la machine pour pouvoir être utilisée, et elle intègre dès son installation un serveur web qui fait que l'on peut lancer le projet en un seul clic. De plus, elle prend moins de place sur le disque dur. Nous avons donc choisi de l'utiliser dans le cadre de ce projet.

Nous pourrions penser que notre logiciel ne nécessite pas de serveur puisqu'il n'y a pas besoin de base de données et que tout le code sera fait du côté client en JavaScript. Mais, puisque notre logiciel doit lire des sons et des images à la volée, ces médias sont chargés en JavaScript avec des requêtes au serveur. Une fois chargées les ressources sont stockées en mémoire dans le navigateur, mais leur chargement préalable nécessite un appel au serveur. Il va donc de soi que pour exécuter notre logiciel, il faut pouvoir le publier sur un serveur.

3.5.1.2 « Versionning » du code source

En ce qui concerne la gestion des sources, nous avons opté pour le logiciel Git Extensions avec un dépôt centralisé en ligne sur la plateforme gratuite [GitHub](#).

Le choix s'est porté sur GIT pour des raisons de connaissances préalables du logiciel et pour son efficacité en matière de gestion des sources qui permet à la fois un « versionning » local et un « versionning » sur le dépôt sur serveur. Ceci nous permettait de pouvoir travailler dans le train sans connexion au dépôt GitHub et avoir tout de même un suivi de nos versions de code grâce au commit local, puis de publier le tout sur le serveur une fois la connexion à internet retrouvée grâce à la fonctionnalité push qui publie sur le serveur.

3.5.2 POC (Proof of Concept)

Dans un premier temps, nous avons décidé d'essayer de développer un seul niveau du jeu afin de valider la possibilité de développer le logiciel en utilisant les éléments définis du langage HTML5 et de valider le fait que le logiciel puisse fonctionner sur différents supports.

Cette phase de développement est appelée Proof Of Concept (POC). L'objectif d'un POC est de valider les choix techniques et la faisabilité du projet. Il permet ainsi de se ré-orienter techniquement ou fonctionnellement en cas de problèmes rencontrés avant d'aller trop loin dans le développement.

Dans le cas de notre logiciel, pour tester la faisabilité du projet nous avons besoin de tester l'exécution des sons et les interactions avec des images sur l'élément Canvas. En effet, il s'agit là du cœur du logiciel puisque celui-ci est entièrement basé sur les instructions données par la voix off et les interactions avec les images à l'écran.

Pour le POC, nous avons donc choisi de développer le premier exercice du domaine Animaux, tel quel. La réalisation du POC s'est fait en 3 étapes, jusqu'à obtention de la solution technique satisfaisante.

3.5.2.1 Version 1: Kinetic JS et Web Audio API²

Il a fallu, dans un premier temps, trouver une librairie JavaScript qui facilite et étend le travail avec le composant Canvas de HTML5, ce composant étant le composant central de notre jeu. L'utilisation d'une librairie pour travailler avec ce composant facilite notamment la superposition de différents éléments (scène, éléments interactifs, navigation, etc.). De plus, les librairies permettent en général d'avoir une gestion simplifiée des interactions, elles font le travail d'interprétation entre une interaction de type clic et une interaction de type toucher, ce qui permet de gérer les deux à la fois sans devoir répéter le même code.

En cherchant sur le web, nous avons trouvé la librairie [KineticJS](#) et avons décidé de la tester. Cette librairie était bien documentée et semblait pouvoir nous donner la possibilité de réaliser les fonctionnalités attendues.

Il s'agissait donc de réaliser le tout premier exercice du jeu existant, le niveau des "Animaux dans à la campagne" (Figure 12 : Exemple Animaux, exercice 1).

² Application Programming Interface

Nous avons commencé par dessiner un Canvas avec une image de fond (la campagne) et une couche supplémentaire contenant les animaux à placer. Sur chaque animal nous avons ajouté la gestion de l'évènement "click" et de l'évènement "tap" en JavaScript.

Dans un premier temps, nous avons simplement fait en sorte que sur chaque clic/tap l'animal soit placé sur la scène. Ceci nous a permis de valider que la gestion de ces deux évènements était fonctionnelle sur tous les supports à disposition (iPad, Tablette Android, Smartphone Android, et PC Windows).

Dans un deuxième temps, nous avons rajouté les sons et la gestion des règles du jeu. En effet il fallait ajouter la mécanique de tests, pour savoir si l'enfant a sélectionné l'animal demandé et lui donner le feedback correspondant.

Pour la gestion du son en JavaScript, nous avons utilisé l'[API Web Audio](#) de HTML5. Dans un premier temps nous nous sommes focalisés sur le fonctionnement des sons sur PC puisque nos outils de développement se trouvent sur PC. Puis nous avons déployé le POC sur le serveur TECFA à disposition des étudiants et l'avons testé sur Android. Il a fonctionné du premier coup sur un téléphone portable Samsung Galaxy S2.

| PC Chrome | Samsung Galaxy S2 | iPad | Samsung Galaxy Tab |
|---|---|---|--|
|  |  |  découverte des limitations du son sur support mobile |  pas de son du tout |

Figure 21 : Résultats du 1er POC

Malheureusement, il n'en était pas de même sur l'iPad (Figure 21). En effet, que ce soit sur Chrome ou sur Safari, aucun son ne sortait sur l'iPad lorsqu'on lançait l'application. Par contre, lorsqu'on tapotait un des éléments interactifs de la scène, on pouvait entendre la voix off.

Nous avons donc passé du temps sur Google à la recherche de solutions dans différents forums et sites d'aides et tutoriels sur l'utilisation de l'audio dans HTML5. Nous avons fini par trouver que le problème ne venait pas de l'implémentation mais de la façon dont l'iPad gère les sons. En effet, par défaut, tous les sons sont bloqués et il faut que chaque son soit déclenché par une interaction de la part de l'utilisateur. Cette limitation est prévue pour éviter de surcharger la bande passante des supports mobiles qui parfois ont des bandes passantes limitées (ex. 3G).

Nous avons ensuite essayé le jeu sur une tablette Samsung Galaxy Tab, afin de voir s'il s'agissait d'un problème uniquement lié à l'iPad, étant donné que le site fonctionnait correctement sur le téléphone mobile Samsung. Malheureusement, il s'est avéré que sur la tablette Samsung que le son n'était pas du tout géré et ce même avec une interaction de la part de l'utilisateur.

Ce premier POC fut donc un échec, mais nous permit d'en apprendre un peu plus sur les limitations des plateformes mobiles en ce qui concerne la gestion des sons.

3.5.2.2 Version 2 : Kinetic JS et Buzz JS

Une fois les limitations des supports mobiles connues, nous avons pensé à utiliser une librairie pour la gestion du son. En effet, les bibliothèques JavaScript disponibles sur internet, sont souvent prévues pour avoir des systèmes de « fallback » en cas de problème avec un support. Un fallback est une solution de contournement.

Après plusieurs tests infructueux sur la tablette Android, nous avons fini par trouver une petite librairie nommée [BuzzJS](#) qui faisait fonctionner le son sur les 4 plateformes testées. Nous avons donc refait le POC initial en intégrant la librairie BuzzJS.

Afin de contourner le problème du son déclenché lors d'une interaction utilisateurs, nous avons utilisé une nouvelle astuce. Il s'agissait d'ajouter un bouton pour lancer le jeu, et en cliquant sur ce bouton, le JavaScript lançait tous les sons du niveau, et les stoppait directement avant même qu'ils aient le temps d'être joués par le navigateur. Ceci faisait croire à la tablette que les sons étaient déclenchés par l'utilisateur, et par la suite il suffisait de les relancer au besoin.

Cette astuce s'est avérée fonctionnelle sur la tablette Android ainsi que sur le smartphone. Par contre sur l'iPad ce n'était malheureusement toujours pas le cas.

| PC Chrome | Samsung Galaxy S2 | iPad | Samsung Galaxy Tab |
|---|---|--|--|
|  |  |  même avec l'astuce, toujours pas |  Astuce de pré-play de tous les sons |

Figure 22 : Résultats du 2ème POC

Au vu du résultat de ce deuxième POC (Figure 22), nous avons décidé de chercher de nouvelles bibliothèques.

3.5.2.3 Version 3 : CreateJS

Pour le 3^{ème} essai, nous avons orienté nos recherches vers des « HTML5 Game Engines », c'est-à-dire, des API JavaScript orientées vers le développement de jeux. Nous en avons essayé plusieurs : Phaser, CocosJS, CAAT, etc. Mais toutes ces API étaient assez complexes à mettre en place et surtout très peu documentées. De plus, il était difficile de savoir s'il s'agissait là d'APIs qui allaient continuer à évoluer ou simplement être abandonnées, comme c'est parfois le cas dans les bibliothèques gratuites.

Cette problématique de choix s'est posée à plusieurs reprises durant la réalisation du POC. En effet, il est très difficile de pencher pour une technologie ou une autre, sachant qu'il y en a des centaines à disposition sur Internet. Notre contrainte était de partir sur une API gratuite, mais comment savoir s'il s'agit d'une librairie sérieuse et pas destinée à disparaître ?

Au final nous avons fait notre choix en nous basant sur le son uniquement. N'ayant pas réussi à faire fonctionner le son avec ces différentes bibliothèques essayées jusque là, nous nous sommes tournés vers [CreateJS](#). Cette librairie, développée sous la licence MIT (Massachusetts Institute of Technology) avait

parmi ses tutoriels un exemple d'approche qui permet de lancer des sons sur les plateformes mobiles à l'aide d'un système de proxy. En effet, dans cette librairie, il existe une manière de faire croire aux tablettes que le jeu se déroule entièrement dans le premier clic fait dans notre page HTML. De plus, la librairie proposait une documentation très complète avec une série de démonstrations et des forums de support très actifs. Ceci nous a convaincu qu'il s'agissait d'une librairie sérieuse.

Ainsi, en suivant le tutoriel proposé sur leur site, nous avons réussi à faire fonctionner le son sur toutes les plateformes. Une fois le son fonctionnel, nous avons refait le POC et avons cette fois obtenu une version qui fonctionnait correctement sur tous les supports.

| PC Chrome | Samsung Galaxy S2 | iPad | Samsung Galaxy Tab |
|---|---|---|--|
|  |  |  |  |

Figure 23 : Résultats du 3ème POC

Suite au succès de ce POC (Figure 23), nous avons conclu que le choix technique se porterait vers la librairie CreateJS.

3.6 Fonctionnalités développées

A la suite du POC, nous avons commencé le développement complet du logiciel. Ce développement s'est déroulé en plusieurs itérations, qui nous ont permis de développer des petites fonctionnalités, de les tester puis d'y revenir si nécessaire pour les améliorer et ainsi de suite jusqu'à obtention d'une solution satisfaisante.

3.6.1 Phase 1 : Adaptation du POC au monde mobile et tactile

Dans un premier temps, nous avons repris le POC et adapté le niveau déjà développé aux nouvelles exigences du logiciel. Nous avons réduit le nombre d'éléments interactifs disponibles, et avons replacé les différents boutons en tenant compte des recommandations pour les écrans tactiles.

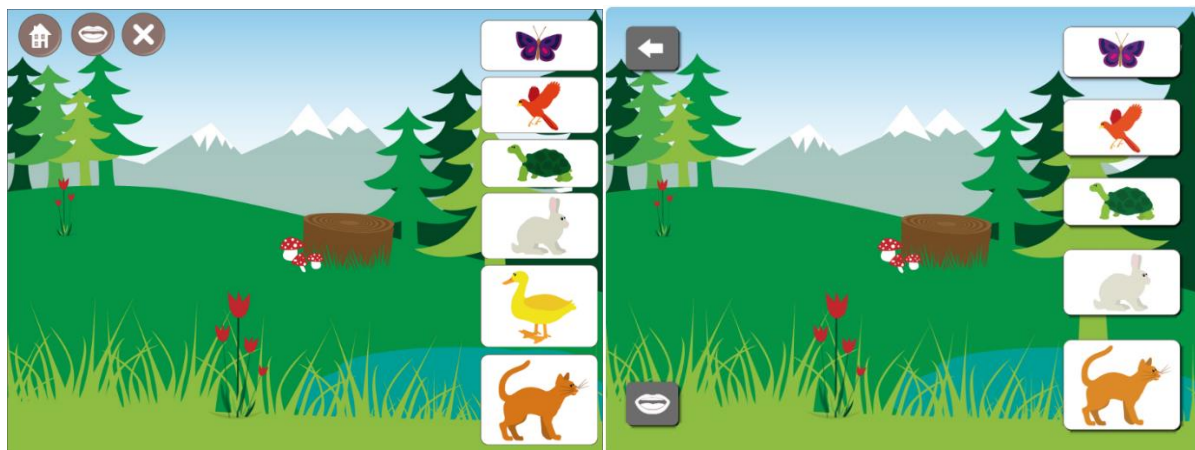


Figure 24 : Comparaison Animaux, exercice 1

Dans l'exemple du premier niveau (Figure 24), nous pouvons voir à gauche la version originale et à droite la nouvelle version.

Dans la nouvelle version, nous avons limité le nombre d'éléments interactifs à 5 par scène afin de pouvoir ajouter plus d'espace entre chaque élément et ainsi éviter de toucher accidentellement deux éléments à la fois. De plus, les éléments interactifs ont été éloignés des bords afin d'éviter un toucher accidentel par le pouce dans le cas où la tablette est tenue dans les mains.

Nous avons également repositionné les boutons de navigation.

Le bouton de retour en arrière a été ajouté en haut à gauche de l'écran. Il a été mis dans une zone difficilement atteignable pour éviter d'être touché par accident. De plus, ce bouton revient seulement d'un pas à l'arrière, c'est-à-dire à l'écran de navigation de deuxième niveau (cf. Figure 11) et non au menu principal comme c'était le cas dans le logiciel d'origine. Nous avons suivi ici les recommandations générales d'utilisabilité de Nielsen qui spécifie qu'il faut toujours permettre à l'utilisateur de revenir en arrière. Ce même bouton est disponible dans l'écran de sous-navigation ce qui permet, le cas échéant, de revenir au menu principal.

Le bouton qui permet de répéter un son a également été repositionné. En effet, nous avons supposé qu'il s'agit là d'un bouton qui risque d'être utilisé de manière plus régulière, nous avons donc décidé de le placer dans une zone plus facile à atteindre en tenant compte des recommandations des zones atteignables sur tablette (cf. Figure 6).

Le bouton qui permet de fermer l'application n'a pas été implémenté. En effet, s'agissant d'une application qui va être accédée par un navigateur web, le bouton de fermeture du navigateur servira également de fermeture pour l'application. De plus, d'après nos observations sur les applications usuelles disponibles sur les tablettes tactiles, celles-ci n'ont pas de bouton exprès pour quitter. En effet, le principe de quitter une application sur les plateformes mobiles semble obsolète. Nous avons tendance à passer d'une application à l'autre sans forcément quitter. Nous avons donc décidé de garder ce comportement dans notre application également.

Une fois ces adaptations faites, nous avons fait en sorte que notre code puisse être industrialisable. Il était important qu'une fois un niveau développé, le code utilisé pour ce niveau puisse servir à générer d'autres niveaux sans que des modifications de code ne soient nécessaires. Nous souhaitons pouvoir simplement ajouter les données du niveau (sons et images) et pouvoir ainsi en créer d'autres sans avoir à toucher à la logique déjà existante. Ceci était d'autant plus important que ce type d'exercices est l'un des plus utilisés dans le logiciel. En effet, nous avons compté 28 exercices de cette catégorie (voir catégorie 1.1 du tableau chapitre 3.2).

Nous avons pu valider cette industrialisation en ajoutant d'ores et déjà 3 exercices du domaine Animaux à ce stade du développement. Une explication étape par étape pour l'ajout d'exercices est disponible en annexe.

3.6.2 Phase 2 : Réalisation du passage d'un niveau à un autre

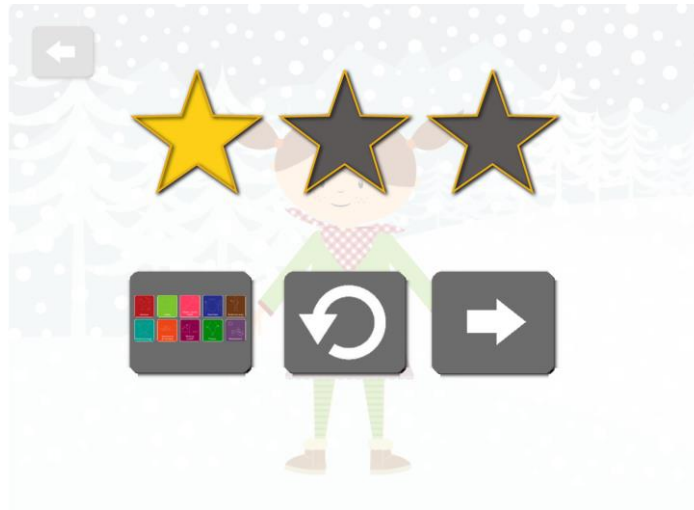


Figure 25 : Ecran de fin de niveau

Dans un deuxième temps, nous nous sommes concentrés sur la gestion de fin de niveau et avons décidé d'y inclure le score réalisé ainsi que la possibilité de naviguer directement vers le niveau suivant (Figure 25), de recommencer le niveau déjà effectué ou de revenir vers le menu principal.

Dans le logiciel initial, une fois le niveau terminé, une animation faisait disparaître le décor du niveau joué et l'utilisateur se retrouvait à nouveau dans l'écran de navigation des sous-menus. Nous avons ici souhaité montrer le score réalisé à la fin du niveau grâce au système d'étoiles afin de répondre à l'objectif de suivi de progrès qui a été fixé au début du projet.

Etant donné l'introduction de cet écran intermédiaire du score, nous avons rajouté 3 boutons, respectivement, un bouton qui permet de revenir au menu principal, un bouton qui permet de refaire le niveau si l'on n'est pas satisfait du score et un bouton qui permet d'aller au niveau suivant du même domaine. Ce dernier n'apparaît pas si l'enfant se trouve dans le dernier exercice du domaine.

Grâce à ce mécanisme, nous avons pu ajouter d'autres niveaux dans le domaine Animaux et tester les enchaînements avant de nous concentrer sur la phase suivante.

3.6.3 Phase 3 : Réalisation d'un exercice de catégorie libre

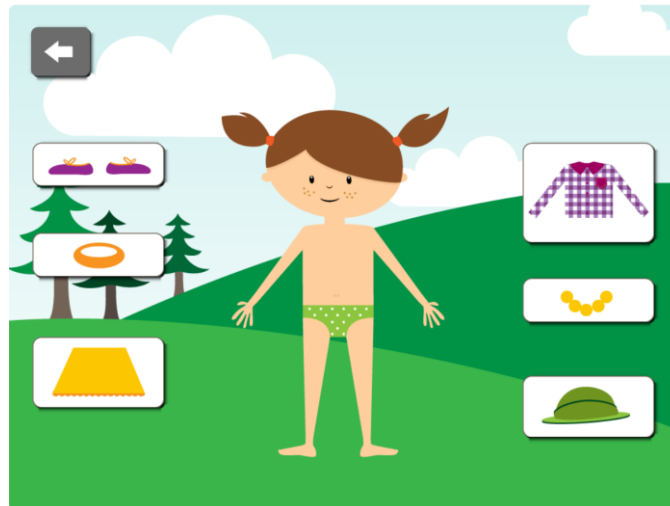


Figure 26 : Exemple d'exercice de type libre

Dans cette phase nous avons souhaité réaliser un exercice de catégorie libre, comme c'est le cas de certains exercices du domaine Habits (Figure 26). Dans ces exercices l'enfant ne doit pas suivre la voix, mais peut choisir les éléments avec lesquels il veut interagir.

Nous avons ici remplacé le survol de la souris par une interaction réagissant au toucher. Dans le logiciel original, le son correspondant à l'élément interactif se déclenchait au survol de la souris, cet événement n'existant plus sur les supports tactiles, nous avons fait en sorte que le son soit lancé au moment où l'élément est touché. Puis une fois l'élément tiré et déposé sur la scène, la voix off donne la confirmation à l'enfant.

Dans ces niveaux, il s'est avéré que la gestion du score n'était pas tout à fait appropriée. En effet l'enfant ne peut pas réellement faire d'erreur puisqu'il choisit lui-même les habits qu'il met. Les erreurs qu'il peut faire c'est par exemple de mettre des chaussures sur la tête. Or il ne s'agit pas là d'erreurs de français. Cela étant, nous avons conservé le mécanisme de score et avons enlevé des points dans le cas où l'enfant plaçait un élément à un endroit qui n'est pas logique.

Ce niveau a également été développé de manière à pouvoir être industrialisé. Ce qui fait que nous avons pu ajouter 2 autres niveaux de ce type sans modifier le code, en ajoutant simplement des données de référence.

Le développement des fonctionnalités de drag'n'drop s'est avéré bien plus difficile qu'il n'y paraît. En effet la gestion de ce type d'interaction, surtout de manière précise comme c'est le cas dans notre logiciel, est assez compliquée dans l'élément Canvas de HTML5. Le développement nous a posé beaucoup de problèmes, et bien que cela ait fini par fonctionner, des petits bugs subsistent.

3.6.4 Phase 4 : Réalisation d'un niveau de type guidé avec éléments sur la scène

La phase suivante consistait à réaliser un autre type d'exercices. Il s'agissait des exercices de catégorie 1.3 qui ont été comptés au nombre de 20 dans le logiciel. (cf. tableau chapitre 3.2)



Figure 27 : Exemple d'exercices du domaine Objets

Ce type d'exercice (Figure 27) étant assez nombreux, il faisait partie de nos priorités de développement. De plus, le troisième domaine (Objets) par ordre de difficultés du logiciel utilise ce type d'exercices.

Dans ce type d'exercices, une différence par rapport aux 2 autres types se présentait. Les éléments interactifs étant déjà sur la scène, une fois qu'ils ont été touchés, ils disparaissent, alors que dans les exercices précédemment développés, les éléments disparaissaient des bords de la scène pour réapparaître sous une autre forme dans la scène.

Cette différence a pu être traitée de manière industrialisable également. Ce qui veut dire qu'une fois qu'un niveau a été créé, nous avons pu en rajouter d'autres sans toucher au code, mais uniquement en ajoutant les sons et images nécessaires à ce niveau.

3.6.5 Phase 5 : Amélioration de la diffusion du contenu

A ce stade du développement, nous avons déjà 9 exercices développés, 3 pour chacun des domaines : Animaux, Habits et Objets.

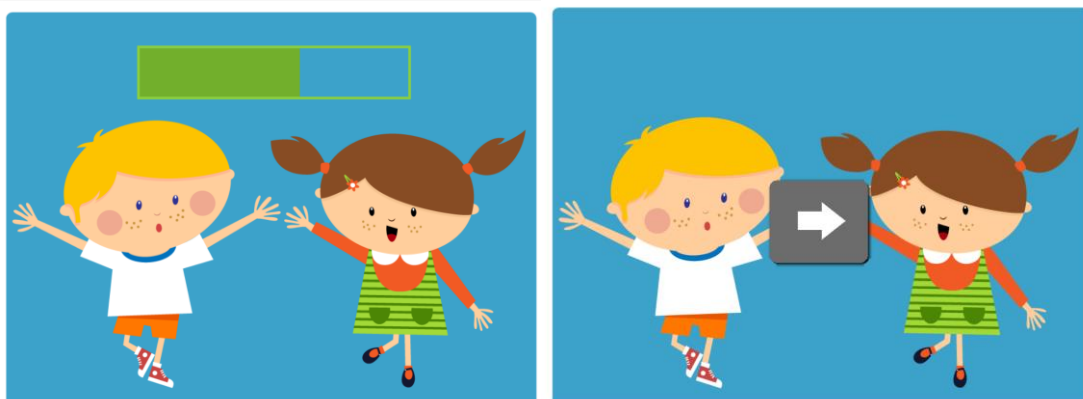


Figure 28 : Ecran de chargement

Afin de pouvoir utiliser les fichiers audio et les images dans le logiciel, il faut d'abord charger ceux-ci dans le navigateur (Figure 28). Ce chargement se faisait à ce stade au lancement du jeu. Dès le début,

Le navigateur chargeait tous les fichiers nécessaires à l'exécution des 9 exercices disponibles dans le logiciel. A ce stade, avec seulement 9 exercices développés, lorsque nous testions l'application avec une connexion de type 3G, le chargement prenait plus de 20 secondes. Nous avons donc décidé de procéder à une refonte du chargement des médias pour améliorer les performances.

Nous avons donc changé la manière de charger les médias de façon à ce que durant le premier écran de lancement, le navigateur ne charge que les médias nécessaires à la navigation en affichant une barre de progression. Ceci a nettement amélioré le temps d'attente du premier chargement.

Nous avons ensuite procédé à un chargement de quelques niveaux en fond de tâche en parallèle au chargement des médias de navigation mais sans afficher de barre de progression. Ceci nous permettait d'avoir déjà quelques niveaux pré-chargés au moment où l'enfant arrive dans le jeu. Ne sachant pas par quel niveau l'enfant commencera puisqu'il n'y a pas d'ordre à respecter, nous avons décidé de charger un niveau sur 4 en mémoire et de gérer le chargement des autres au fur et à mesure que l'enfant les sélectionne.

Si l'enfant choisit un exercice qui n'est pas chargé, une barre de chargement va apparaître. L'exercice qui suit directement l'exercice en cours sera chargé durant l'exécution de celui-ci afin d'éviter à l'enfant d'attendre deux fois de suite. Une fois qu'un exercice est chargé, si l'on revient dessus il ne sera plus rechargé.

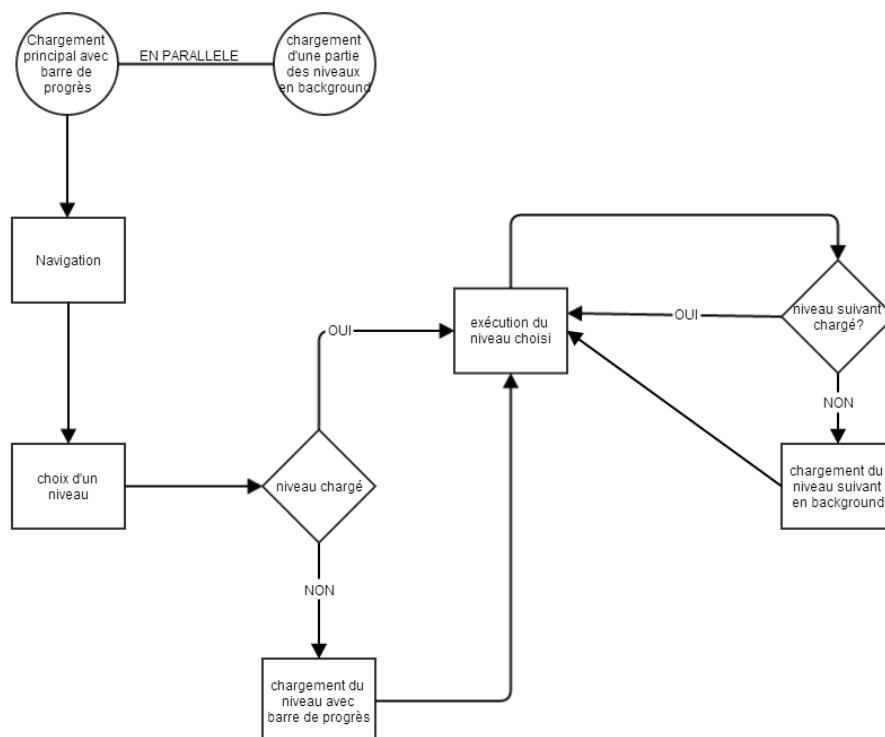


Figure 29 : Diagramme de chargement des médias

Le processus de chargement est représenté graphiquement dans la Figure 29 ci-dessous. Cette refonte a permis une nette amélioration du temps d'attente au début du jeu, mais a ajouté des temps d'attente entre certains exercices. Toutefois, les temps d'attente pour les chargements inter-exercices sont minimes.

3.6.6 Phase 6 : Nouveau type d'exercice

Dans cette dernière phase, nous avons ajouté un autre type d'exercices. Il s'agit d'exercices guidées par la voix off mais où l'enfant doit effectuer un drag'n'drop. Ce type d'interaction est notamment présent dans certains niveaux du domaine Habits ainsi que dans le domaine des Adjectifs. Nous avons opté pour ce choix car la difficulté technique de ce développement nous semblait accessible étant donné que nous avons déjà mis en place des niveaux guidés par la voix et des niveaux avec drag'n'drop, ce qui fait que l'on pouvait combiner les deux pour obtenir de nouveaux types d'exercices.

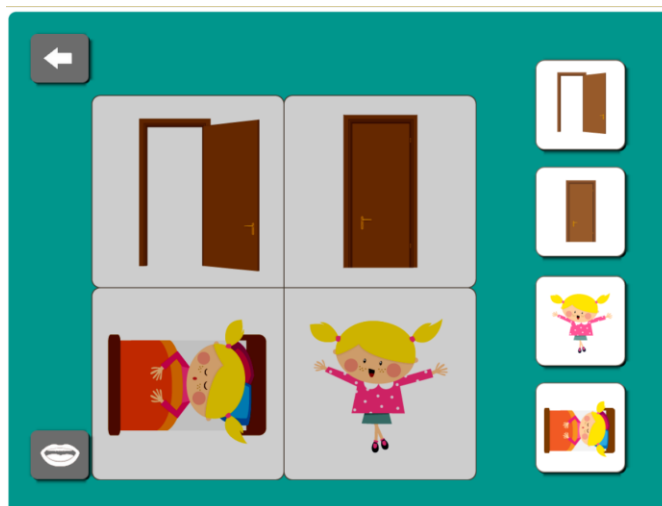


Figure 30 : Exemple d'un exercice des Adjectifs

Ceci a engendré la création d'une nouvelle classe dans notre code et a permis aussi de mettre en place un nouveau domaine, celui des Adjectifs (Figure 30). Bien que ce domaine se situe dans les plus difficiles du jeu initial, nous avons décidé de le développer pour son accessibilité du point de vue technique.

3.6.7 Stockage local

Dans nos objectifs pour la refonte du logiciel, nous avons aussi décidé de stocker les médias dans l'appareil de l'utilisateur afin d'éviter des chargements à chaque exécution et également afin de permettre au joueur de jouer sans aucune connexion internet.

Cette fonctionnalité n'a malheureusement pas pu être réalisée dans son ensemble. En effet, nous avons dans un premier temps stocké les scores dans l'appareil de l'utilisateur, ce qui fait qu'à chaque connexion celui-ci voit ses scores sans qu'ils soient réinitialisés à chaque fois qu'il quitte le jeu. Ceci s'est fait simplement à l'aide de l'API Web Storage (cf. 3.3.4).

Cette API présentait toutefois une certaine limite, puisqu'elle permet de stocker uniquement des chaînes de caractères. Cela ne suffisait donc pas pour le stockage de nos médias. Nous avons donc cherché une autre solution et avons trouvé l'option de stocker nos médias, ainsi que nos fichiers JavaScript dans le cache du navigateur à l'aide de la nouvelle fonctionnalité HTML5 Application Cache.

Cette fonctionnalité permet de lister dans un fichier cache tous les fichiers nécessaires à l'exécution de notre logiciel et d'informer le navigateur qu'il doit les stocker en local afin de pouvoir par la suite y accéder sans connexion à internet. Cette solution est très simple à implémenter et nous avons donc entrepris sa réalisation.

Malheureusement, au cours du développement, il s'est avéré que la librairie choisie pour notre logiciel, CreateJS, ne supportait pas encore le travail avec le cache. En effet, malgré le stockage des médias en local, la librairie cherchait à télécharger le contenu directement depuis le serveur. Nous avons pu confirmer l'incompatibilité de CreateJS et de l'Application Cache directement sur les forums de développement de CreateJS.

Cette limite a fait que nous n'avons pas pu avoir un logiciel qui puisse être accessible hors ligne. Cependant, la librairie CreateJS étant en constante évolution, nous pensons que cette incompatibilité pourra un jour être corrigée et que nous pourrons alors réactiver les options de cache dans le logiciel.

3.6.8 Structure du code (cf. Annexe 2 : code source du logiciel)

Comme pour toute application HTML, nous avons principalement besoin de 3 types de ressources :

- Page HTML : dans notre cas nous en avons qu'une, index.html. Ceci est la page de lancement du projet qui contient l'inclusion de tous les scripts nécessaires
- Scripts JavaScript (.js) : ce sont les fichiers qui contiennent toute la logique de notre application développée dans le langage JavaScript qui est un langage exécuté du côté du client, c'est-à-dire par le navigateur web.
- Fichiers média : c'est-à-dire les fichiers son et les images nécessaires à notre logiciel

Nous avons essayé de structurer notre code en nous inspirant du modèle objet, bien que celui-ci soit géré de manière différente en JavaScript que dans d'autres langages orientés objet.

Nous avons donc découpé nos classes de manière à avoir des objets qui peuvent être instanciés autant de fois que nécessaire, mais aussi pour encapsuler notre code et ne pas avoir des fonctions dispersées partout sans connaître les dépendances des uns des autres. Ce découpage en objet était surtout nécessaire pour la gestion des exercices. Etant donné qu'un certain nombre de comportements était similaire entre les différents exercices il était important de pouvoir réutiliser les fonctions déjà développées et d'éviter ainsi de faire du copier-coller.

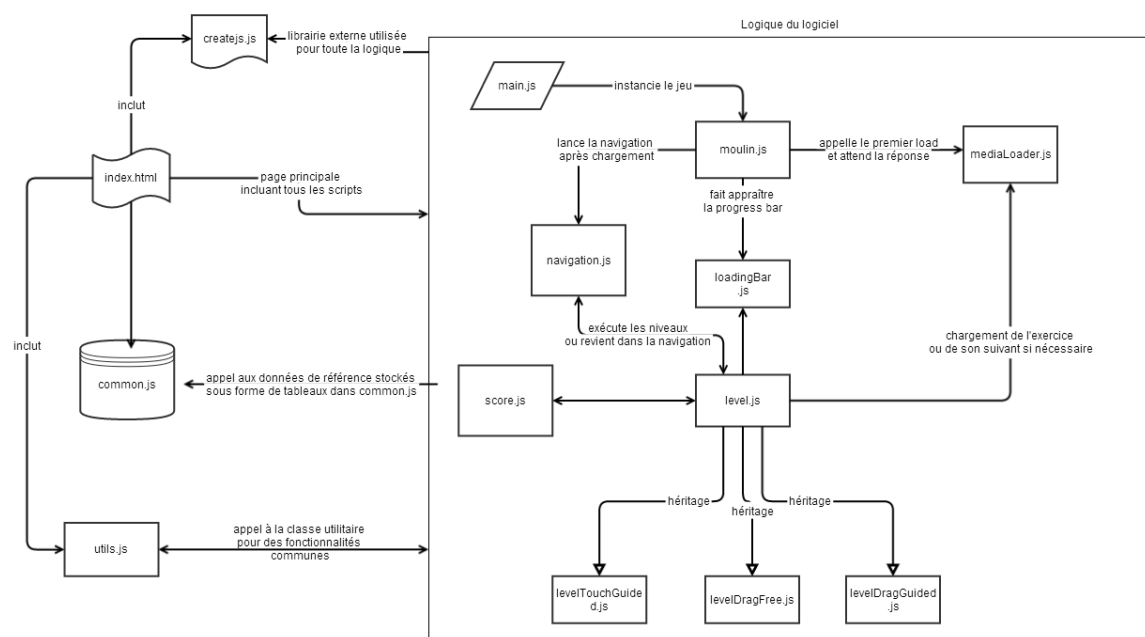


Figure 31 : Structure du code

Une fonctionnalité très utile du navigateur Safari nous a permis d'avoir une meilleure gestion du jeu sur les plateformes Apple, que ce soit l'iPad ou l'iPhone. En effet, en ajoutant seulement 3 lignes de code dans notre page index.html, nous avons pu faire en sorte que lorsque l'utilisateur ajoute un raccourci sur le bureau vers le site depuis le navigateur Safari, une icône apparaisse sur le bureau comme si il s'agissait d'une véritable App (Figure 32).

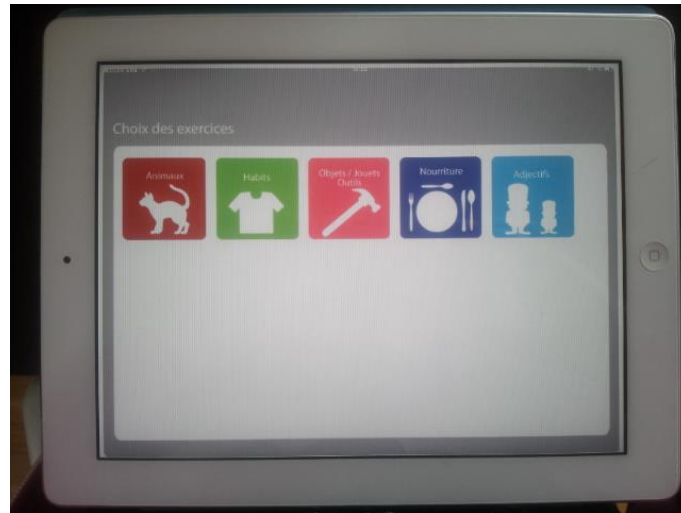


Figure 33 : Jeu en plein écran sur iPad

De plus, grâce à ce mécanisme de raccourci, le jeu est lancé en plein écran et on ne voit même plus que l'on se trouve dans un navigateur (Figure 33). Cette fonctionnalité rend le logiciel d'autant plus ergonomique. Malheureusement elle n'existe que sous Safari, les autres plateformes doivent passer par le navigateur pour accéder au jeu.

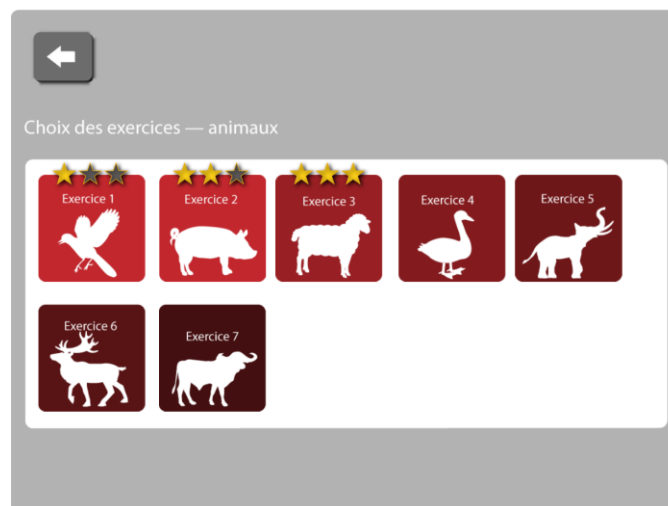


Figure 34 : Suivi de progrès

Le jeu final permet également une gestion du suivi du progrès de l'enfant. Le suivi est sauvegardé et repris d'une connexion à l'autre. Le système de suivi est basé sur un certain nombre d'étoiles :

- Aucune étoile si le niveau n'a jamais été terminé
- 1 étoile si le niveau a été terminé, avec plus d'une erreur
- 2 étoiles si le niveau a été terminé avec une erreur

- 3 étoiles si le niveau a été terminé sans erreurs

Le score est affiché au-dessus de chaque exercice (Figure 34).

Les animations de transition entre les éléments interactifs ont été adaptées au monde HTML5. En effet il est plus difficile de créer des animations de type « Flash » en JavaScript. C'est pourquoi toutes les animations ont été remplacées par un effet de fondu. Les éléments apparaissent et disparaissent doucement de la scène.

3.6.9.1 Pistes d'amélioration

Durant le développement, nous étions seuls à prendre toutes les décisions, ce qui ne correspond pas à une situation courante d'un développement de projet informatique. En effet, dans une équipe de développement, il y a souvent un architecte qui sert de référence pour les décisions de type architecture de code, un responsable d'ergonomie qui décide des types d'interaction, un responsable de base de données qui met en place la structure nécessaire pour gérer les données de l'application, et puis aussi un commanditaire qui décide des fonctionnalités à développer en priorité.

Dans le cadre de ce travail, toutes les décisions ont été prises par une seule personne ce qui peut parfois biaiser le résultat dû à un manque de recul. Maintenant que la phase de développement est terminée nous voyons des points d'améliorations du code qui devraient être prises en compte lors d'un prochain développement.

En ce qui concerne la décomposition des données dans le fichier `common.js`, nous avons opté pour une décomposition stricte par niveau. Or, au fur et à mesure de l'avancement il s'est avéré qu'il y avait plusieurs éléments communs aux niveaux du même domaine, tels que par exemple la scène de base, les consignes en début d'exercice et les consignes de fin. Ces fichiers auraient pu être regroupés et chargés qu'une seule fois pour tous les exercices. Malheureusement, dans la façon dont nous avons conçu le code, ce changement aurait nécessité une refonte totale du projet qui était alors risquée à entreprendre.

De plus, au fur et à mesure du développement, pour corriger certains bugs, le code a parfois été modifié rapidement sans forcément respecter une architecture claire et précise. Nous pensions prendre le temps de faire un « refactoring » pour ces petites corrections mais finalement, nous n'avons pas pu le faire par manque de temps. Il serait donc judicieux de consulter un architecte pour voir quelles sont les améliorations possibles de l'architecture actuelle du projet.

4 Tests d'utilisabilité

4.1 Objectifs

Afin de valider les interactions proposées dans les différents mini-jeux des tests d'utilisabilité ont été prévus auprès d'enfants non lecteurs de 2 à 5 ans.

Le but des tests était de voir si l'enfant est capable de naviguer dans le jeu par lui-même et de comprendre les différentes instructions et les types d'interaction d'un exercice à l'autre.

4.2 Public cible

Le logiciel est destiné à aider les enfants de 2 à 5 ans non-francophones et non-lecteurs. Cependant, le but des tests portant sur l'utilisabilité du logiciel et non sur son apport pédagogique, nous avons élargi le public cible aux enfants de 2 à 5 ans qui sont de langue maternelle française également.

Le test a été effectué auprès de 5 enfants. Selon Nielsen (1990) ainsi que Nielsen & Landauer (1993), cinq testeurs suffisent déjà à identifier 85% problèmes d'utilisabilité qui touchent 31% d'utilisateurs. Nous avons choisi de nous limiter à ce chiffre dans le cadre de ce mémoire.

Les enfants (ci-dessous avec leurs prénoms réels) avaient le profil suivant :

- Gaïa, fille de 5 ans, parlant français et polonais
- Elena, fille de 4 ans, parlant français et italien
- Lévana, fille de 3 ans et demi, parlant français
- Lyam, garçon de 2 ans et 9 mois, parlant français
- Thomas, garçon de 2 ans et 4 mois, parlant français et polonais

Les 5 enfants avaient tous déjà joué à des mini-jeux sur tablette ou sur le téléphone portable de leurs parents.

4.3 Méthode

La méthode utilisée est l'observation. En effet s'agissant d'un jeu libre, il est difficile de donner une tâche à réaliser à l'enfant, comme on peut le faire lors de tests d'utilisabilité d'un site internet ou d'un logiciel de gestion, où l'on demanderait précisément à un utilisateur de trouver une information ou une fonction. Dans le cas de notre jeu, nous avons préféré laisser à l'enfant la liberté de jouer, tout simplement, sans contrainte. Notre démarche s'inscrit dans une méthode qualitative, basée sur des observations de comportements et d'attitudes et non sur le nombre d'actions ou le temps passé.



Figure 35 : Usability tests, Thomas, 2 ans

Il s'agissait donc de donner une tablette iPad à l'enfant (Figure 35) avec le jeu déjà ouvert et de le laisser faire. Il n'a pas été demandé à l'enfant de verbaliser ses gestes, car c'est un peu difficile à cet âge-là d'avoir ce recul. Nous n'avons pas non plus effectué d'entretien post test car il est difficile de demander à un enfant comment il s'est senti durant le test, ce qu'il a aimé, etc. En effet, les enfants ont tendance à vouloir faire plaisir aux adultes, de plus, ils peinent à exprimer ce qu'ils ont apprécié ou non (Hanna, Ridsen, & Alexander, 1997).

Les tests ont été effectués séparément avec chaque enfant, à chaque fois à la maison de l'enfant ou du testeur, avec la présence d'un de leurs parents. Les parents ont été rassurés de manière à comprendre que le but du test n'est pas de tester les capacités de leurs enfants.

Durant le test, il s'agissait d'observer l'enfant et de noter les éventuels problèmes rencontrés. Lorsque l'enfant avait des difficultés pour comprendre quelque chose, il recevait une explication sur le jeu de la part de son parent qui était à ses côtés.

Le temps imparti pour les tests était d'environ 10 à 15 minutes par enfant, en fonction de leur intérêt et capacité de concentration. A la fin de chaque test l'enfant était libre de continuer à jouer s'il le souhaitait sans que nous l'observions, ce qui fût le cas la plupart du temps. Notons qu'à ce stade le jeu comportait une quinzaine d'exercices seulement.

4.4 Résultats

4.4.1 Observations générales

En observant les enfants il était intéressant de constater qu'ils utilisent aléatoirement les doigts pour interagir avec l'écran. En effet d'un exercice à l'autre, voir même d'un « press » à l'autre dans un même exercice, les enfants utilisaient soit l'index soit le majeur, parfois le pouce, parfois même, ils pinçaient le pouce et l'index pour appuyer sur l'écran.



Figure 36 : Usability tests, Gaïa, 5 ans

De plus, lorsqu'ils effectuaient les différents exercices, les enfants faisaient attention à ne pas se tromper. Par exemple, lorsqu'il fallait habiller Lili dans les exercices sur les Habits (Figure 36), ils mettaient les habits dans le bon ordre (d'abord les chaussettes, ensuite les baskets) et au bon endroit. L'application a cependant prévu un mécanisme au cas où l'enfant essaie de mettre la casquette sur les pieds afin de prévenir les erreurs. Il en allait de même dans les exercices guidés par la voix, tous les enfants faisaient très attention de choisir le bon animal ou le bon objet lorsque la voix le demandait. Les plus petits répétaient après la voix off comme pour confirmer que c'est bien ce qu'ils croient. Il aurait certainement été plus intéressant de tester la gestion d'erreurs avec des enfants non-francophones, plus sujets à se tromper.

Nous avons aussi observé un certain plaisir des enfants à jouer. En effet, une fois le jeu commencé, les enfants ne s'en lassaient pas et voulaient constamment recommencer, et ce même lorsqu'ils avaient réalisé tous les exercices disponibles (15 exercices dans la première version testée), cela ne les embêtait pas de refaire les mêmes.

En ce qui concerne les étoiles affichées en fin de niveau ainsi que sur les différents exercices, nous avons constaté que les enfants n'y avaient aucun intérêt. Cela ne les intéressait pas de savoir si ils avaient eu un bon score ou non, ou si ils avaient déjà fait tel ou tel niveau, ils voulaient simplement jouer. Nous pouvons en conclure que le système des étoiles sera surtout utile pour l'enseignant ou le parent. Cependant il serait intéressant de surveiller l'intérêt des étoiles sur le long terme, une fois que l'enfant a commencé à jouer au jeu de manière régulière.

Enfin pour la petite anecdote, le plus jeune des enfants, Thomas, bien qu'il ait déjà interagi avec les smartphone de ses parents a eu une réaction très drôle lorsque l'iPad s'est mis à parler. Il ne s'attendait pas à ce que dans le logiciel, il y ait quelqu'un qui lui adresse la parole. Il a d'abord esquissé un sourire de surprise jusqu'aux oreilles, puis sous l'impulsion a reculé de la table, et a commencé à regarder sur le côté de l'iPad pour voir d'où venait la voix, puis s'en est suivi un petit fou-rire avant de commencer à jouer.

4.4.2 Résultats détaillés

Les résultats sont plutôt satisfaisants. Les enfants ont bien compris comment fonctionne le logiciel, cependant quelques problèmes ont été rencontrés.

4.4.2.1 *Observation 1 : Le menu interactif de fin de niveau n'a pas été compris dès la première utilisation*

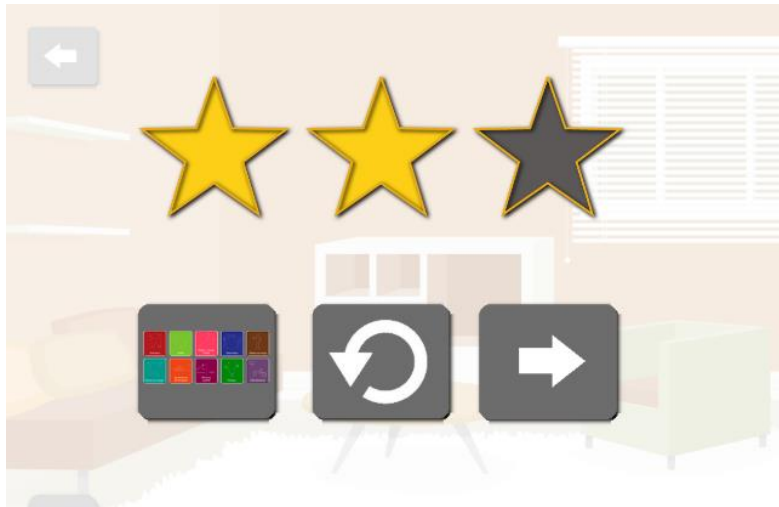


Figure 37 : Menu de fin de niveau

En effet, 2 des 3 boutons présents sur la Figure 37, n'étaient pas clairs pour les enfants. Le bouton qui permet de revenir au menu principal était clair et intuitivement utilisé même par les 2 enfants de moins de 3 ans. Cependant, les 2 autres boutons ont dû être expliqués par le parent. Selon les critères ergonomiques de Bastien & Scapin (1993), ce problème serait un problème de signifiante des codes et dénominations. Il n'était pas clair pour les enfants comment continuer le jeu.

Pour 3 des 5 enfants, ce problème est apparu uniquement lors de la première réalisation d'un niveau. Une fois que les parents expliquaient à leur enfant de quoi il s'agissait, l'enfant utilisait les menus de manière intuitive lors des essais suivants. Notons que les parents ont compris le sens des trois icônes sans que nous ayons à intervenir. Notons également, que les 3 enfants de plus de 3 ans faisaient par la suite appel aux 3 boutons au gré de leurs envies, alors que les 2 enfants de moins de 3 ans avaient tendance à utiliser toujours le retour vers la navigation principale, comme si ce bouton-là était le seul qu'ils avaient vraiment assimilé.

4.4.2.2 *Observation 2 : Les enfants commencent à jouer avant que la voix off n'ait terminé de parler*

Lorsqu'un exercice est lancé, la voix off donne une introduction contextuelle au niveau et dit à l'enfant ce qu'il va devoir faire pour réaliser l'exercice. Par exemple : « Il fait froid, aide Lili à s'habiller. Choisis un habit et mets-le sur Lili ».

Les enfants avaient tendance à commencer à toucher les éléments de la scène avant que les instructions arrivent à la fin. Il s'agit ici d'un problème de guidage, car il n'était pas clair pour les enfants qu'ils doivent attendre la fin des instructions.



Figure 38 : Icône de haut-parleur

Afin de contrer ce problème, nous avons ajouté une icône de haut-parleur (Figure 38) qui apparaît à l'écran lorsque l'enfant essaie de toucher un élément alors qu'un son est en train de jouer. Cette icône apparaît puis disparaît pour fournir un feedback à l'enfant sur ce qui se passe. Ainsi, nous pensons l'aider à comprendre que l'application a répondu à son toucher mais qu'il doit attendre pour réaliser l'exercice.

Après un deuxième test avec un des enfants de 2 ans, cette icône a posé problème, il semblait que l'enfant pensait qu'il avait fait une erreur. Nous avons donc réfléchi et changé l'icône afin de la rendre plus claire pour un enfant qui n'a pas l'habitude des symboles courants du web, et avons décidé d'utiliser le dessin d'une oreille à la place. Nous n'avons pas pu par la suite tester l'efficacité de ce nouveau symbole. Dans le cas où celui-ci ne serait pas compris par les enfants, une autre solution devrait être envisagée, par exemple, la possibilité de flouter l'écran durant les instructions orales.

4.4.2.3 Observation 3 : Les enfants doivent sélectionner les petits éléments à plusieurs reprises



Figure 39 : Exemple d'objets trop petits pour les enfants

Dans les exercices contenant des petits objets, tels que les niveaux dans lesquels Lili et Theo doivent ranger la maison, les enfants ont eu des problèmes pour toucher les éléments tels que le verre, le biberon, le téléphone, car ces objets nécessitent une plus grande précision dans le toucher (Figure 39). Nous avons observé que l'enfant n'est pas très précis lorsqu'il touche un objet. Il a même tendance à glisser du doigt sur celui-ci, et donc, lorsqu'il lève le doigt il se trouve en dehors de la zone de l'objet.

Nous avons, à ce stade, déjà enlevé certains petits objets des exercices initiaux (fourchettes, lunettes, etc.) mais cela ne s'est pas avéré suffisant. Selon les recommandations retenues dans le cadre théorique (cf. 2.4), la taille minimale des éléments interactifs se situe entre 38 et 42 pixels. Cependant, dans notre test, même pour des éléments de taille suffisante selon ces recommandations (ex. biberon taille 42x75px), les enfants ont eu des problèmes pour les toucher. Nous pensons que la taille recommandée par Nielsen doit être recalculée pour s'adapter aux enfants puisque ceux-ci sont moins précis que les adultes.

Afin de résoudre ce problème, nous avons augmenté la taille cliquable des images concernées en rajoutant 25% de la taille initiale autour des images concernées, ce qui permet une marge d'erreur. Un test a été refait avec deux des cinq enfants. Les résultats se sont avérés tout à fait concluants, les enfants n'avaient pas besoin de toucher les petites images à plusieurs reprises pour réussir l'exercice.

4.4.2.4 Observation 4 : Les enfants n'ont pas tout de suite remarqué le bouton de répétition de son et de retour en arrière

Deux des enfants ont posé la question suivante alors qu'ils effectuaient un exercice sur les animaux pour la première fois : « Elle a dit quoi ? ». Ils avaient pas écouté la voix off et ne savaient donc pas sur quel animal cliquer, mais ne voulaient pas se tromper. Ils n'ont pas vu que le bouton avec la petite bouche était là pour répéter un son. Les parents leur ont expliqué qu'ils pouvaient cliquer sur ce bouton pour entendre un son une deuxième fois.

De plus, le testeur leur a fait remarquer l'existence du bouton de retour en arrière, qu'ils n'avaient pas non plus remarqué. Par la suite, le bouton de retour a été assimilé et leur a permis de naviguer dans les exercices et les sous-menus.

Une fois assimilé, ils appuyaient sur le bouton dans les niveaux suivants pour montrer qu'ils avaient compris et pour s'amuser.

4.4.2.5 Observation 5 : Le drag'n'drop pose problème aux plus petits

Dans le cas des trois enfants en dessous de 4 ans, nous avons observé des problèmes pour réaliser les exercices avec une interaction de type drag'n'drop. En effet les enfants ne comprennent pas qu'une fois un objet glissé sur la scène il suffit de le lâcher pour qu'il y reste. Ils essaient plusieurs fois de retaper l'objet comme si ils voulaient confirmer qu'ils veulent le lâcher.



Figure 40 : Lévana, 3 ans et demi devant un exercice du domaine Adjectif

De plus, ils ont tendance à ne pas vouloir passer au-dessus de la scène dans les endroits où l'on peut mettre les objets. Par exemple dans l'exercice du Lotto dans le domaine des Adjectifs, si il faut placer un élément en haut à gauche, par exemple le lapin (Figure 40), normalement on peut sélectionner cet élément puis le passer tout droit par-dessus les autres (ex. tortue) jusqu'à arriver à la cible. Cependant, les enfants évitent soigneusement de toucher l'élément se trouvant sur le passage et le contournent. Ceci rend l'exercice encore plus difficile puisqu'il faut traîner l'objet sur une distance plus longue. Il arrivait ainsi qu'ils lâchent l'élément par accident, et que celui-ci revienne à sa place. Certains essayaient même de le tirer avec un doigt de la main droite jusqu'à un certain point puis ils voulaient continuer l'action avec un doigt de la main gauche, ce qui n'était pas géré par le jeu.

Les deux enfants de plus de 4 ans avaient moins de problèmes et se servaient du drag'n'drop de manière plus intuitive. Cependant elles rencontraient également des problèmes de temps en temps, ces problèmes sont aussi dus au fait que le drag'n'drop tel que développé à l'aide du composant Canvas n'est pas très stable dans notre logiciel.

Pour essayer de régler ce problème, nous pourrions envisager une amélioration qui permettrait de déplacer les éléments avec le doigt à l'écran et les poser ailleurs sur la scène, afin que l'enfant puisse les reprendre et les déplacer ensuite plus loin. Actuellement, les éléments retournent à leur place si ils n'ont pas été placés à l'emplacement attendu, ce qui demande à l'enfant de recommencer l'opération difficile du drag'n'drop depuis le début à chaque fois. Le fait de pouvoir bouger l'élément jusqu'à un certain point permettrait également de le reprendre avec un autre doigt, comme certains des enfants avaient intuitivement essayé de le faire.

4.4.2.6 Observation 8 : Le changement d'interaction d'un exercice à l'autre est mal compris par les enfants

Lorsque les enfants passent d'un exercice de type Guidé par la voix avec comme instruction de toucher un élément pour le faire apparaître ou disparaître de la scène, vers un exercice dans lequel il y a du drag'n'drop ils essaient en premier de toucher les éléments sans les déplacer. Ceci est valable aussi dans l'autre sens, lorsqu'ils sortent d'un niveau drag'n'drop et doivent revenir dans un exercice où il suffit de toucher l'élément, ils essaient de le déplacer.

Les enfants ont des difficultés à savoir quand il faut utiliser quel type d'interaction, bien que dans les instructions données par la voix off du niveau quelques indications sont données. Une fois la première tentative faite, ils comprennent qu'il faut changer le mode d'interaction. Ce comportement a

également été observé auprès d'un public adulte, dont nous nous sommes servis pour faire des tests fonctionnels.

Plusieurs solutions seraient envisageables pour ce problème. On pourrait ajouter une introduction sous forme de vidéo (ex. Angry Birds) au début de chaque domaine afin de montrer à l'enfant ce qui est attendu de lui dans les exercices de ce domaine, ou alors améliorer les phrases d'instruction afin de mieux faire comprendre à l'enfant ce qui est attendu de sa part. Nous avons vu dans le cadre théorique que l'une des recommandations de Druin (2009) portait justement sur la clarté des instructions données aux enfants (cf. 2.3.2).

4.5 Discussion des résultats

Les résultats des tests ont mis en évidence qu'il faut tout de même un temps d'apprentissage pour utiliser le logiciel. En effet, s'agissant de jeunes enfants, ils n'ont pas encore assimilé le concept de boutons de navigation, de bouton suivant ou précédent, comme c'est le cas des adultes qui ont l'habitude de naviguer sur internet et d'avoir ce genre d'interactions sur leur ordinateur.

Cependant, la phase d'apprentissage est très rapide et les enfants comprennent très vite à quoi servent les boutons. Ils sont même fiers ensuite de montrer qu'ils savent les utiliser.

Suite à certaines corrections, notamment le changement de taille des éléments interactifs, des tests ont été refaits et nous ont permis de valider les améliorations ergonomiques effectuées. Nous avons également remarqué lors du deuxième test avec le plus grand des enfants, qu'elle avait compris qu'il fallait attendre la fin des instructions pour jouer. De plus, elle se rappelait également de l'utilité des boutons de navigation entre les niveaux et du bouton de répétition de son. C'était très impressionnant de voir qu'elle utilisait sans problème le bouton de retour en arrière lorsqu'elle arrivait dans un niveau qui ne lui plaisait pas forcément, et qu'elle naviguait ensuite parmi les domaines d'exercices, parmi les différents exercices en se rappelant même quelles étaient les nouveautés par rapport à la dernière fois où elle avait joué.

5 Conclusion

Notre recherche avait pour objectif d'étudier les contraintes à prendre en compte lorsqu'on transforme un logiciel à but pédagogique initialement prévu pour une utilisation sur un ordinateur fixe vers une utilisation plus mobile et disponible sur un support à interaction tactile.

Notre cadre théorique nous a apporté les recommandations nécessaires pour réaliser cette refonte et nous avons essayé de les appliquer dans le cadre du développement du logiciel ludo-éducatif « Le Moulin à Paroles ».

Le développement du logiciel en respectant les recommandations trouvées dans nos recherches a pu se faire partiellement. En effet, par manque de temps, nous n'avons pu réaliser qu'un tiers des exercices du logiciel initial. Ce manque de temps est en partie dû à la complexité technique due à la nécessité de pouvoir exécuter le logiciel sur des plateformes différentes. En effet, il a fallu faire plusieurs essais infructueux pour arriver enfin à une solution qui fonctionne sur toutes les plateformes visées par notre recherche, ce qui nous a fait perdre du temps précieux. Dans un deuxième temps, nous avons également perdu beaucoup de temps lors du développement des exercices avec une interaction de type drag'n'drop qui s'est avérée techniquement complexe.

De plus, un des quatre objectifs fixés pour le développement n'a pas pu être atteint. Il s'agit de l'objectif concernant la possibilité d'exécution sans accès à internet. En effet, afin de rendre le logiciel le plus mobile possible nous souhaitions avoir la possibilité de l'exécuter sans avoir accès à une connexion internet. Malheureusement, la librairie choisie pour développer le logiciel s'est avérée incompatible avec les possibilités de stockage de ressources hors ligne proposées par HTML5. Cet objectif n'a donc pas pu être atteint.

Toutefois, les trois autres objectifs ont été atteints :

- le logiciel est exécutable sur les supports mobiles et fixes
- les interactions ont été adaptées au tactile
- un suivi de progrès a été mis en place

Au vu des problèmes techniques rencontrés durant la phase de développement, nous nous sommes demandé s'il ne serait pas plus efficace de développer des applications natives pour chacune des plateformes proposées. En effet, les problématiques rencontrées, notamment avec la gestion du son, n'aurait pas eu lieu si nous avions décidé de nous concentrer sur une seule plateforme à la fois et à développer une application dite native pour cette plateforme. Les applications natives sont développées en Objective C pour les supports Apple, ou en Java pour les supports Android. Le fait que les applications soient développées en mode natif, enlève la plupart des limites que nous avons identifiées lors d'un développement en HTML5. Cependant, ce choix n'aurait pas été en accord avec notre volonté de rendre le logiciel disponible sur toutes les plateformes malgré leur évolution très rapide. En effet, les applications natives auraient certainement nécessité des mises à jour à chaque évolution du support, ce qui n'est pas le cas d'une application HTML5. De plus, de nouvelles plateformes mobiles pourraient apparaître sur le marché et donc nécessiter un développement natif supplémentaire, alors que dans le cas d'une application HTML5, elle devrait être fonctionnelle.

Afin de permettre son utilisation en domaine scolaire, le logiciel nécessiterait l'ajout de données d'encore quelques exercices afin de proposer une batterie plus large. En effet, bien que nous ayons vu

durant les tests d'utilisabilité que les enfants avaient du plaisir à refaire plusieurs fois les mêmes exercices, nous pensons que sur le long terme cela risque de les ennuyer. De plus, il serait intéressant d'ajouter un système de « traçage » sur ce que l'enfant fait réellement, par exemple, combien de fois il refait le même exercice et avec quels score afin de voir l'évolution de l'enfant et non seulement le résultat final.

Enfin, en ce qui concerne l'utilisabilité du logiciel, les résultats ont été plutôt satisfaisants malgré quelques problèmes rencontrés. Les enfants réagissaient de manière enthousiaste à l'idée de faire les exercices, et même si au début quelques aspects ne leur paraissaient pas très clairs, ils assimilent très vite toutes les fonctionnalités du logiciel. Nous avons remarqué une grande différence entre les enfants de plus de 3 ans et ceux de moins de 3 ans. En effet, les plus âgés se sentaient plus à l'aise dans le logiciel, avaient moins d'hésitations et naviguaient de manière intuitive. Pour ce qui est des enfants de moins de 3 ans, nous recommandons une utilisation avec accompagnement. De plus, pour ce qui est des interactions de type drag'n'drop, celles-ci ont posé des problèmes autant aux enfants qu'au développeur, nous ne sommes donc pas convaincus par leur présence dans le logiciel. Il serait peut-être judicieux de se contenter d'un seul type d'interaction (tap/long press) pour les enfants de cet âge afin d'éviter de les décourager et frustrer lorsque le drag'n'drop ne fonctionne pas comme ils le souhaitent. Toutefois, afin de varier les niveaux et de ne pas rendre les exercices trop répétitifs, on pourrait imaginer d'introduire des éléments qui bougent sur la scène et qu'il faut toucher pendant qu'ils sont en mouvement.

Enfin, malgré l'atteinte partielle de nos objectifs, le logiciel est utilisable sur toutes les plateformes visées, et le nombre d'exercices fourni peut facilement être étendu dans le cas où les écoles souhaiteraient l'utiliser dans l'immédiat.

Nous espérons que le projet sera éventuellement repris par un autre étudiant pour être terminé et qu'il pourra ainsi être utilisé dans les écoles genevoises pour aider les enfants non francophones en âge préscolaire à améliorer leur compréhension orale du français pour les mots et sons les plus fréquents.

6 Bibliographie et Webographie

6.1 Articles

- Alexander, B. (2004). Going nomadic: Mobile learning in higher education. *Educause Review*, 39(5).
- Baron, G.-L., & Bruillard, E. (2008). Technologies de l'information et de la communication et indigènes numériques: quelle situation? *Revue Des Sciences et Technologies de l'Information et de La Communication Pour l'Éducation et La Formation (STICEF)*, 15. Repéré sur <http://telearn.archives-ouvertes.fr/hal-00696420/>
- Bétrancourt, M. (2012a). Du bon usage des tablettes. *Flash informatique D. Vol. Spécial été*, 2012, p. 49-51. Repéré sur <http://flashinformatique.epfl.ch/spip.php?article2573>
- Bétrancourt, M. (2013). Séminaire virtuel « *Les tablettes en éducation : entre phénomène de mode et révolution pédagogique* » *Du bon usage des tablettes en éducation. Ou a-t-on enfin trouvé un outil qui va faire progresser l'intégration des technologies dans les classes ?* CIRT@, Montréal. 14 mai.
- Bidin, S., & Ziden, A. A. (2013). Adoption and Application of Mobile Learning in the Education Industry. *Procedia-Social and Behavioral Sciences*, 90, 720–729.
- Boujol, L. (2014). *Usage pédagogique des tablettes tactiles numériques: une étude de cas en enseignement primaire genevois*. Mémoire de Master MALTT, Université de Genève. Repéré sur <http://archive-ouverte.unige.ch/unige:35124>
- Bozelle, C. (2006). Evaluation de l'acceptabilité d'un environnement d'apprentissage dans le cadre de la rééducation d'enfants sourds ayant reçu un implant cochléaire. Mémoire de Master MALTT, Université de Genève. Repéré sur http://tecaetu.unige.ch/staf/staf-k/bozelle/memoire/memoire_CBozelle.pdf
- Budiu, R., & Nielsen, J. (2011). Usability of iPad apps and websites.
- Buxton, B. (2010). 31.1: Invited Paper: A Touching Story: A Personal Perspective on the History of Touch Interfaces Past and Future. *SID Symposium Digest of Technical Papers*, 41(1), 444–448.
- Couse, L. J., & Chen, D. W. (2010). A Tablet Computer for Young Children? Exploring Its Viability for Early Childhood Education. *Journal of Research on Technology in Education*, 43(1).
- Druin, A. (2009). *Mobile technology for children: Designing for interaction and learning*. Morgan Kaufmann.
- FLUCKIGER C. (2008) La culture numérique des élèves est-elle une culture comme les autres ?, *Revue Française de Pédagogie*, Lyon : INRP, n° 63, p. 51-61
- GIANNOULA, E., & BARON, G.-L. (2002). Pratiques familiales de l'informatique versus pratiques scolaires: représentations de l'informatique chez les élèves d'une classe de CM2. *Sciences et Techniques Éducatives*, 9(3-4), 437–456.
- Gimbert, B., & Cristol, D. (2004). Teaching curriculum with technology: Enhancing children's technological competence during early childhood. *Early Childhood Education Journal*, 31(3), 207–216.

- Hanna, L., Ridsen, K., & Alexander, K. (1997). Guidelines for usability testing with children. *Interactions*, 4(5), 9–14.
- Hamdani, D. S. A. (2013). Mobile Learning: A Good Practice. *Procedia - Social and Behavioral Sciences*, 103, 665–674.
- Hashemi, M., Azizinezhad, M., Najafi, V., & Nesari, A. J. (2011). What is Mobile Learning? Challenges and Capabilities. *Procedia - Social and Behavioral Sciences*, 30, 2477–2481.
- Hourcade, J. P. (2008). Interaction design and children. *Foundations and Trends in Human-Computer Interaction*, 1(4), 277–392.
- Hutchins, E., Hollan, J., & Norman, D. (1985). Direct Manipulation Interfaces. *Human-Computer Interaction*, 1(4), 311–338.
- Jaynes, C. (2013). BYOD in Higher Education: Higher Ed Leading the Way in BYOD [Billet de blog]. Repéré sur : <http://blog.mersive.com/higher-ed-leading-the-way-in-byod/>
- Jobs, S. (2010). Apple Special Event January 2010: Keynote Speech. *Video Accessed*, 31.
- KARSENTI, T., & FIEVEZ, A. (2013). L'iPad à l'école: usages, avantages et défis.
- Kinash, S., Brand, J., & Mathew, T. (2012). Challenging mobile learning discourse through research: Student perceptions of Blackboard Mobile Learn and iPads. *Australasian Journal of Educational Technology*, 28(4), 639–655.
- Kukulka-Hulme, A., & Traxler, J. (2007). *Mobile learning: A handbook for educators and trainers*. Routledge.
- Lachapelle-Bégin, L. (2012). *Vers une révolution tactile? Clic* : Bulletin collégial des technologies de l'information et des communications. N°78, 2012. Repéré sur <http://clic.ntic.org/cgi-bin/aff.pl?page=article&id=2244>
- Lao, S., Heng, X., Zhang, G., Ling, Y., & Wang, P. (2009). A gestural interaction design model for multi-touch displays. In *Proceedings of the 23rd British HCI Group Annual Conference on People and Computers: Celebrating People and Technology* (pp. 440–446). British Computer Society.
- Leclerc J. (21/02/2012). iPad : conception et ergonomie de l'interface [Billet de blog]. Repéré sur <http://www.ergonomie-interface.com/mobile-tactile-nomade/ipad-conception-ergonomie-application/>
- Lueder, R., & Rice, V. (2007). *Ergonomics for children*. CRC Press, Boca Raton, FL.
- Markopoulos, P., Read, J. C., MacFarlane, S., & Hoysniemi, J. (2008). *Evaluating children's interactive products: principles and practices for interaction designers*. Morgan Kaufmann.
- MEDIAPPRO (2006). *A European Research Project: The appropriation of New Media by Youth*. Repéré sur http://www.clemi.org/fichier/plug_download/5942/download_fichier_fr_mediappro_light.pdf
- Melhuish, K., & Falloon, G. (2010). Looking to the future: M-learning with the iPad.
- Meurant, R. C. (2010). The iPad and EFL digital literacy. In *Signal Processing and Multimedia* (pp. 224–234). Springer. Repéré sur http://link.springer.com/chapter/10.1007/978-3-642-17641-8_27

- Michel, C., Sandoz-guermond, F., & Serna, A. (2011). Revue de littérature sur l'évaluation de l'usage de dispositifs mobiles et tactiles ludo-éducatifs pour les jeunes enfants.
- Murphy, G. D. (2011). Post-PC devices: A summary of early iPad technology adoption in tertiary environments. *E-Journal of Business Education & Scholarship of Teaching*, 5(1).
- Naismith, L., Sharples, M., Vavoula, G., & Lonsdale, P. (2004). Literature review in mobile technologies and learning. Repéré sur http://telearn.archives-ouvertes.fr/docs/00/19/01/43/PDF/Naismith_2004.pdf
- Nielsen, J., & Landauer, T. K. (1993). A mathematical model of the finding of usability problems. In *Proceedings of the INTERACT'93 and CHI'93 conference on Human factors in computing systems* (pp. 206-213). ACM.
- Prensky, M. (2001). Digital natives, digital immigrants part 1. *On the Horizon*, 9(5), 1–6.
- Revelle, G. L., & Strommen, E. F. (1990). The effects of practice and input device used on young children's computer control. *Journal of Computing in Childhood Education*, 2(1), 33–41.
- Sachs, L., & Bull, P. H. (2011). USING IPAD2 FOR A GRADUATE PRACTICUM COURSE. *Journal on School Educational Technology*, 7(3).
- Bastien, J.M.C., Scapin, D. (1993) Ergonomic Criteria for the Evaluation of Human-computer interfaces. Institut National de recherche en informatique et en automatique
- Selfe, C. L. (1999). *Technology and literacy in the twenty-first century: The importance of paying attention*. SIU Press.
- Talley, S., Lancy, D. F., & Lee, T. R. (1997). Children, storybooks and computers. *Reading Horizons*, 38(2), 4.
- Tan, T.-H., & Liu, T.-Y. (2004). The mobile-based interactive learning environment (MOBILE) and a case study for assisting elementary school English learning. In *Advanced Learning Technologies, 2004. Proceedings. IEEE International Conference on* (pp. 530–534). IEEE.
- Vavoula, G. N., & Sharples, M. (2002). KLeOS: a personal, mobile, knowledge and learning organisation system. In *Wireless and Mobile Technologies in Education, 2002. Proceedings. IEEE International Workshop on* (pp. 152–156). IEEE.
- Watts, N. (1997). A learner-based design model for interactive multimedia language learning packages. *System*, 25(1), 1–8.
- Wobbrock, J. O., Morris, M. R., & Wilson, A. D. (2009). User-defined gestures for surface computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 1083–1092). ACM.
- Wroblewski L. (02/11/2012) *Responsive Navigation: Optimizing for Touch Across Devices* Repéré sur <http://www.lukew.com/ff/entry.asp?1649>
- Interactions homme-machine. (n.d.). dans *Wikipedia*. Repéré le 12 avril 2014, sur http://fr.wikipedia.org/wiki/Interactions_homme-machine

Why You Only Need to Test with 5 Users. (Nielsen J., 2000). Repéré sur <http://www.nngroup.com/articles/why-you-only-need-to-test-with-5-users/>

6.2 Guides de développement pour les plateformes tactiles

Interaction utilisateur de A à Z, Guides de développement Microsoft. Repéré sur <http://msdn.microsoft.com/fr-fr/library/windows/apps/dn263193.aspx>

Interactivity and feedback, Guides de développement Apple. Repéré sur https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/InteractivityInput.html#//apple_ref/doc/uid/TP40006556-CH55-SW1

Gestures, Guides de développement Android. Repéré sur <http://developer.android.com/design/patterns/gestures.html>

7 Annexes

7.1 Annexe 1 : accès au logiciel en ligne

Le logiciel est disponible en ligne ici :

<http://tecafetu.unige.ch/etu-maltp/R2D2/travnja0/html5/moulin/>

7.2 Annexe 2 : code source du logiciel

Un fichier zip contenant l'ensemble du code source est disponible sur le serveur Tecfa également :

<http://tecafetu.unige.ch/etu-maltp/R2D2/travnja0/html5/moulin.zip>

7.3 Annexe 3 : suivi des versions du logiciel sur GitHub

Le suivi du développement peut être consulté sur la plateforme de versionning GitHub :

<https://github.com/animalsai/moulinCreateJS/commits/master>

7.4 Annexe 3: Recueil de fonctionnalités sous Excel

Le fichier Excel FM_MoulinAParoles.xlsx est disponible [ici](#).

7.5 Annexe 4 Ajout de données pour un nouvel exercice

Pour ajouter de nouveaux exercices dans un domaine existant il suffit de procéder selon les étapes ci-dessous. Nous prenons ici l'exemple de l'exercice 4 pour le domaine Animaux :

1. Ajouter l'icône de l'exercice 4 dans le dossier *media\images\nav*
2. Ajoutez les fichiers du nouvel exercice dans les dossiers appropriés :
media/images/animaux/level4
media/sounds/animaux/level4
3. Ouvrez le fichier *common.js* et exécutez les étapes qui suivent
4. Ajoutez les informations du fichier de sous-navigation ajouté à l'étape 1 dans le tableau nommé *animaux_nav_fileManifest* comme suit :

```
var animaux_nav_fileManifest = [  
  //background  
  {id: "scene", src: IMGFOLDER + NAV_FOLDER + "niveau2_animaux_decor.png"},  
  //level images  
  {id: "animauxEx1", levelId: "animaux1", src: IMGFOLDER + NAV_FOLDER + "niveau2_animaux_ex1.png", "x": 42, "y": 254},  
  {id: "animauxEx2", levelId: "animaux2", src: IMGFOLDER + NAV_FOLDER + "niveau2_animaux_ex2.png", "x": 228, "y": 254},  
  {id: "animauxEx3", levelId: "animaux3", src: IMGFOLDER + NAV_FOLDER + "niveau2_animaux_ex3.png", "x": 408, "y": 254},  
  {id: "animauxEx4", levelId: "animaux4", src: IMGFOLDER + NAV_FOLDER + "niveau2_animaux_ex4.png", "x": 598, "y": 254}  
];
```

Il est important ici de bien donner un ID unique à votre fichier, et de spécifier le *levelId* correctement, le *levelId* doit également exister dans le tableau nommé *levels* (voir étape 6). De plus il faut donner précisément la position du bouton sur la scène à l'aide des coordonnées x et y.

- Ajoutez le tableau contenant les références aux fichiers (*file_manifest*) pour cet exercice avec les images, sons et coordonnées de chaque élément sur la scène comme suit :

```
var animaux4_fileManifest = [
  //background
  {id: "scene", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_decor_ex4.png"},
  //image outlines
  {id: "oie_outline", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_oie_seul.png", "x": 640, "y": 540},
  {id: "dindon_outline", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_dindon_seul.png", "x": 42, "y": 425},
  {id: "agneau_outline", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_agneau_seul.png", "x": 519, "y": 448},
  {id: "chevre_outline", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_chevre_seul.png", "x": 81, "y": 535},
  {id: "poney_outline", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_poney_seul.png", "x": 442, "y": 262},
  //level images
  {id: "oie", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_oie.png", "x": 790, "y": 35},
  {id: "dindon", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_dindon.png", "x": 790, "y": 146},
  {id: "agneau", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_agneau.png", "x": 790, "y": 285},
  {id: "chevre", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_chevre.png", "x": 790, "y": 435},
  {id: "poney", src: IMGFOLDER + ANIMAL_FOLDER + "level4/animaux_poney.png", "x": 790, "y": 600},
  //level sounds
  {id: "oie_snd", src: SNDFOLDER + ANIMAL_FOLDER + "level4/lanimaux.serie4.4.wav"},
  {id: "dindon_snd", src: SNDFOLDER + ANIMAL_FOLDER + "level4/lanimaux.serie4.6.wav"},
  {id: "agneau_snd", src: SNDFOLDER + ANIMAL_FOLDER + "level4/lanimaux.serie4.2.wav"},
  {id: "chevre_snd", src: SNDFOLDER + ANIMAL_FOLDER + "level4/lanimaux.serie4.5.wav"},
  {id: "poney_snd", src: SNDFOLDER + ANIMAL_FOLDER + "level4/lanimaux.serie4.3.wav"},
  //instruction and feedback sounds
  {id: "consignes_animaux4", src: SNDFOLDER + ANIMAL_FOLDER + "level4/lanimaux.serie4.consigne.1.wav"}
];
```

Chacune des images doit avoir une position définie par les axes x et y sur la scène, de plus il faut spécifier son chemin accès ainsi qu'un identifiant unique. Pour les identifiants il est important de garder les suffixes *_outline* et *_snd* selon le type de fichier.

- Ajoutez le nouvel exercice dans le tableau appelé *levels* en spécifiant le nom du *file_manifest* (étape 5) correspondant que le type d'interaction, comme suit :

```
var levels = [
  {id: "score", theme: "score", media: "interLevel_fileManifest"},
  {id: "animaux1", theme: "animaux", media: "animaux1_fileManifest", interaction: InteractionTypeEnum.GUIDED},
  {id: "animaux2", theme: "animaux", media: "animaux2_fileManifest", interaction: InteractionTypeEnum.GUIDED},
  {id: "animaux3", theme: "animaux", media: "animaux3_fileManifest", interaction: InteractionTypeEnum.GUIDED},
  {id: "animaux4", theme: "animaux", media: "animaux4_fileManifest", interaction: InteractionTypeEnum.GUIDED},
  {id: "habits1", theme: "habits", media: "habits1_fileManifest", interaction: InteractionTypeEnum.FREEDRAG},
  {id: "habits2", theme: "habits", media: "habits2_fileManifest", interaction: InteractionTypeEnum.FREEDRAG},
  {id: "habits3", theme: "habits", media: "habits3_fileManifest", interaction: InteractionTypeEnum.FREEDRAG},
  {id: "objets1", theme: "objets", media: "objets1_fileManifest", interaction: InteractionTypeEnum.GUIDED},
  {id: "objets2", theme: "objets", media: "objets2_fileManifest", interaction: InteractionTypeEnum.GUIDED},
  {id: "objets3", theme: "objets", media: "objets3_fileManifest", interaction: InteractionTypeEnum.GUIDED}
];
```

- Rafraîchissez la page d'accueil du logiciel. Le niveau devrait désormais être fonctionnel