



HAL
open science

Danse-doigts, jeu de motricité fine

Jean-Ferdy Susini, Olivier Pons, Nolwenn Guedin, Catherine Thevenot

► **To cite this version:**

Jean-Ferdy Susini, Olivier Pons, Nolwenn Guedin, Catherine Thevenot. Danse-doigts, jeu de motricité fine. Handicap 2016. La recherche au service de la qualité de vie et de l'autonomie, Jun 2016, paris, France. hal-02321088

HAL Id: hal-02321088

<https://hal.science/hal-02321088v1>

Submitted on 20 Oct 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Danse-doigts, jeu de motricité fine

Jean-Ferdinand Susini*, Olivier Pons†
CÉDRIC, CNAM, France.
*jean-ferdinand.susini@cnam.fr,
†olivier.pons@cnam.fr

Nolwenn Guedin
FPSE, UNIGE, Suisse.
nolwenn.guedin@unige.ch

Catherine Thevenot
IP, UNIL, Suisse
catherine.thevenot@unil.ch

Résumé— Ce document décrit la conception, la réalisation et l'expérimentation de « *Danse-doigts* », une application ludothérapeutique à destination d'enfants hémiparétiques.

L'objectif de ce logiciel est double : permettre d'entraîner leur motricité fine sur tablette puis d'étudier l'effet de cet entraînement sur leurs compétences numériques (dénombrement, calcul...).

La population visée, comme l'objectif d'évaluation des compétences de calcul, ont influencé la conception. Le développement logiciel se base sur des technologies Web standard mais repose sur une bibliothèque de programmation parallèle en javascript.

Applications et bibliothèques sont disponibles gratuitement et s'installent très facilement sur la plupart des tablettes tactiles.

I. INTRODUCTION

Dans cet article nous présentons la démarche suivie durant la réalisation de « *Danse-doigts* », une application ludothérapeutique pour la rééducation du membre supérieur d'enfants hémiparétiques. Notre objectif est double : d'abord fournir un outil d'entraînement de la motricité fine de la main, mais aussi mesurer l'influence de cet entraînement sur les compétences numériques (dénombrement, calcul...).

Cette dualité rend la conception de « *Danse-doigts* » particulièrement intéressante à étudier. En effet, nous souhaitons concevoir un jeu dont les aspects ludiques et ergonomiques soient pertinents vis-à-vis de la population ciblée, tout en intégrant de manière fine les intentions thérapeutiques, sans que cela ne vienne perturber l'étude visée.

L'absence de données précises sur la population ciblée (notamment les degrés variés de l'atteinte des capacités motrices, l'importance de la spasticité, la présence de troubles associés comme la taille de la main) a encouragé une conception modulaire et adaptative. L'utilisation de bibliothèques de programmation spécifiques, que nous présentons section V, a grandement facilité cette conception.

Nous avons dû formuler des hypothèses sur la base de nos propres intuitions et expériences, puis les adapter incrémentalement en fonction du retour des expérimentations menées in vivo par les thérapeutes.

Après une présentation des jeux et applications impliquant la motricité de la main, nous présentons le contexte et le cahier des charges puis nous précisons nos hypothèses et la méthodologie utilisée pour créer des mécaniques ludiques, accessibles et motivantes pour un public d'enfants, compatibles avec les contraintes de notre cahier des charges. Nous décrivons les principales caractéristiques du framework et des bibliothèques

de programmation développés [1]. Nous présentons ensuite les résultats d'une expérimentation préliminaire conduite sur deux classes d'enfants sains, dans le but d'adapter le logiciel aux contraintes réelles et d'étudier le lien entre doigts et nombres. Nous concluons cette étude par une discussion sur les résultats de l'expérimentation, la généralisation de notre démarche et les extensions envisagées.

II. L'EXISTANT

Les applications pour tablettes permettant d'exercer la motricité fine sont nombreuses, mais beaucoup impliquent aussi un travail cognitif (puzzle, préparation à l'écriture...), et celles qui visent la seule fonction motrice sont rarement configurables et souvent peu accessibles à un joueur présentant une déficience motrice d'un membre supérieur.

Parmi les applications les plus utilisées en rééducation motrice, deux semblent se détacher :

- 1) Fruit Ninja [2], développé par Halfbrick, est un jeu consistant à découper des fruits jetés en l'air en faisant glisser son (ses) doigt(s) sur l'écran tactile. C'est désormais un grand classique, qui a été porté sur toutes les plateformes et décliné en différentes versions. Il n'était initialement pas à visée thérapeutique mais a beaucoup été utilisé à cet effet [3], [4].
- 2) La suite Dexteria, développé par BinaryLabs, comprend « Dexteria Fine Motor Skill Development » [5] qui se présente explicitement comme

« un ensemble d'exercices thérapeutiques pour les mains (et non des jeux) visant à améliorer la motricité fine et la préparation à l'écriture chez les enfants et les adultes ».

Les exercices de Dexteria utilisent les capacités multi-touches de l'écran pour travailler différents gestes et visent au renforcement (force, contrôle et dextérité) de la main.

Néanmoins si leurs différents *GamePlay* peuvent servir d'inspiration, aucune de ces applications ne semble posséder les caractéristiques qui, en terme de développement logiciel, nous paraissent importantes pour permettre une utilisation optimale des applications de rééducation :

- Être déployable aisément et à moindre coût (sans réécriture) sur tout type de tablette tactile.
- Être facilement personnalisable et configurable en fonction des contraintes thérapeutiques ou expérimentales.

- Permettre de suivre finement toutes les interactions avec l'« utilisateur » à différents niveaux de précision.
- Être extensible afin de permettre de rajouter de nouvelles fonctionnalités sans compromettre l'existant.

La plateforme de développement se doit aussi d'être open-source. Ainsi toute équipe de recherche qui le souhaite pourra l'adapter à ses besoins propres.

III. CONTRAINTES DU CAHIER DES CHARGES

Nous présentons ici la population ciblée et l'objectif de l'étude envisagée par la suite. Nous étudions leur influence sur la conception de l'application.

A. La cible : l'enfant hémiparétique

L'hémiparésie est un déficit partiel d'un hémicorps lié à une lésion cérébrale. Relativement commune chez l'enfant, avec une prévalence de 1 enfant sur 1000, elle peut être congénitale ou acquise, plus ou moins sévère et éventuellement associée à d'autres troubles (épilepsie, troubles spatio-visuels...).

Cette diversité complique la spécification et la conception d'applications de rééducation spécifiques.

Ciblant seulement le membre supérieur, nous avons supposé, dans la version dédiée à l'étude, une mobilité suffisante pour cliquer avec les différents doigts et, dans les versions suivantes, une capacité, même limitée, à effectuer un mouvement de pince entre les doigts et le pouce, le mouvement de pince pouce-index étant le plus difficile pour une main hémiparétique.

Enfin, l'enfant déficient moteur devant déjà assumer, outre ses activités scolaires, de nombreux rendez-vous thérapeutiques réguliers (kinésithérapie, ergothérapie, psychomotricité...), l'entraînement qui vient en complément des thérapies motrices traditionnelles ne doit pas être une surcharge trop importante. Il est donc opportun qu'il puisse se faire à domicile ou sur le temps scolaire. Cela ne fait que renforcer les impératifs de déployabilité, de facilité de prise en main, et de coût déjà mentionnés section II ainsi que la nécessité d'introduire des consignes claires dans le *GamePlay*.

B. L'objectif de l'étude : doigts et capacités numériques

Dès les premières années d'apprentissage et parfois encore à l'âge adulte, nos doigts nous servent à représenter et communiquer des quantités et aussi à effectuer des calculs [6]. Le syndrome de Gerstmann [7] décrit en 1940 suggère qu'une atteinte de reconnaissance digitale puisse être liée à de faibles capacités numériques. Depuis, des approches expérimentales ont permis de tester la nature de la relation entre les habiletés digitales et les performances numériques. Elles ont ainsi montré que la relation entre doigts et nombres est d'origine anatomique comme en témoigne l'intrication des aires neuronales de la discrimination digitale et de celles du traitement numérique [8]–[10]. En complément à cette hypothèse localisationniste, d'autres chercheurs considèrent plus particulièrement que ce serait l'utilisation fonctionnelle précoce des doigts dans certaines routines gestuelles, tels le dénombrement par pointage ou le calcul sur les mains, qui

aiderait à la bonne mise en place des compétences numériques [11], [12]. L'épreuve de gnosie digitale permet de tester la capacité de reconnaissance tactile de ses propres doigts sans contrôle visuel. Comparée à des tests d'intelligence classiques, cette tâche sensorimotrice effectuée par des enfants de 5 ans est un meilleur prédicteur de leur réussite en mathématiques, mesuré un an plus tard [13]. Par ailleurs, l'usage spontané des doigts dans des tâches de dénombrement augmente de 4 à 7 ans, tout en étant corrélé avec un meilleur taux de réussite [14]. Dans la majorité des cas, une gnosie digitale peu développée se révèle être un marqueur des troubles du calcul même chez des enfants de 8 à 11 ans [12], [15], [16]. Pourtant, certains enfants avec de mauvaises gnosies digitales du fait de leur hémiparésie de naissance réussissent à compenser suffisamment leurs difficultés numériques initiales pour finalement atteindre un niveau en arithmétique comparable à leurs pairs [17].

Si les habiletés digitales sont en lien avec les compétences numériques, leur entraînement devrait conduire à des progrès également dans le domaine des mathématiques. Les actions demandées doivent donc solliciter au maximum les dissociations digitales pour tester les répercussions de cet entraînement sur les compétences numériques, d'où les actions de pointage par chacun des doigts de la main. À ce jour, une seule étude a testé cette hypothèse auprès d'enfants de 6 à 8 ans [18]. Dans cette étude, les actions réelles de pointage et de suivi de labyrinthes durant un entraînement de 30 minutes par jours durant 8 semaines ont montré une amélioration des performances en mathématiques mais la validité de ces résultats a été remise en cause [19].

L'un des objectifs de l'étude menée avec « *Danse-doigts* » est de confirmer ces résultats puis d'affiner, par la suite, les apports sur des enfants atteints d'hémiparésie.

C. Contraintes de conception

Nous présentons ici les contraintes (parfois contradictoires) qui ont orienté la conception.

1) *Maintenir la motivation* : L'activité de rééducation devant d'être régulière et poursuivie sur le long terme peut vite devenir répétitive, peu gratifiante et ennuyeuse, entraînant une démotivation des patients qui risquent alors de suivre irrégulièrement le protocole, voire de l'abandonner.

Les méthodologies de game design et de gamification retiennent généralement plusieurs éléments susceptibles de maintenir l'engagement. Les principaux sont : la scénarisation, l'organisation d'objectifs dirigés vers des compétences ciblées, l'importance des récompenses et des commentaires sur les progrès, l'augmentation des niveaux de difficulté, une expérience individualisée, et la possibilité de faire des choix [20], [21].

2) *Éviter le contournement* : Motivé par le jeu, le patient peut être tenté de détourner le protocole. Il peut, par exemple, utiliser systématiquement l'index dans des tâches de pointage ou, dans le cas d'un patient hémiparétique, utiliser sa main valide lorsqu'on lui demande d'utiliser sa main parétique.

Il est essentiel d'essayer au minimum de détecter ces détournements et dans la mesure du possible de les empêcher

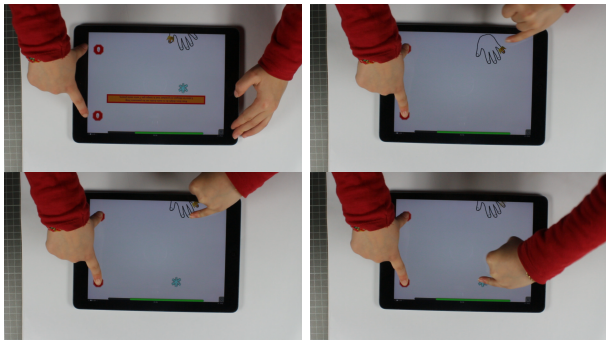


FIGURE 1. Deux doigts de la main qui ne joue pas doivent rester posés sur les panneaux "stop". Une petite main avec une couronne indiquant le doigt qui va jouer s'affiche. L'enfant clique sur la couronne avec le doigt demandé (ici l'auriculaire) pour faire apparaître la cible. Lorsque l'enfant clique sur la cible (ou si le timeout est dépassé), on recommence avec un autre doigt.

en ajoutant au *GamePlay* les contraintes nécessaires. Pour s'assurer de l'utilisation d'une main sur une tâche. On pourra, par exemple, mobiliser l'autre parallèlement dans une tâche statique comme sur la figure 1.

3) *Ne pas biaiser l'étude des compétences numériques* : Puisque l'on veut étudier l'influence de l'entraînement moteur sur les capacités numériques, afin d'éviter tout biais, il ne faut pas que l'application entraîne ces capacités en elles-mêmes.

Pour cela, nous ne devons donc utiliser ou mentionner aucun nombre. Nous ne pouvons donc pas compter les points, les succès ou les échecs. Nous ne devons pas non plus estimer numériquement le nombre de problèmes ou le temps restant.

Il fallait aussi éviter tout biais cognitif lié par exemple à un détournement d'attention (musique de fond, événement parasite, environnement graphique changeant...)

IV. « GAME DESIGN »

A. Concilier les contraintes

Le logiciel conçu pour l'étude numérique correspond à une application ludique d'actions de pointage sur différentes cibles et avec différents doigts.

Les contraintes exprimées dans la section III-C3 réduisent fortement le côté ludique que nous avons néanmoins tenté de maintenir au maximum en scénarisant les tâches répétitives autour de thématiques amusantes pour l'enfant.

Plusieurs thèmes sont proposés contenant chacun plusieurs sous-thèmes. Un certain temps de jeu est nécessaire à l'enfant pour découvrir un thème complet (une semaine dans l'expérience décrite section VI).

Par exemple, nous avons proposé aux enfants le thème « la nature » et ils ont pu réaliser du lundi au vendredi les sous-thèmes suivants : « le ciel », « la ferme », « le jardin », « la forêt », « le lac ». À l'intérieur de ces sous-thèmes, l'enfant peut réaliser 4 jeux différents de deux minutes trente chacun. « le ciel » par exemple est décomposé en « les nuages », « les flocons », « le soleil », « la pluie » (cibles et talles différentes).

Pour chaque challenge de pointage, la réussite est reconnue par un renforcement audio. Enfin, chaque séance se conclut

par une phrase de félicitation et une image de récompense rassemblant tous les jeux réalisés durant la séance de travail. Par exemple, pour « le ciel », l'image se compose d'un dessin de pluie, un dessin de soleil, un bonhomme de neige et un nuage.

La suppression de toute référence au temps restant ayant gêné les enfants lors de la première semaine de l'expérimentation (décrite section VI), nous avons introduit une jauge pour exprimer l'avancement en remplacement de l'information numérique.

Les contraintes de l'étude limitent aussi les capacités que l'on cherche à entraîner. Ainsi dans la version dédiée à l'expérimentation, seules les activités de pointage sont proposées ; les activités nécessitant plusieurs doigts, comme la pince ou la préhension, n'ont donc pas été utilisées pour cette expérience.

B. La phase de jeu

Une fois l'application lancée avec le thème de la semaine, l'enfant voit apparaître les consignes à respecter lors de la session. Lorsque les règles sont passées, il choisit un sous-thème qui lui plaît (ex. : « le ciel »). Sur la tablette, les 4 jeux qui décomposent le sous-thème apparaissent alors (ex. : « la pluie », « le soleil »...). L'enfant choisit celui qu'il veut pour lancer le premier jeu. Deux panneaux « stop » apparaissent alors à l'écran à droite ou à gauche selon la main à travailler. Pour que le jeu puisse continuer, l'enfant doit poser, sur ces panneaux, deux doigts de la main qui ne travaille pas (figure 1). Ils doivent rester posés pendant toute la durée de l'exercice.

Une fois les doigts en contact avec les panneaux « stop », un dessin de l'autre main apparaît en bas de l'écran avec un doigt coiffé d'une couronne. L'enfant doit alors repérer ce doigt sur sa main non bloquée et l'utiliser pour aller pointer les différentes cibles qui lui seront proposées. Les cibles sont immobiles ou mouvantes et de différentes tailles. Le doigt sélectionné change entre chaque cible et il faut donc discriminer ses différents doigts en permanence. Les doigts sélectionnés pour pointer (montrés par l'apparition d'une couronne) ont été choisis aléatoirement dans des cycles de cinq doigts différents. Il n'y a aucun ordre précis, ce qui limite une potentielle anticipation des enfants. Ainsi, la discrimination des différents doigts se fait sans habitude à une série de gestes répétés.

Dès que la cible est touchée, elle disparaît et une autre prend sa place à un endroit différent de la tablette. Cette disparition de cible est toujours accompagnée d'un "son amusant" en lien avec la cible. Cette combinaison image et son permet une meilleure stimulation des enfants car les effets de l'observation sont augmentés quand les inputs sont multimodaux [22]. En effet, à partir du moment où l'enfant peut associer plusieurs sens à l'observation, les apprentissages sont augmentés.

V. CONCEPTION ET RÉALISATION LOGICIEL

Nous présentons maintenant quelques éléments techniques concernant l'implantation de l'application. Nous discutons en particulier des bibliothèques de programmation que nous avons développées afin de faciliter la mise en œuvre d'applications ludo-thérapeutiques.

A. Implication technique des contraintes

Si les contraintes 1 et 3 décrites dans la section III-C relèvent essentiellement de la conception même de l'application, nous avons estimé que le respect de la contrainte 2 devait être en partie garanti par l'application elle-même. Le cahier des charges suppose que nous puissions suivre en permanence le comportement des mains des enfants. Modulo quelques hypothèses simples, la détection de points de contact multiples que l'on trouve sur les tablettes à écran tactile permet, dans une certaine mesure, de « surveiller » durant les exercices le bon respect des consignes. Cependant, au cours de l'expérimentation, nos hypothèses de conception sont validées par le contrôle visuel d'un adulte. Ainsi, pour contrôler la main immobilisée, notre application surveille deux points de contact « suffisamment » éloignés l'un de l'autre, sur une zone de l'écran n'intervenant pas dans le jeu.

Dans le cas du pointage simple des cibles, à chaque instant nous ne devons observer au plus qu'un unique point de contact qui devrait être celui du doigt touchant la cible. Notre application peut détecter des points de contact multiples dans la zone de jeu et alerter sur le non respect de la consigne « un seul doigt utilisé pour pointer ». Cependant, dans nos expériences nous nous sommes seulement concentrés sur la réussite de l'exercice, c'est à dire détecter un point de contact dans la zone de la cible. Les autres points de contact éventuels ont été négligés. Mais nous pourrions techniquement décider d'enregistrer ces contacts parasites.

Remarquons que notre application est également capable de suivre des points de contact glissants par exemple lorsque l'on réalise des exercices comme la pince rapprochant deux cibles ou la pince se refermant sur une cible déterminée.

Pour résumer, notre application doit donc pouvoir :

- suivre au moins 3 points de contact en permanence ;
- détecter d'éventuels points de contact parasites dans la zone de jeu ;
- suivre des points de contacts glissants dans la zone de jeu ;
- gérer les animations graphiques de l'application (affichage des cibles statiques, trajectoires des cibles mouvantes, affichages de consignes, jauge de progression...);
- coordonner l'émission des sons de récompenses ou l'émission des consignes audio ;
- observer les interactions tactiles pour en retirer des données statistiques.

Ce grand nombre d'activités à mener de front (en parallèle), nous a conduit à envisager un paradigme de programmation parallèle pour faciliter l'écriture du code de l'application. Comme nous le verrons dans la sous-section V-C, le langage de programmation choisi pour l'écriture du code de l'application est le langage Javascript, présent dans tout navigateur Web. Cependant, ce langage ne permet pas de manipuler simplement le parallélisme de tâches dans sa version standard. Nous avons donc utilisé le framework *SugarCubesJS* que nous développons et qui apporte cette dimension à Javascript.

B. Parallélisme et bibliothèque de programmation réactive

La bibliothèque *SugarCubesJS* permet d'écrire des programmes conçus comme un agrégat de tâches exécutées en parallèle. Cette bibliothèque repose sur le modèle de programmation réactif/synchrone « à la » Boussinot [23].

Schématiquement, ce modèle de programmation propose de découper l'exécution d'un système informatique dans le temps en portions finies exécutées les unes après les autres. Cette découpe revient à décomposer l'exécution en une succession d'instantanés logiques constituant ce que l'on appelle une horloge, rythmant l'exécution du système. Grâce à cette notion d'horloge, il devient possible de considérer les programmes comme un ensemble de tâches élémentaires s'exécutant en parallèle au cours du même instant. Dans ce modèle de programmation, la synchronisation des tâches parallèles devient une propriété naturelle car la complexité de la communication et de la synchronisation des actions est prise en charge par le modèle de programmation lui-même. Écrire un programme devient alors très simple. Il suffit d'écrire de petites tâches indépendantes entre elles et qui seront placées en parallèle les unes avec les autres dans le système.

Suivant ce paradigme, le framework *SugarCubesJS* propose une notion particulière d'événements qui se différencie des événements Javascript classiques. Un événement est un objet global (accessible par tous les composants parallèles du système) caractérisé par sa présence ou son absence au cours d'un instant. À chaque instant d'exécution, un événement sera exclusivement soit présent soit absent. On dit de ces événements qu'ils sont diffusés instantanément car ils sont accessibles par toutes les tâches parallèles (diffusion) au cours du même instant (instantanéité). Lorsqu'il est présent au cours d'un instant, un événement peut avoir une liste d'objets associée (liste de valeurs communiquées à l'ensemble des composants intéressés). La cohérence de l'état d'un événement à un instant logique donné garantit que tous les composants accèdent à la même liste de valeurs associées exactement au même instant logique. La liste des valeurs n'est accessible qu'à l'instant précis où elle est définie. À chaque instant, une nouvelle liste de valeurs sera définie si l'événement est présent. Les interactions entre les tâches parallèles sont donc prises en charge par la diffusion instantanée d'événements.

Dans le cadre de notre application « *Danse-doigts* », l'utilisation de ce paradigme réactif a grandement facilité la programmation du suivi parallèle des points de contacts (quelques tâches très simples en parallèle permettent de générer des événements rendant compte du comportement supposé du joueur), de la combinatoire issue de la logique du jeu lui-même, des animations et de la diffusion des sons du jeu (renforcement audio en cas de succès, énoncé des consignes...).

Autre avantage très important, cela permet de faciliter l'implantation d'observateurs de comportements. Il est ainsi très simple de rajouter des programmes observateurs en parallèle dans un système réactif écoutant les données émises par le système initial (système observé). Les observateurs traitent en parallèle les données écoutées pour les filtrer, les analyser et

les stocker, sans perturber le système observé.

La collecte des données d'expérience peut être implantée grâce à *SugarCubesJS* de façon totalement indépendante du programme gérant le jeu et des tâches contrôlant le comportement du joueur. Les tâches chargées de la collecte des informations seront écrites sous la forme d'activités parallèles écoutant simplement les événements émis par les programmes gérant les points de contact de l'interface tactile, les événements gérant la logique du jeu et les animations...

En fonction des données statistiques attendues, les tâches de collecte pourront réaliser des agrégations, des corrélations et différents autres prétraitements sur les données avant leur transmission au serveur de collecte sans que cela n'impacte jamais le développement du jeu ou l'interaction avec l'utilisateur. Les tâches *SugarCubesJS* étant indépendantes entre elles, notre application peut-être lancée avec ou sans les tâches de collecte de données car les tâches du jeu émettent leurs événements sans jamais avoir « conscience » de l'existence d'autres tâches en parallèle (techniquement, on ne manipule jamais de références aux autres tâches). Si toute ou partie des tâches de collecte sont présentes dans le système ou même si elle ne sont pas présentes du tout, cela n'impacte absolument pas le comportement de notre application. Pour illustrer cela nous avons poussé notre expérimentation jusqu'à développer nos tâches de collecte sur un serveur différent de celui de l'application.

C. Développement d'une «WebApp»

Afin de faciliter le déploiement de l'application sur un maximum de tablettes du marché, nous avons opté pour une solution indépendante des systèmes et des plateformes des différents constructeurs, à savoir la création d'une « WebApp ». Ainsi, en recourant uniquement à HTML, CSS et Javascript, notre objectif est de donner un accès facile à l'application (au moyen d'un simple navigateur Web). Les standards bien établis de l'industrie du Web facilitent l'écriture de l'application et son portage, sans surcoût, sur les différentes tablettes utilisées dans les familles d'enfants participant à l'expérience.

L'intégration sur les systèmes Android et iOS est renforcée grâce à l'utilisation d'un navigateur Web particulier sur chaque plateforme : Chrome pour Android et Safari pour iOS. En effet, après avoir chargé la page Web de l'application dans le navigateur, il est possible de l'installer sur l'écran de démarrage de la tablette, créant ainsi un raccourci, simulant bien le comportement d'une application native. Lorsque l'utilisateur lance l'application depuis ce raccourci, l'interface de notre application Web, dépourvue des éléments spécifiques du navigateur, est alors exposée en plein écran. L'utilisateur se retrouve immergé dans une expérience similaire à celle d'une application native.

Outre la facilité d'accès à l'application, cette solution technique rend également assez facile la gestion de sa mise à jour, en corrigeant les fichiers directement sur le serveur web. Notre framework permet également d'utiliser l'application hors connexion internet. Cependant, une connexion internet reste nécessaire pour la collecte des données statistiques. En

cas d'absence temporaire de cette connexion, notre framework prévoit de faire résider sur la tablette les données, le temps de retrouver un accès au réseau.

VI. EXPÉRIENCE ET EXTENSIONS

Une étude préliminaire [24], [25] a été menée pendant 9 semaines auprès d'enfants tout-venant, issus d'une classe de CP et d'une classe de CE1.

L'étude a porté sur deux groupes : un groupe réalisant l'entraînement sur tablette et un groupe contrôle. Le groupe entraînement était composé de 17 enfants (9 CP et 8 CE1). Le groupe contrôle était composé d'autant d'enfants. L'âge moyen des enfants était de 6,6 ans. Les enfants des deux groupes étaient répartis harmonieusement selon leur score aux tests de gnose digitale. L'entraînement s'est fait pendant le temps scolaire. L'entraînement sur tablettes était de 10 minutes par jour quatre fois par semaine. Les enfants travaillaient dans la salle informatique et étaient munis de casques audio pour pouvoir profiter des consignes sonores. Pendant ce temps, la maîtresse gardait le groupe contrôle en classe pour faire de la compréhension d'histoire afin de ne pas biaiser les résultats avec une activité en mathématique ou qui pourrait solliciter les doigts. L'école disposant seulement de 11 tablettes et afin de permettre aux enfants d'avoir chacun la leur, les CP et les CE1 ne travaillaient pas au même moment.

Plusieurs types de test ont été faits avant et après la période d'entraînement (Nine Hole Peg Test, gnosies digitales, évaluation des compétences numériques ...).

Pour tester des gnosies digitales de l'enfant, on lui fait porter des mitaines aux doigts marqués de couleurs différentes et on répète l'expérience suivante : ses mains reposant paumes vers le bas sous un cache, l'examineur lui touche un doigt puis soulève le cache ; l'enfant nomme alors la couleur correspondant au doigt touché.

Les compétences numériques sont évaluées sur du matériel non symbolique (comparaison rapide de nuages de cubes), sur des quantités semi-symboliques (reconnaissance de configurations digitales) et enfin sur des représentations symboliques chiffrées (additions).

Dans un premier temps, un traitement statistique des résultats a été fait par des ANOVA à deux facteurs en comparant les deux groupes (entraîné vs. contrôle) et les résultats en pré-entraînement vs. en post-entraînement, sans révéler de différence significative. Des ANCOVA sur le même plan avec les niveaux initiaux en pré-entraînement en covariable n'ont pas permis non plus de mettre en évidence de progrès significatifs ni des gnosies digitales, ni des capacités de calcul des enfants (peut-être en raison du temps trop court d'exposition au logiciel). Cela a cependant permis d'adapter le logiciel aux contraintes réelles, de vérifier sa facilité d'utilisation, de prise en main ainsi que l'intérêt et la bonne acceptation des enfants.

Sur cette base, nous avons expérimenté quelques extensions qui n'ont pas encore fait l'objet d'étude auprès d'enfants :

- Utiliser deux doigts (le pouce et un doigt opposable) de la main qui travaille pour réaliser un mouvement de pince. L'enfant doit ainsi saisir virtuellement deux

RÉFÉRENCES

- cibles (un doigt sur chaque cible) et les faire glisser sous ses doigts sur l'écran pour amener les deux cibles en contact, réalisant le mouvement de pince.
- Une variante de cet exercice a été également réalisée, désignant par une cible statique une zone de l'écran sur laquelle l'enfant doit refermer son mouvement de pince.
 - Il est possible de compliquer l'exercice précédent en transformant la cible statique en une cible mouvante selon une trajectoire prédictible à l'image des cibles mouvantes des exercices de pointage simple.
 - Il est également très simple d'augmenter le nombre de doigts travaillant simultanément à l'exercice de pince. Mais il faut que la tablette tolère une détection réelle des différents points de contact. En général, les tablettes reconnaissent 5 points de contacts différents (autorisant une pince à 3 doigts, 2 contacts étant utilisés par la contrainte d'immobilisation de l'autre main). Les tablettes plus modernes reconnaissent une douzaine de contacts ce qui est très largement suffisant pour ce genre d'exercice.
 - Enfin, nous pouvons facilement réaliser des exercices de pointage ou de glissement des 5 doigts (si le matériel le supporte effectivement), à partir de positions de repos.
- Le suivi de tous les signaux des actions utilisateurs permet d'analyser ces actions à différents niveaux de précision.
- ### VII. DISCUSSION & CONCLUSION
- Diverses versions de l'application « *Danse-doigts* » [26] ainsi que les bibliothèques de programmation *SugarCubesJS* [1] sont disponibles en ligne.
- L'application a été bien accueillie et facilement prise en main tant par les enfants que par les encadrants. L'expérimentation menée en classe ordinaire n'est pas venue confirmer les résultats attendus, peut être en raison des temps d'entraînement trop courts. Cependant, l'étude mérite d'être poursuivie avec des patients hémiparétiques. De plus, le retour des utilisateurs dans la première semaine d'utilisation a permis d'affiner les contraintes et de corriger certaines erreurs de conception. Ces mises au point incrémentales ont permis de tester intensivement le framework *SugarCubesJS* qui a montré sa robustesse et les facilités d'adaptation et d'extensions qu'il permet.
- Nous voulons aussi évidemment étudier l'efficacité de l'entraînement avec nos diverses variantes d'applications sur les fonctions motrices au vu des différentes échelles d'évaluation.
- Pour cela, l'utilisation de la bibliothèque *SugarCubesJS* et le framework développé pour les expérimentations nous permettent maintenant d'envisager de décliner facilement plusieurs variantes dédiées au travail de capacités particulières ou à une étude précise.
- ### REMERCIEMENTS
- Nous remercions l'*Association Hémiparésie*¹ qui a permis d'initier ce projet; ainsi que le personnel et les enfants de l'école Tivoli à Dijon où ont été menées les expérimentations.
1. <http://www.hemiparesie.org/>
- [1] J.-F. Susini. (2015) *Sugarcubesjs*. [En ligne]. Disponible : <https://github.com/LordManta/SugarCubesJS>
 - [2] Fruit ninja. [En ligne]. Disponible : <http://fruitninja.com/>
 - [3] D. Rand *et al.*, "The use of the ipad for poststroke hand rehabilitation; a pilot study," in *Virtual Rehabilitation (ICVR), 2013 International Conference on*. IEEE, 2013, pp. 109–113.
 - [4] D. Rand, G. Zeilig, and R. Kizony, "Rehab-let : touchscreen tablet for self-training impaired dexterity post stroke : study protocol for a pilot randomized controlled trial," *Trials*, vol. 16, no. 1, pp. 1–7, 2015.
 - [5] Dexteria - fine motor skill development app. [En ligne]. Disponible : <http://www.dexteria.net/>
 - [6] G. Ifrah, *Histoire universelle des chiffres*, R. Laffont, Ed. Paris, 1981.
 - [7] J. Gerstmann, "Syndrome of finger agnosia, disorientation for right and left, agraphia and acalculia : Local diagnostic value," *Archives of Neurology and Psychiatry*, vol. 44, no. 2, p. 398, 1940.
 - [8] P. Pinel *et al.*, "Distributed and overlapping cerebral representations of number, size, and luminance during comparative judgments," *Neuron*, vol. 41, no. 6, pp. 983–993, 2004.
 - [9] E. Rusconi, V. Walsh, and B. Butterworth, "Dexterity with numbers : rtms over left angular gyrus disrupts finger gnosis and number processing," *Neuropsychologia*, vol. 43, no. 11, pp. 1609–1624, 2005.
 - [10] M. Sato *et al.*, "Numbers within our hands : modulation of corticospinal excitability of hand muscles during numerical judgment," *Journal of Cognitive Neuroscience*, vol. 19, no. 4, pp. 684–693, 2007.
 - [11] B. Brian, *The mathematical brain*, London, Ed. Macmillan, 1999.
 - [12] M.-P. Noël, "Finger gnosis : a predictor of numerical abilities in children ?" *Child Neuropsychology*, vol. 11, no. 5, pp. 413–430, 2005.
 - [13] M. Fayol, P. Barrouillet, and C. Marinthe, "Predicting arithmetical achievement from neuro-psychological performance : A longitudinal study," *Cognition*, vol. 68, no. 2, pp. B63–B70, 1998.
 - [14] A. Lafay *et al.*, "The role of fingers in number processing in young children," *Frontiers in psychology*, vol. 4, 2013.
 - [15] A. J. Costa *et al.*, "A hand full of numbers : a role for offloading in arithmetics learning ?" *Handy numbers : finger counting and numerical cognition*, p. 13, 2011.
 - [16] R. Reeve and J. Humberstone, "Five-to 7-year-olds' finger gnosis and calculation abilities," *Frontiers Psychology*, vol. 2, 2011.
 - [17] C. Thevenot *et al.*, "Numerical abilities in children with congenital hemiplegia : An investigation of the role of finger use in number processing," *Developmental neuropsychology*, vol. 39, no. 2, pp. 88–100, 2014.
 - [18] M. Gracia-Bafalluy and M.-P. Noël, "Does finger training increase young children's numerical performance ?" *Cortex*, vol. 44, no. 4, pp. 368–375, 2008.
 - [19] J.-P. Fischer, "Numerical performance increased by finger training : A fallacy due to regression toward the mean ?" *Cortex*, vol. 46, no. 2, pp. 272–273, 2010.
 - [20] T. Baranowski *et al.*, "Playing for real : video games and stories for health-related behavior change," *American journal of preventive medicine*, vol. 34, no. 1, pp. 74–82, 2008.
 - [21] K. M. Kapp, *The gamification of learning and instruction fieldbook : Ideas into practice*. John Wiley & Sons, 2013.
 - [22] K. Alaerts, S. P. Swinnen, and N. Wenderoth, "Interaction of sound and sight during action perception : evidence for shared modality-dependent action representations," *Neuropsychologia*, vol. 47, no. 12, pp. 2593–2599, 2009.
 - [23] F. Boussinot, *La programmation réactive - Application aux systèmes communicants, Collection technique et scientifique des télécommunications*. Masson, 1996.
 - [24] L. Nogier, "Entraînement des gnosies digitales et influence sur des tâches numériques," Master's thesis, Faculté des sciences du sport, DIJON, 2015.
 - [25] R. Picard, "Projet d'amélioration des compétences numériques chez l'enfant par entraînement des habiletés et des gnosies digitales sur tablette tactile," Master's thesis, Faculté des sciences du sport, DIJON, 2015.
 - [26] J.-F. Susini, O. Pons, and N. Guedin. (2015) *Danse-doigts*. [En ligne]. Disponible : <http://jeanferdysusini.free.fr/motricite.html>