



HAL
open science

Permettre l'autonomie dans l'activité de programmation par blocs pour des enfants non-voyants

Ken Andriamahery-Ranjalahy, Philippe Truillet

► To cite this version:

Ken Andriamahery-Ranjalahy, Philippe Truillet. Permettre l'autonomie dans l'activité de programmation par blocs pour des enfants non-voyants. 12ème Conférence Handicap 2022 : Humaines et artificielles, les intelligences au service du handicap, Institut Fédératif de Recherche sur les Aides Techniques pour personnes Handicapées (IFRATH), Jun 2022, Paris, France. pp.155-160. hal-03716993

HAL Id: hal-03716993

<https://hal.science/hal-03716993v1>

Submitted on 7 Jul 2022

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.

Permettre l'autonomie dans l'activité de programmation par blocs pour des enfants non-voyants

Ken Andriamahery-Ranjalahy
LLA-CREATIS, Université de Toulouse,
UT2J
Toulouse, France
ken.andria@univ-tlse2.fr

Philippe Truillet
IRIT, Université de Toulouse,
CNRS, UT3
Toulouse, France
Philippe.Truillet@irit.fr

Abstract— Au collège, les langages visuels de programmation tels que Scratch sont utilisés en raison de leur simplicité de manipulation et ceux-ci ont même intégré les enseignements généraux depuis la réforme de 2016.

Nous avons développé le prototype TaBGO, permettant à des enfants malvoyants ou non-voyants de concevoir et d'exécuter des programmes Scratch en autonomie ou en collaboration. Dans cet article, nous présentons le prototype en nous focalisant sur les feedbacks (retours utilisateur) multimodaux audio-tactiles liés à l'exécution des programmes.

Nous discuterons des perspectives selon deux axes : le premier s'organise autour des aspects logiciel d'une part afin de permettre l'adaptation des activités de programmation au collège pour les non-voyants et d'autre part, le second axe se concentre sur les outils à mettre en place et sur leurs usages en classe inclusive.

Mots-clés—programmation, construction d'algorithmes, objets tangibles, multimodalité, autonomie

I. INTRODUCTION

Depuis 2009, de nombreuses enquêtes professionnelles prévoient une augmentation importante du nombre d'offres d'emploi dans le domaine de l'informatique [1]. Cette croissance trouve sa source dans l'importance grandissante du numérique dans le monde actuel. En corrélation avec ces changements technologiques, l'informatique commence également à atteindre les jeunes apprenants et les contextes éducatifs liés. Il est important de souligner la réforme de 2016 en France, qui intègre les notions d'informatique au programme de collège. Afin d'introduire ces notions, les professeurs de collège utilisent des plateformes en ligne (comme Scratch, Blockly ou Alice par exemple), car elles sont faciles à comprendre et à utiliser pour les néophytes. En raison de leurs syntaxes simples, leurs associations entre couleurs et fonctions, et leurs tutoriels intégrés, ces plateformes et les approches associées sont faciles à prendre en main. Cependant, ces supports reposent essentiellement sur des interactions visuelles, qui restent inaccessibles pour les non-voyants.

Il est néanmoins important de souligner les bénéfices de ces activités pour les non-voyants au-delà de la nécessaire inclusion

des personnes à besoin spécifique. Premièrement, il est important de préciser que la période du collège est très importante dans le développement sensori-moteur des jeunes apprenants, particulièrement pour les non-voyants en raison des troubles associés à leur handicap qui sont susceptibles d'apparaître [2]. Une approche simple et intégrable en milieu scolaire peut alors représenter une opportunité pour les élèves non-voyants de participer à diverses activités susceptibles de stimuler leurs compétences motrices, sensorielles et de favoriser leurs développements sensori-moteurs [3]. Deuxièmement, ces activités de programmation peuvent également représenter des opportunités de stimuler la collaboration entre élèves voyants et non-voyants dans un contexte éducatif. Ces collaborations sont susceptibles de faciliter l'inclusion des personnes non-voyantes, notamment en milieu scolaire, dès le niveau collège [4].

Ainsi, le présent article aborde cette problématique : est-il possible de proposer un médium non-visuel pour les non-voyants, afin de leur permettre un accès facile à la plateforme Scratch au même titre que les voyants ? De plus, est-il possible d'intégrer cette solution dans un environnement éducatif afin de favoriser l'inclusion de ces populations ?

Pour répondre à ces interrogations, le présent document propose une interface utilisateur tangible (TUI) composée d'une valise contenant des blocs physiques Scratch, d'une webcam et d'un logiciel de reconnaissance. En connectant les blocs, n'importe quel utilisateur (voyant ou non-voyant) peut construire des algorithmes. Ces algorithmes sont ensuite analysés par la webcam et traduits par le logiciel en fichier .sb3 (extension de fichier exploitable par la plateforme Scratch). En utilisant cette interface physique, les utilisateurs construisent des algorithmes de façon tangible, comme ils l'auraient fait visuellement (avec un clavier et une souris) sur la plateforme Scratch.

Afin d'explicitier cette solution, le présent article expose d'abord un bref état de l'art des technologies qui ont influencé la réalisation du projet TaBGO (acronyme de "Tangible Blocks Go Online"). Ensuite, une présentation rapide du prototype TaBGO est proposée et détaillée selon trois axes : matériel,

logiciel et un dernier se concentrant sur les fonctionnalités additionnelles utiles à la réalisation des exercices.

II. ETAT DE L'ART

A. Communication non-visuelle et multimodalité

Le projet TaBGO s'inscrit dans le domaine des technologies d'assistance [5, 6]. Si ces technologies ne sont pas nouvelles pour les non-voyants – le système à points de Louis Braille date des années 1850 par exemple –, ce domaine connaît cependant une récente augmentation des travaux de recherches liés. Cette augmentation peut s'expliquer par les progrès dans le domaine des composants électroniques libres (open source), comme Arduino, BeagleBone et Raspberry Pi. Ces compagnies proposent de la technologie accessible en matière de prix et de simplicité d'usage. Pour cette raison, ce type de technologie est de plus en plus utilisé dans les recherches universitaires depuis les vingt dernières années [7] qui peuvent aussi permettre de développer de nouvelles interfaces accessibles faciles à implémenter. En utilisant des blocs Peltier ou des buzzers vibrants, il est ainsi possible de construire une interaction non-visuelle, adaptée pour des utilisateurs non-voyants [8].

Cette interaction non-visuelle vise à être aussi efficace que l'interaction visuelle, mais ce but peut être difficile à atteindre, car les sens humains ne sont pas équivalents en matière de vitesse de transmission et de réflexes [9]. Néanmoins, il est possible d'approcher cette efficacité en utilisant la multimodalité en tant que ligne directrice principale [10] en utilisant des stimuli audio et tactiles par exemple. Cette utilisation de multiples modalités s'est révélée efficace si elle est correctement mise en place, notamment en évitant les effets délétères tels que ceux causés par le phénomène de redondance par exemple [11]. Pour être correctement compris, chaque stimulus de chaque modalité doit encoder des informations complémentaires, et non pas des informations redondantes. De plus, la zone ciblée par ces stimuli non-visuels revêt une importance particulière. Par exemple, les stimuli tactiles ciblent souvent les bras et les mains car ces zones sont particulièrement sensibles aux vibrations et aux textures [12]. La multimodalité au sein des technologies d'assistance peut être remarquée dans le projet ReacTable [13] par exemple : un instrument de musique en forme de table, qui associe des cubes (retours tactiles) et des procédés de *sound design* (retours audio) dans un but de création musicale.

Les retours audios liés au langage sont déjà employés au sein de nombreuses technologies d'assistance. On peut par exemple mentionner les technologies centrées autour de la synthèse vocale. Un tel procédé est également utilisé dans de nombreux lecteurs d'écran pour les non-voyants, comme JAWS¹ ou NVDA². Dans le domaine de l'informatique, la synthèse vocale est utilisée pour aider les développeurs non-voyants. Par exemple Javaspark [14, 15], Emacspeak [16] et CAITLIN [18] sont des environnements de développement (IDE) qui intègrent la synthèse vocale dans leur éditeur de texte et permettent leur accessibilité pour les non-voyants.

Les retours audios utilisant des stimuli audios courts – ou earcons [19, 20] – sont souvent superposés à des stimuli visuels : les exemples sont nombreux dans les études dans le domaine

de l'audiovisuel [21]. Ces stimuli audio sont souvent basés sur les principes de consonances et dissonances, définies dans un contexte tonal et/ou psychoacoustique [22]. Par exemple, les stimuli de validation emploient des intervalles ou des harmoniques consonants, alors que les stimuli négatifs (comme les messages d'erreur par exemple) utilisent des intervalles ou des harmoniques dissonants. Le sens de la mélodie peut également être associé avec les informations correspondantes ; une mélodie ou un intervalle ascendant est souvent perçu(e) comme plus engageant(e) qu'une mélodie ou intervalle descendant [23]. Une telle logique peut être employée dans la construction d'une interaction non-visuelle efficace, en utilisant des earcons en tant que stimuli additionnels.

Enfin, les retours tactiles sont également de plus en plus utilisés dans la communauté des technologies d'assistance, notamment dans le cas de cartes et de navigateurs internet [24] rendus accessibles pour des non-voyants en utilisant des vibrations. Des stimuli tactiles comme des textures (TeslaTouch) [25] ou des messages thermiques [26] peuvent également être intégrés. L'utilisation d'une interaction haptique peut représenter une opportunité pour les non-voyants d'utiliser leurs capacités motrices, surtout si celle-ci est intégrée à un contexte éducatif.

B. Contexte éducatif

La plateforme Scratch (projet du MIT) est utilisée en tant qu'outil éducatif pour faciliter l'apprentissage de l'informatique [27] mais il est important de noter que ce n'est pas la seule : ces outils sont fréquemment dérivés de jouets mécaniques. Ces artefacts représentent alors d'excellents exemples de TUI (Tangible User Interface) intégrables en milieu scolaire. Par exemple, on peut citer le Bee-bot [28] qui est un contrôleur en forme d'abeille (associé aux logiciels "*Lady Beetle*" et "*World of Sounds*"), dont l'une des principales fonctions est d'aider les élèves à programmer des séquences musicales.

Les compétences de programmation sont également introduites dès l'école primaire, comme le montre le projet Torino [29]. Ce dernier se base sur des unités rondes que l'on connecte afin de construire des algorithmes. De par cette connectique particulière, il est possible d'affirmer que la logique utilisée par Microsoft est proche de celle utilisée au sein de Scratch et de Blockly. La logique centrée autour des blocs peut également être remarquée dans certaines TUI en utilisant des blocs semblables à ceux de LEGO et une grille associée [30] ou en utilisant des blocs pour créer et s'échapper de labyrinthes basés sur des algorithmes [31]. L'utilisation simultanée de la multimodalité (stimuli audio et tactiles) et de TUI peut être également soulignée dans toutes ces plateformes.

Il convient alors de noter que d'autres projets plus analogues à des androïdes ont également été adaptés pour des activités informatiques car les mouvements et actions haptiques sont faciles à programmer. Le Roamer, un robot en forme de tortue, est un exemple représentatif [32]. Cette approche basée sur les androïdes permet de traduire des algorithmes en actions et en stimuli tactiles. Ainsi, cette traduction ne dépend donc pas de

¹ <https://www.freedomscientific.com/products/software/jaws>

² <https://www.nvaccess.org/download>

stimuli visuels, et peut être implémentée dans la construction d'une communication non-visuelle multimodale.

Au niveau du lycée, beaucoup de technologies d'assistance se sont concentrées sur les plateformes en ligne, et particulièrement sur Blockly. Blocks4All [33] présente une IDE sensiblement "augmentée" en utilisant une tablette Android (qui émule alors différentes textures). AccessibleBlockly [34] permet d'avoir un accès non-visuel à chacun des modules Blockly (ArduBlockly, OzoBlockly et BlockyTalky [35]) grâce à des retours audio.

Les chercheurs se sont également penchés sur l'accessibilité de Scratch pour les utilisateurs non-voyants. Ces recherches représentent des influences majeures pour le présent projet [36], utilisant les concepts de TUI et de multimodalité, de même que l'utilisation de composants électroniques *open source* (arduino). Parmi ces influences, il convient de citer la valise Accessi-DV [37] basée sur des blocs tangibles analogues à ceux utilisés par Scratch ; ainsi que le prototype CodeBox64 [38], un contrôleur Scratch inspiré des manettes de jeu, utilisant la technologie arduino.

Ces prototypes favorisent l'accès des non-voyants à la plateforme Scratch mais nécessitent l'intervention d'une tierce personne voyante pour contrôler et exécuter les programmes. Par exemple le prototype CodeBox64 utilise la technologie arduino, nécessitant une personne dotée de compétences en soudure et en prototypage électronique afin d'être utilisée. D'un autre côté, la valise Accessi-DV ne requiert pas de soudure, mais en revanche, elle ne permet pas non plus de communiquer directement avec la plateforme Scratch. Cette communication ne peut alors se faire que par la reproduction de l'algorithme par une personne voyante, directement sur la plateforme en ligne. Enfin, si la programmation débranchée (« *unplugged coding* ») représente une solution qui peut s'avérer utile pour l'apprentissage de l'informatique, celle-ci reste limitée par l'impossibilité d'accéder aux ressources numériques en ligne.

Notre projet s'inspire énormément de ces prototypes et l'un des principaux objectifs de notre travail vis-à-vis de ces technologies est de surmonter les obstacles sus cités liés à l'autonomie. Cette amélioration de l'autonomie se base sur la construction d'une communication non-visuelle efficace, un autre objectif majeur de ce projet.

Grâce à cet état de l'art, il a été possible de souligner quelques caractéristiques-clés qui ont favorisé la création de solutions viables afin de rendre accessibles les notions d'informatique aux collégiens (voyants et non-voyants). Ces caractéristiques incluent l'utilisation d'une interaction non-visuelle, basée sur des stimuli multimodaux (stimuli audio et tactiles). Cette communication est rendue possible en implémentant de la technologie libre, peu coûteuse et accessible pour les communautés associées aux contextes éducatifs (étudiants, chercheurs, enseignants). D'autres caractéristiques incluent également l'intégration en milieu scolaire (en utilisant des TUI en forme de jouets), et la nécessité d'une solution

autonome. Ces éléments-clés représentent des lignes directrices fortes pour la solution implémentée par le projet TaBGO

III. LE PROTOTYPE TABGO

Le prototype TaBGO (Tangible Blocks Go Online)³ comporte trois grandes parties (voir Figure 1) : des blocs Scratch tangibles (inspirés par la plateforme Scratch online et le projet Accessi-DV), une webcam et un logiciel associé.

Ces trois parties interagissent entre elles comme montré dans la figure 2 ci-dessous :

- L'utilisateur construit des algorithmes en connectant les blocs ;
- Les blocs sont ensuite reconnus via une webcam et notre solution traduit l'algorithme tangible en fichier .sb3 (extension de Scratch),
- Le fichier est généré automatiquement et exploité pour effectuer l'exécution multimodale de l'algorithme.

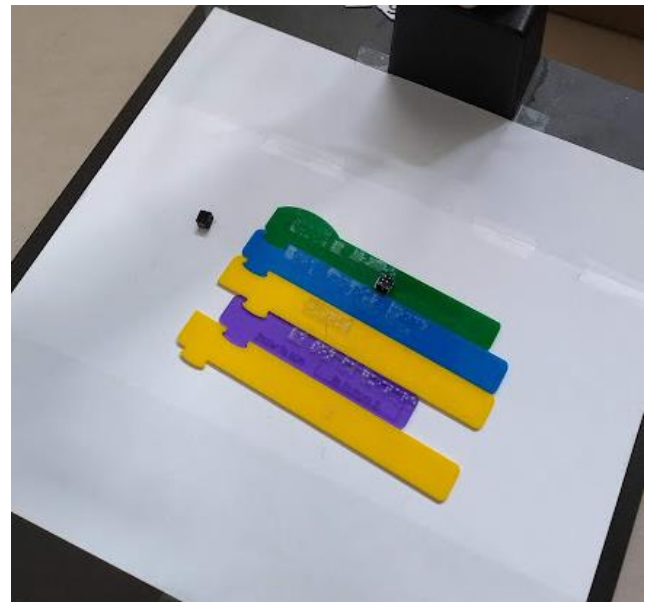


Fig. 1. : Vue générale du prototype physique TaBGO.

Le prototype TaBGO peut être détaillé en se concentrant sur trois de ses aspects fondamentaux : les composants tangibles (hardware), les composants logiciels (software) et l'équipement additionnel inclus.

A. Partie matérielle

Le premier aspect fondamental du prototype TaBGO dont il est question dans cet article est en lien avec les composants tangibles.

Le prototype actuel est composé de blocs tangibles en plexiglas, ayant une hauteur de 3,5 mm pour une longueur de 28,5 mm et une largeur de 7,5 mm de large. Ces dimensions ont été spécialement choisies afin de rendre les blocs faciles à agripper tout en restant suffisamment solides pour être maniés

³ Le code de notre prototype est accessible ici : <https://github.com/truillet/tabgo>

par des collégiens. Les blocs partagent les fonctions et les couleurs de leurs équivalents sur la plateforme Scratch. Les blocs peuvent être connectés grâce à leur forme rappelant celle des puzzles. Ce système de connexion vise à reproduire la syntaxe et les fonctionnalités utilisées dans la plateforme Scratch. Dans le cas de boucles conditionnelles ou booléennes, une cordelette est intégrée aux blocs afin de permettre un assemblage facile de ceux-ci, ainsi qu'une compréhension aisée des notions liées.

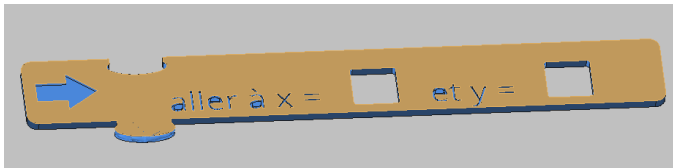


Fig. 2. exemple de modélisation 3D d'un bloc utilisé par TaBGO

Chaque bloc possède des inscriptions braille sur son recto, une traduction des fonctions permises par les blocs. Ces marques en braille représentent un premier pas vers une communication non-visuelle. De plus, certains blocs disposent de trous (cf. Figure 2) afin d'intégrer des cubarithmes (dés en braille) en leur sein. Ceux-ci peuvent être configurés afin d'afficher une valeur et associer celle-ci aux blocs. Grâce à ces améliorations, ces blocs peuvent facilement être lus et maniés par les non-voyants.

Dans le langage Scratch, le code couleur est directement lié aux catégories des blocs (opérateurs, musique, synthèse vocale, etc.). Celui-ci a été pris en compte lors de la construction des blocs tangibles. Cependant, ce code couleur repose sur des stimuli visuels inaccessibles pour les non-voyants. Afin de pallier ce problème, des gravures ont été utilisées afin d'associer chaque catégorie de blocs (et donc chaque fonction) à des stimuli tactiles. De plus, les blocs sont contenus dans une valise à six compartiments. La valise souligne les différentes catégories, et aide les utilisateurs (voyants ou non-voyants) à être plus organisés et efficaces dans la construction d'algorithmes.

Étant donné que les retours utilisateur tiennent une place importante dans la communication non-visuelle suggérée, il est important de souligner le parti pris du projet de se concentrer sur les extensions Scratch. Ces catégories additionnelles de blocs comprennent les extensions Scratch Music et Scratch Text-to-Speech (TTS, synthèse vocale), et permettent d'utiliser des retours audios pour les collégiens, qu'ils soient voyants ou non-voyants. De plus, le projet utilise également l'extension `micro:bit`⁴, qui se base sur une carte électronique à bas coût qui permet d'implémenter des retours haptiques. Toutes ces extensions permettent au projet d'implémenter simplement et rapidement des stimuli multimodaux, afin de construire une communication non-visuelle efficace.

B. Partie logicielle

En plus de ces composants physiques, notre solution comprend également un composant logiciel dont les fonctions principales comprennent l'analyse des photographies prises par la webcam et la traduction de cette analyse en un fichier `.sb3` (voir Figure 3)

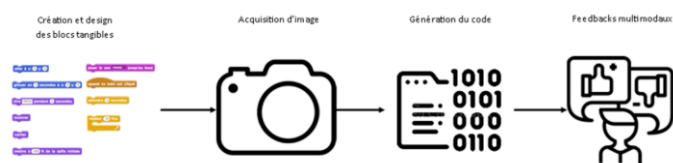


Fig. 3. Schéma résumant le processus de reconnaissance de TaBGO

Ce processus d'analyse détecte la fonction et la configuration de chaque bloc, du texte en braille et des cubarithmes. Ces analyses permettent une détection précise des algorithmes construits, et permettent donc une traduction robuste vers un fichier `.sb3`.

Le processus de traduction est la deuxième partie de l'aspect logiciel du prototype. Celle-ci représente un moyen efficace pour les utilisateurs non-voyants de bénéficier de Scratch de manière totalement autonome. Un fichier `.sb3` (extension de Scratch, résultat du processus TaBGO) peut facilement être ouvert dans la plateforme Scratch afin que l'utilisateur exécute son algorithme autant de fois qu'il/elle le souhaite. L'accès facile aux extensions non-visuelles permet à l'utilisateur de comprendre plus aisément si l'algorithme qu'il/elle construit est conforme à ce qui est attendu (vis-à-vis de l'activité organisée). L'installation d'une boucle de feedback efficace est particulièrement utile dans un contexte éducatif, étant donné que cette boucle conforte l'apprenant dans une méthode d'essai/erreur, et lui permet ainsi de devenir de plus en plus familier avec les notions et les syntaxes de programmation utilisées.

Ces boucles de retours nécessitent d'être rapides afin d'être efficaces, mais ne sont volontairement pas intégrées dans un processus en temps réel au sein du prototype. En effet, cet usage présente de nombreux désavantages, comparé aux possibilités qu'elle permet. Tout d'abord, l'utilisation du temps réel est généralement gourmande en ce qui concerne les ressources disponibles, et donc nécessite un ordinateur suffisamment puissant pour pouvoir en bénéficier. Étant donné que les classes françaises n'ont pas toutes des ordinateurs récents et/ou puissants, l'utilisation du temps réel peut ne pas être adaptée. De plus, ces ressources technologiques ne sont pas nécessairement accessibles pour tous les élèves, et peuvent ainsi représenter un frein au niveau de l'accessibilité, ce qui est antithétique à l'objectif du projet TaBGO. Ensuite, l'utilisation du temps réel peut s'avérer inadaptée pour les utilisateurs néophytes de Scratch (qu'il s'agisse de la version tangible ou en ligne) car celle-ci peut perturber l'utilisateur par excès d'information. Cet excès d'information contribuerait alors à augmenter la charge cognitive expérimentée [40, 41], et serait à même de rendre l'approche Scratch plus difficile à comprendre et à intégrer pour des collégiens. Ainsi, et bien que l'utilisation du temps réel puisse être intéressante, celle-ci doit être correctement implémentée afin d'être facilement appréhendée, surtout si elle s'ajoute à l'utilisation de la multimodalité. Enfin, l'utilisation du temps réel peut présenter des problèmes de compatibilité quand

⁴ <https://microbit.org>

il est question d'utiliser des équipements additionnels, tels que les lecteurs d'écran ou les cartes micro:bit.

C. Fonctionnalités additionnelles

L'implémentation de fonctions additionnelles représente le troisième aspect du prototype TaBGO. Afin de construire une interaction non-visuelle efficace, le présent article a déjà souligné l'importance des stimuli multimodaux et des boucles de retours associées. Pour permettre l'implémentation de stimuli audio et tactiles, les extensions Scratch sur lesquelles le projet se concentre comprennent Scratch Music, Scratch Text-to-Speech et micro:bit. Ces retours permettraient d'employer de nouvelles modalités d'accès en ce qui concerne l'approche Scratch de l'informatique, et ce indépendamment du fait que l'utilisateur soit voyant ou non-voyant.

Ces retours multimodaux peuvent être particulièrement utiles dans le cas de l'avatar Scratch représenté par un dessin de chat. Cet avatar ne permet pas sa manipulation par les non-voyants. Néanmoins, les épreuves du Diplôme National des Collèges incluent souvent un trajet à faire effectuer par l'avatar ou une figure géométrique à dessiner : l'accès à la fenêtre de l'avatar est donc particulièrement primordial dans ce cadre.

En effet, une solution possible serait de faire ressentir aux utilisateurs une sensation d'espace audio sans utiliser de stimuli visuels. Les stimuli audios et leur spatialisation peuvent alors s'avérer riches de possibilités. Avec l'utilisation de la stéréophonie, il est déjà possible de donner une direction (gauche-droite) aux stimuli audios afin de recréer cette sensation d'espace. Par exemple en utilisant une paire d'écouteurs, un mouvement de l'avatar Scratch vers la droite peut être associé à un bip sonore à droite de sorte à ce qu'un non-voyant puisse percevoir ce mouvement. Il est également possible de recréer une sensation de profondeur (distance de la source par rapport à l'auditeur) en utilisant des procédés de réverbération. En utilisant une réverbération numérique, les stimuli audios ont la possibilité de se déployer sur deux axes (gauche/droite, devant/derrière), qui peuvent être associés aux axes horizontaux et verticaux de l'avatar. Afin d'augmenter cette sensation d'espace sonore perçu, nous avons pris le parti d'utiliser des procédés binauraux : ces procédés s'appuient sur des théories et des effets psycho-acoustiques afin de recréer la sensation d'un son 3D avec seulement une paire d'écouteurs.

Cette approche binaurale peut donc représenter un moyen facile et accessible de préciser la communication audio utilisée. Avec les éléments-clés cités plus haut en tant que lignes directrices, un logiciel de spatialisation audio a été développé en utilisant Max/MSP qui permet d'ajouter de la profondeur, une direction et une localisation aux stimuli audios utilisés au sein de la plateforme Scratch. Il est important de préciser que le logiciel de spatialisation intervient en sortie de chaîne audio et non pas pendant l'analyse, pour les raisons de vitesse et d'efficacité mentionnées plus haut. Grâce à ce placement dans la chaîne, les problèmes éventuels de compatibilité avec Scratch sont évités : l'outil de spatialisation s'interface directement avec le périphérique de sortie sans utiliser la plateforme Scratch.

Grâce à diverses commandes intégrées dans l'interface utilisateur, les créateurs d'activités basées sur Scratch ont la possibilité de prévoir des parcours sonores à réaliser par les élèves via la plateforme en ligne. Vis-à-vis du prototype TaBGO, il est alors possible d'associer des commandes de spatialisation aux blocs détectés, de sorte à ce que le parcours soit entièrement traduit en même temps que l'algorithme. Simultanément à la traduction de l'algorithme tangible en fichier .sb3, des instructions de spatialisation sont produites, qui contrôlent le logiciel de spatialisation via un bus logiciel. Ce logiciel de spatialisation représente donc un moyen facile pour les enseignants et éducateurs de programmer des retours audios adaptés : il est alors possible pour eux de construire des activités Scratch se basant sur l'avatar tout en restant accessible pour les non-voyants.

En résumé, notre solution, avec ses nombreux composants, vise à une approche alternative non-visuelle à la plateforme Scratch en proposant une communication multimodale et en utilisant des outils intégrables en contexte éducatif.

CONCLUSION

En s'appuyant sur des recherches connexes antérieures dans le milieu des technologies d'assistance, le projet TaBGO propose une solution d'accessibilité de Scratch en trois parties : des blocs tangibles pour construire les algorithmes, une webcam pour identifier les blocs et un logiciel de traitement associé pour produire un fichier scratch.

Étant donné que le principal objectif est la construction d'une interaction non-visuelle efficace, le projet a mis l'accent sur les stimuli audios et tactiles, en intégrant notamment des technologies telles que la spatialisation audio pour représenter les déplacements dans l'espace.

Tout au long de cet article, le projet a souligné l'utilisation de stimuli multimodaux en utilisant de la technologie open source et des TUI.

La phase expérimentale est en cours et a pris beaucoup de retard dû à la pandémie de COVID. Des pré-tests et les tests en milieu scolaire ont été réalisés, et ont révélé des résultats encourageants et des retours utiles à propos du prototype et du protocole expérimental à mettre en œuvre.

Il est néanmoins important de souligner les possibilités qui sont encore à explorer en ce qui concerne la construction de communications non-visuelles efficaces : relecture rapide des algorithmes produits par exemple et l'utilisation en complète autonomie de notre prototype de la conception à l'exécution et modification des programmes.

REMERCIEMENTS

Le présent projet et les recherches associées sont soutenus par l'association française UNADEV (Financement UNADEV 2019.49).

REFERENCES

- [1] Das, M., Marghitu, D., Jamshidi, F., Mandala M. and Howard A. (2020). Accessible Computer Science for K-12 Students with Hearing Impairments. 10.1007/978-3-030-49108-6_13.
- [2] Flammant, J. (2016). De l'oeil au Regard [Monographie, CSEMSH].
- [3] Hatwell, Y, Le développement perceptivo-moteur de l'enfant aveugle, *Enfance* 2003/1, Vol. 55, pages 88 à 94, Paris, 2003.
- [4] Metelja O., Thieme A., Brûlé, E, Benett C., Serrano M., Jouffrais Ch., Toward classroom experiences inclusive of students with disabilities, vol XXVI.1, ACM Interactions, january-february 2019.
- [5] Truillet Ph., « L'informatique, pour un monde plus accessible ». *Revue* 1024, Dossier Accessibilité informatique, 2020, numero 15, Article 7, pp. 45-57. (hal-03116648)
- [6] Senjam S., Assistive technology for students with visual disability: Classification matters, *Kerala Journal of Ophthalmology*, 2019, Bombay, India.
- [7] Archambault, D., Interaction et usages des modalités non visuelles, accessibilité des contenus complexes, (Thèse) Université Pierre et Marie Curie - Paris VI, Paris 2010.
- [8] Andriamahery Ranjalaly K. H., La PACMAM : une approche multimodale accessible des supports de composition musicale, *Journal de Recherche en Education Musicale* vol.12 n°2-2021, Février 2022.
- [9] Dale E. 1969. Cone of experience, in *Educational Media: Theory into Practice*. Wiman RV (ed). Charles Merrill: Columbus, Ohio
- [10] Adami, E., "Multimodality", as published in O. Garcia, N. Flores and M. Spotti (2016) *Oxford Handbook of Language and Society*. Oxford: Oxford University Press
- [11] Le Bohec, O. & Jamet, É. (2005). Les effets de redondance dans l'apprentissage à partir de documents multimédia [1]. *Le travail humain*, 68, 97-124. <https://doi.org/10.3917/th.682.0097>
- [12] Hoffmann, R., Valgeirsdottir, V. V., Johannesson, O. I., Unnthorsson, R., & Kristjansson, A. (2018). Measuring relative vibrotactile spatial acuity: Effects of tactor type, anchor points and tactile anisotropy. *Experimental Brain Research*, 236, 3405-3416. <https://doi.org/10.1007/s00221-018-5387-z>
- [13] Jordà, S., Geiger, G., Alonso, M., & Kaltenbrunner, M. (2007). The reacTable : Exploring the Synergy between Live Music Performance and Tabletop Tangible Interfaces.
- [14] Cheong, C.; Andrew Burge (ed.), Coding without sight: Teaching object-oriented java programming to a blind student », 8th Annual Hawaii International Conference on Education, Honolulu, Hawaii, 7-10 January 2010, pp. 1-12.
- [15] Francioni, J. M., Matzek, S. D. & Smith, A. C., A Java Programming Tool for Students, with Visual Disabilities, ASSETS '00, November 2000, Arlington, Virginia, USA.
- [16] Raman, T.V., Michael J. Tauber (Ed.) « Emacspeak—A speech interface » *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'96)*, p.66-71, ACM, New York, USA.
- [17] Vickers, P. , Alty, J. "Musical Program Auralisation: A Structured Approach to Motif Design", *Interacting with computers*, n°14, p.457-485, october 2002, USA
- [18] Owen Ch. B., Teaching Modern Object-Oriented Programming to the Blind: An Instructor and Student Experience, Michigan State University; Sarah Coburn, Michigan State University, Ms. Jordyn Castor
- [19] Collins, Karen, Game Sound, MIT Press, Cambridge, Massachusetts, Etats-Unis, 2008
- [20] Lequai, Cl., Muse #5 « Industries Culturelles Et Créatives : Musiques & Jeux Vidéo », conférence de 2018, Toulouse, France.
- [21] Johannesson, G., Auditory Icons vs Earcons in Games: What Makes the Most Efficient Nonverbal Sound Cue? 2016, Lulea
- [22] Terhardt, E. "The Concept of Musical Consonance: A Link between Music and Psychoacoustics" *Music Perception: An Interdisciplinary Journal*, Vol. 1, No. 3, Dedicated to Helmholtz, Printemps 1984, pp. 276-295, University of California Press, Californie, Etats-Unis
- [23] Meredith, D., "Statistical Regularities in Melodic Phrases: Effects on Aesthetic Ratings" (2018). CUNY Academic Works., https://academicworks.cuny.edu/gc_etds/2445
- [24] Bigham, J. P., Kane, S. K. "Tracking @stemxcomet: Teaching Programming to Blind Students via 3D Printing, Crisis Management, and Twitter", SIGCSE '14, Mars 2014, Atlanta, Etats-Unis.
- [25] Bau, O., Israr, A., Poupyrev, I., Harrison, C. (2010). Teslatouch : Electro vibration for Touch Surfaces. *UIST'10*. <https://doi.org/10.1145/1866029.1866074>
- [26] Sato, K., & Maeno, T. (2012). Presentation of Sudden Temperature Change Using Spatially Divided Warm and Cool Stimuli (p. 468). https://doi.org/10.1007/978-3-642-31401-8_41
- [27] Maloney, J., Resnick, M., Rusk, N., Silverman, B. and Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*. 10. 16. 10.1145/1868358.1868363.
- [28] Jaskova, L., Kaliakova, M., Programming Microworlds for Visually Impaired Pupils, Conférence de Constructionism 2014, 2014, Vienne, Autriche.
- [29] Morrison, C., Villar, N., Hadwen-Bennett, A., Regan, T., Cletheroe, D., Thieme, A. & Sentance, S.. (2019). Physical Programming for Blind and Low Vision Children at Scale. *Human-Computer Interaction*. 1-35. 10.1080/07370024.2019.1621175.
- [30] Utreras, E., Pontelli, E.. (2020). Design of a Tangible Programming Tool for Students with Visual Impairments and Low Vision. 10.1007/978-3-030-49108-6_22.
- [31] Wang, D, Zhang, Ch, Wang, H.. (2011). T-Maze: A tangible programming tool for children. 127-135. 10.1145/1999030.1999045.
- [32] Renaud P., Virey M., Le Roamer: un robot déjà ancien au service d'apprentissages bien actuels, dans la nouvelle revue de l'adaptation et de la scolarisation (N°52), pp 231-239, INSHEA, 2010
- [33] Ladner, R. E., Milne, L. R., Blocks4All: Overcoming Accessibility Barriers to Blocks Programming for Children with Visual Impairments, CHI 2018, Avril 2018, Montréal Canada
- [34] Ludi S., Merchant W.; Simpson, J., Exploration of the Use of Auditory Cues in Code Comprehension and Navigation for Individuals with Visual Impairments in a Visual Programming Environment, ASSETS'16, 23-26 October 2016, Reno, Nevada, USA
- [35] Deitrick, E.; Sanford, J., Shapiro, B. R., BlockyTalky: A Low-Cost, Extensible, Open Source, Programmable, Networked Toolkit for Tangible Creation, IDC2014, 17-20 juin 2014, Aarhus, Danemark
- [36] Andriamahery-Ranjalaly K, Berquez L., Jessel N., Truillet Ph., TaBGO: towards accessible computer science in secondary school. 23rd International Conference on Human-Computer Interaction (HCI International 2021), Jul 2021, virtual place, United States, pp. 12-24. (hal-03168307)
- [37] Boissel S., Mallette Accessi DV Scratch « Scratch débranché en braille et gros caractères », dans la nouvelle revue de l'adaptation et de la scolarisation (N°77), pp 183-192, INSHEA, 2017
- [38] Wagner A., Wang Z., Evaluating a Tactile Approach to Programming Scratch, ACMSE 2019, Avril 2019, Kennesaw, USA
- [39] Metin, S. (2020). Activity-based unplugged coding during the preschool period. *International Journal of Technology and Design Education*. 10.1007/s10798-020-09616-8.
- [40] Puma, S., Matton, N., Paubel, PV. et al. Cognitive Load Theory and Time Considerations: Using the Time-Based Resource Sharing Model. *Educ Psychol Rev* 30, 1199–1214 (2018). <https://doi.org/10.1007/s10648-018-9438-6>
- [41] Sweller J. et al., Cognitive Load Theory, Explorations in the Learning Sciences, *Instructional Systems and Performance Technologies* 1, DOI 10.1007/978-1-4419-8126-4_6, © Springer Science+Business Media, LLC 2011