

EXPLORER LE RÔLE DE LA COMPRÉHENSION DES CONCEPTS MATHÉMATIQUES DANS LE CONTEXTE DU DÉBOGAGE DANS LE DESIGN EN INGÉNIERIE ASSOCIÉ À L'INFORMATIQUE PHYSIQUE ET AU CODAGE INFORMATIQUE

| KAMBA* JACQUES ET FREIMAN** VIKTOR

Résumé | Au Nouveau-Brunswick, au Canada, plusieurs écoles offrent aux élèves la possibilité d'apprendre à coder et d'utiliser des appareils informatiques physiques dans le cadre d'activités de design en ingénierie. Le présent article analyse spécifiquement le rôle de la compréhension des concepts mathématiques lors du débogage des erreurs de code informatique lors de la création d'artefacts d'ingénierie. Les résultats de l'article montrent plusieurs épisodes de travaux d'étudiants où des liens mathématiques sont présents explicitement ou implicitement.

Mots-clés : design en ingénierie, *débogage*, *codage*, appareils informatiques physiques (AIP) et modèle.

Abstract | In New Brunswick, Canada, several schools offer students the opportunity to learn to code and use physical computing devices as part of engineering design activities. The present article specifically analyzes the role of understanding mathematical concepts during debugging of computer code errors when creating engineering artifacts. The results of the article show several episodes of student work where mathematical connections are explicitly or implicitly present.

Keywords: Engineer design, *debugging*, coding, physical computing devices (PCD) and model.

Remarque : Afin d'alléger le texte de l'article et d'en faciliter la lecture, veuillez prendre note que le genre masculin sera utilisé pour désigner à la fois le masculin et le féminin, et ce, sans aucune discrimination.

I. ENJEUX SOCIÉTAUX LIÉS AUX USAGES DES RESSOURCES NUMÉRIQUES ET NON NUMÉRIQUES DANS LE DESIGN EN INGÉNIERIE DANS L'ÉCOLE

Dans notre contribution au même groupe de travail lors du Colloque 2018 (Freiman et Chiasson, 2018), nous avons repéré la tendance de complexification et de diversification de ressources sur le plan trans-, inter-, et disciplinaire. Les auteurs ont ainsi mis en évidence que les nouveaux types d'espaces, riches en technologies, dynamiques, mobiles munis des affordances de rétroaction instantanée créent des opportunités pour les enseignants d'enrichir le répertoire de stratégies de résolution de problèmes, de plus en plus complexes qui sont axées, entre autres, sur l'exploration, le bricolage (*tinkering*) et la pensée computationnelle (informatique) dans le contexte de création de défis de robotique par les élèves (Freiman et Chiasson, 2018). Cette tendance semble s'amplifier, depuis 2016, avec la croissance de labos de fabrication numérique (aussi connus comme *makerspaces*) dans les écoles du Nouveau-Brunswick (N.-B.) (Canada), marquée par un enthousiasme notable de la part de certains enseignants qui voient leurs salles de classe se transformer en véritables labos créatifs pour engager les élèves dans un apprentissage intégré des sciences, technologies, ingénierie et mathématiques (STIM) en les initiant à explorer une variété de technologies de pointe (Freiman, 2020). Afin d'aider les enseignants à mettre en place des activités de fabrication pour stimuler l'intérêt des élèves pour les STIM, plusieurs

* Université de Moncton – Canada – cjk2904@umoncton.ca

** Université de Moncton – Canada – viktor.freiman@umoncton

associations à but non lucratif ou caritatives, telles que Labo Créatifs (Brilliant Labs en anglais) (<https://laboscreatifs.ca/>) ou Place aux compétences (<https://pacnb.org/fr/>), mettent à la disposition des enseignants trois types de ressources : Les outils numériques tels que des environnements de *codage informatique*, de *micro:bit*, des servomoteurs et des hélices en plus de fournir les matériaux usuels tels que des cartons, de la peinture et de la colle. Enfin, de personnes-ressources provenant de ces organismes se portent volontaires pour assister les enseignants et leurs élèves dans leurs activités de création informatisée. Les labos créatifs deviennent ainsi un objet d'intérêt de recherche, notamment au N.-B., au sein de l'équipe du Réseau des partenaires CompeTI.CA qui s'est intéressée, entre autres, aux compétences du 21^e siècle, à l'approche de tinkering (Furlong et Léger, 2022), au design en 3D (Freiman et Lingley, 2023) et au rôle de mathématiques ('M') dans les activités STIM aux labos créatifs (LeBlanc et al., 2022). Notre étude poursuit dans la même veine en s'intéressant particulièrement au *débogage* et ses liens avec les mathématiques lors d'une utilisation autonome par les élèves de diverses ressources matérielles et technologiques.

Les progrès de la technologie numérique accroissent la complexité du monde moderne et engendrent des enjeux sociétaux d'envergure dans presque toutes les sphères de la société moderne (p. ex. : l'éducation, la fabrication et l'économie). Les technologies numériques peuvent créer des incidents et des failles (p. ex. : des erreurs de code, des pannes informatiques et des mises à jour défectueuses) qui peuvent affecter des personnes humaines, des entreprises et la Société tout entière (George, 2024). Les risques et les vulnérabilités liés aux technologies numériques et à leurs usages appellent aux citoyens d'aujourd'hui la nécessité d'acquérir de nouveaux types d'habiletés qui pourraient leur permettre d'identifier et de trouver solutions aux erreurs de code et aux problèmes liés aux technologies. Parmi ces habiletés, plusieurs auteurs évoquent le *débogage* (Ismail et Hassan, 2019 ; Li et al., 2019). Le *débogage* est un processus qui concerne la correction des erreurs de code et la résolution de divers problèmes liés aux appareils informatiques physiques (AIP) et aux modèles physiques (Hennessy Elliott et al., 2023).

Dans notre étude, nous avons pu observer les élèves d'une classe de 4^e année du primaire (9-10 ans) travailler sur un projet interdisciplinaire partant du programme de science (exploration du monde animal), en effectuant les recherches sur un animal choisi et en présentant les résultats de leurs recherches au public (leurs écoliers et les parents). Entre autres, leur tâche plus spécifique consistait en création, en équipe de 3-4 élèves, d'une maquette d'animal en y intégrant un *micro:bit*, une carte micro-contrôleur programmable, ayant des capteurs et actionneurs intégrés (<https://microbit.org/fr/>).

Ainsi, plusieurs questions émergent dans ce contexte ; entre autres, quel est le type des erreurs de code et des problèmes liés aux AIP que les élèves rencontrent lors de la réalisation de ce projet ? Comment les élèves identifient-ils et corrigent-ils les erreurs de code ? De quelle manière la compréhension de mathématique pourrait-elle soutenir le *débogage* des erreurs de code et d'ajustement de maquettes physiques produites par les élèves ? Dans notre communication, on va tenter de répondre à ces questions à partir d'un travail de 3 élèves qui ont choisi de produire une maquette d'un tigre blanc dont une partie (les dents) ferait un mouvement de rotation de 90 degrés à l'aide d'un servomoteur intégré. Notons que pour ces élèves, tout comme pour leurs pairs, cette expérience avec le *codage informatique* et le *micro:bit* était la première (du moins, dans le contexte scolaire).

II. CADRE CONCEPTUEL : RESSOURCES DES DESIGN EN INGÉNIERIE ET LA GENÈSE INSTRUMENTALE DES ÉLÈVES

Dans les années 1990-2000s, une conceptualisation d'usage des outils informatiques prend de l'élan avec l'émergence de l'approche instrumentale en enseignement de mathématiques puisant dans les

idées de l'ergonomie cognitive (Rabardel, 1995) et dans l'approche anthropologique en didactique des mathématiques (Chevaillard, 1991). Dans sa contribution à l'EMF 2006, Artigue (2006) explique l'apport de chaque théorie : la première (ergonomie cognitive) permet de faire une distinction entre l'objet logiciel – l'artefact – et l'instrument pour celui qui en fait usage, tout en reflétant la complexité de processus individuels et sociaux de la genèse instrumentale tandis que la 2^e (approche anthropologique) qui apporte une dimension institutionnelle (apprentissage et enseignement, normes et valeurs,...) permettant de situer les genèses instrumentales individuelles au sein de genèses institutionnelles (artefact introduit dans le système éducatif ; transformation de l'artefact en instrument). (Artigue, 2006, p. 8).

Trouche (2005), à son tour, apporte de l'éclairage sur la fertilité de l'approche instrumentale pour ressortir le rôle (1) des artefacts dans les pratiques, les apprentissages et les enseignements mathématiques, à la fois comme l'expérience et l'assistant à l'activité humaine ; (2) du contexte dans lequel les artefacts peuvent devenir les instruments au cours de processus complexes, qui ont des dimensions individuelles et collectives ; et 3) de l'instrument, comme résultat d'un processus de transformation, par l'individu, de l'artefact (l'instrumentalisation) et d'un processus de structuration de l'activité (l'instrumentation).

Dans les années 2010-2020s, avec l'arrivée de nouveaux environnements d'apprentissage informatisés et de nouveaux outils à disposition des enseignants et des élèves, l'approche instrumentale semble gagner d'importance dans le contexte de l'enseignement des mathématiques. Par exemple, en mettant de l'emphasis sur le rôle de l'enseignant dans la genèse instrumentale des élèves (entre autres, le choix de logiciel à utiliser), Drijvers et al. (2014) ont repris le concept d'orchestration instrumentale de Trouche (2005) pour identifier différents types d'organisation de travail des élèves, dont un nommé *Work-and-walk-by* nous intéresse particulièrement (dans le contexte de labo créatif), car dans ce cas, les élèves travaillent seuls ou avec leurs pairs sur une tâche impliquant l'usage de technologies et l'enseignant se déplace dans la salle de classe en répondant aux questions des élèves et suivant leur progrès. À leur tour, Gueudet et al. (2014) y ajoutent l'utilisation autonome (sans intervention de l'enseignant) et l'utilisation accompagnée lorsque l'enseignant reste à côté des élèves qui ont plus de difficultés d'usage d'un logiciel et les aide plus explicitement dans leur travail.

Selon King et English (2016), le design en ingénierie est une activité qui consiste à produire des artefacts, à les tester et à les reconcevoir en fonction des tests précédents. Les activités d'apprentissage qui offrent aux élèves des défis d'ingénierie et des possibilités de conception sont considérées comme bénéfiques pour soutenir et développer davantage les processus de modélisation, de discours et d'échafaudage naturel dans lesquels les élèves collaborent avec leurs pairs pour résoudre des problèmes de la vie réelle (DeJarnette, 2012). Dans ces activités, les élèves développent leur compréhension par l'interaction et l'observation de leur environnement (Vygotsky, 1978). Les projets de design aident les élèves à voir les liens entre les concepts scientifiques et les solutions aux problèmes du monde réel (Sadler et al., 2014). Parmi les différents types de matériaux et de technologies disponibles pour soutenir les activités de design en ingénierie, il y a des environnements de *codage informatique*, des AIP (*micro:bit*) et des matériaux usuels tels que le carton.

Quant au processus de design, nous avons adapté les étapes de la pensée design proposées par Pruneau et al. (2019). Ainsi, lors de l'étape 1 (Figure 1) les élèves, ayant déjà construit un modèle physique de leur animal, commencent leur processus de conception d'un système informatisé de mouvement en choisissant une partie du corps de l'animal à dynamiser (observation-inspiration) et en imaginant comment cette partie bougerait et comment elle pourrait être automatisée (définition-synthèse-idéation). Dans un deuxième temps (étape 2) un prototype de système informatique physique est construit, y compris le *codage*, le transfert du programme vers un AIP et l'intégration dans la

maquette. L'étape 3 permet aux élèves d'effectuer un test pour présenter, finalement leur prototype final au public (étape 4). Au cours du processus de création et de test de leurs prototypes, les élèves ont été confrontés à un certain nombre de problèmes qui nécessiteront un *débogage*.

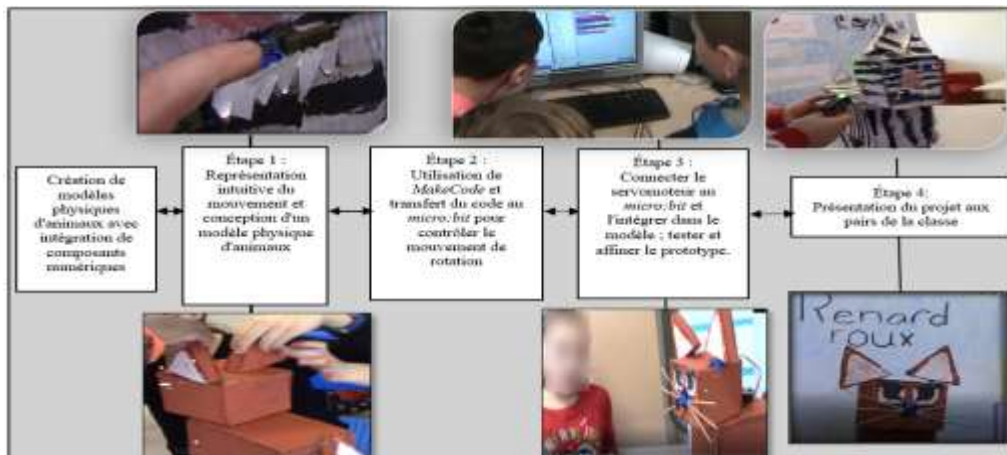


Figure 1 – Processus de design en ingénierie en 4 étapes

Le *débogage* est défini de différentes manières et sous plusieurs angles. Dans le cadre du présent article. Le *débogage* demeure une pratique de la pensée informatique (PI) (Grover et Pea, 2018), il est un processus cognitif qui consiste à identifier et à corriger des erreurs de codes et aussi à résoudre divers problèmes liés aux AIP (Hennessy Elliott et al., 2023), et aux artefacts physiques. Selon la littérature, il semble y avoir un lien étroit entre la PI et les mathématiques. Shute et al. (2017) expliquent que « la pensée mathématique se compose de trois parties : les croyances sur mathématiques, processus de résolution de problèmes et justification des solutions » (p. 4). Plusieurs compétences de la PI, telles que le séquençage (algorithmes), la modularité et le *débogage* pourraient être considérés comme « fondamentales » pour la compréhension mathématique par des élèves, dès leurs jeunes âges (4-5 ans) (Lavigne et al., 2020). Les auteurs soutiennent que lorsque les jeunes élèves sont engagés dans le *débogage* (ce qui signifie identifier les erreurs de code, décomposer les séquences d'un programme et tester le fonctionnement du programme), ils peuvent utiliser des connaissances mathématiques (comme utiliser une régularité, compter ou comparer) pour corriger les erreurs de code (Lavigne et al., 2020).

À cet égard, nous avons examiné quelques études susceptibles de révéler les liens entre l'apprentissage des mathématiques et le *débogage*. Par exemple, Shumway et al. (2021, p. 21) ont étudié les concepts de comptage, de raisonnement spatial, d'unités de mesure et d'opérations sont apparus comme des forces favorables ou défavorables lorsque des élèves ont débogué des programmes. Aussi, Riche et al. (2019) ont montré comment impliquer des élèves dans le processus de *débogage*. En effet, lorsque des élèves observent, émettent des hypothèses, modifient et testent, ils peuvent créer un contexte fructueux pour soutenir mutuellement l'apprentissage des mathématiques et le développement de la PI de manière intégrée. De la même manière, Kilhamn et al. (2021) ont étudié le potentiel de la programmation informatique, lorsqu'elle est intégrée au programme de mathématiques, pour soutenir l'exploration de différents concepts et processus mathématiques.

III. EXEMPLE DE DÉBOGAGE DES ERREURS DE CODE ET DE MAQUETTE

Le cas choisi pour notre communication présente le processus de *débogage* suivi par les élèves de 4e année qui ont été regroupés dans une équipe de trois personnes. Ils ont conçu et créé une maquette

d'un tigre blanc, une espèce très rare dans la nature. Nous avons fait des observations en classe lors de la réalisation du projet par les élèves avec l'enregistrement vidéo de quelques moments de leur travail. Nous avons également fait un entretien avec les élèves et leur enseignante à la fin du projet. L'analyse de données est réalisée en fonction de quatre étapes de la pensée design que les élèves ont suivi pour créer la maquette avec un système informatisé intégré.



(Source : https://commons.wikimedia.org/wiki/File:Tigre_d%27Asie_%C3%A0_robe_blanche.jpg)

Figure 2 – Modèle de l'animal pour imiter le mouvement naturel de mâchoires d'un tigre,

Étape de la création de la maquette : D'après nos observations, les élèves se sont engagés dans un processus de design où ils ont utilisé divers matériaux pour créer une maquette de tigre blanc (Figure 2). Au départ, les élèves ont choisi de dynamiser les mâchoires de la maquette qui devraient modéliser le mouvement naturel d'un tigre lui permettant d'ouvrir et de fermer sa bouche par le mouvement de servomoteur attaché au *micro:bit* programmé. À ce propos, E1 dit ceci : *On va bouger sa bouche.*

Les élèves se sont mis à discuter de comment créer la bouche sur la tête de leur maquette (l'image 2 de la Figure 3 ci-dessous). Une forme rectangulaire a ainsi émergé de leur discussion :

- E1 : Comment on va couper la bouche ? Comme un rectangle ?
- E2 : Oui, on va faire de même [il prend la feuille et fait le dessin pour montrer comment ils vont couper la bouche].
- E1 : Ça, c'est la tête, les yeux. Est-ce qu'on va couper la bouche comme ça ? [Il dessine une ligne droite sur la tête du dessin pour représenter la bouche, voir l'image 1 de la Figure 3].
- E2 : Non, plus couper en ligne [il ajoute deux lignes en largeur de la ligne droite de la bouche, voir l'image 2 de la Figure 3].
- E1 : Comment ? ça peut aller comme ça et comme ça [il dessine deux flèches pour montrer comment les dents vont monter et descendre. Voir l'image 3 de la Figure 3]
- E2 : Ça va descendre de même. On fait un rectangle [ils vont couper la bouche à la forme d'un rectangle].

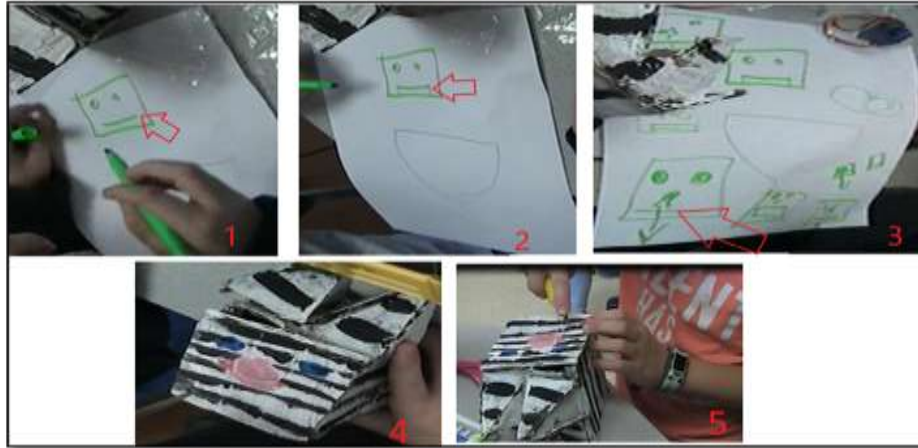


Figure 3 – Dessin de la bouche, idée intuitive de mouvement et coupure de bouche

Pendant que les deux élèves de cette équipe essaient de faire un sketch de la bouche (forme et position), leur collègue commence à découper le rectangle à l'aide d'un couteau (l'image 5 de la Figure 3 ci-dessus). D'ailleurs, l'image 3 de la même Figure 4, montre la représentation intuitive du mouvement de mâchoires (flèche vers le haut et vers le bas). Les élèves semblent déjà avoir des idées intuitives d'algorithme de mouvement contrôlé par le *micro:bit* : lorsque le bouton de démarrage A du *micro:bit* serait pressé. Les élèves se sont, par la suite, rendu compte que la réalisation d'un tel mouvement sera trop complexe, voire impossible. Ils décident donc de se limiter au mouvement plus simple de rotation de dents (de 90°). Les élèves devaient alors programmer le *micro:bit* pour que ce mouvement soit reproduit sur la maquette. La section suivante présente l'étape de *codage informatique* et les efforts de *débogage* des erreurs de code liées à ce programme.

Étape de la conception de programme informatisé : À l'aide d'une plateforme de *codage* MakeCode, les élèves ont tenté de créer les séquences du code informatique (blocs) opérant les mouvements des mâchoires de leur animal qui seront exécutés par des servomoteurs contrôlés par le *micro:bit*. La première version du programme contenait cinq blocs de code, comme le montre la Figure 4 : Le premier bloc signifie le démarrage du programme. Les deux blocs en couleur rouge déterminent la position de servo broche allant de 90 à 180° ainsi assurant une rotation de 90 degrés. Enfin les deux blocs bleus signifient les pauses entre les mouvements de servo broche (la pause de 10 000 ms, soit 10 secondes).

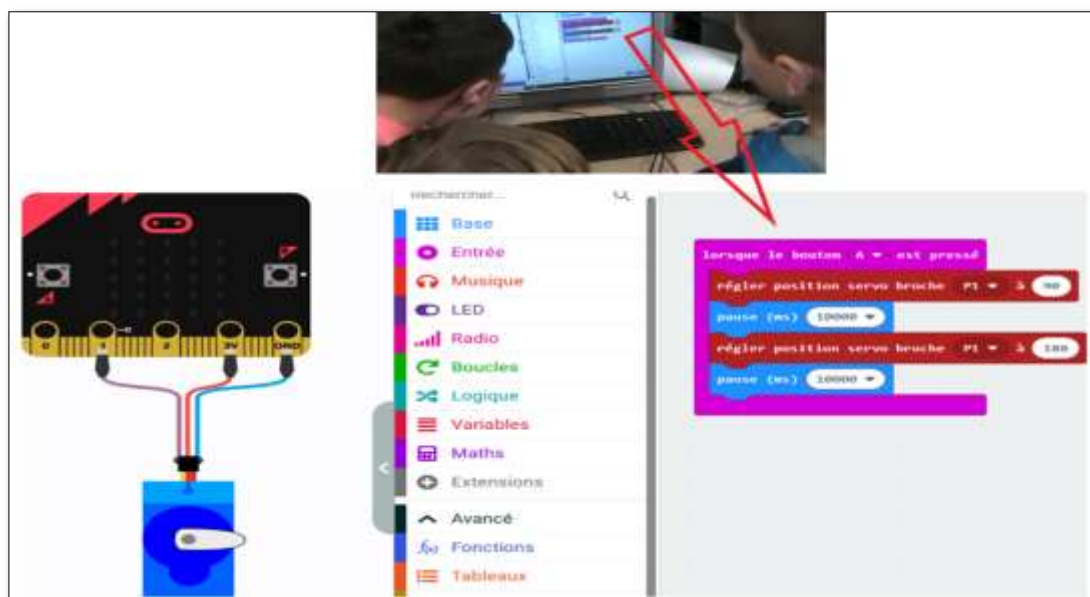


Figure 4 – Efforts des élèves pour coder le mouvement des mâchoires à l'aide de MakeCode

Nous avons pu observer que lors de tests réalisés par les élèves que l'hélice de simulateur de servomoteur pouvait effectuer un mouvement de rotation de 90 degrés dans le sens horaire. À cette étape, l'usage de la ressource a permis aux élèves d'explorer les concepts de temps (de pause) et d'angle de rotation dans un contexte de codage informatique.

Selon le journal de bord du chercheur, les élèves ont réalisé que leurs codes ne fonctionnent pas correctement. Entre autres, ils ont jugé que leur pause était trop longue. Leur enseignante qui a vu le problème a aidé les élèves en leur suggérant de diminuer la durée de la pause (à 1 seconde) pour que le servomoteur puisse mieux effectuer des mouvements aller-retour. Interrogés par le chercheur à la fin du projet, les élèves se sont rappelé de problème, sauf la durée d'une seconde (ils ont dit 1 minute, à la place de 1 seconde, ce qui plus court que 10 secondes initiales.

- E3 : Oui, on faisait sur l'ordinateur, on faisait l'affaire (le codage), on a fait comme 10 secondes, on ne voulait pas faire une longue pause. Mais on a mis comme une minute
- CH : Qui vous a aidé à corriger l'erreur ?
- E1 : C'est madame.

Également, il manquait le bloc de boucle, alors l'élève devait redémarrer le programme pour que l'hélice du simulateur de servomoteur répète le mouvement, mais les élèves n'y ont porté l'attention. Ils avaient donc besoin d'aide de l'enseignante pour comprendre qu'il fallait modifier la durée de la pause.

Étape de la mise à l'essai et validation : À cette étape, les élèves devaient intégrer le servomoteur et le *micro:bit* dans la maquette pour tester le fonctionnement de leur système informatisé de mouvement. Cependant, les élèves ont éprouvé certaines difficultés liées à la dynamisation des mâchoires de la maquette selon leur idée initiale. En effet, leur bouche était conçue de manière statique de sorte qu'on ne pouvait pas faire bouger les mâchoires. Les élèves ont ainsi trouvé un compromis prévoyant de dynamiser les dents de la maquette au lieu des mâchoires. À ce propos, l'E2 s'exprime comme suit : « On a trouvé des solutions, à la place d'ouvrir et fermer la bouche, on a fait bouger les

dents ». Les élèves ont alors coupé un morceau de carton en la forme de zigzag pour représenter les dents de la maquette (Figure 5). Les dents ont été collées sur l'hélice du servomoteur intégré dans la bouche de maquette, comme le montre la Figure 6 suivante. Les élèves ont vérifié, par la suite, le mouvement de dents, une rotation de 90 degrés (Figure 6)



Figure 5 – Mise à l'essai et création et intégration de dents



Figure 6 – Test de mouvement de dents

Étape de la présentation du projet : Les élèves ont présenté leur produit final aux pairs de la classe et à la communauté scolaire interne de l'école (Figure 7). La présentation était accompagnée d'un bref exposé oral expliquant l'animal et son mode de vie, prouvant l'intérêt et le choix portés sur l'animal, et rendant compte du déroulement de la réalisation du projet en matière d'apprentissages acquis, de méthodologie du travail, de répartition des tâches, des difficultés rencontrées et des solutions apportées.



Figure 7 – Tableau de présentation et produit final

IV. DISCUSSION ET CONCLUSION

L'article partage quelques résultats obtenus dans le cadre d'une étude doctorale portant sur le *débogage* dans un contexte de design en ingénierie. Dans le présent exemple, les élèves ont été introduits, pour la première fois, dans un *codage informatique* pour programmer une partie de leur maquette d'effectuer un mouvement de rotation à l'aide d'un servomoteur intégré contrôlé par un *micro:bit*. La complexité de la tâche réalisée par les élèves est ainsi mise en évidence dans le processus

d'instrumentation (au sens de Trouche, 2005) dans lequel l'utilisation de diverses ressources a été prise en charge par les élèves lors d'une exploration autonome (avec peu d'aide de la part de l'enseignante) (Gueudet et al., 2014). Bien à l'aise dans la construction de leur maquette initiale (en 3D), les élèves devaient alors faire face à une multitude de problèmes au niveau d'informatisation de leur système.

En cherchant à programmer leur mouvement, les élèves ont bien repéré les blocs de code nécessaires. Mais une fois rendu compte d'un fonctionnement incorrect de leur modèle (une pause trop longue), les élèves semblent avoir de la difficulté à comprendre les concepts mathématiques liés aux variables de temps et d'angle de rotation et savoir comment convertir des mesures de temps (millisecondes en secondes). Ces concepts vont cependant au-delà de ce que le programme d'études en mathématiques du N.-B. prévoit pour le niveau de la 4^e année.

De plus, vivement discutée à l'étape de conception collective de la bouche de la maquette, l'idée d'un mouvement coordonnée de deux mâchoires de tigre (vers le haut et vers le bas), bien présente dans les schémas construits par les élèves s'est heurtée à une impossibilité physique de dynamiser une forme rectangulaire trop statique de la bouche qu'ils ont conçue. Il semble ainsi avoir un décalage au niveau de coordination des actions centrées sur le codage et le transfert de code vers l'AIP (*micro:bit*) et la structure géométrique de leur modèle (la tête de leur maquette).

Étant une pratique de la PI qui a, selon Shute et al. (2017), une relation étroite avec les mathématiques, le débogage a permis aux élèves de mobiliser divers concepts mathématiques à différents stades de leur projet créatif. Les élèves ont notamment utilisé des concepts mathématiques liés à la géométrie pour résoudre le problème lié à la maquette en ajustant la forme de la bouche et de dents permettant d'assurer le mouvement désiré. Ils se sont également attardés aux problèmes liés aux paramètres de code en MakeCode, notamment, pour ajuster la durée de la pause. Bien qu'utile pour la réalisation des actions concrètes de *débogage*, ce lien entre le *débogage* et les mathématiques demeure encore peu explicité dans ce type d'activités informelles ; encore moins, on y voit un lien direct avec le curriculum. Également, on note l'importance de l'aide de l'enseignante (et d'autres personnes ressources présentes en classe), leur rôle doit également être clarifié davantage.

Ces observations, bien qu'utiles pour comprendre la genèse instrumentale de l'élève (Trouche, 2005), ne fournissent pas toutefois suffisamment de données pour comprendre comment le contexte plutôt informel d'exploration de ressources STIM interagit avec le processus de construction de savoirs mathématique au sens institutionnel (Chevallard, 1991). Il y a donc une nécessité de mener des études plus approfondies pour comprendre comment rendre plus explicite (et efficace) le rôle de mathématiques dans un projet interdisciplinaire impliquant le *codage informatique* et l'usage de AIP.

RÉFÉRENCES

- Artigue, M. (2006). The integration of symbolic calculators into secondary education: some lessons from didactical engineering. Dans G. Guin, K. Ruthven et L. Trouche (dir.), *The didactical challenge of symbolic calculators: Turning a computational device into a mathematical instrument* (p. 231-294). Springer.
- Ball, T., Chatra, A., Peli de Halleux, J., Hodges, S., Moskal, M. et Russell, J. (2019). Microsoft MakeCode: Embedded programming for education, in blocks and TypeScript. Dans *Proceedings of the ACM SIGPLAN2019, Symposium on SPLASH-E, 25 octobre 2019, Athènes, Grèce* (p. 7-12). Association for Computing Machinery. <https://doi.org/10.1145/3358711.3361630>
- Csapó, G. (2019). Placing event-action-based visual programming in the process of computer science Education. *Acta Polytechnica Hungarica*, 16(2), 35-57. http://acta.uni-obuda.hu/Csapo_89.pdf

- Chevallard, Y. (1991). *La transposition didactique. Du savoir savant au savoir enseigné* (2^e éd.). La Pensée sauvage.
- DeJarnette, N. (2012). America's children: Providing early exposure to STEM (science, technology, engineering and math) initiatives. *Education*, 133(1), 77–84.
- Deed, C., Cardellino, P., Matthews, E. et Outhall, A. (2022). A qualitative exploration of cardboard architecture in post-pandemic schools. *International Journal of Educational Research Open*, 3, article 100186. <https://doi.org/10.1016/j.ijedro.2022.100186>
- De Zoysa, S. (2024) *Microsoft global outages caused by CrowdStrike software glitch*. https://www.researchgate.net/publication/382625564_Microsoft_global_outages_caused_by_CrowdStrike_software_glitch
- Drijvers, P., Tacoma, S., Besamusca, A., Van den Heuvel, C., Doorman, M. et Boon, P. (2014). Digital technology and mid-adopting teachers' professional development: A case study. Dans A. Clark Wilson, O. Robutti et N. Sinclair (dir.), *The mathematics teacher in the digital era* (p. 189–212). Springer.
- Freiman, V. (2020). Issues of teaching in a new technology-rich environment: Investigating the Case of New Brunswick (Canada) School Makerspaces. Dans Y. Ben-David Kolikant, D. Martinovic et M. Milner-Bolotin (dir.), *STEM teachers and teaching in the digital era: Professional expectations and advancement in the 21st century schools* (p. 273-292). Springer. https://doi.org/10.1007/978-3-030-29396-3_15
- Freiman, V. et Chiasson, M. (2018). Enjeux de diversification et complexification de ressources : un cas de programmation informatique d'un robot dans une école de demain. Dans M. Abboud (dir.), *Actes du colloque EMF2018 « Mathématiques en scène, des ponts entre les disciplines », 22-26 octobre 2018, Gennevilliers, Paris* (p.712-720). IREM de Paris. https://emf.unige.ch/files/9615/7545/7017/GT6_FREIMAN-CHIASSONCA_actes_EMF2018.pdf
- Freiman, V. et Lingley, J. (2024). Micro-tensions from students' prototyping in a school makerspace: Lessons from an unfinished work. Dans K. Plakitsi et S. Barma (dir.), *Sociocultural approaches to STEM education: An ISCAR international collective issue* (p. 259-293). Springer International Publishing. https://doi.org/10.1007/978-3-031-44377-0_12
- Furlong, C. et Léger, M. T. (2022). Le *tinkering* au cœur du processus de résolution de problèmes en contexte de fabrication numérique à l'école. *Revue hybride de l'éducation*, 5(2), 127-149. <https://doi.org/10.1522/rhe.v5i2.1227>
- George, A. S. (2024). When trust fails: Examining systemic risk in the digital economy from the 2024 CrowdStrike outage. *Partners Universal Multidisciplinary Research Journal*, 1(2), 134-152. <https://doi.org/10.5281/zenodo.12828222>
- Grover, S. et Pea, R. (2018). Computational thinking: A competency whose time has come. Dans S. Sentance, E. Barendsen et C. Schulte (dir.), *Computer Science Education : Perspectives on Teaching and Learning in School* (p. 19-38). Bloomsbury Academic. <http://dx.doi.org/10.5040/9781350057142.ch-003>
- Guedet, G., Buteau, C., Mesa, V. et Misfeldt, M. (2014). Instrumental and documentational approaches: From technology use to documentation systems in university mathematics education. *Research in Mathematics Education*, 16(2), 139-155. <https://hal.science/hal-01006510v1>
- Hennessy Elliott, C., Gendreau Chakarov, A., Bush, J. B., Nixon, J. et Recker, M. (2023). Toward a debugging pedagogy: Helping students learn to get unstuck with physical computing systems. *Information and Learning Sciences*, 124(1/2), 1-24. <https://doi.org/10.1108/ILS-03-2022-0051>

- Ismail, A. A. et Hassan, R. (2019). Technical competencies in digital technology towards industrial revolution 4.0. *Journal of Technical Education and Training*, 11(3).
<https://doi.org/10.30880/jtet.2019.11.03.008>
- Jang, D. J. et Kim, S. A. (2016). Performance-oriented kinetic façade design based on interworking between physical and digital models. Dans N. Yabuki et K. Makanae (dir.), *Proceeding of the 16th international conference on computing in civil and building engineering (ICCCBE 2016)*, 6-8 juillet 2016, Osaka, Japon (p. 504-511). http://www.sec.eng.osaka-u.ac.jp/secit/icccb2016/Proceedings/Full_Papers/064-192.pdf
- Juskeviciene, A., Stupurienė, G. et Jevsikova, T. (2020). Computational thinking development through physical computing activities in STEAM education. *Computer Applications in Engineering Education*, 29(1), 175-190. <https://doi.org/10.1002/cae.22365>
- Kim, C., Yuan, J., Vasconcelos, L., Shin, M. et Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science*. <https://doi.org/10.1007/s11251-018-9453-5>
- King, D. et English, L. D. (2016). Engineering design in primary school: Applying STEM concepts to build an optical instrument. *International Journal of Science Education*, 38(18), 2762-2794.
<https://doi.org/10.1080/09500693.2016.1262567>
- Kvaššayová, N., Mansell, M., Cápay, M. et Bellayová, M. (2021). The BBC micro:bit in Slovakia. Dans *Proceedings of the international conference on Information and Digital Technologies (IDT2021)*, Zilina, Slovaquie (p. 359-365). <https://doi.org/10.1109/IDT52577.2021.9497573>
- Kilhamn, C., Bråting, K. et Rolandsson, L. (2021). Teachers' arguments for including programming in mathematics education. Dans G. A. Nortvedt, N. F. Buchholtz, J. Fauskanger, F. Hreinsdóttir, M. Häikiöniemi, B. E. Jessen, J. Kurvits, Y. Liljekvist, M. Misfeldt, M. Naalsund, H. K. Nilsen, G. Pálsdóttir, P. Portaankorva-Koivisto, J. Radišić et A. Wernberg (dir.), *Proceedings of NORMA 20, The 9th Nordic Conference on Mathematics Education « Bringing Nordic mathematics education into the future »*, Juin 2021, Oslo, Norvège (p. 169-176). http://matematikdidaktik.org/wp-content/uploads/2021/04/NORMA_20_preceedings.pdf
- Lavigne, H. J., Presser, A. L. et Rosenfeld, D. (2020). An exploratory approach for investigating the integration of computational thinking and mathematics for preschool children. *Journal of Digital Learning in Teacher Education*, 36(1), 63-77.
- LeBlanc, M., Freiman, V. et Furlong, C. (2022). From STEm to STEM: Learning from students working in school makerspaces. Dans C. Michelsen, A. Beckmann, V. Freiman, U. T. Jankvist, A. Savard (dir.), *Mathematics and its connections to the arts and sciences (MACAS). 15 years of interdisciplinary mathematics education* (p. 179-203). https://doi.org/10.1007/978-3-031-10518-0_10
- Li, C., Chan, E., Denny, P., Luxton-Reilly, A. et Tempero, E. (2019). Towards a framework for teaching debugging. Dans *ACE19 Proceedings of the twenty-first Australasian computing education conference*, 29-31 janvier 2019, Sydney, Australie (p. 79-86). <https://doi.org/10.1145/3286960.3286970>
- Pruneau, D. (2019). *La pensée design pour le développement durable. Applications de la démarche en milieux scolaire, académique et communautaire*. Livres en ligne du CRIRES.
https://lel.crires.ulaval.ca/works/guide_pedagogique.la_pensee_design_pour_le_developpement_durable_0.pdf
- Rabardel, P. (1995). *Les hommes et les technologies, approche cognitive des instruments contemporains*. Armand Colin.

- Rich, K. M., Strickland, C., Binkowski, T. A. et Franklin, D. (2019). A K–8 debugging learning trajectory derived from research literature. Dans *Proceedings of the 50th ACM technical symposium on computer science education (SIGCSE19)*, 27 février au 2 mars 2019, Minneapolis, USA (p. 745–751).
- Sadler, P. M., Coyle, H. P. et Schwartz, M. (2014). Engineering competitions in the middle school classroom: Key elements in developing effective design challenges. Dans W. C. Newstetter (dir.), *Design Education* (p. 299-327). Routledge. <https://doi.org/10.4324/9781315799308>
- Shute, V. J., Sun, C. et Asbell-Clarke, J. (2017). Demystifying computational thinking. *Educational Research Review*, 22, 142-158. <https://doi.org/10.1016/j.edurev.2017.09.003>
- Shumway, J. F., Welch, L. E., Kozlowski, J. S., Clarke Midura, J. et Lee, V. R. (2021). Kindergarten students' mathematics knowledge at work: The mathematics for programming robot toys. *Mathematical Thinking and Learning*, 25(4), 380-408. <https://doi.org/10.1080/10986065.2021.1982666>
- Trouche, L. (2005). Construction et conduite des instruments dans les apprentissages mathématiques : nécessité des orchestrations. *Recherches en Didactique des mathématiques*, 25(1), 91-138.
- Vygotsky L. S. (1978). *Mind in society: Development of higher psychological processes*. Harvard university press.