

DÉNOUER

LES

CIRCUITS

DÉNOUVER

LES

CIRCUITS

Adelina Dimitrova

Mémoire DNMADE

Innovation Sociale

Lycée de Corbusier

Illkirch-Graffenstaden

Promotion 2022-2025

SOMMAIRE

Carte Mentale
Pages 6 à 9

État de l'art - "Comment faire apprendre la robotique et/ou la programmation aux enfants débutants?"
Pages 10 à 20

ANNEXES

Études de cas

Études de Cas — Design
Pages 22 à 47

Études de Cas — Technique
Pages 49 à 71

Études de Cas — Art
Pages 73 à 91

Synthèses de lecture

Pages 92 à 105

Entretien Sociologique

Pages 106 à 134

Outil de récolte

Page 135 à 150

Bibliographie commenté

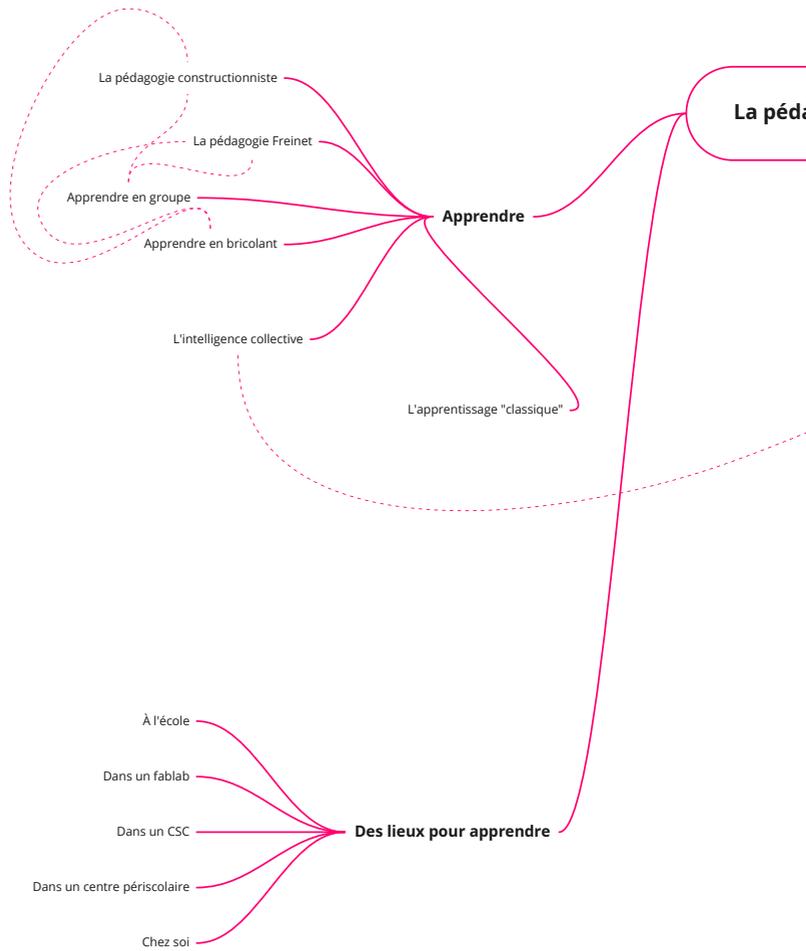
Page 151 à 155

Remerciements

Page 156 à 158

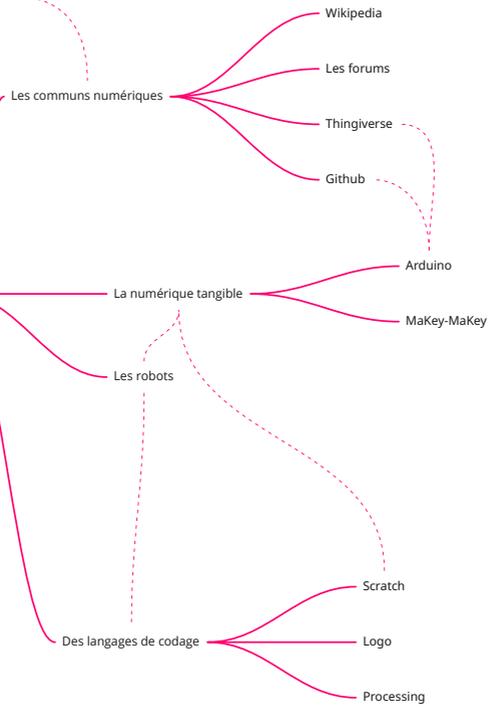
CARTE

MENTALE



Pédagogie numérique

Outils numériques



ÉTAT

DE

L'ART

“Comment faire apprendre la robotique et/ou la programmation aux enfants débutants ?”

La question du numérique en pédagogie n'a jamais été aussi pressante. Face à des outils qui ne cessent d'évoluer, les enfants peuvent se sentir de plus en plus démunis. Experts de l'objet, ils n'en perçoivent pas toujours le fonctionnement

Et si l'apprentissage du numérique permettait de considérer les objets comme des outils au service d'une autre pratique ?

Quel place à l'apprentissage de la robotique dans ces apprentissages ?

Pour affiner ces questions, nous allons premièrement nous pencher sur l'histoire de l'enseignement de l'informatique et du numérique en France à l'École depuis les années 1980, puis explorer la place du numérique dans les autres espaces éducatif, et enfin interroger la place du designer dans les outils de pédagogie numérique pour déterminer des pistes pour poursuivre cette recherche-projet.

Enseigner avec l'informatique

Le plan informatique pour tous a été initié le 25 Janvier 1985 face à l'essor de l'ordinateur domestique¹. Il avait pour but de moderniser le pays et de rendre l'informatique accessible à tous les citoyens français.

Malgré le temps qu'il a pris pour se mettre en place, il s'agissait d'une initiative ambitieuse pour son temps. Équiper chaque salle de classe avec un nanoréseau d'ordinateurs permettait de réduire les coûts financiers du plan mais aussi d'anticiper les utilisations de l'ordinateur en réseau².

Ce plan a été contesté au sein de l'Éducation Nationale, les enseignants l'accusant d'avoir brutalisé le système éducatif français³. On reprochait aussi au *Plan informatique pour tous* de ne pas être à destination des citoyens, mais plutôt d'être mis en place pour "relancer une industrie électronique française à travers la société Thomson"⁴.

Pour certains psycho-pédagogues, faire programmer des enfants en BASIC, "c'était massacrer des jeunes esprits"⁵. En plus de cela, si les classes étaient équipées en matériel informatique, les enseignants manquaient eux de formation, ce qui ne leur permettait pas de mettre en pratique l'enseignement de l'informatique de manière adaptée.⁶

Par la suite, l'ordinateur va malgré tout se banaliser dans l'enseignement traditionnel occidental jusqu'à aujourd'hui où les enjeux numériques sont plus que prégnants.

1 Jacques BAUDÉ, « Le plan « Informatique pour tous » », 1024 - Bulletin de la Société Informatique de France, no 5, mars 2015, p. pages 95-108, <https://doi.org/10.48556/SIF.1024.5.95>.

2 Ibid.

3 Ibid., p. 103-104.

4 Ibid., p. 104.

5 Ibid.

6 Ibid.

L'informatique dans l'enseignement aujourd'hui

Encore aujourd'hui, dans l'enseignement traditionnel les ordinateurs sont utilisés comme des gadgets à faire apprendre de vieux concepts abstraits comme calculer des racines carrées. On ne s'en sert pas pour leur capacité à se métamorphoser en plusieurs outils, mais plutôt comme un outil pour faire ses devoirs académiques plus "efficacement", ou comme un tuteur, qui montre les consignes à suivre et qui évalue l'élève.¹

Loin de bousculer les pratiques d'enseignement, l'informatique sert aujourd'hui à donner l'illusion d'une pratique innovante, ainsi le manuel devient un manuel numérique, l'exposé devient une présentation avec support Power Point, la dissection de grenouille devient une application interactive. Mais le cadre et les méthodes d'enseignements eux restent inchangés.

L'ensemble reste structuré, presque rigide, s'articulant entre séquences et chapitres dictés par le manuel. En dehors de la cour de récré, à l'École, on ne s'amuse pas. On est sérieux et studieux dès lors qu'on rentre en classe².

Cette situation est aussi dû à une vision française de l'enseignement.

"Si un professeur ne fait pas de cours magistraux qui durent 45 minutes, est-ce qu'il enseigne vraiment?"³

Cette question est souvent posée par des personnes extérieures à l'enseignement car elles estiment qu'un enseignant est un sachant qui transmet ses connaissances à des apprenants silencieux et réceptifs.

Il existe pourtant d'autres mouvements éducatifs qui envisagent l'apprentissage autrement, et qui incluent les outils informatiques dans leurs pratiques.

On retrouve ainsi la pédagogie Freinet, où le professeur occupe un rôle différent. En éducation nouvelle, l'élève devient acteur de son apprentissage et est encouragé à chercher, créer et apprendre par lui-même.⁴

L'enseignant devient alors un guide, un documentaliste, ou un chef d'atelier pour les élèves, afin de les amener à apprendre en collaborant tous ensemble.

1 Voir synthèse de lecture - *Invent to Learn* dans les annexes

2 *Ibid.*

3 *Ibid.*

4 Manutan COLLECTIVITÉS, « Comprendre la pédagogie Freinet en 10 points clés », dans *Classe de Demain - Manutan Collectivités*, <https://www.classe-de-demain.fr/accueil/education/comprendre-la-pedagogie-freinet-en-10-points-cles> (Page consultée le 11 mars 2025).

La place du numérique dans les espaces éducatifs

Seymour Papert, mathématicien, informaticien, et éducateur au *Massachusetts Institute of Technology (MIT)* a fondé le mouvement *constructionnisme* en s'inspirant du travail effectué avec *Jean Piaget* sur le *constructivisme*.

Le *constructionnisme* est un mouvement visant à faciliter l'apprentissage de l'informatique.⁵ Il encourage l'enfant à chercher par lui-même et à expérimenter librement l'utilisation de l'ordinateur. Il incite également les enfants à travailler en groupe.

On peut considérer le constructionnisme comme un des prolongements de la *pédagogie Freinet*, en effet ces deux mouvements placent l'élève au centre des apprentissages et mettent en avant la collaboration entre enfants.

Dans la théorie du *constructionnisme*, on pense plutôt projet que leçon et contrôle.⁶ Les projets permettent aux groupes d'enfants d'expérimenter des principes numériques, d'explorer les mécanismes des outils, et de documenter leurs propres recherches.

À l'inverse d'un morcellement des enseignements, le constructionnisme va ainsi proposer des projets sollicitant différentes compétences et champs d'application. Ainsi, plutôt que d'apprendre à utiliser Scratch en cours de technologie, puis l'heure d'après de découvrir le principe de germination des plantes en Sciences et Vie de la Terre (SVT), et enfin de dessiner des natures mortes en arts plastiques, on va plutôt encourager l'élève à prendre en main Scratch, puis à réaliser un jeu vidéo permettant d'expliquer comment pousse une plante en utilisant ses propre dessins.

Le numérique devient alors un outil au service du projet.

Il permet d'accompagner les enfants dans leurs réalisations et d'ouvrir de nouvelles possibilités.

Malheureusement, il est compliqué d'appliquer les théories constructionnistes dans l'École française contemporaine et notamment dans les collèges.

En effet, le projet doit permettre à l'élève d'aller dans différentes directions, d'explorer de nombreuses pistes, et de se tromper, alors même que le système encourage l'enseignant à évaluer des compétences et des savoirs standardisés.

⁵ « Constructionism (learning theory) », dans Wikipedia, [s.l.], 2024, [https://en.wikipedia.org/w/index.php?title=Constructionism_\(learning_theory\)&oldid=1203563760](https://en.wikipedia.org/w/index.php?title=Constructionism_(learning_theory)&oldid=1203563760) (Page consultée le 11 mars 2025).

⁶ Voir synthèse de lecture - Invent to learn dans les annexes.

Mais l'École n'est pas la seule sphère éducative que rencontre l'enfant.

Il existe aussi des "tiers lieux" en dehors de la famille et de l'École qui peuvent jouer ce rôle.

Les FabLabs ou certains accueils périscolaires invitent ainsi les enfants à pratiquer la programmation et la robotique. On y retrouve différents milieux qui s'y rencontrent, bidouillent, bricolent et fabriquent ensemble des objets numériques à leur rythme sans la pression de l'évaluation.

En 2005, au moment de sa création, *Scratch* va d'abord être testé dans un centre périscolaire informatique nommé *The Clubhouse*. *The Clubhouse* est situé dans un quartier populaire de Los Angeles et permet aux jeunes issus de milieux défavorisés d'accéder librement à des ordinateurs et différents logiciels.¹

Lors de ce test, les chercheurs du *MIT Média Lab* laissent les enfants libres de choisir ce qu'ils veulent faire avec le logiciel. Les animateurs eux, sont encouragés à collaborer et à aider les jeunes pour accomplir leurs projets.

Dans ce cadre extérieur aux normes de l'École, les jeunes proposeront des projets variés : images animées, jeux vidéos, clip vidéo...² Lors de cette expérimentation *Scratch*, les jeunes disent percevoir le logiciel comme un cahier d'esquisse, où tout est possible.

Ils n'associent pas *Scratch* avec de l'informatique, mais plutôt avec des arts visuels ou littéraires. Et pourtant lors de la réalisation de leurs projets, ils mettent en œuvre les notions fondamentales de la programmation comme les boucles et les variables par exemple.³

Mais si les volontés des chercheurs étaient celles-ci au départ, les usages et le regard porté sur Scratch ont aujourd'hui changés. En effet, l'utilisation du logiciel en cours de technologie au collège souvent pour pratiquer l'arithmétique a tendance à ternir l'image du logiciel et à le résumer à ce seul usage.

1 Voir synthèse de lecture - *Programming by Choice* dans les annexes

2 *Ibid.*

3 *Ibid.*

Yann Misse, fabmanager et facilitateur-animateur à *la CabAnne des créateurs* à *Schiltigheim* évoque un manque de moyens (financier, humain et temporel) des enseignants pour mettre en place des activités individualisées.⁴

Il envisage *les Fablabs* comme des espaces permettant à tous les enfants de découvrir et d'alimenter leur propre curiosité. Ces espaces permettent de s'adapter aux particularités de chacun, *Yann Misse* mentionne l'intelligence personnelle de l'individu. La pluralité des activités proposées permet de répondre aux besoins de chaque enfant. Par exemple, certains enfants ont une intelligence corporelle qui va les amener à s'intéresser aux activités sportives, d'autres auront une intelligence manuelle qui les amènera à pratiquer des activités techniques.⁵

Les activités numériques peuvent, elles, investir tous les champs, et ne pas se destiner uniquement (comme on pourrait l'imaginer) à des enfants ayant une intelligence logico-mathématique.⁶ *La CabAnne des Créateurs* propose par exemple des activités plastiques utilisant la robotique. L'activité de découverte du *Ozobot* consiste ainsi à réaliser un dessin qui va guider un robot équipé d'un système de capteurs. Ces animations avec les *Ozobots* permettent de susciter la curiosité chez les enfants et de les amener vers l'informatique pas à pas.⁷

Les fablabs jouent un rôle pivot dans l'éducation numérique. Ils comblent les lacunes de l'École qui ne s'adapte pas aux différents profils d'élèves, en proposant une approche plus active et plus ludique des outils numériques.

4 Voir *Entretien sociologique - Yann Misse dans les annexes*

5 *Ibid.*

6 *Ibid.*

7 *Ibid.*

Comment le design peut-il faciliter les apprentissages et réduire les inégalités face au numérique ?

En design, l'enseignement du numérique et de la robotique vont être essentiellement développés dans le champ de l'industrie. C'est ainsi les ingénieurs, les design industriels et les investisseurs qui vont s'en emparer.

On peut observer deux tendances dans ce développement.

Tout d'abord, une tendance qu'on pourrait qualifier de "gadget", c'est-à-dire des professionnels qui vont concevoir des outils numériques comme une réponse miraculeuse aux problématiques pédagogiques (inclusion, accès à l'éducation, mixité...).

Ces recettes magiques sont souvent des échecs, comme le montre l'initiative *One Laptop per Child* (Un Ordinateur Portable par Enfant) lancée en 2005, par *Nicholas Negroponte* qui dit s'être inspirée de la pensée constructionniste.¹

Dans ce projet, l'idée est de créer un ordinateur portable pour enfant n'excédant pas 100 dollars. Ces ordinateurs seraient ainsi distribués dans les pays en voie de développement pour que les enfants puissent apprendre l'informatique.

Cette initiative, très alléchante pour les investisseurs occidentaux qui faisaient ainsi "une bonne action" fût très critiquée par les diplomates issus des pays concernés. En effet, au-delà de l'objet un certain nombre de problématiques se posent : mettre en place des infrastructures pour alimenter les ordinateurs, en assurer la maintenance, former les enfants à l'utilisation de l'outil...²

Cette approche jugée déconnectée des enjeux témoigne d'une méconnaissance, et d'une non prise en compte des contextes culturelles et géographiques des pays visés par le projet.³ Dans *One Laptop per Child*, on retrouve les mêmes problématiques que celles générées par le plan "*Informatique pour tous*" : un commanditaire imagine que la démocratisation de l'outil numérique suffit à régler la question de l'accès.

Mais il ne suffit pas de donner un ordinateur aux enfants pour qu'ils sachent s'en servir et qu'ils puissent l'utiliser pour d'autres usages... La prise en compte des particularités des personnes et du contexte, ainsi que leur accompagnement dans la démarche semble essentielle pour arriver à ces objectifs.

¹ « *One Laptop per Child* », dans *Wikipedia*, [s.l.], 2025, https://en.wikipedia.org/w/index.php?title=One_Laptop_per_Child&oldid=1280173160#Teacher_training_and_ongoing_support (Page consultée le 20 mars 2025).

² *Why Western Designs Fail in Developing Countries*, 2024, 27:36, <https://www.youtube.com/watch?v=CGRtyxEpoGg> (Page consultée le 19 mars 2025).

³ « *One Laptop per Child* », *loc. cit.*

Mais on peut observer une autre tendance dans la conception d'outils pour l'apprentissage du numérique. Une approche matérielle qui consiste à réaliser des artefacts manipulables pour mieux saisir des concepts abstraits.

C'est le cas de *LEGO Mindstorms* par exemple conçu par le *MIT Media Lab* en collaboration avec la compagnie *LEGO*. Ce kit permet de créer des robots programmables de manière ludique avec un objet familier (les pièces *LEGO*), tout en initiant les enfants aux principes de la robotique. On retrouve ce type de démarche dans d'autres projets comme la carte *MaKey-MaKey* et les objets *Cubetto* qui utilisent le tangible pour mieux enseigner la programmation et la robotique.⁴

En plus, de cette matérialité on constate que ces outils sont la plupart du temps accompagnés d'une plateforme qui fédère une communauté autour de l'objet. Ces communautés indépendantes permettent aux usagers de partager des ressources et des connaissances. Ainsi il n'est pas seul face à l'outil, il a toujours à sa disposition des ressources qui lui permette de s'approprier le dispositif. La communauté Arduino par exemple qui permet d'accompagner la carte du même nom. Celle-ci est open source ce qui lui a permis de se développer et de favoriser des usages propre à chaque utilisateur.

Un autre type de projet est de matérialiser la communauté elle-même.

Le magazine Make: par exemple est un outil de diffusion de projets numériques qui valorise le mouvement Maker, et permet à ses lecteurs d'agir et de réaliser des projets là où ils sont.

Pour construire un projet, il est donc nécessaire de s'intéresser aux usagers qui vont interagir avec.

Aussi, lors de l'atelier outillé mené à *la CabAnne des Créateurs à Schiltigheim* avec un groupe d'enfants, nous avons interrogé la perception des robots chez les enfants, pour identifier des manières de les initier à la robotique.

Lors de l'atelier, nous avons constaté que les enfants avaient tendance à concevoir des robots anthropomorphiques avec comme rôle d'aider ou de remplacer les humains dans des tâches demandant de la précision et de la force, comme le jardinage ou la construction. Les adultes accompagnants avaient eux tendance à proposer des usages plus variés pour leurs robots. Par exemple, un des adultes a créé un robot thérapeutique pour guider les aveugles dans leur vie quotidienne. L'atelier outillé a permis de pointer une perception assez restreinte des fonctions des robots chez les enfants. Cependant, il ne s'agissait pas là d'un robot-outils qui pourrait les aider à concevoir ou à imaginer d'autres projets.⁵

⁴ Voir *Études de cas design dans les annexes*.

⁵ Voir *Atelier Outillé dans les annexes*.

Aussi pour la suite de la recherche, il pourrait être nécessaire de viser à changer la perception qu'ont les enfants des robots, en partant du postulat que ce changement leur permettrait d'entretenir un autre lien avec l'apprentissage du numérique, un lien les rendant acteurs de cet apprentissage.

La problématique serait alors...

Comment le design social peut permettre aux enfants d'agir leur apprentissage du numérique en mobilisant des compétences de robotique ?

Nous pourrions alors imaginer des outils pour mettre en œuvre des ateliers robotique.

Les enfants s'inscrivent alors à la CabAnne des Créateurs et seraient amenés à manipuler différents types de robots. Le numérique serait alors l'objet de l'activité, et le designer réfléchirait à la manière d'outiller ses ateliers, et observerait l'évolution des participants sur la question du numérique.

Ou alors, nous pourrions au contraire interroger la place des robots dans d'autres types d'activités (jeux sportifs, activités plastiques, activités d'expression...).

Créer des outils pour pouvoir jouer au robot lors d'une activité sportive, ou produire des réalisations plastiques à l'aide de robots.

Cette piste permettrait à la fois de travailler sur la place de la robotique et du numérique dans d'autres champs, et de déconstruire la vision qu'ont les enfants de ce type d'activités.

ÉTUDES

DE

CAS

DESIGN



Arduino

Interaction Design Institute Ivrea
2005

Arduino est une plateforme open-source de logiciel et de matériel informatique créée en 2005 par l'*Interaction Design Institute d'Ivrée en Italie*.¹

Cette plateforme destinée au prototypage de projets est ouverte à tous (avec ou non une connaissance de l'informatique).

La plateforme comporte des cartes microcontrôleurs licenciées sous un modèle copyleft qui permet de créer ses propres clones compatibles avec les autres modules et logiciels de programmation *Arduino*, qui eux sont aussi ouverts.²

L'IDE³ *Arduino* permet de programmer la carte avec la langue *Wiring*, qui est inspirée de *Processing* et qui se base sur le *C++*.⁴

L'*Arduino* comporte aussi une très grande communauté de passionnés qui partagent leurs projets et leur expertise sur différents sujets, que ce soit l'électronique, la robotique, la programmation, etc.⁵

1 « *Arduino* », dans *Wikipedia*, [s.l.], 2024, <https://en.wikipedia.org/w/index.php?title=Arduino&oldid=1262713605> (Page consultée le 16 décembre 2024).

2 *Ibid.*

3 *IDE* = Un environnement de développement intégré, qui aide à créer des programmes pour l'*Arduino* et à les importer sur la carte.

4 « *Arduino* », *loc. cit.*

5 « *Arduino Projects* », dans *Instructables*, <https://127.0.0.1:8001/circuits/arduino/projects/> (Page consultée le 18 mars 2025).



Cette plateforme est très intéressante pour la robotique, car elle ouvre à tous la possibilité de créer son propre robot, grâce à une carte universelle et open-source qui peut contrôler presque tous les capteurs et se programmer de différentes manières.

Sa grande communauté permet d'apprendre avec les autres, de créer un lien social autour d'un même sujet, et une intelligence collective, qui n'est pas restreinte à un seul public de sachants



Processing

MIT Media Lab
2001

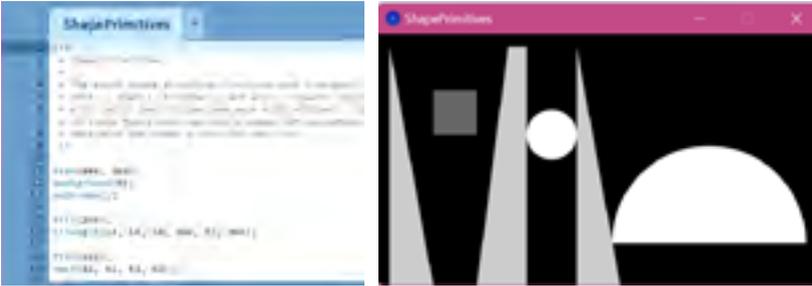
Processing est un environnement de développement libre (sous licence GNU GPL), créé par Benjamin Fry et Casey Reas, deux artistes américains.

Processing est le prolongement «multimédia» de *Design by numbers*, l'environnement de programmation graphique et d'apprentissage du code créatif développé par leur enseignant *John Maeda* au *MIT Media Lab*.¹

Processing est tout particulièrement adapté à la création plastique et graphique interactive et au graphisme de données, mais sa simplicité d'utilisation en fait aussi un bon support pour l'apprentissage de la logique de la programmation pour les non-programmeurs.²

¹ « Processing », dans Wikipédia, [s.l.], 2024, <https://fr.wikipedia.org/w/index.php?title=Processing&oldid=218959104> (Page consultée le 17 décembre 2024).

² *Ibid.*



Un programme en *Processing* s'appelle un "*sketch*", qui se traduit par esquisse en français. Cette nomenclature permet d'approcher le codage d'une différente manière, qui permet d'utiliser le code plus librement pour créer des expérimentations graphiques, tel comme une esquisse.³

L'idée du nom de *Processing* vient de la volonté de se concentrer sur le processus de créer une esquisse, et pas de sa finalité, ainsi que d'indiquer l'état actif des programmes qui sont toujours en processus de tourner.⁴

³ Processing FOUNDATION, « A Modern Prometheus », dans Processing Foundation, 8 juin 2018, <https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85> (Page consultée le 16 décembre 2024).

⁴ Ibid.



Makey-Makey

Jay Silver + Eric Rosenbaum

2010

Makey-makey est un kit d'invention et de prototypage électronique conçu pour relier des objets de la vie quotidienne à des touches de clavier.¹ Il a été créé par Jay Silver et Eric Rosenbaum, deux étudiants du *MIT Media Lab* qui expérimentaient des manières d'utiliser les objets du quotidiens pour interagir avec un ordinateur.²

Tous deux affirment qu'ils souhaitent encourager les gens à voir le monde autour d'eux comme un gigantesque jeu de construction³.

Cette carte électronique est souvent utilisé en conjonction avec Scratch pour contrôler des jeux vidéos de manière peu conventionnel : que ce soit avec des bananes, des cannettes, de la peinture conductive, ou tout autre matériau qui conduit de l'électricité⁴.

Makey-Makey encourage ses utilisateurs à partager leurs créations en ligne et à discuter ensemble des différents projets qu'ils pourraient créer avec le kit d'invention.⁵

1 « Makey Makey », dans Wikipedia, [s.l.], 2024, https://en.wikipedia.org/wiki/Makey_Makey (Page consultée le 5 mars 2025).

2 « MIT students' invention turns bananas into keyboard », BBC News (1 juin 2012), <https://www.bbc.com/news/technology-18303012> (Page consultée le 5 mars 2025).

3 Ibid.

4 « Makey Makey », loc. cit.

5 « Teachers Projects », dans Instructables, <https://127.0.0.1:8001/teachers/makeymakey/projects/> (Page consultée le 5 mars 2025).



La simplicité de *Makey-Makey* ouvre les portes de l'expérimentation électronique à tous. Sa simplicité d'usage permet à chacun de pouvoir imaginer des interfaces accessibles, notamment les personnes ayant des troubles de la motricité.⁶

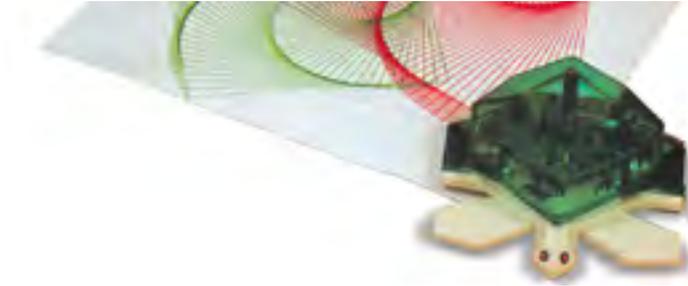
Le fait que la carte peut être utilisée directement sans avoir recours à des explications trop longues facilite cet accès.⁷ Et permet même à de jeunes enfants de fabriquer des circuits sans l'aide d'un adulte, ou d'imaginer des manettes alternatives pour un jeu vidéo.⁸

En terme d'apprentissage, *Makey-Makey* permet d'expérimenter le fonctionnement d'un circuit électrique, et d'envisager différemment les interfaces avec lesquelles nous interagissons habituellement.

6 « MIT students' invention turns bananas into keyboard », *loc. cit.*

7 *Ibid.*

8 *Ibid.*



La tortue Logo

Seymour Papert
1969

La tortue *Logo* est un robot éducatif destiné à l'apprentissage du langage de programmation *Logo*.¹

Elle a été créée en 1969 par *Seymour Papert, Wallace Feurzeig, et Cynthia Solomon* afin de rendre la programmation visuelle et interactive. Le robot-tortue obéit aux programmes de *Logo*, un langage de programmation créé dans le but d'apprendre le codage aux enfants.

Le langage *Logo* se base sur *Lisp*², un ancien langage de programmation toujours aussi utilisé maintenant.

Pour créer *Logo*, les chercheurs *Logo* se sont basés sur la théorie d'apprentissage de Jean Piaget³, où l'apprentissage encouragé par l'exploration et la participation à des projets menés en classe, au lieu de suivre des cours classiques.

1 CYBERNET1, « 1969 - The Logo Turtle - Seymour Papert et al (Sth African/American) », dans *cyberneticzoo.com*, 10 janvier 2010, <http://cyberneticzoo.com/cyberneticanimals/1969-the-logo-turtle-seymour-papert-marvin-minsky-et-al-american/> (Page consultée le 19 mai 2024).

2 « Logo (langage) », dans Wikipédia, [s.l.], 2024, [https://fr.wikipedia.org/w/index.php?title=Logo_\(langage\)&oldid=221518753](https://fr.wikipedia.org/w/index.php?title=Logo_(langage)&oldid=221518753) (Page consultée le 6 janvier 2025).

3 *Ibid.*



La tortue *Logo* et le langage *Logo* ont permis à la création et le développement de différents outils et langages par la suite.

Par exemple, le langage *Scratch* a été inspiré par *Logo* et garde l'idée d'un langage simple, facile à prendre en main tout en étant puissant quand on le maîtrise.⁴

La tortue *Logo* a permis aux enfants de pouvoir interagir avec un objet physique lors de leurs séances de programmation et de mieux s'y projeter quand ils programment.

⁴ *Ibid.*



Scratch

Mitchel Resnick + MIT Media Lab
2007

Scratch est un langage de programmation graphique à but éducatif, manipulable et exécutable par le logiciel du même nom.

Ainsi, *Scratch* est à la fois un environnement de développement, un moteur d'exécution du langage Scratch et un site web.¹ Il a été créé en 2007 par *Mitchel Resnick* avec l'équipe de *Lifelong Kindergarten* à *MIT Media Lab* avec une volonté de prioriser l'apprentissage créatif chez les enfants et de rendre la programmation accessible et ludique².

Scratch permet de nombreuses applications notamment la création d'image animée et de jeux vidéo,³ ainsi que la découverte de la robotique lorsqu'on y ajoute des plug-ins.

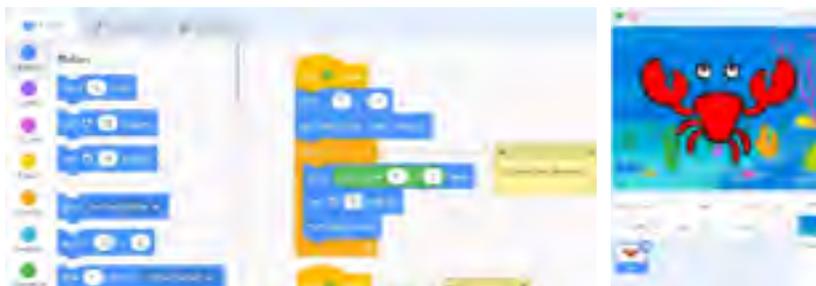
Depuis sa mise en ligne pour le public en 2007⁴, *Scratch* est devenu un incontournable de l'apprentissage de l'informatique. Répandu dans de nombreux milieux éducatifs, on le retrouve aujourd'hui aussi bien à l'École (l'enseignement de *Scratch* est aujourd'hui dans les programmes scolaires du Collège), que dans les accueils périscolaires, les CSC (Centres Sociaux Culturels) ou encore les fablabs.

1 « *Scratch (langage)* », dans Wikipédia, [s.l.], 2025, [https://fr.wikipedia.org/w/index.php?title=Scratch_\(langage\)&oldid=223919564](https://fr.wikipedia.org/w/index.php?title=Scratch_(langage)&oldid=223919564) (Page consultée le 18 mars 2025).

2 « *Our Story* », dans Scratch Foundation, <https://www.scratchfoundation.org/our-story> (Page consultée le 18 mars 2025).

3 John H. MALONEY et al., « *Programming by choice: urban youth learning programming with scratch* », *SIGCSE Bull.*, vol. 40, no 1, 12 mars 2008, p. 367-371, <https://doi.org/10.1145/1352322.1352260>.

4 « *Scratch (programming language)* », dans Wikipedia, [s.l.], 2025, [https://en.wikipedia.org/w/index.php?title=Scratch_\(programming_language\)&oldid=1279452108](https://en.wikipedia.org/w/index.php?title=Scratch_(programming_language)&oldid=1279452108) (Page consultée le 18 mars 2025).



En effet, la simplicité d'utilisation de *Scratch* lui permet d'être accessible dès 8 ans, et les nombreuses possibilités qu'il offre, permettent d'accompagner l'enfant jusqu'à ses 16 ans,⁵ en introduisant des concepts de plus en plus difficiles, comme les variables et les boucles.

De plus, *Scratch* dispose de beaucoup de documentation et de ressources pédagogiques, sa grande communauté de programmeurs permet d'accéder à de nombreux programmes déjà codés, et d'être aidé par d'autres pour mieux comprendre la plateforme.⁶

Tous ces paramètres facilitent sa prise en main et son enseignement quel que soit le niveau scolaire.⁷

Cette communauté *Scratch* maintient un lien entre ses usagers, et leur permet d'apprendre les uns des autres. Cette ubiquité du logiciel a certainement permis une démocratisation du code. Ainsi, tout le monde, quel que soit son milieu, son âge, son lieu d'habitation peut aujourd'hui (à partir du moment où il a un ordinateur) fabriquer un programme informatique.

5 *Scratch - About*, <https://scratch.mit.edu/> (Page consultée le 18 mars 2025).

6 *Scratch - Explore*, <https://scratch.mit.edu/> (Page consultée le 18 mars 2025).

7 *Scratch - Educators*, <https://scratch.mit.edu/> (Page consultée le 18 mars 2025).



Poppy Project

Matthieu Lapeyre
2012

Le *Poppy Project* est un projet robotique open-source créé en 2012 par *Matthieu Lapeyre*. Il a développé ce projet quand il était élève du laboratoire *Flowers à l'INRIA Bordeaux Sud-Ouest* sous la direction de *Pierres-Yves Oudeyer*, chercheur en robotique et intelligence artificielle.¹ Le projet a été conçu comme une plateforme open-source low-cost pour expérimenter le rapport entre humain et machine à travers la morphologie du robot.² Il est très personnalisable grâce à l'impression 3D et il donne aussi la possibilité d'expérimenter avec d'autres matériaux que le plastique dur.³ Il n'est pas coûteux à fabriquer ni à modifier, et comme il est sur une licence open-source, on peut rajouter des nouvelles fonctionnalités comme on veut.⁴

Poppy peut se décliner en plusieurs versions, les plus connues étant *Poppy Humanoid*, *Poppy Torso* *Poppy Ergo Jr*.

Poppy Humanoid, comme le nom indique, est la version humanoïde pendant que *Poppy Torso* est juste une version sans les jambes, tandis que *Poppy Ergo Jr* est plus en forme de lampe, avec une possibilité d'avoir une pince pour manipuler des objets.⁵

1 *Poppy Project - À propos*, <https://www.poppy-project.org/fr/about/> (Page consultée le 6 janvier 2025).

2 *Matthieu LAPEYRE, Poppy: open-source, 3D printed and fully-modular robotic platform for science, art and education, phd thesis, [s.l.]*, Université de Bordeaux, 2015, <https://theses.hal.science/tel-01104641> (Page consultée le 6 janvier 2025).

3 *Ibid.*

4 *Ibid.*

5 *Poppy Project - plateforme open source pour la robotique*, <https://www.poppy-project.org/fr/> (Page consultée le 6 janvier 2025).



Il a aussi une librairie *Python* appelé *Pypot*,⁶ qui permet de le programmer librement de bas-niveau⁷, mais on peut aussi choisir d'autres langages, comme *Scratch* afin de faciliter plus la prise en main du robot pour les débutants.⁸

Poppy a une grande volonté de se dissocier des plateformes traditionnelles robotiques d'études comme le *Nao d'Aldebaran*, qui lui est beaucoup trop coûteux et trop fermé à la modification, avec des règles rigides à sa utilisation et une morphologie qu'on ne peut pas modifier ni expérimenter avec.⁹

Il peut être un peu plus difficile à utiliser mais offre une très grande flexibilité avec sa modularité et avec sa plateforme open-source. Il existe aussi une très grande communauté d'utilisateurs *Poppy* qui permet d'avoir une intelligence collective très grande. Il est intéressant pour utiliser avec un jeune publique novice

⁶ Matthieu LAPEYRE, *Poppy*, op. cit.

⁷ Bas-niveau signifie qu'on peut le programmer jusqu'aux composants électroniques individuels.

⁸ *Programmation avec Scratch · Documentation of the Poppy Platform*, [s.d.], <https://docs.poppy-project.org/fr/programming/scratch> (Page consultée le 6 janvier 2025).

⁹ Matthieu LAPEYRE, *Poppy*, op. cit.



Bittle

Petoi
2020

Bittle est un chien robot open-source créé par la société Petoi en se basant sur **l'Arduino**. **Bittle** a été conçu en 2020 avec une volonté de créer un kit robotique accessible à tous.¹

Il est très simple à construire avec des pièces plastiques qui s'assemblent comme un puzzle 3D et ensuite il est rapide à calibrer avec l'application mobile.² Le chien robot est programmable avec plusieurs langages, allant d'un langage similaire à **Scratch**³ pour les débutants jusqu'à **Python** pour des utilisateurs plus avancés⁴.

Il est aussi très mobile et durable malgré sa petite taille⁵, qui le rend particulièrement pertinent à utiliser avec un jeune public tels que des enfants.

Son utilisation avec le langage basé sur **Scratch** le rend aussi très accessible aux débutants⁶ et sa ressemblance avec un chien rend sa forme familière et permet aux débutants de mieux se projeter.

On peut aussi facilement lui rajouter d'autres fonctionnalités, comme une caméra pour voir son environnement et interagir avec, qui pourrait être intéressant pour certains projets.⁷

1 « Realistic Open Source Programmable Bittle Robot Dog », dans Petoi, <https://www.petoi.com/pages/bittle-open-source-bionic-robot-dog> (Page consultée le 6 janvier 2025).

2 « We test out Bittle, a pet robot dog that will teach you how to code - review », dans BBC Science Focus Magazine, <https://www.sciencefocus.com/reviews/bittle-robot-dog-review> (Page consultée le 6 janvier 2025).

3 « Petoi Coding Blocks Released for Block-based Scratch Programming Suppo », dans Petoi, 17 avril 2023, <https://www.petoi.com/blogs/blog/petoi-coding-blocks-released-for-block-based-programming-support> (Page consultée le 6 janvier 2025).

4 Python API | Petoi Doc Center, 15 novembre 2024, <https://docs.petoi.com/apis/python-api> (Page consultée le 6 janvier 2025).

5 « We test out Bittle, a pet robot dog that will teach you how to code - review », loc. cit.

6 « Petoi Coding Blocks Released for Block-based Scratch Programming Suppo », loc. cit.

7 « Realistic Open Source Programmable Bittle Robot Dog », loc. cit.



Il repose aussi sur la plateforme *OpenCat*, qui le permet d'être facilement programmable sous une licence open-source et qui lui permet aussi d'avoir une grande communauté qui forme une intelligence collective autour du robot, et qui partagent tous leurs projets et leur programmes pour *Bittle*.⁸

Pour initier ou sensibiliser un jeune public à la robotique, *Bittle* permet un apprentissage par la pratique. Au lieu de simplement lire ou suivre des théories, *Bittle* permet de programmer et de voir immédiatement les résultats en action.

Par exemple, on peut programmer des mouvements simples, puis les tester en observant le robot exécuter les commandes. Lorsque les usagers modifient le code de *Bittle*, ils obtiennent un retour immédiat en observant ses réactions.

Cela aide à comprendre comment la programmation affecte directement le comportement du robot, renforçant ainsi la compréhension des concepts de base. *Bittle* peut, par exemple, être utilisé par un designer social, dans des ateliers ou des événements pour sensibiliser les communautés à des sujets comme la robotique, l'intelligence artificielle et l'impact social de ces technologies.

Cela peut favoriser une meilleure compréhension des avancées technologiques et de leurs implications sociales, notamment dans les quartiers moins favorisés ou les populations moins exposées à ces domaines.

8 *Ibid.*



Cubetto

Primotoys
2020

Cubetto est un set de jeu éducatif fourni avec un robot en bois, un tapis de jeu, une carte de programmation et des jetons d'instruction colorés.¹ Il permet aux enfants à partir de 3 ans d'apprendre les bases de programmation sans écran.²

Les jetons de couleur correspondent à des ordres formels : tourner, avancer, exécuter une séquence prédéfinie.³ **Cubetto** invite à raisonner de manière rigoureuse, notamment grâce à son quatrième et dernier type de jeton, le bleu, qui donne un ordre formel de suivre la ligne de bas, qui se retrouve sur la carte de programmation.⁴ La ligne en bas permet de complexifier les commandes, et de programmer une séquence d'ordres qui peut être répétée. Maîtriser celle-ci est fondamental pour pouvoir traverser entièrement les tapis de jeu en réalisant des trajets longs ou complexes, notamment en répétant certaines séquences.⁵ Le **Cubetto** ainsi que sa carte de programmation et ses jetons sont fabriqués à partir de bois robuste, qui permet au jouet de "collecter de l'histoire", à travers des signes d'usage.⁶ Son design a pour intention d'être interculturel et neutre en genre, afin d'encourager les filles et les garçons âgés de 3 à 5 ans à apprendre les principes de la programmation.⁷ **Cubetto** a été inspiré des méthodes de Maria Montessori et du langage de programmation **LOGO**, créé par **Seymour Papert**.⁸

1 « *Cubetto is a wooden robot that teaches children how to code* », dans *Dezeen*, 9 mars 2016, <https://www.dezeen.com/2016/03/09/cubetto-primotoys-preschool-children-coding-programming-wooden-robot/> (Page consultée le 26 février 2025).

2 « *Cubetto: A toy robot teaching kids code & computer programming* », dans *Primo Toys*, <https://www.primotoys.com/> (Page consultée le 26 février 2025).

3 « *On a testé... Cubetto, le robot jouet qui apprend aux enfants à programmer* » (6 mai 2016), https://www.lemonde.fr/pixels/article/2016/05/06/on-a-teste-cubetto-le-robot-jouet-qui-apprend-aux-enfants-a-programmer_4914784_4408996.html (Page consultée le 26 février 2025).

4 *Ibid.*

5 *Ibid.*

6 « *Cubetto is a wooden robot that teaches children how to code* », *loc. cit.*

7 *Ibid.*

8 *Ibid.*



Cubetto a été créé par *Primotoys*, une entreprise londonienne fondée par *Filippo Yacob* et *Matteo Loglio*, qui sont deux designers.⁹ *Filippo* a été inspiré par la naissance de son fils pour créer *Cubetto*, afin de propager l'apprentissage de la programmation et de la robotique aux plus jeunes.

Cubetto propose une autre approche de l'apprentissage de la programmation, il n'essaie pas d'être très performant ou précis, mais d'être le plus simple possible, afin d'être accessible dès le plus jeune âge.

Cependant, il permet aux jeunes enfants de travailler leur conscience spatiale car ils doivent donner des ordres précis en respectant les limites techniques du robot pour aller dans des cases spécifiques sur le tapis de jeu.

Ces limites techniques, au contraire, sont avantageuses pour *Cubetto* car elles demandent beaucoup de réflexion de la part du jeune usager. Son design coloré et fait à partir de matériaux naturels lui donne un aspect réconfortant, comme un jouet qu'on retrouverait à l'école maternelle.

Par contre, cette simplicité d'utilisation ne lui donne pas la fonctionnalité de grandir en même temps que les compétences de l'enfant, ce qui ne la rends adaptée qu'aux débutants. Sans cette capacité, on ne peut pas apprendre des concepts de programmation plus complexes et donc l'apprentissage peut stagner.

Cubetto est adapté pour un jeune public, qui lorsqu'il aura passé l'âge, devra trouver un autre dispositif pour acquérir des concepts plus complexes, tels que les variables et les capteurs.

⁹ « *Primo Toys* », dans Wikipédia, [s.l.], 2025, https://fr.wikipedia.org/w/index.php?title=Primo_Toys&oldid=222985426 (Page consultée le 26 février 2025).



LEGO Mindstorms

LEGO + MIT Media Lab
1988

LEGO Mindstorms est une gamme de produits destinée à créer des robots programmables basés sur les briques de construction *LEGO*.¹ Chaque kit comporte une brique "intelligente", des attachements pour cette brique tels que des moteurs et des capteurs, et des pièces de *LEGO Technic*, pour construire la mécanique et donner forme au robot.²

Au contraire des kits de *LEGO* traditionnelle, il ne dispose pas d'un modèle défini à construire. Bien qu'il dispose d'exemples de construction, le kit est ouvert dans l'intention que l'utilisateur crée et programme ses propres robots.³

LEGO Mindstorms a été créé à partir d'une longue collaboration entre *MIT Media Lab* et la compagnie *LEGO*. Sa conception commence dans les années 80, avec *LEGO tc Logo* créé en 1988, accessible uniquement aux Écoles⁴. *LEGO tc Logo* avait pour but de permettre la programmation des constructions *LEGO*, mais il nécessite d'être branché à un ordinateur.⁵

Cette même année, la *MIT Media Lab* conçoit une brique programmable et indépendante, qui est devenue le cerveau du système robotique de *Mindstorms*.⁶ Cette brique programmable va donner vie à aux robots construits avec ce kit sans être branché constamment à l'ordinateur, lui permettant de bouger librement.

1 « *LEGO Mindstorms* », dans *Wikipedia*, [s.l.], 2025, https://en.wikipedia.org/wiki/LEGO_Mindstorms (Page consultée le 26 février 2025).

2 *Ibid.*

3 *Ibid.*

4 Mitchel RESNICK, « *Member collaboration: LEGO Mindstorms* », dans *MIT Media Lab*, <https://www.media.mit.edu/posts/member-collaboration-lego-s-mindstorms/> (Page consultée le 26 février 2025).

5 *Ibid.*

6 *Ibid.*



LEGO Mindstorms est très fortement inspiré par le livre *Mindstorms* de *Seymour Papert*, où il détaille l'enseignement constructionniste pour apprendre la programmation et la robotique aux débutants.⁷

LEGO Mindstorms est très important dans l'enseignement de la robotique et de la programmation. Il permet d'apprendre comment construire un robot complètement fonctionnel et comment ensuite le programmer d'une manière familière, celle de la construction *LEGO*. Cette familiarité des pièces *LEGO* permet aux usagers, petits ou grands, de mieux comprendre ce qu'ils font concrètement. Le fait d'avoir une base commune et standard facilite la collaboration entre usagers et leur permet d'utiliser le kit sans avoir toujours recours à un expert "savant".

Le kit est aussi souvent utilisé pour réaliser des prototypages rapides dans différentes entreprises⁸, y compris des entreprises de robotique. Le kit de construction est ludique et éducatif à la fois.

Il est entouré d'une grande communauté d'utilisateurs en ligne, qui s'adjoignent à de nombreuses ressources et tutoriels gratuits pour apprendre la construction et la programmation.

Il reste flexible et ouvert d'usage, malgré les pièces standardisées et son coût prohibitif.

⁷ Mitchel RESNICK, « *The Seeds That Seymour Sowed* », dans MIT Media Lab, <https://www.media.mit.edu/posts/the-seeds-that-seymour-sowed/> (Page consultée le 26 février 2025).

⁸ Mitchel RESNICK, « *Member collaboration* », loc. cit.



Make: magazine

Dale Dougherty
2005

Make: est un magazine américain publié depuis février 2005. Il se consacre à présenter des projets DIY pour des individus et des groupes¹. Le magazine se concentre sur des projets axés autour de l'informatique, du travail des métaux, de l'ébénisterie, ou de la robotique, ainsi que d'autres disciplines.²

Make: mets en avant les étapes de projets DIY en lien avec le thème du tome. Il inclut également des sections qui parlent de différents outils contemporains comme les imprimantes 3D. Ces sections s'adressent à un public intéressé par le mouvement Maker.³

Make Community LLC, qui publie le magazine *Make:*, a pour but de donner les moyens techniques à tous, peu importe leur âge et leur milieu, d'agir, d'apprendre de leur propre mains, et ainsi de devenir Maker à leur tour.⁴

Avec le succès de *Make: Magazine*, *Make Community* a lancé plusieurs *Maker Faires* depuis 2006.⁵ *Les Maker Faires* sont des événements célébrant le mouvement Maker, et rassemblant la communauté du magazine.⁶

Make:, même s'il est restreint à un public anglophone, permet la propagation d'une nouvelle manière d'apprendre, celle du "faire".

1 « *Make (magazine)* », dans *Wikipedia*, [s.l.], 2024, [https://en.wikipedia.org/w/index.php?title=Make_\(magazine\)&oldid=1250002184](https://en.wikipedia.org/w/index.php?title=Make_(magazine)&oldid=1250002184) (Page consultée le 27 février 2025).

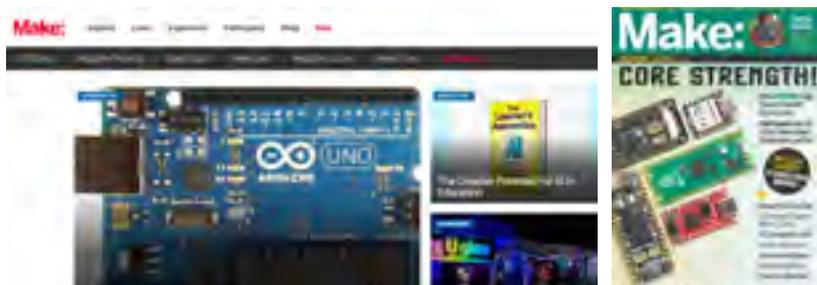
2 *Ibid.*

3 *Ibid.*

4 « *Make: Community* », dans *Make: Community*, <https://make.co/> (Page consultée le 18 mars 2025).

5 « *Make (magazine)* », loc. cit.

6 « *The Best of the Global Maker Movement* », dans *Maker Faire*, <https://makerfaire.com/> (Page consultée le 18 mars 2025).



Il est même considéré comme le bible des bricoleurs, des bidouilleurs, et des créateurs⁷, car il contient de nombreuses ressources : outils, présentation de personnes marquantes du mouvement...

La manière dont le magazine *Make:* diffuse ses idées permet de les rendre accessibles à tous quelque soit le niveau du lecteur.

Le site *Make:* permet aux non abonnés d'accéder à de nombreux articles gratuits et libres des projets à faire, des avis sur des outils, ou des événements *Maker Faire* à venir prochainement.⁸ Ce travail de recueil et de diffusion d'information permet à tous de participer au mouvement Maker et de s'essayer à la programmation et à la manipulation de concept informatiques.

⁷ « *Make: Community* », dans *Make: Community*, <https://make.co/> (Page consultée le 18 mars 2025).

⁸ « *The Latest DIY Projects and News for Makers | Make* », dans *Make: DIY Projects and Ideas for Makers*, 17 mars 2025, <https://makezine.com/blog/> (Page consultée le 18 mars 2025).



Computer Science Unplugged

Tim Bell
1990

Computer Science Unplugged est un ensemble gratuit d'activités d'apprentissage qui enseigne la science de l'informatique à travers des jeux et des casse-têtes captivants en utilisant des cartes, des cordes, des crayons de couleur et beaucoup de remue-ménages¹.

CS Unplugged a d'abord été créé dans les années 90 par *Tim Bell* afin de pouvoir enseigner l'informatique à l'école primaire de son enfant.² Aujourd'hui *Computer Science Unplugged* se trouve librement sur internet ce qui facilite l'accès aux enseignants et aux élèves.³

Computer Science Unplugged est sous la license Creative Commons, ce qui permet aux usagers de créer des variantes, par exemple, certaines activités qui nécessitent un grand terrain de jeu peut être transformé en un jeu de plateau pour les écoles qui ont très peu d'espace⁴.

Généralement, les explications de l'enseignant sont très brèves et les élèves sont tout de suite mis en action et relèvent de nombreux défis, tels que des énigmes, des concours, ou des résolutions de problème.⁵

1 À propos - *CS Unplugged*, <https://www.csunplugged.org/fr/about/> (Page consultée le 27 février 2025).

2 BRETT KERR-LAURIE, « Computer science with a twist: Tim Bell's long list of achievements », *The Press* (29 décembre 2023), <https://www.thepress.co.nz/nz-news/350134595/computer-science-twist-tim-bells-long-list-achievements> (Page consultée le 27 février 2025).

3 *Ibid.*

4 *Principes - CS Unplugged*, <https://www.csunplugged.org/fr/principles/> (Page consultée le 27 février 2025).

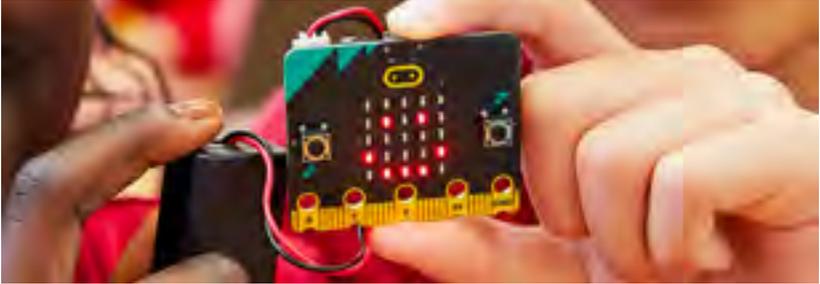
5 *Ibid.*

CS Unplugged propose de changer l'approche de l'apprentissage de l'informatique, et défend l'apprentissage d'une "pensée informatique".

Le fait d'utiliser un matériel tangible permet de dédramatiser l'approche du numérique. De plus l'aspect ludique des objets va quant à lui amener les usagers à mobiliser d'autres compétences que celles développer d'habitude lors de l'apprentissage de l'informatique.

Ensuite, le fait d'avoir des objets permet de réduire les coûts, ce qui rend l'outil accessible à de nombreux espaces scolaires, et à de nombreux publics n'ayant pas les moyens d'accéder à un ordinateur.

Enfin, il permet de changer l'espace d'apprentissage de l'informatique en étant utilisé dans cours de récréation ou dans un parc.



Micro:Bit

BBC
2015

Le *Micro:Bit* est un nano-ordinateur créé par la *British Broadcast Channel (BBC)* en 2015 pour inciter les enfants à apprendre l'informatique et la programmation en faisant leur propre projets.¹

À sa sortie en Royaume-Unis, il a été distribué gratuitement à 1 million d'enfants de 11 ans² afin de faciliter l'apprentissage du numérique, même dans des milieux défavorisés. La carte est très petite et simple ce qui permet de nombreux usages et d'opportunités d'utilisation.

Par exemple, *Micro:Bit* peut être utilisé pour créer un projet centré sur les textiles électroniques, par exemple un tote bag qui peut afficher des emoji, ou même créer des jouets.³

Le *Micro:Bit* s'inspire en partie du *BBC Micro*, un ordinateur personnel des années 80, que l'on trouvait dans chaque foyer britannique pour découvrir l'informatique à la maison.⁴

1 « *Micro:bit* », dans Wikipédia, [s.l.], 2024, <https://fr.wikipedia.org/w/index.php?title=Micro:bit&olddid=219800757#Historique> (Page consultée le 6 janvier 2025).

2 Stuart DREDGE, « *BBC to give away 1m Micro:bit computers to schoolchildren* », *The Guardian* (7 juillet 2015), <https://www.theguardian.com/technology/2015/jul/07/bbc-give-away-1m-micro-bit-computers-schoolchildren> (Page consultée le 7 janvier 2025).

3 « *Projects* », dans Microsoft *MakeCode*, <https://makecode.microbit.org/projects/> (Page consultée le 7 janvier 2025).

4 Stuart DREDGE, « *BBC to give away 1m Micro* », loc. cit.



Le *Micro:Bit* a aussi donné lieu à beaucoup de projets mis en ligne afin de pouvoir inspirer ses utilisateurs pour fabriquer des projets simples et ludiques.⁵

Le *Micro:Bit* permet aux enfants d'avoir une plateforme simple et peu chère à utiliser pour créer différentes choses.

Sa simplicité d'utilisation est encore plus accentuée par le fait que l'enfant qui l'utilise n'a pas besoin de souder quoi que ce soit, et on peut y accrocher différents modules avec des pinces crocodiles et des câbles amovibles. Ceci diminue le danger de se brûler et élimine une grande partie des erreurs.

Dans des quartiers sensibles et qui n'ont pas forcément accès aux outils numériques, le *Micro:Bit* peut venir en aide pour apprendre à manier l'électronique et à la programmation à un faible coût et avec une très grande flexibilité d'usages notamment grâce au logiciel de développement similaire à *Scratch*, et qui est donc familier au plus grand nombre.

5 « Projects », *loc. cit.*

ÉTUDES

DE

CAS

TECH.

FAIRE CONDUIRE L'ÉLECTRICITÉ...

Objectif

Créer un circuit électrique de manière rapide et esthétique, en utilisant d'autres éléments que des fils électriques.

Faire conduire de l'électricité avec de la laine

Protocole

- Tremper des fils de laine dans de la peinture conductive
- Sécher les fils pendant 1 heure.
- Connecter les fils à la pile et à la LED à l'aide de pinces crocodiles.
- Veiller que les fils ne soient pas trop longs, car sinon il n'aura pas assez d'électricité pour allumer la LED.

Observations

La LED brille très faiblement.

Conclusion

- Le courant passe.
- La LED s'éclaire faiblement.
- Plastiquement, c'est intéressant si on travaille la méthode un peu plus et qu'on coude le fil dans un tissu.
- Le temps de réalisation est trop long et trop salissant.



Faire conduire de l'électricité avec un dessin

Protocole

- Dessiner deux traits en utilisant du scotch pour masquer et pour avoir des lignes précises.
- Appliquer plusieurs couches si nécessaire. (cf photos)
- Laisser sécher le dessin.
- Poser les contacts de la batterie dessus et scotcher la DEL.

Observations

La LED brille assez faiblement, malgré un bon contact avec le dessin conductive.

Conclusion

- Conductivité d'électricité assez convenable
- L'esthétique est intéressante et permet d'être polyvalente dans les usages.
- Il faut faire attention à la haute résistance électrique du papier.
- L'application a été très rapide et facile, et permet de créer un petit circuit simple sans souder.



Faire conduire l'électricité avec le papier

Protocole

- Découper des bandes de papier cartons avec une épaisseur maximum de 1 cm.
- Peindre des bandes de papier avec de la peinture conductive.
- La couche appliquée doit couvrir le papier sur un seul côté.
- Laissez sécher pendant 10 minutes
- Appliquer des couches si il n'y a pas assez de conductivité
- Relier la pile et la LED avec le papier.
- Maintenir le contact de la LED avec du ruban adhésif.

Observations

La LED s'éclaire un peu faiblement. Le test a été fait avec deux bandes de papier. La peinture conductive, même étant sèche, est très salissante.

Conclusion

- Le papier peint par la peinture conductive semble intéressant pour conduire de l'électricité, car elle permet de créer des circuits simples sans souder.
- Esthétiquement, ça pourrait être intéressant
- Il faut toujours faire attention à la résistance qu'il a naturellement pour certains circuits sensibles au voltage.



Faire conduire l'électricité avec des pelures végétales

Protocole

- Découper des fines lamelles d'une peau de banane
- Attacher les deux lamelles à la pile à l'aide d'un ruban adhésif.
- Piquer la LED dans les lamelles.

Observations

La LED brille sans besoin d'une résistance pour limiter le courant. La peau de banane conduit assez bien l'électricité mais est salissante et peut pourrir.

Conclusion

- La peau de banane peut conduire l'électricité convenablement
- Il faut faire attention qu'elle ne pourrit pas et qu'elle ne salisse pas trop.
- Elle est peu recommandable pour créer des circuits électriques.
- Mais, elle est intéressante pour créer des circuits éphémères et avec une esthétique plus brute.



DESSINER EN PROGRAMMANT...

Objectif

Dessiner un cercle rouge à l'aide d'un programme numérique de la manière la plus simple et la plus accessible.

Dessiner en programmant avec BASIC

Protocole

- Initier le programme BASIC en définissant les variables à utiliser (R, X, Y, N, et C) avec la commande LET.
- Écrire une définition pour un cercle avec la commande REM et en utilisant des équations pour définir comment tracer un cercle avec la commande PLOT. La commande REM permet de définir un mini-programme à exécuter dans le programme principal.
- Placer la commande pour tracer un cercle avec GOSUB en précisant la ligne 5000
- Finir le reste du programme avec END avant la commande REM, et l'exécuter.

Observations

Le cercle se dessine et est très précis. Par contre, il fallait expliquer mathématiquement c'est quoi un cercle au langage BASIC avec beaucoup de différentes variables qu'on a définies au début.

Conclusion

- Il est archaïque et requiert des notions de mathématiques avancées pour certains usages.
- Pour dessiner un cercle, il fallait expliquer à BASIC c'est quoi avec une formule mathématique.
- Ce n'est pas adapté aux débutants car il n'est pas facile à utiliser, et requiert quelques notions de programmation.

>Quite BASIC
Free, beautiful, and powerful

Quite BASIC is an all web-based (free) BASIC online programming environment.

- Click "Play" and see what happens!
- Or start coding right away. Just add the BASIC program below!

BASIC Program

```

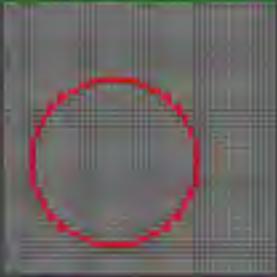
1000 GOTO 1100
1010 INPUT "Name: "; N$
1020 PRINT "Name: " + N$
1030 INPUT "Age: "; A$
1040 PRINT "Age: " + A$
1050 INPUT "Address: "; A$
1060 PRINT "Address: " + A$
1070 INPUT "City: "; C$
1080 PRINT "City: " + C$
1090 INPUT "State: "; S$
1100 PRINT "State: " + S$
1110 INPUT "Zip: "; Z$
1120 PRINT "Zip: " + Z$
1130 INPUT "Phone: "; P$
1140 PRINT "Phone: " + P$
1150 INPUT "E-mail: "; E$
1160 PRINT "E-mail: " + E$
1170 INPUT "Favorite Color: "; F$
1180 PRINT "Favorite Color: " + F$
1190 INPUT "Favorite Food: "; F$
1200 PRINT "Favorite Food: " + F$
1210 INPUT "Favorite Sport: "; F$
1220 PRINT "Favorite Sport: " + F$
1230 INPUT "Favorite TV Show: "; F$
1240 PRINT "Favorite TV Show: " + F$
1250 INPUT "Favorite Music: "; F$
1260 PRINT "Favorite Music: " + F$
1270 INPUT "Favorite Book: "; F$
1280 PRINT "Favorite Book: " + F$
1290 INPUT "Favorite Movie: "; F$
1300 PRINT "Favorite Movie: " + F$
1310 INPUT "Favorite Animal: "; F$
1320 PRINT "Favorite Animal: " + F$
1330 INPUT "Favorite Plant: "; F$
1340 PRINT "Favorite Plant: " + F$
1350 INPUT "Favorite Flower: "; F$
1360 PRINT "Favorite Flower: " + F$
1370 INPUT "Favorite Fruit: "; F$
1380 PRINT "Favorite Fruit: " + F$
1390 INPUT "Favorite Vegetable: "; F$
1400 PRINT "Favorite Vegetable: " + F$
1410 INPUT "Favorite Drink: "; F$
1420 PRINT "Favorite Drink: " + F$
1430 INPUT "Favorite Dessert: "; F$
1440 PRINT "Favorite Dessert: " + F$
1450 INPUT "Favorite Candy: "; F$
1460 PRINT "Favorite Candy: " + F$
1470 INPUT "Favorite Game: "; F$
1480 PRINT "Favorite Game: " + F$
1490 INPUT "Favorite Hobby: "; F$
1500 PRINT "Favorite Hobby: " + F$
1510 INPUT "Favorite Sport: "; F$
1520 PRINT "Favorite Sport: " + F$
1530 INPUT "Favorite TV Show: "; F$
1540 PRINT "Favorite TV Show: " + F$
1550 INPUT "Favorite Music: "; F$
1560 PRINT "Favorite Music: " + F$
1570 INPUT "Favorite Book: "; F$
1580 PRINT "Favorite Book: " + F$
1590 INPUT "Favorite Movie: "; F$
1600 PRINT "Favorite Movie: " + F$
1610 INPUT "Favorite Animal: "; F$
1620 PRINT "Favorite Animal: " + F$
1630 INPUT "Favorite Plant: "; F$
1640 PRINT "Favorite Plant: " + F$
1650 INPUT "Favorite Flower: "; F$
1660 PRINT "Favorite Flower: " + F$
1670 INPUT "Favorite Fruit: "; F$
1680 PRINT "Favorite Fruit: " + F$
1690 INPUT "Favorite Vegetable: "; F$
1700 PRINT "Favorite Vegetable: " + F$
1710 INPUT "Favorite Drink: "; F$
1720 PRINT "Favorite Drink: " + F$
1730 INPUT "Favorite Dessert: "; F$
1740 PRINT "Favorite Dessert: " + F$
1750 INPUT "Favorite Candy: "; F$
1760 PRINT "Favorite Candy: " + F$
1770 INPUT "Favorite Game: "; F$
1780 PRINT "Favorite Game: " + F$
1790 INPUT "Favorite Hobby: "; F$
1800 PRINT "Favorite Hobby: " + F$
1810 INPUT "Favorite Sport: "; F$
1820 PRINT "Favorite Sport: " + F$
1830 INPUT "Favorite TV Show: "; F$
1840 PRINT "Favorite TV Show: " + F$
1850 INPUT "Favorite Music: "; F$
1860 PRINT "Favorite Music: " + F$
1870 INPUT "Favorite Book: "; F$
1880 PRINT "Favorite Book: " + F$
1890 INPUT "Favorite Movie: "; F$
1900 PRINT "Favorite Movie: " + F$
1910 INPUT "Favorite Animal: "; F$
1920 PRINT "Favorite Animal: " + F$
1930 INPUT "Favorite Plant: "; F$
1940 PRINT "Favorite Plant: " + F$
1950 INPUT "Favorite Flower: "; F$
1960 PRINT "Favorite Flower: " + F$
1970 INPUT "Favorite Fruit: "; F$
1980 PRINT "Favorite Fruit: " + F$
1990 INPUT "Favorite Vegetable: "; F$
2000 PRINT "Favorite Vegetable: " + F$

```

Compiler

Compiler generated by
[GBC to HTML app](#)

Executable



Output

Dessiner en programmant avec LOGO

Protocole

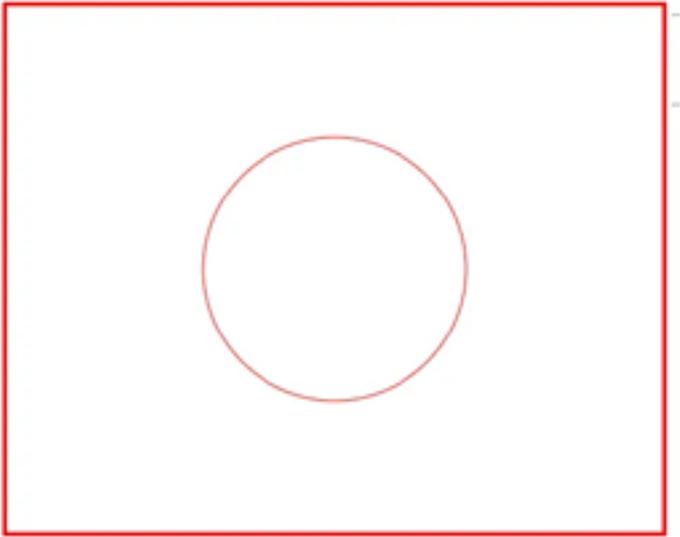
- Centrer la "tortue" (le curseur) avec la commande ct
- Effacer l'écran avec la commande cs (clear screen)
- Cacher le curseur avec la commande ht
- Configurer le crayon à être rouge avec la commande setpencolor "red"
- Poser le crayon avec la commande pd
- Dessiner un arc de 360 degrés avec 100 de rayon avec la commande arc 360 100 afin de dessiner un cercle.

Observations

Le cercle a été très facile à dessiner. Par contre, on ne peut pas varier l'épaisseur de trait. On peut facilement lui donner le rayon de l'arc et lui dire de dessiner un arc de 360 degrés, qui nous donne un cercle.

Conclusion

- LOGO a été très facile à prendre en main pour dessiner un simple cercle, mais il faut savoir que pour lui, un cercle c'est un arc de 360 degrés.
- Les commandes du langage sont assez simples à maîtriser car elles sont très courtes et donc sont adaptées aux débutants.
- Par contre, il n'offre pas beaucoup de flexibilité au niveau de l'épaisseur de trait.



```
ct cs ht setpencolor "red" pd arc 360 100 ↗
```

Run
Clear

Dessiner en programmant avec Scratch

Protocole

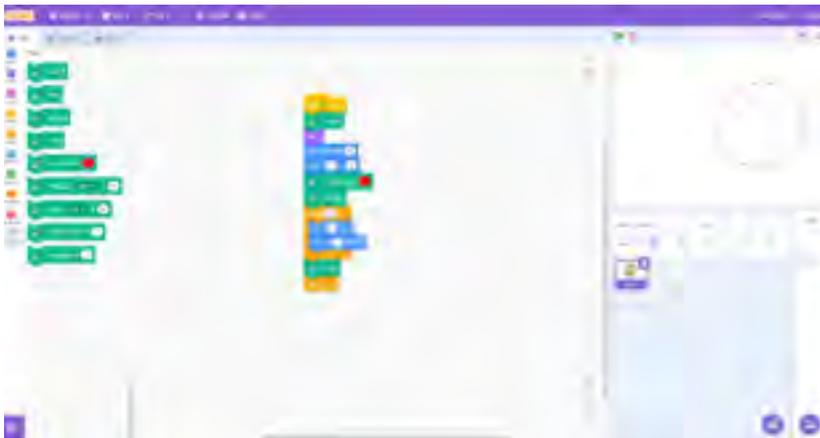
- Effacer tout avec la commande `clear all` qui se retrouve chez les blocs verts émeraudes sous "*Pen*".
- Cacher le chat avec la commande `hide` qui se trouve chez les blocs violets sous "*Looks*".
- Se positionner au centre et mettre la rotation à 0 degrés avec les commandes `point in direction 0` et `go to x: 0 y:0`, qui se trouvent chez les blocs bleus sous "*Motion*".
- Définir la couleur du stylo en étant rouge avec la commande `set pen color to` qui se retrouve chez les blocs verts émeraudes sous "*Pen*".
- Poser le stylo avec la commande `pen down`.
- Tourner en rond pendant 130 fois avec un bloc `repeat 130` qui contient `move 5 steps` et `turn 3 degrees`, qui se retrouvent chez les blocs bleus sous "*Motion*".
- Lever le stylo avec la commande `pen up` qui se retrouve chez les blocs verts émeraudes "*Pen*."
- Ajouter le bloc "`stop all`" pour signaler la fin du programme.

Analyse

Ça a été un peu plus difficile de faire quelque chose de bien précis. Malgré la définition des origines du cercle, il se dessine un peu partout et on n'arrive pas à deviner quelle taille il va avoir à la fin.

Conclusion

- La simplicité de Scratch est très intéressante et la rend accessible à tout le monde.
- Pas assez de précision.
- La taille invariable de l'écran de sortie n'est pas idéal non plus.



Dessiner en programmant avec Processing

Protocole

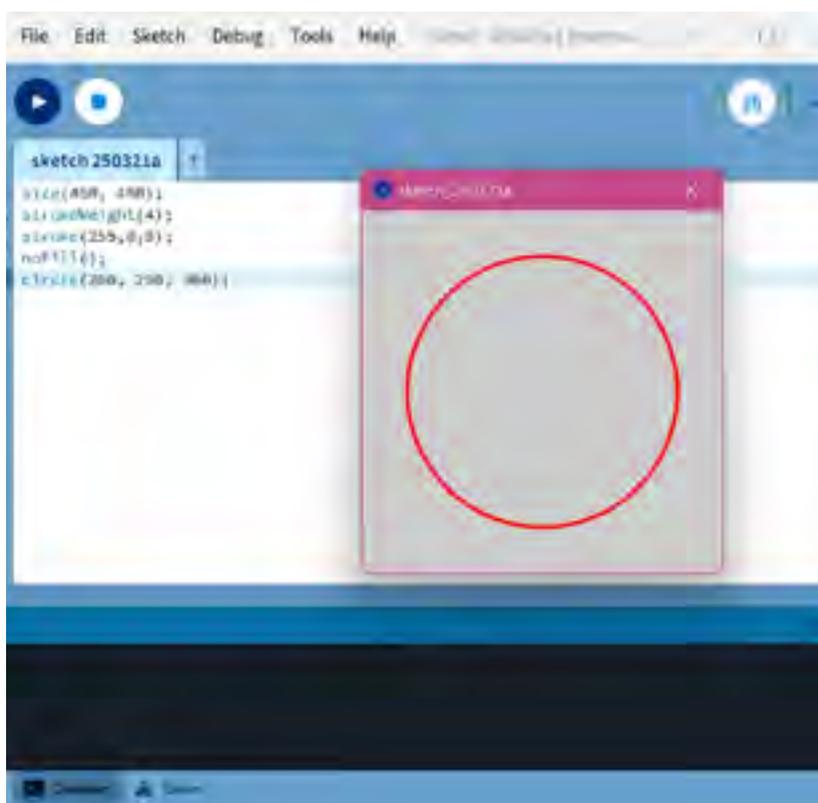
- Définir la taille du plan de travail avec la commande `size`.
- Définir l'épaisseur du trait avec la commande `strokeWeight`.
- Définir la couleur du trait pour qu'elle soit rouge avec la commande `stroke`.
- Définir qu'on ne veut pas que le cercle soit rempli avec la commande `noFill`.
- Dessiner un cercle en précisant sa position de départ X et Y, ainsi que son rayon avec la commande `circle`.
- Exécuter le programme avec le bouton jouer.

Analyse

Processing, étant un logiciel créé pour dessiner avec du code, a été très facile pour dessiner un cercle. Il y a déjà une commande "cercle" qui est intégrée dans le langage de codage donc pas besoin de le définir manuellement à l'aide des formules géométriques compliquées.

Conclusion

- Processing est très adapté pour dessiner avec du code, car c'est son usage premier.
- Il offre beaucoup de précision, jusqu'à choisir la taille de son plan de travail et l'épaisseur de son trait.
- Facile à prendre en main.



Dessiner en programmant avec AIBO

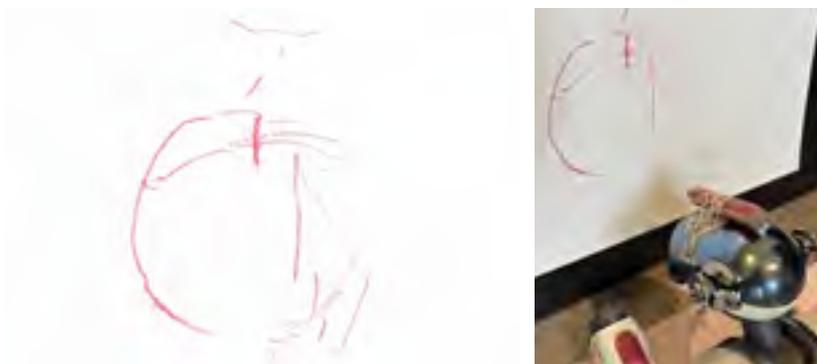
Protocole

- Attacher à l'aide d'un morceau de ruban adhésif un crayon-craie rouge sur le museau d'un Sony AIBO ERS-7.
- Placer l'AIBO devant une feuille A3 blanche accroché à un mur.
- Allumer l'AIBO.
- Se connecter à l'AIBO par WI-FI.
- Ouvrir le logiciel de télécommande AIBO Entertainment Player sur son PC.
- Commander l'AIBO à tourner la tête afin de dessiner un cercle sur la feuille avec la manette de contrôle de vue sur le logiciel.

Explication de l'interface Navigator de AIBO Entertainment Player V2.0

- À gauche, un modèle 3D qui montre la posture de l'AIBO.
- En centre, le point de vue de l'AIBO.
- À droite, le point orange permet de contrôler la position de la tête de l'AIBO afin de regarder autour de soi.
- En bas à droite, est la télécommande pour manoeuvrer le corps de l'AIBO





Analyse

Le robot a été très difficile à contrôler précisément car on a un contrôle trop grossier sur le mouvement de la tête. Il y a souvent des fois où, si il tourne la tête trop vers la gauche, on perd le tracé du crayon. Ce n'était pas très ergonomique, ni pour le robot, ni pour l'utilisateur.

Conclusion

- Aucune précision dans les traits.
- L'AIBO n'a pas été codée comme les autres études de cas techniques, mais contrôlée par un logiciel télécommande sur le PC.
- Pas facile à commander pour dessiner un cercle parfait.
- L'AIBO reste quand même intéressant car il permet de dessiner sur des supports physiques, et on peut envisager d'autres moyens de dessiner avec lui

ÉTUDES

DE

CAS

ARTISTIQUE



XENODOG
Dog 50%
Pig 45%
Human 5%



COPYCAT
Dog 50%
Cat 50%



DOLLY
Dog 80%
Cow 20%



JELLYBOGGY
Pig 40%
Jellyfish 35%
Chameleon 25%



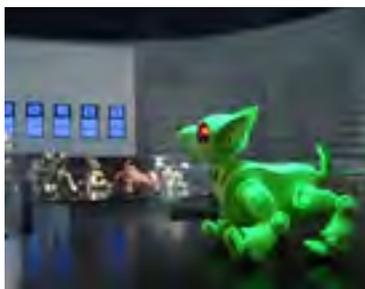
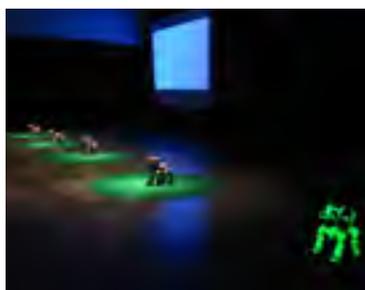
GFP PUPPY
Dog 95%
GFP 5%



SCHIZOGG
Pig 50%
Mr. Magill 25%
Mister Hyde 25%



FlyingPig
Dog 50%
Pig 50%
Human 5%



MILYBOGGY
Chien 90%
Mole 5%
Cameleon 5%

COPYCAT
Chien 50%
Cat 50%

FLYINGPIG
Chien 50%
Pig 50%
Human 5%

Dog[LAB]01

France Cadet
2003

France Cadet est une artiste française qui explore la robotique et les nouveaux médias. Son travail aborde de façon ironique diverses questions de la science qui font débat, révélant nos peurs actuelles sur les biotechnologies et l'intelligence artificielle. Elle explore parodiquement la relation de l'homme à l'animal et de l'humain à l'androïde, en jouant sur la frontière, de plus en plus poreuse, entre le naturel et l'artificiel.¹

Dog[LAB]01 est une installation artistique créée en 2003 qui mélange le piratage robotique avec de la modification et expérimentation génétique.²

France Cadet va modifier et hacker 7 chiens robot *i-Cybies*³ afin de créer des chiens hybrides qui seront ensuite accompagnés de fiches signalétiques détaillant leur nom, leur origine génétique et leurs principales caractéristiques. Ils évoluent dans des espaces d'observation individuels, tels ceux d'un laboratoire.⁴

Dog[LAB]01 me semble intéressant dans la manière dont l'artiste *France Cadet* reprend des chiens robots destinés à être des simples jouets robotiques pour enfant et détourne leur usage afin de créer un œuvre jouant avec le sujet de la modification génétique. À travers l'utilisation et de piratage électronique et esthétique du chien robot, elle leur donne des nouvelles formes et de fonctions qu'ils ne sont pas prévues pour, comme si elle faisait de la modification génétique sur ces chiens robots.

Le détournement et le piratage du robot dans cette installation est utilisé comme un outil pour critiquer et remettre en question la place de l'Homme à créer des nouvelles espèces et à jouer "dieu" avec la nature, qui est d'autant plus soulignée par les différentes espèces créées par l'artiste, qui sont inspirées d'exemples vraies, comme *GFP Puppy*, qui est directement inspirée de *GFP Bunny/Alba*, une lapine vivante fluorescente créé par *Eduardo Kac*.⁵

Dans un sens, l'artiste a aussi joué au dieu avec la modification des chiens robots pour son installation, en leur donnant un nouvel usage.

1 Site de l'artiste France Cadet, https://www.francecadet.com/index_fr.html (Page consultée le 4 février 2025).

2 France CADET, « Biographie - Démarche artistique », [s.d.], https://www.francecadet.com/pdf/2023_France_Cadet_monographie_web.pdf (Page consultée le 4 février 2025).

3 *i-Cybie est un jouet robotique créé par Silverlit dans les années 2000 sur 4 pattes avec une apparence canine. Originellement on ne peut pas le reprogrammer mais à travers le hacking on peut ajouter de nouvelles fonctionnalités.*

4 France CADET, « Biographie - Démarche artistique », loc. cit.

5 Ibid.



Drawing Operations

Sougwen Chung
2015

Sougwen Chung est une artiste et chercheuse canadienne-chinoise qui se spécialise dans la collaboration entre humain et machine. Elle explore le tracé-fait-par-la-main et le tracé-fait-par-la machine comme une approche de mieux comprendre la dynamique entre l'humain et des systèmes informatiques.¹

Drawing Operations est une collaboration entre un artiste et un bras robotique nommé *D.O.U.G._1 (Drawing Operations Unit: Generation_1)*. Le robot *D.O.U.G._1* imite le geste de l'artiste grâce à une caméra et dessine en même temps que lui, qui donne lieu à une performance synchronisée entre les deux.²

Cette installation est une première étape d'une recherche qui vise à examiner l'interaction humain-robot comme une collaboration artistique.³

D.O.U.G._1 imite chaque geste que l'artiste va faire, tel comme un chant improvisé en duo.

L'IA de ce robot observe et transmet chaque bogue, erreur, et glitch qu'il trouve. Cette session de dessin, sans aucune harmonie ou de règles préétablies par l'artiste ou la machine, va se libérer de quelques contraintes esthétiques, en examinant l'essence et le phénomène de beauté en même temps.⁴

La collaboration entre humain et robot est un sujet important dans cette œuvre d'art, ainsi que la place de l'IA et du robot dans le monde de l'art.

Ici, le robot et l'artiste travaillent ensemble, avec le robot qui imite et apprend à partir de l'humain d'une manière programmée.

D'une manière, on peut qualifier ce robot en étant un apprenti à l'art, avec l'artiste le guidant et lui apprenant des nouvelles fonctionnalités, comme si on le programmait en dessinant.

1 Info - Sougwen Chung (愷君), <https://sougwen.com/info> (Page consultée le 4 février 2025).

2 *Drawing Operations (2015)* - Sougwen Chung (愷君), <https://sougwen.com/project/drawing-operations> (Page consultée le 4 février 2025).

3 *Ibid.*

4 *Ibid.*



The Woodpeckers

Marco Barotti

2019

Marco Barotti est un artiste multimédia. Après ses études de musique au *Siena Jazz Academy*, il a commencé à mélanger le son avec les arts visuels. Ses travaux sont dirigés par une volonté d'inventer un langage artistique où on exprime une ère fictionnelle post-futuriste à travers des interventions auditives kinétiques dans des environnements naturels et urbains. Ses installations artistiques mélangent la technologie audio, les objets de consommation et les déchets en sculptures mobiles déclenchées entièrement par le son. ¹

Les woodpeckers sont de curieux oiseaux robotiques qui déambulent dans les rues. Ils sont des petits robots qui réagissent aux ondes électromagnétiques émis par divers objets électroniques² en tapant sur la surface qu'ils sont mis.

C'est là le projet interactif de *Marco Barotti* : retranscrire en rythmes sonores et visuels les ondes générées par nos appareils de communication mobiles, et sensibiliser le public à la présence de ces radiations invisibles. ³

Les woodpeckers sont intéressants dans la manière dont le robot est utilisé pour sensibiliser les gens à des phénomènes qu'ils ne peuvent pas percevoir.

Ici, *Les woodpeckers* en tant que robots ne sont pas forcément mis en avant, mais leur utilité de détecter et d'alerter les autres d'ondes électromagnétiques invisibles qui sont émises en permanence par nos appareils quotidiens.

Les woodpeckers fonctionnent comme les canaries qu'on retrouvait dans les galeries des mineurs, en alertant les mineurs des gaz toxiques ou de manque d'oxygène. Ils rendent l'invisible visible, tel comme un robot qui exécute un programme qu'au début on comprend pas en la lisant simplement, mais en l'exécutant, on comprend et on voit la programme en mouvement, grâce au robot.

¹ About - [marcoarotti.com](https://www.marcoarotti.com), <https://www.marcoarotti.com/about-1> (Page consultée le 4 février 2025).

² Chloé CURA, « The Woodpeckers », dans *Bien Urbain*, 17 juillet 2019, <https://juste-ici.fr/fr/the-woodpeckers/> (Page consultée le 4 février 2025).

³ Ibid.



Nonsense Machines

MAYWA DENKI
2001 - 2010

Maywa Denki est une unité artistique dirigée par *Novmichi Tosa*. Elle porte le nom de la société que le père de l'artiste dirigeait en son temps. Le costume de scène du collectif *Maywa Denki* est calqué sur l'uniforme de travail typique des magasins d'électricité japonais – l'incarnation des petites et moyennes entreprises qui avaient tant contribué à l'économie du Japon à l'époque de son incroyable croissance.¹

Le style unique de *Maywa Denki* est mis en évidence par un terme que l'artiste *Novmichi Tosa* utilise : par exemple, chaque pièce du projet *Maywa Denki* est appelée un « produit », et les expositions ou les spectacles sont organisés en tant que « démonstrations de produit ».²

Dans *Nonsense Machines*, on retrouve plusieurs “produits” ludiques et un peu farfelus qui tournent autour du son et de la mécanique.

Dans cette “démonstration de produits”, on retrouve le *Seamoon*, qui est un robot chantant contrôlé par ordinateur.³ Le robot *Seamoon* a des cordes vocales artificielles créées en caoutchouc semblables à un accordéon qui lui permettent de chanter avec d'autres robots-instruments.⁴

Comme le nom “*Nonsense Machines*” indique, c'est des machines et des robots créés sans aucun but précis. C'est des œuvres absurdes qui interrogent la place de la machine et de sa fonction.

Avec le vocabulaire et la présentation utilisés par *Maywa Denki* en appelant leurs œuvres des produits et leurs expositions des démonstrations de produit, on peut aussi questionner la place de l'art dans une ère industrielle. On peut aussi se demander l'intérêt des robots aussi absurdes qui n'accomplissent pas des tâches “utiles” mais qui sont là juste pour exister en tant que chanteurs et des musiciens automatisés et contrôlés par un opérateur de machine qui les programme pour jouer ensemble.

Ici, *Maywa Denki* devienne une usine pour des machines de non-sens qui critique notre époque actuelle centrée sur la surconsommation et le capitalisme.

1 YOSHIMOTO CREATIVE AGENCY CO., LTD, *Biographie de Maywa Denki, Cité des Sciences, [s.d.]*, https://www.cite-sciences.fr/fileadmin/fileadmin_CSI/fichiers/au-programme/expos-temporaires/art-robotique/_documents/Maywa-Denki.pdf (Page consultée le 2 avril 2025).

2 *Nonsense Machines - Maywa Denki - L'exposition - Art Robotique - Expositions passées - Ressources - Cité des sciences et de l'industrie*, <https://www.cite-sciences.fr/fr/ressources/expositions-passees/art-robotique/l'exposition/nonsense-machines-maywa-denki> (Page consultée le 4 février 2025).

3 *Ibid.*

4 MAYWADENKI, « セーモンズ / Seamoons », dans *明和電機 - Maywa Denki*, 29 août 2012, <https://www.maywadenki.com/products/voice/seamoons/> (Page consultée le 24 février 2025).



A Mathematical Theory of Communication

Casey Reas
2015

Casey Reas est un artiste américain dont les œuvres conceptuelles, procédurales et minimales explorent le potentiel créatif de la programmation informatique. Reas est notamment connu pour avoir créé, avec *Ben Fry*, le langage de programmation *Processing*.¹

Casey Reas a étudié le design à l'Université de Cincinnati. Il a ensuite passé deux ans à explorer le potentiel artistique de la programmation et de l'électronique.²

"*A Mathematical Theory of Communication*" est une œuvre créée en 2014, qui est une grande fresque qui couvre deux murs à côté de l'atrium du complexe de *Gates Dell*, qui abrite le département des sciences de l'informatique à l'université de Texas au campus de la ville d'Austin.³ Prenant son nom de l'œuvre de *Claude Shannon*, cette œuvre mélange l'information et l'art conceptuel pour peindre une cartographie de données numérique.⁴ Pour pouvoir fabriquer cette cartographie numérique, Reas a dû capturer les images télévisées avec une antenne et il les a traitées à travers des algorithmes créés par lui, ensuite ces images abstraites sont de nouveau re-traitées par un autre algorithme afin de générer l'œuvre finale.⁵

L'œuvre "*A Mathematical Theory of Communication*" est intéressante dans la manière dont elle utilise la programmation et la captation de données d'images télévisées pour créer une cartographie de différentes ondes électroniques. Comme une grande partie des œuvres ici, la programmation est utilisée ici pour nous montrer l'invisible à travers un dessin réalisé purement par des algorithmes et de données récoltés par l'artiste. Le nom de l'œuvre est aussi tiré d'un article scientifique écrit par *Claude Shannon*, qui a fondé la théorie de l'information ainsi que la théorie des télécommunications⁶. *A Mathematical Theory of Communication* démontre bien les théories complexes de l'information et de télécommunication à travers les processus que Casey Reas a utilisé pour la créer et les a rendu visible à travers la fresque créée par des milliers de bits d'information.

1 « Casey Reas », dans Wikipédia, [s.l.], 2024, https://fr.wikipedia.org/w/index.php?title=Casey_Reas&oldid=220861121 (Page consultée le 20 février 2025).

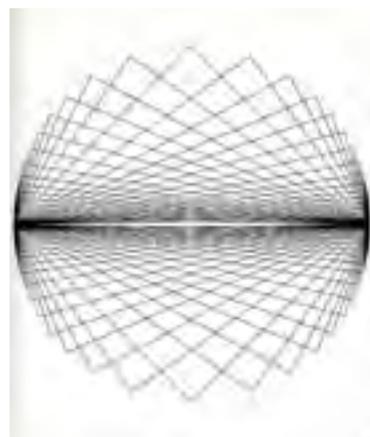
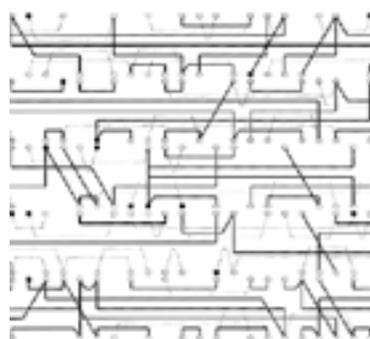
2 Ibid.

3 *A Mathematical Theory of Communication* | Americans for the Arts, <https://www.americansforthearts.org/by-program/networks-and-councils/public-art-network/public-art-year-in-review-database/a-mathematical-theory-of-communication> (Page consultée le 9 février 2025).

4 Casey Reas, « *A Mathematical Theory Of Communication* » | LANDMARKS, <https://landmarks.utexas.edu/artwork/mathematical-theory-of-communication> (Page consultée le 20 février 2025).

5 Ibid.

6 « *A Mathematical Theory of Communication* », dans Wikipédia, [s.l.], 2024, https://fr.wikipedia.org/w/index.php?title=A_Mathematical_Theory_of_Communication&oldid=213731942 (Page consultée le 24 février 2025).



Computer Graphics, une esthétique programmée

Manfred Mohr

1971

Manfred Mohr est un artiste allemand considéré comme un des pionniers de l'art numérique.¹ Il a commencé à utiliser un ordinateur depuis 1969 parce qu'il était très intéressé à créer de l'art algorithmique. Il va passer de l'art expressionniste à de l'art géométrique et algorithmique. *Pierre Barbaud*, un pionnier de la musique créé par ordinateur, va aussi beaucoup l'encourager à créer de l'art algorithmique.

Computer Graphics, une esthétique programme est la première exposition d'art algorithmique qu'il fait en 1971 au *A-R-C Musée d'Art Moderne* de la Ville de Paris.²

Cette exposition montre les nombreuses œuvres programmées créées par *Manfred Mohr* sur le traceur électronique Benson. Chaque œuvre dans l'exposition est un programme différent qui est tracé sur du papier.³ Pour cette époque, l'utilisation de l'ordinateur dans la création de l'art est loin d'être une conception traditionnelle⁴. *Manfred Mohr* construit plusieurs tableaux complexes à travers l'aléatoire de la programmation et en utilisant la précision du traceur.⁵

Computer Graphics, une esthétique programme est une exposition assez importante dans l'histoire de l'art numérique, car elle est la première à sensibiliser un public à des nouvelles méthodes de création d'art en utilisant la programmation et les traceurs industriels.⁶

Avec cette exposition, de nombreuses personnes ont pu découvrir un nouveau monde de conception et l'art génératif créé par la programmation informatique et des algorithmes. Les œuvres présentées dans l'exposition sont toutes des résultats de différents programmes et qui nous donnent un visuel concret sur les décisions que l'algorithme a prises.

Le fait que c'est aussi en 1971 que l'exposition se déroule est aussi très intéressant niveau historique, car l'ordinateur restait inconnu pour la plupart des personnes et avait une connotation stricte avec des grandes entreprises et de la domaine scientifique. Ici, l'exposition aide à démystifier la programmation et les ordinateurs à travers l'art numérique.

¹ « Manfred Mohr », dans *Wikipedia*, [s.l.], 2025, https://en.wikipedia.org/w/index.php?title=Manfred_Mohr&oldid=1268923043 (Page consultée le 7 février 2025).

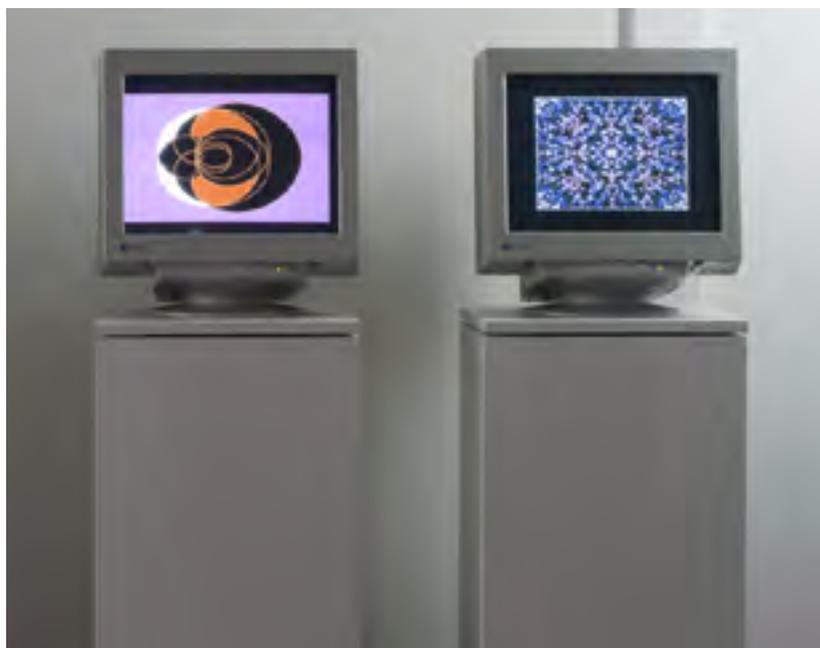
² Manfred MOHR et al., *Manfred Mohr Computer Graphics, Une esthétique programmée, A-R-C Musée d'Art Moderne de la Ville de Paris, 1971*, <http://www.emohr.com/paris-1971/index.html> (Page consultée le 7 février 2025).

³ *Ibid.*

⁴ *Ibid.*, p. 10.

⁵ Manfred MOHR et al., « Manfred Mohr Computer Graphics, Une esthétique programmée », *loc. cit.*

⁶ Luca BENDANDI et al., *Tracing the line: the art of drawing machines and pen plotters, Berlin, Vetro, 2023*, p. 23.



ORCHID

Herbert W. Franke

1984 - 1992

Herbert W. Franke est un auteur de science-fiction et un pionnier du graphisme informatique autrichien. Il étudie la physique, les mathématiques, la chimie, la psychologie et la philosophie à *l'Université de Vienne*.¹ Il a commencé à expérimenter avec l'art informatique dans les années 60 pour créer des dessins au traceur et de l'art interactif sur l'ordinateur² dans un but de recherche esthétique et de créer des méthodes analytiques de savoir pourquoi que fait la beauté dans une œuvre d'art.³

Dans l'œuvre interactive *ORCHID* réalisée par *Herbert W. Franke* avec un micro-ordinateur de type DEC Professional 350, des champs de couleur rectangulaires se forment petit à petit, en partant de la partie inférieure de l'écran. Ils se superposent, s'alignent les uns à côté des autres. Ici, il est possible de déplacer le graphique en bloc vers le haut ou vers le bas en utilisant le clavier.⁴

Parmi *ORCHID*, il y a aussi *TROPIC*, *WALDWOHN*, *ABROLL3* et *RAHMEN4* qui sont tous des programmes qui créent des œuvres abstraites géométriques qui s'inspirent de l'aléatoire.⁵

Herbert W. Franke avait créé ces œuvres entre 1984 et 1992 en expérimentant différentes notions de mathématiques et de physique.

Il met en scène la programmation pour créer des nouveaux œuvres chaque fois qu'on interagit avec *ORCHID* avec le clavier, laissant le spectateur choisir la position du bloc et ainsi de la composition finale de l'œuvre.

Le fait que *ORCHID* se trouve sur un écran et non sur du papier tracé par un traceur lui donne beaucoup de liberté de pouvoir évoluer comme le spectateur veut. *ORCHID* nous surprend aussi car le programme est aléatoire dans les décisions qu'il prend, tel comme une fleur qui pousse à sa manière.

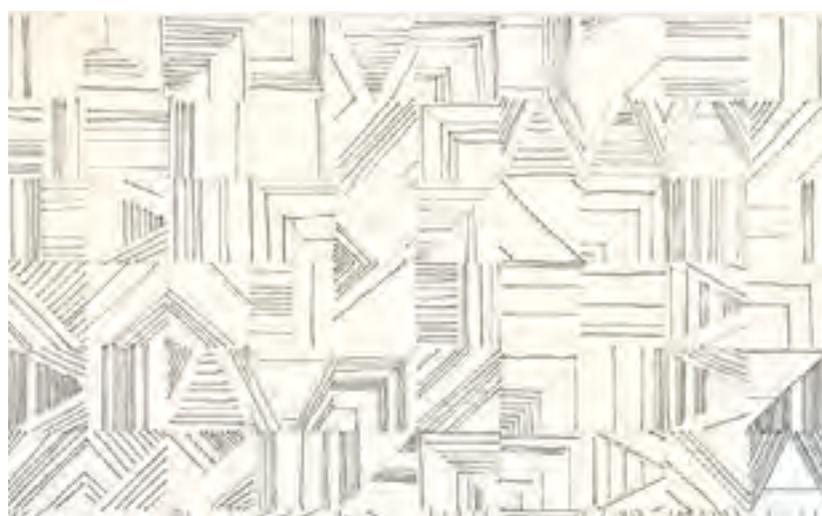
¹ « Herbert W. Franke », dans Wikipédia, [s.l.], 2023, https://fr.wikipedia.org/w/index.php?title=Herbert_W._Franke&oldid=210764106 (Page consultée le 9 février 2025).

² Brian DROITCOUR, « Herbert W. Franke & Casey Reas », dans *Outland*, 13 mai 2022, <https://outland.art/herbert-w-franke-casey-reas/> (Page consultée le 9 février 2025).

³ *Ibid.*

⁴ « Herbert W. Franke », dans CEAAC | Centre Européen d'Actions Artistiques Contemporaines, <https://ceaac.org/fr/artiste/herbert-w-franke-2/> (Page consultée le 7 février 2025).

⁵ *Ibid.*



À la recherche de Paul Klee

Vera Molnar
1970

Vera Molnar est une artiste plasticienne d'origine hongroise. Elle est connue en tant que pionnière de l'art numérique.¹ Ses œuvres, en s'enrichissant de connaissances sur la psychologie de la forme et des lois de la vision, deviennent des questionnements plastiques de l'optique.

Cybernéticienne puis informaticienne, **Molnár** met en place dans les années 1960 un mode de production qu'elle nomme « machine imaginaire » avant d'être la première artiste en France à produire des dessins numériques en utilisant un ordinateur relié à une table traçante en 1968.²

À la recherche de Paul Klee est justement une des premières séries d'œuvres créées par **Vera Molnar** qui utilise la "**machine imaginaire**".

Cette œuvre est inspirée de l'oeuvre "**Variations**" créé par **Paul Klee** en 1927, avec le motif principal étant un échiquier³.

"**La machine imaginaire**" de **Vera Molnar** est une programme simple qui impulse une transformation des formes, combinant consignes et interdits. Elle décide des formes, des couleurs, des textures, des matériaux, des supports et dessine sur un rouleau les propositions qu'elle retient.⁴

Cette machine imaginaire est conçue en 1959 pour anticiper l'outil informatique.⁵

La machine imaginaire est très importante pour la création de l'art numérique car elle donne un exemple d'un programme destiné à concevoir et à créer de l'art à travers différentes règles donnés par l'artiste qui le code. Avant l'ordinateur et le traceur, ce sera l'artiste qui suit les instructions données par la machine imaginaire, tel comme un problème mathématique.

À la recherche de Paul Klee est une œuvre qui se sert beaucoup de la machine imaginaire pour décider des lignes qui sont dessinées et qui permet de mieux visualiser le programme. Cette œuvre va aussi permettre de mieux étendre l'idée de **la machine imaginaire** aux autres artistes.

1 Vera Molnár - Parler à l'œil, 5 février 2024, https://www.centrepompidou.fr/fr/programme/agenda/evenement/PA7jRZ5?utm_medium=EMAIL&utm_source=INFORMATION (Page consultée le 9 février 2025).

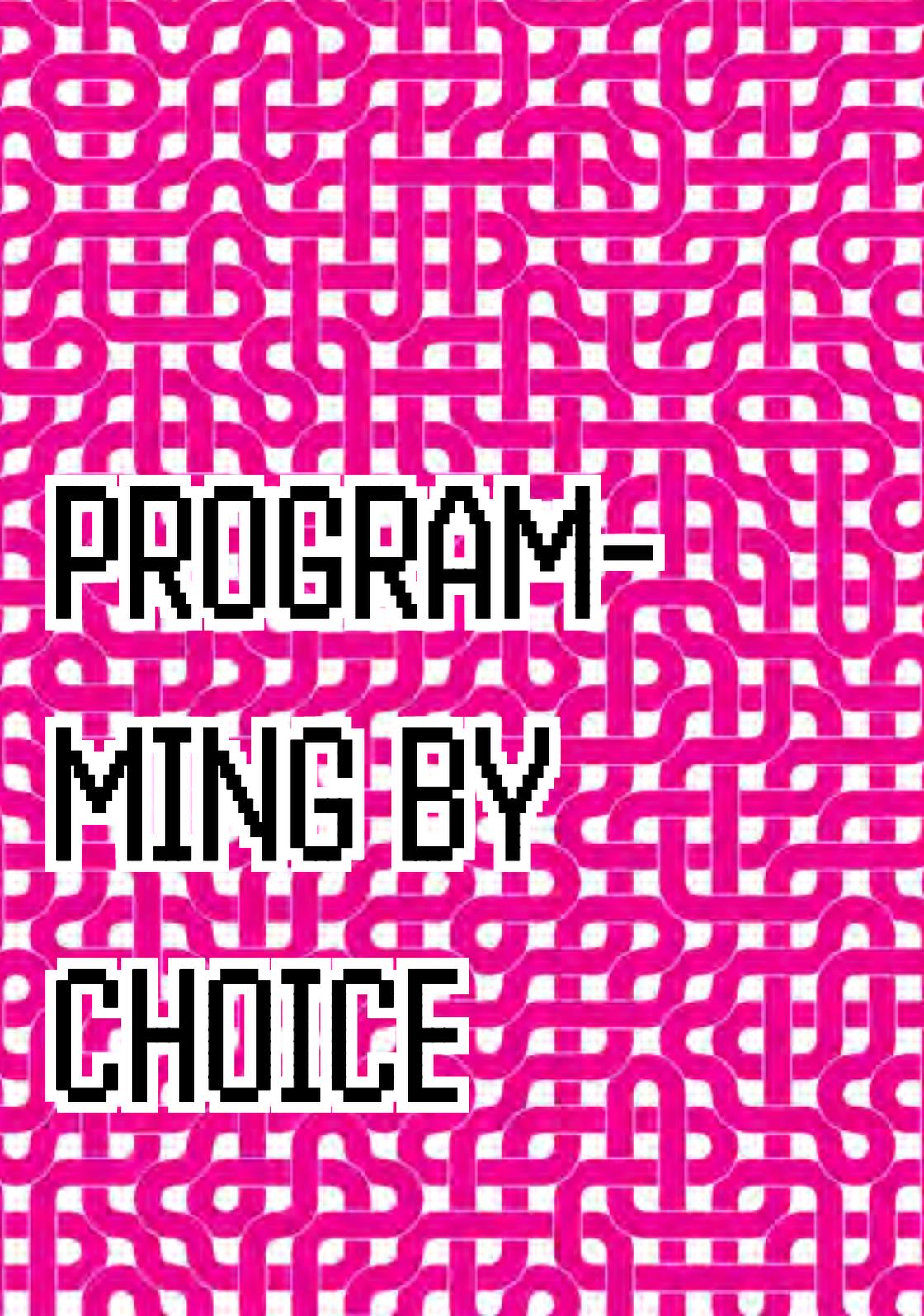
2 Ibid.

3 À la Recherche de Paul Klee (Looking for Paul Klee) | The Anne + Michael Spalter Digital Art Collection, <https://spalterdigital.com/artworks/223/> (Page consultée le 20 février 2025).

4 Vera Molnár, aux sources du code, 28 novembre 2023, <https://www.centrepompidou.fr/fr/magazine/article/vera-molnar-aux-sources-du-code> (Page consultée le 9 février 2025).

5 Vera Molnar, <https://radicalweb.design/ressources/css/layout/processus/examples/05.html> (Page consultée le 24 février 2025).

SYNTHÈSES
DE
LECTURE



**PROGRAM-
MING BY
CHOICE**

Programming by choice

John Maloney, Kylie Peppler, Yasmin B. Kafai, Mitchel Resnick et Natalie Rusk
2008

Programming by choice est un article scientifique publié par L'Association for Computing Machinery en 2008 qui explore l'expérience des jeunes âgés entre 8 et 18 ans avec Scratch.

Contexte

Les recherches ont commencé en 2005 dans un quartier populaire de Los Angeles dans un centre d'informatique périscolaire qui sert les jeunes âgés entre 8 et 18 ans issus de familles défavorisées, qui sont souvent issus d'immigration¹. La recherche a duré 18 mois durant une période de plein essor de l'utilisation de l'informatique et de la programmation. Les chercheurs utilisent le logiciel Scratch pour permettre aux jeunes d'apprendre la programmation, pratique encore peu accessible au grand public.² Il convient de noter qu'avant cette recherche, personne dans le centre d'informatique périscolaire ne savait ce qu'était Scratch.

Méthodologie de recherche

Les auteurs introduisent Scratch en 2005 au centre informatique périscolaire pour l'apprentissage des bases du codage. Les chercheurs laissent les enfants libres de choisir leur projet pour le travailler sur Scratch et les jeunes peuvent demander de l'aide aux animateurs pour certains aspects, notamment artistiques.³

Après quelques semaines, les chercheurs organisent des "Scratch-a-thons", des Hackathons, où dans un temps limité les jeunes doivent rendre un projet complet. Les enfants disposent 3 ou 4 heures pour chaque "scratch-a-thon", et ils présentent leur travail au groupe.⁴

Les enfants sont libres de réaliser des jeux vidéo, des clips musicaux, des cartes de vœux et des animations avec le logiciel.

Pendant ces 18 mois, les chercheurs collectent tous les projets Scratch réalisés par les jeunes pour analyser les concepts de programmation appris en autonomie.⁵

À la fin, les chercheurs interrogent 30 jeunes pour avoir leur ressenti face à Scratch.⁶

1 John H. MALONEY et al., « Programming by choice: urban youth learning programming with scratch », *SIGCSE Bull.*, vol. 40, no 1, 12 mars 2008, p. 367-371, <https://doi.org/10.1145/1352322.1352260>.

2 *Ibid.*

3 *Ibid.*

4 *Ibid.*

5 *Ibid.*

6 *Ibid.*

Les concepts de programmation appliqués par les jeunes

Les chercheurs ont collectionné 34% des projets (sur un total de 536 projets Scratch) et ils les ont analysés pour voir quelles commandes les enfants ont utilisées le plus dans leur projet.¹

Ils ont remarqué que les jeunes faisaient beaucoup appel à l'interaction de l'utilisateur et aux boucles en programmation, car il y a beaucoup de projets qui sont centrés sur les jeux vidéo.

Par contre, certaines commandes plus complexes comme les opérations booléennes et les nombres aléatoires n'étaient pas aussi évidentes à acquérir en autonomie.²

Il y a eu un cas où un jeune voulait se servir des variables dans son projet, mais il ne savait pas comment procéder.

Quand Mitchel Resnick, qui était un des chercheurs, a visité le centre périscolaire, il a pu expliquer au jeune comment les variables fonctionnent et le jeune a résolu son problème par lui-même.³

Petit à petit, davantage de jeunes ont expérimenté avec des concepts plus complexes et ils ont pu faire des projets encore plus aboutis en utilisant leurs nouvelles connaissances.

Le ressenti des jeunes face à Scratch

Après leur étude sur les données quantitatives à partir des projets récoltés, ils interrogent 30 jeunes pour avoir leur avis sur Scratch.

Les chercheurs ont demandé aux jeunes "Si Scratch devait être quelque chose qui n'est pas sur un PC, qu'est-ce cela serait ?" pour guider leur réflexion.⁴

La réponse la plus fréquente est celle du "papier" ou "de cahier d'esquisse". En effet, selon eux, "Scratch permet de faire ce que l'on veut, comme le papier".⁵

Il y a eu des réponses similaires, en disant que Scratch peut être tout car ça peut "être ton propre monde créatif".⁶ Pas mal d'enfants avaient des avis assez positifs par rapport à Scratch, sauf un, car lui il était plus habitué à utiliser Flash et Scratch ne lui laissait pas faire ce qu'il voulait.⁷

1 *Ibid.*

2 *Ibid.*

3 *Ibid.*

4 *Ibid.*

5 *Ibid.*

6 *Ibid.*

7 *Ibid.*

Après, les chercheurs ont demandé aux jeunes si Scratch leur rappelait quelque chose en lien avec ce qu'ils étudient à l'école.

Une des réponses les plus courantes était que Scratch ressemblait aux cours d'art, car il est proche des diverses formes d'art, que ce soit le dessin, la sculpture, le théâtre, la musique, ou la danse.

Scratch ressemble aux activités qui permettent de s'exprimer personnellement et créativement, comme l'art ou les cours de langue.⁸

À la question sur leurs connaissances en programmation, les jeunes répondent qu'ils "ne connaissent pas la programmation", et qu'ils ne voient pas Scratch comme un langage de programmation.⁹

Avec du recul, l'avantage de ne pas voir Scratch comme de la programmation, a permis au logiciel de mieux se propager dans le centre périscolaire.

L'importance du lieu

Dans cet article scientifique, le centre périscolaire d'informatique "The Clubhouse" dans le quartier populaire de Los Angeles a été crucial pour les recherches.

The Clubhouse a permis aux jeunes du quartier d'accéder librement aux ordinateurs, pendant une période où l'informatique commençait à se démocratiser, mais restait difficile à aborder pour les familles défavorisées.¹⁰

Les jeunes qui ont participé aux recherches venaient de ces familles défavorisées qui étaient souvent issues d'immigration, et avant la période de recherche, la plupart n'avaient pas d'expérience avec l'informatique, mais The Clubhouse leur a permis d'accéder à l'informatique, en leur proposant une grande sélection de logiciels et sans imposer une structure rigide d'activités.¹¹

Il est aussi important de noter que les animateurs qui travaillent au centre périscolaire étaient principalement issus des études d'arts et n'étaient donc pas du tout des programmeurs, mais ils ont pu accompagner les jeunes à travers la création de divers projets Scratch en se reposant sur leur compétences d'art et leur créativité.¹²

Les chercheurs ont pu voir que par moment, c'était les jeunes qui apprenaient Scratch aux animateurs, ce qui change la dynamique habituelle de l'animateur qui accompagne un jeune dans une activité¹³.

8 *Ibid.*

9 *Ibid.*

10 *Ibid.*

11 *Ibid.*

12 *Ibid.*

13 *Ibid.*

En lien avec mes recherches

Programming by choice m'a beaucoup aidé pour mes recherches sur l'apprentissage de la programmation par la méthodologie employée par les chercheurs et les découvertes qu'ils ont faites en étudiant les jeunes qui utilisent Scratch.

Je trouve leur méthodologie très pertinente, car elle laisse les jeunes faire ce qu'ils veulent avec Scratch sans imposer des commandes à utiliser impérativement. Ils laissent les jeunes apprendre par eux-mêmes et ils encouragent la collaboration entre jeunes, ainsi que la collaboration entre animateurs et jeunes. Les chercheurs au début avaient une posture d'observateur et de collecteurs d'artefacts numériques.

Et à la fin, pour interroger les jeunes, le fait de lier Scratch avec des concepts qui leurs sont déjà familiers est très intéressant, car cela leur permet d'avoir un point de référence et de mieux structurer leur propos.

Les jeunes pendant les entretiens ne sont pas laissés devant une feuille blanche et les chercheurs leur donnent des pistes adaptées à ce qu'ils connaissent déjà.

Le fait de demander aux jeunes de trouver un objet dans le monde physique qui rappelle leur expérience de Scratch est pertinent pour avoir une idée de comment ils ressentent leur rapport au logiciel.

La posture des animateurs du Clubhouse, même si elle n'est pas souvent mentionnée dans l'article, est fascinante, car ils apprennent avec les jeunes et collaborent.

Les animateurs se détachent de leur posture habituelle de guide et ils deviennent des apprenants en même temps que les jeunes.

INVENT TO LEARN

Making, Tinkering, and
Engineering in the Classroom

Sylvia Libow Martinez
Gary Stager, Ph.D.

Invent to Learn

Sylvia Libow Martinez et Gary Stager
2013 - Constructing Modern Knowledge Press

À propos des auteurs

Sylvia Libow Martinez était ingénieure en aéronautique avant de devenir productrice et vice-présidente de plusieurs éditeurs dans le monde des jeux vidéo, où elle a pu rencontrer des personnes qui étaient exclues des mathématiques “traditionnelles” qu’on apprend à l’école et qui l’ont amenée à concevoir des expériences d’apprentissages ouverts à tous sans forcément passer par les cours magistraux.¹

Gary Stager est un professeur, éducateur, développeur, etc. En 1990 il a mené des recherches sur des plateformes d’apprentissage en ligne et a travaillé avec Seymour Papert pour documenter ses projets de recherches d’apprentissage du numérique.²

Tous les deux ont formé la Constructing Modern Knowledge Press et ont collaboré pour écrire Invent to Learn.³

Invent to Learn critique le mode d’éducation traditionnelle et propose une solution alternative inspirée de Jean Piaget et de Seymour Papert, qui est le constructivisme. La livre propose aussi beaucoup de ressources afin d’alimenter la recherche pour propulser l’apprentissage de l’informatique et de la fabrication.

1 « Meet the Influencer: Sylvia Martinez », dans *Literacy Beat*, 15 mai 2016, <https://literacybeat.com/2016/05/15/meet-the-influencer-sylvia-martinez/> (Page consultée le 2 décembre 2024).

2 « About Gary », dans Gary Stager, <http://professorgarystager.com/about-gary> (Page consultée le 4 mars 2025).

3 « About the authors », dans *Invent To Learn*, <https://inventtolearn.com/about-the-authors/> (Page consultée le 4 mars 2025).

Le constructivisme et la critique de l'école actuelle

Dès le 1er chapitre l'accent est mis sur la théorie d'apprentissage nommée le constructivisme. Cette théorie formalisée par Jean Piaget explique que l'apprenant construit son savoir par ses propres expériences, et non par quelqu'un d'autre qui lui transmet l'information sans lui donner l'opportunité de bien le saisir et de faire sens avec.

Invent to Learn reflète bien la théorie de Piaget, car des élèves qui montrent des difficultés avec des mathématiques "traditionnelles" ont une toute nouvelle approche quand le problème vient d'une situation concrète et est en lien avec d'autres centres d'intérêt.¹ Les professeurs ne doivent donc pas donner des leçons didactiques, mais plutôt les ressources et l'environnement pour comprendre le sujet qu'ils enseignent.²

Aujourd'hui, l'ordinateur, les microcontrôleurs, les programmes et l'Internet rendent ce type d'apprentissage plus facile, et même, selon Seymour Papert³, l'ordinateur c'est le Protée de Machines, car comme la divinité marine dans la mythologie grecque qui peut se métamorphoser⁴, l'ordinateur est universel et peut changer sa forme pour servir des milliers de fonctions, et plaire à des milliers des goûts.⁵

1 Sylvia LIBOW MARTINEZ et Gary STAGER, *Invent to learn: making, tinkering, and engineering in the classroom*, Torrance, Calif, Constructing Modern Knowledge Press, 2013, p. 13.

2 *Ibid.*, p. 14.

3 Seymour Papert est un chercheur, un mathématicien et un activiste pour le numérique.

4 « Protée (mythologie) », dans Wikipédia, [s.l.], 2024, [https://fr.wikipedia.org/w/index.php?title=Prot%C3%A9e_\(mythologie\)&oldid=221634821](https://fr.wikipedia.org/w/index.php?title=Prot%C3%A9e_(mythologie)&oldid=221634821) (Page consultée le 10 mars 2025).

5 Sylvia LIBOW MARTINEZ et Gary STAGER, *Invent to learn*, op. cit., p. 19.

Mais malheureusement, l'éducation traditionnelle utilise l'ordinateur juste comme un nouveau gadget à faire apprendre des vieux concepts abstraits comme trouver les racines des nombres impaires et d'autres "problèmes mathématiques"¹. Pour les écoles traditionnelles occidentales, 2 manières de voir l'ordinateur dominant. Dans la première, l'ordinateur est vu comme un outil permettant à l'élève de faire ses devoirs académiques plus efficacement. Et dans la deuxième, il est vu comme un tuteur, en montrant des consignes et en évaluant l'élève.² À l'école, c'est les manuels et les chapitres qui mettent en motion l'apprentissage d'une manière structurée³, avec les enseignants qui suivent attentivement ce que le cursus dicte. Le jeu dans l'école est seulement dans le cours de récré, et au moment de la sonnerie, il faut redevenir sérieux.⁴

Le *tinkering*, que l'on peut traduire par le bidouillage en français, est une pratique guidée par la curiosité et l'imagination. Quand on bidouille il n'y a pas d'instructions, il n'y a pas de mauvais choix, ni d'échecs. L'importance est mise sur le fait de comprendre comment les choses fonctionnent.⁵ Le *tinkering* est une activité unique à l'humain, elle combine des forces sociales et créatives qui incluent l'apprentissage et le jeu.⁶

Comme le dit Jean Piaget, "*Le jeu est le travail des enfants*"⁷, mais le jeu n'est pas une corvée ni l'opposé du travail, en revanche, c'est de l'apprentissage et nous permet de recréer nous et notre monde.⁸ Dans Invent to Learn, le jeu est déterminant pour l'éducation de l'enfant. Le jeu et le design se ressemblent dans le fait qu'ils se détachent des manières de pensées habituelles, et permettent d'accomplir beaucoup plus.⁹

1 *Ibid.*

2 *Ibid.*, p. 34.

3 *Ibid.*, p. 36.

4 *Ibid.*

5 *Ibid.*, p. 38.

6 *Ibid.*, p. 36.

7 *Ibid.*, p. 38.

8 *Ibid.*

9 *Ibid.*

Apprendre à penser par le design et les projets

Pour se distancier des structures rigides habituelles comme des manuels et des plans de leçons prédéfinies, et de détacher l'enseignant de son rôle de manager et d'administrateur du savoir¹, Invent to Learn propose de faire des projets en classe, et d'apprendre à penser par le design.² C'est-à-dire, qu'il faut engager les élèves dans des pratiques authentiques de design et leur donner des projets concrets avec du matériel adapté.³ Dans la salle de classe, la méthode de design itérative est pertinente, car elle permet d'expérimenter plusieurs fois et de tester plusieurs possibilités pour résoudre un problème concret. ⁴ Un modèle proposé par le livre est la méthode TMI, qui peut dire "trop d'informations" (Too Much Information), ainsi que : trop d'instructions, trop d'interruptions, trop d'interventions, parce que les méthodes de design actuelles ne sont pas assez adaptées pour une salle de classe, car elles sont souvent trop complexes et justement, elles contiennent trop d'interruptions, trop d'interventions, et trop d'instructions de la part du professeur.

L'acronyme TMI de cette nouvelle méthode de design itérative en revanche permet de souligner 3 verbes principaux et donc de créer une méthodologie de design simplifiée: Think (Penser), Make (Faire), Improve (Améliorer).⁵

Think signifie penser en français, et cela inclut le brainstorming, se fixer des objectifs, et le fait de prendre des décisions pour son projet.

Make, ou faire en français, englobe la fabrication, le jeu, la programmation... En gros c'est à cet endroit que l'on trouve le plus d'action

Improve, ou améliorer en français est une étape obligatoire qui se produit quand l'élève se rend compte que son projet ne s'est pas passé comme il a voulu et/ou qu'il n'est pas satisfait de son travail. À ce moment l'élève va discuter avec les autres, il va rechercher de nouvelles idées, il va expérimenter de nouveaux matériaux... etc.⁶

1 *Ibid.*, p. 36.

2 *Ibid.*, p. 45.

3 *Ibid.*

4 *Ibid.*, p. 50.

5 *Ibid.*, p. 52.

6 *Ibid.*, p. 53-54.

Pour revenir sur la place du professeur, ici, il ne va pas donner des cours magistraux où il va parler pendant 45 minutes, mais il va pouvoir incarner plusieurs rôles dans une salle de classe constructionniste.¹ Par exemple, il peut devenir ethnographe et il va découvrir ce que ses élèves savent déjà. Il peut aussi être documentaliste, et récolter des preuves qui rendent les pensées invisibles de ses élèves visibles. Il peut aussi incarner un manager de studio et rendre des outils et des ressources disponibles à la classe pour permettre que les idées des élèves deviennent réalité. Et finalement, il peut être un "guide sage", en guidant ses élèves à chercher loin dans leurs grandes idées sans les forcer.²

Ici, le professeur idéal ne donne pas des cours magistraux, mais s'intéresse à ses élèves et à leurs compétences.³ Il limitera les instructions trop longues afin de laisser l'opportunité à ses élèves d'explorer⁴, et il va encourager ses élèves à collaborer et penser ensemble.⁵

1 *Ibid.*, p. 75.

2 *Ibid.*, p. 76.

3 *Ibid.*, p. 75.

4 *Ibid.*, p. 77.

5 *Ibid.*, p. 78.

Comment cet ouvrage m'a aidé dans mes recherches et a débloqué mes préjugés?

Le livre *Invent to Learn* a été crucial dans mes recherches, car il m'a permis de penser en dehors des simples outils techniques pour enseigner la robotique et la programmation. *Invent to Learn* m'a permis de découvrir la méthodologie d'enseignement constructionniste de Seymour Papert et m'a donné des pistes de recherches très intéressantes par rapport à la pédagogie alternative pour enseigner l'informatique en dehors des salles de classe traditionnelles. Je me suis beaucoup plus penché sur la partie méthodologie que sur la partie technique et matérielle du livre, qui regroupe le choix des outils et des matériaux pour créer une salle de classe constructionniste¹, et comment aménager sa salle afin d'inviter la créativité et le bricolage, sans imposer des stéréotypes masculins.² Dans le livre, les auteurs insistent beaucoup sur l'importance d'une informatique tangible, qui permet qui permet aux élèves de créer leurs propres machines qui interagissent avec l'environnement³, et donc de se libérer des habitudes informatiques qu'on a à l'école. Par exemple, utiliser Word pour taper de longs essais à propos d'un sujet imposé par un professeur, utiliser Excel pour calculer des statistiques, etc. Pour faire bref ce livre m'a aidé à découvrir de nouvelles méthodologies pour enseigner et m'a permis de m'ouvrir à de nouveaux moyens de voir l'informatique et de percevoir le makerspace, qui avant mes recherches, se limitaient aux communautés geeks savantes.

1 *Ibid.*, p. 88.

2 *Ibid.*, p. 161.

3 *Ibid.*, p. 91.

ENTRETIEN SOCIO.

Contexte

Lors du mois d'octobre, j'étais accueilli par la CabAnne des Créateurs à Schiltigheim pour effectuer un stage en fablab chez eux.

Pendant mon stage, j'ai pris l'opportunité d'entretenir avec Yann Misse, qui est un animateur-facilitateur à la CabAnne des créateurs ainsi que fabmanager. L'entretien, même s'il était assez court (environ 26 minutes), a été très intéressant car ça m'a permis de mieux connaître le parcours, ainsi que le travail de Yann et aussi du public qu'il travaille avec.

L'entretien s'est passé très calmement, même s'il y a eu pas mal d'hésitations parfois car certaines questions étaient vastes et nécessitent un peu plus de réflexion.

Mon rôle pendant l'entretien était d'interroger et d'écouter ce que Yann me disait, ainsi que de le relancer et de le diriger vers mon sujet assez subtilement.

Il faut aussi souligner qu'il y a eu pas mal de mouvement dans le Fablab pendant ce temps et nous étions assez distraits parfois, que ce soit par l'imprimante 3D qui travaillait dans l'arrière-plan, ou par des personnes qui passaient.

Mais cette distraction a permis d'alléger la lourdeur de l'entretien et de le rendre plus décontracté.

Transcription de l'entretien sociologique

Moi

Et voilà, présente-toi un peu.

Yann

Huh?

Moi

Présente-toi un peu. T'es qui ? Tu as quel âge ?

Yann

OK. Euh moi je m'appelle Yan Misse, j'ai 26 ans, je suis animateur facilitateur à la Cabane des créateurs. Euh voilà. Est-ce que tu veux mon parcours ?

Moi

Oui, ça serait très intéressant justement.

Yann

Moi, j'ai un DUT en génie électrique et informatique industrielle. Euh j'ai euh... j'ai fait une année de licence en énergie électrique électronique automatique et une année d'école d'ingénieur en informatique et réseau de télécommunication en alternance.

Moi

Hm. Et justement, est-ce que ce parcours t'aide justement là où tu es maintenant ?

Yann

Oui. Clairement parce que bah du coup j'ai fait beaucoup de enfin j'ai fait beaucoup de programmation d'électronique. J'ai toujours bricolé...

Interruption par une dame

Moi

Continuons?

Yann

hésitation Euh... Donc oui, mes études elles m'ont appris à bricoler donc surtout dans le domaine de l'électronique, de la robotique, de la programmation de manière générale.

Et après à côté de ça, ça m'a aussi donné une curiosité avec tout ce qui est en lien avec les nouvelles technologies.

Les outils, les outils à commande numérique euh et aussi sur du mécanique, modélisation 3D, tout ça pour pouvoir euh pour pouvoir aboutir à des projets concrets en ayant un peu les bases dans tous les domaines.

Moi

Hmm.

Yann

Mais surtout du coup en électronique et en informatique.

Moi

Très bien.

Yann

Voilà.

Moi

Un peu une question un peu plus personnelle. Est-ce que tu étais toujours aussi curieux en tant qu'enfant ? Est-ce que tu as bricolé en tant qu'enfant ?

Yann

Euh la période je pense où j'ai j'étais le plus curieux, c'était quand je suis rentré au lycée et que j'ai eu mon premier ordinateur.

Moi

C'était quoi ton premier ordinateur ?

Yann

C'était un Asus.

Moi

Quel était le système d'exploitation ?

Yann

Windows. Asus, Windows à l'époque.

Moi

XP?

Yann

Il rigole Non non non, quand même! Je ne suis pas si vieux que ça.

Moi

Comment ça tu es pas aussi vieux que ça !? Moi aussi j'avais XP quand j'étais jeune.

Yann

Ouais mais le premier ordi sur lequel j'ai... Enfin le premier ordi auquel j'ai touché, bien sûr qu'il était sur Windows XP mais mon **premier ordinateur personnel à moi**, il devait être sous Windows 10, je pense.

Moi

10 ?

Yann

Ouais, je pense. Ouais. Ou 8 peut-être. 8

Moi

7 peut-être?

Yann

Non, je pense c'était Windows 8.

Moi

OK. Ah, effectivement

Yann

Et du coup, c'est à ce moment-là que j'ai été le plus curieux de l'informatique, de par le biais du jeu vidéo aussi, de la programmation, de l'électronique, de tout ça. Et après, bah ça c'est ça a toujours continué. Je dirais que je bricole peut-être moins qu'avant à tester des nouveaux trucs. Et encore non, je suis toujours curieux.

Moi

Et pourquoi tu bricoles pas autant ?

Yann

Parce que j'ai déjà mon travail qui est de bricoler beaucoup, et du coup en perso j'ai moins la motivation de me lancer dans des projets euh ... Parce que du coup j'en ai fait un peu mon métier quoi.

Moi

Ah...

Yann

...mais je m'éclate toujours autant à faire des projets dans mon travail mais aussi des fois ça il m'arrive d'en faire en perso peut-être moins qu'avant.

Moi

Hmm. Et vu que tu bricoles, tu es bien au courant de mouvement open source?

Yann

Oui, bien sûr.

Moi

C'est quoi ton avis ? C'est qu'est-ce que tu en penses ? Tu penses que c'est quelque chose d'utopique ? C'est quelque chose de fiable ? Enfin, bref.

Yann

Bah utopique non parce que il y a déjà beaucoup de choses open source qui existent et qui fonctionnent.

Bah la preuve en est le logiciel Blender qui est un logiciel open source et qui fonctionne très très bien et qui est même très performant et euh pour lequel on va trouver beaucoup de ressources.

En fait, l'open, j'aime bien parce que du coup voilà, il y a ce côté communautaire et où on va trouver beaucoup de ressources mais parfois l'open source donne un peu ce sentiment de pas abouti complètement.

Moi

Hm.

Yann

Ce qui est normal parce que c'est des choses qui sont vachement évolutives de par la communauté.

Moi

Mhm.

Yann

Mais euh... mais la dynamique est bonne. Moi dans la vie de tous les jours euh je ne suis pas à 100 % open-source même pas du tout. Enfin, il y a certains logiciels que j'utilise qui sont open source, notamment le Slicer pour l'impression 3D qui est open source, certains logiciels de modélisation comme Blender.

Moi

Hm.

Yann

Mais à côté de ça, la plupart du temps, j'utilise quand même des logiciels qui sont privés.

Moi

Et qu'est-ce que t'empêche vraiment d'être d'aller complètement open source ?

Yann

Euh je dirais le la manque de synergie avec les environnements de travail,

Moi

C'est-à-dire?

Yann

La complexité des fois d'utilisation d'open source pourlier entre certains logiciels ou... en fait je ne sais même pas trop euh... ouais des fois la complexité d'utilisation... je trouve qu'en open source des fois c'est plus complexe que certains logiciels qui ont des interfaces utilisateur... je dirais que c'est globalement c'est ça. C'est l'UX et l'interface utilisateur.

Moi

Ouais c'est si on compare Inkscape à Illustrator, ce n'est pas la même chose.

Yann

Oui, voilà, c'est ça. Ils font le même café, on va dire. Enfin, le job est le même, mais euh en terme d'utilisation, c'est des fois moins intuitif. Donc, je dirais que l'interface utilisateur qui, des fois, est moins accueillante que sur certains logiciels.

Moi

Hm.

Yann

Je trouve que globalement, même quand tu compares un Linux à du Windows, j'ai plus de facilité. Après, je pense c'est une question d'habitude aussi hein, mais des fois, j'ai parfois plus de facilité d'utilisation d'un Windows qu'un Linux. Même je trouve que des fois, c'est plus fluide aussi en terme d'utilisation.

Moi

Ouais.

Yann

Voilà.

Moi

Et...

Yann

C'est vraiment de l'interface.

Moi

Ouais. Encore une autre des questions un peu sur en tout qui est un peu liée avec l'open c'est un peu tout ce qui est idée des communs.

Yann

Ouais.

Moi

hésitation Les communs, les... tout ce qui est communs

Yann

Les licences?

Moi

Bah en soit plus les... comment s'appelle déjà ?

Yann

Euh qu'est-ce ça dire ? Les comuns...?

Moi

Les communs... comment dire... un peu l'intelligence collective.

Yann

L'intelligence collective?

Moi

Un peu comme Wikipédia... Tout ce qui est ressources en ligne comme Thingiverse.

Yann

Ouais carrément. Oui ben, moi je trouve que c'est justement... je trouve que c'est très très chouette. C'est arriver à créer des communs de par une communauté qui s'investit et euh... et du coup bah... Si le mouvement est trop chouette de de rendre accessible des...

Par exemple, moi j'ai un exemple. C'est de la réparation de pièces en impression 3D bah si tu as un fichier euh qui a déjà été conçu pour une marque donnée pour un appareil donné, ben... De le mettre à disposition d'autres gens pour préparer...

Enfin il y a une dimension un peu écologique là-dedans.

Moi

Hm.

Yann

Et euh... du coup je trouve que c'est intéressant.

Moi

Est-ce que toi tu contribues à des projets?

Yann

J'aimerais bien mais je ne le fais pas assez... et de part mon travail, il faudrait que je crée des communs et de la documentation sur les projets que je fais mais étant donné que je suis tout seul sur certains projets, c'est une charge de travail en plus.

Moi

Ouais.

Yann

Que des fois je prends pas le temps de... je prends pas le temps de faire.

Mais dans un idéal oui, j'aimerais beaucoup comme le projet bah des dessins de l'animation de dessin genre...

D'ailleurs l'algorithme que j'utilise est open source mais si j'arrive à créer un projet autour de ça, j'aimerais bien le rendre open source lui aussi, accessible et documenté.

Moi

Ouais. Mais quand on parle d'accessibilité, qu'est-ce qu'on entend par accessible justement ?

Yann

Bah qu'il y a des bases de données de... bah que ce soit pour des algorithmes comme GitHub où tu peux aller récupérer des algorithmes de programmation, ou alors ben comme Thingiverse quand je modélise des pièces, bah prendre le temps de les mettre sur Thingiverse et de les référencer et voilà.

Pour moi rendre accessible c'est le mettre en ligne sur des plateformes.

Moi

Hm.

Yann

Après, ça peut être aussi partager euh,,, partager euh... de en physique.

Moi

Hm.

Yann

Par exemple, enfin de partager de la documentation physique à quelqu'un ou de lui donner des fichiers physiques, enfin des fichiers de lui transférer les fichiers directement.

Moi

Ouais.

Yann

Sans passer par les plateformes.

Moi

OK. Et justement, je vais m'orienter un peu plus sur tout ce qui est animation-méditation.

Yann

OK.

Moi

Est-ce que t'avais déjà travaillé avec des enfants ?

Yann

Oui, beaucoup même.

Moi

Tu peux m'expliquer un peu plus ?

Yann

Euh bah du coup à la Cabanne, on fait de l'animation et donc j'ai eu à plusieurs moments euh.. enfin on fait globalement, on fait beaucoup d'animations sur les périodes quand on s'approche un peu de l'hiver et surtout en été.

Et... du coup j'ai déjà travaillé avec les enfants sur plein de thématiques différentes.

Ça va de la robotique à la peinture à la photo avec le soleil à de l'animation sportive.

Euh... de l'animation créative, culturelle, voilà, globalement. Mais moi, j'aime bien montrer le fonctionnement des machines en animation.

Expliquer, essayer de vulgariser un peu comment ça fonctionne pour que les enfants possèdent des bases.

Après j'aime bien aussi voir les enfants s'exprimer créativement parlant, que ce soit sur de la peinture, sur de même au niveau sportif. Je suis un peu allé loin dans la question mais...

Moi

Non, c'est parfait parce que justement j'allais aussi justement vous demander comment vous percevez l'intelligence d'un enfant ? Est-ce que vous pensez qu'il sait faire des choses dont on ne se rend pas compte ?

Yann

Euh bah je dirais que ça dépend des enfants. Mais euh je pense que bah les enfants, ils ont pas tous la même curiosité et le même intérêt pour les mêmes animations.

Moi

Hm.

Yann

On le voit, il y a des enfants qui vont être beaucoup plus réceptifs sur des animations sportives, d'autres beaucoup plus sur des animations créatives, d'autres pour beaucoup plus sur des animations en lien avec le numérique.

Et après en terme d'intelligence, ben justement je trouve ces intelligences, elles se retrouvent dans les différents domaines d'animation.

C'est que, un enfant qui aime bien le sport, bah il va avoir cette intelligence un peu sportive, du collectif et de l'utilisation de son corps. Mais à côté de ça, ben un enfant qui aime bien l'animation créative, bah lui, il va avoir plus une intelligence manuelle.

Donc là, on est encore sur l'utilisation du corps, mais aussi sur l'utilisation de l'esprit, de la créativité. Euh voilà, ça dépend.

Pareil, l'intelligence numérique aussi, je trouve c'est on va être plus en lien avec de l'intelligence liée à la logique, aux mathématiques.

Moi

Hm.

Yann

Et après, il y a des animations qu'on fait qui mêlent un peu tout ça. Quand on fait des animations Ozobot, bah il y a un côté un peu créatif mêlé à l'intelligence numérique et après... bah pas sportif pour le coup mais voilà

Moi

Hm.

Yann

Enfin voilà, je trouve que chaque profil d'enfant a son propre intelligence un peu. Enfin moi j'aime pas trop ce truc de mettre dans une case "ah cet enfant...", par exemple, "il est autiste et il fonctionne comme ça" enfin.

Moi

Ouais.

Yann

Tout le monde a sa complexité à lui.

Moi

Hm.

Yann

Même si bien sûr il y a il y a des pathologies qui existent qui sont à prendre en compte.

Moi

Ouais. Et justement avec ces différentes formes d'intelligence, est-ce que vous pensez que le milieu de l'enfant peut jouer un rôle dessus peut-être ?

Yann

Ah oui, carrément. Oui.

Moi

Comment ça ?

Yann

Bah oui, l'éducation, les activités qu'un enfant va pouvoir faire étant jeune, qui vont lui être présentés ou enfin.. son accès à certaines activités, l'utilisation d'un ordinateur, l'utilisation des écrans globalement et les permissions qu'on va lui donner sur ces utilisations là.

Moi

Hm.

Yann

Je pense que ça joue, ça a un grand impact. Donc oui, le milieu globalement a un impact sur le cerveau de l'enfant et sur comment il va être curieux et sa manière de réfléchir et d'agir avec son environnement.

Moi

Hm.

Yann

Et les gens qui l'entourent aussi.

Moi

Ouais. Et justement pour basculer sur tout ce qui est euh... robotique, bricolage, un peu le côté un peu plus numérique.

Les enfants qui comme tu dis ont cette intelligence numérique, est-ce qu'il y a un stéréotype qui se dégage ? Est-ce que c'est un petit enfant issu d'une famille bourgeoise ou...

Yann

Choqué Waouh ! Alors euhh..

Moi

Pardon, c'est mal dit comme ça, mais...

Yann

Non mais, je comprends la question.

Nous on travaille souvent dans les quartiers prioritaires. Donc oui, on ressent avec les enfants des quartiers que ben parfois... euh... Tu vas avoir un peu un manque de compréhension face au numérique.

Et à contrario, ça m'arrive aussi des fois de faire des animations à la CabAnne avec des enfants qui pour le coup eux ont accès à un ordinateur, viennent sûrement de famille un peu plus aisée et qui ont cette intelligence numérique qui est peut-être un peu plus développée, ouais.

Moi

Hm. Et justement pour les enfants qui n'ont pas forcément cette intelligence numérique, est-ce que justement la robotique ça aide à mieux comprendre certaines notions, ou juste avoir une sorte de truc physique qui leur montre les conséquences de leurs actions en programmation ?

Yann

Euh je pense oui, carrément. Oui. Oui. Oui, c'est sûr même. Euh je pense que oui, après ça dépend avec quel robot tu agis, quelle complexité de robot aussi tu proposes en animation avec ces enfants-là. Mais euh mais oui, globalement, je pense que ça permet de bien comprendre la logique numérique, la robotique, même si parfois je trouve que c'est quand même des fois un peu complexe.

Moi

Comment ça ?

Yann

Ben, déjà, il y a des par rapport au langage, la programmation reste souvent du langage, c'est souvent en anglais par exemple.

Donc des fois en termes d'accessibilité, un enfant qui a du mal déjà parfois à s'exprimer en français, ça va être pareil en anglais.

Du coup, il va peut-être avoir du mal à juste comprendre le langage. En fait, ça dépend des interfaces de programmation, du langage de programmation qui est utilisé et oui, du robot qui est mis en œuvre derrière que tu programmes.

Moi

Hm

Yann

Mais oui, globalement, je pense que... En fait du moins moi je pense que ça permet de susciter un intérêt chez les enfants d'utiliser la robotique.

En fait, tu vas susciter un intérêt pour tout ce qui est logique, parfois mathématique.

De part cet objet ludique sur lequel, comme tu disais, voir tes actions, on va parler de programmation en physique. Je ne sais pas si j'ai été clair.

Moi

Oui.

Yann

Ouais.

Moi

Ouais. C'était clair. Et encore sur l'enfant et la robotique?

Yann

Hm?

Moi

Dans tous les robots de la cabane, c'est lesquels qui ont marché le mieux et pourquoi?

Yann

Euh bah souvent bah après, c'est ceux qu'on utilise le plus parce que il est le plus adapté à en terme de tranche d'âge, mais c'est l'ozobot qu'on utilise souvent en animation...

Bon souvent c'est un grand mot mais c'est celui qui revient régulièrement quand même en animation bah parce que c'est celui qui est le... on va dire qui permet de mettre un pied dans la programmation sans non plus avoir besoin d'un ordinateur parce que tu vas juste utiliser des codes couleurs pour opérer des actions et c'est celui qui s'adapte du coup à des publics très variés en termes de tranche d'âge.

Donc pouvant aller de... je ne sais pas, 6-7 ans jusqu'à 12- 14.

Puis c'est un robot qui est évolutif, c'est que il y a la partie programmation avec du suivi de ligne et des codes couleur mais tu peux aussi le prendre en main avec un ordinateur. Du coup bah passer sur de la programmation plus complexe.

Moi

Hm.

Yann

Voilà.

Moi

Est-ce que vous avez déjà utilisé la programmation complexe avec les ozobots en atelier? qu'est-ce que ça donner ?

Yann

Non, moi pour l'instant je n'ai pas eu le temps de me pencher dessus et je l'ai jamais utilisé en atelier.

Le seul truc un peu évolué qu'on a pu faire, c'est les feuilles pédagogiques de Scratch qu'on a déjà pu tester avec des enfants.

Moi

Ouais, justement. Moi je les ai testé ce matin justement pour voir qu'est-ce que ça donne et je trouve c'est assez bien vu que c'est étape par étape et c'est des grandes feuilles.

Yann

Elles sont super hein. Elles sont très belles, très cool ces feuilles.

Moi

Et justement parlant de Scratch, quel est votre avis dessus, en général ?

Yann

Euh.

Moi

Parce que souvent, quand on dit Scratch au collègue...

Yann

Les enfants n'aiment pas trop ça.

Moi

Ils n'aiment pas. Ouais,

Yann

Bah justement, je pense que cet avis, il démontre bien que ça permet de mettre un pied dedans mais on reste quand même ça reste quand même un peu complexe.

Moi

Hm

Yann

Il faut quand même bien vulgariser les bases de la programmation, même si tu utilises Scratch parce que ça reste au final de la programmation mais avec des blocs.

Donc on va dire que en termes d'interface, c'est un peu plus attractif mais ça reste de la programmation.

Donc il y a quand même la complexité liée à la logique de programmation avec les boucles, les "si alors ça", Voilà.

Moi

Hm.

Yann

Donc je ne sais pas, je trouve que l'interface c'est quand même c'est une jolie interface qui est quand même assez intuitive qui donne envie mais ça reste encore complexe si tu n'as pas les bases du tout de programmation.

Moi

Hm.

Yann

Bon après ça ouais je ne sais pas. Je me suis un peu perdu. Je pense que c'est un bon outil quand même. C'est un bon outil pour mettre les pieds dans la programmation mais ça parlera peut-être pas à tous les enfants.

Moi

Pourquoi ça ne parlait pas à tous les enfants ?

Yann

Ben parce qu'on a revient à ce truc de d'intelligence un peu numérique et il y a certains enfants qui ont pas cette intelligence là, sans parler d'intelligence, mais de curiosité aussi numérique.

Moi

Hm.

Yann

Curiosité de programmation, Ça parle pas à tout le monde.

Moi

Ah, ouais.

Yann

Mais ça reste un bon outil Scratch pour pour débiter.

Moi

Hm.

Yann

Enfin, sans parler de scratch, juste la programmation en bloc en tant que tel, quoi.

Moi

Ouais. Vu que c'est assez simple à comprendre si tu es déjà curieux.

Yann

Oui. Oui. Voilà, c'est ça.

Moi

Mais alors pour les gens qui sont pas déjà curieux, on fait comment pour enseigner le numérique ?

Yann

C'est une bonne question. Je ne sais pas si j'ai la réponse. Euh...

Pause longue

Moi

Parce qu'il y a déjà les Ozobot qui peuvent susciter un peu.

Yann

Oui, voilà. Je pense qu'on peut faire du de la robotique sans forcément programmer et bah le Ozobot, il le montre parce que bah on va pouvoir faire effectuer un comportement au robot sans passer par de la programmation.

Donc enfin entre guillemets parce qu'on va utiliser des codes couleur pour pour lui faire exécuter des actions.

Moi

Hm.

Yann

Mais oui. Je pense que c'est un bon exemple. On peut susciter de l'intérêt pour le numérique et la programmation mais sans utiliser de code quoi.

Je pense que c'est peut-être comme ça déjà dans un premier temps qu'on peut insuffler cette curiosité chez l'enfant.

Sinon c'est pas forcément facile.

Je trouve que ce n'est pas pour l'avoir utilisé.

Même Scratch en quartier prioritaire, l'enfant peut vite décrocher et justement ça lui arrive parce que nous on est quand même censé faire des animations pas forcément visé éducative même si des fois c'est bien mais c'est on est plus là pour divertir.

Moi

Hm.

Yann

Enfin ouais divertir et en même temps si on peut attiser de la curiosité pour le numérique c'est bien.

Mais du coup par exemple, j'ai déjà utilisé Scratch et je sais que ça ne suscitait pas un intérêt énorme mais comme tu disais, il y a cette réflexion qui revient de "Ah bah c'est ce qu'on fait à l'école".

Moi

Ouais.

Yann

Donc voilà, il cette je pense il y a cette notion de d'école et de scolaire et de qui ne va peut-être pas susciter de l'intérêt chez tous les enfants.

Moi

Hm.

Yann

Parce que pas tous les enfants sont scolaires.

Moi

Ouais.Justement parlant de scolaire, l'éducation nationale.

Yann

Ouais.

Moi

Vous en pensez quoi ? Même si vous n'étiez pas forcément instituteur, vous étiez quand même dedans.

Yann

Euh, moi j'ai un avis très mitigé. Je pense qu'il y a du bon en termes de programme.

Enfin ouais, le programme éducatif, on va dire français que ce soit dans les études supérieures ou même l'école primaire, le collège lycée, je pense que le programme n'est pas mauvais.

Mais dans l'exécution de ce programme là, je pense qu'il y a **pas assez de moyens qui sont mis en place** et pas assez d'adaptation en fonction de ce qu'on disait un peu tout à l'heure de différentes intelligences.

Moi

Hm.

Yann

Et pas assez de temps en fait, pas assez de temps pour s'occuper de tout le monde pour prendre en compte justement cette différence de milieu social.

Moi

Hm.

Yann

Ils n'ont pas assez de moyens.

Donc peut-être en passant par de la robotique pour faire des mathématiques, passant par... je ne sais pas, par des animations culturelles pour tout ce qui est histoire géo.

Enfin voilà, peut-être rendre plus vivant un peu l'école.

Moi

Hm.

Yann

Mais actuellement il n'y a pas de moyen et beaucoup de professeurs sont débordés.

Moi

Ouais.

Yann

Peut-être que moi ce que j'ai pu ressentir à certains moments c'est le manque de passion de certains professeurs qui du coup ont du mal à transmettre parce que bah ils ne sont pas forcément passionnés par ce qu'ils font et du coup ils n'en ont pas l'envie.

Moi

Hm.

Yann

Mais ce manque de passion des fois il s'explique de par le manque de moyens parce que bah si on leur met pas les moyens à disposition bah la passion elle s'éteint vite. Ils ne peuvent pas se renouveler dans leur pratique.

Moi

Ouais

Yann

Voilà.

Moi

Merci beaucoup pour votre temps et pour l'entretien.

Yann

Pas de soucis.

Le parcours de Yann Misse, et comment ça lui a aidé en tant que animateur-facilitateur

Pour commencer l'entretien, j'ai posé des questions assez personnelles sur le parcours de Yann Misse, afin de mieux le connaître en tant que personne et mieux comprendre son passé.

Avant de devenir fabmanager et animateur-facilitateur à la CabAnne des créateurs, il a beaucoup étudié dans le domaine de l'informatique industrielle et de l'ingénierie civile, et a fait une alternance en réseau de télécommunication. Ses expériences lui ont permis d'apprendre à programmer et à bricoler dans la domaine de l'électronique et la robotique, et lui ont aussi donné une curiosité envers les nouvelles technologies.

Avec son expérience, il peut aboutir à des projets concrets en utilisant les bases qu'il a dans l'électronique et l'informatique.

Pour creuser un peu plus loin, je lui ai demandé s'il avait toujours cette curiosité envers le numérique en tant qu'enfant, et il affirme que c'est depuis qu'il a reçu son premier ordinateur personnel pendant le lycée qu'il est aussi curieux face au numérique, avec les jeux vidéos qui alimentait encore plus cette curiosité. Mais, depuis qu'il est fabmanager à la CabAnne des Créateurs, il n'a pas eu la motivation comme avant de se lancer dans des projets personnels car il bricole déjà beaucoup au travail.

Je trouve que le parcours de M. Misse est intéressant, même s'il reste assez classique. Quand je parle de parcours classique, je parle du fait qu'avec ses études en informatique et en ingénierie, il travaille maintenant dans un fablab, à transmettre le savoir qu'il a accumulé pendant des années aux autres.

Son parcours lui a aussi beaucoup nourri sa curiosité envers la programmation et l'électronique.

Open-Source et les Communs, une utopie ou non?

En connaissant son parcours qui est très centré sur l'informatique, j'ai voulu connaître son opinion sur le mouvement open-source et sur les communs, et voir s'il pense que c'est une utopie irréalisable selon lui, et si l'Open-Source et les communs peuvent avoir une place dans le monde réel.

Selon Yann Misse, l'open source n'est pas une utopie inatteignable, car il existe beaucoup de bons logiciels open-source, tels que Blender, et que l'esprit communautaire des logiciels open-source permettent d'avoir beaucoup de ressources pour pouvoir les utiliser.

Par contre, vu que l'open-source est évolutif, il va avoir ce sentiment d'être pas assez abouti parfois. Concernant ses usages, il n'utilise pas tout le temps des logiciels open-source, car parfois elles sont trop complexes à utiliser entre elles. Par exemple, avec Lightroom, on peut facilement passer à Photoshop pour modifier et ajuster les photos qu'on a importées et travaillées avec Lightroom.

On ne trouve pas cette facilité et cette synergie entre les logiciels open-source.

Un autre point a été fait sur la complexité de certaines interfaces utilisateurs, qui peuvent être assez compliquées pour des utilisateurs débutants.

Pour les communs, on a eu un peu de mal au début car on n'en parle pas aussi souvent que l'open-source, mais, après quelques clarifications et la mention des sites comme Wikipedia et Thingiverse, on a pu débloquer une conversation fascinante dessus.

Yann Misse, trouve que c'est très intéressant justement de pouvoir mettre à disposition des fichiers et de l'information aux autres usagers librement sur des plateformes dédiées comme Thingiverse.

Yann souligne un côté écologique qui est mis en avant avec la modélisation et la mise à disposition des pièces de rechange imprimables en 3D pour réparer certains appareils.

Dans son travail de fabmanager, il veut bien contribuer aux communs, mais étant donné qu'il travaille souvent tout seul, c'est une charge de travail en plus pour lui qu'il ne peut pas assumer.

Les différentes animations par la CabAnne, et les différentes intelligences chez les enfants.

Pour basculer sur un tout autre sujet, je voulais en savoir plus sur le travail qu'il fait avec les enfants qui visitent la CabAnne des créateurs et voir quels types d'activités ils font ensemble.

La plupart des animations se passent pendant les périodes de vacances été et hiver, et elles sont axés autour de différentes thématiques, que ce soit de la robotique, la peinture, ou l'animation sportive.

Même s'il aime bien vulgariser comment fonctionnent certaines machines en animation, il aime bien voir les enfants s'exprimer créativement.

Pour Yann, chaque enfant a une intelligence différente qui influence sa curiosité et l'intérêt qu'il porte pour certaines animations proposées par la CabAnne des créateurs.

Par exemple, il y a des enfants qui seront beaucoup plus réceptifs sur des animations sportives, d'autres sur des animations créatives, des enfants qui préféreront les animations en lien avec le numérique, etc.

Un enfant qui aime bien l'animation créative, aura une intelligence beaucoup plus manuelle qui met en mouvement son esprit, sa créativité, et ses mains.

Cependant, ils font aussi des animations qui mêlent les thématiques ensembles, comme les animations Ozobots, où ils mêlent le côté créatif avec le numérique.

Chaque profil d'enfant a sa propre intelligence, et ne doit pas être mis dans une case pré-défini.

Le milieu de l'enfant peut aussi jouer un très grand rôle sur comment il va être curieux et sur sa manière de réfléchir.

Pour creuser un peu plus, j'ai demandé s'il y a des stéréotypes qui peuvent se dégager par rapport à certaines intelligences, comme par exemple, si l'enfant a une intelligence numérique, est-il forcément issu d'une famille assez riche ou pas?

Vu que à la CabAnne des créateurs ils travaillent beaucoup avec des enfants issus des quartiers populaires, Yann a ressenti qu'ils ont un peu une manque de compréhension face au numérique en comparant avec des enfants qui ont un accès à un ordinateur chez eux et qui viennent d'une famille un peu plus aisée.

Pour essayer de débloquer ce manque de compréhension face au numérique, le choix du robot, ainsi que la manière dont on l'utilise peut jouer un grand rôle, même si elle reste compliquée.

Une raison de cette complexité est le rapport au langage, et le fait que les langages de programmation sont en anglais. Ce dernier point est important car si l'enfant a du mal à s'exprimer déjà en français, ça va être pareil en anglais, et donc il aura du mal à comprendre le langage de programmation.

Donc, les robots qui marchent le mieux dans la CabAnne pour susciter la curiosité chez les enfants débutants sont les Ozobots, car ils utilisent aucun langage de programmation, juste des lignes et des codes couleurs.

Par contre, quand Scratch est mentionné en animation, les enfants ont souvent la réflexion que c'est quelque chose de scolaire et ils ont tendance à vite décrocher. Le mot scolaire ici est utilisé comme un péjoratif, car on l'associe à quelque chose d'ennuyeux, et à quelque chose qu'on ne comprend pas.

Le manque de moyens dans l'éducation nationale.

Une explication de cette mauvaise réputation envers le scolaire est le fait que dans l'éducation nationale, ils veulent proposer des nouveaux programmes qui donnent envie à apprendre le numérique.

Mais, la plupart des écoles (que ce soit primaire, collège, lycée ou même études supérieures) dans le pays n'ont pas assez de moyens qui sont mis en place et ne sont pas adaptés aux différentes intelligences qu'un enfant peut avoir.

Il n'y a pas assez de temps non plus pour s'occuper de tout le monde et de prendre en compte les différences de milieux sociaux parmi les élèves.

Les professeurs sont débordés, et certains manquent de passion face aux sujets qu'ils enseignent. Ce manque de passion et ce débordement des professeurs fait que les enfants décrochent vite de ce qu'on essaie de leur faire apprendre, et les enseignants ne peuvent pas renouveler leur pratique car ils n'ont pas les moyens.

Il y a des idées pour essayer de rendre plus vivant l'enseignement, comme proposer la robotique pour apprendre les mathématiques ou des animations culturelles pour mieux apprendre l'histoire et la géographie, mais on est toujours heurté par le manque de moyens dans l'éducation nationale.

Conclusion

Ce entretien sociologique avec Yann Misse m'a beaucoup aidé dans mes recherches, car il m'a permis de connaître le parcours complexe d'un fabmanager et de animateur-facilitateur, ainsi de voir comment son parcours lui a permis d'avoir une aussi grande curiosité envers le numérique et d'aider les autres de mieux comprendre l'univers informatique.

Son point de vue sur l'open-source et les communs est aussi pertinent car il est très nuancé et permet de mieux comprendre les avantages et les désavantages du open-source pour les utilisateurs débutants à l'informatique, ainsi que ses propres difficultés à soutenir le mouvement de open-source.

Malgré son parcours d'études très axé autour de l'informatique et les facilités en informatique qui lui sont accordés par son expérience, je trouve que c'est intéressant qu'il anime quand même des activités sportives et créatives afin de répondre aux besoins de chaque enfant dans les quartiers populaires de Schiltigheim.

L'idée des différentes intelligences pour chaque enfant a été aussi importante dans mes recherches et m'a aidé à mieux voir le problème avec l'éducation tournée autour la programmation et la robotique.

On ne peut pas forcer chaque profile d'enfant dans une case et espérer qu'ils vont apprendre ce qu'on leur inculque.

ATELIER

OUTILLÉ

Atelier outillé

Chez la CabAnne des Créateurs à Schiltigheim

30/12/2024 - tout l'après-midi [13h à 18h]

4/01/2025 - tout l'après-midi [13h à 18h]

Lieu et contexte

J'ai eu l'opportunité de pouvoir mener mes recherches à La CabAnne des Créateurs à Schiltigheim pendant deux animations de quartier qui se sont déroulées pendant les vacances de Noël le 30 décembre et 4 janvier 2025.

Je faisais partie d'un des nombreux stands, qui étaient plus accentués sur les jeux vidéo et le coloriage.

Il avait un public varié, mais c'était souvent des mères qui venaient avec leurs enfants ou des jeunes qui venaient seuls avec leurs amis. Le 30 décembre, il n'y avait pas beaucoup de monde, car la communication n'était pas assez précise dans le quartier, malheureusement vu que je me suis prise un peu en retard, je n'ai pas pu aider à la communication dans le quartier, donc on a du refaire un communiqué pour le 4 janvier pour ramener plus de monde.

La plupart du temps, les participants étaient présents pour les animations autour des jeux vidéo en collectif (type FIFA, Just dance, minecraft... etc).

Quand je leur ai demandé à l'oral avant de faire mon atelier s'ils étaient sensibles à la robotique et à la programmation, la plupart des gens m'ont répondu non, mais ils sont quand même curieux.

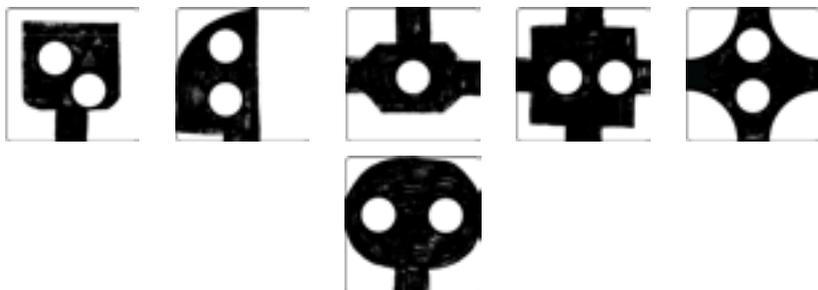
Au lieu de toucher les usagers attendus. qui sont des enfants de 6 à 8 ans, j'ai aussi eu des adolescents et des adultes qui étaient parfois plus intéressés que les enfants. Les enfants étaient un peu plus timide à interagir avec le dispositif et souvent étaient accompagnés de leur frère ou de leur sœur, qui ensuite faisait l'atelier après eux.

J'ai pu alors récolter 14 réponses, qui sont mélangées entre adultes, adolescents et enfants.

L'objectif de la recherche était de connaître la façon dont les gens perçoivent les robots et leur forme, en créant leur robot personnel par eux-mêmes et en me racontant une petite partie de la vie du robot.

Scénario d'usage

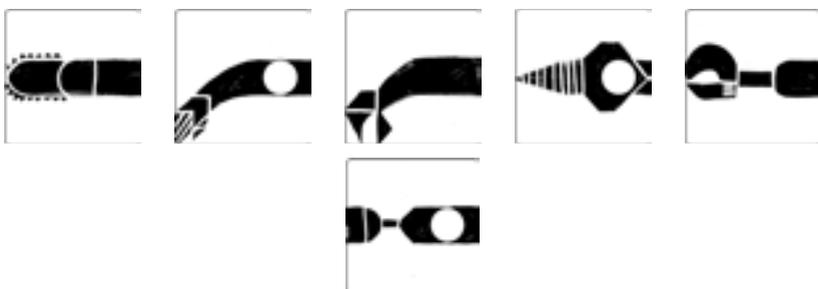
Têtes et corps



Jambes

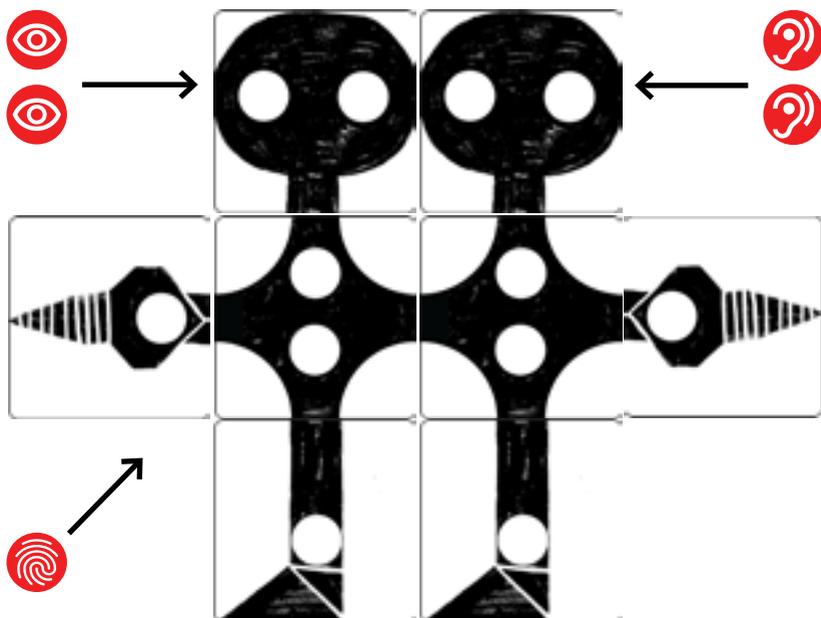


Bras



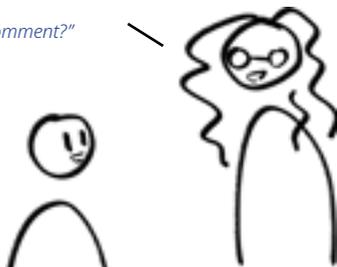
L'utilisateur doit choisir au moins une tête, un corps, deux bras, et une jambe.

Sur un bloc de corps, on a maximum 4 emplacements pour imbriquer d'autres parties.



Quand le robot est assemblé, usager peut prendre des jetons de vue, ouïe, et toucher pour ajouter des sens rudimentaires à son robot en les plaçant dans les espaces vides prévus à cet effet.

*"Que fait ton robot? Qu'est ce qu'il fait dans sa vie?"
 "Est-il gentil ou méchant?"
 "A-t-il un nom? Il s'appelle comment?"*



Quand l'assemblage est fini, il est posé quelques questions à l'orale. L'usager est libre de donner des réponses longues ou courtes

Quelques images en plus...





Résultats



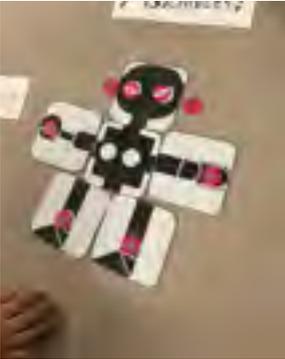
Femme

- Le robot s'appelle Arachnobot
- Constructeur de cabane dans la forêt
- Ses bras d'outil lui permettent une bonne manipulation du bois et une grande précision.
- Il n'est pas violent, malgré son apparence



Homme

- Le robot s'appelle Jaal
- Un oeil "Sharingan" pour connaître l'humeur de ses collègues humains.
- Aime taquiner et se balader
- Adore embêter ses collègues humains
- Coupe du pain, ouvre des bouteilles...
- Ni gentil, ni méchant... Neutre et équilibré



Garçon

- Le robot n'a pas de nom.
- Un robot gentil
- Il répare des maisons et coupe du bois



Garçon (frère du dernier participant)

- Aucun nom aussi.
- Aussi gentil.
- Jardinier, coupe les arbres mort
- Tout-terrain, lui permettant une grande mobilité pour son lieu de travail



Fille

- Le robot n'a pas de nom
- Visse des clous
- Peureux des pompiers, car ils le chassent.
- Neutre, ni gentil ni méchant.



Femme

- Le robot s'appelle Lartiste
- Polyvalent
- Écoute avec le coeur, adoptant un rôle d'ami et d'ecouteur
- Des différents bras pour différentes matières d'art
- Aime créer et est très gentil



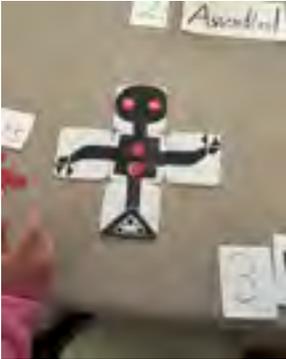
Homme

- Le robot s'appelle Valerian
- Utilitaire
- Guide d'aveugle
- N'a pas de bras mais il peut établir une barrière pour protéger son usager aveugle
- Rôle thérapeutique



Fille

- Le robot s'appelle Elisa
- Gentille
- Elle aime bien jouer et est très curieuse
- N'a pas une rôle précise



Fille

- Le robot s'appelle Tom.
- Danseur de Hip-Hop.
- Gentil.



Garçon [frère du dernier participant]

- Le robot n'a pas de nom.
- Est aussi danseur.
- Est aussi gentil.



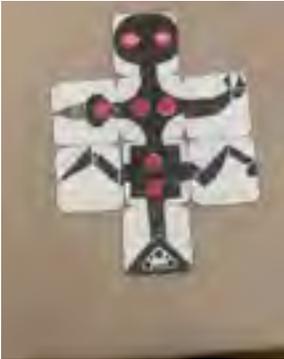
Adolescente

- Le robot n'a pas de nom
- Travailleur dans le BTP, manie des outils et est constructeur
- Travaille avec des humains et d'autres robots, c'est un bon équipier



Adolescente

- Le robot s'appelle C-borg
- Combattant, robot tueur
- Cherche la vengeance aux humains.



Femme

- Le robot s'appelle Multicia
- Un robot multifonction
- Outils à chaque bras
- Il est gentil et au service de l'homme



Homme

- Le robot s'appelle XYZ
- Aime les formes géométriques
- Choregraphe
- Réceptif au public, adore les applaudissements
- Gentil
- "Fils" de bûcheron, a pu dépasser sa fonction primaire

Observations globales des résultats

- On peut observer que tous les robots ont une forme plus ou moins anthropomorphe, et c'est souvent chez les enfants que l'on retrouve cette volonté. Les éléments proposés peuvent avoir une influence, mais il est aussi envisageable de créer des robots avec des formes non-humanoïdes. On peut retrouver cette volonté chez les adultes.
- Il y a une prédominance des roues et des chenilles pour la locomotion.
- La fonction est souvent un travailleur dans le domaine de la construction, parfois on y retrouve quelques robots thérapeutiques et aussi certains qui sont plus intéressés à détruire l'humanité.
- Quand ce sont des frères ou des sœurs qui créent un robot, ils ont tendance à juste changer quelques parties et reprendre les mêmes fonctions. C'est souvent le petit qui copie le grand.
- Parfois, des personnes équipent les robots avec des grands outils jugés dangereux, comme la tronçonneuse et la perceuse, mais leur donnent des rôles qui n'ont rien à voir avec les outils qu'ils ont. Par exemple le robot "fils de bûcheron" qui a voulu devenir chorégraphe.
- Il y a eu pas mal de gens qui avaient du mal à s'imaginer un robot spontanément et ils ont dû réfléchir un peu. Il y en avait certains qui se sentaient un peu intimidés car ils étaient débutants et donc ils ne s'y connaissaient pas trop.
- Un biais vers la forme anthropomorphe est induite avec certaines cartes proposées, comme celles des bras et des jambes humaines, ainsi que la catégorisation des parties qui peut entraîner un biais vers la forme anthropomorphe.

Conclusion

Les personnes interrogées non-initiés à la robotique ont une vision très anthropomorphique du robot. Le robot, pour les débutants, est là pour faire un travail précis qui lui est attribué grâce à la manière dont il est construit. Le robot est un travailleur, et souvent un ami à l'humain, ou au moins un collaborateur, un guide... Les personnes débutantes ont une vision très positive du robot, même si parfois ils estiment que la robotique à l'air très complexe et difficile

BIBLIO

GRAPHIE

COMMENTÉ

BELL, Tim, et al. « Computer Science Unplugged: school students doing real computing without computers », https://www.researchgate.net/publication/266882704_Computer_Science_Unplugged_school_students_doing_real_computing_without_computers. (Page consultée le 16 mai 2024).

Le "Computer Science Unplugged Project" est un projet qui vise à apprendre des notions des sciences de l'informatique à des élèves par des activités qui nécessitent aucun ordinateur et qui met souvent les élèves en mouvement. Je le trouve intéressant parce qu'il met les gens en mouvement pour comprendre des notions assez avancées de l'informatique sans devoir utiliser des ordinateurs.

CLEMENTS, Douglas H, et Julie S MEREDITH. « Research on Logo: Effects and Efficacy », https://www.researchgate.net/publication/234570408_Research_on_LOGO_Effects_and_efficacy. (Page consultée le 19 mai 2024).

Dans cet article, on étudie l'efficacité et les effets de Logo, qui est un ancien langage de programmation créé en 1967 par Seymour Papert, Wally Feurzeig, Cynthia Solomon et Daniel Bobrow. L'article date de 1992 mais il est pertinent pour étudier les nombreux bienfaits du langage Logo et comment on peut l'enseigner pour pouvoir utiliser ses notions en dehors des cours d'informatique.

CYBERNE1. « 1969 - The Logo Turtle - Seymour Papert et al (Sth African/American) », dans [cyberneticzoo.com](http://cyberneticzoo.com/cyberneticanimals/1969-the-logo-turtle-seymour-papert-marvin-minsky-et-al-american/), 10 janvier 2010, <http://cyberneticzoo.com/cyberneticanimals/1969-the-logo-turtle-seymour-papert-marvin-minsky-et-al-american/> (Page consultée le 19 mai 2024).

La tortue robotisée de Logo est un robot créé dans les années 70 pour aider les enfants à apprendre la programmation et la géométrie. Elle est accompagnée avec le langage de programmation Logo, qui est une des premières langages éducatifs et donc on peut la considérer comme l'ancêtre de Scratch. Je trouve cette référence très pertinente par le fait qu'elle utilise un dispositif robotisée avec un langage simple pour apprendre le codage, ainsi de savoir que ma question existe depuis longtemps.

KATTERFELDT, Eva-Sophie, et al. « Talkoo: A new paradigm for physical computing at school », Manchester United Kingdom, IDC '16: Interaction Design and Children, ACM, 21 juin 2016, <https://doi.org/10.1145/2930674.2935990> (Page consultée le 20 mai 2024).

Le kit Talkoo est un kit DIY qui comporte des modules basées sur l'Arduino ainsi qu'un IDE visuel qui peut parler aux différentes modules, qu'elles soient des capteurs ou des moteurs. Il permet les élèves à créer des prototypes très poussés en une heure. Je trouve cette modularité et cette tactilité très pertinentes pour mes recherches car elle fournit un objet tangible et manipulable.

MALONEY, John, et al. « Programming by Choice: Urban Youth Learning Programming with Scratch », https://www.researchgate.net/publication/221537160_Programming_by_Choice_Urban_Youth_Learning_Programming_with_Scratch. (Page consultée le 19 mai 2024).

Dans un centre d'informatique périscolaire on observe les expériences des jeunes de quartier populaire avec le logiciel de programmation scratch après 18 mois d'utilisation. On observe l'importance d'un logiciel accessible mais puissant pour pouvoir apprendre à coder. Vers la fin, ils interrogent comment les jeunes ressentissent Scratch.

AMES, Morgan G. *The Charisma Machine: The Life, Death, and Legacy of One Laptop per Child*, [s.l.], The MIT Press, 2019, <https://doi.org/10.7551/mitpress/10868.001.0001> (Page consultée le 2 décembre 2024).

Le livre parle d'un projet qui s'appelait "One Laptop per Child" (OLPC) et comment il a été une "machine à charisme". Ça veut dire que l'ordinateur portable avait une image très positive et des promesses de grands changements. Les gens de l'occident croyaient beaucoup en son pouvoir de transformer l'éducation et de résoudre des problèmes sociaux. Par contre, ils n'ont pas pris en compte les usagers qui vont s'en servir dans les pays en voie de développement. Le Livre "The Charisma Machine" nous montre les dangers à créer des outils technophiles sans prendre en compte pour qui on a créé cet outil.

BENDANDI, Luca, et al. Tracing the Line: the art of drawing machines and pen plotters, Berlin, Vetro, 2023, 239 p.

Un recueil de différents artistes numériques qui utilisent les traceurs pour créer des œuvres programmées avec une petite partie historique et technique au début. On peut retrouver beaucoup d'artistes qui utilisent le code pour créer des œuvres dessinées uniques.

LIBOW MARTINEZ, Sylvia, et Gary STAGER. Invent to learn: making, tinkering, and engineering in the classroom, Torrance, Calif, Constructing Modern Knowledge Press, 2013, 237 p.

Invent to learn est un livre qui documente l'histoire de l'apprentissage du numérique et offre une nouvelle perspective sur l'utilisation des outils numériques dans la salle de classe. Les auteurs, issus de l'éducation et de l'ingénierie, mettent en avant la pédagogie constructionniste de Seymour Papert et la pensée Maker pour mieux apprendre le numérique aux enfants.

Vera Molnár, aux sources du code, 28 novembre 2023, <https://www.centrepompidou.fr/fr/magazine/article/vera-molnar-aux-sources-du-code> (Page consultée le 9 février 2025).

“Dès 1959, comme une préfiguration de la suite, elle crée sa « machine imaginaire », soit un programme simple qui impulse une transformation des formes, combinant consignes et interdits. Elle décide des formes, des couleurs, des textures, des matériaux, des supports et dessine sur un rouleau les propositions qu'elle retient.”

L'éducation au numérique, une question de forme scolaire | L'Entretien, 2023, <https://www.youtube.com/watch?v=fjJC2Y5n-7bM> (Page consultée le 10 octobre 2024).

Le passage traite de l'intégration du numérique dans l'éducation à travers une interview de Louis Derrac, un acteur clé dans le domaine de l'éducation numérique. Il partage son expérience de 12 ans en tant qu'animateur de l'accès au savoir, initialement centrée sur l'éducation aux médias, avant d'élargir son champ d'étude à la culture numérique. Louis met en lumière le flou entourant le terme “numérique”, notant son évolution vers un concept plus large englobant des systèmes éducatifs. Il offre un aperçu historique, reliant les développements technologiques – des ordinateurs, nés de besoins de calcul, à l'internet, fruit d'une nécessité de communication – aux enjeux sociaux.

REMERCIEMENTS

Remerciements

À propos des couvertures

- La couverture de ce memoire et celle de Programming by Choice a été fait avec p5.js, une variante javascript de Processing.
- Je veux remercier Charmonder sur Openprocessing pour le code source.

Merci...

À *M. Gross* et *Mme. Slaghuis* pour m'avoir corrigée et encouragée.

À *M. Gross*, *Mme. Slaghuis*, *Mme. Buteau*, et *M. Menault* pour m'avoir aidé à créer et monter l'outil de récolte.

À *Mme. Stankoski* pour m'avoir relu et aidée avec la compréhension.

À *M. Kâaniche* pour m'avoir aidée avec la prise des notes lors de l'outil de récolte, ainsi à me soutenir moralement lors des moments difficiles.

À l'équipe de *la CabAnne des Créateurs à Schiltigheim* de m'avoir accueillie lors d'un stage de 1 mois en fablab, ainsi que les deux ateliers outillés.

À *Haricot*, mon cher compagnon robotique, qui m'a permis de découvrir la programmation chez un vrai robot, de m'aider dans mon étude de cas technique et aussi de me soutenir moralement durant les longues nuits d'écriture et de mise en page.

Et dernièrement...

Merci à mes parents de m'avoir encouragé et soutenu pendant l'écriture et la création de ce mémoire.

Sans l'aide de tout le monde cité ici, ce mémoire n'aurait jamais vu le jour.

