



Tiers-lieux, territoire apprenant et recherche : M'L@B, incubateur de l'académie de Montpellier

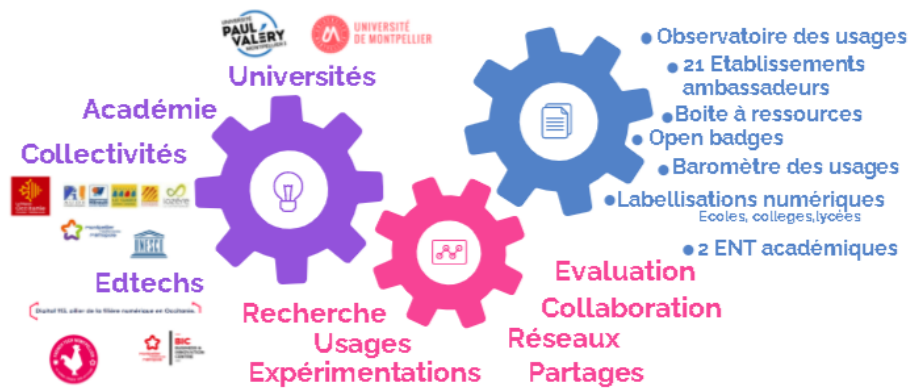
Auteur : Kevin Sigayret, doctorant de Université Montpellier 3, sous la direction de Nathalie Blanc et André Tricot du laboratoire [EPSYLON](#), qui contribuera aussi aux [prochains groupes thématiques numériques](#) - GTnum 2 (hybridation) et 4 (numérique et disparités).

Cette contribution est proposée par la [DANE de Montpellier](#) dans le cadre de l'incubateur M'L@B (thématique « Territoire apprenant »).

Présentation générale de l'incubateur académique

L'incubateur de projets pédagogiques numériques de l'académie de Montpellier M'L@B est né en 2018 de la réponse de la DANE (Délégation Académique au Numérique Éducatif) à l'appel à projet de la DNE dans la [thématique « Territoire apprenant »](#).

Tous les projets pédagogiques portés par la DANE sont intégrés à notre incubateur et [font système](#). Nous avons ainsi pu porter avec la ville de Montpellier le [dossier Unesco Ville apprenante qui a été remporté en juillet 2019](#).



Le projet « Tiers-Lab »

Dans le cadre de l'incubateur académique, le projet « **Tiers-Lab** » vise à accompagner les Tiers-Lieux éducatifs, comme les FabLabs dans les établissements scolaires. On appelle Tiers-Lieu un espace de travail collaboratif dans lequel les compétences de tous sont mises à la disposition de la communauté dans le but de réaliser des projets et de les pérenniser.

Les laboratoires de fabrication « FabLabs » permettent la **réalisation des solutions techniques innovantes de chaque projet** conçues au sein des Tiers-Lieux. **Apprendre en faisant et partager son savoir**, tels sont les principes des « Tiers-Labs ». Ces activités permettent de développer la pensée critique, la créativité, la culture informatique, la résolution de problème, et la collaboration, qui sont les **compétences du 21ème** siècle identifiées entre autres par l'UNESCO et l'OCDE. Ainsi, les élèves et/ou des adultes (de l'établissement, ou externes à l'établissement), seront à même de réaliser tout type de projets grâce à la **mise à disposition de matériel et de compétences**. Les projets peuvent prendre des formes très variées et contribuer à la mise en place de nouvelles pratiques pédagogiques : conception et réalisation d'un objet 3D, programmation d'un objet connecté, amélioration du cadre de vie scolaire, élaboration d'un exposé, conception d'un « escape game », accueil d'un conférencier, etc. Le thème dépend de la communauté capable d'apporter de l'aide au porteur du projet.



Les Tiers-Labs favorisent ainsi l'intelligence collective pour résoudre des problèmes actuels et se préparer aux défis de demain.



Ce projet est accompagné par Réseau Canopé (directions territoriales académiques de Montpellier et Toulouse) avec la **co-animation de formations hybrides** (visioconférence, puis deux journées de formations entre janvier et juin, visioconférence en juin) et concerne **31 établissements**, qui maillent le territoire académique.

A la rentrée 2020, le projet [« Fablab à l'école »](#) avec la Cité des sciences viendra compléter le dispositif Tiers-Lab.

« Tiers-Lab » et projet de recherche

Dans les Tiers-Labs les élèves peuvent créer des robots et les programmer, thème qui est l'objet de la **thèse en cours de Kevin Sigayret**, financée par la DNE dans le cadre de l'incubateur M'L@B.

Son **projet de recherche** propose d'interroger les bénéfices d'un enseignement de la programmation à l'école. Différents dispositifs d'apprentissage peuvent être mis en place pour permettre cet enseignement, en fonction des outils utilisés (logiciels de programmation à vocation éducative, robots pédagogiques, objets programmables connectés...). Les activités débranchées, qui ne font appel à aucun outil numérique, présentent aussi certains avantages qu'il convient



d'identifier et de tester par rapport aux autres techniques. La question de l'efficacité relative de ces différents dispositifs peut alors être posée. Quels bénéfices peut-on espérer de l'apprentissage de la programmation et quels outils sont les plus efficaces pour garantir les meilleures conditions d'apprentissage ?

Quels bénéfices de la programmation en contexte scolaire ?

De nos jours, les compétences en programmation et en codage « se classent explicitement parmi les résultats d'apprentissage pour l'enseignement primaire dans moins de la moitié des systèmes éducatifs européens, mais pour les premier et deuxième cycle de l'enseignement secondaire dans une trentaine de pays » selon le [rapport Eurydice \(2019\)](#). En France, la compétence numérique est définie selon deux axes dont le premier la définit comme une langue : les langages de programmation et les algorithmes. Ce même rapport classe la programmation comme une compétence issue d'un domaine plus large qui est celui de la création de contenus numériques.

L'apprentissage de la programmation, et plus globalement de l'informatique, à l'école connaît un certain essor depuis quelques années. Cet élan peut s'expliquer principalement par l'accès à des **logiciels de programmation visuelle** (par exemple le réputé logiciel Scratch), qui rendent plus accessible une discipline parfois jugée trop complexe, ainsi que par la nécessité pour le futur citoyen de pouvoir appréhender et démystifier le numérique et ses enjeux par une réflexion critique (Romero, Viéville, Duflot-Kremer, de Smet et Belhassein, 2018).

Pour Scherer, Siddiq et Sánchez Viveros (2019), la programmation est une activité similaire aux **activités de résolution de problèmes** dans d'autres domaines, tels que les sciences ou les mathématiques. Les compétences nécessaires à sa maîtrise impliquent différents aspects du raisonnement et, plus généralement, de la cognition humaine. Il est donc tout à fait logique de chercher à explorer les bénéfices que l'on peut retirer de l'apprentissage de la programmation sur le plan cognitif. Par ailleurs, la question du transfert possible des connaissances assimilées en programmation peut être posée. Les interventions visant à favoriser les compétences de codage entraînent-elles des effets indirects positifs dans d'autres disciplines ou d'autres situations ?



Scherer et al. (2019) tentent de répondre à cette vaste interrogation par le biais d'une **méta-analyse reposant sur 105 études distinctes**. Les auteurs définissent les compétences en programmation comme la capacité de créer, modifier et évaluer un code informatique ainsi que d'utiliser les compétences conceptuelles et procédurales nécessaires pour appliquer ces connaissances dans l'optique de résoudre un problème. Ils essaient alors de mesurer ce qu'ils nomment l'« **effet de transfert** », lié à ces compétences en programmation, c'est-à-dire l'impact de celles-ci sur les aptitudes cognitives des individus. Ils distinguent d'ailleurs l'effet de transfert proche, qui concerne les facultés cognitives mobilisées par la pratique de la programmation, de l'effet de transfert distant sur les facultés cognitives sans lien direct avec la programmation.

Les résultats de cette analyse indiquent un **effet positif de l'apprentissage de la programmation**. Ces bénéfices sont observés à la fois sur les compétences en programmation elles-mêmes et sur la maîtrise de concepts en informatique, qui sont améliorées par la pratique, mais aussi dans d'autres domaines de la cognition, ce qui témoigne du transfert possible des compétences acquises en programmation dans d'autres situations.

L'effet de transfert proche est fort et peut être observé dans des situations nécessitant une pensée créative, des compétences en Mathématiques, en métacognition ou en raisonnement logique. L'étude ne permet cependant pas de conclure de manière définitive sur l'effet de transfert distant, qui semble exister néanmoins, mais pas pour toutes les compétences cognitives et pas dans les mêmes proportions. Il est aussi à noter que les effets sont davantage importants lorsque le groupe contrôle est inactif en comparaison avec un groupe contrôle actif.

Toujours dans l'optique de mesurer les bénéfices secondaires de la programmation, Popat et Starkey (2019) proposent une **revue de littérature**, regroupant une dizaine d'articles, focalisée sur l'influence du codage sur les résultats scolaires des enfants. Les chercheurs mettent en exergue une influence de la programmation sur certaines compétences dites « de haut niveau ». La capacité à raisonner semble ainsi améliorée par la pratique du code, comme en témoignage l'étude de Psycharis et Kallia (2017), citée dans cette revue de littérature. 66 élèves de lycée était répartis en deux groupes de 33 élèves et un seul de ces deux groupes participait à des activités de programmation.

On observe que la pratique de la programmation semble responsable d'un accroissement significatif des capacités de raisonnement des élèves qui y ont pris part en comparaison avec le groupe témoin. Il en va de même pour le sentiment d'auto-efficacité en mathématiques qui augmente significativement. En



revanche, l'amélioration de l'aptitude à résoudre des problèmes mathématiques n'est pas significative pour les élèves qui ont participé aux activités de programmation.

Popat et Starkey (2019) remarquent par ailleurs que les résultats éducatifs de l'apprentissage de la programmation dépendent principalement de son intégration dans le programme scolaire et de la **conception pédagogique des activités** proposées qui seront susceptibles de favoriser ou non le développement de compétences secondaires.

Les différents outils pour l'enseignement de la programmation et leurs effets

Messer, Thomas, Holliman et Kucirkova (2017) ont voulu tester l'influence de **l'apprentissage de la programmation sur les compétences des enfants en mathématiques, sur leur habileté visuo-spatiale et sur leur mémoire de travail**. Pour ce faire, les chercheurs ont réparti aléatoirement 41 enfants de 5 ou 6 ans dans trois conditions expérimentales. Le premier groupe s'exerçait à la programmation via l'utilisation d'une tablette tactile, un second groupe s'entraînait de manière plus classique sur papier alors que le dernier groupe (groupe contrôle) ne participait à aucune activité de programmation, celles-ci étant remplacées par des exercices d'addition et de soustraction. Pour l'ensemble des enfants, ces activités se déroulaient deux fois par semaine, à raison de 10 minutes par session, pendant une durée totale de 6 semaines.

Les 3 groupes de participants manifestent **globalement une amélioration de leurs compétences spatiales et de leur niveau en mathématiques**, les résultats post-test étant supérieurs à ceux obtenus pré-test. Selon les auteurs, cette amélioration ne pourrait probablement pas être imputable à un simple effet de maturation, indépendant des activités réalisées, étant donnée la durée relativement faible de l'étude. Aucun effet significatif n'est en revanche observé sur la mémoire de travail.

Plus important encore, ce protocole expérimental ne montre **aucune différence significative entre les trois groupes**. L'apprentissage de la programmation ne semble donc pas plus efficace que des activités plus directes pour développer les compétences étudiées. Cependant, les auteurs remarquent que ces résultats sont tout de même encourageants car ils font état d'une réelle capacité de la programmation à **développer des compétences transférables dans d'autres domaines** plus ou moins reliés. A priori, l'usage d'outils numériques pour enseigner la programmation ne semble pas non plus indispensable.



Cette dernière constatation, qui peut paraître contre-intuitive au vu de l'interdépendance entre le numérique et la programmation, fait écho aux travaux de Romero et al. (2018). Les auteurs identifient un ensemble de bénéfices supposés à **l'apprentissage de la programmation de manière débranchée**, sans faire appel aux outils technologiques. La charge cognitive liée à l'utilisation d'une machine, qui nécessite certaines compétences techniques essentielles, serait un premier obstacle à l'apprentissage réalisé directement sur logiciel. Par ailleurs, l'informatique débranchée s'accorderait davantage avec les théories de la cognition incarnée qui soulignent **l'importance des actions sensori-motrices concrètes** dans le processus d'apprentissage. Enfin, les activités débranchées permettraient de construire des analogies tangibles et concrètes en relation avec les notions abstraites rencontrées en algorithmique, ce qui en faciliterait l'appropriation.

Les auteurs comparent de manière quasi-expérimentale l'apprentissage de la programmation par des activités débranchées, dans une première condition avec un apprentissage plus classique et dans une seconde condition via un logiciel à vocation éducative. L'échantillon d'élèves de CM2 mis à contribution reste assez faible (13 enfants dans la première condition, 10 dans la seconde). Ce protocole ne fait état d'aucune différence significative entre les deux groupes. Ceci montre, dans la continuité de l'étude réalisée par Messer et al. (2017), qu'en ce qui concerne les activités branchées et débranchées pour l'apprentissage de la programmation, **la question de la supériorité d'une approche sur une autre est encore loin d'être tranchée**.

De même, Hermans et Aivaloglou (2017) proposent une autre étude qui tente de comparer l'impact d'une première exposition à la programmation avec ou sans **Scratch**, langage visuel de programmation et, par extension, logiciel qui permet aux enfants de facilement créer leurs propres contenus interactifs (Olabe, Olabe, Basogain et Castaño, 2011).

35 élèves de primaire sont séparés en deux groupes. Dans une condition, les élèves s'initient au codage sur Scratch pendant 4 semaines alors que dans la seconde condition cette initiation est réalisée par le biais d'activités débranchées. Au bout de ces 4 semaines, la formation est poursuivie pendant 4 semaines supplémentaires mais, cette fois-ci, tous les élèves dans les deux groupes utilisent le logiciel Scratch.

À l'issue de cet entraînement, les deux chercheurs constatent une absence de différence significative entre les performances des élèves dans les deux conditions sur la compréhension des concepts de programmation. Cependant, les participants du groupe centré sur les activités débranchées montrent une plus grande confiance dans leurs capacités.



De même, Sáez-López, Román-González et Vázquez-Cano (2016) ont étudié l'intégration d'un langage de programmation visuel auprès de 107 élèves de CM2 et 6ème en Espagne en s'appuyant sur le logiciel Scratch, afin d'évaluer les bénéfices de l'utilisation de ce type de logiciel sur l'attitude et les processus d'apprentissages des élèves. L'étude est réalisée sur deux années consécutives avec un groupe témoin non randomisé de 32 élèves. 20 sessions d'une heure sont intégrées en cours de « Sciences » et d'« Art » tout au long de ces deux années.

Les résultats montrent premièrement qu'**une approche pédagogique active améliore l'apprentissage des concepts de programmation**, la logique et la maîtrise de l'outil informatique. Du côté des données subjectives, les élèves ayant participé à cette recherche, tout comme les observateurs, estiment que travailler avec un logiciel de programmation visuel est amusant, motivant et engageant pour l'apprenant. Ces résultats positifs incitent les auteurs à recommander l'usage de Scratch en fin de primaire et au début du secondaire.

Xu, Ritzhaupt, Tian et Umapathy (2019) réalisent une **méta-analyse qui vise à comparer l'effet de différents langages de programmation** sur l'apprentissage de cette discipline à l'école. Les langages basés sur des « blocs » (langages visuels, comme Scratch) sont comparés aux langages plus classiques, basés sur du texte. L'analyse se fonde sur l'étude de 13 publications regroupant 52 comparaisons de tailles d'effets sur des variables cognitives et affectives.

Un faible effet positif est obtenu en faveur de la programmation par blocs sur la cognition. Pour les dimensions affectives, l'effet positif du langage par blocs est négligeable. Les auteurs déplorent un manque de données rendant toute réponse définitive impossible concernant les plus-values de ces « nouveaux » langages de programmation. L'étude en question soulève plus de questionnements qu'elle n'apporte de réponses concrètes.

Enfin, une méta-analyse récente de Scherer, Siddiq et Vivares (2020) propose d'**interroger l'usage des outils visuels et physiques disponibles pour enseigner la programmation aux élèves**. Certaines recherches antérieures témoignent d'une possible supériorité des langages de programmation visuels sur les langages de programmation basés sur du texte, en terme d'accessibilité notamment. En outre, l'utilisation de robots physiques programmables est supposée renforcer la compréhension de concepts informatiques ainsi que l'engagement et la motivation.



Les résultats montrent que le codage est une compétence que l'on peut entraîner et améliorer par la pratique. L'apprentissage de la programmation à l'école a un effet fort et évident sur cette compétence. L'analyse confirme par ailleurs une augmentation de la taille de l'effet lorsque les langages de programmation visuels ou la robotique sont utilisés, bien que l'influence des robots pédagogiques, et des outils physiques en général, soit encore trop peu étudiée expérimentalement.

La robotique et les objets concrets au service de l'enseignement de la programmation

Pour Olabe et al. (2011), l'usage combiné de Scratch et de la robotique s'inscrirait dans le **paradigme d'apprentissage constructionniste**, dans lequel l'élève est le protagoniste principal de son apprentissage et apprend principalement par la pratique. L'usage des robots à des fins éducatives pourrait bien se révéler précieux pour favoriser le développement des facultés cognitives chez l'enfant. La recherche expérimentale identifie un ensemble de compétences pour lesquelles la robotique pédagogique, souvent couplée avec des activités de programmation, présenterait un intérêt.

Récemment, Nogry (2018) a filmé puis analysé des **séances de robotique pédagogique avec des enfants de CP âgés de 6 ans**. Au début de l'activité, les élèves apprennent pas à pas à manipuler le robot. Ils constatent rapidement un écart entre leurs intentions et déplacement réel du robot. Cela les conduit à adopter une stratégie basée sur de **multiples essais et erreurs**. Sans jamais être verbalisé par l'enseignante, le concept de programme (ou de séquence) est progressivement assimilé par les élèves. La structuration de la tâche à accomplir et les médiations de l'enseignante conduisent les élèves à adopter une démarche de planification des déplacements du robot, de réflexion sur la séquence d'instructions à donner et d'observation des écarts entre le déplacement prévu et le déplacement observé. Ainsi, les enfants comprennent le caractère différé de l'exécution du programme.

La robotique éducative présente la particularité de **concilier l'usage d'un dispositif numérique avec la manipulation d'un objet physique et concret**. Mais l'utilisation d'objets concrets pour favoriser l'enseignement de concepts abstraits n'est pas nouvelle. Tous les élèves sont plus ou moins confrontés à ce genre d'outils, notamment en mathématiques où de nombreuses études ont été réalisées pour tenter d'en mesurer l'effet.

Kaminsky, Sloutsky et Heckler (2006) ont mis en place une étude avec 19 élèves de 6ème (6th grade) répartis aléatoirement en deux groupes. Un concept mathématique nouveau était enseigné à l'ensemble



des élèves mais, dans une seule des deux conditions, cet apprentissage était réalisé dans une situation concrète pertinente. Par la suite, les élèves étaient confrontés à un nouveau problème impliquant le même concept mathématique.

Les résultats de cette étude montrent que **les enfants n'ont pas nécessairement besoin d'une situation concrète pour acquérir un concept abstrait**. L'apprentissage initial est légèrement supérieur dans la condition concrète mais lorsqu'un problème nouveau intervient, on observe que l'effet de transfert est plus important dans la condition contrôle et, par conséquent, les élèves non impliqués dans une situation concrète sont meilleurs pour le résoudre. Tous les sujets étaient explicitement prévenus que ce qu'ils venaient d'apprendre était utile pour faire face à cette nouvelle situation.

Les auteurs interprètent l'échec plus important des élèves en condition concrète par leur incapacité à reconnaître la structure du concept étudié et à faire correspondre les éléments dans les deux situations. Par conséquent, si l'utilisation d'objets et de situations concrètes peut faciliter l'apprentissage immédiat, elle pourrait aussi entraver la possibilité de transférer les connaissances acquises à d'autres situations similaires. .

Plus récemment, Carbonneau, Marley et Selig (2013) ont mené à bien une **méta-analyse sur les preuves empiriques en faveur de l'usage des objets matériels pour l'apprentissage des mathématiques**. 55 études distinctes ont été sélectionnées et chacune d'entre elles proposait une comparaison avec un groupe contrôle dans lequel l'apprentissage était réalisé uniquement à l'aide de symboles abstraits. Au total, ce sont les résultats de plus de 7000 élèves, tous âges confondus (de la maternelle à l'université), qui sont évalués et analysés.

Globalement, cette analyse met en exergue **un effet faible à modéré en faveur de l'apprentissage des mathématiques par le biais des objets concrets**. Cependant, les données obtenues semblent indiquer que l'usage d'objets matériels a un impact différent selon les objectifs d'apprentissage. On peut notamment observer un effet de taille importante en ce qui concerne la rétention de l'information mais lorsqu'il s'agit de la résolution de problèmes, du transfert des connaissances acquises à des situations nouvelles ou encore de la capacité à justifier ses réponses en fournissant une explication sur la méthode employée, l'effet mesuré est plus faible.

Sullivan et Heffernan (2016) définissent le matériel de manipulation éducatif comme un référent concret à des concepts abstraits. Ils s'intéressent plus particulièrement au **matériel de manipulation informatique**



(ou numérique) qui possède une capacité informatique embarquée, tels que les **kits de construction robotique** par exemple. Pour les auteurs, la relation directe entre le programme qui est exécuté et le mouvement du robot facilite la compréhension de la nature duelle de l'objet, en tant qu'objet physique et concret se déplaçant dans l'espace d'une part, mais aussi en tant que représentation d'un programme, qui n'est rien d'autre qu'une suite de symboles et de processus abstraits. Le feedback immédiat permis par la robotique induit l'observation de divergences entre le mouvement attendu du robot et son mouvement réel, ce qui favorise l'association entre le robot concret et le programme abstrait dans l'esprit de l'enfant, tout en incitant l'élève à générer des inférences sur les raisons qui sous-tendent ces divergences et donc à potentiellement **améliorer ses aptitudes au raisonnement causal**.

Si les objets physiques, matériels et concrets peuvent présenter des avantages certains, ceux-ci ne constituent pas non plus une panacée qui permettrait systématiquement de faciliter l'apprentissage de concepts abstraits complexes à appréhender. La littérature scientifique s'est principalement concentrée sur l'usage de ces objets en mathématiques et identifie un problème majeur : la difficulté pour l'enfant de faire correspondre l'objet qu'il manipule ou la situation concrète dans laquelle il est impliqué avec un phénomène plus général, indépendant de cette situation. Cependant, dans le domaine de la robotique éducative, la quasi transparence entre les mouvements et déplacements du robot et le contenu du programme qui les détermine pourrait faciliter la capacité des apprenants à intégrer la dimension duelle de l'objet qui agit comme incarnation concrète d'un processus abstrait sous-jacent. Cette compréhension pourrait avoir un impact sur l'acquisition des notions et des compétences en programmation notamment.

Le potentiel de la robotique éducative pour améliorer la maîtrise, parfois délicate, de notions en programmation est étayé par certaines observations et certains résultats expérimentaux mais globalement la recherche souffre d'un manque d'études comparatives solides sur l'apprentissage de la programmation suivant l'utilisation ou non des objets concrets programmables. Un effet des robots sur la motivation et l'engagement des élèves est parfois observé même si les liens qui unissent la motivation de l'apprenant et performances d'apprentissages restent ténus (voir à ce sujet : Amadiou et Tricot, 2014). Le caractère attrayant du robot pourrait aussi susciter un intérêt plus prononcé pour les disciplines scientifiques bien que, là encore, des données contradictoires doivent modérer notre propos. **La recherche expérimentale sur les bénéfices de la robotique éducative n'en est encore qu'à ses débuts et il convient de faire preuve de prudence quant à ses mérites potentiels.**



La recherche qui sera réalisée tout au long de ce travail de thèse s'inscrit dans le champ de la **psychologie expérimentale**, méthodologie fondée sur l'observation et l'acquisition de connaissances par le biais d'expérimentations. Au cours de l'année scolaire 2020-2021, un ensemble d'observations sont prévues, avec l'aide de la DANE de l'Académie de Montpellier, pour tenter de répondre aux interrogations que suscitent les différents outils au service de l'apprentissage de la programmation. Notre **hypothèse** défend l'idée que **la robotique pédagogique présente l'avantage majeur de permettre la manipulation directe par l'apprenant d'un objet physique concret**, susceptible de se déplacer et de bouger réellement dans l'espace. Cette caractéristique particulière du robot permettrait ainsi de faciliter l'apprentissage et la maîtrise des notions fondamentales de la programmation.

Références

Liens

Quelques ressources issues des Tiers-Labs :

@cadémie du numérique : <https://academienumerique.fr/>

Mobilab 66 : <http://mobilab.fablab66.fr>

Exemples d'usages :

<https://usagesdunumerique.ac-montpellier.fr/tiers-lab-1>

<https://usagesdunumerique.ac-montpellier.fr/programmation-0>

<https://usagesdunumerique.ac-montpellier.fr/activites-debranchees-0>

Le réseau des Tiers-Lab s'organise aussi autour de FabLab grand public :

- Carcassonne - Les forges numériques : <https://fablabcarcassonne.wordpress.com>



- Nîmes - **le FabLab** : <https://lefablab.fr/>
- Montpellier - **LabSud** : <https://www.labsud.org/>
- Perpignan - **Squaregolab** : <https://www.squaregolab.com/>

Partenaires universitaires : IUT de Nîmes et de Perpignan

Bibliographie

- Amadiou, F. & Tricot, A. (2014). *Apprendre avec le numérique. Mythes et Réalités*. Paris : Retz.
- Carbonneau, K. J., Marley, S. C., & Selig, J. P. (2013). A meta-analysis of the efficacy of teaching mathematics with concrete manipulatives. *Journal of Educational Psychology*, 105(2), 380.
- Commission européenne/EACEA/Eurydice, 2019. L'éducation numérique à l'école en Europe. Rapport Eurydice. Luxembourg: Office des publications de l'Union européenne.
- Hermans, F., & Aivaloglou, E. (2017, November). To scratch or not to scratch? A controlled experiment comparing plugged first and unplugged first programming lessons. In *Proceedings of the 12th workshop on primary and secondary computing education* (pp. 49-56).
- Kaminski, J. A., Sloutsky, V. M., & Heckler, A. F. (2006, July). Do children need concrete instantiations to learn an abstract concept. In *Proceedings of the XXVIII Annual Conference of the Cognitive Science Society* (pp. 1167-1172). Erlbaum Mahwah, NJ.
- Messer, D., Thomas, L., Holliman, A., & Kucirkova, N. (2018). Evaluating the effectiveness of an educational programming intervention on children's mathematics skills, spatial awareness and working memory. *Education and Information Technologies*, 23(6), 2879-2888.
- Nogry, S. (2018). Comment apprennent les élèves au cours d'une séquence de robotique éducative en classe de CP?. De 0 à 1 ou l'heure de l'informatique à l'école, 235.
- Olabe, J. C., Olabe, M. A., Basogain, X., & Castaño, C. (2011). Programming and robotics with Scratch in primary education. *Education in a technological world: communicating current and emerging research and technological efforts*, 356-363.
- Popat, S., & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365-376.



- Psycharis, S., & Kallia, M. (2017). The effects of computer programming on high school students' reasoning skills and mathematical self-efficacy and problem solving. *Instructional Science*, 45(5), 583-602.
- Romero, M., Viéville, T., Dufлот-Kremer, M., de Smet, C., & Belhassein, D. (2018, August). Analyse comparative d'une activité d'apprentissage de la programmation en mode branché et débranché.
- Sáez-López, J. M., Román-González, M., & Vázquez-Cano, E. (2016). Visual programming languages integrated across the curriculum in elementary school: A two year case study using "Scratch" in five schools. *Computers & Education*, 97, 129-141.
- Scherer, R., Siddiq, F., & Sánchez Viveros, B. (2019). The cognitive benefits of learning computer programming: A meta-analysis of transfer effects. *Journal of Educational Psychology*, 111(5), 764.
- Scherer, R., Siddiq, F., & Viveros, B. S. (2020). A meta-analysis of teaching and learning computer programming: Effective instructional approaches and conditions. *Computers in Human Behavior*, 106349.
- Sullivan, F. R., & Heffernan, J. (2016). Robotic construction kits as computational manipulatives for learning in the STEM disciplines. *Journal of Research on Technology in Education*, 48(2), 105-128.
- Xu, Z., Ritzhaupt, A. D., Tian, F., & Umapathy, K. (2019). Block-based versus text-based programming environments on novice student learning outcomes: a meta-analysis study. *Computer Science Education*, 29(2-3), 177-204.