

# Introduction de l'algorithmique dans le cadre d'enseignements interdisciplinaires au cycle 2 du primaire en France

**Michèle COUDERETTE**<sup>49</sup>

LDAR, Univ Paris Est Créteil, F-94010 Créteil, France  
IRL CRM-CNRS, Université de Montréal, Canada

**Dominique LAVAL**<sup>50</sup>

CY Cergy Paris Université, F-95000 Communauté d'Agglomération de Cergy-Pontoise, France  
INSPE de l'académie de Versailles, F-78100 Saint-Germain-en-Laye, France

**Résumé.** Depuis la rentrée 2016, une initiation à la programmation informatique est inscrite dans les programmes de l'école primaire en France. Celle-ci permet d'associer des activités de repérage dans l'espace, comme le codage du déplacement d'un robot. Par ailleurs, une telle initiation offre l'opportunité d'un travail interdisciplinaire dans le domaine de la littérature de jeunesse, avec la compréhension et la transcription de récits en programmes informatiques implémentables dans un environnement numérique. Dans cet atelier, les participants sont amenés à résoudre deux tâches de programmation informatique distinctes, pour des élèves de 6 à 8 ans : déplacement d'un robot dans un tunnel et transcription d'un conte en des séquences algorithmiques afin de créer une animation implémentable dans Scratch Junior. La résolution de ces tâches nécessite un cycle de modélisation connectée à l'informatique. Se pose alors la question des obstacles que les élèves peuvent rencontrer, lors des différentes phases de modélisation connectées à l'algorithmique et à la programmation.

**Mots-clés.** Algorithmique au cycle 2, Interdisciplinarité, Modélisation, Robotique, Déplacement dans un tunnel, Scratch Junior, Espace de travail algorithmique, Codage, Logique, Raisonnement, Conte.

**Abstract.** Since the start of the 2016 school year, an introduction to computer programming has been included in the French primary school curriculum. This makes it possible to associate spatial location activities, such as coding the movement of a robot. Furthermore, such an initiation offers the opportunity for interdisciplinary work in the field of children's literature, with the understanding and transcription of stories into computer programs implementable in a digital environment. In this workshop, participants are asked to solve two distinct computer programming tasks, for students aged 6 to 8: moving a robot in a tunnel and transcribing a story into algorithmic sequences to create an animation implementable in Scratch Junior. Solving these tasks requires a modeling cycle connected to IT. The question then arises of the obstacles that students may encounter during the different modeling phases connected to algorithms and programming.

**Keywords.** Algorithmic, Interdisciplinary work, Modelling, Robotic, Moving in a tunnel, Scratch Junior, Algorithmic Working Space, Coding, Logic, Reasoning, Tale.

**Resumen.** Desde el inicio del año escolar 2016, se ha incluido en el plan de estudios de la escuela primaria una introducción a la programación informática. Esto permite asociar actividades de localización espacial, como por ejemplo codificar el movimiento de un robot. Además, dicha iniciación ofrece la oportunidad de realizar un trabajo interdisciplinario en el campo de la literatura infantil, con la comprensión y transcripción de cuentos en programas informáticos implementables en un entorno digital. En este taller, se pide a los participantes que resuelvan dos tareas distintas de programación informática, para estudiantes de 6 a 8

---

<sup>49</sup> [michele.couderette@u-pec.fr](mailto:michele.couderette@u-pec.fr)

<sup>50</sup> [dominique.laval@cyu.fr](mailto:dominique.laval@cyu.fr)

años: mover un robot en un túnel y transcribir una historia en secuencias algorítmicas para crear una animación implementable en Scratch Junior. Resolver estas tareas requiere un ciclo de modelado conectado a TI. Surge entonces la pregunta de los obstáculos que los alumnos pueden encontrar durante las diferentes fases de modelado relacionadas con los algoritmos y la programación.

**Palabras clave.** Algorítmica, Trabajo interdisciplinario, Modelado, Robótica, Moverse en un túnel, Scratch Junior, Espacio de trabajo algorítmico, Codificación, Lógica, Razonamiento, Cuento.

**Zusammenfassung:** Seit Beginn des Schuljahres 2016 ist eine Einführung in die Computerprogrammierung Teil des französischen Grundschullehrplans. Dadurch ist es möglich, räumliche Standortaktivitäten zuzuordnen, um beispielsweise die Bewegung eines Roboters zu kodieren. Darüber hinaus bietet eine solche Einführung die Möglichkeit zur interdisziplinären Arbeit im Bereich der Kinderliteratur mit dem Verstehen und Schreiben von Geschichten in Computerprogramme, die in einer digitalen Umgebung umsetzbar sind. Die Teilnehmer sollen zwei unterschiedliche Computerprogrammieraufgaben für Schüler im Alter von 6 bis 8 Jahren lösen: einen Roboter in einem Tunnel bewegen und eine Geschichte in algorithmische Sequenzen umwandeln, um eine in Scratch Junior umsetzbare Animation zu erstellen. Zur Lösung dieser Aufgaben ist ein mit der IT verbundener Modellierungskreislauf erforderlich. Es stellt sich dann die Frage, auf welche Hindernisse Lernende während der verschiedenen Modellierungsphasen im Zusammenhang mit Algorithmen und Programmierung stoßen können.

**Schlüsselwörter:** Algorithmen, Interdisziplinarität, Modellierung, Robotik, Bewegung in einem Tunnel, Scratch Junior, Algorithmischer Arbeitsbereich, Kodierung, Logik, Argumentation, Märchen.

## *Introduction*

Depuis la rentrée 2016, une initiation à la programmation informatique est inscrite dans les programmes de l'école primaire en France. Alors qu'au cycle 4, la programmation est un thème d'étude à part entière, au primaire, algorithmique et programmation sont désignés comme des objets transversaux pouvant être travaillés dans le champ des mathématiques, des sciences et technologies, du langage au travers d'activités de codage, de repérage et déplacements dans l'espace utilisant ou non des objets numériques tels que des robots de planchers ou des applications logicielles. Le document Eduscol « initiation à la programmation au cycle 2 et 3 » (MEN, 2016, p. 1) précise qu'une « initiation à la programmation pourra être une opportunité pour des travaux interdisciplinaires : avec le champ questionner le monde au cycle 2, par exemple, autour de la question du repérage ou avec le français, dans le développement des usages du langage oral ou écrit, notamment en créant des histoires illustrées par de courtes animations créées par les élèves ».

Dans notre atelier, nous avons présenté deux situations distinctes introduisant des concepts informatiques au cycle 2, l'une utilisant des robots Blue-Bot (situation des tunnels présentée par Michèle Couderette), l'autre l'environnement numérique ScratchJr (le conte « Le lion et les trois buffles » de Dhouib et Angeli présentée par Dominique Laval). Il s'agit dans ces recherches de réfléchir à l'introduction de concepts algorithmiques dans des contextes d'interdisciplinarité. L'atelier s'est focalisé plus particulièrement sur les obstacles rencontrés par les élèves lors des phases de modélisation.

### *1. Programmer la traversée d'un tunnel par une Blue-Bot (M. Couderette)*

Nous présentons dans cette première partie une situation, que nous appelons « traversée de tunnels » par commodité de langage. Les deux tâches issues de cette situation sont testées lors de deux séances de classe au cycle 2 (6, 7 ou 8 ans).

Précisons tout d'abord que la situation des tunnels prend son origine dans une recherche collaborative en cours avec trois enseignants d'école primaire. Le questionnement initial porte sur « comment l'informatique et notamment l'algorithmique peuvent-elles contribuer à la construction d'autres apprentissages, en particulier en mathématiques ? ».

### Mise en contexte

Dans un premier temps, une vidéo d'un métro automatique entrant dans un tunnel est présentée aux élèves. Cette vidéo montre clairement l'absence de conducteur. Cette mise en contexte permet aux élèves d'entrer plus rapidement dans la compréhension des tâches telles qu'elles sont ensuite énoncées aux élèves.

Les deux situations de classe proposent de programmer un Blue-Bot pour que celui-ci se déplace d'une station A à une station B en traversant un tunnel droit ou un tunnel courbé (fig.1).



Figure 1 : mise en contexte

Les tunnels proposés aux élèves sont fabriqués de manière contraignante : (i) en carton sans la moindre marque pouvant indiquer leur longueur, (ii) suffisamment longs pour que les élèves ne puissent pas agir sur le Blue-Bot au cours de son déplacement. De ce fait, les élèves sont amenés à anticiper les déplacements dans leur intégralité et donc à produire puis tester un programme complet. Les situations répondent ainsi à un premier principe au cœur de la programmation, l'anticipation.

Deux séances précèdent les séances de « traversée de tunnels ». L'objectif de ces deux séances est de faire émerger le fonctionnement du robot, en particulier le repérage des différentes touches ainsi que leurs fonctions, en particulier les touches « effacer » pour vider la mémoire, « pause » pour suspendre quelques secondes le déplacement du Blue-Bot et la touche « GO » pour lancer le programme (fig.2). Le robot avance pas à pas, chacun des pas étant de 15 cm.

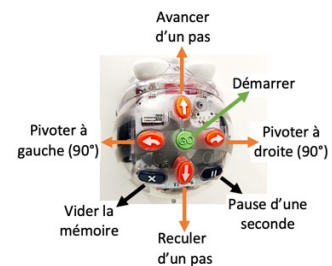
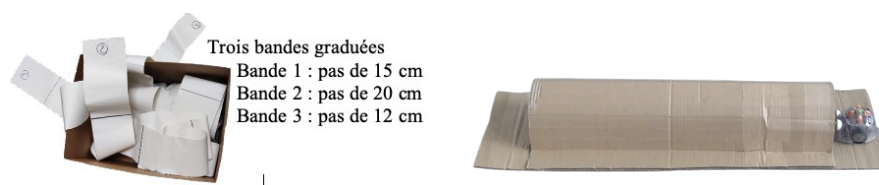


Figure 2 : Robot Blue-Bot

## 1.1. Description des tâches

### Première tâche : programmer un robot pour traverser un tunnel droit de 60 cm de longueur

Pour résoudre cette tâche, les élèves disposent de bandes graduées différentes (fig. 3).



**Figure 3 :** Tunnel droit et bandes graduées.

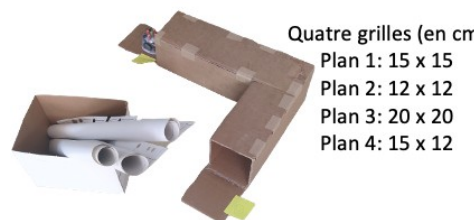
Ces bandes sont construites de telle façon qu'elles peuvent toutes convenir pour mesurer les tunnels : avec la bande A, le tunnel mesure 4, mais avec la bande B, il mesure 3 et avec la bande C, il mesure 5. Pour autant, seule la bande A, graduée tous les 15 cm, convient car dans le pas du robot Blue-Bot.

La consigne formulée aux élèves est de « programmer le déplacement d'un robot d'une station A (à l'entrée du tunnel) à la station B (à la sortie du tunnel) ».

Pour pouvoir répondre à cette consigne, les élèves doivent donc basculer d'un cadre mathématique (mesure dans un système d'unité non conventionnel puis démarche d'investigation pour déterminer la bande idoine), à un cadre de programmation informatique (programmer les déplacements du robot).

### Deuxième tâche : programmer un robot pour qu'il traverse un tunnel coudé

Les dimensions du tunnel sont indiquées sur la figure 4 ci-contre. Les élèves ont cette fois-ci à leur disposition des plans quadrillés différents. A l'inverse de la tâche précédente, plusieurs plans (plans 1 et 4) permettent de résoudre la situation et de produire un programme correct.



**Figure 4 :** Tunnel coudé et plans quadrillés.

Si la première situation a une dominante mathématique, cette deuxième situation met l'accent sur la programmation des robots. Il s'agit de réinvestir les connaissances et compétences construites lors de la résolution de la première tâche pour se focaliser sur la programmation des déplacements du robot.

La programmation demande une instruction supplémentaire (pivoter) ce qui peut paraître sans réelles difficultés... Nous y revenons dans la section « premiers constats ».

## 1.2. Premiers constats

### Un objet se révélant problématique : le Blue-Bot !

Signalons tout d'abord que le Blue-Bot, bien que son nom sous-entende qu'il soit un robot, est en réalité un automate. Il réagit à la pression de touches positionnées sur le capot et non à des événements provenant de son environnement. Par commodité, nous gardons cette terminologie.

La manipulation du Blue-Bot fait apparaître plusieurs sources potentielles de difficultés pour des élèves de cycle 2.

Les Blue-Bots sont des objets orientés, ce qui demande à l'élève de repérer chaque déplacement dans le repère du Blue-Bot et non pas par rapport à lui-même.

Les déplacements sont commandés différemment selon que l'on change de direction ou pas : pour se déplacer en ligne droite (avancer ou reculer), une seule commande suffit alors que pour se déplacer en tournant, deux sont nécessaires, l'une pour indiquer la direction (pivoter à droite ou à gauche) l'autre pour avancer.

Vider la mémoire s'opère par la touche « X ». Si la mémoire contient un premier programme, les nouvelles instructions se rajoutent à celles du programme initial. Or rien ne permet à l'élève de savoir si la mémoire est vide. Cet aspect cumulatif de saisie des commandes ne facilite ainsi pas l'interprétation des rétroactions du Blue-Bot.

Enfin, la longueur du pas du Blue-Bot est très proche de sa taille ce qui, nous le verrons par la suite, induira une technique non attendue de résolution du problème « traversée du tunnel droit ».

### Des contributions réciproques aux apprentissages

Si, pour les élèves, l'objectif consiste à ce que le robot traverse le tunnel, les intentions didactiques des enseignants diffèrent.

**La première tâche** donne l'occasion aux enseignants de revenir sur des apprentissages mathématiques, en particulier ceux liés à la mesure : mesurer dans des systèmes non conventionnels, insister sur le lien entre étalon et résultat de la mesure, conforter les techniques de mesurages pour les élèves les plus faibles.

L'opportunité d'un prêt d'un deuxième robot « robot-souris » d'un pas différent de celui du Blue-Bot (pas de 12 cm), permet de se rendre compte que la mesure des tunnels dépend de l'étalon choisi, et que cette donnée est particulièrement importante pour programmer les déplacements des robots. D'où la nécessité de développer une démarche pour déterminer laquelle des trois bandes graduées serait l'outil permettant de résoudre le problème et ce, en fonction du robot Blue-Bot ou Robot-Souris.

Plusieurs procédures sont observées lors de la recherche de la bande idoine :

Désigner une case sur la bande quadrillée assez éloignée de la case de départ puis programmer le robot pour qu'il se déplace sur cette case. Si le robot se déplace bien sur la case prévue, la bande convient.

Appuyer sur les boutons flèches pour que le robot se déplace. S'il se déplace de case en case, alors la bande peut convenir, mais s'il s'arrête « à cheval » sur deux cases, c'est-à-dire sur une graduation, le pas du robot est trop petit ou trop grand et la bande ne convient pas.

Poser le robot sur les cases, et comparer la taille du robot et la taille des cases. Bien que cette procédure ne soit pas correcte (*cf.* paragraphe précédent), cette technique est retenue par de nombreux élèves...

En cherchant à résoudre une tâche de programmation simple, les élèves travaillent ainsi des compétences et savoirs mathématiques relatifs à la mesure. Réciproquement, la programmation informatique des déplacements du robot demande une capacité d'anticipation : pour un tunnel de mesure  $n$ , il faut anticiper et donc programmer  $n+1$  déplacements unitaires (en pas du robot).

**Pour la résolution de la deuxième tâche**, l'intention didactique est cette fois-ci principalement d'ordre informatique. Les connaissances et compétences construites lors de la résolution de la première situation-problème font milieu — didactique — pour la deuxième situation. Si la plupart des élèves repèrent rapidement le plan quadrillé

approprié 15 cm × 15 cm, certains utilisent le plan 15 cm × 12 cm et s'en servent comme d'une bande en ne retenant qu'un des côtés de la grille pour mesurer le tunnel.

La principale difficulté des élèves réside au niveau du repérage de la case sur laquelle le robot pivote. Le fait de proposer une « situation tunnel » où l'on ne peut agir à l'intérieur et non une « situation pont » où tout est visible oblige les élèves à se représenter un espace non accessible et à mobiliser l'outil « plan quadrillé » pour prévoir la case de pivotement. Les figures ci-dessous présentent différentes techniques

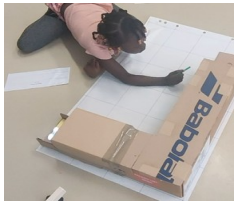


Figure 5



Figure 6



Figure 7



Figure 8



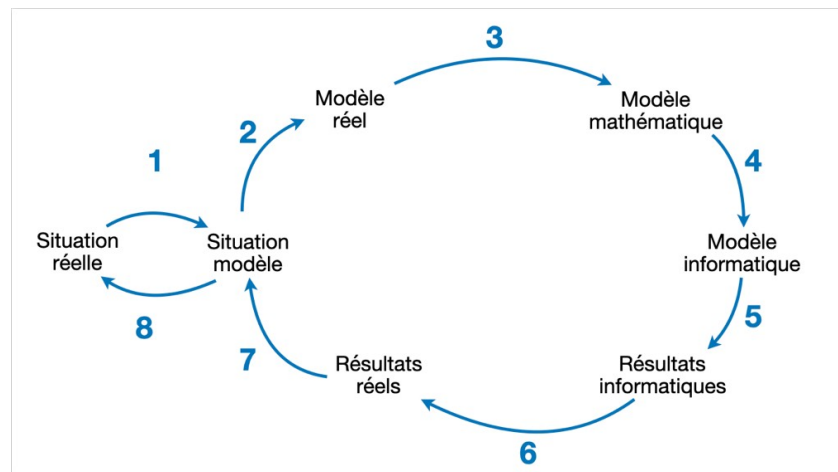
Figure 9

Les élèves positionnant le quadrillage sous ou du côté intérieur des tunnels (cf. fig. 5 et 6) rencontrent des difficultés à repérer la case de pivotement. Certains tentent de résoudre cette difficulté en positionnant un quadrillage sur le tunnel (cf. fig. 7), d'autres en positionnant le plan sur le côté extérieur du tunnel en se servant du plan comme d'une bande (cf. fig. 8), d'autres encore (cf. fig. 9) en prenant l'empreinte du tunnel sur le quadrillage.

### 1.3 Brève conclusion à propos des situations des tunnels

Lors de cette première partie de l'atelier, nous avons présenté ces deux situations afin de discuter de la manière dont chaque discipline contribue à la construction de savoirs. En mathématiques, les élèves ont réinvesti ou consolidé des savoirs et compétences sur le plan de la mesure, du repérage dans l'espace, tandis qu'en programmation informatique, les notions d'algorithme et de codage ont été initiées. La thématique du colloque portant sur la modélisation, nous nous sommes alors posé la question : est-il possible d'initier un travail de modélisation au cycle 2 du primaire ?

Le modèle de Blum et Leiss permet d'avancer une réponse positive : en suivant le cycle du modèle, nous avons cherché à expliciter chacune des étapes constituant le travail de modélisation.



**Figure 10** : cycle de modélisation de Blum et Leiss adapté à la situation des tunnels

La situation de départ est issue du monde réel. Il s'agit d'une rame de métro automatique. Celle-ci a été programmée pour qu'elle s'arrête en différentes stations sans intervention directe humaine.

L'étape 1 consiste à faire « comme si » Blue-Bot représentait un train automatique. Il s'agit alors d'anticiper et donc programmer le robot pour qu'il s'arrête automatiquement à la station d'arrivée.

La situation est ensuite simplifiée (étape 2) pour ne considérer que l'essentiel. Blue-Bot et tunnels

A l'étape 3, il s'agit d'associer le quadrillage idoine aux déplacements du robot, ce que l'on peut voir comme une première activité de modélisation mathématique.

L'étape 4 demande de modéliser les déplacements par un codage (modélisation informatique).

S'en suit un traitement informatique (étape 5) fournissant le résultat sous la forme d'un programme.

Le programme informatique est testé à l'étape 6. L'étape 7 valide ou invalide le programme. L'invalidation conduit alors à un nouveau cycle modifiant le modèle mathématique ou informatique.

Dans la classe, comment ce travail de modélisation pourrait-il prendre corps ? Il nous a semblé qu'aller au-delà de l'explicitation par les élèves des procédures de résolution des problèmes en faisant exprimer et argumenter le choix des outils permettant de résoudre les situations pouvait être un premier pas vers un travail de modélisation.

## **2. ScratchJr et la pensée algorithmique. Une nouvelle représentation mentale au service de la compréhension du récit (D. Laval)**

Avec l'enseignement du numérique, certaines tâches impliquent plusieurs domaines, et pour chacun de ces domaines, nous avons un Espace de Travail (EA). Nous faisons alors le choix de prendre comme cadre théorique pour les analyses des tâches, celui des « Espaces de Travail Connectés » au sens de Lagrange et Laval (2019). Nous partons de

l'hypothèse que ce cadre peut permettre de rendre compte de la manière dont les connexions entre les EA vont donner du sens aux concepts impliqués. Par ailleurs, nous rappelons que l'enseignement de l'algorithmique et de la programmation, associé à la logique et au raisonnement, s'inscrit dans les objectifs du socle commun de connaissances, de compétences et de culture. Au regard de ces constats, lors de cette seconde partie, nous nous intéressons à une pratique de la pensée algorithmique et du codage, à travers la programmation sur Scratch Junior, dans le cadre d'un enseignement interdisciplinaire (littérature de jeunesse, logique, raisonnement, algorithmique, codage, programmation) avec la compréhension d'un conte au niveau du cycle 2 de l'enseignement primaire.

## 2.1. Ebauche d'une ingénierie didactique

Nous présentons, une ingénierie didactique expérimentée dans deux classes du cycle 2 : un double niveau CE1/CE2 et un CE2 dans deux écoles de la région parisienne. Elle se découpe en deux séquences. Une première où les élèves analysent le conte, *Le lion et les trois buffles* de Dhouïb et Angeli (2014), à l'aide du cadre de la morphologie du conte (Propp, 1922) et de la sémantique structurale de Greimas (1966). Cette séquence se situe dans le domaine de la littérature de jeunesse. Bien qu'elle soit évoquée lors de l'atelier, elle ne pas fait l'objet d'un temps de réflexion avec les participants de l'atelier. Nous faisons donc le choix de ne pas la présenter ici<sup>51</sup>. Lors de la seconde séquence se situant dans le domaine de l'informatique, les élèves transcrivent le conte en des algorithmes qu'ils implémentent ensuite dans l'environnement Scratch Junior, afin d'obtenir une animation. C'est cette séquence qui a fait l'objet d'une présentation lors de l'atelier.

### Le conte : Le lion et les trois buffles de Dhouïb et Angeli

Ce conte est issu d'une fable traditionnelle arabe. Il raconte l'histoire malheureuse de trois buffles. Trois buffles frères (fig. 11), un blanc, un noir et un jaune, forts de leur union, décident de partir à l'aventure. Les chemins de l'aventure n'étant pas sans danger, ils chassent une meute de chacals (fig. 12) puis finissent par se reposer sous un grand arbre où un lion accepte leur présence (fig. 13). Le temps passant, le lion sème la discorde parmi les buffles et isole à chaque fois l'un d'entre eux pour le manger.



**Figure 11** : Les trois buffles (Extrait du conte de Dhouïb et Angeli)



**Figure 12** : Les buffles chassent la meute de chacals (Extrait du conte de Dhouïb et Angeli)



**Figure 13** : Le lion sous le baobab (Extrait du conte de Dhouïb et Angeli)

### La séquence 2 autour du numérique

Cette séquence est découpée en trois séances. Dans la première, l'élève propose, à l'aide de sa mémoire et des illustrations (si elles existent) du conte, de raconter le récit avec ses

<sup>51</sup> Elle fera l'objet d'un futur article qui sera publié fin 2024, début 2025.

mots en mettant l'accent sur les différents protagonistes du conte (par la suite, nous parlerons de variables informatiques au format de « chaînes de caractères » ou de « listes ») et des lieux. Puis, au cours de la deuxième séance, l'élève reformule le récit en utilisant des structures informatiques : variables (les protagonistes), boucle « Répéter », codage, etc. Ici, nous nous situons dans un travail algorithmique « papier-crayon ». Lors de la troisième séance, l'élève dans le cadre d'une activité branchée, implémente les algorithmes « papier-crayon » construits à la séance 2, dans l'environnement Scratch Junior afin d'obtenir une animation et de la tester.

### **Le choix de l'environnement numérique Scratch Junior au niveau du cycle 2**

Nous situant au cycle 2, nous faisons le choix de l'environnement numérique Scratch Junior dont nous examinons son appropriation et sa pertinence pour l'apprentissage de l'algorithmique et de la programmation par les élèves, dans le cadre d'un enseignement transdisciplinaire. Initialement, l'environnement Scratch Junior est une application pour tablette<sup>52</sup>, accessible aux enfants de 5 à 8 ans. Cependant, depuis quelques années, il existe des émulateurs de cet environnement qui permettent une utilisation de Scratch Junior sur des ordinateurs<sup>53</sup>. Cet environnement dispose d'une interface graphique basée sur un langage visuel fonctionnant par des manipulations gestuelles de la part de son utilisateur. Cet environnement permet à de très jeunes enfants, débutant en informatique, de programmer des jeux, des histoires, etc., dans l'esprit d'une première approche de résolution de problèmes, en lien avec le raisonnement et la logique, en considérant le codage comme une autre forme d'alphabétisation (Komis, 2016). Scratch Junior permet une programmation de type événementiel. Le travail algorithmique et de programmation conduit l'élève à entrer dans le raisonnement et la logique, à rechercher des bugs, à développer des idées de projet. De nombreux concepts de programmation sont abordés : séquences, une première structure de boucle (Répéter), variables informatiques, etc.

### **2.2. Description de l'environnement Scratch Junior**

L'interface ergonomique de l'environnement Scratch Junior est d'une grande simplicité adaptée aux jeunes enfants. Elle est basée sur des aspects ludiques permettant à de jeunes élèves de programmer sous forme de codage. Pour cela, ils ont des instructions par blocs de programmation, regroupées en palettes de couleurs différentes. Grâce à ces différents blocs, Scratch Junior possède une structure syntaxique, au sens de « grammaire », constituée de trois composantes : les éléments de base avec les différents blocs de programmation ; les règles de composition de ces éléments en commandes ; les règles de composition des commandes en scripts. Cette syntaxe est simple d'utilisation pour les élèves qui n'ont pas nécessairement une bonne maîtrise de la lecture. Afin d'écrire les scripts des programmes, ils procèdent par des « Glisser » dans la zone de programmation de la page, des blocs choisis (fig. 14 à 19).

Les scripts obtenus sont construits alors en assemblant les différents blocs sous forme d'un puzzle permettant une représentation spatiale du script (programme) (fig. 22).

Par ailleurs, ces blocs traduisent plusieurs aspects de l'action : - différentes possibilités de déclenchements des programmes (fig. 14), - différents types de déplacements (fig. 15), - des allures modifiables (agrandir, rétrécir, disparaître, apparaître, bulle de texte) (fig. 16), - générer des sons (possibilité d'enregistrer des paroles) (fig. 17), - possibilité de répétitions indéfiniment et de changements de scènes (fig. 18), - utiliser des blocs de contrôles (le protagoniste peut avancer selon trois vitesses, on peut aussi stopper l'action,

<sup>52</sup> <http://www.scratchjr.org/>

<sup>53</sup> <https://jfo8000.github.io/ScratchJr-Desktop/>

mettre un temps de pause, utiliser une boucle Répéter) (fig. 19). L'élève voit ainsi de façon conviviale et simple les premiers concepts fondamentaux de l'algorithmique et de la programmation (fig. 20 et 21).

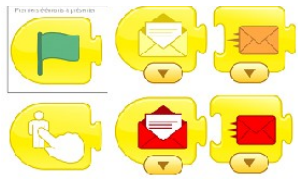


Figure 14 : Les blocs de déclenchements

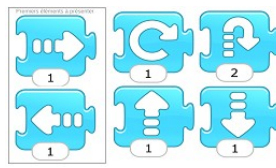


Figure 15 : Les blocs de déplacements



Figure 16 : Les blocs d'« allures »



Figure 17 : Les blocs de sons



Figure 368 : Les blocs Fin, Répéter indéfiniment, Changements de page



Figure 19 : Les blocs de contrôles



Figure 20 : Le chat et la poule d'après le travail d'un élève de CE2



Figure 21 : Principe d'une page de programmation



Figure 22 : Scripts de trois quadrilatères (Jaune, orange et marron) dans un champ

Comme le rapporte Komis (2016), « ScratchJr comporte [donc] plusieurs fonctionnalités sémantiques inscrites dans le paradigme de la programmation impérative ([...] séquence de commandes pour exécuter un mouvement dans l'espace ou le branchement non conditionnel ALLER sur une autre page) ». De même, d'autres paradigmes sont possibles : « la programmation multi-agent, étant donné que plusieurs personnages peuvent être programmés indépendamment et agir de manière concurrente » (Komis, 2016.). De plus, Scratch Junior permet d'accéder à « des rudiments de la programmation orientée-objets étant donné que le programmeur peut définir plusieurs personnages et leur interaction à partir des messages ou des événements » (Ibid.). Cet environnement permet aussi de déclencher une action d'un personnage ou d'un objet par le fait que celui-ci est touché par un autre personnage ou un autre objet. Ainsi, nous avons aussi une « programmation par événements » (Ibid.)

### 2.3. Découverte en amont de l'environnement Scratch Junior par les élèves

En amont de ces deux séquences, pendant cinq mois, à raison de 45 minutes par semaine, les élèves ont été familiarisés avec les différentes possibilités de codage et d'animations qu'offre l'environnement Scratch Junior. Pour cela, ils ont construit des petits programmes, comme :

Un déplacement dans l'espace (par exemple, une fusée décollant de la lune après que le spationaute eut reçu l'ordre de rentrer dans la fusée pour rentrer sur la Terre), avec reconnaissance de deux scènes. La première se situe sur la surface de la lune où la fusée stationne : le spationaute se déplace pour rentrer dans la fusée et décolle. Dans la seconde, nous avons le vol de la fusée dans l'espace en direction de la Terre (fig. 23 et 24).

Une transcription de la fable *Le Corbeau et le Renard* de Jean de La Fontaine, avec reconnaissance des différents protagonistes : personnages (corbeau, renard, narrateur), objet (fromage), lieu (un arbre avec une branche apparente) (fig. 25).

Création d'un personnage « abstrait » (avec des figures géométriques usuelles) qui se déplace dans l'espace d'un jardin (fig. 26).



**Figure 23 :** La surface de la lune avec le spationaute (Travail d'un binôme de CE1/CE2)



**Figure 24 :** Déplacement de la fusée dans l'espace (Travail d'un binôme de CE1/CE2)



**Figure 25 :** Le corbeau et le renard (Travail d'un élève de CE2)



**Figure 26 :** Promenade d'un personnage « abstrait » (Travail d'un élève de CE2)

#### Les séances 1 et 2 de la séquence informatique

Dans le cadre de cet article, nous présentons très brièvement le travail demandé aux élèves d'une classe de CE2 de Nanterre (92), lors de la séquence informatique, découpée en 2 séances de 45 minutes chacune.

Lors de la séance 1 (préparation/anticipation pour l'animation Scratch Junior), chaque élève résume d'abord le conte par étapes (sur le principe du schéma narratif), puis découpe le conte en huit parties et donne un titre pour chaque partie. Ensuite, il doit anticiper le nombre de paysages et leurs représentations sur Scratch Junior. De même, il anticipe le choix des formats des différents personnages sur Scratch Junior. Ainsi, les élèves proposent d'associer les personnages à différentes figures usuelles issues de la géométrie plane : buffles (un triangle de couleur pour chaque buffle, possibilité d'envisager les trois buffles réunis sous le principe d'une variable liste) ; meute de chacals (rectangle marron) et le lion (carré orange).

La seconde séance alterne différents champs : algorithmique, codage et utilisation de Scratch Junior. Pendant cette séance, les élèves sont répartis en deux groupes hétérogènes (14 élèves par groupe) et travaillent en salle informatique. Chaque élève a un ordinateur à sa disposition, avec Scratch Junior. L'observation porte sur le travail de médiation algorithmique du conte. Chaque élève transcrit le conte en algorithmes écrits en langage naturel, tenant compte de la logique du conte. Ensuite, les élèves passent au codage en l'implémentant dans Scratch Junior, afin de créer une animation et de la tester.

## 2.4. Cadres théoriques et méthodologie de recherche

### Utilisation de cadres théoriques

Dans la continuité des travaux de Minh et Lagrange (2016), de Lagrange (2018) et de Lagrange et Laval (2019 et 2022), nous nous référons à l'idée d'*Espaces de Travail Connectés*. En effet, nous élargissons cette approche aux *Espaces de Travail Algorithmique* (ETA) (Laval, 2018) qui seraient connectés à différentes méthodes d'analyse structurale (MAS) venant de cadres théoriques de la littérature. Ainsi, une analyse fine des données recueillies lors des observations des séquences en classe, est menée à l'aide d'un tel cadre théorique se situant aux connexions entre le modèle d'un ETA spécifique à Scratch Junior et différentes MAS, la morphologie du conte de Propp (1965) qui « construit [...] une grammaire des formes narratives qu'il présente comme des formules mathématiques » (Farahnak, 2010) et le schéma actantiel de Greimas (1986), où le modèle permet de déchiffrer la nature sémiolinguistique du récit. Nous supposons qu'il existe « une structure matricielle de l'histoire, fondée sur l'enchaînement logique des séquences » (ibid.). Ainsi, l'étude des connexions entre ETA et MAS donne du sens au travail des élèves dans des tâches impliquant différents domaines : logique, algorithmique, programmation et littérature de jeunesse, en tenant compte des genèses sémiotique, instrumentale et discursive, et de la structure narrative du conte.

### Espaces de travail algorithmique

Nous référant aux *Espaces de Travail Mathématiques* (Kuzniak, 2011, Kuzniak et Richard, 2014), nous adaptons une structure semblable aux ETA, c'est-à-dire avec les trois dimensions suivantes :

Sémiotique, avec une utilisation de symboles, de graphiques et d'objets concrets compris comme des signes.

Instrumental, avec la construction et l'expérimentation de modèles représentant les relations entre objets dans une configuration donnée (figures géométriques, graphiques, programmes, etc.). Ici, des artefacts vont être utilisés.

Discursif, avec la justification et l'idée de preuve (validité de l'algorithme) à l'aide d'un cadre de référence théorique (programmation sur Scratch Junior afin de tester la robustesse de l'algorithme).

Au niveau du primaire, nous restons pour l'essentiel sur les deux premières dimensions : sémiotique et instrumental. La dimension sémiotique dans l'ETA se caractérise par une notation proche des systèmes de symboles mathématiques, même si les similitudes peuvent être souvent trompeuses (Lagrange et Laval, 2022). De même, l'affectation est spécifique à l'ETA, puisqu'elle est liée à l'idée de changement d'état. En effet, une valeur va être donnée à une variable qui remplace une valeur précédente. Pour plus d'information, nous renvoyons le lecteur aux travaux de Samurçay (1985) qui a constaté que ces différences entre variables informatiques et mathématiques prêtent à confusion pour des débutants en informatique. Quant à la dimension instrumentale d'un ETA, elle implique l'exécution de l'algorithme, soit sur une machine, soit mentalement avec l'aide du papier et du crayon. L'exécution n'implique pas l'accomplissement passif d'un programme, mais plutôt un processus de construction, où les implémentations sont exécutées avec des données sélectionnées, afin de réaliser une exploration empirique (Lagrange et Laval, 2022).

### 2.5. Nos attentes, nos hypothèses didactiques et question de recherche

Supposant que la transcription du conte et de sa logique en algorithmes simples favorise la mobilisation du langage et du raisonnement, mais aussi la mémorisation du récit et sa reformulation, nous attendons que l'élève élabore au « papier-crayon » des algorithmes basés sur la logique des structures algorithmiques, permettant une nouvelle représentation mentale du conte, et qu'il implémente le codage de l'algorithme dans Scratch Junior. Par ailleurs, nous supposons que l'élève schématise la compréhension du conte par une représentation sémiotico-instrumentale, favorisant l'exploration du conte. L'élève va au-delà d'une simple lecture du conte, en proposant une activité de programmation informatique. Ainsi, cette organisation implicite de conduites et de gestes forme un schème (Vergnaud, 2011). En effet, selon Vergnaud, les schèmes sont des structures mentales qui intègrent les actions et les connaissances. Dans ce cas, nous pouvons observer que l'utilisation d'une représentation sémiotico-instrumentale pour explorer le conte et créer une activité de programmation informatique crée une mémoire de l'action. Les gestes, les manipulations et les réflexions impliqués dans cette activité deviennent partie intégrante du schème de l'élève. L'élève ne se contente pas de lire passivement le conte, il prend une part active en interagissant avec le texte de manière dynamique. Il s'implique activement en explorant le texte, en l'analysant, en le décomposant, et en le manipulant pour créer quelque chose de nouveau, en l'occurrence le traduire en termes de langage de programmation. Cela encourage l'élève à interpréter le contenu du conte de manière critique, à en extraire des éléments clés et à les relier à d'autres connaissances ou compétences. De plus, cela favorise une meilleure maîtrise du langage en lui permettant de traduire les idées et les concepts du récit en termes de langage de programmation, ce qui nécessite une réflexion et une compréhension approfondies.

Faisant l'hypothèse que l'institution situe l'algorithmique dans une approche *outil* (au sens de Douady, 1986) avec des algorithmes pour donner sens à un certain nombre de concepts mathématiques, nous nous questionnons : Comment dépasser ce stade pour que l'algorithmique devienne un vrai *objet* (ibid.) d'apprentissage, quel que soit le domaine ?

## 2.6. Exemples de travaux algorithmiques faits dans la classe de CE2

Nous présentons brièvement les travaux des élèves Hugo et Constantin<sup>54</sup> de la classe de CE2 de Nanterre. Dans un premier temps, lors de la séance 1, chacun des élèves a découpé, le conte en huit « chapitres » : (1) Les trois buffles (représentés par des triangles de couleurs), rois de leur vallée ; (2) Le désir des trois buffles de partir à l’aventure ; (3) Rencontre avec la meute de chacals (représentée par un rectangle marron) ; (4) Arrivée des trois buffles dans la savane où se trouve un baobab ; (5) Rencontre avec le lion (représenté par un carré orange) au repos sous le baobab ; (6) Le lion sème la peur chez les buffles noir et jaune, le buffle blanc est mis à l’écart et le lion finit par le manger ; (7) Après un retour au calme sous le baobab, le lion évoque à nouveau sa faim et sème la jalousie chez le buffle noir qui s’éloigne de son frère le buffle jaune que le lion dévore ; (8) Après un nouveau temps de calme sous le baobab, le buffle noir prend conscience de la manipulation du lion pour manger ses deux frères. Se retrouvant seul, il se laisse alors dévorer par le lion. Puis, lors de la seconde séance, les élèves se sont réparti les chapitres pour le passage à la programmation sur Scratch Junior. Hugo a traité les 5 premiers chapitres (de la vallée à l’arrivée dans la savane et la rencontre avec le lion au repos) et Constantin les 3 derniers (la mort du buffle blanc suivie d’un temps calme avec ensuite la mort du buffle jaune et pour terminer, celle du buffle noir). Le choix de cette répartition est laissé aux enfants, avec l’accord de l’enseignante. Celle-ci parle de tableaux pour les scènes que les enfants définissent dans l’environnement Scratch Junior. Par exemple, nous avons les extraits<sup>55</sup> ci-dessous pour Hugo (fig. 27) et pour Constantin (fig. 28).



Figure 27 : Arrivée dans la savane et rencontre avec le lion.

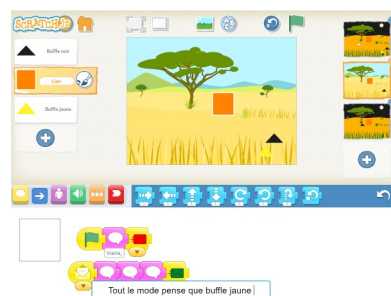


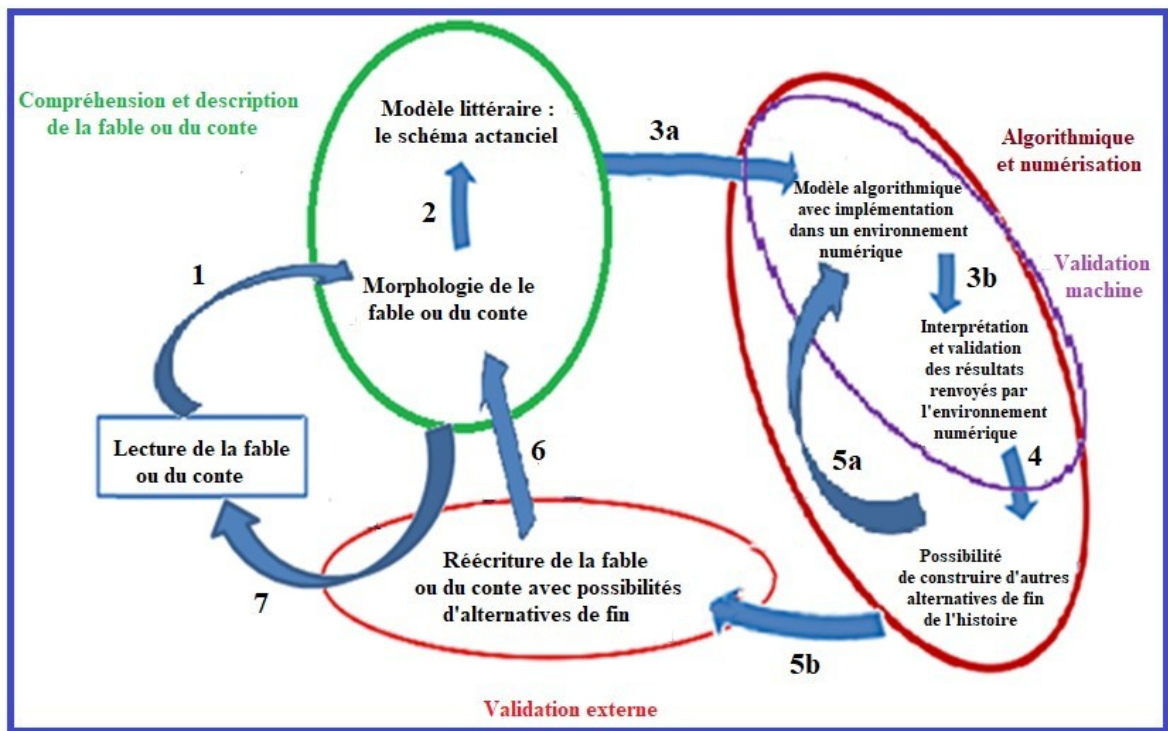
Figure 28 : Extrait où le lion mange le buffle jaune après avoir semé la jalousie dans l’esprit du buffle noir

## 2.7 Brève conclusion

Lors de cette seconde partie de l’atelier, nous avons pu observer, à travers une nouvelle médiation, que l’élève modifie son expérience et sa relation au conte et à l’algorithmique. En effet, un modèle de transcription algorithmique peut s’apparenter à une nouvelle forme d’adaptation du conte, après celles du théâtre jeunesse ou du dessin animé. Cette nouvelle perspective permet d’aborder une ouverture de l’algorithmique au-delà du seul domaine des mathématiques. Cela justifie l’algorithmique comme *objet* de savoir à part entière dès le début de la scolarité de l’élève. Par ailleurs, nous situons notre approche du modèle de transcription algorithmique comme pouvant être une adaptation du modèle de Blum et Leiss (2007) aux spécificités de l’algorithmique connectée à la compréhension du récit en littérature de jeunesse (fig. 29).

<sup>54</sup> N.D.L.R. : Les prénoms des élèves ont été changés

<sup>55</sup> N.D.L.R. : Les textes des animaux sont tels que les enfants les ont écrits, d’où la possibilité de fautes d’orthographe.



**Figure 29** : Cycle de modélisation avec un modèle littéraire de type algorithmique  
Ainsi, sur ce schéma, nous pouvons observer les différentes étapes suivantes :

- (1) La compréhension du texte.
- (2) La construction d'un modèle littéraire.
- (3) La transcription numérisée avec reconnaissance des personnages, des objets, du (ou des) lieu(x).
- (3a) Modèle algorithmique et implémentation dans un environnement numérique.
- (3b) Interprétation et validation des résultats renvoyés par l'animation numérisée.
- (4) La possibilité de construire des fins alternatives à l'histoire.
- (5) Une réécriture de la fable ou du conte.
- (5a) Choix de modifier le modèle algorithmique initial.
- (5b) Réécriture au « papier-crayon » de l'histoire.
- (6) Une interprétation.
- (7) La lecture.

### Références bibliographiques

Blum, W., Leiss D. (2007). How Do Students and Teachers Deal with Mathematical Modelling Problems? The Example Sugaloaf und the DISUM Project. In C. Haines, P. L. Galbraith, W. Blum, & S. Khan (Eds.), *Mathematical Modelling (ICTMA12)-Education, Engineering and Economics*. Chichester: Horwood.

- Couderette, M. (sous presse). *Introduction de l'informatique au primaire: quelle organisation didactique ?* Dans Y. Buyck, M. Sudriès, F. Ligozat et C. Marlot (Eds.). *Les didactiques face à l'évolution des curriculums. Savoir(s) et pratiques pour entrer dans la complexité du monde.* Actes du 6ème Colloque international de l'ARCD. Université de Genève.
- Dhouib, M., & Angeli, M. (2014). *Le lion et les trois buffles*. Seuil Jeunesse.
- Douady, R. (1986). Jeux de cadres et dialectique outil–objet. *Recherches en didactique des Mathématiques*, 7.2, 5–31. Editions La Pensée Sauvage.
- Farahnak, N. (2010). Analyse structurale de deux contes de Perrault selon les schémas de Propp, Larivaille et Greimas. *Etudes de Langue et Littérature Françaises*, 2(1), 5-15.
- Greimas, A.-J. (1986). *Sémantique structurale*. Formes sémiotiques. Puf.
- Jouve, V. (2006). Les métamorphoses de la lecture narrative. *Protée*, 34(2-3), 153–161. <https://doi.org/10.7202/014273ar>
- Komis, V. (2016). Une analyse cognitive et didactique du langage de programmation *Scratch Jr*. Présenté à Didapro6-DidaSTIC, Namur, Belgique. <https://docplayer.fr/57266243-Une-analyse-cognitive-et-didactique-du-langage-de-programmation-scratchjr.html>
- Kuzniak, A. (2011). L'espace de Travail Mathématique et ses genèses. *Annales de Didactiques et de Sciences Cognitives*, 16, 9-24. <https://hal.archives-ouvertes.fr/halshs-01060043>
- Kuzniak, A., Richard, R. P. (2014). Spaces for mathematical work: Viewpoints and perspectives. *Relime*, 17(4.1), 17–26.
- Lagrange, J.-B., Laval, D. (2019). *Connected Working Spaces: The case of computer programming in Mathematics Education*. Eleventh Congress of the European Society for Research in Mathematics Education, Utrecht, Netherlands.
- Lagrange, J.-B., Laval, D. (2022). Connecting algorithmics to mathematics learning: a design study of the intermediate value theorem and the bisection algorithm. *Educ Stud Math* 112, 225–245. <https://doi.org/10.1007/s10649-022-10192-y>
- Larivaille, P. (1974). L'analyse (morpho)logique du récit. *Revue Poétique*, n° 19, pp. 368-388.
- Laval, D. (2018). *L'algorithmique au lycée entre développement de savoirs spécifiques et usage dans différents domaines mathématiques* (Thèse). Université Sorbonne Paris Cité. <https://hal.archives-ouvertes.fr/tel-01943971>
- Minh, T.K., Lagrange, J. B. (2016) Connected functional working spaces: a framework for the teaching and learning of functions at upper secondary level. *ZDM Mathematics Education*, 48(6), pp. 793–807.
- MEN (2016). Initiation à la programmation au cycle 2 et 3. Document Eduscol consulté en ligne <https://eduscol.education.fr/document/15409/download>
- Propp, V. (1965). *Morphologie du conte*, Seuil, Paris, 1965.
- Samurçay, R. (1985). *Signification et fonctionnement du concept de variable informatique chez des élèves débutants*. *Educationnal Studies in Mathematic*, n° 16-2, pp. 143-161.
- Vergnaud, G. (2013). *Pourquoi la théorie des champs conceptuels ?* (Vol. 36, Numéro 2). Informa UK Limited. <https://doi.org/10.1174/021037013806196283>