



**HAL**  
open science

# Organiser l'apprentissage de la programmation au cycle 3 avec des activités guidées et/ou créatives

Christophe Declercq, Françoise Tort

## ► To cite this version:

Christophe Declercq, Françoise Tort. Organiser l'apprentissage de la programmation au cycle 3 avec des activités guidées et/ou créatives. RJC EIAH 2018, Apr 2018, Besançon, France. hal-01765408

**HAL Id: hal-01765408**

**<https://hal.science/hal-01765408v1>**

Submitted on 12 Apr 2018

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Organiser l'apprentissage de la programmation au cycle 3 avec des activités guidées et/ou créatives

Christophe Declercq<sup>1</sup>, Françoise Tort<sup>2</sup>

<sup>1</sup> Espe de l'académie de Nantes, CREN  
Christophe.Declercq@univ-nantes.fr

<sup>2</sup> ENS Paris-Saclay, STEF  
Francoise.Tort@ens-cachan.fr

**Résumé.** On analyse l'activité des élèves dans deux types de situations d'apprentissage de la programmation au cycle 3 : des situations inspirées des documents d'accompagnement et des situations issues du concours Castor. On en déduit une typologie des comportements des élèves ainsi que des indications pour l'enseignant. On présente ensuite un micro-monde permettant une hybridation entre les deux types de situations présentées au début. On s'attache alors en particulier à décrire et analyser les possibilités d'organisation, de retour instrumental et de suivi des activités des élèves dans ce nouvel environnement.

**Mots-clés.** didactique de l'informatique, scénario pédagogique, micro-mondes, informatique créative, pensée informatique.

## 1 Introduction

En France, depuis la rentrée 2016, l'enseignement de l'informatique est inscrit dans les programmes des cycles 2, 3 et 4.

Au cycle 3, le programme précise : "Une initiation à la programmation est faite à l'occasion notamment d'activités de repérage ou de déplacement (programmer les déplacements d'un robot ou ceux d'un personnage sur un écran), ou d'activités géométriques (construction de figures simples ou de figures composées de figures simples)".

Les documents d'accompagnement [1] proposent, entre autres, des activités utilisant l'environnement Scratch [2] dont le scénario pédagogique se caractérise par une activité de programmation semi-guidée encadrée par l'enseignant. L'environnement de programmation utilisé est ouvert et permet des activités de programmation créative.

Par ailleurs, depuis 2011, de nombreux élèves de cycle 3 participent au concours Castor [3,4] et poursuivent avec le concours Algorea. Ils résolvent des défis de programmation dans des micro-mondes. Le scénario pédagogique est différent : l'élève avance seul à son rythme selon une progression pré-définie de défis.

Après l'étude exploratoire [3] sur la possibilité de considérer les ressources du concours comme des ressources pédagogiques pour l'enseignement, nous avons,

dans ce travail, voulu expérimenter des variations autour de situations du concours qui avaient été identifiées comme particulièrement constructives par rapport aux notions au programme du cycle 3 - principalement la séquence et la boucle.

Notre problématique consiste à analyser l'impact du choix de l'environnement d'apprentissage de la programmation, sur l'activité des élèves et sur la nécessité ou non d'un guidage par l'enseignant. La contribution principale consiste à proposer un micro-monde inspiré d'une situation du concours Castor, dédié à l'apprentissage de la séquence et de la boucle.

Pour étudier l'intérêt de notre proposition, nous avons d'abord analysé l'activité des élèves dans les situations de référence au programme du cycle 3, et avons alors identifié différentes postures possibles pour l'élève dont une - celle de programmeur - vers laquelle notre proposition tend à guider les élèves.

Nous présentons successivement, les situations de référence étudiées, leur analyse a priori, puis notre proposition et les modalités prévues pour en étudier l'usage en classe.

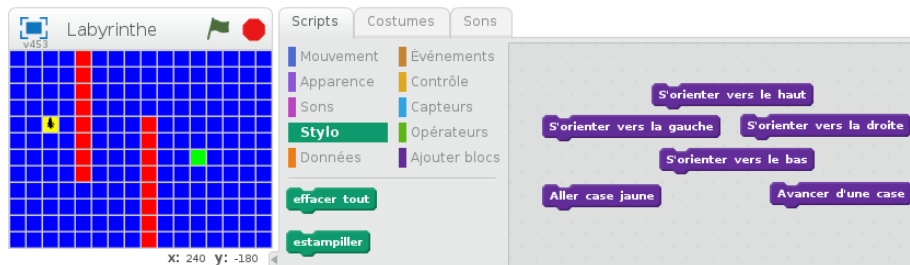
## 2 Situations didactiques et environnements interactifs

Dans les documents d'accompagnement du cycle 3, les situations de programmation avec Scratch sont principalement de deux types : déplacement d'un personnage sur une grille pour aller d'un point à un autre et reproduction d'une figure géométrique par un traceur. Elles visent à faire découvrir les instructions de déplacement, la notion de séquence puis à introduire la répétition bornée.

### 2.1 Programmer le déplacement d'un personnage sur une grille

On s'intéresse tout d'abord à la situation du déplacement sur une grille. Nous avons construit une situation analogue à celle du document d'accompagnement avec une image de labyrinthe en fond d'écran et des blocs préprogrammés de déplacements absolus : avancer d'une case, s'orienter vers le haut / le bas / la droite / la gauche.

FIGURE 1. Activité de déplacement sur une grille



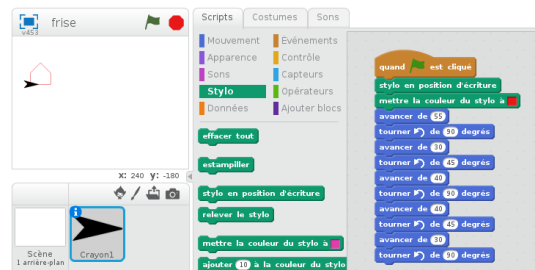
La tâche de l'élève consiste à programmer le déplacement du personnage - ici un insecte - d'une position de départ - la case jaune - vers une position d'arrivée - la case verte.

Le document d'accompagnement précise bien l'objectif de ce type d'activité : "Déplacer un personnage sur un écran peut sembler être une activité déjà beaucoup pratiquée par de nombreux élèves lorsqu'ils jouent à des jeux vidéo, mais dans ces jeux les élèves déplacent en général des personnages par petites touches successives, en procédant à des micro-déplacements successifs pour obtenir le déplacement voulu. L'objectif est ici différent car il s'agit d'anticiper un déplacement complet, de prévoir à l'avance l'ensemble des instructions permettant d'obtenir un déplacement complexe souhaité."

## 2.2 Répéter un motif élémentaire pour dessiner une frise

De la même manière, nous avons proposé une activité de dessin, permettant à l'élève de s'appropriier les instructions élémentaires manipulant le crayon, en lui fournissant un programme effectuant un dessin élémentaire - une maison - et en lui confiant la tâche de faire une frise avec ce motif.

FIGURE 2. Activité de dessin élémentaire



L'objectif de l'activité est d'amener l'élève à réutiliser le programme du motif simple, en le faisant répéter le nombre de fois nécessaire. Remarquons qu'il faut aussi penser à ajouter, à la fin du dessin du motif, un déplacement positionnant le personnage au point de départ du motif suivant.

## 2.3 Reproduire une frise en identifiant le motif à répéter

Au concours Castor, le défi-type visant à faire utiliser une répétition bornée est celui du robot-peintre devant reproduire une frise. La frise à reproduire est affichée, les seules instructions disponibles sont les instructions de déplacement, il est annoncé que le programme sera répété, et le nombre de répétitions est indiqué. Le jeu d'instructions est volontairement simplifié : déplacements absolus d'un pas à chaque déplacement.

FIGURE 3. Le robot peintre du concours Castor

Pour reproduire le dessin objectif, construisez une séquence de commandes. Elle sera répétée 6 fois.

**Exemple**  
(la séquence est répétée 6 fois)

1 pas à droite  
1 pas à droite  
1 pas en haut  
Essayer

**Objectif**

Prenez les instructions ici

1 pas à droite  
1 pas à gauche  
1 pas en haut  
1 pas en bas

Glissez les instructions là

Résultat de l'exécution  
(la séquence est répétée 6 fois)

Pause Arrêter

L'objectif ici est de centrer l'activité de l'élève sur le repérage du motif à répéter. Cette activité permet de développer une compétence de généralisation, dans la mesure où le motif élémentaire n'est pas fourni et n'est pas particulièrement facile à deviner.

### 3 Analyse a priori de l'activité des élèves

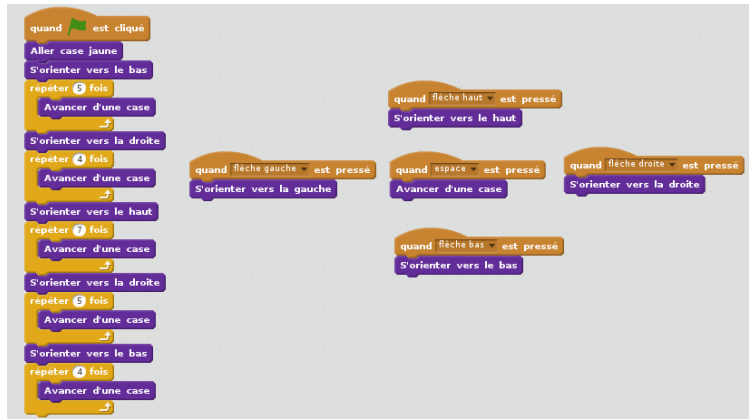
#### 3.1 Activité de déplacement : programmer ou télécommander ?

Dans l'activité de déplacement proposée en 2.1, l'appropriation la plus immédiate par les élèves consiste à tester les blocs de déplacement en cliquant dessus pour déplacer immédiatement le personnage. Ce comportement est induit par le pré-positionnement des blocs utiles sur l'espace de travail.

Un second type d'activité de l'élève consiste à tester les blocs un par un puis à les accrocher dans l'ordre pour construire la séquence. Ce mode de fonctionnement est facilité par Scratch : le retour instrumental permet à cette tâche d'être effectuée sans nécessiter de compétence d'anticipation, mais produit cependant un programme effectuant le déplacement voulu. Ce programme peut cependant ne pas avoir un comportement reproductible, si le positionnement à un état initial a été omis, ce qui est courant dans ce type de programmation par l'exemple.

La tâche prescrite par l'enseignant consiste à anticiper l'ensemble du déplacement en le programmant et en prévoyant aussi le départ à la position initiale. Pour y parvenir, le guidage de l'enseignant peut consister à demander que le programme puisse s'exécuter, plusieurs fois, depuis l'interface plein écran en cliquant sur le drapeau vert.

FIGURE 4. Programmer un déplacement / programmer la télécommande



Un dernier type de comportement peut alors être observé avec des élèves déjà initiés à la programmation événementielle en Scratch : en affectant à chacun des événements déclenchés par les flèches du clavier, l'action de déplacement correspondante, il est facile de programmer la télécommande du personnage, de passer en mode plein écran, puis de guider le personnage avec les flèches, sans avoir écrit le programme de déplacement !

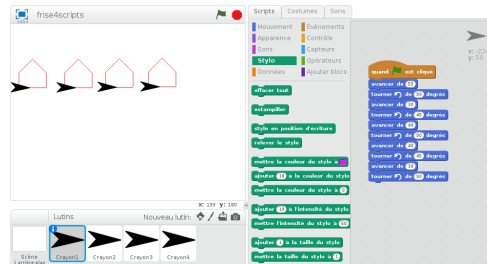
### 3.2 Activité de dessin : répéter ou dupliquer

Pour s'appropriier le programme fourni par l'enseignant, l'élève peut exécuter une 1ere fois le programme, déplacer le curseur interactivement avec la souris, puis l'exécuter à nouveau : il peut ainsi construire la frise demandée en utilisant le programme fourni en mode "télécommande". La répétition est effectuée par le geste mais n'est pas programmée.

La tâche souhaitée par l'enseignant consiste à faire programmer la répétition. Cependant il est aussi possible pour l'élève de copier quatre fois le personnage, de placer chaque personnage au point de démarrage de chacune des maisons, puis de déclencher l'exécution. En effet, la copie d'un personnage crée un personnage analogue disposant du même script : à l'exécution les scripts s'exécuteront en parallèle. On obtient la frise désirée, sans avoir recours à une structure de répétition.

Dans cet exercice, afin de guider les élèves sur la piste de la répétition, il est nécessaire de contraindre la solution en ajoutant au moins deux contraintes : ne devoir appuyer qu'une seule fois sur le drapeau vert, et ne devoir utiliser qu'un seul personnage.

FIGURE 5. Programmer une frise par duplication



### 3.3 Les postures de l'élève

En analysant les situations d'utilisation de l'environnement Scratch, on distingue 4 postures possibles de l'élève :

- l'**utilisateur de télécommande**, effectue des manipulations directes, mais ne programme pas ;
- le **programmeur pas à pas**, peut réussir à écrire un programme simple mais ne développe pas de compétence d'anticipation ;
- le **programmeur de télécommande** est dans une posture de programmeur mais reporte la définition de la stratégie de résolution du problème, sur le futur utilisateur de la télécommande ;
- le **programmeur** est dans le rôle voulu pour l'élève par l'enseignant.

Ces 4 postures construites dans la résolution d'une même situation, expliquent pourquoi le guidage par l'enseignant est indispensable, s'il veut engager ses élèves dans une véritable activité de programmation.

Les situations du concours Castor ne nécessitent pas de guidage par l'enseignant et induisent nécessairement une posture de programmeur. Elle ont été conçues spécifiquement pour cet usage, en contexte de concours. Mais pour une utilisation en classe, la situation générique de reproduction de motifs mérite d'être adaptée avec une progression de motifs adaptés au niveau du cycle concerné.

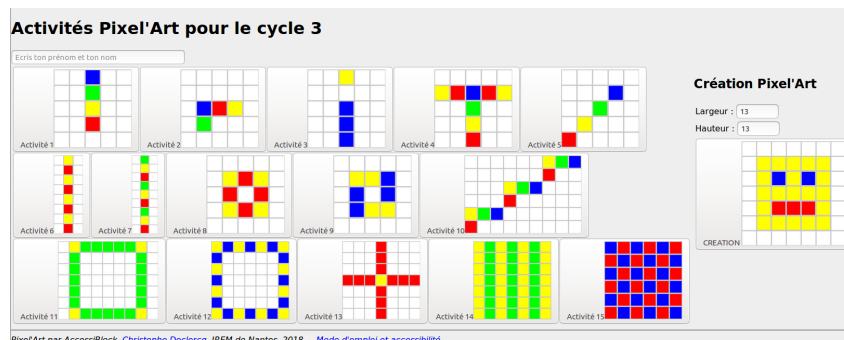
## 4 Proposition de séquences intégrant des défis et des activités créatives

### 4.1 Du robot peintre au pixel'art

Pour décliner la situation du robot peintre en différentes versions et la prolonger par une activité créative qui permette à l'élève de conserver la trace de son oeuvre, on propose une activité de **Pixel'Art** en utilisant la bibliothèque AccessiBlock [5] - variante accessible de la bibliothèque Blockly de programmation par blocs [6].

L'application web PixelArt3, développée en javascript côté client, est diffusée sous licence libre et est disponible au téléchargement à l'adresse : <https://gitlab.univ-nantes.fr/declercq-c/PixelArt3>

FIGURE 6. Activités Pixel'Art pour le cycle 3



L'organisation proposée ici comprend 5 activités autour de la séquence, puis 5 activités conçues pour amener progressivement la nécessité de la répétition bornée. Les 5 activités suivantes peuvent être traitées avec l'usage de répétitions imbriquées. L'activité créative est disponible à tout moment de la progression : elle utilise les mêmes primitives de programmation, et se distingue seulement des activités guidées par le fait qu'elle ne comporte pas de motif à reproduire et que l'élève a le choix de la taille de son dessin. Cette activité est destinée à permettre le réinvestissement des notions abordées pendant les activités guidées, dans un mode où l'élève a plus de liberté au niveau du dessin à réaliser.

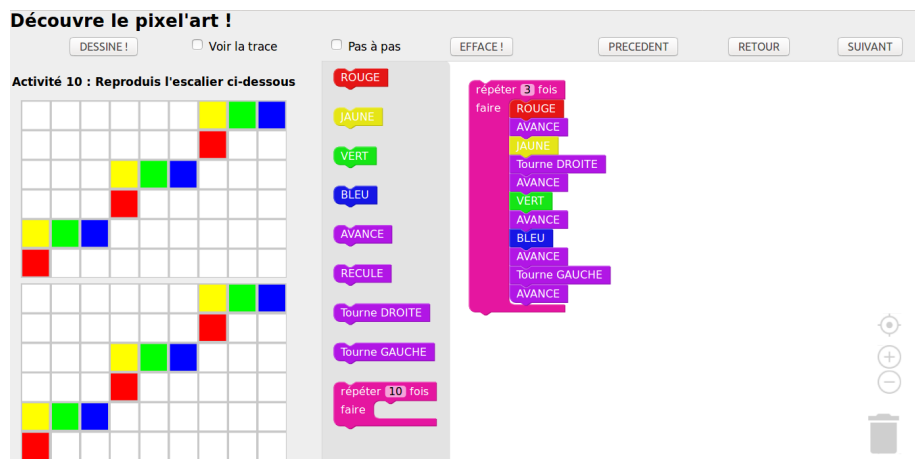
L'enseignant peut aussi définir son scénario en dessinant lui-même - avec le mode créatif - les motifs qu'il souhaite donner à reproduire aux élèves. Puis il inclut dans son scénario, la progression souhaitée d'activités de reproduction et d'activités créatives. Pour cela il doit simplement éditer un fichier nommé SCENARIO et y noter pour chaque activité : la phrase consigne, le nom du fichier contenant l'image à reproduire, la taille de la grille à utiliser et les coordonnées de départ.

Par rapport au robot peintre, on a conservé un jeu d'instructions de déplacement mais en choisissant des déplacements relatifs, et on a ajouté les couleurs. La répétition figure explicitement en tant que bloc de programmation.

Dans cette activité, le motif à reproduire - un escalier - est fourni. L'élève dispose des instructions permettant de peindre la case courante en rouge, jaune, vert ou bleu, d'avancer ou de reculer le curseur par rapport à la direction courante et de tourner à droite ou à gauche (à angle droit). Le bloc **répéter n fois** permet de programmer une répétition bornée sans possibilité d'utiliser une variable. Le nombre de répétitions est une constante à saisir dans le programme.



FIGURE 7. Activité de reproduction d'un motif



La figure 7 illustre l'activité terminée, on peut comparer le motif à reproduire - en haut à gauche - et le motif effectivement dessiné - en bas à gauche - par le programme saisi par l'élève - à droite. Les blocs disponibles sont visibles dans le bandeau central.

#### 4.2 Interaction, retour instrumental et interface

Pour amener systématiquement l'élève dans la posture du programmeur, l'interface ne lui permet que d'exécuter la totalité du programme saisi, qu'il peut cependant construire progressivement. A chaque exécution, le dessin est effacé et le curseur est repositionné à sa position de départ, ce qui permet d'assurer que - contrairement à Scratch - toute exécution d'un même programme aura toujours le même effet.

Pour procurer à l'élève un retour instrumental efficace, deux options sont disponibles : le mode **trace** permet de visualiser les positions et orientations successives du curseur ; le mode **pas à pas** permet d'exécuter instruction par instruction en faisant afficher le dessin au fur et à mesure de l'exécution, avec l'instruction courante qui est affichée en surbrillance. Les deux modes peuvent être combinés pour obtenir une trace et une exécution pas à pas. Ceci est particulièrement utile pour comprendre l'exécution de programmes superposant plusieurs couleurs sur la même case, ou dont le déplacement sort du cadre. L'exécution instantanée du programme entier ne permettrait pas de donner un retour expliquant ces deux phénomènes.

L'enregistrement des traces de l'activité, estampillées de la date et de l'heure, est effectué systématiquement par l'application qui conserve, le temps de la session, chaque début et fin d'activité, chaque création ou destruction de bloc et

chaque tentative d'exécution. L'enseignant ou le chercheur, qui souhaite analyser ces traces, peut les conserver.

### 4.3 Expérimentations en classe et recherches en cours

Plusieurs expérimentations sont menées actuellement en cycle 3.

Les premières utilisations en classe ont été effectuées par les participants du groupe IREM algorithmes et programmation de Nantes : quatre enseignants de mathématiques ont ainsi testé les activités avec leurs classes de sixième et ont contribué à améliorer l'interface en signalant les difficultés rencontrées par les élèves.

Une étude exploratoire a été menée par un professeur stagiaire avec une classe de CM1 de 22 élèves au premier trimestre 2018. Les élèves, travaillant en binôme, ont utilisé PixelArt pendant une séquence de 4 séances. Les enregistrements des traces d'activité dans PixelArt sont en cours d'analyse.

Un second recueil de données plus important a été effectué en collaboration avec un groupe d'enseignants expérimentés, accompagnés en formation par leur conseiller pédagogique qui participe à cette recherche. Il porte sur deux types de situations : les binômes travaillant sur les ordinateurs, les élèves et l'enseignant interagissant au sujet de la résolution du problème posé. Les données recueillies - traces d'activité et vidéos - alimentent des travaux en cours cherchant à étudier en particulier la genèse chez l'élève du concept de boucle.

## 5 Conclusions et perspectives

Ces premiers recueils de données montrent déjà que le scénario proposé est opérant et que le retour instrumental produit par la trace et le pas à pas compensent l'absence d'exécution bloc par bloc, fonctionnalité que nous avons écartée volontairement pour éviter des postures de télécommande. La réinitialisation systématique a aussi été appréciée - en particulier par les enseignants - pour qui la non répétabilité des exécutions d'un programme Scratch constituait un obstacle complexifiant leurs tentatives d'institutionnalisation.

La possibilité d'activité créative a été plébiscitée par les élèves qui, une fois les défis terminés, occupent tout le temps disponible en donnant libre cours à leur imagination.

L'analyse quantitative de toutes les données recueillies est en cours. Les métriques d'analyse des traces sont en cours d'élaboration.

Une autre perspective que nous étudions actuellement concerne le cycle 4. L'analyse comparée de situations du concours Castor - la course de grenouilles - impliquant une synchronisation entre 2 activités, et de situations utilisant Scratch avec des scripts déclenchés par des événements et des activités en parallèle pourrait nous conduire à la définition d'un micro-monde mêlant défis et activités créatives autour de la programmation de jeux, à la manière de [7].

Nous pourrions alors envisager de comparer aussi au cycle 4, l'approche hybride proposée ici, avec l'approche de programmation créative proposée en particulier en France par [8].

## Références

1. Ministère de l'Education Nationale, [eduscol.education.fr/ressources-2016](http://eduscol.education.fr/ressources-2016) : Initiation à la programmation aux cycles 2 et 3 (2016)
2. Maloney, J., Resnick, M., Rusk, N., Silverman, B., Eastmond, E. : The Scratch programming language and environment. *ACM Trans. Comput. Educ.* **10(4)** (2010)
3. Drot-Delange, B., Tort, F. : Concours Castor, ressource pédagogique pour l'enseignement de l'informatique? Etude exploratoire auprès d'enseignants. In : *Didapro 7 - DidaSTIC*. (février 2018)
4. Drot-Delange, B., Tort, F. : Résolution de défis et pensée informatique. In : *10e rencontres scientifiques de l'ARDiST*. (mars 2018)
5. Declercq, C. : AccessiBlock : propositions pour un environnement accessible et observable d'apprentissage de la programmation pour l'école élémentaire et le collège. In : *Didapro 7 - DidaSTIC*. (février 2018)
6. Pasternak, E., Fenichel, R., Marshall, A.N. : Tips for creating a block language with Blockly. In : *IEEE Blocks and Beyond Workshop*. (2017)
7. Reffay, C., Dadeau, F., Follet-Locatelli, B., Michaud, P., Greffier, F. : Apprentissage de la programmation en cycle 2 avec un jeu vidéo collaboratif. In : *EIAH 2017*. (2017)
8. Romero, M. : De l'apprentissage procédural de la programmation à l'intégration interdisciplinaire de la programmation créative. *Formation et profession* **24(1)** (2016) 87–89