

Introduction d'éléments d'informatique dans l'enseignement des mathématiques, défis et opportunités

Le cas d'une pratique ordinaire en classe
troisième

Simon Modeste

University of Montpellier, France

simon.modeste@umontpellier.fr

Contexte

- **Table ronde Mathematics Education incorporating Coding: Practical Challenges and Opportunities**
 - Symposium « Coding, Computational Modeling, and Equity in Mathematics Education » à Brock University (Ontario, Canada) 27-29 avril 2023
Celia Hoyles (Chair), George Gadanidis, Oh Nam Kwon, Simon Modeste, Elena Prieto-Rodriguez
- **Handbook of Digital Resources in Mathematics Education, ed: Prof. Dr. Birgit Pepin, Ghislaine Gueudet, Jeffrey Choppin,**
 - Section : How Digital Resources Transform Content, Michal Tabach, Jana Trgalova**
 - Computational thinking and mathematics
Laura Broley, Chantal Buteau, Simon Modeste, Maryna Rafalska, Max Stephens

Informatique et pensée informatique en mathématiques, en France et à l'étranger

- **Pisa 2022 (OCDE)**

12. The increasing and evolving role of computers and computing tools in both day-to-day life and in mathematical literacy problem solving contexts is reflected in the recognition in the PISA 2022 framework that students should possess and be able to demonstrate computational thinking skills as they apply to mathematics as part of their problem-solving practice. Computational thinking skills include pattern recognition, designing and using abstraction, pattern decomposition, determining which (if any) computing tools could be employed in analysing or solving a problem, and defining algorithms as part of a detailed solution. By foregrounding the importance of computational thinking as it applies to mathematics, the framework anticipates a reflection by participating countries on the role of computational thinking in mathematics curricula and pedagogy.

Informatique et pensée informatique en mathématiques, en France et à l'étranger

- **Ontario (Canada)**

- Grades 1-9 : « pensée computationnelle » intégrée au curriculum de mathématiques, il est attendu que les élèves soient capables d'apprendre et d'utiliser la programmation au sein de la partie algèbre.

« The thought process involved in expressing problems in such a way that their solutions can be reached using computational steps and algorithms »

- Grades 10-12 : enseignements d'informatique, technologie, et programmation (+ changements à venir)

Informatique et pensée informatique en mathématiques, en France et à l'étranger

- **Australie**

- Introduction de la « pensée computationnelle » en mathématiques, présentée comme une des nombreuses démarches clés dans l'enseignement et l'apprentissage des mathématiques.

« The capacity to purposefully select and effectively use the functionality of a digital device, platform, software or digital resource is a key aspect of Computational Thinking in the Mathematics curriculum. ... Different digital tools or platforms can carry out computations and implement algorithms using numerical, textual, statistical, probabilistic, financial, measurement, geometrical, graphical, logical and symbolic functionalities. »

“Programmation” dans le curriculum en France (en maths et en info.)

Grades	Niv. scol.	Organisation		Langage
1-5	École	Mathématiques Informatique (un seul enseignant)		Scratch Jr / Robots
6-9	Collège	Informatique (partagé entre)	Mathématiques ("algorithmique et programmation")	Scratch
			Technologie	Scratch / Robots
10-12	Lycée	Informatique (spécialité NSI)		Python & autres
		Mathématiques ("Algorithmique et programmation")		Python

Curriculums prescrits, implémentés... et atteints

- **Implémentation des curriculums**

- Réalité du matériel et contraintes de temps
- Diversité des expériences des enseignants avec l'informatique et la programmation, et besoin de formation continue
- Gérer l'équilibre et les liens entre enjeux d'apprentissage informatiques et mathématiques

- **Curriculums atteints**

- Hétérogénéité des pratiques enseignantes (incluant ne pas enseigner l'algorithmique et la programmation)
- Objectifs d'apprentissages difficiles à atteindre (fin du cycle 3, du cycle 4, du lycée)
- Importante hétérogénéité d'expérience de connaissances pour les élèves à la fin de la scolarité obligatoire

Collège – cycle 4 (grades 7-9)

Programmes de mathématiques – 5 Thèmes

Thème E – Algorithmique et programmation

Au cycle 4, les élèves s'initient à la programmation, en développant dans une démarche de projet quelques programmes simples, sans viser une connaissance experte et exhaustive d'un langage ou d'un logiciel particulier. En créant un programme, ils développent des méthodes de programmation, revisitent les notions de variables et de fonctions sous une forme différente, et s'entraînent au raisonnement.

Exemples d'activités possibles : jeux dans un labyrinthe, jeu de Pong, bataille navale, jeu de nim, tic tac toe, jeu du cadavre exquis.

Attendus de fin de cycle

- Écrire, mettre au point et exécuter un programme simple.

Écrire, mettre au point, exécuter un programme

Connaissances

- Notions d'algorithme et de programme.
- Notion de variable informatique.
- Déclenchement d'une action par un événement.
- Séquences d'instructions, boucles, instructions conditionnelles.

Compétences associées

- Écrire, mettre au point (tester, corriger) et exécuter un programme en réponse à un problème donné.

Une classe “ordinaire”

Mathématiques – 3ème

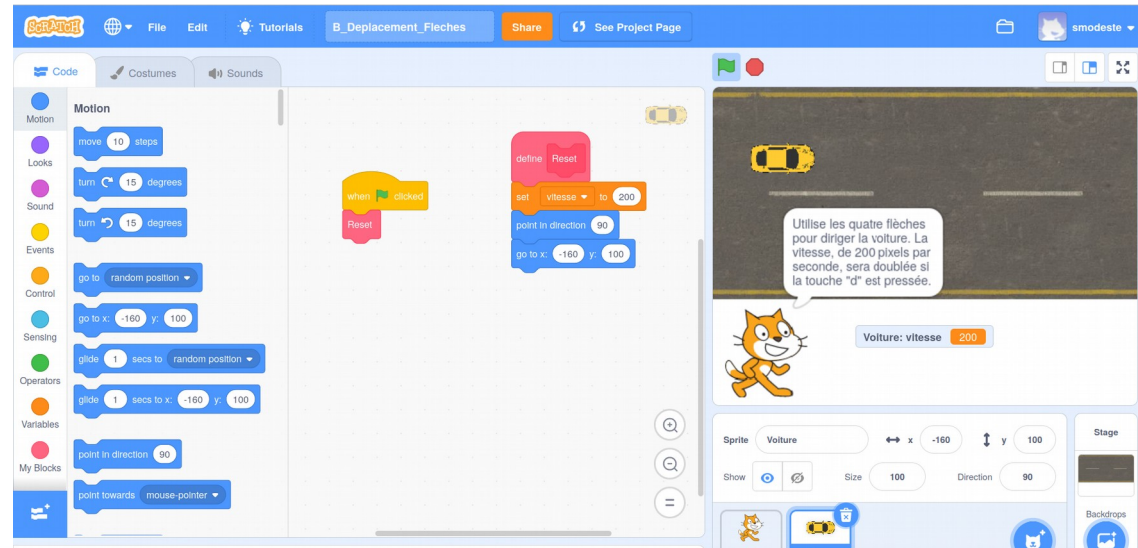
- L'enseignant de mathématiques, Damien, ancien ingénieur avec une expérience de la programmation
- Dans cette séquence : une focalisation spécifique sur la programmation « indépendamment du contenu mathématiques » et basé sur la conception de jeux vidéos.
- Contraintes : pas assez d'ordinateurs dans la salle informatique, pas de cours en demi-groupes
 - ½ groupe travaille en autonomie sur des mathématiques, ½ groupe sur Scratch
 - Travail autonome des élèves, sur la base de courtes vidéos enregistrées par l'enseignant, et des *projets* Scratch pré-remplis (avec les consignes et de l'aide)



Une classe "ordinaire"

Mathématiques – 3ème

- Séquence organisée en 6 « exercices », avec l'objectif de faire concevoir un jeu vidéo aux élèves comme projet final.
 - 1) Déplacer une voiture en utilisant les coordonnées
 - 2) Déplacer la voiture en utilisant les touches du clavier**
 - 3) Faire des demi-tours réalistes
 - 4) Gérer des événements aléatoires
 - 5) Gérer des obstacles avec des points de vie
 - 6) Faire se déplacer l'arrière plan



Une classe "ordinaire"

Mathématiques – 3ème

VIDÉO

Exercice 2: Déplacer la voiture en utilisant les touches du clavier

- Faire se déplacer la voiture avec les touches
- Fixer la vitesse à 200 pixels par seconde
- Doubler la vitesse à 400 lorsque la touche « d » est appuyée

Extraits vidéos sélectionnés de la classe et du travail de deux collégiennes (avec des coupes)

```
when clicked
  Reset
  forever
    if key right arrow pressed? then
      point in direction 90
      move 10 steps
    if key left arrow pressed? then
      point in direction -90
      move 10 steps
    if key down arrow pressed? then
      point in direction 180
      move 10 steps
    if key up arrow pressed? then
      point in direction 0
      move 10 steps
```

The image shows a Scratch script for a car movement exercise. The script starts with a 'when clicked' event, followed by a 'Reset' block. A 'forever' loop contains four 'if' blocks, each checking for a specific arrow key press. Each 'if' block contains a 'point in direction' block and a 'move 10 steps' block. The directions are 90 (right), -90 (left), 180 (down), and 0 (up). To the right of the main script, a separate 'define Reset' block is shown, containing a 'set speed to 200' block, a 'point in direction 90' block, and a 'go to x: -160 y: 100' block.

Quelques observations et sujets de discussion

- **Syntaxe et sémantique**

- Rôle important des rétro-actions du logiciel (approche expérimentale)
- Risques de travail purement syntaxique (effet *presse-bouton*, recopiage de solutions...) : Qu'est-ce qui est vraiment appris ?

- **Enjeux de contrat didactique**

- Qu'est-ce qui est attendu ? un code (formel), une solution à un problème ?
Comment une solution est-elle validée ? et par qui ?
- Rôle important de l'enseignant, et des interactions avec l'enseignant

- **Mathématiques et programmation**

- Exemple de mathématiques qui apparaissent dans et pour la programmation (vitesse)
- Difficulté pour les élèves de réaliser un travail « papier-crayon » pendant l'activité de programmation