

# MATHÉMATIQUE+CODAGE, 6e à 10e années

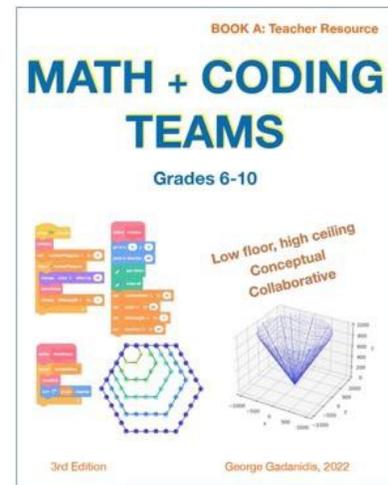
Dans un environnement d'apprentissage collaboratif

par George Gadanidis, Western University

## APERÇU

Cette session présentera une ressource basée sur la recherche qui offre :

- des tâches requérant des connaissances préalables minimales tout en proposant des relations complexes (« low floor, high ceiling »);
- des programmes fonctionnels au départ déjà encodés pour faire des mathématiques vivantes et dynamiques;
- des surprises conceptuelles et un développement de la compréhension;
- une culture de classe revitalisée.



Les participants recevront une copie PDF gratuite de la ressource, *Math + Coding Teams, Gr. 6-10*.

Si vous trouvez cette ressource utile, vous pouvez réserver une session en ligne gratuite de 45 à 60 minutes pour vos responsables de districts et vos enseignants, et recevoir une licence de district gratuite.

## PRÉSENTATEUR

George Gadanidis est professeur en didactique des mathématiques à la Western University.



Il a travaillé pendant de nombreuses années dans des classes de recherche et de vulgarisation en Ontario (ainsi qu'au Brésil), collaborant avec des éducateurs pour concevoir des expériences mathématiques qui offrent aux élèves un sentiment d'émerveillement mathématique.

Pour en savoir plus sur son travail, consultez le site [imaginethis.ca](http://imaginethis.ca).

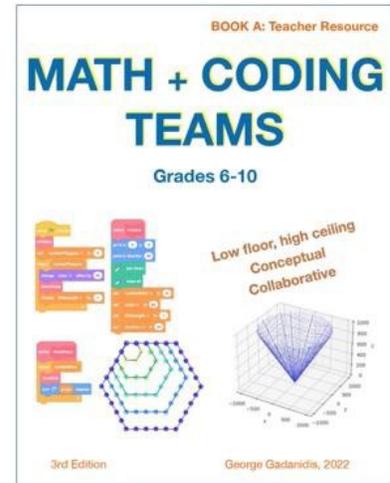
# MATH + CODING TEAMS

La séquence d'activités Math + Coding propose :

- des tâches riches en mathématiques et en programmation;
- des niveaux d'engagement variés;
- de développer une compréhension conceptuelle;
- un environnement d'apprentissage collaboratif.

Les activités sont basées sur l'enquête en classe.

## EN UTILISANT CETTE RESSOURCE



La ressource « Math + Coding Teams » peut être utilisée de deux manières complémentaires.

- Pour l'enseignement des mathématiques, pour donner vie aux concepts et aux relations mathématiques de manière dynamique grâce à la programmation.
- Pour l'enseignement de l'informatique, afin d'introduire les concepts de programmation dans le contexte de la résolution de problèmes mathématiques.

Les tâches peuvent être utilisées comme :

- Activités collaboratives en équipe.
- Activités d'approfondissement ou d'enrichissement.
- Points de départ pour l'apprentissage par projet.

*Math + Coding Teams* comprend 2 livres :

- *Le livre A* (120 pages) est une ressource pour les enseignants. Il contient : (a) les tâches de l'élève et (b) les notes de l'enseignant (et les solutions).
- *Le livre B* (56 pages) est une ressource pour les élèves. Il ne contient que les tâches de l'élève. Le livre B est conçu pour être partagé avec les élèves.

## ORGANISER DES ÉQUIPES

- Équipes de 2 à 4 étudiants.
- Une atmosphère non compétitive.
- Résolution de problèmes collaborative.
- Un sens de l'objectif commun.
- Aucun membre de l'équipe n'est laissé pour compte.

Une expérience de partage des apprentissages et d'interrogations!

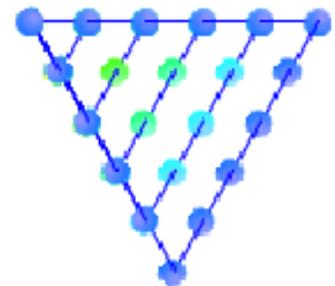
## DES APPRENTISSAGES « LOW FLOOR, HIGH CEILING »

Les tâches des équipes Math + Coding sont conçues pour offrir un point de départ à chaque élève pour s'engager dans les mathématiques et la programmation.

Les tâches offrent également aux élèves la possibilité d'étendre leurs apprentissages en posant leurs propres questions et en utilisant le code pour explorer de nouvelles avenues.

Gardez les points suivants à l'esprit lorsque vous guidez les élèves dans leurs apprentissages.

- **Un code qui fonctionne.** Les élèves reçoivent un programme qui fonctionne au départ.
- **Résoudre des énigmes.** Les activités impliquent souvent de modifier ce code pour créer différentes représentations, comme des ensembles de points pour différents nombres triangulaires (comme illustré à droite), ou autres nombres polygonaux.
- **La tentation de l'explication.** Vous serez tenté d'expliquer le code aux élèves, ou de commencer par leur apprendre les différentes parties du code. Laissez les élèves apprendre en expérimentant le plaisir de la résolution de problèmes, l'éclair de compréhension, la surprise et le développement de la perspicacité.
- **Comprendre le code.** En résolvant de tels problèmes, les élèves découvrent la signification et la fonction des différents blocs de code, et commencent à apprendre à écrire leur propre code.
- **Comprendre les concepts mathématiques.** Le code représente dynamiquement des concepts et des relations mathématiques. Cela donne vie aux concepts mathématiques, les rend manipulables et leur donne un aspect tangible.
- **Extension des tâches.** Encouragez les élèves à se poser des questions et à les explorer de manière autonome. S'ils le souhaitent, ils peuvent travailler dans plus d'un environnement de codage. La plupart des tâches sont présentées à la fois en Scratch et en Python.

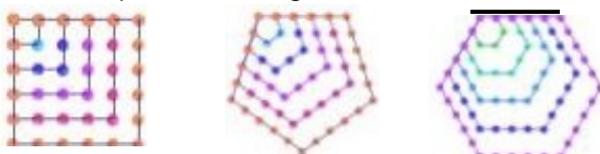


# SOMMAIRE DES CONTENUS

## Math + Codage en équipes 4

### 1. Nombres polygonaux avec Scratch

- Tâches pour les élèves 7
- Notes pour les enseignants 18

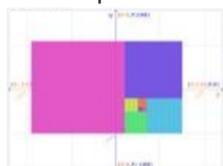


**Math:** Motifs croissants; motifs géométriques; expressions algébriques; propriétés des polygones

**Scratch:** tortue graphique; boucles; sous-programmes; variables; listes

### 2. Nombre d'or avec Scratch

- Tâches pour les élèves 24
- Notes pour les enseignants 28



**Math:** Motifs croissants; motifs géométriques; expressions algébriques; histoire

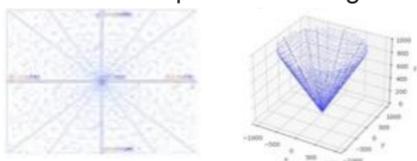
**Scratch:** tortue graphique; variables; listes

### 3.A Triples Sumériens avec Scratch

- Tâches pour les élèves 35
- Notes pour les enseignants 36

### 3.B Triples Sumériens avec Python

- Tâches pour les élèves 42
- Notes pour les enseignants 43



**Math:** Relation entre les côtés dans un triangle rectangle; triplets pythagoriciens; représentations graphiques; histoire; nombres complexes; vecteurs

**Scratch:** Boucles imbriquées; instructions conditionnelles; sous-programmes; variables; listes

**Python:** Listes; boucles imbriquées; instructions conditionnelles; esquisses 2D; esquisses 3D

### 4.A Principe d'infini avec Scratch

- Tâches pour les élèves 53
- Notes pour les enseignants 54

### • 4.B Principe d'infini avec Python

- Notes pour les enseignants 50
- Tâches pour les élèves 49



**Math:** Représentation visuelle des fractions; motifs décroissants; limite d'une série infinie; histoire

**Scratch:** tortue graphique; sous-programmes; variables; 54 sprites dynamiques (lutins programmables)

**Python:** Boucles; variables; graphiques 2D; représentation de données en tableaux

### 5.A Relations linéaires avec Scratch

- Tâches pour les élèves 59
- Notes pour les enseignants 63

### 5.B Relations linéaires avec Python

- Tâches pour les élèves 69
- Notes pour les enseignants 73



**Math:** Relations linéaires et non linéaires; opérations sur les équations linéaires; représentation graphique des relations

**Scratch:** Boucles imbriquées; instructions conditionnelles; sous-programmes; variables

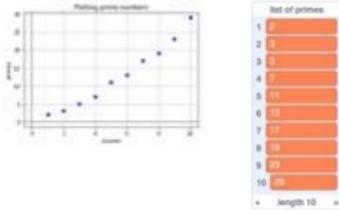
**Python:** Listes; boucles imbriquées; instructions conditionnelles; graphiques 2D

### 6.A Nombres premiers avec Scratch

- Tâches pour les élèves 80
- Notes pour les enseignants 86

### 6.B Nombres premiers avec Python

- Tâches pour les élèves 92
- Notes pour les enseignants 98



**Math:** Nombres premiers; théorie des nombres; expressions algébriques; modulo arithmétique; histoire

**Scratch:** Boucle; instructions conditionnelles; sous-programmes; variables; efficacité du code; modulo arithmétique

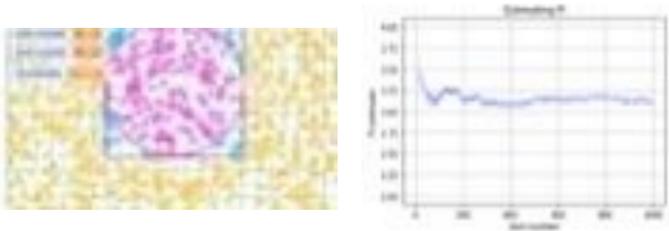
**Python:** Listes; boucles; instructions conditionnelles; graphiques 2D; efficacité du code; modulo arithmétique

### 7.A Estimation de $\pi$ avec Scratch

- Tâches pour les élèves 102
- Notes pour les enseignants 111

### 7.B Estimation de $\pi$ avec Python

- Tâches pour les élèves 115
- Notes pour les enseignants 119



**Math:** Signification de  $\pi$ ; probabilité expérimentale (fréquentielle)

**Scratch:** Boucles; listes; instructions conditionnelles; sous-programmes; variables

**Python:** Listes; boucles; instructions conditionnelles; graphiques 2D

