COMPUTER PROGRAMMING

Votre gouvernement

Québec

20-178-05A-4

TABLE OF CONTENTS

INTRODUCTION

In May 2018, the Ministère de l'Éducation et de l'Enseignement supérieur launched the Digital Action Plan for Education and Higher Education (DAP). The plan is based on the "effective integration and optimal use of digital technologies to foster the success of all Quebecers in order to promote lifelong skills development and maintenance." This plan is consistent with the changes schools have undergone over the last 20 years, specifically in regards to the Québec Education Program and education reform. The integration of information and communication technologies has taken different forms, ranging from teaching aids to learning tools used directly by students. Certain uses now provide new learning opportunities that are more engaging for students, specifically the opportunities for openness to the world that they offer.

In the context of pedagogical innovation using digital technology, Measure 2 of the Digital Action Plan for Education and Higher Education focuses particularly on computer programming and aims to "increase [its] use . . . in education"(DAP, 27). Initiatives carried out under this measure have made it possible to identify benchmarks of the potential and limits of programming for engaging students in meaningful activities while simultaneously developing these young people's technological skills, an increasingly useful tool to have in the digital world.

Computer programming is an element of Dimension 2 of the Digital Competency Framework (DCF), published in April 2019: "Developing computational thinking, particularly through the improvement of one's programming skills and enhancement of one's understanding of computer programming" (DCF, 14). This document aims to more clearly define computer programming for school staff of Québec educational institutions. It provides a short definition, a discussion on the educational potential, explicit connections with the Digital Competency Framework, starting points for the recommended pedagogical approach and some strategies to undertake in the classroom.

WHAT IS COMPUTER PROGRAMMING?

Computer programming is a human activity that involves articulating problems and generating creative solutions by making use of the possibilities and respecting the constraints of the programming language. This can be examined from various perspectives, including a technical (writing of code), creative (creative approach) or reflective (computational thinking) approach. This complex activity is remarkably social, as it requires direct or indirect collaboration and sharing.[1]

This activity involves, in particular:

- processing information
- debugging
- creating algorithms
- iterative thinking
- conditional logic
- using visualization tools

It can automate tasks, solve complex problems in a creative way, generate predictions or simulate events that have yet to happen.

FIGURE 1
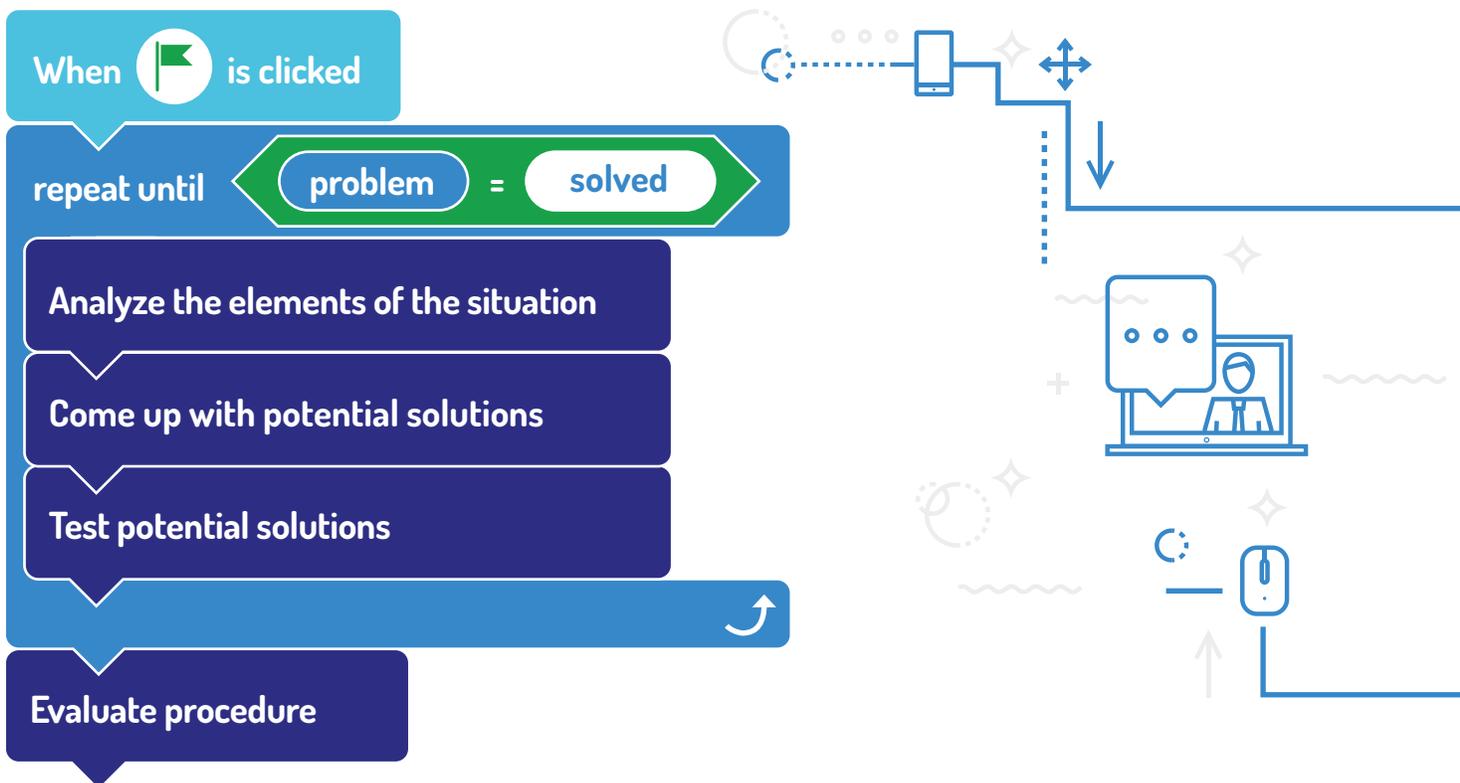
Figure 1 can serve as the starting point for exploring the educational potential of computer programming. It provides a visual representation of a possible creative process for solving problems through programming.

---

1. Definition based on analysis completed by Sylvie Barma, *Rapport final: Réaliser une étude de cas multiples qui vise à affiner les connaissances sur l'usage pédagogique ou didactique de la programmation dans les écoles du Québec* (Québec: Ministère de l'Éducation et Enseignement supérieur, 2018), **https://www.periscope-r.quebec/full-text/barma_mees_rapport_final_-_lel.pdf**. (in French only)

CONNECTIONS WITH THE DIGITAL COMPETENCY FRAMEWORK

Computer programming can be a way to make learning even more meaningful for students by creating a context full of opportunities to develop multiple dimensions of digital competency, such as producing content, solving problems, and fostering creativity and innovation. It improves their understanding of the digital world and of their technological skills.

WHAT IS THE EDUCATIONAL POTENTIAL OF COMPUTER PROGRAMMING?

Although programming and robotics can be taught and learned for their own ends, they can also be used as pedagogical tools to approach other subjects, similarly to how one might use visual arts, video design, games or research. When used in this way, they enable the development of several cross-curricular and subject-specific competencies of the Québec Education Program.

Adult general education

Computer programming can also be used as a learning tool in adult general education. The Computer Science course offered exclusively in adult general education in Secondary Cycle Two offers students the opportunity to expand their knowledge of computer programming.

Vocational training

In vocational training, computer programming can be treated as an additional skill for future professions. It also helps students develop digital literacy, which is useful to everyone.

Programming and robotics are additional teaching tools in a one's didactic and pedagogic tool kit. They allow for variety in the approaches that are used to accommodate a larger number of learning needs. While the reasons for using these tools vary from one teacher to another, some of the following effects may be observed:

ENGAGEMENT

- Increases motivation by providing an authentic context for students to complete projects
- Allows some students to experience success and develop a sense of accomplishment
- Fosters or leverages students' interest in science and technology
- Emphasizes perseverance by developing a positive attitude toward errors in the context of learning
- Fosters a love of learning in students with learning difficulties

CREATIVITY

- Encourages the development of students' creative potential
- Allows students to express their ideas and make choices
- Transforms students from users to creators of digital tools or artifacts

STRUCTURED THINKING

- Develops logical reasoning and critical thinking
- Promotes the undersanding of a concept or process
- Develops the ability to question and adjust an approach
- Promotes the understanding and use of the concept of a variable
- Develops the ability to create a structured and appropriate procedure
- Helps students learn to break down complex ideas into simpler parts
- Helps students understand that learning is a process not a destination

ADOPTION OF A POSITIVE APPROACH TO LEARNING

- Encourages the risk-taking required for innovation
- Provides differentiated challenges adapted to the students' instructional needs
- Develops students' autonomy
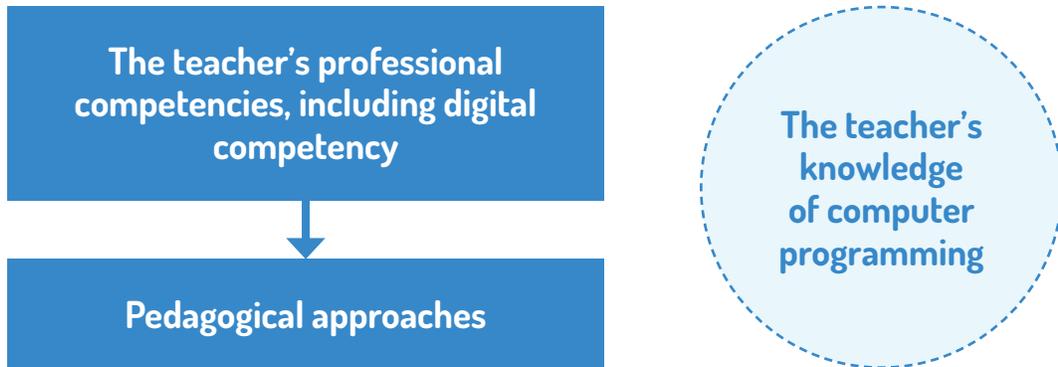- Empowers students with respect to their own learning

OTHER POSSIBLE BENEFITS

- Allows students to explore possible careers in information technology
- Fosters the development of cross-curricular competencies in authentic situations
- Allows students to become familiar with the workings of the digital world
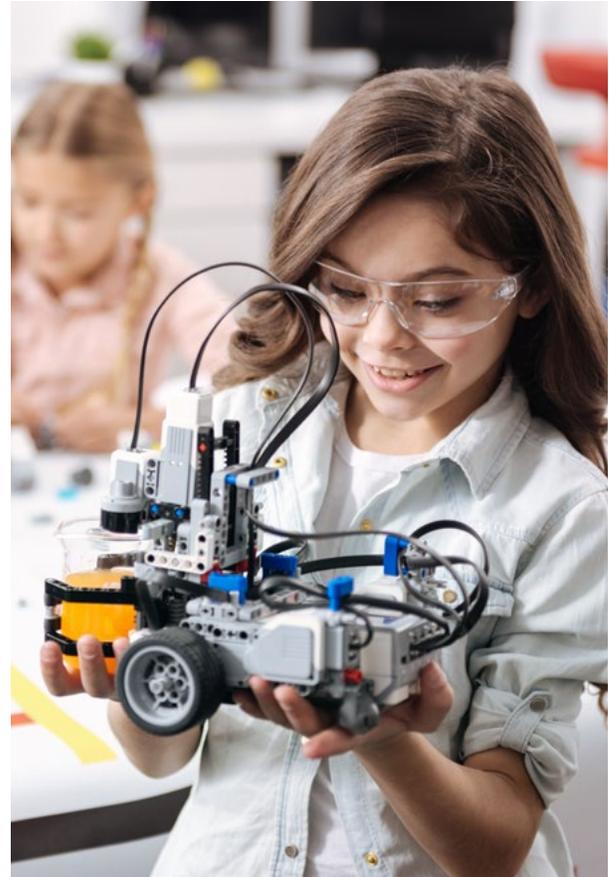
WHAT APPROACH SHOULD TEACHERS TAKE DURING COMPUTER PROGRAMMING ACTIVITIES?

Having teachers adopt a particular approach when undertaking computer programming activities in class can help increase the benefits for students.

> **The teacher's professional competencies, including digital competency**
>
> ↓
>
> **Pedagogical approaches**

The teacher's knowledge of computer programming

The first element of this approach is related to teachers' professional competencies and their digital competency. In order to make the most of programming as a pedagogical tool, a teacher must become more of a guide, accompanying the student in the use of tools, carrying out tasks and understanding documents.

The following are questions teachers can ask themselves before undertaking programming activities with their students:

- What is my level of competency development (e.g. problem-solving, digital competency)?
- What are my pedagogical approaches and how does coding fit into them?
- What is the pedagogical relationship I have built with my students and how can it help me develop appropriate methods to carry out coding activities?

By considering these questions, many teachers realize they are better equipped than they think to carry out programming activities for the benefit of the students.

When it comes to teachers' knowledge of computer programming—the syntax of a particular programming language or the understanding of loops and conditional structures (IF/THEN), for example—teachers should embrace not having perfect mastery. During their initial explorations of programming and robotics in the classroom, they will learn alongside their students, who can, themselves, become experts. The students' creative achievements can be highlighted in different ways, such as by having students present them to the class or to their peers.

There are several strategies that can help teachers identify the possibilities, challenges and solutions before jumping in with the whole class.

For example, teachers can:

- observe a programming activity in a colleague's classroom
- ask to take part in training on the subject
- put together a group of student experts to explore and test activites, which will help when working with the whole class

Another aspect of teachers' didactic and pedagogical approach is related to the evaluation of a programming or robotics project. As the student can solve the problem by taking several different "paths," it is not the final product, but the process (approach, algorithm, efficiency of the program, debugging ability, etc.) that should be the focus of the evaluation. Without being specifically evaluated itself, programming can sometimes be used to create a context for the evaluation of a subject-specific competency.

DIFFERENT TYPES OF COMPUTER PROGRAMMING ACTIVITIES

Unplugged activities

Unplugged activities are games that are played without computer equipment and that aim to introduce concepts, such as variables, loops and sorting. They can be played using playing cards, marbles, a mat, etc.

Educational robotics

Educational robotics activities use programming to make robots perform actions. Different types of robots exist and can be adapted to different educational levels.

Visual programming

Visual programming languages, which use blocks or icons, allow students to overcome the barriers related to complex syntax when being introduced to programming, particularly in the case of younger students.

Text-based programming

Text-based programming languages allow for advanced and varied uses of programming. They require rigorous syntax and the use of symbols.

Online exercises

Online exercises allow users to quickly master new programming techniques. Their potential for developing creativity is more limited, but, if used judiciously, they will allow students to learn quickly.

Creative programming

Creative programming takes an approach where, from the very start, the creative potential of programming is emphasized. This is accomplished by giving students the freedom to work on projects that are meaningful to them.

RESOURCES FOR UNDERTAKING PROGRAMMING AND ROBOTICS IN THE CLASSROOM

As part of the Digital Action Plan for Education and Higher Education, several training pathways were developed to assist teachers in familiarizing themselves with the educational robotics equipment, makerspaces and digital devices.:

http://www.education.gouv.qc.ca/dossiers-thematiques/plan-daction-numerique/formations/autoformations/ (in French only)
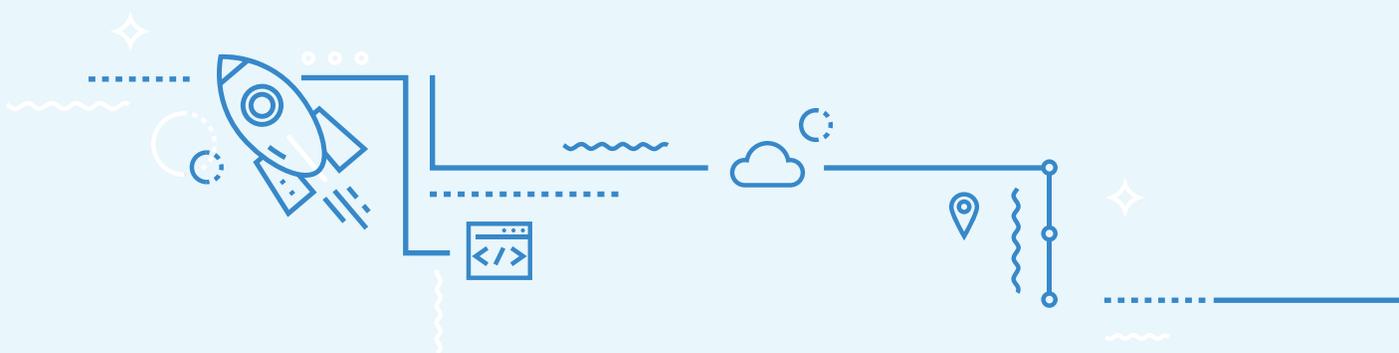
RÉCIT

RÉCIT is a network of experts in the integration of digital technologies working in Québec school service centres, school boards and private schools, coordinated by the Ministère de l'Éducation. Since 2001, the members of RÉCIT have supported teachers in the development of students' competencies through the integration of ICT. All teachers can identify and directly contact their school service centre or school board's education consultants on the RÉCIT website and request support, particularly for the pedagogical use of coding.
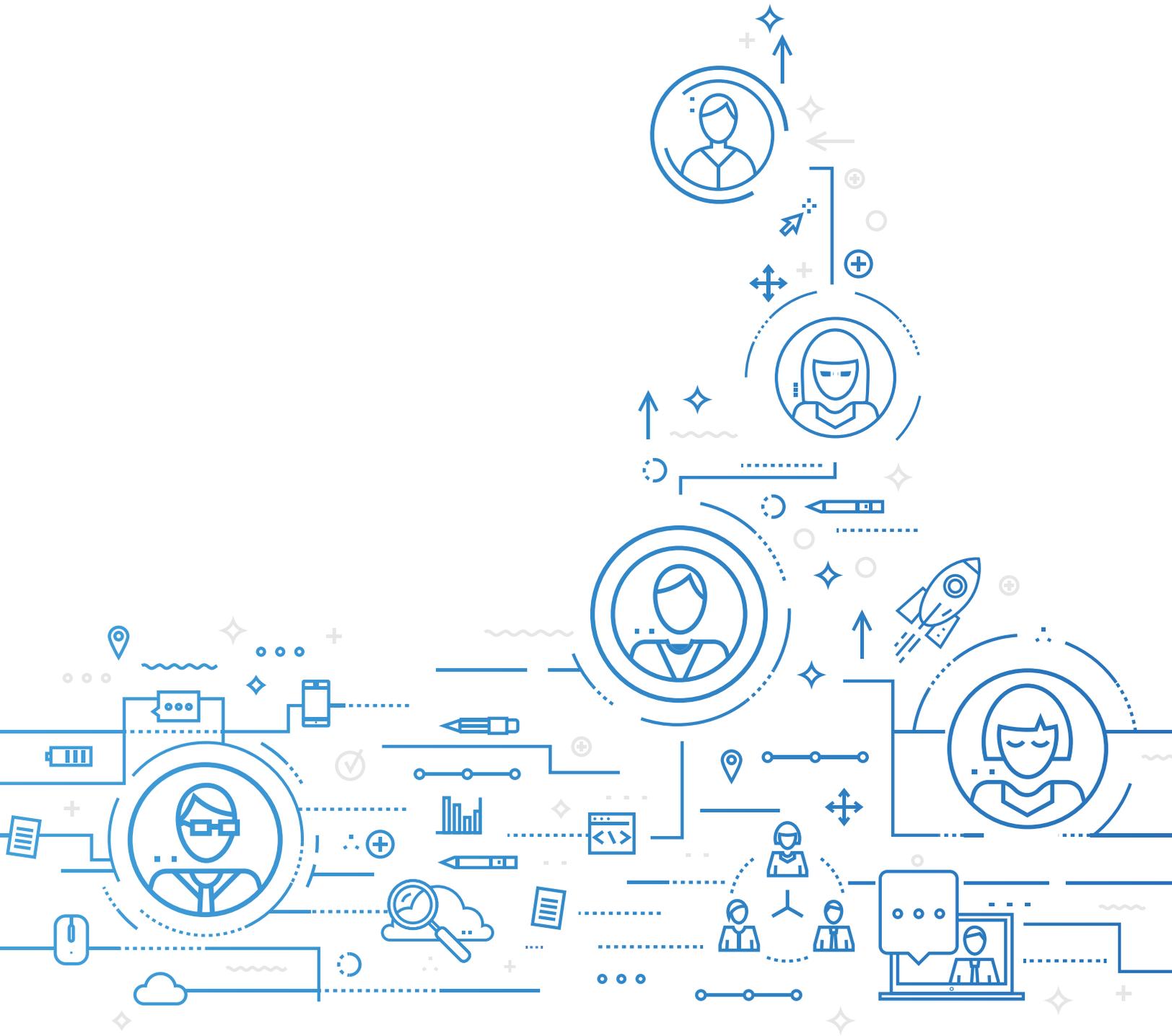
http://recit.qc.ca (in French only)

CADRE21

The Centre d'animation, de développement et de recherche en éducation pour le 21e siècle (CADRE21) offers online training for Québec teachers. Some of this training involves computer programming and its implementation in schools.

https://www.cadre21.org/ (in French only)

EDUCATION.GOUV.QC.CA

Éducation
Québec