

# Engineering projects "Savanturiers de l'ingénierie"

Vincent Ribaud  
Lab-STICC UMR 6285  
[ribaud@univ-brest.fr](mailto:ribaud@univ-brest.fr)

Cyrielle Feron, Vincent Leildé  
ENSTA Bretagne  
[First.Last@ensta-bretagne.org](mailto:First.Last@ensta-bretagne.org)

Ashwarya Arora, Sarah Le Bihan  
Université de Bretagne Occidentale  
[First.Last@etudiant.univ-brest.fr](mailto:First.Last@etudiant.univ-brest.fr)

## Abstract

*The "Savanturiers - School of Research" initiative is a French nationwide educational program that aims to introduce investigative approaches into schools through the creation of science projects. During the course of several Savanturiers projects, primary school pupils struggled with program design, among other difficulties, and we felt the need for methodical, technical and human support. In a computer science degree, the capstone project offers students the possibility of a "learning by doing" approach to software development, from the requirements to the product qualification. To overcome the difficulties mentioned, we have mobilized Bachelor of Computer Science students by giving them the classes involved in the Savanturiers projects as "clients" for their capstone projects. This has reduced the technical scope of the capstone projects but has increased their social utility. Although some Bachelor students did not take advantage of this situation, it was a positive endeavor as attested by the student feedback.*

## 1. Introduction

The Savanturiers<sup>1</sup> project is an education-through-research project, orchestrated by teachers, led by pupils and accompanied by an academic or engineer mentor. The "Savanturiers - School of Research" initiative is a French nationwide educational program that advocates learning-through-research projects in primary classes. The approach revolves around eight stages, ranging from the establishment of scientific questioning to the restitution of the work accomplished. For projects focusing on engineering, called "Savanturiers de l'ingénierie", the approach is restricted because such projects are intended to respond to a need. Around our university, five classes carried out a "Savanturiers de l'ingénierie" project presented

<sup>1</sup>In French, "savant" means "scientist" and "aventuriers" means "adventurers". A possible equivalent in English of "Savanturiers" might be "Adventifics", a contraction of adventurers and scientific.

in Table 1. Classes and mentors experienced some difficulties, such as teaching design, and we felt the need for methodical, technical and human support. We tried to provide this support through the work and help of Bachelor of Computer Science students during their capstone project. This article is an attempt to relate the students' work and pupils' projects mainly through a discussion about the artifacts produced by the students. Section 2 depicts classes and their projects, and difficulties encountered. Section 3 presents the academic context of the capstone project, the artifacts engineered for Savanturiers projects and the students' self-assessment. We conclude in Section 4.

## 2. The classes and their projects

### 2.1. The "Savanturiers de l'ingénierie" initiative

According to the Savanturiers website, education through research refers to both the initiation of pupils to methods and issues of research and the position of the teacher as a pedagogue-researcher. In the particular case of the "Savanturiers de l'ingénierie" projects, our goal is to enable pupils and teachers to learn and use the methods, techniques and know-how of the engineer, in their creative effort to shape tomorrow's world. Unlike an introduction to research, "Savanturiers de l'ingénierie" projects have to follow a logic that responds to a need. For the projects presented in this article, the approach is structured in four steps: 1. statement of work, 2. design, 3. coding and 4. assessment. Project management places each step in time and in space and associates each step with a set of tasks. Comprehension of a task operates on (at least) two orthogonal and complementary dimensions: representational structures and the activities that operate upon them. It is often easier to define the representational structures that are required at the beginning and end of a task rather than to define the task itself. Furthermore, assessment is generally centered more on products than processes,

ie focuses on the artifacts produced or consumed by tasks rather than what is going on during the task. Consequently, the assessment step operates on artifacts produced during the project.

## 2.2. A "Savanturiers de l'ingénierie" project

We will illustrate our point with an example of a Savanturiers de l'ingénierie" project. Paragraphs written in italics are related to work products, inputs or outputs of an engineering step. Paragraphs written in normal font discuss steps activities.

### *Statement of work.*

*A class of 7-9 year old pupils study the language that bees use to transmit information on the foraging areas. When an exploratory bee has found a food source, it is able to convey different information to its fellow bees with a dance called the waggle dance. For the project, we are interested in two pieces of information: the orientation and the distance of the food source in respect to the hive. This information will allow other bees to go to the food source and forage there. The project aims to reproduce the bees' behavior using mbot robots and Scratch programming.*

**Requirements elicitation activity.** We define that the simulation behavior includes four interrelated parts: (a) Representation of the simulation: setting up the hive, robots and the field of flowers; (b) Exploration: the explorer robot looks for some food, finds it and comes back to the hive; (c) Communication: the explorer robot transmits useful information to other robots; (d) Harvesting: the forager robots go to the field of flowers.

### *Requirements elicitation result.*

*The exploratory bee is simulated using an mbot robot that walks around the classroom to discover a food source such as a field of flowers. The exploratory bee, having arrived at the food source, memorizes or calculates information needed to go from the hive to the source. The exploratory bee transmits the information to other bees. Foraging bees are simulated using mbots which, using information received, go to the food.*

**Requirements analysis activity.** We recognize a typical communication problem between two entities, which requires (at least) three stages: (a) the transmitter synthesizes the information to be exchanged, (b) transmission of a message, (c) the receiver interprets meaningful information from the message received.

**Problem decomposition activity.** Once the standard situation is recognized, we can break down the problem and structure its resolution into sub-problems. According to the pupils' familiarity with the problem of foraging bees, the teacher intervenes more or less in the building knowledge process: what is an exploratory

bee, what is it looking for, what should it collect as information to be exchanged with its peers, how is the message transmitted, how are messages understood, what are the other bees doing?

**Analysis activity regarding the sub-problems and reflection.** The teacher and mentor guide the pupils toward the formulation of sub-problems and resolution hypotheses. A broad solving scheme can be prepared by the mentor that might be the following. *Simulation of the field:* the field of flowers to discover is a sheet of paper placed somewhere in the class. *Setting off to explore:* a robot explorer leaves the hive, it stops when it has detected the sheet; it measures the angle and the distance towards the hive. *Return from mission:* after returning to the hive, the robot-explorer transmits the angle and distance to other robots by infrared message. *On the way:* Forager robots leave in the direction of the "field" and stop on the sheet.

The investigation process starts again for each sub-problem. If we take the example of the search for the "field", this is a typical situation we are faced with in robotics projects: how to explore an unknown space. Once the problem of exploration has been set, it is necessary to break it down into sub-problems: memorize the starting point, criss-cross the class, avoid obstacles, find the "field".

**Synthesis activity for possible solutions.** The analysis (usually descending) gives possible solutions - for example, using the distance sensor to recognize the obstacles and the color sensor to recognize the "field" - solutions must be assembled to get a working schema.

### *Design result.*

*The food source is detected using the line follower sensor. The exploratory bee goes through the classroom to discover the color requested. The possible obstacles are detected using the ultrasonic sensor. A compass sensor can be used to measure the direction between the hive and the source. The distance is measured by calibrating the robot speed at the beginning of the session and measuring the time of displacement. The exploratory bee transmits the direction and distance to foraging bees. Foraging bees use the direction and distance information or the course memorized by the explorer to reach the food. Two programs need to be created, for the explorer and for the forager.*

**Coding activity.** This is the development activity of the solution, then the packaging in the form of a set of Scratch programs that can be used to run the simulation.

### *Coding results.*

*The different programs. The simulation is started and filmed; audio commentary or written comments explain the different parts of the simulation.*

**Assessment activity.** Assessment only makes sense

if it helps direct the pupils' attention to the knowledge derived and the skills developed.

*Assessment results.*

*Moments focused on meta-cognition, led by the teachers. The aim is for the pupils to realize their evolution and their progress and that they can assign them a certain degree of confidence, thus developing autonomy for subsequent actions.*

**In practice ...** The engineering cycle presented above is an ideal view. In fact, the project failed: globally and at almost each step.

Regarding the Statement of Work, pupils had to investigate the domain (bees' language) and formulate a need. Either the domain offered too many study possibilities, or the pupils were too young, or both; but it did not work. The mentor had to impose his Statement of Work artifact to move on to the next step.

The mentor envisaged the design activity following a "Divide-and-conquer" approach, but the pupils were unable to divide the problem up correctly. Even when sub-problems were set by the mentor, pupils were unable to see them as part of a whole. Hence the project was divided into three different sub-projects, each of them coded separately.

Coding sub-problems was too hard for almost all the pupils. Mathematical notions such as angle and distance are seen at the end of the primary school curriculum and thus were missing. Compass programming in Scratch is difficult and compass calibration is tricky. In the best case, some pupils correctly reproduced the Scratch code provided to solve sub-problems but most of them failed. Savanturiers projects are not intended to be assessed as other school work, but presented and explained to other pupils, parents and teachers. Because it was the first year we had performed Savanturiers projects, we did not organize such a presentation.

### 2.3. The difficulty in learning design

Computer Science in primary school is strongly influenced by constructionism, the theory of learning developed by Papert and Harel, who sees learning as a process where the learner builds his knowledge interacting with the object of study [1], hence the idea that we teach programming through the construction of programs, ie a consciously hands-on approach. It is difficult to convey complex algorithms to children, whereas we can get them to manipulate parts of complex programs. One goal of the Scratch online community, according to Brennan and Resnick, is to support young designers in reusing and remixing, by helping them to come up with ideas and code that will serve as a basis

for creating things which are much more complex than they could have created on their own [2]. If we consider a program as an object with which we can interact, thus an algorithm is an abstraction of this object, and writing an algorithm is a design activity. The main modality of this design activity is the decomposition of a seemingly difficult problem into sub-problems that we know how to solve, which Jeannette Wing states as: "Computational thinking is using abstraction and decomposition when attacking a large complex task or designing a large complex system [3]." In theory, learning design is performed during a round trip between the questions of the problem space set up by the projects and the elements of the solution space that we are building. In theory, writing a statement of work and eliciting requirements are activities where one highlights the important elements of the information available and the required knowledge. In theory, depending on the pupils' level of knowledge, hierarchically structuring the problem is always led by the project supervisors or delegated to the pupils under the supervisors' control. Moreover, there should be a subsequent period of reflection where pupils apply their knowledge and skills as they would instruments, tools or resources [4]. In practice, we found that the pupils knew how to reuse know-how in fairly similar situations, at the same abstraction level. However, almost all the pupils were stuck when they moved between abstraction levels, for instance when they had to design sub-problems or to assemble solution elements.

## 3. Producing artifacts for the Savanturiers projects

### 3.1. Context

The curriculum developed by the ACM (Association for Computing Machinery) states the following terminal requirement for a program in computer science: "Demonstration that each student has integrated the various elements of the undergraduate experience by undertaking, completing, and presenting a capstone project [5]." The capstone projects provide students with a "learning-by-doing" approach for the development of software, from requirements to qualification testing of the software product. Indeed, the capstone project progress is supported and supervised by software engineering processes. These processes help students to become aware and to improve what they do, especially when processes are placed in a general perspective and when continuous assessment informs students about the maturity of their practices and the quality of their production.

School and location	Number of pupils and (age range)	Savanturiers project
<i>J. S.</i> , a small (coastal) town	14 pupils (7-9)	Biology / Robotics: bee language study and simulation with mbot robots
<i>P. D.</i> , our town (priority area)	22 pupils (9-11)	Robotics: reconnaissance and rescue post-disaster mission
<i>Q.</i> , our town (priority area)	24 pupils (9-11)	Sustainable development / Robotics : what kind of pollution do we have in our school ?
<i>J. S.</i> , a small (coastal) town	22 pupils (10-11)	Mathematics / Programming: a machine to cipher and decipher messages
<i>du P.</i> , another small (coastal) town	22 pupils (9-11)	Robotics : my mbot robot does my work

**Table 1. Classes performing a Savanturiers project**

### 3.2. Organization

Our capstone project takes place at the end of the Bachelor of Computer Science curriculum. The two week project related in this article took place in April 2018, between final exams and an internship period. Students were supposed to know and be able to implement the engineering cycle presented below.

We used a colored representation of phases, that proved to help students' understanding:

**statement of work (red).** The statement of work established the project specifications, seen from the customer's point of view and expressed them in business terms.

**requirements analysis (orange).** The requirements analysis formulated the customer requirements for the software in the form of functional and technical specifications.

**high-level design (yellow).** The high-level design identified a solution that met the requirements, defined its component decomposition, architectures and allocated high-level requirements to each component. At the end of the high-level design, requirements validation was organized in the software test plan.

**detailed design (green).** The detailed design of a system or a sub-system took the component requirements as inputs and broke down, architected and allocated the requirements to lower-level components, possibly at the programming language level.

**coding and unit testing (blue).** This was the coding and testing of the components independently of their integration in higher-level components.

**integration and integration tests (indigo).** This was the activity that assembled the components and

tested them functioning as a whole. The tests organization was written in the tests description.

**software qualification (violet).** This was the activity of verifying software compliance with its reference framework and in particular that it complied with all the specified requirements.

There were 45 students, divided into three classes of 14, 15 and 16 students. The project was carried out in pairs, or (exceptionally) alone. Inside a class, each pair of students or single student worked on a different project. Available projects were assigned to students on a first-in, first-out basis. Projects were reused from one class to another. During the capstone project, each class had a dedicated room and each student had to be present for at least six hours per day. A teacher was present four hours per day in the classroom, provided guidance and carried out either formative or summative continuous assessment. Students were therefore permanently informed about the progress of their project and the grading related to the different phases and products. At the end of the capstone project, a demonstration allowed students to present and defend their project.

### 3.3. An engineering approach for primary school pupils

Among the authors of this article, two were the teachers of the capstone project. Thanks to a new accreditation established with supervisory authorities, we reoriented the capstone project by introducing two major changes: (i) "customers" of projects were the classes engaged in the Savanturiers projects; (ii) the technical scope was reduced in favor of usability, i.e. that technical aspects - programming educational robots in Scratch - are quite easy for Bachelor of Computer Science students. On the other hand, the educational dimension and the purpose of providing the Savanturiers

projects with useful support brought additional and complex questions.

Schools, classes and Savanturiers projects are presented in Table 1. Two classes belonged to schools located in priority education areas where the socio-economic environment is difficult and where additional means are provided to schools. Three classes belonged to more privileged schools, located in small towns on the coast. Typically, a class was divided into several groups of 4-5 pupils and all groups in a class performed the same Savanturiers project. The five Savanturiers projects were thus proposed to the Bachelor students (the Bee Language project was split into two slightly different projects), and we added two exploratory projects for students carrying out the project alone. Exploratory projects were based on educational robot kits and intended to explore the kits' possibilities for future Savanturiers projects.

The final project documentation had to be uploaded to the community website <https://openlab.makeblock.com> where mblock users share their projects. Students had the possibility to visit school classes and to train pupils in their products. This activity was rewarded with a bonus mark. Unsurprisingly, the most successful projects were two projects performed by students who played the game through to the end and who went to classes to train pupils. They are the two projects discussed later in this article.

When we designed the new setting for the capstone project, we had in mind the difficulty for students in understanding the purpose, the outcomes and the activities of each life cycle phase. Because "Savanturiers de l'ingénierie" projects are intended to expose pupils to a straightforward engineering life cycle, we made the assumption that the study and the realization of artifacts intended to support the straightforward life cycle would help students to become aware of a "real" life cycle. A possible drawback - which happened in some student groups - is that students confused "real" phases to be simplified phases. Confusion might also have been due to the fact that we intentionally kept the color scheme in the simplified cycle for pupils:

**statement of work (red).** The statement of work described the project in a few lines. It established what to do and what materials (e.g. sensors) and software (e.g. mblock) would be used.

**design (yellow).** The design divided the system into different programs and described what each program should do.

**coding (blue).** We created programs and eventually we assembled them in a whole.

**assessment (violet).** We verified that the project worked : approximately, satisfactorily or very well. We also examined the artifacts.

### 3.4. Mission: Pollution

The class from a school located in an education priority area in our town, carried out the research on the environmental conditions around the school and in particular on noise and atmospheric pollution. It was a double level class with 9-11 years old pupils. Most of them were foreign-born and a few may have had difficulties with spoken and written French. The purpose of the Savanturiers project was to collect environmental data using an mbot robot and display data on a computer located in the classroom.

Remember that Bachelor students were divided into three groups. In each group, a student pair chose the Mission : Pollution project, projects results are available at <https://openlab.makeblock.com/> A pair was not interested in working with pupils. The two other pairs cooperated in converging towards a common solution that is presented in Table 2.

**Student support of the project.** The solution was deployed in the classroom during several training sessions. In the first session, the two pairs of students were made aware of the level of attention and understanding of a class in a priority education area. Students trained pupils to take a single measurement, display a value and compare the value to a threshold. The students also interviewed the class teacher on her project and had a better idea about the project difficulties and what kind of artifacts could work for pupils. The second session was held the following week with a pair of students, where pupils coded a small Scratch program performing noise or gas measurements. They also worked on the concept of threshold and alert, but it was too difficult for them.

The two last sessions took place at the end of the school year with three students. The project was reoriented because pupils wanted to measure the noise during canteen and break times. The pupils learned to control the robot with a tablet app and collected measurements. The measurements series were then studied in class to understand the use of curves. During the last session with the students, the pupils defined a protocol for collecting noise measurements in the canteen. Then the pupils collected measurements over a week, and under the teacher's supervision, they compared data to the initial hypotheses they had made and established conclusions.

Phase	Students' work	Artifacts for pupils
Statement of work	Study of the different actuators, sensors and communication means of mbot robots that will be used for data management.	List of electronic modules with their features to use : sound sensor and gas sensor.
Design	The project includes several aspects: <b>Mobile measurement:</b> On demand, the robot goes to the measuring point, collects the environmental data and returns to its base. <b>Fixed measurement:</b> On a regular and programmed basis, the robot collects the environment data. <b>Recording:</b> Fixed or mobile measurements made by the robot are transmitted and stored persistently. <b>Processing:</b> from the collected data and according to the parameters chosen by the user (time interval, spatial zone, type of measurement), the data are displayed in the form of curves and diagrams.	Diagram of the decomposition of the solution as well as the description of the different elements of the solution that would be intelligible to a primary pupil. <i>Example : The program is divided into three parts. The first part (initialize block) allows the program to be started properly. Then, we use an indefinite repeat loop that contains two other parts. The second part (robot control block) moves the robot and takes measurements. The third part (display measurement block) displays the measurements graphically.</i>
Coding	Development of the solution then "packaging" in the form of a set of Scratch programs that can be assembled by the pupils to carry out the project.	Completed programs : <i>initialize, drive the robot, take the measurements, display the measurements in a chart, display the measurements in a curve.</i>
Assessment	<a href="https://openlab.makeblock.com/topic/5ac5ecb9d441d3dd0d06f9eb">https://openlab.makeblock.com/topic/5ac5ecb9d441d3dd0d06f9eb</a> Description of the project on the shared site	Online: videos, photos, programs.

**Table 2. Engineering the project "Mission: Pollution"**

Phase	Students' work	Artifacts for pupils
Statement of work	Study of different coding techniques : Fleissner Grille, Caesar cipher, Scytale, Knights Templar cipher, Freemason's cipher. Study of the communication means between mbot robots.	Choice of Caesar cipher. Tests of the capabilities of mbot robots to send and receive messages using an infrared link.
Design	The project has three related parts: <b>Encryption:</b> programming encryption using a Caesar cipher. <b>Communication:</b> a transmitting robot transmits a secret message to a receiving robot. <b>Decryption:</b> programming decryption using a Caesar cipher.	Decomposition and description of the solution intelligible to a primary school pupil. <i>Example : For centuries, encryption has stopped messages from being intercepted. One of the best known is the "Caesar cipher". The principle is to replace each letter by another, following a fixed offset that is the same for each letter.</i>
Coding	Development of the solution then "packaging" in the form of a set of Scratch programs that can be assembled by the pupils to carry out the project.	Completed programs : <i>message offset, character look-up, modulo override, encryption, decryption, display.</i>
Assessment	<a href="https://openlab.makeblock.com/topic/5ac8ecced441d3dd0d0774ec">https://openlab.makeblock.com/topic/5ac8ecced441d3dd0d0774ec</a> Description of the project in the shared site	On line: documentation, programs.

**Table 3. Engineering the project "A machine to cipher and decipher messages"**

### 3.5. A machine to cipher and decipher messages

A class from a school by the seaside studied how secret messages can be transmitted from a transmitter to a receiver using cryptography techniques. Pupils were 10-11 years old. The class had a good education level and pupils liked projects and Scratch programming. The project aimed to cipher messages, transmit them using robots, and decipher them. The restitution of the project, under the guidance of the doctoral student in cryptography, mentor of the project, was done during the visit of the class to their future secondary school, during a cryptography workshop jointly organized with a sixth grade class and its mathematics teacher.

**Student support of the project.** In each of the three student groups, a pair chose this project. The final restitution being complex, several working sessions were held between the teachers and different student pairs. It led students to understand what was required and students also trained teachers to program a Caesar cipher using Scratch and to exchange messages between mbot robots. During a session at the school, a pair of students trained the primary school pupils in the same techniques.

The workshop at the college mobilized the project mentor and four volunteer Mathematics students and required 24 robots. Each primary school pupil was paired with a sixth grade pupil and each child had a robot that allowed him or her to send and receive the encrypted messages. Encryption and decryption were done using Scratch programs made by the pairs of computer science students.

### 3.6. Learning support

The process naturally used by students in computer science is the practice of functional analysis with black boxes, which breaks down functions into simpler functions. "This decomposition allows the complexity to be dominated, especially considering black boxes, that is, components that are not needed, at a certain level of use and understanding, to analyze the inner workings [6]." As pointed out by Bordallo and Ginestet [7], the use of black boxes is pedagogically justified because it preserves the interest of the students by not lengthening the projects excessively and because it corresponds to a systemic functioning of the human mind.

Students also had to train teachers. In a research intended to assess students' accompaniment in primary schools, Lafosse-Marin and Jeanbart state: "The traditional dual situation corresponding to an antisymmetry of competences between teacher-expert and pupils-novices turns into a triadic and multiform

situation [8]." In the few cases we are aware of, and especially for teachers with little literacy in digital technologies, it has been observed that teachers joined pupils in a "novice" status and brought students into an expert position, even though they are undergraduate students. We came to the [obvious] observation that the introduction of digital sciences in primary school could not be done without strong support for primary school teachers.

### 3.7. Students feedback

At the beginning of the project, students were asked for their consent for their work to be observed, and for their participation by means of a questionnaire composed of three parts. The first part was about the students' perception of the capstone experience. The second part was a self-assessment of student roles during the capstone project. We used the four roles proposed by Tardif as significant of the learning paradigm [4]. The third part allowed students to express their feelings about the objectives of the project.

Table 4 presents the different items in the questionnaire for the first and second parts, on a Likert scale with 5 possible answers: strongly agree (OK), agree (+ OK), neither agree nor disagree (+/-), disagree (-OK), strongly disagree (KO). The last but one column synthesizes all of the answers by ranking the answers as follows: strongly agree (5), agree (4), neither agree nor disagree (3), disagree (2), strongly disagree (1). Because we drastically changed the focus and the content of the capstone project, the last column presents the answers synthesis for the last year. This year, 41 out of 45 students responded to the questionnaire. Last year, 22 students over 25 answered a similar questionnaire.

Table 5 presents the different items of the third part, on the same Likert scale (the third part did not exist last year). The students evaluations were very positive, particularly the objective of making projects for primary schools classes.

As mentioned above, there is a meta-cognitive issue with the engineering phases of the projects. Generally speaking, the main goal of a capstone project is to learn-by-doing a simplified cycle of software development through a fairly realistic project. However, in our case, the "customers" of the project do not ask for a solution to their problem, but for guidance and helpful resources. In other words, pupils do not have to be provided with the final product but with pieces of the solution that will help them to carry out the project. For some groups, Bachelor students did not realize (or did not want to take in account) the issue and completed the project instead of the pupils. Obviously, the work

The capstone project	OK	+OK	+/-	-OK	KO		avg	<
I had time to learn and do the project.	29	11	0	1	0		4,66	3,90
I found the project complex.	1	10	16	14	0		2,95	3,86
I was committed to performing the project.	19	22	0	0	0		4,46	4,50
I found the project realistic.	14	20	6	0	1		4,10	4,22
I had to deepen my knowledge and skills to perform the project.	3	21	12	3	2		3,48	4,45
I improved my working methods thanks to the project.	2	12	22	3	2		3,22	3,95
My roles in the capstone project	OK	+OK	+/-	-OK	KO		avg	i
<i>investigator</i> : I discussed my questions about the project and/or I defended my solutions with the other students.	17	21	2	1	0		4,29	4,49
<i>co-operator sometimes expert</i> : I explained some project points to other students and/or I obtained explanations from others.	15	22	2	1	0		4,22	4,36
<i>clarifying actor</i> : I asked the teacher or other students to ensure that I had understood my project well and to verify the adequacy of my proposals and differences.	13	21	5	2	0		4,10	4,59
<i>strategic user of available resources</i> : I used the available resources and/or supplementary resources and I verified their relevance.	20	17	3	1	0		4,37	3,86

**Table 4. Questionnaire about the project and self-assessment of student roles.**

My feeling about the Savanturiers project	OK	+OK	+/-	-OK	KO		avg
Requirement elicitation: project topics were suitable for primary school pupils.	12	16	10	3	0		3,90
Design: project design was suitable for primary school pupils.	5	14	14	7	1		3,37
Coding: project realization was suitable for primary school pupils.	6	12	14	7	2		3,27
I appreciate the fact that my project is used by primary school classes.	24	17	0	0	0		4,58
I appreciate the fact that I can record a video or perform a demonstration of my project for primary school pupils.	15	18	7	1	0		4,15

**Table 5. Questionnaire on the students' feelings about projects objectives and interest.**



was easier than in a typical capstone project and students completed the project in half of the allocated time. For sure, they missed one of the pedagogical objectives we had in mind and they learned probably less than in a typical capstone project. This can be seen in Table 4 by comparing the last but one column (average for this year's capstone project) with the last column (average from the previous year capstone project). On the other hand, these students had fun and enough time to polish their projects and it helped other students struggling with the pedagogical issues.

The previous year, the capstone project was a client-server management information system intended to manage personal address books and agendas and a central directory and a social network. The system was implemented using Java, JSF relying on an Oracle RDBMS. Sub-systems interacted through a protocol based on UDP.

The last column of Table 4 shows the average obtained the previous year for questions of the first and second parts of the questionnaire. A comparison of the two last columns synthesizes the pros and cons of the new version of the capstone project. In the first part, we see that the project for primary schools was perceived as less complex, and that compared to the previous year, students felt that they had had more time this year, they had lessened their knowledge and less improved their working methods. In the second part, we also see that there was less investment in the first three roles, probably because the project was less complex this year. On the other hand, there was much more investment in the strategic user role this year, which is probably related to the meta-cognitive issue discussed above and that could mean that students were more sensitive to the quality and usefulness of the resources provided to them or found on the internet.

Without the resources produced by the Bachelor students, the Savanturiers projects presented in Section 3.4 and Section 3.5 would probably have failed. However, there is no guarantee of success, the mentor produced the same kind of resources for the Bee language project (Section 2.2) and the project failed.

### 3.8. Suggestions for improvement

#### A Scratch-based computational thinking framework

Brennan and Resnick developed a computational thinking framework that emerged from their studies of the activities of young Scratch programmers [2]. The authors defined three key dimensions of the computational thinking framework: (1) *computational concepts*, (2) *computational practices*, (3) *computational perspectives*.

The concepts that are common to many Scratch projects are transferable to other programming contexts or to other areas: sequences, loops, events, parallelism, conditionals, operators and data [2]. The 'scratcher' adopted a variety of strategies and practices that the authors grouped into four main sets of practices [2]: **Being incremental and iterative** - developing a little, then trying it out, and then developing further; **Testing and debugging** - be sure that things are working well, find and correct problems; **Reusing and remixing** - building projects based on other people's work to create things that are much more complex than the scratchers could have created on their own; **Abstracting and modularizing** - building something large by putting together collections of smaller parts. Finally, the authors added a dimension they called perspective [2]: **Expressing** - Realize that computation is something that they can use for design and self-expression. **Connecting** - Be aware of the power and the value of creating with others, and for others. **Questioning** - Feel empowered to ask questions about and with technology.

**Articulating the framework for pupils** Our university and National Education representatives for our region signed a 3-year agreement that let our students teach programming (Scratch Jr, Scratch and mblock) in primary school classes. During the academic year 2017-2018, 26 classes benefited from the initiative, including all classes performing a Savanturiers project. The programming course was delivered over 6 consecutive weeks, with a 1-hour lesson each week. The course covered all the computational concepts mentioned above. Pupils coded either using Scratch or mBlock, a Scratch-derived environment for programming educative robots called mbot. According to the teachers and our students, almost all the children loved Scratch programming and acquired a minimal set of programming knowledge. However, we observed that only a few pupils saw programming as an activity they could use on their own. When the programming class was over, most pupils moved on to the next school subject, as they probably did for other subjects.

The dimension of practices is helpful to think about improvements for the future. The goal of a Savanturiers project is to get pupils involved in a rigorous, productive and team-based learning process. We devised the projects presented here using a simplified waterfall cycle, but regarding the dimension of practices presented above, it might not have been appropriate. If we want children to be incremental and iterative, should we switch from a top-down to a bottom-up approach? In this case, we have to define small problems that pupils solve using iterative design

and coding. Then, we present a larger picture where pupils can combine their solutions. Briefly stated, should we replace our "divide-and-conquer" principle by a "tinker-and-assemble" approach?

In our opinion, testing and debugging practices are strongly related to the pupils' motivation to achieve a result, a key component in successful learning. It might be the case that a bottom-up approach fosters the pupils' goal to use the piece of code s/he is currently working on, hence developing the pupils' will to fix the code if it does not work as expected.

Reusing and remixing support the development of critical code-reading capacities and provoke important questions about ownership and authorship [2]. Throughout the project, we should encourage conversations with and between pupils about their code and how they can cooperate and give credits to others' work.

Abstraction and modularization is an important practice for all design and problem solving and Bachelor students used it intensively when they prepared resources for the pupils' project. The pupils were able to handle a stack of code as a new block, abstracting and modularizing its behavior in a single component. We should develop this ability to promote the "tinker-and-assemble" approach.

**Towards a mentoring role** In a Savanturiers project, the role of a mentor is to give pupils and their teacher the benefit of learning from an engineers thought process in his/her field of expertise. According to the Savanturiers website, engineers involved with schools are responding to a vital need to develop a spirit of innovation, creativity, critical thinking and a sense of responsibility in today's youth, and helping young people to think about their future in concrete terms. This year, Bachelor students produced resources for the pupils' projects but interacted with pupils only if they volunteered. However, mentors availability is limited and Bachelor students may act as mentor-assistants, the role being part of the capstone experience. We envisage associating Bachelor students with primary classes, their teacher and their mentor at the beginning of the semester when the Savanturiers project starts. Students will accompany the mentor when he or she visits the class and will interact with the class by e-mail or instant messaging, asking for help from the mentor if needed. At the beginning of the capstone project, students will have to report on the class progress and about the simplified engineering cycle the class is following and the artifacts produced. Our goal is to foster a meta-cognitive work on engineering life-cycle, steps and work products. In addition, if students are made aware

of pupils progress, we expect that students will produce artifacts that are more suited to the project needs. In terms of outcomes for Bachelor students, they will have the possibility to raise awareness among young people of the engineering-related scientific challenges society faces. They will share their knowledge and engineering skills with teachers and pupils and help them to carry out an engineering project in their classroom.

## 4. Conclusion

When carrying out the Savanturiers projects, three difficulties quickly made themselves felt: the teachers' technological maturity, learning about design and availability of support. To overcome these difficulties, we mobilized student capstone projects and gave them the classes engaged in the Savanturiers projects as customers. The technical scope was reduced in favor of usability, meaning that the technical aspects - programming educational robots in Scratch - were easily acquired by computer science graduates. However, the educational dimension and the support objectives of Savanturiers projects brought additional and complex issues. Two projects particularly benefited from this mobilization, but all the students appreciated the concrete and fun aspects of the Savanturiers projects as well as the educational purposes of their capstone projects.

## References

- [1] S. Papert and I. Harel, *Constructionism*. Ablex Publishing Corporation, 1991.
- [2] K. Brennan and M. Resnick, "New frameworks for studying and assessing the development of computational thinking," in *Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada*, pp. 1–25, 2012.
- [3] J. M. Wing, "Computational thinking," *Commun. ACM*, vol. 49, no. 3, pp. 33–35, 2006.
- [4] J. Tardif and A. Proulx, *Intégrer les nouvelles technologies de l'information: quel cadre pédagogique?* ESF, 1998.
- [5] R. Shackelford, A. McGettrick, R. Sloan, H. Topi, G. Davies, R. Kamali, J. Cross, J. Impagliazzo, R. LeBlanc, and B. Lunt, "Computing curricula 2005: The overview report," in *ACM SIGCSE Bulletin*, vol. 38, pp. 456–457, ACM, 2006.
- [6] C. Pair, "La technique dans l'école d'aujourd'hui," *Cibles*, Mai 1987.
- [7] I. Bordallo and J.-P. Ginestet, *Pour une pédagogie du projet*. Hachette Education, 2006.
- [8] M.-O. Lafosse-Marin and P. Jeanbart, "L'accompagnement en sciences et technologies à l'école primaire: un enseignement collaboratif pour un meilleur partage des savoirs," *Revue GN n*, vol. 83, 2009.