

“Designing an introductory lesson to Scratch programming language within the framework of STEAM Education to reinforce primary school pupils’ Computational Thinking”

Schina, Despoina¹; Esteve-González, Vanessa²

¹PhD student, Faculty of Education. University of Rovira and Virgili,
despoina.schina@estudiants.urv.cat

²Associate Professor, Faculty of Education. University of Rovira and Virgili,
vanessa.esteve@urv.cat

ABSTRACT

This project proposes a well-structured lesson plan for an introductory lesson to Scratch Programming Language for Primary School pupils from 8 to 11 years old. The lesson plan incorporates exploratory, story-telling and project-based teaching approaches and it is based on STEAM educational framework integrating the teaching of Science, Technology, Engineering Art and Mathematics. Objective of this lesson is to introduce students to Computational Thinking and get them acquainted with data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices. The present research follows the Design Based Methodology and the lesson designed is implemented in one context-a Greek non-formal educational institution. The lesson designed is implemented several times in the same context but with different pupils. In total, 27 pupils participate to the research (56% boys and 44% girls). The teaching materials and strategies are amended and improved throughout the research process to provide future teachers with suitable teaching materials and guidelines for teaching Scratch Programming Language.

KEY WORDS: *Scratch, STEAM, Computational Thinking*

INTRODUCTION

Nowadays, there is a growing demand for personnel in the field of Computer Programming. The European Commission estimates there are 700,000 unfilled vacancies for IT practitioners, of which programmers outnumber other IT professionals by ratio 5 to 1 (Depryck, 2016). Introducing coding in the curriculum at an early age is a long-term investment in bridging the skills gap between the technology demands of the labour market and the availability of people to fill them (Depryck, 2016). For Greece that is suffering a deep debt crisis and financial stagnation since 2009, investing into Computer Programming Education could possibly lead to the country's economic recovery. There is an urgent demand for children getting familiarized with

Computer Programming at an early stage and being equipped with skills in the field of Computational Thinking. The European Union has taken a lot of initiatives to promote Computer Programming and STEM/STEAM education, one of them being TACCLE project in which 'Best practices and experiences that focused on the development of computational thinking and related skills in any level of pre-university education were identified, shared and valorized' (Garcia Peñalvo, 2016). However, in the context of Greece, there is room for improvement when it comes to designing and implementing and integrating computer programming into the school curriculum.

RESEARCH PROBLEM

The necessity of further research in this field is evident as quality teaching materials are needed now more than ever to be used at schools aiming at reinforcing primary school pupils' Computational Thinking skills. This is exactly the purpose of this study; create a lesson proposing certain content, structure, teaching methods and materials for Scratch Programming language, design it specifically for the Greek context, implement it and improve it. Schools need a suitable Curriculum for computer programming and teachers need guidelines and teaching materials.

LITERATURE REVIEW

STEAM- Computational Thinking

The present research proposes the design of an introductory lesson to Scratch programming language in the framework of STEAM Education. STEAM combines 5 specific disciplines; Science, Technology, Engineering, Arts and Mathematics. STEAM originates from STEM teaching model in which the discipline of Art is not included. STEM is a teaching model that is a currently recognized application in cross-content-area disciplines to create knowledge as a whole (Bazler & Van Sickle, 2017). STEAM education is not in opposition to STEM education but, it enriches and expands the scope of STEM education by integrating the discipline of Art (Taylor, 2016). As far as Scratch is concerned, it is a programming language, developed by the MIT Media Lab and can be used in the framework of STEAM education. Scratch is distinguished for the following characteristics; it is suitable for children, it is specifically designed for young programmers (Resnick et al., 2009), it makes the learning procedure fun and arouses learners' interest and enthusiasm (Wilson & Moffat, 2010; Saez Lopez et al., 2016), Scratch offers an intuitive way into programming and leaves lots of space for creativity (Romeike, 2008). Scratch also promotes the learning of important mathematical concepts (Resnick et al., 2009; Calao et al., 2015) and promotes pupils' computational thinking skills (Saez Lopez et al., 2016; Calao et al., 2015).

Kotsopoulos et al., (2017) elaborated a Pedagogical Framework for Computational Thinking (CTPF), including four pedagogical experiences: (1) unplugged, (2) tinkering, (3) making, and (4) remixing. The integration of these

experiences in the school curriculum would bring a lot of benefits to students. Unplugged activities are kinaesthetic activities, activities in which objects are introduced, activities possibly based around a game or challenge and activities containing elements of fantasy and story-telling that aim at increasing children's interest, curiosity, motivation and creativity (Nishida et al., 2009). Tinkering experiences do not involve constructing a new object but, taking things apart and engaging in changes and/or modifications to existing objects. (Kotsopoulos et al., 2017). Tinkering is a valid and valuable style of working, characterized by a playful, exploratory, iterative style of engaging with a problem or a project (Resnick & Rosenbaum 2013). When it comes to making experiences, students are required to problem-solve, make plans, select tools, reflect, communicate, and make connections across concepts (Kotsopoulos et al., 2017). Remixing experiences involve sharing an object and modifying or adapting it in some way and/or embedding it within another object to use it for substantially different purposes (Kotsopoulos et al., 2017). Unplugged, Tinkering, Making and Remixing experiences favor the development of Computational Thinking.

Teaching Scratch

Scratch is an amazing programming environment that fosters students' creativity and computational thinking. Teachers should adopt an appropriate methodology and teaching strategies so that students take a full advantage of all that Scratch offers. Scratch can be taught through exploratory, project-based and storytelling teaching techniques. First, students should be allowed to explore the Scratch features as the backdrop and sprite library and programming blocks. Smith and Neumann (2014) pointed out that 'teachers can select premade coding blocks from the extensive online library to allow students to explore concepts such as properties, attributes, rotations, reflections, and translations by making incremental or small changes to existing programs. Not only should students be allowed to experiment with different blocks, but they should also be encouraged to tinker existing programs. By exploring and trying out the different blocks, they will be able to understand the blocks' functions and the principles of programming. Additionally, Project Based Learning enables an active teaching approach (Saez-Lopez et al., 2016). The active approach gets positive results and stands out through a teaching methodology centered on the student to develop projects and creations (Saez-Lopez et al., 2016). Project-Based methodology is ideal for teaching programming, as it encourages collaboration among students and facilitates teaching in mixed abilities classes. In addition, the research of Demo, G, B., & Williams, L. (2014) is innovative as a Scratch course was implemented through story-telling to address learners who were not familiar with informatics.

RESEARCH OBJECTIVES - QUESTIONS - METHODOLOGY

The objective of the research was to design an introductory lesson to Scratch to introduce primary school pupils to Computational Thinking and teach them how to (1) use Scratch digital environment and basic programming blocks (2) solve basic programming problems and (3) use algorithms creatively in the framework of STEAM Education. We also wanted to explore in which way the teaching materials and teaching strategies should change to improve the lesson. The research questions are the following:

- RQ1: To what extent does the introductory lesson to Scratch, designed in the framework of STEM Education, teach primary school pupils how to use Scratch digital environment and basic programming blocks, solve basic programming problems and use algorithms creatively?
- RQ 2: To what extent did the teaching materials and strategies have to be amended throughout the iterations?

The research follows a Design Based Research methodology. As defined by Plomp and Nieveen (2007), educational design research is the systematic study of designing, developing and evaluating educational interventions (such as programs, teaching-learning strategies and materials, products and systems) as solutions for complex problems in educational practice, which also aims at advancing our knowledge about the characteristics of these interventions and the processes of designing and developing them. This project does not have a tight linear structure but, it is open-ended and iterative; it evolves as improvements are introduced in every cycle. According to Susan McKenney & Thomas C Reeves (2013), the interventions of educational research design evolve over time through multiple iterations of investigation, development, testing and refinement. The research is divided in several stages according to De Benito & Salinas, (2016) scheme of work. First, the situation was analyzed and the problem was defined, then, the solutions were developed and the research instruments were selected. Typically, several cases are selected to implement the research, however, due to limited amount of time, the present study only reaches its implementation in one case. In this study, the researcher has the double role of teacher-researcher. In the present research there are 5 research stages as shown in Table 1 below.

Research stages	Description
Stage 1 Definition of the problem and documentation	Carrying out literature review, considering the context, setting the research problem, objectives and questions.
Stage 2 Functional Analysis	Selecting instruments and materials that would fulfill the research objectives.
Stage 3 Design and Validation of Research Materials and	Designing and Validating the lesson plan and the worksheets

Instruments	
Stage 4 Implementation	Carrying out the lessons and implementing changes per cycle.
Stage 5 Production of documentation and principles design	Studying the results and drawing conclusions.

Table 1: Research Stages and Description.

CONTEXT - RESEARCH INSTRUMENTS

The research took place in a non-formal educational institution specialized in the field of foreign languages. The school is located in Thessaloniki, Greece. 8 introductory lessons to Scratch were offered in a period of 4 weeks, from the 18th of March to the 8th of April 2017. The participants of the research were self-selected pupils of a private language school in Thessaloniki, Greece and joined the class voluntarily. They were 27 pupils, 15 boys and 12 girls. 11 pupils were on the 3rd year of Primary School, 9 on the 4th year, 2 on the 5th year and 5 on the 6th year of Primary School.

When it comes to the research instruments, an Observation Report was completed during and/or after the lesson by the teacher. The observation report contained statements about students' attitudes, behavior and about the difficulties that they encountered. It also included statements about the teaching materials and methods employed during the lesson. The teacher read the statements and marked the appropriate answer in a scale from 1 to 5. The report was designed to be short and easy in order to be completed quickly while teaching. Qualitative data obtained was analyzed on Atlas.ti. In addition, Focus Groups were introduced into the research, having as an objective to obtain feedback from the pupils. The focus group discussions were carried out right after each lesson, lasted around 10 minutes and were recorded by the researcher. The focus group discussions provided the researcher with feedback about the lesson and helped the researchers to improve the lesson plan and materials.

LESSON PLAN AND TEACHING MATERIALS

As far as the teaching materials are concerned, a Scratch lesson plan was created including the skills, the content, the learning objectives of the lesson (Table 3) and the methodology (Table 4). In addition, a worksheet with programming challenges was created for the second Part of the lesson (Annex). In the first and second iteration the children had to complete the worksheet on their own whereas in the third iteration children carried out the activity in pairs.

INTRODUCTORY LESSON TO SCRATCH	
General Skills	Specific Skills
<p>-Ability to use Scratch digital environment.</p> <p>-Develop computational thinking. Acquire basic programming skills. Develop problem solving skills and logical reasoning.</p> <p>-Develop creative thinking.</p>	<p>Science: introduction to the concepts of speed and time, spatial awareness: Left, Right, Up and Down, geometry: axis x and y and angles.</p> <p>Technology and Engineering: create and debug simple programs, use logical reasoning to predict the behaviour of simple programs, develop problem solving skills and computational thinking.</p> <p>Mathematics: calculations, measurements, solving practical problems</p> <p>Art and Design: use a range of materials creatively to design and make products.</p>
Content	
PART A- familiarization with Scratch digital environment and basic programming blocks	
Introduction to Scratch main features (sprites/backdrop/script) and basic blocks MOTION (move, turn, change x and y) EVENTS (when ... clicked/pressed) and CONTROL (wait, repeat, forever). Creating scripts stopping and running programs. Familiarization with the following programming terms: programming, instructions, sequence, algorithm, repetition and loops.	
PART B - Creation of short programs and development of programming skills and debugging	
The pupils will be asked to debug the following programs from the worksheet; 1) Blocks not attached 2) Wrong event block, 3) Sprite upside-down. The pupils will be asked to create the following short programs to help Scratchy; 1) Scratchy climbing up the tree (change y axis, glide to or move to), 2) Scratchy finishing the race (change x axis, glide to or move X steps). The pupils will be encouraged to experiment on their own with blocks from looks category: 1) Change colour, 2) Say something, 3) Think of something. The pupils will be asked to use the correct loop to create and improve programs: 1) Use repeat block to improve programs and make them shorter, 2) Use forever loop to create a program.	
PART C- Creation of longer program, using algorithms creatively	
The pupils will be asked to create an animation on Scratch using various blocks from the following block categories: event, control, motion, looks, sound. Each sprite they insert must be programmed to perform an action.	
Learning Objectives	
<ul style="list-style-type: none"> • Understand how simple algorithms work. • Detect, correct and explain errors in algorithms and programs. • Design, write and improve programs that accomplish specific 	

<p>goals.</p> <ul style="list-style-type: none"> • Get familiarized with Scratch digital environment and programming. • Learn basic programming terms: programming, instructions, sequence, algorithm, repetition and loops • Acquire skills in the field of STEAM.
--

Table 2: Lesson Plan- Skills- Content-Learning Objectives.

METHODOLOGY	Time
<p>Introduction to Scratch</p> <p>Get the children excited about Scratch introducing the character Scratchy through a video made on Scratch. Show them the script that lies behind the animation. Manipulate code and demonstrate. Explain that programming is giving instructions.</p>	2 min
<p>PART A</p> <p>A.1 Introduction to Scratch key features</p> <p>Teach the main features of Scratch by demonstrating on the projector/interactive board. Sprites are the objects or the characters that can perform actions, sprites can be modified, rotated and you can even create your own. Backdrop is the background it can be named, edited, created, and deleted. Script is a collection of blocks that all interlock with one another. The blocks and their order are very important, as they determine how sprites interact with each other. Show them how to drag the blocks and put them together.</p> <p>Activity 1- familiarization with sprites and backdrop</p> <p>Ask pupils to explore the Scratch library. Afterwards, play 3 sounds and ask them to choose a suitable backdrop and 2 sprites.</p> <p>Activity 2- experimentation with blocks.</p> <p>Ask pupils to experiment with some block categories (e.g. motion, looks, sound, events, control). Afterwards, ask them what they have discovered.</p> <p>A.2 Teach the main blocks:</p> <p>Explain the main EVENT blocks. Event blocks make scripts run, they are essential for every project: without the hat blocks from this category, a project would not be able to begin. Demonstrate. Tell them that the event blocks are also called Hat blocks. Then explain MOTION blocks. Motion blocks control a sprite's movement. Explain and demonstrate what the following blocks do (turn, move, change x, change y, go to).</p> <p>Activity 3- familiarization with repeat and forever block</p>	<p>A.1 10min</p> <p>Act. 1+2 10min</p> <p>A.2 10min</p> <p>Act. 3 3 min</p>

Ask the kids to stand up and shout <i>JUMP! JUMP! JUMP!</i> The kids jump 3 times. Then, ask them to find an easier way to give the same instructions. The right answer is to shout <i>JUMP 3 TIMES</i> Explain that the repeat block 'hugs' other blocks. Show them where Forever block is and ask them what it does.	
Part B Practice with graded challenges Ask the pupils to complete the worksheet. The challenge sheet contains various graded challenges, that the pupils should be able to carry out after the first part of the lesson. The teacher will urge the pupils to try to solve the challenges on Scratch programme. After solving each challenge on Scratch, the pupils write down the answer on the worksheet.	20-25 min
Part C Create your own animation Students already know from the 'Meet Scratchy Video' that Scratchy loves swimming and he's an excellent swimmer. Ask students to create an underwater animation with Scratchy and his underwater friends. Each sprite inserted must perform an action.	20 min

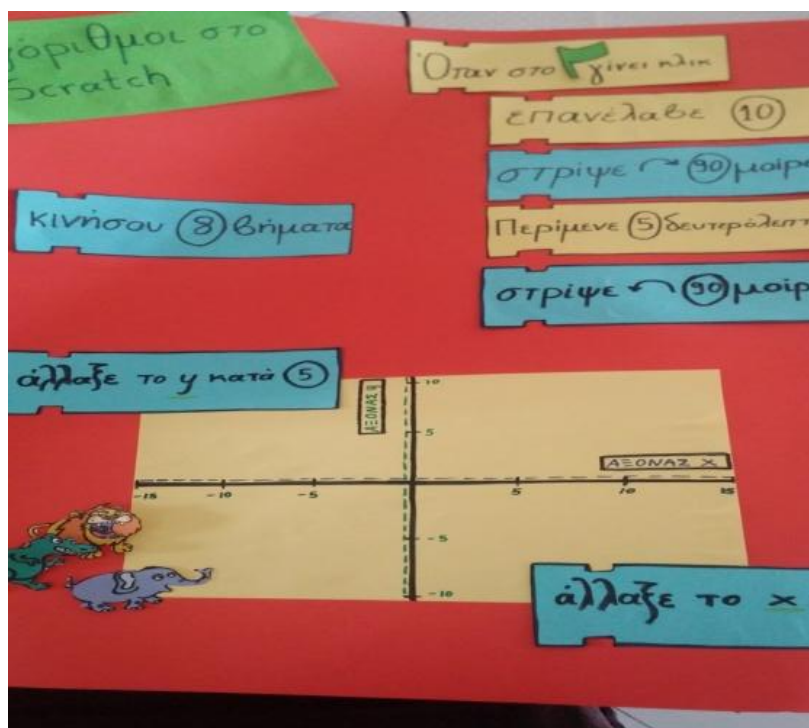
Table 3: Lesson Plan- Methodology.**RESULTS**

The introductory lesson to Scratch was designed according to the National and European Curriculum and guidelines for Primary School Education in the field of Computer Science. The lesson was also designed in the framework of STEAM education, proposing the development of skills in the field of Science, Technology and Engineering, Mathematics and Art and Design. The lesson designed aims at boosting pupils' practical mathematical knowledge through teaching programming as proposed by Calao et al., (2015). It also targets at raising pupils' Computational Thinking abilities by introducing data practices, problem solving computational practices, modelling and simulation practices and systems thinking practices. In the first part of the lesson students studied a complex system as whole, understood the relationships within a system, communicated information about a system and assessed computational models. In the second part, students worked on trouble-shooting and debugging, assessing different approaches/solutions to a problem, programming/ choosing effective computational tools. In the third part of the lesson students chose effective computational tools, defined systems and managed complexity and thought in levels. In addition, unplugged, tinkering and making experiences were employed in the lesson having as a goal to foster pupils' Computational Thinking. As far as the unplugged are concerned, the children carried out a kinaesthetic task; performed the teacher's instructions- algorithm and then assessed and improved the instructions- algorithm. As far as tinkering experiences are

concerned, pupils tinkered algorithms, for example, they tinkered motion blocks in the first part of the lesson. Finally, when it comes to making experiences, students were created short and long programs in the second and third part of the lesson.

When it comes to the teaching strategies, after studying several scholar articles since 2009 that were related to teaching Scratch programming language to primary school pupils, the following teaching strategies were incorporated in the teaching process. First, the teacher/researcher followed Smith's (2014) proposal for letting students explore concepts by making incremental or small changes to programs. What is more, Saez-Lopez's Project Based Learning teaching approach was adopted by assigning to the pupils to create an animation on Scratch on their own. This teaching approach was adopted because it facilitates teaching in mixed abilities classes and that was exactly the case of these classes; the age of the learners varied and included children from the 3rd to the 6th year of primary school. Additionally, story-telling was incorporated in the teaching process; pupils created their own short personalized stories with Scratchy and his underwater friends. What is more, in the third Part of the lesson a short animation in which Scratchy was the protagonist and was talking about his adventures was created animation. The whole lesson was successfully based on story-telling and made engaging for little children. Finally, collaborative learning was introduced in the third cycle of the research. In the first and second cycle of the research, the pupils carried out the worksheet activities on their own and during the focus group discussions mentioned that the worksheet activities seemed like a test and that they were intimidated. It was also observed by the teacher and noted in the observation report that the pupils seemed to be stressed out throughout the second part of the lesson. For this reason, the second part of the lesson was amended and carried out in pairs in the third cycle, so that it would be more enjoyable for the pupils.

As far as the teaching materials are concerned, changes were implemented in the worksheet and lesson plan and supplementary visual materials were created to improve the lesson as presented in Table 5. First, when it comes to the worksheet, there were some extra lines on the second page that generated confusion among the students. These extra lines were deleted in the following cycle. Apart from that, this text was added on the top of the worksheet; 'read the instructions carefully and try to find the solutions to the following challenges' to make it clearer for the students. When it comes to Supplementary Teaching Materials, in the first cycle, Year 3 and year 4 pupils had trouble with the x and y axis. They found it hard to grasp the concept. They would be able to better understand that concept through visual materials and for this reason a poster was created and used in the second and third cycle, as you may see in Picture 1. The above-mentioned changes were implemented based on the teacher's comments on the observation report.



Picture 1: Poster.

	2 nd cycle	3 rd cycle
Worksheet	The extra lines were deleted and the instructions were improved.	x
Supplementary teaching materials	A poster for teaching X and Y axis was created.	x
Lesson Plan	x	Part B to be carried out in pairs.
Teaching strategies	x	Collaboration in Part B

Table 4: Changes Implemented per cycle.

CONCLUSIONS

The conclusions are presented according to the research objectives.

O1. When it comes to the design of an introductory lesson to Scratch, a three-part introductory lesson was designed according to the National and European Curriculum, proposing the development of skills in STEAM Education, integrating Computational Thinking activities according to Kotsopoulos et al., (2017) Pedagogical Framework.

O2. Teaching materials and teaching strategies were improved throughout the process of the three cycles. The instructions of Worksheet 1 were improved and a supplementary poster was designed to support children during the learning process. In the 3rd cycle of the research, collaboration was added to the existing teaching approaches (exploration, tinkering, project-based teaching and story-telling).

Below, suggestions are given to researchers carrying out the research in another context:

- It is highly recommended to introduce Computational Thinking with unplugged activities and incorporate in their teaching a variety of visual materials.
- Researchers should study in greater depth collaboration in a Scratch Introductory lesson, possibly by asking pupils to carry out more tasks in pairs.
- Researchers would be advised not to adopt the role of the teacher. It would be better for another person to teach the class and the researcher to observe the pupils and the lesson.
- It is highly recommended to implement this research in other contexts. For example, to carry it out in a formal educational context, in a public school and with a larger sample for a longer period of time.

REFERENCES




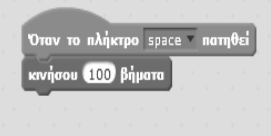

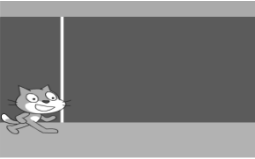



- Bazler, J., & Van Sickle, M. (2017). Cases on STEAM education in practice. <http://doi.org/10.4018/978-1-5225-2334-5>
- Calao L.A., Moreno-León J., Correa H.E., & Robles G. (2015) Developing Mathematical Thinking with Scratch. In: Conole G., Klobučar T., Rensing C., Konert J., Lavoué É. (eds) Design for Teaching and Learning in a Networked World. Lecture Notes in Computer Science, vol 9307. Springer, Cham
- De Benito Crosetti, B., & Salinas Ibáñez, J. M. (2016). La Investigación Basada en Diseño en Tecnología Educativa. *Revista Interuniversitaria de Investigación En Tecnología Educativa*, 0(0), 44–59. <https://doi.org/10.6018/riite2016/260631>
- Demo, G. B., & Williams, L. (2014). The many facets of Scratch. *Issep'14*, 8–16.
- Denning, P. J. (2009). The profession of IT Beyond computational thinking. *Communications of the ACM*, 52(6), 28. <http://doi.org/10.1145/1516046.1516054>
- Depryck, K. (2016). From computational thinking to coding and back, 16–18. <http://doi.org/10.1145/3012430.3012492>
- García Peñalvo, F. J. "A brief introduction to TACCLE 3 — coding European project," *2016 International Symposium on Computers in Education (SIIE)*, Salamanca, 2016, pp. 1-4. doi: 10.1109/SIIE.2016.7751876
- Kotsopoulos, D., Floyd, L., Khan, S., Namukasa, I. K., Somanath, S., Weber, J., & Yiu, C. (2017). A Pedagogical Framework for Computational Thinking.

- Digital Experiences in Mathematics Education. <http://doi.org/10.1007/s40751-017-0031-2>
- Linn, M. C., Aho, A. V, Blake, M. B., Constable, R., Kafai, Y. B., Kolodner, J. L., & Bradley, S. (2010). Report of a Workshop on The Scope and Nature of Computational Thinking. *Thinking* (Vol. 1). <http://doi.org/10.17226/12840>
- McKenney, S.E., & Reeves, T. (2013). *Conducting Educational Design Research*. NY: Routledge
- Nishida, T., Kanemune, S., Idosaka, Y., Namiki, M., Bell, T., & Kuno, Y. (2009). A CS unplugged design pattern. *Proceedings of the 40th ACM Technical Symposium on Computer Science Education - SIGCSE '09*, 41(1), 231–235. <http://doi.org/10.1145/1508865.1508951>
- Plomp, T., & Nieveen, N. (Eds.). (2007). *An introduction to educational design research. An Introduction to Educational design research*. Shanghai: SLO •Netherlands institute for curriculum development Tjeerd. Retrieved from <http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:An+Introduction+to+Educational+Design+Research#0>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52, 60–67. <http://doi.org/10.1145/1592761.1592779>
- Resnick, M., & Rosenbaum, E. (2013). Designing for Tinkerability. *Design, Make, Play: Growing the next Generation of STEM Innovators*, 163–181.
- Romeike, R. (2008). Workshop: A creative introduction to programming with scratch. *IFIP International Federation for Information Processing*, 281, 341–344. http://doi.org/10.1007/978-0-387-09729-9_49
- Sáez López, J. M., González, M. R., & Cano, E. V. (2016). Visual programming languages integrated across the curriculum in elementary school: A two-year case study using “scratch” in five schools. *Computers & Education*, 97, 129–141. <http://doi.org/10.1016/j.compedu.2016.03.003>
- Schina, D., & Esteve-González, V. (2017). Designing an introductory lesson to Scratch programming language within the framework of STEAM Education to reinforce primary school pupils' Computational Thinking. *XX Congreso Internacional EDUTEC 2017: Investigación, Innovación y Tecnologías, la triada para transformar los procesos formativos*. Universidad de Santiago de Chile
- Smith, C. P., & Neumann, M. D. (2014). Scratch it out! Enhancing geometrical understanding. *Teaching Children Mathematics*, 21(3), 185–188.
- Smith, N., Sutcliffe, C., & Sandvik, L. (2014). Code club: bringing programming to UK primary schools through scratch. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education - SIGCSE '14*, 517–522. <http://doi.org/10.1145/2538862.2538919>
- Taylor, Peter Charles, "Session N: Why is a STEAM curriculum perspective crucial to the 21st century?" (2016). 2009 - 2016 ACER Research Conferences. 6. http://research.acer.edu.au/research_conference/RC2016/9august/6
-

Wilson, A., & Moffat, D. C. (2010). Evaluating Scratch to introduce younger schoolchildren to programming. Proceedings of the 22nd Annual Workshop of the Psychology of Programming Interest Group, 64–75.

ANNEX-WORKSHEET

Challenges -Read the questions below carefully and reply briefly.

<p>1. Why is this script not working? How can you fix it?</p> <p>_____</p> <p>_____</p> <p>_____</p>		<p>2. I press the green flag but it is not working. Why?</p> <p>_____</p> <p>_____</p>	
<p>3. Can you give the same instructions but using less blocks?</p> <p>_____</p> <p>_____</p> <p>_____</p>		<p>4. Can you correct the script to make Scratchy spin forever?</p> <p>_____</p> <p>_____</p>	
<p>5. Scratchy wants to climb up the tree. Can you help him?</p> <p>_____</p> <p>_____</p> <p>_____</p>		<p>6. Scratchy is tired but he wants to finish the race. Can you help him?</p> <p>_____</p> <p>_____</p>	
<p>7. Scratchy is upside down. He is dizzy! Help him stand upright!</p> <p>_____</p> <p>_____</p> <p>_____</p>		<p>8. Scratchy can change his colour. Do you know how?</p> <p>_____</p> <p>_____</p>	
<p>9. Scratchy wants to tell you that he loves you! Help him say it!</p> <p>_____</p> <p>_____</p> <p>_____</p>		<p>10. Scratchy has been walking the whole day. Now he is thinking he is tired. Try to find the right block.</p> <p>_____</p> <p>_____</p>	