



<http://www.diva-portal.org>

Postprint

This is the accepted version of a paper presented at *European Conference on e-Learning - ECEL 2017*.

Citation for the original published paper:

Mozelius, P., Öberg, L-M. (2017)

Play#based learning for programming education in primary school: The Östersund model.

In: Anabela Mesquita and Paula Peres (ed.), *Proceedings of the 16th European Conference on e-Learning ECEL 2017* (pp. 375-383). Reading: Academic Conferences and Publishing International Limited

Proceedings of the European Conference on e-Learning

N.B. When citing this work, cite the original published paper.

Permanent link to this version:

<http://urn.kb.se/resolve?urn=urn:nbn:se:miun:diva-31955>

Play-based learning for programming education in primary school- the Östersund model

Peter Mozelius and Lena-Maria Öberg

Mid Sweden University, Department of Computer and Systems Sciences, Östersund, Sweden

Abstract: Learning to program has been classified as problematic learning with high drop-out rates and low motivation at university level. Like the learning of a natural language the learning of syntax and basic techniques in a programming language is easier and more natural if started at a younger age. This study is based on an evaluation of a pilot project for students in a Primary school where computational thinking and programming concepts have been introduced as play-based learning. Students have learnt concepts such as 'Bubble sort' by playing the algorithm without any computer. Later the learnt concepts and algorithms have been implemented with Scratch and the Python programming language as main tools.

The aim of the study is to describe and discuss a model for implementing computational thinking and programming for fifth grade students by play-based learning. As the overall research strategy the case study approach was used to evaluate this pilot project. Data has been collected in a combination of observations, interviews and group discussions during a 15 session pilot course and three workshops on teacher training. Findings have been analysed thematically and presented using the SWOT framework to identify and discuss strengths, weaknesses, opportunities and threats in the Östersund model.

Learning outcomes of the pilot were promising but with individual variations in the student group. The idea of introducing programming and computational thinking as early as in primary school seems like a good idea, but the recommendation is to keep sessions play-based and with enjoyment as the key feature to engage primary school students. Conducted sessions in the pilot are worth replicating and so are the teacher training workshops. However, the challenge that remains is to create a sustainable and scalable implementation of the described model including primary school teachers' professional development.

Keywords: Play-based learning, Programming education, Computational thinking, Play-based learning, Teacher training, NOOC

1. Introduction and Aim

Computer programming has in many studies been classified as problematic learning with high drop-out rates and low motivation reported in university courses (Lahtinen, Ala-Mutka & Järvinen, 2005; Wiedenbeck Labelle & Kain, 2004; Bennedsen & Caspersen, 2007; Gomes & Mendes, 2007; Olsson & Mozelius, 2015). Considering syntax and grammatical rules artificial languages have much in common with natural languages (Chomsky & Schützenberger, 1963), and basic constructions in both types of languages seem easier to grasp and use if learning is started at a younger age (Johnson & Newport, 1989; Gibson, 2003). Several studies have reported on promising results on the idea of game-based or play-based programming for all in primary school (Kynigos, 1992; Gibson, 2003; Wilson, A., Hainey, & Connolly, 2012).

Learning to program computers in primary schools would also provide students with twenty-first-century skills that includes more than just writing code. Students should also get the

opportunity to develop computational thinking (CT), a concept that involves problem-solving with the use of generic computer science techniques (Lawanto et al., 2017). This can be done in activities that does not necessarily need to involve any computers.

A country where CT has been introduced in primary as well as in secondary school is New Zealand (Heintz, Mannila & Färnqvist, 2016; Duncan, Bell & Atlas, 2017) and the first research studies reports on a successful start (Bolstad, 2016) opening up opportunities for cross-curricula learning (Bell, Duncan & Atlas). An interesting initiative in New Zealand are the game-coding workshops given at public libraries around the island (Bolstad, 2016).

In Sweden on the other side of the globe the government gave in 2015 the National Agency for Education (Skolverket) the assignment to develop a new curriculum for primary school and to update the one for secondary school. Briefly, the proposal is based on the idea of strengthening the digital competence and introducing programming in school ((Heintz, Mannila & Färnqvist, 2016). Programming will not be a subject on its own, and rather a part of mathematics and technology. Curricula will be updated to enable new activities aimed to strengthen students' digital skills and knowledge. (Swedish National Agency for Education, 2017) Less is stated in the field of teachers' professional development. How should an appropriate teacher training be funded and organised?

1.2 Aim

The aim of the study is to describe and discuss a model for implementing computational thinking and programming for primary school students that is built around play-based learning. This is only a study of a minor pilot project in one primary school, but with the long-term perspective of developing a model that could be useful for other schools in the Östersund region.

2. Didactics, tools and related research

Teaching and learning activities for learning to program might be implemented in various way, with different concepts of instructional design and with different tools. This chapter describes some choices that were made in the Östersund initiative.

2.1 Computational thinking

Computational thinking (CT) is the main objective in the Östersund model, playful programming or other play-based activities are just two of several ways to improve students CT. Scratch is a mostly used as a tool to visualise computational thinking and the Python programming language is a tool to implement basic CT concepts by writing code. The relation between writing code in Python and CT has been described by Perkovic (2015) in a book for Python programming in higher education as:

“Computational thinking is a term used to describe the intellectual approach through which natural or artificial processes or tasks are understood and described as computational processes. This skill is probably the most important one you will develop in your training as a computer scientist.”

According to Brennan & Resnick (2012) some important CT concepts to include in play-based programming for children are:

- Sequences (series of instruction)

- Loops (iteration)
- Events (triggered actions)
- Parallelism (simultaneous execution)
- Conditionals (selection)
- Operators (arithmetic, logic and string operators)
- Data (storing, retrieving, and updating)

More or less the same fundamental CT concepts that are taught and learnt in an introductory course at university level, but a difference is in the order that they are introduced.

2.2 Play-based learning

Game-based learning for programming education seem to have a strong potential for courses in higher education both when it comes to game construction and students playing educational games (Malliarakis, Satratzemi & Xinogalos, 2013; Mozelius, 2014). The use of games and game is often also a part of play-based learning (Sanchez, 2016), but so are also other ludic resources (Fernández-Oliveras & Oliveras, 2015). Play-based learning could be defined as *"... young learners constructing knowledge as they explore, experiment, discover and solve problems in playful and unique ways"* (Ebbeck, Yim, & Lee, 2013). As an example, one of the startup activities in the Östersund pilot students are playing the Bubble sort algorithm.



Image 1. Playing the Bubble sort algorithm

In Denmark the term playful programming have been used for a joyful learning of CT concepts and game construction in Scratch. Playful programming is also the name of a Danish project with the interesting idea of connecting developers, educators, parents, learners and researchers, to build a common vocabulary for describing, developing and teaching (Allsopp & Ejsing-Duun, 2016). This multi-stakeholder approach is also a fundamental idea in the Östersund model.

2.3 Scratch

Scratch has been developed of the Lifelong Kindergarten Group at the MIT Media Lab, used in more than 150 countries and available in more than 40 languages (Scratch, 2017). It is a free visual puzzle programming environment that can be accessed online. The programming environment is also available as a free and platform independent application. Source code is open and released under a GPLv2 licence.

Programs are built by dragging and dropping puzzle blocks to create programming concepts such as conditions, statements, variables, selection and iteration. The absence of syntax errors makes Scratch suitable for younger students. In the Scratch environment it is possible to create animated stories, games, music and implement most of the fundamental programming concepts (Maloney et al., 2010; Meerbaum-Salant et al, 2010; Ozoran et al, 2012). Finally, the Scratch environment seems to decrease untrained teachers' reluctance toward programming (Yukselturk & Altiok, 2017).

2.4 Python

Python is an interpreted high-level multi-purpose programming language first released in 1991 that can be used for imperative, object-oriented and functional programming. Compared to most other programming languages Python has higher readability as well as writability, where complex processes can be written with few lines of code. Python interpreters are free and available for most platforms to support platform independency and the idea of 'write once run everywhere'. (Perkovic, 2015; Lam, & Rossiter, 2016). However Python is often hard to install on tablets and other mobile devices.

Python has built in high-level data structures and dynamic typing which together with a large standard library enables fast creation of various types of programmes. In the Idle editor expressions and constructions can be instantly tested and evaluated. Furthermore, Python supports a wide variety of add-on modules and packages, which encourages program modularity and effective programming. Interactive digital games can relatively easy be created with use of the add-on packages Tkinter and PyGame. (Perkovic, 2015; Lam, & Rossiter, 2016; Python.org, 2017)

2.5 Related research and some lessons learnt

Programming education have a long history leading back to the 1950s. However, in most countries the focus is mainly on tertiary education and partly on secondary school. Few countries have a nation-wide implementation of programming or computational thinking for primary school. Sweden has only recently begun the digitalisation of primary school and there are many reasons for looking at lessons learnt by pioneers in other countries.

An early well-known pioneer was Seymour Papert and with the use of turtle graphics in the Logo programming language. What can be learnt from Papert (1993), is among many other things, the importance of joyfulness and to construct real world artefacts. What has been shown in later research studies is that the idea of constructing digital games can be a successful way to teach and learn basic programming in primary school (Maloney et al., 2008; Denner, Werner & Ortiz, 2010; Wilson, Hainey, & Connolly, 2012).

Another interesting legacy from Seymour Papert is to concretise computational thinking by robot programming for younger children, which have been tested in various countries (Sipitakiat & Nusen, 2012). An interesting setup tested in Denmark is the multi stakeholder approach, involving parents as well as the IT industry (Allsopp & Ejsing-Duun, 2016). Finally, there seems to be reasons for opening up the educational sandbox with activities outside school like it has been carried out in Australia (Bolstad, 2016).

3. Method and data collection

The overall research strategy was a case study with in-depth studies of a pilot project with 10 teaching and learning sessions for students and 3 workshops for teacher training and awareness raising. All activities have been conducted at the primary school *Östbergskolan* in Östersund, Sweden during the 2017 spring semester. Data was collected in a combination of observations with photo documentation as field notes, interviews and informal group discussions during workshops and coffee breaks.

Collected data was analysed using the SWOT framework to categorise a deductive thematic analysis to identify strengths, weaknesses, opportunities and threats with an internal and an external dimension divided into positive and negative aspects as depicted in Table 1 below.

| SWOT Model | Helpful to achieve objectives | Harmful to achieve objectives |
|--------------------------------------|-------------------------------|-------------------------------|
| I n t e r n a l | S trengths | W eaknesses |
| E x t e r n a l | O pportunities | T hreats |

Table 1. The used SWOT analysis framework

The basic idea with SWOT is that the future pattern of actions should match found strengths and opportunities at the same time as weaknesses and threats should be concerned (Stacey, 1993). Used simplistically the SWOT framework is a naïve tool, the recommendation is to use complementary analytic frameworks (Pickton & Wright, 1998). In this study the SWOT components have been used as categories in a deductive thematic analysis.

4. The Östersund model

Swedish primary school now has a detailed proposal for digitisation where computational thinking and programming are part of the concept. Programming will not, in the near future, be a subject of its own, but rather taught and learnt in sessions for Mathematics and Technology (Swedish National Agency for Education, 2017). There exists a consensus on the digitisation and the importance of introducing CT concepts at a younger age, but when it comes to instructions on how there is a lack of clear guidelines.

In Östersund, a city with around 60 000 inhabitants in Mid Sweden, the municipality has put up some minor funding to explore how. A two month pilot project has been designed and implemented in a collaboration between the municipality's IT-advisor (A) and a local entrepreneur (E) with good programming skills and a vision of a future city where most inhabitants have basic programming skills. The concept was built on a 10 session series of activities for a 5th grade primary school class at Östbergskolan, a public school in Östersund with 25 students and Teacher (T) as the pilot teacher. T had no previous programming experience and was rather chosen by her curiosity more than by ITC skills or insight in computational thinking.

The started redesign of Swedish primary and secondary schools should in a longer perspective include all schools and also rural regions all over the nation. As a part of the pilot three workshops were given for primary school teacher with a wide variety of backgrounds and not only teachers in Mathematics or Technology. All primary school teachers in Östersund were invited and around 55 teachers showed up at the first workshop. After an initial discussion on how society and citizens in general will be affected of the ongoing digitisation, fundamental CT concepts were introduced and visualised in the Scratch environment. Learning activities for the students and the workshop sessions for the teachers shared the same thematic mix with sorting algorithms, game construction and creating animations for both audiences.



Image 2. Students writing pseudo code for the Bubble sort algorithm

A consistent setup, if considered that the pilot activities for the students are meant to be replicated by the workshop participants. Another idea is to repeat a concept with multiple implementations, where a concept can be played, described, written in pseudo code, implemented in Scratch or in the Python programming language.

Furthermore, a fundamental idea in the Östersund model is to open up for collaborations between primary, secondary and tertiary education. As an example university students are supposed to create programming exercises for secondary school and secondary school students should in the same way construct activities and learning material for primary school. Finally, as in the Danish ‘Playful Programming Project’, professional developers should be connected with students, teachers and researchers with a multi-stakeholder approach that also should include rural areas of the Jämtland-Härjedalen region.

The overall aim for all activities is to build for the future with a new generation that will be better prepared for the new tasks and professions that will be a fundamental part of the ongoing and inevitable digitisation. Hopefully, there will be benefits for all stake-holders and what is done today in primary school will pay off in ten years for universities a IT-companies. At the same time a generally higher level of digital skills and CT knowledge could also be seen as a democratic way to prevent exclusion in an increasingly digitised society.

5. Findings and discussions

Main findings have been categorised into strengths, weaknesses, opportunities and threats in Table 2 below.

| Findings | Helpful to achieve objectives | Harmful to achieve objectives |
|--|---|---|
| I n t e r n a l | Strengths: Ambitious pilot Motivated pilot teachers Engaged teacher workshops Initial learning outcomes Support from municipality | Weaknesses: Replication Teacher drop out Entrepreneur E dependency Lack of learning content Lack of technical support |
| E x t e r n a l | Opportunities: Resource sharing Experience sharing Inter-school collaboration Further funding Professional development | Threats: Scalability Sustainability Rural area coverage Heterogeneous equipment Professional development |

Table 2. Categorised findings

5.1 Strengths

The ambitiously designed pilot project has undoubtedly kick-started an engagement among primary schools in general and the fifth grade class at Östbergskolan in particular. Both Entrepreneur E and Teacher T have led teaching and learning sessions in a charismatic way

and in collaboration with IT-advisor A, sessions have been well-organised. Motivation and engagement are of course key factors also when it comes to students. The initial learning outcomes are impressive even if there are large individual variations. Like it has been reported from similar initiatives for chess training in schools, around 10 per cent of the students get an instant understanding and an intrinsic motivation for further exploration.

The support from the municipality has been a good support even if it is a relatively small amount of money invested. Workshop venues have been used for free at the primary school and the workshop sessions have been given during daytime as a part of teachers' ordinary work without any extra rewards than coffee and sandwiches. All teachers that have attended all three workshops have their own Scratch account and a basic understanding of CT and how play-based learning can be a part of their daily school activities. This has been a cost effective way to raise awareness in around 50 primary schools in the region, but how about the other 85 in the surrounding Jämtlands län?

5.2 Weaknesses

Like creating a strong motivation among the teachers that attended all three workshops, the ambitious design of the pilot has also resulted in a high drop-out rate. Of the initially 55 teachers at Workshop 1, around 40 showed up at Workshop 2 and finally with 25 teachers remaining at Workshop 3. The high tempo in the activities has scared off more than half the initial group and some of the remaining teachers found the last workshop a bit stressful and that it was hard to follow all the exercises in the Scratch environment. From the initial heterogeneous teacher group the remaining 25 had a stronger orientation toward Natural sciences and Mathematics. The coming reformation of Swedish primary school must be a 'Digitalisation for All' and also region-wide as well as nation-wide.

Several teacher at the last workshop brought up their worries about technical issues and how new learning content should be created. Most participants agreed upon the necessity of follow-up activities in the autumn, which also was promised by IT-advisory A. But how could the successful pilot activities with Teacher T be replicated? There is only one Entrepreneur E, and he has also an IT-company to run. Play-based learning, Scratch and Python are relatively uncomplicated concepts to handle, but who should take care of software and hardware issues? One teacher suggested that Entrepreneur E can develop a series of learning activities that are shared online but to achieve sustainability there ought to be another model.

5.3 Opportunities

Learning content was a general worry at Workshop 3 and since there are no realistic single provider the idea of sharing seemed to be the best solution. There exists of course a plethora of code resources on the Internet, but who should evaluate and customise. An online platform with possibilities to share content as well as experiences and lessons learnt is appealing. However, an online platform or a web portal are not built without knowledge and funding.

A fundamental concept in the Östersund model is inter-school collaboration and not only primary schools sharing ideas and resources. An opportunity is to give secondary school students assignments where they produce and evaluate some of the learning content for primary school. Especially for students on technical programmes this would be realistic real world projects that have a good probability to be successful. In the same way could these projects be designed and supervised by student on university programmes on computer

science or informatics. The necessary professional development should not only include primary school and with the Mid Sweden University in Östersund academia has to take action as well.

With the successful pilot it must be possible to get further funding to create a networked and open solution for professional development. The concept that has been discussed has been named as a *NOOC*, an acronym that is built around the idea of a Networked Open Online Collaboration. A concept that should avoid the top-down MOOC design and instead be built bottom-up based on the involved stakeholders' needs. In the discussions after Workshop 3 some teachers pointed out their doubts for a pure online solutions and rather argued for some kind of blended design including shorter face-to-face sessions.

5.4 Threats

Experiences from other successful pilots in primary school shows that the scaling up is not always the next natural step where success guarantees further funding and sustainability. Without further funding it seems hard to find how the suggested model can be implemented region-wide to fulfil the idea of 'Programming for All'. If professional development stays focused to the city region of Östersund the model could not be seen as successful. If one of all discussed factors should be seen as the most crucial, it must be professional development. Teachers programming proficiency and CT knowledge must reach higher levels than what can be achieved in three workshops, otherwise the student activities will be suffering. The idea must be teachers learning from each other to create content and activities where students can learn from each other.

Primary schools heterogeneous ICT equipment is another issue that needs to be addressed, there is far from one computer per child and as an example one primary school only has 15 computers for 80 students.



Image 3. Workshop 1, high tempo activities with a low resolution projector

Several schools also have a mix of computers and iPad tablets, where tablets are not always the optimum for programming but otherwise very popular since they can be used for taking pictures and capture sounds in outdoor activities. There exists probably a general need for better equipment if primary school should be digitalised in an efficient way. An example is the old and a bit blurry projector that was used in the workshops.

6. Conclusion

The pilot has been successful both for student sessions and teacher workshops but the high tempo and the challenging activities have resulted in a high teacher drop-out. Follow-up sessions in the autumn are welcome but here is a need for further discussions to figure out how all high quality activities in the pilot can be replicated region-wide. Furthermore, it is not always that a successful primary school pilot necessary will get further funding to scale up (Mozelius, Rahuman & Wikramanayake, 2012).

Authors' recommendation is to support professional development in a networked open online collaborative solution, a NOOC, involving shorter face-to-face sessions between the various stakeholders. Sharing, content and experiences, is crucial and something that also could be part of a NOOC. Finally, an inter-school exchange where higher grade students develop learning content for primary school seems like a promising concept.

7. Future work

An important follow-up would be to analyse the official evaluation and to compare outcomes and teacher attitudes with other similar initiatives in other countries. The Östersund pilot is in several aspects unique in Sweden, but there are well-designed initiatives in countries like New Zealand and Finland to get inspiration and best practices from. Authors' long term goal is to elaborate, design and implement the described NOOC concept to enable a networked open online collaboration for the various stakeholders in the Östersund model. The upcoming digitalisation of primary school is a complex process where academia has an important, but not necessary leading role.

References

- Allsopp, B. B., & Ejsing-Duun, S. (2016). Programming Concepts in Playful Programming Products. In *10th European Conference on Games Based Learning: ECGBL 2016* (p. 1).
- Bennedsen, J., & Caspersen, M. E. (2007). Failure rates in introductory programming. *ACM SIGCSE Bulletin*, *39*(2), 32-36.
- Bell, T., Duncan, C., & Atlas, J. (2016). Teacher Feedback on Delivering Computational Thinking in Primary School. In *Proceedings of the 11th Workshop in Primary and Secondary Computing Education* (pp. 100-101). ACM.
- Bolstad, R. (2016). Game-Coding Workshops in New Zealand Public Libraries: Evaluation of a Pilot Project. *New Zealand Council for Educational Research*.

Brennan, K., & Resnick, M. (2012). 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).

Chomsky, N., & Schützenberger, M. P. (1963). The algebraic theory of context-free languages. *Studies in Logic and the Foundations of Mathematics*, 35, 118-161.

Denner, Werner and Ortiz (2011). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts?, *Computers & Education*, Volume 58, Issue 1, January 2012, 240-249,

Duncan, C., Bell, T., & Atlas, J. (2017). What do the Teachers Think?: Introducing Computational Thinking in the Primary School Curriculum. In *Proceedings of the Nineteenth Australasian Computing Education Conference* (pp. 65-74). ACM.

Ebbeck, M., Yim, H. Y. B., & Lee, L. W. M. (2013). *Play-based learning* (pp. 185-200). Allen & Unwin.

Fernández-Oliveras, A., & Oliveras, M. L. (2015). Conceptions of science, mathematics, and education of prospective kindergarten teachers in a play-based training. *International Journal on Advances in Education Research*, 2(1), 37-48.

Gibson, J. P. (2003). A noughts and crosses Java applet to teach programming to primary school children. In *Proceedings of the 2nd international conference on Principles and practice of programming in Java* (pp. 85-88). Computer Science Press, Inc.

Gomes, A., & Mendes, A. J. (2007). An environment to improve programming education. In *Proceedings of the 2007 international conference on Computer systems and technologies* (p. 88). ACM.

Heintz, F., Mannila, L., & Färnqvist, T. (2016). A review of models for introducing computational thinking, computer science and computing in K-12 education. In *Frontiers in Education Conference (FIE)*, 2016 IEEE (pp. 1-9). IEEE

Johnson, J. S., & Newport, E. L. (1989). Critical period effects in second language learning: The influence of maturational state on the acquisition of English as a second language. *Cognitive psychology*, 21(1), 60-99.

Kynigos, C. (1992). Insights into pupils' and teachers' activities in pupil-controlled problem-solving situations: A longitudinally developing use for programming by all in a primary school. In *Mathematical Problem Solving and New Information Technologies* (pp. 219-238). Springer Berlin Heidelberg.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005, June). A study of the difficulties of novice programmers. In *ACM SIGCSE Bulletin* (Vol. 37, No. 3, pp. 14-18). ACM.

Lam, G., & Rossiter, D. (2016). Interactive Python Programming for Beginners. *Interactive Python Programming for Beginners*, 248.

Lawanto, K., Close, K., Ames, C., & Brasiel, S. (2017). Exploring Strengths and Weaknesses in Middle School Students' Computational Thinking in Scratch. In *Emerging Research, Practice, and Policy on Computational Thinking* (pp. 307-326). Springer International Publishing.

Malliarakis, C., Satratzemi, M., & Xinogalos, S. (2013). Towards a new massive multiplayer online role playing game for introductory programming. In *Proceedings of the 6th Balkan Conference in Informatics* (pp. 156-163). ACM.

Maloney, J. H., Peppler, K., Kafai, Y.B., Resnick, M. and Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. In *Proceedings of the 39th SIGCSE technical symposium on Computer science education*. ACM, 367–371.

Maloney, J., Resnick, M., Rusk, N., Silverman, B. & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*

Meerbaum-Salant, O., Armoni, M. & Ben-Ari, M. (2010). Learning computer science concepts with Scratch. In *Proceedings of the Sixth International Workshop on Computing Education Research (ICER '10)*(pp. 69–76). New York: ACM.

Mozelius, P., Rahuman, K., & Wikramanayake, G. (2012). Two Years of One-to-one Computing in Sri Lanka—The Impact on Formal and Informal Learning in Primary School Education. In *Proceedings of World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education (ELEARN)* (pp. 1192-1201).

Mozelius, P., (2014). Two Promising Approaches to Game-based Learning in Programming Courses. In: *Lucas Mitchell ed. Computer Games: Technology, Educational Uses and Effects on Cognitive Development*. Nova Science Publishers, Inc.

Olsson, M., & Mozelius, P. (2015). Visualization of concepts and algorithms in programming education-A design theoretic multimodal perspective. In *Proceedings of the International Conference on e-Learning, ICEL* (pp. 257-264).

Ozoran, D., Cagiltay, N. E. & Topalli, D. (2012). Using Scratch in introduction to programming course for engineering students. In *Proceedings of 2nd International Engineering Education Conference (IEEC2012)* (pp. 125–132). Antalya, Turkey.

Papert, S. (1993). The children's machine. *TECHNOLOGY REVIEW-MANCHESTER NH-*, 96, 28-28.

Perkovic, L. (2015). *Introduction to computing using Python: An application development focus*. Wiley Publishing.

Pickton, D. W., & Wright, S. (1998). What's swot in strategic analysis?. *Strategic change*, 7(2), 101-109.

Python.org (2017). "What is Python" (Accessed 21/ 05/ 2017) from: <https://www.python.org/doc/essays/blurb/>

Sanchez, E., Piau-Toffolon, C., Oubahssi, L., Serna, A., Marfisi-Schottman, I., Loup, G., & George, S. (2016). Toward a Play Management System for Play-Based Learning. In *European Conference on Technology Enhanced Learning* (pp. 484-489). Springer International Publishing.

Sipitakiat, A., & Nusen, N. (2012). Robo-Blocks: designing debugging abilities in a tangible programming system for early primary school children. In *Proceedings of the 11th International Conference on Interaction Design and Children* (pp. 98-105). ACM.

Scratch (2017). "About scratch", (Accessed 20/ 05/ 2017) from: <https://scratch.mit.edu/about>

Stacey, R. (1993). *Strategic Management and Organisational Dynamics*, Pitman, London.

Swedish National Agency for Education (2017). "Digital kompetens och programmering ska stärkas i skolan" (Digital competency and programming should be strengthened in school) (Accessed 18/ 05/ 2017) from: <https://www.skolverket.se/laroplaner-amnen-och-kurser/nyhetsarkiv/nyheter-2016/nyheter-2016-1.247899/digital-kompetens-och-programmering-ska-starkas-i-skolan-1.247906>

Wiedenbeck, S., Labelle, D., & Kain, V. N. (2004). Factors affecting course outcomes in introductory programming. In *16th Annual Workshop of the Psychology of Programming Interest Group* (pp. 97-109).

Wilson, A., Hainey, T., & Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. In *European Conference on Games Based Learning* (p. 549). Academic Conferences International Limited.

Yukselturk, E., & Altıok, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789-801.