Informatics in Education, 2020, Vol. 19, No. 4, 569–587

DOI: 10.15388/infedu.2020.25

Teaching Computing without Computers: Unplugged Computing as a Pedagogical Strategy

Leonard BUSUTTIL, Marquita FORMOSA

Department of Technology and Entrepreneurship, Faculty of Education, Malta
e-mail: leonard.busuttil@um.edu.mt, marquita.c.formosa.17@um.edu.mt

Received: June 2020

Abstract. This paper investigates unplugged computing as a formal pedagogical strategy to teaching computing to a Maltese secondary class of Year 9 students. It aims at identifying the effectiveness of this pedagogy outlining the strengths and weaknesses in its application, taking into consideration the level of attainment for students, as well as the impact on teachers' lesson preparation. This research study is based on the delivery of five unplugged activities. It analyses students' reaction when exposed to such unplugged activities to assess the viability of using this pedagogy when teaching computing concepts in a formal setting. The study concludes that unplugged computing is an effective pedagogical strategy that helps attain a high level of engagement and student involvement, encouraging teamwork and collaboration. Students experience a wide attention span and good retention through the constant link of computing scenarios to real-life examples and the use of tangible non-computing related objects. Notwithstanding, the study also identifies certain limitations of this pedagogy, mainly that it is not sufficient as a standalone pedagogy, but needs to be applied in conjunction with other pedagogies to be able to cover all the expected learning objectives of the curriculum.

Keywords: unplugged, computational thinking, computing, secondary school.

1. Introduction

Teaching computer science in school is quite a challenge, which is evident from both the learners' as well as the educators' perspective. Unfortunately, many students view computer science negatively; some feel that this field lacks excitement and is considered monotonous. There's the impression that professionals in the field are required to spend endless time in front of a computer without any actual interaction (Taub, Armoni, & Ben-Ari, 2012).

From the other side of the class, teachers find Computing too hard for most students to comprehend. They feel that the abstract part of Computing is complex to teach, especially with examples like algorithms, programming and computational thinking skills (Sentance & Csizmadia, 2017). Another challenge is that the teacher needs to be able to help the student understand what the computer is doing behind the scenes. Students might know how to use a computer before starting computer science at school, but they do not know how it works. Guzdial (2015), states that the "inhumanness of computers makes them harder to understand" (Guzdial, 2015, p. 32).

A pedagogical strategy, referred to as unplugged computing, also known as CS Unplugged, aims at addressing challenges faced when teaching computing concepts. This strategy involves the use of practical activities to help students understand the key principles of computer science without using an actual computer (Bell, Alexander, Freeman, & Grimley, 2009).

The term CS Unplugged originated in a series of activities published online and later collated into a free online book primarily to be used in outreach activities (Bell, Witten, & Fellows, 1998). A teachers' edition version was later published in 2000 and updated in 2002 and 2015 (Bell, Witten, & Fellows, 2015). This newer version still did not assume that computer science would be part of a curriculum (Bell & Vahrenhold, 2018). Unplugged computing activities vary substantially when it comes to preparation and delivery since there are several genres of such activities that one can adopt. The types of unplugged activities vary from Magic Tricks, Storytelling, Music, Dance, Kinaesthetic Activities, Code Breakers, Contests and Puzzles amongst other genres (Bell & Vahrenhold, 2018; Curzon & McOwan, 2017; Dagienė, Futschek, & Stupurienė, 2019; Delal & Oner, 2020; Ghafoor, Brown, Rogers, & Hines, 2019; Rosamond, 2018; Sapientia, n.d.; Shelton, 2016; Waite, Maton, Curzon, & Tuttiett, 2019). Unplugged activities are also used as part of other activities such as in the algomotricity strategy where students work in groups to verbally guide a blindfolded robot through a simple path and later use a slightly modified version of Scratch to write code to guide a sprite through a maze (Lonati, Malchiodi, Monga, & Morpurgo, 2015).

Unplugged computing has several benefits, such as, being independent from the use of computer hardware, enabling the use of activities that are easy to implement, and supports the teacher in teaching problem-solving and computational thinking skills using manipulatives. Such activities are meant to inspire group work, as well as engaging students as much as possible (Cortina, 2015)

This pedagogical strategy is being used in quite a number of countries across the globe, such as in various European countries, the US and Korea (Bell *et al.*, 2009). Several studies were carried out to assess its application, amongst which, Rodriguez, Rader, and Camp (2016) explored the teaching of Computational Thinking concepts in two American middle schools whilst Rijke, Bollen, Eysink, and Tolboom (2018) discussed the introduction of unplugged activities in primary schools. Rodriguez *et al.* concluded that CS unplugged activities increased students' interest in the subject and their level of engagement in class (Rodriguez *et al.*, 2016). Rijke et al conclude that unplugged activities have a future in primary schools, especially in the upper primary education (grade 4 through 6) since at this age students start to make progress in their abstract reasoning**.** Tonbuloğlu & Tonbuloğlu (2019) stated that unplugged coding activities had a positive effect on the improvement of computational thinking skills of the students. They also stressed that students engaged in unplugged activities were motivated and participated in class activities.

This study aims to assess CS Unplugged as a pedagogical strategy in teaching national computing curriculum topics to secondary school students. In Malta, where this research was conducted, students can choose to study computing as from year 9, aged 13, and sit for a national exam aged 16. Through this study, the authors wanted to identify if unplugged computing pedagogy can be used to teach areas of the curriculum in a formal setting and the benefits or pitfalls that the use of this pedagogy had when applied in such a setting. In this paper we try to answer the following research questions:

1. How effective is the unplugged computing pedagogy when used in a formal school setting?
2. What are the drawbacks of using this pedagogy in a formal school setting as part of a prescribed syllabus?

2. Method

Qualitative case study methodology (Merriam, 2009) was used in this study. Given that the study focuses on observing the reaction of a class of students when exposed to CS Unplugged, this research methodology was deemed appropriate since case studies "enable researchers to examine data at the micro-level" and "provide better insights into the detailed behaviours of the subjects of interest" (Zainal, 2007, p. 5)

A teaching pack of unplugged activities were prepared for a year 9 class made up of 13 to 14-year-old children. These students were in their first year of studying Computing in a secondary school.

Twenty-five children took part in these activities, led by one of the researchers, whilst the regular teacher and second researcher stayed in class and acted as observers. In order to allow for intercoder reliability and to enhance consistency between observers, an observation schedule was used by the observers (McKechnie, 2008). The observation schedule asked the observers to comment on the engagement levels of the students in class and on the right level of difficulty that the activity presented to the audience. Given the qualitative nature of this study, the observation schedule also included space to record subjective data which informed the study. Following each activity, the researchers and the regular teacher discussed their observations of the learning that ensued in each session. The discussions were audio-recorded and later transcribed.

2.1. *Participants*

The participation of secondary school students depended on whether they have chosen Computing as one of their options at school. Year 11 students were not considered to take part since their academic year finishes earlier than the other years. Hence, the class needed to be any Computing class either from Year 9 or from Year 10.

Cortina (2015) highlights the fact that CS unplugged allows the learners to “learn through activity outside of a standard academic setting” (Cortina, 2015, p. 26). To be able to conduct the activities in a different learning setting yet remain on the school grounds since the sessions took place according to the school timetable, it was vital that the school would allow the possibility to conduct such activities outside the computer lab if necessary, such as in a gymnasium or an outdoor space.

The school’s resident Computing teacher had to be qualified and experienced. During the five, eighty-minute long activities, the resident Computing teacher was invited to be present in class and act as a co-researcher by observing the students and record such observations in an observation schedule. Hence it was important that through his/her experience, the teacher would be able to give constructive feedback about the application of this pedagogical strategy in the classroom and highlight benefits or pitfalls that would seem evident.

While discussing the prospective topics, it transpired that CS unplugged could not be applied to teach the topics scheduled according to the teacher’s scheme of work. During the time frame of delivering the activities, logic gates, input and output devices and conditionals (using Java) were planned to be covered in class. However, while researching unplugged activities for such topics, we could not find any appropriate activities to cover all curriculum learning objectives for logic gates and input and output devices. This unveiled one of the limitations of CS unplugged – that it cannot be used for all the topics in the curriculum.

Subsequently, it was agreed with the teacher that input and output devices would not be tackled through these activities and will be replaced by covering the topic of parity check, for which CS unplugged could be used. With respect to logic gates, CS unplugged was only used to introduce the topic whilst conducting an activity on binary addition in class.

Furthermore, an activity covering computational thinking skills was used to introduce students to such skills, even though these are not specific requisites of the curriculum. Notwithstanding, the teacher agreed with covering such skills since they are considered essential for computing students.

A readily available CS unplugged activity was used to cover conditionals, in line with the teacher’s scheme of work.

2.2. *Data Analysis*

For this study, the thematic analysis approach was used to analyse the data collected through the activities to address the research questions of this study. In this analytic method, one needs to accustom himself well with the collected primary data to be able to take note of any possible “patterns” present within the data collected (Braun & Clarke, 2006, p. 81). These patterns are then analysed for similar characteristics to determine different groups of data referred to as “codes” (Braun & Clarke, 2006, p. 87). These codes are analysed once again for further similarities so that these are sorted according to “themes” (Braun & Clarke, 2006, p. 87). “A theme captures something important about the data about the research question, and represents some level of patterned response or meaning" within the analysed data (Braun & Clarke, 2006, p. 82). The process of identifying themes will be concluded once the researcher has exhausted any additional groupings such that themes "are internally coherent, consistent and distinctive" (Braun & Clarke, 2006, p. 96). At this point, all primary data for each theme will be analysed and assessed for formulating the report's finding (Braun & Clarke, 2006).

3. Unplugged Activities

The unplugged activities covered in this research study consisted of five activities targeting separate areas of the year 11 Computing curriculum, specifically logic gates, problem solving, error handling through parity, conditionals and adding numbers through logic circuits. After each activity students were asked to complete a worksheet to assess the knowledge gained through the activity. The five activities used were:

- Logic Gates through Twister based on the Twister game (Ward, n.d.).
- Building a tower based on the activity by Caldwell (2016).
- Parity magic using cards adapted from Bell, T., Witten, I., & Fellows, M. (2015).
- Conditionals using cards adapted from the Hour of Code activity conditionals with cards (Hour of Code, n.d.).
- Adding numbers adapted from the BBC series The joy of logic (Cliff, 2013).

3.1. *Logic Gates through Twister*

The aim of this activity was to help students familiarise themselves for the first time with the concepts of the Logic OR, AND and NOT together with their respective inputs and outputs. To introduce students to the logic of the AND, OR and NOT gates, the Twister game(Ward, n.d.) was used. This game is normally played by having a coloured mat as shown in Fig. 1, and a spinner to determine commands by random. Each player spins the spinner to determine which body part needs to be used from the hands and the feet, and on which colour. This is a multiplayer game whereby the winner is determined by being the only one left standing.

Fig. 1. Twister mat.

The Twister game was played in the school's gymnasium, but the rules were slightly amended. The spinner was not used and instead, students were shown commands such as the following:

- Red: NOT (Right Hand).
- Blue: (Left Hand) AND (Right Hand).
- Yellow: (Right Hand) OR (Left Hand).

For the NOT commands, they were expected to place any body part on the specified colour apart from the one mentioned in the brackets of the NOT command. When an AND command was shown, students were expected to use both body parts mentioned in the command, whereas for the OR commands, students were expected to choose one of the mentioned body parts. The game was played using twelve commands with two different difficulty levels. The first set consisted of six single commands similar to the above. The second six commands had a combination of either logic OR and AND, logic OR and NOT, or logic AND and NOT. After playing the game in the gym, the students were asked to complete a worksheet. The worksheet contained the same commands that were used in the game. Students were requested to mark the body parts to satisfy the command.

3.2. *Building a Tower*

This activity was meant to cover the problem-solving process through specific Computational Thinking skills – problem-solving, pattern recognition, evaluation and Decomposition. This activity was adapted from Caldwell & Smith (2016).

The activity started by eliciting from the students what they know about the term persistence. A discussion evolved on episodes in their lives where persistence needed to be present, such as when learning how to ride a bike. Eventually, the highlight was on the importance of learning when we fail and being persistent even though a task might seems to be impossible.

The students were invited to split into two groups of four and two groups of

three to be able to work together to build a tower. The rules for this challenge were for the students to build a tower by using the materials provided, and that the tallest tower would win.

For this activity, each group was given 30 spaghetti sticks and 30 marshmallows. The main aim behind this activity was that the students needed to be persistent while working together to build the tallest standing tower. The students were given 30 minutes to build the tower. During this part of the activity, students were continuously observed and encouraged to continue persisting on building the tallest tower. Following the activity, a discussion was held to be able to link the steps in building the tower with specific skills. Firstly, the importance of planning was highlighted. Pattern recognition was then brought to the discussion to outline that when students started "to recognise the best way to join the spaghetti beams" they had to "repeat this method to build a stronger structure" (Caldwell & Smith, 2016, p. 184). Students needed to evaluate their work continuously, which might have also decomposed the challenge into smaller tasks to be tackled by each member separately (Caldwell & Smith, 2016). Students were also introduced to the term problem solving as a skill which featured a lot in this task. During the discussion, other examples were shared for the introduced computational thinking skills so that students can relate further with their everyday life scenarios.

3.3. *Parity Magic Using Cards*

To introduce the topic, a magic trick was used. As soon as the students entered the classroom, the researcher mentioned that a magic trick will be performed and asked for two assistants. Twenty-five small paper squares having one side black and the other side white were distributed to one student and 25 small magnets to the second student assisting in this task. The two students were asked to form a 5 x 5 grid by attaching the small papers on the whiteboard with the magnets provided. They were asked to use whichever side they wished, either black or white. The grid formed by the students can be seen in Fig. 2.

Following the students' grid, the researcher added a row at the bottom of the grid and a column to the right of the grid turning the 5X5 grid into a 6X6 grid. The extra row and column were added in a way that the number of black and white squares were even in all the rows and columns. The researcher left the class and asked a volunteer to overturn one of the cards, irrespective of the position whilst the researcher was not in class. Once the researcher came back into class the flipped card was identified using the Parity rules. The students were not told the algorithm followed but were asked to try and figure out how the flipped square was identified. Some of the children managed to identify the patterns and identified the routine used. The trick was then explained to the rest of the class. A discussion ensued to evaluate the limitation of parity in situations where more then one card is flipped at the same time, without breaking the validity of the parity check.

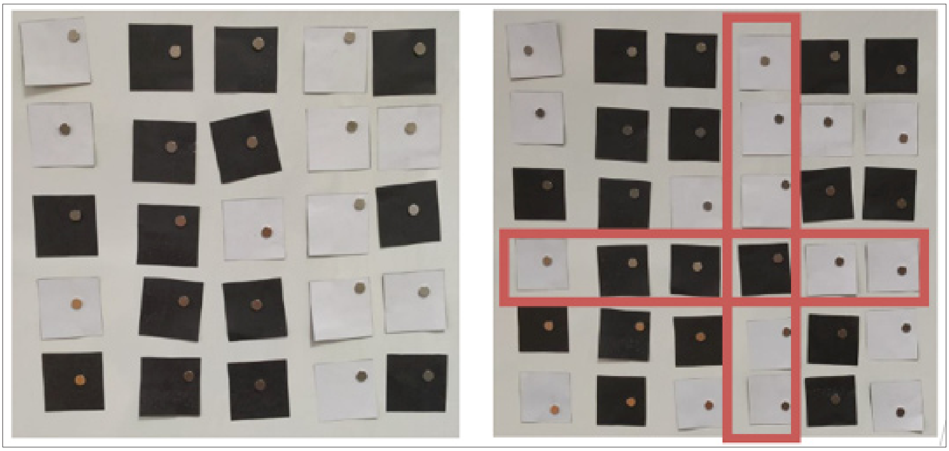


Fig. 2. Parity Magic using cards.

3.4. *Conditionals Using Cards*

For this part of the activity, a game was played using a deck of cards. Students had to sit around the teacher's table and play this game in pairs. The deck of cards were all facing down and each pair of students had to flip one card in each round. Six if/else game rules, were adapted from the website code.org. These rules were shown to the students, one for each round, as depicted in Fig. 3 in the case of Round 1.

These were left available for students as a reference during this activity. According to the rule for Round one, if the flipped card is red, then the player would earn one point, otherwise, if black, the player would earn two points. For every command shown to students, one pair of students was asked to elicit the meaning of each rule. For each round, each pair of students was given points according to the commands. The commands for Round 2 and Round 3 were the same but presented differently. These are displayed in Fig. 4.

The main idea behind the above was to help students start getting familiar with the Java syntax when programming conditionals. The game rules for the rest of the rounds were based on the same concept. Following Round 6, all points for each team were added and the team with the highest number of points won the game.

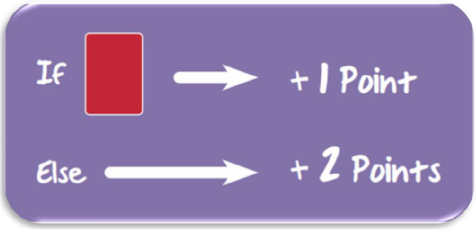
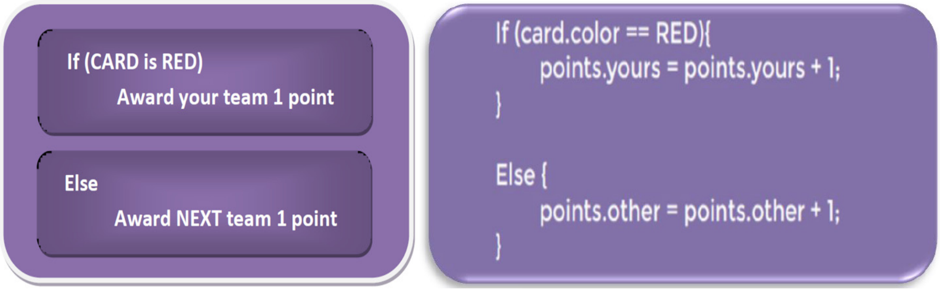


Fig. 3. Card Rules Round 1.



If (CARD is RED)
Award your team 1 point

Else
Award NEXT team 1 point

```
If (card.color == RED){
        points.yours = points.yours + 1;
}

Else {
        points.other = points.other + 1;
}
```

Fig. 4. Card rules listed in different formats.

3.5. *Adding Numbers*

In this activity, students had to be introduced to the Binary Addition using Logic Gates. They were already familiar with both topics but they were not knowledgeable about how binary addition can be done using Logic Gates.

The main idea for this activity was adopted the BBC series The joy of logic (Cliff, 2013). Students were taken to the school's gymnasium and instructed to represent logic gates and binary digits (bits), hence processing information like a computer. To be able to do this, blueprints of circuits were taped on the floor together with the AND, OR and NOT gates. The circuits varied from very simple ones with just one gate to the more complex circuits that carried out binary addition.

The circuit that was used to show students how the addition of two bits can be carried out using logic gates, was the half adder circuit. The half adder circuit is the one shown in the image Fig. 5.

Since students were only introduced to the AND, OR and NOT gates, the version of the half adder used only consisted of the OR, NOT and AND gates as in Fig. 5.

In order to process information like a computer, some students were asked to represent the logic gates while some students were asked to represent bits. The latter students would flow through parts of the circuit by following the taped directions on the floor and displaying their respective bits, i.e. either 1 or 0. For those students representing the logic gates, they were asked to wear orange high visibility vests, together with a sign stating the name of the gate, and were stationary on the place of the logic gate.

Students displaying the initial input bits would start walking on the tape until they reached a logic gate student. Depending on which binary bit is being displayed, the logic gate student determines which binary bit (1 or 0) ought to be displayed by another student standing in front of him representing bits. This latter student then starts walking on the tape displaying the determined bit by the logic gate student until he reaches a second logic gate student. The second logic gate student then determines the bit to be

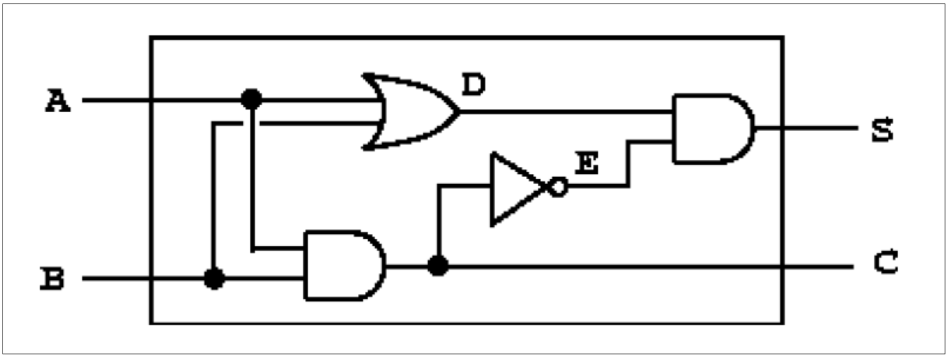


Fig. 5. Half Adder using OR, NOT and AND.

displayed by another student standing in front of him. This process continues until students work their way through the whole circuit and the final output bits are determined. During this process, one student was asked to take note of the initial input values being put into the circuit and the final output values, i.e. the sum and the carry values. This was done to construct a table of initial inputs and final outputs. Eventually, students were assisted in trying to identify the manipulation of the input bits and reach the conclusion that binary addition is taking place.

4. Findings

The observations noted by the researchers and the students' customary teacher, were analysed using thematic analysis. Each worksheet at the end of each activity was also used to assess students' knowledge after each activity. These too were used to gain an insight into the knowledge gained through the unplugged activities. Six main themes emerged from the thematic analysis of the observations and recorded discussions between the researchers and the class teacher, as well as the assessment worksheets worked by the students following each unplugged activity. In this section each theme will be explored and compared to the findings in literature. The themes will be linked the research questions outlined in the introduction section.

4.1. *Engagement*

One of the key aspects of CS unplugged is engagement whereby participants feel engaged in each activity by being attractive for them (Bell, Alexander, Freeman, & Grimley, 2008). From the class teacher's observations, it can be noted that students' engagement was present in all five activities that were delivered. This is clear from the class teacher's score on the level of engagement during each activity. It was noted that various aspects affected the students' level of engagement. The environment, the prior knowledge possessed by the students, the game like fun element and the hard fun (as described by (Kafai, 2018)) all contributed to the engagement theme. We will now delve into each aspect.

4.1.1. *Environment*

The environment played an important role in these activities. The twister, tower and binary addition activities were held in the school's gymnasium, while the conditionals and parity check activities were held in a computer lab.

It emerged that when activities were held in the gymnasium, students were more engaged. This was most evident during the Twister activity, especially when students were asked to remove their shoes in order to be able to play. This activity offered the possibility of learning an academic subject in a more physically active way. According to the class teacher, this activity managed to engage students holistically. Students were given the opportunity to learn in a different setting, hence being able to learn from "observations and experiences". This is an important characteristic of CS unplugged (Cortina, 2015, p. 26).

However, during the binary addition activity, we noted that the environment offered some drawbacks due to the lack of alternative resources. Since the activity was performed in the gymnasium, there were no portable boards or flip charts available to display the whole circuit being 'acted out' by students for binary additions.

Had such alternative resources been available, students might have been in a better position to better understand the nature and true purpose of the activity, as outlined by the class teacher. The logic circuit taped on the floor might have not been so clear for students to interpret their actions. This might have been mitigated if such alternative resources were available within the setting environment.

4.1.2. *Prior Knowledge*

Students' prior knowledge is another aspect that determined students' level of engagement. When students had no knowledge of the topic covered in the activity, i.e. the twister, tower, conditionals and parity check activities, students seemed to be more engaged.

On the other hand, in the binary addition activity, students were quite familiar with both logic gates as well as binary addition. Hence the only new knowledge for them was the fact that binary addition was being performed through logic gates. Therefore, since this activity was a consolidation of what they already knew, students seemed to be less engaged.

According to the class teacher's observation, this activity did not challenge students well enough, since it started with things that they were quite knowledgeable about. In fact, when the students were asked for feedback about this activity, several mentioned that it was too basic and plain. Some students even remarked that this activity was somewhat boring.

Feaster *et al.* (2011), highlights this aspect and hence consolidates that when students are already knowledgeable about a topic or aspect in computer science, they tend to be less interested in learning. This holds even though this pedagogical strategy is used since students would find such activities "less rewarding and unnecessary" (Feaster, Segars, Wahba, & Hallstrom, 2011, p. 252).

4.1.3. *Hard Fun*

The fun factor played an important part for students to be engaged in the activities. During the delivery of such activities, students regularly mentioned that they used to have fun while doing these activities since these were different from what they usually do during their regular Computing lessons. This was, in fact, coherent with the class teacher's observations. In her observation sheet for the Tower activity, she stated that "students were able to understand the somewhat abstract concept of computational thinking in a very fun and tangible way".

It was evident that the students were very much engaged when they were motivated to think. This was seen in the Tower activity, where students were provided with marshmallows and spaghetti to build the tallest standing tower. In this activity, students were encouraged to think outside the box to be able to reach their goal. When commenting about this activity, the class teacher mentioned that from all five activities, students were most intrigued by this one and were very eager to start this challenge. It was interesting to note the various methods that students came up with in trying to succeed in this activity. In the Tower activity, students were encouraged to work in teams to reach their goal whereby they needed to think creatively and work together. Both of these traits are vital aspects of CS unplugged (Cortina, 2015).

The Parity Check activity also concluded that it highly motivated students to think. The students were observed having fun but at the same time working actively to try to decipher how the researcher was guessing which card was turned and always getting this right. They confirmed this during a short feedback session at the end of the activity. When the extra row and column were added as part of this activity, one could hear some students remarking that the additional black or white squares were being placed randomly. This shows that students were already thinking of the possibility that there must be a pattern that was being used to determine whether to add a black card or a white one to each row or column.

The fact that students were motivated to think might also be a key motivator for students to learn. Nishida, Idosaka, Hofuku, Kanemune, & Kuno (2008) mention that CS unplugged proved to help in "enhancing students' motivations, thinking abilities and imaginations" (Nishida et. al, 2008, p. 246)

4.1.4. *Genres of Unplugged Activities*

For all five activities, different adaptations of the CS Unplugged pedagogy were used, in line with Caldwell's and Smith's work (2016). The Tower and Binary Addition activities adopted a kinaesthetic approach, whereby students learn through action. On the other hand, the Twister and Conditionals activities focused on playing games in class, while magic was used for the Parity Check activity.

These different genres played an important part in the students' level of engagement. The Tower activity was the most preferred kinaesthetic activity with students enjoying the fact that it was hands-on and that they had to come up with creative methods on how to build the highest standing tower.

Both the Twister and Conditionals activities encouraged curiosity among students. In fact, students started to question what was going to happen during class and how the game was related to Computing, especially when they saw the Twister mat set up for the Twister activity. These activities managed to keep students engaged and curious about what was the link to their academic subject.

For the Parity Check activity, it was evident that as soon as the researcher mentioned at the start of the lesson that the students were going to participate in a magic trick, students were immediately interested and paid more attention. They were also eager to be involved since almost all of the students wanted to participate as helpers. The trick was performed a number of times in order for more students to be able to take part.

These observations highlight the fact that this pedagogical strategy was quite effective when it comes to teaching secondary school students. This supports the fact that "the practical teaching materials with various games and hands-on activities" of this pedagogical strategy, make it more effective (Nishida *et al.*, 2008).

4.1.5. *Knowledge Retention and a Longer Attention Span*

The engagement aspect throughout the activities helped students to prolong their attention span and focus more on what was being covered in class. This confirms the conclusions reached by Rodriguez *et al.* (2016) that such CS Unplugged lessons have proved to be "sufficiently engaging to keep kids' attention in the lesson" (Rodriguez *et al.*, 2016, p. 100).

Such activities also improved students' retention of the academic knowledge acquired. This was evident from the fact that although the last activity was delivered after a two-month interval the students still remembered clearly what had been covered in the previous activities. This further reinforces the findings of Murphy Paul's work (2015, when she states that using CS Unplugged activities "may increase the odds that what they learn later will be absorbed in a deep lasting way" (Murphy Paul, 2015, p. 25).

The class teacher emphasized that the CS Unplugged activities were important for such students' retention. When introducing the Parity Check, as the students' customary teacher would normally cover the parity bit by giving examples of both even and odd parity checks, without going into too much detail. She admitted that this might be at the expense of students not remembering the concept well enough. Following the Parity Check activity, she remarked that the fact that students had to think on how to solve the magic trick might help them to memorise the parity check better. The class teacher also remarked that she will be willing to use such activities to teach future students given that such activities "help students to recall the knowledge in an easier manner".

4.2. *Level of Difficulty*

It was noted that the level of difficulty in all five activities was quite adequate for students. The class teacher mentioned that all activities started off with a challenge which all students needed to tackle. During these activities, there were some students who mastered the challenge while some others struggled or failed to complete it. A clear example was the Tower activity, where three out of four groups managed to build the tower but the fourth group did not. This shows that not all students can equally tackle the same level of difficulty and hence the importance of having different exercises with incremental levels of difficulty.

Similarly, in the Binary Addition activity, the class teacher pointed out that the introductory circuits could have been avoided since these were too simple for the students' level of current knowledge. This would have made more time available for a better explanation on how bits were being manipulated by the more complex logic circuits in order to perform binary addition. This proved to be somewhat difficult for some students to understand, as noted when analysing the responses to the students' worksheets

Overall, it was noticed that students found almost all these activities challenging but which at the end, most students managed to master. This portrays one of the strengths of this pedagogical strategy; the fact that it allows teachers to build each activity according to the students' capability levels. Feaster *et al.* (2011) mention that CS Unplugged activities can be adapted to "suit a range of learning levels", hence adapting the activity's level of difficulty according to the audience in order for it to be somewhat challenging (Feaster *et al.*, 2011, p. 248).

4.3. *Clear Instructions*

An important theme that emerged from the observations is the importance of having clear instructions conveyed to the students for them to understand what is being expected from them. This was confirmed by the class teacher, who remarked that students seemed to understand well the necessary instructions they needed to follow in almost all five activities.

During the Parity Check activity, three out of fourteen students managed to anticipate the pattern used to perform the magic trick. According to the class teacher's observations, this was due to the fact that the researcher managed to explain clearly the previous examples for which students were very attentive.

In her observations for the Twister activity, the class teacher mentioned that students understood the concepts of the AND, OR and NOT gates quite easily. In her opinion though, in order to facilitate the game's association with computer science, it would have been more beneficial had it been limited to only two body parts rather than four, for example, left hand and right hand only, rather than including right foot and left foot as well. This would have made it simpler and eliminate confusion, especially with the NOT gate. According to the observer, the confusion with the NOT gate became evident when the students were shown the command NOT (Left Hand). Some students thought the answer would be right hand while for others the answer had to be anything but the left hand. The class teacher believed that since binary has two states, having just two body parts in the activity would have made the transition to binary easier. For example, if students are presented with the statement NOT (Left Hand), then the result would have to be right hand. In the class teacher's viewpoint, this is more easily assimilated to Computing, such that if we have NOT (0), then the answer is 1.

Although this observation by the class teacher was a valid point, we are of the opinion that the presented Twister activity showed that students also understood the concept of the logic NOT in view of Boolean expressions. In this case, students will be asked to choose from more than just two options. Hence this activity helped the students familiarise themselves with the NOT logic in other Computing aspects, not just binary.

4.4. *Time and Resources*

Different types of resources were needed for the five activities. Some were bought ready to use, such as the life vests for the Binary Addition activity and playing cards for the Conditional activity. Others required time to prepare them, such as cutting the black and white squares for the Parity Check activity together with all the worksheets for the five activities, except for the Conditionals activity worksheet that was taken from the source indicated.

Even though such activities required considerable preparation time to build such resources, from past teaching experiences, lessons with more conventional pedagogical strategies require more or less the same preparation. Considering that the prepared resources can eventually be shared and used over and over again, as suggested by Cortina (2015), CS Unplugged activities are "easy to present" and can be collaborated with other teachers. Working together with other Computing teachers can effectively decrease the preparation time for such resources (Cortina, 2015, p. 25).

From the five activities, the Binary Addition activity was the most time consuming to prepare and dismantle. Given the nature of the activity, it entailed taping the shape of the circuit on the gymnasium floor and which took some 30 minutes to set up and another approximately 15 minutes to dismantle. This might not be feasible to have such activities on a day-to-day basis, especially when the teacher has a heavier lesson load, which might not allow the teacher to have such time before and after each lesson.

4.5. *Real World Scenarios*

It was deemed important to relate the subject material to real world examples which the students are familiar with in order to help students comprehend better the bigger picture.

In the Twister activity, students were quite familiar with the game and this allowed them to focus on learning and understanding the AND, OR and NOT commands rather than learning the game. Students were subsequently introduced to such logic gates topic, which through the CS Unplugged activity, students seem to have understood quite well. This was also confirmed by the class teacher when she covered the material in more detail at a later stage following this activity.

The class teacher highlighted the fact that in teaching Computing, sometimes simple concepts are difficult to convey to an audience, such as conditionals. This was very successful with the Conditional activity, which the class teacher suggested that it was very innovative and the game itself helped students to conceptualise better the concept of conditionals and their link to programming.

Nishida *et al.* (2009) mention the fact that games are very useful in CS Unplugged. They raise students' "interest, curiosity and motivation" (Nishida *et al.*, 2009, p. 232) to learn. Such aspects were evident in both the Twister as well as the Conditional activity, since on both occasions students immediately started to ask questions on how the games were related to Computing.

When students had to think about how conditional statements can be used in real life through the conditional activity, students were able to identify that features from programming are used in everyday life. This aspect might have portrayed programming in a logical way to understand, rather than emphasizing solely on the programming syntax.

4.6. *Collaboration and Teamwork*

CS Unplugged activities are meant to "encourage collaborative work" (Cortina, 2015, p. 25). In most of the activities conducted in this research study, students had the opportunity to discuss and solve problems in a group such as during the Tower activity and the Parity Check activity. This is a key aspect of CS Unplugged activities as students need to be "motivated to work towards a goal, either as part of a team, or to try to find a better/ faster solution than another group" (Bell *et al.*, 2008, p. 131).

However, not all students were equally ready to work in teams. For example, in the tower activity, only three out of the four teams managed to successfully work as a group. The class teacher's observations on this point suggested that the main problem with the fourth group was the lack of teamwork, potentially because they were not comfortable working together. Hence, the teamwork aspect in this activity was not achieved by all students.

5. Discussion

In the introduction section of this paper we set out the research questions that guided us throughout this research. In this section we are revisiting the questions in light of the findings outlined in section 4 above.

5.1. *Effectiveness of Unplugged Pedagogy in a Formal School Setting*

A key finding of this research was that the participants felt engaged in each of the activities that were conducted as part of the formal class syllabus. This reinforced the findings of other research (e.g. Bell *et al.*, 2008; Threekunprapa & Yasri, 2020) . This engagement led students to be more focused during the lessons and for the knowledge gained to be more long lasting. The out of class, yet in school environment, where some of the activities were conducted, proved to be an important factor that contributed to the effectiveness of this pedagogy in a formal school setting even though drawbacks were associated with this setting. The unplugged activities proved to be most effective when used as the introductory activities to new topics. This too reinforced findings in other studies (Feaster, Segars, Wahba & Hallstrom, 2011). There are multiple genres of unplugged activities and all the genres used in this research were deemed to be fun by the students. Although students described these activities as fun, the students were still engaged in a thinking process conducive to learning. This led us to describe these activities as being not just fun but hard fun.

5.2. *Drawbacks of Using the Unplugged Pedagogy as Part of a Prescribed Curriculum*

One of the main motivations on conducting this study was to identify drawbacks of using computational thinking in a formal setting. From the onset the difficulty to identify unplugged activities for all the areas of the curriculum was noted by the researchers. Teachers planning to use this methodology must choose well the curricular areas where the use of this methodology would be mostly beneficial.

Although as outlined above using a different physical learning setting and different resources raised the level of engagement, this too posed additional drawbacks. Conducting activities out of the classroom means that the teacher does not have access to resources which are usually available in class such as a whiteboard or flip chart.

Each activity required preparation time to not only create resources but to also set the props to be used in the activities. Some activities required circuit blueprints to be drawn and taped on the floor, together with the different labels for students to wear. Such additional time to set up and dismantle activities might not always be possible for a teacher in a formal school environment, especially when having back to back lessons with different classes. Access to space is very important especially for the kinaesthetic activities and space might not always be available in schools since such space is usually used by different subjects in school. Similarly, the preparation time required is not only to plan for the lessons but also to setup the props before the class. Unplugged activities generally take time to conduct and are quite hard to fit in the timetabled slots. The secondary school timetable is usually very rigid since subjects are taught by different teachers and flexibility to move lessons is not always afforded to teachers. Teachers also required preparation time needed to create the props to be used in the unplugged activities. Such preparation time can be reduced by adapting readymade resources found in games such as in the case of the twister activity.

6. Conclusion

The prepared teaching pack of five unplugged activities seemed to be very effective in the classroom. It appeared to be age-appropriate, and most of the activities seemed to be at the right level of difficulty for the students.

The Binary Addition activity might have been slightly too easy for students since they already knew most of the basics highlighting the suitability of unplugged activities to introduce topics rather than as a consolidation exercise. The CS Unplugged activities motivated students to think through various challenges, which in turn helped to increase the students' attention. The connection to real-world examples helped the students to establish a more apparent connection between the computer and the real world the students enjoyed most of the activities where the fun factor was evident in each activity. However, the fun factor on its own might not necessarily signify learning experiences. Hence, when using CS Unplugged, it is essential that students are allowed to enjoy the activity but the learning objectives are emphasised throughout this process. All students were very much engaged in all activities. The fact that CS Unplugged allows the teacher to use various ways to teach Computing, such as games and magic, helps in engaging students further. Using a different learning setting location and the use of unexpected resources to raise curiosity, such as playing cards and magic tricks, helped to increase this level of engagement. CS Unplugged promotes teamwork, which was highly noticed during each activity. Every Computing student should be encouraged to work in teams, and this pedagogical strategy enables such characteristic.

No pedagogy can be used in all scenarios and using unplugged activities in a formal educational setting was found to have its own set of setbacks. Notwithstanding the issues identified in this research, the benefits outweighed the drawbacks identified and the pedagogy was found to be beneficial especially when used to introduce complex topics, common to most secondary level curricula.

References

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2008). Computer science without computers: new outreach methods from old tricks. In: *Proceedings of the 21st Annual Conference of the National Advisory Committee on Computing Qualifications*.

Bell, T., Alexander, J., Freeman, I., & Grimley, M. (2009). Computer science unplugged: School students doing real computing without computers. *The New Zealand Journal of Applied Computing and Information Technology*, 13(1), 20–29.

Bell, T., & Vahrenhold, J. (2018). CS Unplugged How Is It Used, and Does It Work? In: *Adventures Between Lower Bounds and Higher Altitudes* (pp. 497–521). Springer.

Bell, T., Witten, I., & Fellows, M. (1998). *Computer Science Unplugged: Off-line activities and games for all ages*. Citeseer.

Bell, T., Witten, I., & Fellows, M. (2015). *CS Unplugged: An enrichment and extension programme for primary-aged students*. self-published.

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative Research in Psychology*, 3(2), 77–101.

Caldwell, H., & Smith, N. (2016). *Teaching Computing Unplugged in Primary Schools: exploring primary computing through practical activities away from the computer*. Learning Matters.

Cliff, D. (2013). How to build a computer out of children. Retrieved from `https://www.bbc.co.uk/programmes/p01m5xfs`

Cortina, T. J. (2015). Reaching a broader population of students through" unplugged" activities. *Communications of the ACM*, 58(3), 25–27.

Curzon, P., & McOwan, P. W. (2017). *The Power of Computational Thinking: Games, Magic and Puzzles to Help you Become a Computational Thinker*. World Scientific.

Dagienė, V., Futschek, G., & Stupurienė, G. (2019). Creativity in Solving Short Tasks for Learning Computational Thinking. *Constructivist Foundations*, 14(3).

Delal, H., & Oner, D. (2020). Developing Middle School Students' Computational Thinking Skills Using Unplugged Computing Activities. *Informatics in Education*, 19(1), 1–13.

Feaster, Y., Segars, L., Wahba, S. K., & Hallstrom, J. O. (2011). Teaching CS unplugged in the high school (with limited success). In: *Proceedings of the 16th Annual Joint Conference on Innovation and Technology in Computer Science Education*, 248–252.

Ghafoor, S. K., Brown, D. W., Rogers, M., & Hines, T. (2019). Unplugged Activities to Introduce Parallel Computing in Introductory Programming Classes: an Experience Report. In: *Proceedings of the 2019 ACM Conference on Innovation and Technology in Computer Science Education*, 309.

Guzdial, M. (2015). *Learner-centered Design of Computing Education*. California;USA: Morgan Claypool.

Hour of Code. (n.d.). Conditionals With Cards: An Hour of Code Tutorial. Retrieved from `https://code.org/hourofcode/unplugged-conditionals-with-cards`

Kafai, Y. B. (2018). Constructionist visions: Hard fun with serious games. *International Journal of Child-Computer Interaction*, 18, 19–21.

Lonati, V., Malchiodi, D., Monga, M., & Morpurgo, A. (2015). Is coding the way to go? In: *International Conference on Informatics in Schools: Situation, Evolution, and Perspectives*, 165–174.

McKechnie, L. E. F. (2008). Observation Schedule. In *The SAGE Encyclopedia of Qualitative Research Methods* (p. 576). Sage London.

Merriam, S. B. (2009). *Qualitative Research: A Guide to Design and Implementation*. San Francisco: Jossey-Bass.

Nishida, T., Idosaka, Y., Hofuku, Y., Kanemune, S., & Kuno, Y. (2008). New methodology of information education with "computer science unplugged." In: *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*, 241–252.

Rijke, W. J., Bollen, L., Eysink, T. H. S., & Tolboom, J. L. J. (2018). Computational Thinking in Primary School: An Examination of Abstraction and Decom-position in Different Age Groups. *Informatics in Education*, 17(1), 77.

Rodriguez, B., Rader, C., & Camp, T. (2016). Using student performance to assess CS unplugged activities in a classroom environment. In: *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, 95–100.

Rosamond, F. (2018). Computational Thinking Enrichment: Public-Key Cryptography. *Informatics in Education – An International Journal*, 17(1), 93–103.

Sapientia, U. (n.d.). AlgoRythmics. Retrieved from `https://algorythmics.ms.sapientia.ro/`

Sentance, S., & Csizmadia, A. (2017). Computing in the curriculum: Challenges and strategies from a teacher's perspective. *Education and Information Technologies*, 22(2), 469–495.

Shelton, C. (2016). Time to plug back in? The role of "unplugged" computing in primary schools. *ITTE Newsletter*, (2016).

Taub, R., Armoni, M., & Ben-Ari, M. (2012). CS unplugged and middle-school students' views, attitudes, and intentions regarding CS. *ACM Transactions on Computing Education (TOCE)*, 12(2), 1–29.

Threekunprapa, A., & Yasri, P. (2020). Unplugged Coding Using Flowblocks for Promoting Computational Thinking and Programming among Secondary School Students. *International Journal of Instruction*, 13(3), 207–222.

Tonbuloğlu, B., & Tonbuloğlu, İ. (2019). The Effect of Unplugged Coding Activities on Computational Thinking Skills of Middle School Students. *Informatics in Education*, 18(2).

Waite, J., Maton, K., Curzon, P., & Tuttiett, L. (2019). Unplugged Computing and Semantic Waves: Analysing Crazy Characters. In: *Proceedings of the 1st UK & Ireland Computing Education Research Conference*, 1–7.

Ward. (n.d.). Twister. Retrieved from `https://www.thefactsite.com/history-of-twister/`

Zainal, Z. (2007). Case study as a research method. *Jurnal Kemanusiaan*, 5(1).

L. Busuttil is a senior lecturer in Computing Education at the Department of Technology & Entrepreneurship Education (TEE) within the Faculty of Education. He received a PhD from the University of Sheffield in 2014. As part of his work in TEE, Leonard is involved in the formation of pre-service and in-service Computing educators. His research interests include Computational Thinking, Computing education, design of educational software, game-based learning and human computer interaction.

M. Formosa received her Master degree in Teaching and Learning in Education with Computing from the University of Malta and a BSc degree in Software Development from the Malta College of Arts, Science and Technology of Malta. She has experience working in the software industry and later as a Computing teacher in a Maltese secondary school. Her research interests include unplugged computing as well as teaching software development to both secondary school students and adult students.