

The notion of Lego[©] Mindstorms as a powerful pedagogical tool: Scaffolding learners through computational thinking and computer programming

Jacqui Chetty, University of Johannesburg, South Africa

ABSTRACT

This paper documents a study that was conducted with regard to the use of robotics as an innovative pedagogical tool for computer programming. The robotics in question relate to the Lego Mindstorms robots that were introduced as a means to develop further learners' problem-solving skills and motivate learners to have fun while learning. Vygotsky's philosophy regarding the Zone of Proximal Development supports the notion of Lego Mindstorms robots as a pedagogical tool. A mixed methods study was conducted and the aim of the project was twofold. Firstly, to determine whether robots reinforce fundamental computer programming concepts that were taught in the classroom environment. Secondly, to determine whether robots provide motivation and interest in computer programming. The issues regarding the pedagogical approach are discussed and feedback from learners is analysed. The results are positive and encouraging.

Keywords: Pedagogy, Lego Mindstorms robots, Computer programming

INTRODUCTION

Learners enrolled for a computer programming course for the first time often find it challenging to understand the fundamental concepts surrounding the discipline. Equally, educators find it difficult to teach such learners. Research indicates that traditional pedagogical approaches do not lend themselves towards a positive experience for both learner and educator (Lister, 2011). Given that pedagogical approaches are a powerful determinant as to whether a learner will be successful in a course, it is worthwhile investigating the variety of pedagogies and related tools available to determine whether there is an approach that could adequately scaffold learners studying computer programming courses. Research indicates that using games as a pedagogical tool to teach learners the art of algorithmic problem solving as well as programming has successfully been implemented to scaffold learners.

The idea of using games to teach fundamental computer programming concepts is not new (Lawhead et al., 2002). Using games as a learning tool is advocated as games have the potential to contribute positively to successful learning (Piteira & Haddad, 2011). Lego Mindstorms robots is one such game that provides an innovative teaching tool for building learners' computer programming skills. Amongst others, the game provides two necessary elements for learning, namely understanding and motivation (Piteira & Haddad, 2011). It provides a platform for learners to build, reinforce and practise fundamental computer programming concepts, while adding an element of fun. Lego Mindstorms scaffolds learners' learning because it uses action instead of explanation; accommodates a variety of learning styles and skills; reinforces mastery skills; provides an opportunity to practise; and affords an interactive, decision-making context.

Lego Mindstorms robots may prove to be an effective teaching tool to *scaffold* learners as they refine the art of computational thinking and learn to design complex algorithms. Pedagogical approaches are still the most powerful predictor of how successful or unsuccessful learners can be (deRaadt, 2008). This is particularly true of teaching approaches that focus on the notion of 'scaffolding'. The scaffolding becomes an instrument for educators to 'bridge' learners who are navigating a new discipline as they clasp frantically to fragile knowledge in their minds. Structured scaffolding that is maintained for a reasonable period of time allows the fragile knowledge to solidify and become entrenched in the mind of the learner.

The paper is structured as follows: firstly, a review of literature is presented and includes the difficulties that learners face when learning to program; the pedagogical approaches for computer programming; and the use of Lego Mindstorms robots as a pedagogical tool. Secondly, the methodological approach used is described. Thirdly, the results are presented and discussed, and finally the limitations as well as the conclusion are presented.

LITERATURE REVIEW

Difficulties faced by learners

The skills expected for computer programming are complex and learners worldwide find it very difficult to solve problems (Mead et al., 2006). The problem arises from learners needing to articulate a problem into a programming solution (Lahtinen, Ala-Mutka & Jarvinen, 2005; Garner, Hadden & Robins, 2005) by combining syntax and semantics into a valid program (Winslow, 1996) through the construction of mechanisms and explanations (Soloway, 1986). In order to achieve this, learners need to be able to apply fundamental computer programming concepts (Robins, Rountree & Rountree, 2003; Garner et al., 2005) and understand abstract concepts (Lahtinen et al., 2005). When learners are faced with trying to absorb and understand too many new concepts at one time, their working memory may become overloaded. Their overloaded working memories make it very difficult for them to understand the concepts taught to them. The idea of working memory and load capacity is also known as cognitive load theory (Mason, Cooper, Simon & Wilks, 2015).

Cognitive load

Humans are limited to a working memory capacity that is strictly bounded and relatively small (Mason et al., 2015). This means that due to our limited working memory capacity our memories can become overloaded and our cognitive performance can decline. This is particularly true when novice learners are faced with the fundamentals associated with computer programming concepts as these concepts are fraught with abstract ideas or higher order thinking skills. Such concepts are often layered, one on top of the other, before a learner is able to design and construct computer programs. Given that such learners are new to the discipline (novices) their cognitive load increases exponentially, often exceeding their critical threshold level of cognitive capacity (Mason et al., 2015).

Given this scenario, it is therefore not unexpected for research to indicate that the results linked to computer programming courses aimed at novices more often than not have a particularly high failure rate. The situation in South Africa is worse as many learners in South Africa often do not acquire a sound education at primary and secondary educational level (Jansen, 2012). This means that when such learners are presented with a subject, such as computer programming at tertiary level, they struggle as their cognitive load is pushed to capacity. These learners often cannot adapt as they are expected to learn concepts that require abstract reasoning, also known as computational thinking (Bower & Falkner, 2015).

Computational thinking

Computational thinking (DBR_Collective) can be defined as the ability of a learner to develop problem-solving strategies and techniques that assist in the design and use of algorithms and models (Falkner, 2015). According to Lister (2011) such thinking needs time to develop. In fact, most learners possess limited computational thinking (CT) in the early stages of their lives, but such skills should develop and mature, given that learners are educated and receive formal training (Lister, 2011). Given the importance of CT, Kramer (2007) asks whether it is possible to improve learners' CT at tertiary level. He also advocates that unless learners' CT is well developed, they should not be allowed admission into computing courses. Given the South African context, Kramer's advice cannot be implemented, as the nation has a social responsibility to encourage education amongst previously disadvantaged learners. Nelson Mandela felt very strongly about this and insisted that learners have the right to tertiary education (Jansen, 2012). Therefore, innovative pedagogical approaches as well as tools that support learning and favour the development of CT must be investigated. Lego Mindstorms is one such tool that is investigated in this study.

The next section explores the different pedagogical approaches used for computer programming.

PEDAGOGICAL APPROACHES TO COMPUTER PROGRAMMING

Pedagogical approaches relate to the manner in which teaching and learning takes place in order to facilitate desired learning outcomes (Pears et al., 2009). There are many pedagogical approaches to teaching and learning (Boyer, Langevin & Gaspar, 2008; Pears et al., 2009). For example, the teacher-centric approach consists of activities, such as lecturing, questioning and demonstration. The lecturer is the expert who transfers their knowledge across to learners (Xiaohui, 2006). This approach is used extensively to teach not only computer programming courses, but also other disciplines of study. Although the teacher-centric approach is the most popular approach, there are other pedagogical approaches that are unique to teaching and learning computer programming. For example, the learner-centred pedagogical approach involves philosophies that have been around for many decades and even a century. These include social constructivism, peer-led learning, collaborative learning and problem-based learning, to name a few.

Regardless of the pedagogical approach used in the classroom, learning computer programming can take on many forms. Firstly, teaching a particular language, such as Java, where the structure, syntax and semantics of the programming language itself is taught (Pears et al., 2009). Most textbooks are structured according to the constructs of a particular programming language. For example, learners may learn how to make use of variables by applying the eight primitive data types known to Java.

Secondly, teaching problem-solving techniques applicable to computer programming is another approach. The idea is that if a learner is able to solve one type of problem, that learners should be able to solve other problems of a similar nature (Pears et al., 2009; Winslow, 1996). Very precise computer programming structures are taught within this context. For example, instead of learning how to make use of variables by applying the eight primitive data types known to Java, variables can be learned by learners developing pseudo code or flowcharts.

Thirdly, teaching programming through the introduction of graphical user interface (Guillory) tools, such as Scratch, Greenfoot or Alice provide a simulated computer programming environment that is user-friendly. It provides ease-of-use when trying to develop computer programs (Maloney, Resnick, Rusk, Silverman & Eastmond, 2010).

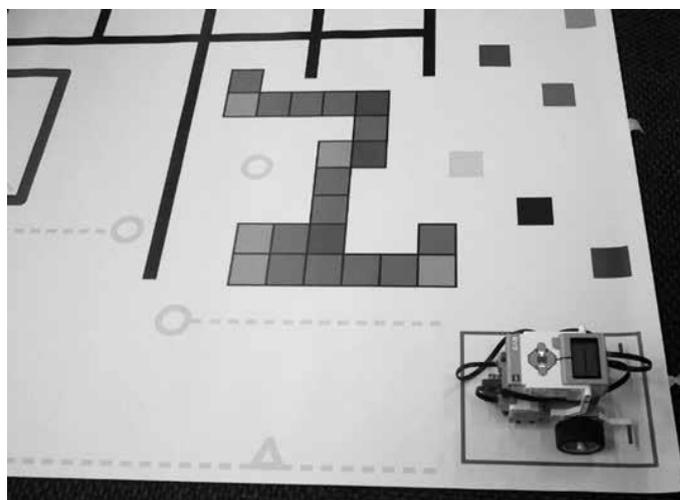
Lastly in this paper, teaching learners how to read, trace and debug existing programs (Patton, 2004; Miliszewska & Tan, 2007) before they embark on writing their own programs is also very effective. Tracing a computer program reveals underlying concepts to learners that they most probably would not have thought of themselves. Learners then learn to mimic these revolutionary ideas and make them their own.

Lego Mindstorms as a pedagogical tool

Lego Mindstorms robots have become a popular pedagogical tool to teach and learn introductory computer programming concepts (Lui, Ng, Cheung & Gurung, 2010; Lawhead et al., 2002). The emphasis is on the word 'tool', where robots create a rich environment that provides a platform for novices as well as experienced teachers to implement a laboratory experience for students to learn programming skills in an interesting, unique and challenging manner. In effect, Stein (1998) challenges the computer science teaching community to move from the premise that computation is calculation to the notion of computation is interaction. Robots would be a natural way to explore such a concept.

Lego Mindstorms robots form part of Lego education and can be bought through a representative responsible for retailing such toys. As illustrated in Figure 1, the Mindstorms consist of building components, a programmable brick, active sensors and motors. There is software for which both Graphical User Interface (GUI) and command line interfaces are available. The robots, together with their associated interfaces provide an opportunity for educators to transform classrooms into rich laboratory or software studios, where learners can experience learner-centred learning, collaborative learning and peer-to-peer programming experimentation (Yamazaki, Sakamoto, Honda, Washizaki & Fukazawa, 2015). This environment provides an opportunity for learners to 'put their programming skills to the test' as what they program comes to life through the Lego Mindstorms robot. They can visually understand 'what works', 'what does not work' and 'why'.

*Figure 1:
An EV3 Lego Mindstorms robot*



Lego Mindstorms robots provide an opportunity for learners to understand fundamental computer programming concepts that are, by their very nature, abstract (deRaadt, 2008). These concepts are not analogies with the real world (Piteira & Haddad, 2011) and learners find it challenging to relate to real-world problems. Moreover, traditional pedagogical approaches to teaching computer programming exacerbate this problem (Rountree, Rountree & Robins, 2002).

Introducing Lego Mindstorms robots provides a unique opportunity to transform a classroom environment in which (Piteira & Haddad, 2011): learners are given an opportunity to 'grapple' with real-world problems; the Lego Mindstorms robot becomes a learning tool that can scaffold learners; fragile knowledge of abstract programming concepts can be reinforced; and learners are given an opportunity to experiment, explore and enjoy programming. These aspects can create a degree of motivation.

The motivation factor

Research indicates that emotions, such as hope, anger, relief, anxiety and boredom are significantly related to motivation, learning strategies, cognitive resources, self-regulation, and academic achievement, as well as personality and classroom antecedents (Pekrun, Goetz, Titz & Perry, 2002). According to Jenkins & Davy (2002) motivation in particular is a crucial component related to learners' success. Although motivation is difficult to quantify, Jenkins has identified expectancy and value as two factors, which when multiplied can predict learners' motivation (Jenkins, 2001). Expectancy is related to the extent to which learners feel that they are able to succeed. Value is related to what they expect to gain. For example, confident learners who feel that they are able to succeed will attach a value or goal related to high marks. They will most likely score high in the area of motivation as: $\text{motivation} = \text{expectancy} * \text{value}$ (Jenkins, 2001).

A motivated learner would therefore experience emotions related to hope, enjoyment and pride, whereas an unmotivated learner would experience emotions related to anger, frustration, anxiety and boredom. Lego Mindstorms robots provide an opportunity for learners to experiment and explore. The idea of learning-through-play is an effective tool to create personal motivation and satisfaction of learning (Piteira & Haddad, 2011).

METHODS AND MATERIALS

The research methodology approach adopted was a Design-based research (DBR_Collective) approach, which included two iterative cycles of testing. Although much data has been collected using this approach, the article focuses on two sets of data collection, namely qualitative as well as quantitative data, aimed at answering the following questions that relate to the aims and objectives of Lego Mindstorms robots: firstly, did the learners indicate that the robot programming further improved their partially developed fundamental computer programming concepts? Secondly, are learners more motivated to become computer programmers due to their exposure to Lego Mindstorms robots?

The qualitative study focused on collecting data by means of a focus group interview. The aim of the interview was to identify the extent to which the Lego Mindstorms robots assisted learners in improving their fundamental computer programming skills as well as to identify whether learners felt that the robots introduced an element of fun and motivation. The data was analysed using Atlas.Ti and a number of codes, categories and themes were established. The themes were expanded upon to provide guidelines for educators wanting to make use of Lego Mindstorms robots as a learning tool to improve problem-solving skills amongst learners.

Additionally, the study made use of a quantitative analysis, where learners answered closed-ended types of questions from a questionnaire developed by the researcher. The questions also focused on the aims and objectives of the article. A Likert-type scale was included with scores, from Strongly Disagree (1) to

Strongly Agree (4). These scores, means and standard deviations calculated statistical data aimed at answering the research questions.

A rich collection of both qualitative as well as quantitative data emerged. Both sets of results indicate that Lego Mindstorms robots provide a platform on which learners' fragile computer programming knowledge can be improved upon using a language independent tool that captures the persistent truths about problem solving and programming techniques.

This article is an extension of a conference paper written by the researcher (Chetty, 2015) and it differs from the conference paper in that the themes that have been developed from the qualitative analysis have been expanded upon for educational practice. Furthermore, the article presents a set of quantitative results pertaining to learners' opinions regarding Lego Mindstorms robots being a tool that can be used to teach generic programming skills. The quantitative analysis surveyed a larger number of learners.

Participants

Of the 135 learners enrolled for a computer programming module 56 learners participated in a Lego Mindstorms project that focused on improving their problem-solving skills as well as to determine whether Lego Mindstorms robots provide the motivation needed to learn a difficult task, such as computer programming. The participants worked collaboratively every week for two hours (within a six month semester), under the supervision of a facilitator as well as each group being led by a tutor.

Course content

The Lego Mindstorms course included explanations regarding the various components; instructions for building a standard EV3 robot; guidelines to program the robot to perform the basic tasks; and exercises for practising. Each session included suggestions for learners to explore further. Figure 2 illustrates the learning in action.

*Figure 2:
Demonstration in action*



The learners were tasked to complete a variety of activities aimed at:

- building the Lego Mindstorms robot according to specifications
- introducing the Lego Mindstorms robot from a physical perspective as well as from a software-driven perspective

- providing learners with a simple task that required them to learn about movement (rotations, power, angles, timing), where the task was to program the robot to move in the shape of a square
- providing learners with more complex tasks within the movement learning area, where learners needed to change their programs and include looping
- introducing learners to the colour sensor, which would require them to make use of looping, such as move until...
- introducing learners to the touch and ultra sound sensor, which would allow them to program the robots regarding distance and object avoidance
- providing a variety of tasks every week to simulate the above mentioned robots concepts that reflect the programming environment
- introducing a theme, such as the solar system where learners were asked to program the robots to move from 'earth' to the 'moon' and the robot had to avoid any 'alien spaceships'.

DATA COLLECTION TECHNIQUES

As discussed, this study comprised both qualitative and quantitative data collection techniques. Firstly, a focus group interview was a suitable approach to enrich the evaluation of the pedagogical design as it allows a particular case to be examined to provide insight into the extent to which learners felt that the pedagogical intervention increased their knowledge and supported their learning regarding the module (Denzin & Lincoln, 1994). Focus group interviews capture the complexities of a phenomenon; such detailed observations can very rarely be captured in surveys or experimental designs (Falkner, Vivian & Falkner, 2014; Hartley, 2004). Although the sample size is small, the number of participants in the study being 21, it is sufficient for qualitative research that ranges from 1 to 99, with an average sample size of 22 (Troskie-de Bruin, 2013). The focus group interview was conducted and recorded by the researcher. Its aim was to gather data regarding the research questions presented above in this article.

Secondly, a questionnaire was constructed that contained Likert-type items to measure students' extent of agreement with each item on a scale ranging from Strongly Disagree to Strongly Agree. Using SPSS v.22, measures of central tendency were computed to summarise the data of each item. Measures of dispersion were computed to understand the variability of responses for each item. Items with low reliability scores (Cronbach alpha <.70) were excluded from the computation. Care was taken to reverse code items where applicable.

RESULTS

Qualitative results

The audio recording was transcribed to text and the data were subjected to the process of open coding and axial coding. The qualitative software ATLAS.ti was used (Troskie-de Bruin, 2013) to code the units as well as derive specific categories. A first-cycle coding was conducted using the Initial Coding (open coding) method as it allows the researcher to 'code quickly and spontaneously, pay meticulous attention to the rich dynamics of data through line-by-line coding – a "microanalysis" of the corpus' so that the researcher can 'search for processes – participant actions that have antecedents, causes, consequences and a sense of temporality' (Saldana, 2013; Strauss & Corbin, 1990). This type of coding was chosen as the researcher wanted to determine learners' experiences regarding the use of Lego Mindstorms robots to improve problem solving as well as to determine whether the robots provide an element of motivation for learners. A number of codes were identified as the data was searched sentence-by-sentence.

As the code list was quite extensive Morse (1994: 25) advises to reorganise and reanalyse data to 'link seemingly unrelated facts logically, of fitting categories one with another'. During the second cycle of coding, 'axial coding' was conducted. Boeije (2010) explains that the purpose of axial coding is 'to

determine which [codes] in the research are the dominant ones and which are the less important ones ... [and to] reorganise the data set: synonyms are crossed out, redundant codes are removed and the best representative codes are selected' (Boeije, 2010: 109). Saldana (2013) succinctly explains that the 'axis' of Axial Coding is a category(ies) that is (are) derived from First Cycle coding. Table 2 tabulates the development of categories from the codes. These categories were further sub-divided into themes, namely solving real-world authentic activities, computational thinking, collaborative learning, and motivating learners through Lego Mindstorms.

*Table 1:
Classification of categories*

Category	Frequency	Frequency %
Improve problem-solving skills	21	100%
Testing programming abilities using real-world problems	5	23.8%
Fun way to develop programming skills	21	100%
Collaboration with peers	21	100%
Computational thinking	11	52.4%
Motivation factor	10	47.6%
Made new friends	7	33.3%

Additionally, learners were also asked to indicate by way of a 'Yes' or 'No' whether the Lego Mindstorms robots provided scaffolding, given their fragile knowledge related to computer programming concepts learned in the classroom environment. As seen in Table 2 all learners indicated that the robots provided an opportunity to further reinforce problem-solving skills. Of the 21 learners, 20 learners felt that the robot programming provided them with an opportunity to learn how to break a problem into smaller steps.

*Table 2:
Responses regarding Mindstorms reinforcing learning*

Programming concept	'yes'	'no'
Problem solving	21	
Breaking problems into small steps	20	1
Practise algorithmic skills	20	1
Assist in understanding how to solve problems better algorithmically	21	
Assist in coding programs that require input-processing-output	19	2
Assist in coding programs that require methods	21	
Assist in coding programs that require repetition	21	

Table 2 shows a classification of the codes that were generated using Atlas Ti. As discussed, during the axial coding process the codes that best represented and supported the notion of Lego Mindstorms robots being beneficial to improve programming skills were highlighted and grouped into themes. The main themes that emerged related to solving real-world authentic activities, computational thinking, collaborative learning, and motivation. Each is now discussed.

Solving real-world authentic activities

Researchers and experts worldwide agree that an authentic learning activity represents a problem that has real-world relevance, is ill-defined, and needs to be completed over a period of time (Herrington, 2013; Lombardi, 2007; Herrington, 2006; Brannock, Lutz & Napier, 2013).

Real-world relevance relates to problems that match 'every day' tasks of professionals in practice. Such problems are normally 'messy' or ill-defined. Ill-defined problems are problems that when described to learners are open to interpretation, as opposed to problems that are developed by following step-by-step solutions. Instead of being highly prescriptive, ill-defined problems provide an opportunity for learners to identify the steps needed to complete the activity (Herrington, Reeves & Oliver 2006). As ill-defined problems are more complex, learners need a longer period of time to complete such activities. A longer time period allows learners to reflect on the choices they make regarding the solution. This enhances their metacognitive skills (Lombardi, 2007).

Learners in particular embraced and acknowledged the idea of using the Lego Mindstorms robots to program in the real world. For example, learners were asked to develop a 'car' using their robot. The car had to be able to navigate from point A to point B, while avoiding obstacles and adhering to traffic robots (red – stop; green – go). As learners understood the real-world problem in the context of their own lives, this made it easier for them to develop and implement the coding instructions. Some of the responses from learners included the following responses, where the bolded responses highlight the themes:

- Yes I enjoyed it because of the observation of **using concepts of programming logic in physical situations**
- It's a fun programming logic, whereby you can solve problems in different ways through that robot. It is realistic, **because you can just imagine the robot in real-life situations**

Another learner acknowledged the notion of using the Lego Mindstorms robot to benchmark his personal programming skills.

- Yes, I really did. When I joined the project I was curious and I wanted **to test my abilities** by trying and taking an opportunity that was presented to us. I am glad I did because now I know where I stand and I know I can

Asking learners to solve problems that involved authentic tasks that they could relate to, provided essential scaffolding so that the gap between 'understanding the problem' and determining 'how to solve it' was not too wide.

Computational thinking

Vygotsky (1978) suggested this as any higher order thinking skill (including computer programming skills) evolves in the construction of joint social activities (collaborative tasks), prior to developing into skills that can be applied to independent problem solving (Beck & Chizhik, 2013). The social spaces or 'zones of proximal development' are critical if higher order thinking skills are to be achieved (Vygotsky, 1978).

For many of the learners at the university, critical thinking and deeper level learning is difficult to achieve. However, the learners acknowledged the notion of critical thinking when learning how to program the robots.

- It challenged our thinking and encouraged us to **make use of critical problem-solving skills** to accomplish our tasks

- It was different from the usual work we do in the labs. It provided the platform to test what we learnt in class – **it tested our analytical skills**

Collaborative learning

The idea of constructing knowledge through a social setting can be a powerful educational and learning tool (Vygotsky, 1978; Kozulin et al., 2003; Stetsenko, 2010; Ben-Ari, 1998). This type of learning involves two important concepts, namely social constructivism and collaboration. Social constructivism is a philosophy and a learning theory that has established itself in the last few decades (Karagiorgi & Symeou, 2005). It attempts to take the individual into account by viewing them as unique, having their own personal and subjective experiences (Karagiorgi & Symeou, 2005). The individual imposes meaning on the world by constructing knowledge based on past experiences, goals, curiosities and beliefs (Cole, 1992). In this way, the individual adapts and constructs knowledge as they make sense of their world.

It is widely acknowledged that there are many educational advantages that can be derived from learners working in collaboration with one another (Brown, 2005; Preston, 2006). The expression 'I learn what I believe as I hear myself speak' is very powerful. One advantage is that learners are more successful when learning occurs in the midst of others. Learners learn from one another as they discuss problems and formulate solutions (Ben-Ari, 1998; Lombardi, 2007).

Collaborative learning is one of the most powerful characteristics of Lego Mindstorms. Learners remark:

- I made new friends who **helped me on my programming skills** and now I can take shortcuts to make the best out of programming
- I was a person who liked to isolate herself but **now I can't even count how many friends I have**, so it boosted my social life
- We were discussing and **bringing our knowledge into one**

Collaboration with others is a powerful pedagogical tool. Such collaboration often leads to increased motivation among learners.

Motivating learners and Lego Mindstorms robots

Hirumi & Kebritchi (2008) argue that games are effective tools for learning because, amongst other advantages, games create personal motivation and satisfaction. One of the main advantages of using games, such as Lego Mindstorms to motivate learners, is due to the inherent characteristics associated with the games, namely energy, direction, persistence and equifinality – all aspects that stimulate and motivate learners (Serrano-Camara, Paredes-Velasco, Alcover & Velezques-Iturbide, 2014).

Motivation forms part of a conceptual model or framework known as the self-determination theory. This theory emphasises the importance of the development of internal human resources for personal development and self-regulation. Such self-determination refers to something because it is interesting or enjoyable, such as the Lego Mindstorms. Such games provide a natural wellspring of learning and achievements that can either be encouraged or discouraged by facilitators' practices. Such motivation results in high quality learning and creativity (Ryan & Deci, 2000).

The Lego Mindstorms robots inspired and motivated learners to think about programming in a very real way. They reported:

- Because **now I want to program bigger** and better robots

- **It motivates me** because it showed me that what I am studying is relevant to today
- Better programmer, **makes me want to perfect the robot** and make it do exactly what I want it to do

Quantitative results

*Table 3:
Results of Lego Mindstorms robots promoting learning*

Question	N	M	MODE	SD
Q1 The use of Lego Mindstorms robots helped me to write better problem-solving solutions	56	3.09	3.00	.794
Q2 I found better problem-solving solutions after playing with Lego Mindstorms robots.	56	3.07	3.00	.815
Q3 The more I 'played' with the Lego Mindstorms robots the better my problem-solving solutions became.	56	3.03	3.00	.844
Q4 I understood if/else (selection) better after using Lego Mindstorms	56	3.23	3.00	.572
Q5 I understood looping (repetition) better after using Lego Mindstorms	56	3.30	3.00	.570
Q6 I understood programming better after using Lego Mindstorms	56	3.28	3.00	.564
Scale	56	3.17	3.00	.693

It is immediately apparent from Table 3 that students either Agreed (3) or Strongly Agree (4) with each item, as the mean of all the responses is above 3 Agreed (3). Items 1 (n = 56, M = 3.09, SD=.794) and 2 (n = 56, M=3.07, SD=.815) indicate that learners made use of Lego Mindstorms robots to learn problem-solving skills and that they enjoyed the use of the tools. Items 2 and 3 show that learners 'use of the Lego Mindstorms robots helped me to write better problem-solving solutions'. Table 3 also indicates that the Lego Mindstorms robots promoted learning. Most of the students either Agreed (3) or Strongly Agree (4) indicated by items 4 (n = 56, M = 3.23, SD = .572), 5 (n = 56, M = 3.30, SD = .570) and 6 (n = 56, M = 3.28, SD = .564) that after completing the Lego Mindstorms project their problem-solving skills had improved.

DISCUSSION

The research questions related to this article focused on two aspects, namely, did learners feel that Lego Mindstorms robots improved fragile knowledge; and were the learners motivated to learn programming due to the interaction with the Lego Mindstorms robots? These questions are now answered as part of a discussion, based on the qualitative as well as the quantitative data presented earlier in the paper.

Firstly, learners indicated that the idea of learning programming using authentic activities proved successful, as they were often able to relate to the real-world problems that they were expected to solve. They felt as if they were real-life programmers. The authentic activities were fun, which motivated them to keep trying.

Secondly, learners felt that the tasks challenged their thinking and it was expected of them to reason in a critical manner, where knowledge had to be transferred from one context to another. As higher order

thinking skills (HOTS) forms an important aspect of learning, one in which lifelong learning skills are formed, the development of programs through robotics assisted in the development of HOTS.

Thirdly, the notion of constructing knowledge in collaboration with one another is one of the most powerful learning tools. Learners constructed solutions in groups where ideas were discussed, a number of solutions were formulated and explained and learners shared their knowledge socially. Learners learned from one another, sometimes they were the ones teaching and other times they were the ones learning.

Lastly, motivation cannot be underestimated within the educational process. Learners who are motivated often achieve more than learners who are not. The learners were very excited to be part of the class every week and an element of competitiveness began to develop from one week to the next. As an educator it could clearly be seen that learners were enjoying the learning experience. The experience of being part of the class motivated learners to develop better solutions that were more and more powerful as the weeks progressed.

LIMITATIONS OF THE STUDY

Providing learners with necessary scaffolding as they learn is essential. As discussed, there are many benefits for both the educator and the learner. However, many lessons were learned from this study. As Lego Mindstorms robots will be fully integrated into the computer programming course from 2015, it is beneficial to reflect on the limitations of making use of Lego Mindstorms robots. While some limitations may be overcome, as the educators become more experienced, other limitations remain as challenges, either to be conquered or not. The limitations of the study include:

- sufficient time available for learners to take full advantage of learning through robots
- a suitable software lab that encourages play
- sufficient time to learn and prepare for the robot lessons
- limited expert knowledge regarding the use of robotics
- support for the use of robots at management level.

Most of these challenges have been overcome to the extent that a further seven sets of Lego Mindstorms robots have been purchased. Additional time has been scheduled on learners' timetables (semester A) and the robots will form part of the computer programming course, where marks will be allocated towards each project. Most importantly, educators have been given the opportunity to register for an official course through UNISA. This will mean that educators have the expertise to extend learners.

CONCLUSION

In the analysis, a variety of categories related to whether Lego Mindstorms robots provided a platform that reinforces fragile computer programming concepts, were identified. Additionally, a quantitative analysis provided further evidence suggesting that learners confirmed that Lego Mindstorms robots assisted them with learning computer programming concepts. Although not a pedagogy within its own right, Lego Mindstorms robots can be a platform that provided an opportunity for the focus of learning to shift from traditional learning, to learning that engages learners actively, using real-world problems. It can allow learner to develop dynamic programming skills in a language independent environment, thus seen as a pedagogical tool. In today's rapidly changing world, providing innovative ways to scaffold learners and challenge them to think in other ways is essential.

As this project was a pilot study, future work will focus on an additional Lego Mindstorms project, where robotics is integrated into the computer programming course presented to first-year learners (novices).

Additionally, the Lego Mindstorms robots course will be presented to Grade 4 learners at a school close to the university where the study was conducted. The focus will be on girls in Information Technology as well as the development of problem-solving skills at a younger level. Lastly, the course will also be presented to student teachers at a university overseas.

REFERENCES

- Beck, L. & Chizhik, A. (2013) 'Cooperative Learning Instructional Methods for CS1: Design, Implementation, and Evaluation' *ACM Transactions On Computing Education* 13(21) pp.470-474.
- Ben-Ari, M. (1998) 'Constructivism in Computer Science Education' Paper presented at the SIGSCE'98 Conference. Atlanta GA USA: ACM.
- Bower, M. & Falkner, K. (2015) 'Computational Thinking, the Notional Machine, Pre-service Teachers, and Research Opportunities' *Australasian Computer Education (ACE) Conference*, edited by Daryl D'Souza and Katrina Falkner. Sydney, Australia: Australasian Computer Society Inc.
- Boyer, N.R., Langevin, S. & Gaspar, A. (2008) 'Self Direction & Constructivism in Programming Education' *SIGITE'08*. Cincinnati, Ohio, USA: ACM.
- Brannock, E., Lutz, R. & Napier, N. (2013) 'Integrating Authentic Learning into a Software Development Course: An Experience Report' *SIGITE'13*. Orlando, Florida, USA: ACM.
- Brown, J.S. (2005) 'New Learning Environments for the 21st Century' Forum for the Future of Higher Education Aspen Symposium.
- Chetty, J. (2015) 'Lego Mindstorms: Merely a Toy or a Powerful Pedagogical Tool for Learning Computer Programming' *Australian Computer Science Conference (ACSC) 2015*. Sydney, Australia.
- Cole, P. (1992) 'Constructivism revisited: A search for common ground' *Educational Technology* 33(34) pp.27-34.
- DBR_Collective. (2003) 'Design-based research: An emerging paradigm for educational inquiry' *Educational Researcher* 32(4) pp.5-8.
- Denzin, N.K. & Lincoln, Y.S. (1994) *Introduction: Entering the field of qualitative research*. Thousand Oaks: Sage.
- deRaadt, M. (2008) 'Teaching Programming Strategies Explicitly to Novice Programmers' University of Southern Brisbane, Queensland, Australia.
- Falkner, K., Vivian, R. & Falkner, N.J.G. (2014) 'Identifying Computer Science Self-Regulated Learning Strategies' *ITICSE'14*. 19th Annual Conference on Innovation and Technology in Computer Science. Uppsala, Sweden: ACM.
- Falkner, K. (2015) 'Teaching Computational Thinking in K-6: The CSER Digital Technologies MOOC' *Australasian Computer Education (ACE) Conference*. Sydney, Australia: Australia Computer Society Inc.
- Garner, S., Haden, P. & Robins, A. (2005) 'My Program is Correct But it Doesn't Run: A Preliminary Investigation of Novice Programmers' Problems' *Australasian Computing Education Conference*. Newcastle, Australia: Australian Computer Society, Inc.

- Guillory, B.A. (2011) 'Teaching through problem solving' <http://www.slideshare.net/bbienemy/teaching-through-problem-solving1> (Accessed February 2014).
- Hartley, J. (2004) 'Case Study Research' In C. Cassell & G. Symon (Eds.) *Essential Guide To Qualitative Methods In Organisational Research*. London: SAGE.
- Herrington, J. (2006) 'Authentic e-learning in higher education: Design principles for authentic learning environments and tasks' AACE '10. Vancouver, Canada.
- Herrington, J. (2013) 'Its the task that matters most' http://authenticlearning.ino/AuthenticLearning/Authentic_Task.html (Accessed November 2013).
- Herrington, J., Reeves, T.C. & Oliver, R. (2006) 'Authentic Tasks Online: A synergy among learner, task and technology' *Distance Education* 27(15) pp.1-15.
- Hirumi, A. & Kebritchi, M. (2008) 'Examining the Pedagogical Foundations of Modern Educational Computer Games' *Computers & Education* 51 pp.1729-1743.
- Jansen, J. (2012) 'I would seriously consider not sending my child to school in SA' *Moneyweb* <http://www.moneyweb.co.za/mw/view/mw/en/page292681?oid=562851&sn=2009> Detail. (Accessed 24 February 2012).
- Jenkins, T. (2001) 'Teaching Programming - A Journey From Teacher To Motivator' *2nd Annual LTSN-ICS Conference*. London: LTSN Centre for Information and Computer Sciences.
- Jenkins, T. & Davy, J. (2002) 'Diversity and Motivation in Introductory Programming' *Innovation in Teaching and Learning in Information and Computer Sciences* 1(1) doi: 10.1120/ital.2002.01010003
- Karagiorgi, Y. & Symeou, L. (2005) 'Translating Constructivism into Instructional Design: Potential and Limitations' *Educational Technology & Society* 8(11) pp.117-127.
- Kozulin, A., Gindis, B., Ageyev, Vladimir S., Miller & Suzanne M. (Eds.) (2003) *Vygotsky's Educational Theory in Cultural Context*. Cambridge: Harvard University Press.
- Kramer, J. (2007) *Magazine Communication of the ACM* 50(4) pp.36-42.
- Lahtinen, E., Ala-Mutka, K. & Jarvinen, H. (2005) 'A Study of the Difficulties of Novice Programmers' *ITICSE '05*. Monte de Caparica, Portugal: ACM.
- Lawhead, P.B., Bland, C.G., Barnes, D.J., Duncan, M.E., Goldweber, M., Hollingsworth, R.G. & Schep, M. (2002) *A Road Map for Teaching Introductory Programming Using LEGO Mindstorms Robots*. New York USA: *ITICSE-WSR*.
- Lister, R. (2011) 'Concrete and Other Neo-Piagetian Forms of Reasoning in the Novice Programmer' *Australasian Computer Education Conference*. Perth, Australia: Australia Computing Society Inc.
- Lombardi, M.M. (2007) 'Authentic Learning for the 21st Century: An Overview' *Educause* pp.1-12.
- Lui, A.K., Ng, S.C., Cheung, H.Y. & Gurung, P. (2010) 'Facilitating Independent Learning with Lego Mindstorms Robots' *acmInroads* 1(5) pp.49-53.

Maloney, J., Resnick, M., Rusk, N., Silverman, B. & Eastmond, E. (2010) 'The Scratch Programming Language Environment' *ACM Transactions On Computing Education* 10 Article 16 pp.1-15.

Mason, R., Cooper, G., Simon & Wilks, B. (2015) 'Using Cognitive Load Theory to select an Environment for Teaching Mobile Apps Development' *Australasian Computer Education (ACE)*. Sydney, Australia: Australia Computing Society Inc.

Mead, J., Gray, S., Hamer, J., James, R., Sorva, J., St.Clair, C. & Thomas, L. (2006) 'A Cognitive Approach to Identifying Measurable Milestones for Programming Skill Acquisition' *ITICSE'06*. Bologna, Italy: ACM.

Miliszewska, I. & Tan, G. (2007) 'Befriending Computer Programming: A Proposed Approach to Teaching Introductory Programming' *Issues in Information Science and Information Technology* 4(13) pp.277-289.

Pallant, J. (2007) *SPSS Survival Manual. A step-byStep Guide to Data Analysis Using SPSS for Windows (Version 15)* Milton Keynes: Open University Press.

Patton, Peter C. (2004) 'New Methods for Teaching Programming Languages to both Engineering and Computer Science Students' http://www.micsymposium.org/mics_2004/Patton.pdf (Accessed January 2014).

Pears, A., Seidman, S., Malmi, L., Mannila, L., Adams, E., Bennedsen, J., Devlin, M. & Paterson, J. (2009) 'A Survey of Literature on the Teaching of Introductory Programming' *ITICSE-WGR '07* 39(4) pp.204-223.

Pekrun, R., Goetz, T., Titz, W. & Perry, R.P. (2002) 'Academic Emotions in Students' Self-Regulated Learning and Achievement: A Program of Qualitative and Quantitative Research' *Educational Psychologist* 37 pp.91-105.

Piteira, M. & Haddad, S.R. (2011) 'Innovate in Your Program Computer Class: An approach based on a serious game' *OSDOC'11*. Lisbon, Portugal.

Preston, D. (2006) 'Using Collaborative Learning Research to Enhance Pair Programming Pedagogy' *ACMSIGITE3* pp.16-21.

Robins, A., Rountree, J. & Rountree, N. (2003) 'Learning And Teaching Programming: A Review and Discussion' *Computer Science Educational Journal* 13 pp.137-172.

Rountree, N., Rountree, J. & Robins, A. (2002) 'Predictors of Success and Failure in a CS1 Course' *ACM SIGCSE Bulletin* 34(4) pp.189-196.

Ryan, R.M. & Deci, E.L. (2000) 'Intrinsic and Extrinsic Motivations: Classic Definitions and New Directions' *Contemporary Educational Psychology* 25 pp.54-67.

Saldana, J. (2013) *The Coding Manual for Qualitative Researchers*. LA, London, New Delhi, Singapore: SAGE.

Serrano-Camara, L.M., Paredes-Velasco, M., Alcover, C. & Velazquez-Iturbide, J.A. (2014) 'An evaluation of students' motivation in computer-supported collaborative learning of programming concepts' *Computers in Human Behaviour* 31 pp.499-508.

Soloway, E. (1986) 'Learning to Program = Learning to Construct Mechanisms and Explanations' *Communications of the ACM* 29(9) pp.850-858.

Stein, L. (1998) 'What We've Swept Under the Rug: Radically Rethinking CS1' *Computer Science Education* 8(2) pp.118-129.

Stetsenko, A. (2010) 'Teaching-learning and development as activist projects of historical Becoming: expanding Vygotsky's approach to pedagogy' *Pedagogies: An International Journal* 5 pp.6-16.

Strauss, A.L. & Corbin, J.M. (1990) *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. London: SAGE.

Troskie-de Bruin, C. (2013) 'Qualitative research: Requirements for post-graduate study projects' edited by Research and Development Consultants (ASEV). Stellenbosch, South Africa.

Vygotsky, L. (1978) *Mind and Society*. Cambridge, MA: Harvard University Press.

Winslow, L.E. (1996) 'Programming Pedagogy -- A Psychological Overview' *SIGCSE Bulletin* 28(6) pp.17-22.

Xiaohui, Hu. (2006) 'Improving teaching in Computer Programming by adopting student-centered learning strategies' *The China Papers* pp.46-51.

Yamazaki, S., Sakamoto, K., Honda, K., Washizaki, H. & Fukazawa, Y. (2015) 'Comparative Study on Programmable Robots as Programming Educational Tools' *Australasian Computer Education (ACE) Conference*. Sydney, Australia: Australia Computer Society Inc.