Lego® Mindstorms: Merely a Toy or a Powerful Pedagogical Tool for Learning Computer Programming?
Jacqui Chetty
University of Johannesburg
jacquie@uj.ac.za

Abstract
Research world-wide indicates that computer programming is a difficult subject to teach and learners find such modules challenging. The result is that attrition rates are high. South Africa experiences similar problems. These problems are exacerbated by learners entering tertiary institutions under-prepared. Such institutions are under much pressure to transform learners’ academic capabilities. Transformation can be realised through alternative pedagogical approaches as it is well known that these approaches can influence learning outcomes in a positive manner.

This paper documents an innovative pedagogical approach to teach computer programming through the use of robotics. Lego Mindstorms robots have been used to further develop learners’ problem solving skills as well as encourage learners to “think out of the box”. The pedagogical approach supports Vygotsky’s philosophy regarding the Zone of Proximal Development. A case study was conducted and learners were expected to complete different projects using the Lego Mindstorms EV3 robots. The aim of each project was twofold. Firstly, to further reinforce fundamental computer programming concepts that had been partially developed. Secondly, to increase interaction between learners and generate higher motivation and interest in computer programming. A qualitative analysis was performed subsequent to the case study. The issues regarding the pedagogical approach are discussed and feedback from learners is analysed. The results are positive and encouraging.

Keywords: Lego Mindstorms robots, innovative pedagogical approach, programming, problem solving

1 Introduction
The idea of games as a pedagogical approach to teaching-and-learning computer programming is not new (Lawhead 2002). This said, using games as a learning tool is advocated as games have the potential to positively contribute to successful learning (Piteira 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 2011).

It provides a platform for learners to build, reinforce and practice 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 practice; and affords an interactive, decision making context.

This paper investigates the benefits of using Lego Mindstorms as a pedagogical tool to reinforce learners’ fragile knowledge related to the fundamental concepts of computer programming. As motivation often predicts how successful a learner is academically (Jenkins 2000, Serrano-Camara 2014), the paper also investigates whether Lego Mindstorms provides a high level of motivation.

The paper is structured as follows: section 2 presents a review of literature related to computer programming and includes difficulties faced by learners learning to program. Pedagogical approaches in use today are also discussed, as well as Lego Mindstorms as a pedagogical tool. Section 3 describes the methodology approach used and section 4 evaluates and analyses results produced from the methodological approach. Section 5 highlights observations noticed by the educator. Section 6 concludes the paper.

2 Background

2.1 Difficulties faced by learners
The skills expected for computer programming are complex. These skills include the ability to:
- Solve problems (Development-OECD 2004, Mead 2006);
- Articulate a problem into a programming solution (Garner 2005, Lahtinen 2005);
- Construct mechanisms and explanations (Soloway 1986);
- Combine syntax and semantics into a valid program (Winslow 1996);
- Understand larger entities of a program instead of smaller details (Lahtinen 2005);
- Apply fundamental computer programming concepts (Robins 2003, Garner 2005);
- Understand abstract concepts (Lahtinen 2005); and
- Properly estimate their level of understanding (Lahtinen 2005).

Many of these skills require learners to think in an abstract manner, or require higher order thinking skills (HOTS). Unfortunately, learners in South Africa often do not acquire such skills at primary and secondary educational level (Jansen 2012). This means that when such learners are presented with a subject, such as computer programming, they struggle to deliberate concepts in an abstract manner (Mason 1999). For
example, although learners may understand how to solve a particular problem mathematically, they may not be able to articulate the problem into a programming solution by applying fundamental computer programming constructs learnt in the classroom.

2.2 Abstract reasoning

Abstract reasoning is one of the most important mental tools that learners must have to become competent computer programmers (Corney 2012). Lister (2011) proposes that learners may possess limited skills 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). However, research indicates that very few learners’ are able to engage with and solve programming problems that involve abstract reasoning. As seen in Table 1, which illustrates Lister’s Neo-Piagetian theory, learners’ progress as programmers, in a similar fashion to the developmental process described above.

Given the importance of abstract reasoning Kramer (2007) asks whether it is possible to improve learners’ abstract reasoning through education and training. He also advocates that unless learners’ abstract reasoning is well developed, they should not be allowed admission into computing courses.

In the South African context, Kramer’s advice cannot be implemented, as the mandate from the government is to be inclusive of previously disadvantaged learners’. Nelson Mandela felt very strongly about this and insisted that learners have the right to tertiary education (Jansen 2012). Unfortunately, many learners’ attend schools that are over populated and under resourced (Mason 1999). Consequently, tertiary educators cannot adopt the approach of testing for abstract reasoning ability as many learners’ do not meet the requirements. Therefore, innovative pedagogical approaches that favour the development of abstract reasoning have to be investigated and implemented. Lego Mindstorms is one such tool that is investigated in this study.

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

2.3 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 2009). There are many pedagogical approaches to teaching-and-learning (Boyer 2008, Pears 2009). However, the traditional teacher-centric pedagogical approach is still the most popular (Nicolaides 2012).

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 modules, 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.

The learner-centred pedagogical approach involves philosophies that have been around for many decades and even a century. These include social constructivism, peer-lead learning, collaborative learning and problem-based learning, to name a few.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor motor</td>
<td>Students who are able to trace code with less than 50% accuracy are at this stage.</td>
</tr>
<tr>
<td>Pre-operational</td>
<td>Students are able to competently trace code but they are not able to derive meaning or explanation regarding the outcome of the code.</td>
</tr>
<tr>
<td>Concrete operational</td>
<td>Students are able to trace and describe code but they are only able to do so in familiar situations. A student would not be able to successfully transfer their skills from one context to another.</td>
</tr>
<tr>
<td>Formal operational</td>
<td>Students are deemed to be competent programmers at this stage.</td>
</tr>
</tbody>
</table>

Table 1: Lister’s (2011) Neo-Piagetian stages

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 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 (Winslow 1996, Pears 2009). 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 learnt by learners developing pseudo code or flowcharts.

Thirdly, teaching programming through the introduction of graphical user interface (GUI) 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 2010).

Lastly in this paper, teaching learners how to read, trace and debug existing programs (Patton, Miliszewska 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.

All of the above-mentioned approaches can embrace the use of peer collaboration. Peer collaboration is the process of dividing learners into groups of between five and eight, or pairs. They proceed to solve problems and develop computer programs collaboratively (Preston 2006, Teague 2007).
2.3.1 Lego Mindstorms as a Pedagogical Approach

Lego Mindstorms robots have become a popular pedagogical tool to teach and learn introductory computer programming concepts (Lawhead 2002, Lui 2010). In effect, Stein (Stein 1998) challenges the computer science teaching community to move from the premise that computation is calculation to the notion of computation as interaction. Robots would be a natural way to explore such a concept.

Lego Mindstorms 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 (EV3), active sensors and motors. There is software for which both GUI and command line interfaces are available. These simple tools 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. 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” and “what does not work” and “why”.

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 2011) and learners find it challenging to relate to real-world problems. Moreover, traditional pedagogical approaches to teaching computer programming exacerbate this problem (Rountree 2002).

Introducing Lego Mindstorms robots provides a unique opportunity to transform a classroom environment in which (Piteira 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. This creates a degree of motivation.

Figure 1: An EV3 Lego Mindstorms robot

2.4 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 2002). According to Jenkins (Jenkins 2000) 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: motivation = expectancy * 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 provides 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 2011).

3 Methodology

In this paper, a qualitative analysis regarding learner reflections of Lego Mindstorms robots was undertaken in order to identify how the robots assisted learners in improving their fundamental computer programming skills. This was explored within a case study of an introductory computer programming course. The research questions that were asked relates to: firstly, did the learners reflections 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?

3.1 Research Method

An instrumental case study was a suitable approach for answering the research questions as it allows a particular case to be examined to provide insight into the extent to which robot programming can scaffold learners’ learning of fundamental computer programming concepts (Denzin 1994). Case studies capture the complexities of a
Although the sample size is small, the awareness and application of ion represented a problem solving stage. In order to achieve this, the process of open coding, before proceeding to axial coding. Grounded theory involves establishing a coding framework and it differs from other types of qualitative analysis in that a specific, structured coding framework is not employed (Charmaz 2006). The first stage in grounded theory development is open coding, where the data is broken down into distinct segments in order to obtain the full collection of ideas and concepts present in the data, without regard to how it will be used. Subsequently, axial coding is employed, where the coding framework developed during the open coding stage is refined and reorganised into specific categories, informed by theoretical frameworks and comparison within the data (Henning 2004).

The case considered in this research project forms part of an introductory computer programming course at a university in Johannesburg, South Africa. Learners within this course have already completed a 6 month (semester A) course related to the development of problem solving skills. Furthermore, they are currently in their second 6 months (semester B) learning a computer programming language, namely C#. Their computer programming competencies are still developing. The learning objectives for both courses include awareness and application of simple data structures, related algorithms and algorithm complexity, as well as initial experiences in small-scale problem solving and software development.

The Lego Mindstorms robots project has been included as a supplementary course. One hundred and twenty six learners were approached about the project and a small demonstration was conducted to illustrate to them some of the skills that they would learn. Learners were informed that the Lego Mindstorms project was not mandatory. A total of 63 learners volunteered to be part of the project. Learners were divided into 3 groups of 21 each, where each group was further sub-divided into 7 groups of 3. Each group of 21 were presented with a variety of Lego Mindstorms projects that they had to complete over a period of 4 weeks (a 2 hour session per week).

The Lego Mindstorms course includes explanations of various components, instructions for building a standard EV3 robot, guides to program the robot and perform the basic tasks, and exercises for practicing. Each session included suggestions for learners to explore further. Figure 2 illustrates an educator demonstrating an action.

### 3.2 Coding Framework

The basic unit of analysis in this project were coding units (Charmaz 2006). During the open coding stage, sections of text, such as a word, phrase, sentence or paragraph, were coded while the selection represented a single idea or concept related to using robots as an innovative pedagogical approach. During the axial coding stage, the researcher refined the established codes into categories, merging codes where appropriate and identifying specific categories as derived from the data. The qualitative software ATLAS.ti was used (Troskie-de Bruin 2013) to code the units as well as derive specific categories. Table 2 illustrates the specific categories as defined by the axial coding. Table 3 illustrates the extent to which learners felt that the Lego Mindstorms robots provided them with an opportunity to reinforce fundamental programming concepts that they had already learnt.

### 4 Qualitative Analysis

The analysis so far indicates that firstly, learners recognise the benefit of Lego Mindstorms as a tool to further reinforce fragile fundamental programming concepts. Secondly, Lego Mindstorms motivates learners to become better programmers and a passion for programming is developed. As illustrated in Table 2 the following categories emerged.

#### 4.1 Solving real-world authentic activities

Researchers and experts world-wide 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 2006, Lombardi 2007, Brannock 2013, Herrington 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 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).

Authentic learning activities provide an opportunity for learners to construct new knowledge instead of reproducing existing knowledge. In order to achieve this, learners are provided with multiple sources from which they can draw information, examine the problem from many angles, distinguish relevant information from irrelevant information, and formulate a solution (Lombardi 2007). These tasks are usually completed in collaboration, and learners are given the opportunity to discuss problems, ideas and solutions, thus learning from one another, before completing the task.

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:
Table 3: Responses regarding Mindstorms reinforcing learning

- 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 related to, provided essential scaffolding so that gap between “understanding the problem” and determining “how to solve it” was not too wide.

4.2 Critical thinking

Computer programming is a subject that is rife with abstract concepts. For learners to better understand such concepts, it may be adventitious to provide them with meaningful problems to solve. As per the Neo-Piagetian theory any improvement in abstract reasoning skills results in improved computer programming competencies. Furthermore, allowing learners to solve such problems in a collaborative manner may further enhance abstract reasoning.

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 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.

4.3 Collaborative learning

The idea of constructing knowledge through a social setting can be a powerful educational and learning tool (Vygotsky 1978, Ben-Ari 1998, Kozulin 2003, Stetsenko 2010). This type of learning involves two important concepts, namely social constructivism and collaboration. Social constructivism is a philosophy and a learning theory (Vygotsky 1978). When a misperception occurs the individual will either ignore the conflict; construct a better model of prior understanding; or reflect on the existing model and create a new way of thinking about that model (Perkins 1999). However, ‘misperceptions’ may be dealt with in another way, one in which the individual works in collaboration with others.

It is important to note that the newly constructed knowledge may not necessarily be ‘correct’ as the individual’s past experiences may be based on ‘misperceptions’ or ‘misunderstandings’ (Karagiorgi 2005). When a misperception occurs the individual will either ignore the conflict; construct a better model of prior understanding; or reflect on the existing model and create a new way of thinking about that model (Perkins 1999). However, ‘misperceptions’ may be dealt with in another way, one in which the individual works in collaboration with others.

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. The biggest 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 that:
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 isolated 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.

4.4 Motivating learners and Lego Mindstorms

Hirumi (Hirumi 2008) argues 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 2014).

Motivation forms part of a conceptual model or framework known as the self-determination theory. This theory emphasizes 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 achievement that can either be encouraged or discouraged by facilitators practices. Such motivation results in high quality learning and creativity (Ryan 2000).

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

- 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.

5 Observations

The following observations came to light during the study:

- Learners collaboration increased significantly, relationships where formed not only with peers, but also with tutors and educators;
- Although learners had not been exposed to Lego Mindstorms prior to the case study, they quickly understood what was expected of them. Instead of being intimidated they embraced the experience;
- There was a high level of excitement as learners completed complex programming tasks. This in turn, lead to the promotion of a high level of motivation;
- There was an element of competitiveness amongst learners’ as they competed against each other. This further added to the motivation factor; and
- All of the above factors lead to an increase in learners’ confidence.

6 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. Learners’ reflections illustrate that Lego Mindstorms robots provide them with an opportunity to:

- Improve their problem solving / algorithmic skills;
- Engage with authentic real-world activities to test their programming abilities;
- Stimulate creative thought and develop critical thinking;
- Collaborate with peers and learn socially; and
- Motivate them to achieve and become creative in their thinking.

Lego Mindstorms robots is a platform that allows the focus of learning to shift from traditional learning to one where learners can engage in real-world problems. It allows them to develop dynamic programming skills in a language independent environment. In today’s rapidly changing world, providing innovative ways to scaffold learners and challenge them to think in other ways is essential.

7 Acknowledgements

The financial assistance of the National Research Foundation (NRF) in South Africa under Grant number 84244 is hereby acknowledged. Any opinion, findings and conclusions or recommendations expressed in this material are those of the authors and the NRF does not accept any liability thereto.

The assistance of Dr E.C. Anderssen, who was responsible for the classroom facilitation, the management thereof, and support of the Lego Mindstorms case study is also hereby acknowledged.

8 References


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(2).


