Learning Programming using Lego Mindstorms: Analysis of Learner Experiences

Farah Hanum Noor
Faculty of Cognitive Science and Human Development,
Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak
Malaysia
62770@siswa.unimas.my

Fitri Suraya Mohamad
Faculty of Cognitive Science and Human Development,
Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak,
Malaysia
mfitri@unimas.my

Jacey Lynn Minoi
Faculty of Computer Science and Information Technology,
Gamification Centre,
Universiti Malaysia Sarawak
94300 Kota Samarahan, Sarawak
Malaysia
jacey@unimas.my

2020 IEEE 8th R10 Humanitarian Technology Conference (R10-HTC) | 978-1-7281-1110-0/20/$31.00 ©2020 IEEE | DOI: 10.1109/R10-HTC49770.2020.9357024

***Abstract*— We experimented with the viability of using Lego Mindstorms as an alternative approach to supplement the learning of Basic Programming among novice university students. The study captured reactions and responses from ten undergraduates who had no background in Programming and were facing difficulties coping with an introductory course in Programming. Four Lego Mindstorms units were deployed to examine the participants' problem- solving and computational thinking skills. Findings indicate a high positive response (t= -3.88 and p = 0.004 for Problem Solving; t= -4.13 and p=0.003 for Computational thinking) to the use of Mindstorms as a strategy to encourage higher-order thinking, a critical skill in learning Programming. Participants reported clarity in understanding computational thinking concepts and claimed to be better able to articulate problem-solving tasks when prompted using Mindstorms.**

Keywords*— *Computational thinking, LEGO Mindstorms, Robotics, problem-solving skills

I. INTRODUCTION

For more than two decades, there has been a persistent reporting of high failure rates in introductory programming courses; some suggested to be as high as one-third of the course enrolment [1]. A recent study showed that despite advances in pedagogy, the worldwide failure rates have not improved over time [2]. Despite decades of research, internal factors based upon traditional learning theories have also failed to explain the CS1 (First Introductory Programming Course) failure rate phenomenon, and no factor to date has been shown to influence programming performance across a range of different teaching contexts [3]. There have been studies which explored the connections between desirable aspects of programming behaviour and performance [4]. However, although some of the proposed interventions yielded promising results, to date, there is still no overall understanding as to why so many students endlessly struggle in programming courses.

Many assumptions have been put forward as to why learning to program is a difficult task. Some have attributed to the nature of programming itself. Programming is not a single skill, but rather a complex cognitive activity, where a student must simultaneously build and apply several higher-order cognitive skills to solve a particular problem [5]. Other reasons are attributed to the aspects of the students. Students may have lacked the motivation to learn to code, and they may not be able to create a mental model of how programs relate to the underlying system [6] or create an explicit model of program flow [7].

Other reasons include the teaching methodology used by course instructors. Many students fail programming courses. Instructors face an additional challenge of adjusting their expectations to the students' level of ability [8]. These teaching challenges have been acknowledged, and decades of research effort has been put into creating and applying teaching interventions that facilitate students' learning [5,9]. These interventions can include moving from a traditional lecture and lab-based approach to using pair programming, game-based learning, or extreme apprenticeship. However, to date, there is minimal evidence of studies which have attempted to quantitatively compare the impact that these proposed instructional approaches on how they can improve the pass rates of failing programming courses. Without any quantitative evidence on the relative strengths of different instructional approaches, the research community will continue to have a dearth of a clear consensus of precisely which instructional strategies would provide the most effective means of teaching programming and consequently saving failing programming students.

This study was designed to observe the effect of LEGO Mindstorms EV3 on participants' problem-solving and computational thinking skills. The study also explored the impact on motivation and interest in computer programming. LEGO Mindstorms EV3 is a robot set which is commonly used in programming education. EV3 consists of building components, programmable brick, active sensors, and motors. There is a software programme for which both Graphical User Interface (GUI) and command-line interfaces are available. The robots, together with their 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 [10]. The 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'. In other words, LEGO Mindstorms is an interactive robot that produces output according to a learner's input. Its immediacy in responding provides an opportunity to visualise how codes affect action. EV3 is different from Scratch (block-based visual programming language) in a sense where it uses action instead of explanation and affords an interactive, decision making context