# Supporting Young Students to Learn Computer Programming in an Early Schooling

Mohammed Alghamdi, Dhiya Al-Jumeily, Abir J Hussain
*Applied Computing Research Group*
School of Computing and Mathematical Sciences
Liverpool John Moores University, Liverpool, UK
Email: M.Y.Alghamdi@2012.ljmu.ac.uk, {D.Aljumeily, A.Hussain}@ljmu.ac.uk

***Abstract*— The deployments of technology across the global toward efficient learning environment are growing rapidly. In United Kingdom educational system, the government is investing 1.1 million pounds in primary and secondary school for early programming lessoning starting from age five upwards. The ideology behind this innovation is to make younger generation linked to innovation and digital industries and improve the pace with the most successful education system in the world. In this paper, an assessment-driven educational programming tutoring system for young students is described in a high-level overview (which provides an abstract design of the main components of the proposed system), as well as a detailed discussion of some of the existing tutoring programming tools for kids. The intention will be to develop this given proposed design fully functional and then evaluate it in different primary schools from the UK and other Middle East countries. One of our main intended aims of this study would be to offer a suitable educational programming system that could ease the process of teaching and learning programming for kids in primary schooling.**

***Keywords— Technology and Education; Coding; young students; Teaching and Learning; Computer programming; Programming tutoring systems for kids; Adaptive software***

## I. INTRODUCTION

In today's increasingly technological society, kids would benefit from understanding computational thinking and problem solving [1]. Furthermore, once kids have good knowledge of computational thinking and problem solving, this would enable them to easily understand how and why technology works. Teaching computer programming to kids would be one of the ways to equip them with problem solving skills [2]. Furthermore, kids from western countries such as the UK have been encouraged to start learning how to code while they are in a primary school [3]. Consequently, their governments have changed their computer curricula. Additionally, teachers who are teaching kids how to code, they would require some training programs as well as a proper technological programming tool that can aid their students to learn how to code smoothly. This on-going research work therefore focuses on investigating challenges associated with programming/computational thinking tools for teaching and learning kids in a primary school. The proposed system aims to solve some of these issues that are associated with some of the existing educational programming tools for kids. This paper situates our work as a position paper, giving an early overview of the authors' plans for this research project. As such, the rest of the paper is organised as follows: Section II describes Teaching Programming in an early Schooling, while Section III discusses the importance of coding for young students. Section IV illustrates some of the existing educational programming tools for kids. Section V describes the authors' considerations of relevant pedagogical concerns, while Section VI shows how they will be reflected in the system. Section VII describes the design of the proposed system and Section VIII summarises the paper.

## II. TEACHING PROGRAMMING IN ERARLY SCHOOLING

There are several definitions for the programming concept, with the simplest start by defining programming as a form of a problem solving. This involves knowing where the problem is (locating the problem), analysing the problem, designing the solution, writing the actual code and then testing the solution [4].

There are two levels of programming Languages [5]. The first one is a low level of programming language that can be understood by the machine. For example, machine code. Languages belong to this level are hard to learn [5].

The second level is a high level of programming languages, which include C, Java, Visual Basic and others. Those languages are converted into a low level programming languages. However, some other programming languages are interpreted by another application. For example, java script which uses web browsers for interpreting the programme [5].

For this paper, our focus will be on the simple programming language that can be used for school level students to learn computer programming easily.

As the UK government has decided to make the year of 2014, the year of code for the kids in a primary schooling [6]. Furthermore, children today are born digitally native and use technological tools such as tablets before even they talk [6]. With this new change in the computing subject, there would be a need for good preparation for both kids and teachers to be ready to take this new challenge. Teachers would require some support and training in how to deliver coding skills to their young students. Kids would also need some help from their teachers in the classroom as well as a tool that can motivate their learning either at home or in the classroom.

The importance of coding in early schooling will be discussed in the following section.

978-1-4799-7186-2/15/$31.00 ©2015 IEEE

## III. THE IMPORTANCE OF CODING IN EARLY SCHOOLING

Before discussing the importance of programming in early schooling, it would be quite important to briefly discuss here some of the challenges that novice programmers are facing when they are doing a programming course at any level of learning [7]. One of the challenges would be a lack of knowledge of the basics of programming, for instance, looping, if statements and etc. Another challenge could be that novice programmers have not been practically taught how to code in their previous studies, for example, in a primary school [7]. Consequently, educationalists and parents are required to pay attention to those above discussed issues and work together to find an optimal solution. One of the possible solutions would be starting teaching a programming in a primary school not leaving those students until they get to a university level. There are some existing programming tools for teaching kids how to code (those will be in discussed in the next section) however, those tools are still having some pedagogical challenges. One of those challenges would be the idea of assessment for learning is not considered. Another challenge could be the idea of cooperative learning between parents, and teachers are missing from those available tools. Therefore, the planned aims of this ongoing research are to tackle some of those above discussed issues and develop an automated tutoring tool for teaching kids programming (Software development). Additionally, the planned tool is designed for kids age of 7 to 11 and one of the main reasons to target those pupils at this early age is to simplify the process of learning programming for them as early as possible (Nowadays kids are having a technology in their hands while they are in an early age). Another one is to make them at least they know the fundamental aspects of programming before reaching a college/university level. In the planned system, those kids will be given various methods to get access to the learning environment (this environment will be considered in section VII). One of the given choices is to access learning materials through some smart devices. Other choices are to learn through either the tablet or the main website. By using our planned tool in a primary school on kids, it would be expected many benefits for those kids. One of the main benefits would be kids will be able to think logically, solve problems intelligently and do so many important skills (developing their computational thinking skills). Also, coding skill is similar to a writing skill where kids can write down their thoughts, drawing diagrams and organize their ideas and etc.

## IV. SOME OF EXISTING EDUCATIONAL PROGRAMMING TOOLS FOR EARLY SCHOOLING

Educational programming languages/ tools would be those tools that are mainly designed as a learning tool for kids not as tool for developing applications for real world work. There are many tools for teaching kids how to program [8, 9, 10]. Some of these educational programming languages are detailed in table 1.

| System | Overview | Date | Shortcoming |
|---|---|---|---|
| **Logo** | It is an educational tool that can be used to introduce kids to programming. | 1967 | The idea of assessment for learning is missing from those tools. Those tools have not considered the importance of informing parents about their children's performances. (Cooperative/Shared learning). |
| **RoboMind** | It is a programming environment that would allow kids to learn the basic of computer science. | 2005 | |
| **Scratch.** | It is a graphical tool that can be used by kids to make animated stories, games and etc. | 2006 | |

TABLE I: Existing programming teaching tools for kids.

"Logo" is the first example of an educational programming tool. It was used to teach the programming concepts associated with LISP [9].
"RoboMind" is an educational programming environment. It enables novices to learn the basic of computer science. This programming environment offers a simple scripting language (a set of rules) and here where the movement of robot can be controlled (forward-backward) [10].
"Scratch" is another learning language that can be used by kids to make their own interactive stories, games, animation and share their designs with one another [9].
Those above discussed educational programming tools are good to be used by kids however, they require some further improvements. For example, the idea of assessment for learning is not included. So, monitoring the progress of kids is important to be considered by any educational programming tool. Furthermore, the idea of updating parents about their coding progress is not included in those above discussed tools. Therefore, in the planned programming tool for kids, it will be aimed to focus on those mentioned limitations. The outcome of this study would be developing an easily educational accessible programming language for kids to practically learn the fundamental aspects of programming (computational thinking). Pedagogical aspects in education will be discussed in the next section.

## V. PEDAGOGICAL CONCERNS

This work focuses on the pedagogical requirements of a software system that can adapt curriculum and learning to a student's needs. The following subsections discuss these high-level pedagogical concerns.

### *A. The Taxonomy of Education*

Bloom's taxonomy, a pillar in pedagogical science, is a classification of educational goals, which help teachers and lecturers in structuring their approach to learning. This can be reflected in both preparation and delivery of materials to students; also how to assess and encourage students to increase their attainment levels. Bloom divided

these goals into three domains: Cognitive, Affective and Psychomotor [11]. In 2001, Krathwohl et al revised Bloom's taxonomy, changing the cognitive domain; updating the six levels based on feedback from teaching practitioners and their interactions with students. The new levels were: Remembering, Understanding, Applying, Analysing, Evaluating and Creating [12]. Thompson et al further interpreted Bloom's taxonomy, citing the difficulties in applying these levels of cognition to software engineering and programming; explaining the taxonomy with examples specific to programming [13].

### *B. Assessment*

Assessment is used to evaluate student performance and to provide feedback for students to reflect on their work. However, when setting an assessment, its purpose must be clearly established. Teachers should therefore select the assessment type most suitable for the student cohort and the learning outcomes being evaluated [14]. The rest of this section briefly describes several assessment types considered relevant to this work. They are: formative, summative, diagnostic, and finally continuous assessment. Formative assessment helps teachers and students to see shortcomings in submitted work; enabling helpful feedback so students can identify and focus on their weak areas. Additionally, it helps teachers to measure their own performance for a given cohort. Summative assessment normally occurs at the end of a module of study; providing a quantitative measure of performance [15][16]. Diagnostic assessments are performed at the start of a learning plan, and identify learners' current understanding and attainment levels, and in some cases, a student's learning difficulties [14]. Finally, Continual assessment may occur several times during a course; providing an on-going measure of student performance and is used to direct future learning [17].

### *C. Learning styles in education*

Rebecca & James [18] defined learning styles as the characteristic techniques in which learners learn, understand and get information. Some researchers defined a learning style as an approach of learning a concept. This is because each learner has a different preferred approach to understanding or learning things. For example, some learners prefer to, and perform better when learning visually, while others may prefer to learn aurally [19] [18].

Considering learning styles of all students in the traditional classroom can be a challenging issue for teachers. Teachers have only a limited time in preparing their materials and delivering their classes, lectures and tutorials [20]. Established pedagogical theory specifies several learning style models [21], including Kolb Experiential Learning Theory [19], the VARK Model [19, 22] , Felder–Silverman Learning/Teaching Style Model [19] and Dunn and Dunn Learning Style Model [19]. Moreover, each of these models has different descriptions for the learning styles [21]. The following subsection discusses VARK model in more detail.

### *D. The VARK Model*

The acronym VARK stands for Visual (V), Aural (A), Read/Write (R), and Kinesthetic (K). Learning style has been defined in this model as a learner's preferred ways of remembering, understanding, and reasoning about knowledge [19]. The VARK model has been used for advising teachers for knowing the preferred learning styles of their students [23, 24] Significantly, this model has a supporting validated questionnaire (and website version at http://www.vark-learn.com/english) that allows a reasonably quick (self) assessment of learning style preference. This can be done by filling in the online questionnaire, which then shows the website or allows calculation of the VARK learning style. VARK defines four learning styles [19, 22, 23], as follows:

**Visual:** one of the original basic learning styles. In this particular type, a learner learns best by seeing. For example, flowcharts, diagrams, maps and so on [19].

**Aural:** another significant learning style in traditional classroom education. Here, a learner prefers to learn best through listening to lectures, discussion, tapes and etc [19].

**Read/Write:** These learners prefer self-directed learning – e.g. reading textbooks, reports, or webpages and then summarize or write what they have understood [19].

**Kinaesthetic:** This is another primary learning style in the classroom. Kinaesthetic learners do so best through experience; undertaking experiments, carrying out case studies, practical sessions, etc [19].

The next section discusses the pedagogy in the planned system.

## VI. PEDAGOGY IN THE PROPOSED SYSTEM

The main inspiration for the proposed system is the *Assessment for Learning* initiative, comprising diagnostic and continual assessment [25] and practised widely by England's school teachers [26]. This defines a structured learning approach based on a student's prior knowledge and ability, followed by learning informed by a student's assessment performance. This methodology is applied in the proposed system, such that curriculum sequencing and material generation is fully integrated into an adaptive, student-centric learning tool. This is shown in Figure 1 as an overview process flow diagram.

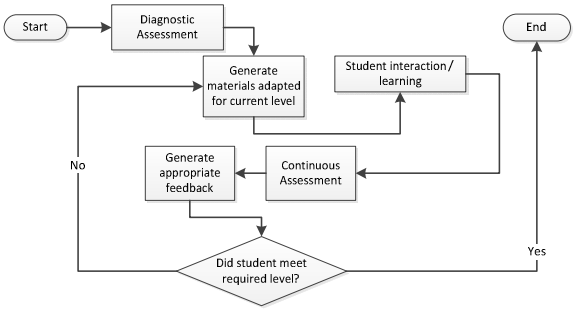


Fig. 1: Learner-followed process in proposed system

When a student first engages in a course, they undertake a Diagnostic Assessment to establish the entry level of programming ability (e.g. control flow statements). A learning level appropriate to the student's needs is

selected to generate appropriate curriculum material – providing guided, self-directed learning. The student is engaged in formative "Continuous Assessment", providing appropriate feedback and adapting the curriculum appropriately. Parents will also be informed about coding progress of their kids.

## VII. DESIGN OF OUR PROPOSED PROGRAMMING TOOL

The design of this planned electronic computational thinking system will be divided into three environments. One will be called a "Management Environment". The second one would be a "Learning and Engagement". The last one could be the "Evaluation Environment". Those above environments will be detailed as follows:

### *A. Management for staff*

This environment is fully controlled by an administrator. This administrator will be given the highest privileges in this area. Some of its tasks would be adding as many teachers as it is required, assigning a particular teacher to a certain class, providing added teachers some other privileges to for instance adding students to their classes and managing learning contents, monitoring the other two created environments and doing other various duties.

### *B. Learning and engagement for students*

This learning area will be organised by those teachers who were added in the previous environment. Those teachers will take the responsibility for assessing the initial level of their students before using this environment and knowing what they need to learn from the fundamental aspects of programming (looping-sequence and others). Some of the planned pedagogical activities in this area will be summarized in the following points.

*1) Specified learning outcomes:*

There will be some designed learning outcomes for the students. An example would be a learner needs to know the looping concept or other such fundamental programming concepts.

*2) A list of Lessens would be as follows:*

a) Sequence (Right-left-up-down).
b) Repeat (looping).

*3) Activities*

There could be four activities. For each activity or exercise, learners would be given a time to think how to reach the goal, analyse the situation ahead and then solve the problem. The teacher will set those activities and each activity would be different from the next one in terms of the complexity and others. Those learners will be assessed and given a score on their performances against the generated exercises. Furthermore, the idea of learning styles will be applied here where visual and physical learners will get the chance to see the things and learn by doing.

Figure 2 shows one of the activities of how our system can teach the concept of Sequence to the learner.

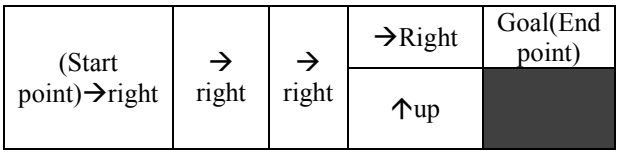


Fig 2: shows an example of the Sequence lesson.

It can be seen from the above illustration (figure1) that a learner needs to do a number of steps to reach the desired goal. First, he/she needs to turn right. Second, it is required to turn right again and then repeat the same action again (turn right). After that, he goes up and then followed by turning right to get to the destination or the desired goal.

However, the subsequent figure (2) illustrates another different activity of how our system can teach the concept of looping to the student.

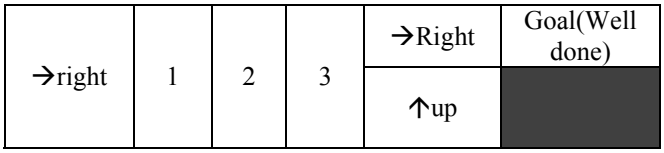


Fig 3: shows an example of the Repeat lesson.

In figure 3, the learning process will be done in an intelligent way where the learner is not required to repeat the action that he/she did in figure 1. All they need to do is to think of solving the situation ahead and then, inform the developed tool how many times to turn right (for instance 3 times) and this will be performed by the users.

### *C. Evaluation for students and parents*

This designed environment will completely be related to the assessment of student's activities. This means that in this environment teachers and parents can monitor the performance of their students or kids and discuss what it should or should not be done in the future (the idea of cooperative learning is applied in our online school where parents and teachers can see how well kids are coding). Furthermore, the educational concept of assessment for learning is also considered in our online technology (as kids will be assessed continually and helping them to find out what to do next).

## VIII. SUMMARY

Teaching programming to kids or young students has both costs and advantages. One of the advantages would possibly be enabling those young learners to have a better understanding of the fundamental aspects of programming before they reach to college/university level. The costs would include that teachers in a primary school would require an additional training in how to deliver the aspects of programming in a way that suites those young students as well as they would need a support from a technological tool that aids their learners to easily be engaged while they are learning how to program. Consequently, this work examines the issues surrounding the development of an

educational programming tool for kids. This paper discussed issues surrounding educational programming tools for kids, and examined surrounding pedagogical issues to help determine failings and incompleteness in current technological teaching and learning tools for young learners.

During the course of this research, a prototype of the proposed system will be implemented and evaluated in real-life student contact situations. These evaluations will occur in primary schools in both the UK and two other countries from Middle East.

## IX. REFERENCES


[1] J. Bransford, J., Sherwood, R., Vye, N., and Rieser, "Teaching Thinking and Problem Solving: Research Foundations.," in American Psychologist, 1986, pp. 1078–1089.

[2] S. Tarkan, V. Sazawal, A. Druin, E. Golub, E. M. Bonsignore, G. Walsh, Z. Atrash, and C. Park, "Toque□: Designing a Cooking-Based Programming Language For and With Children," pp. 2417–2426, 2010.

[3] C. Duncan, T. Bell, and S. Tanimoto, "Should your 8-year-old learn coding?," in Proceedings of the 9th Workshop in Primary and Secondary Computing Education on - WiPSCE '14, 2014, pp. 60–69.

[4] M.-W. C. Dictionary, "Overview of Programming and," Tenth., Merriam-Webster Inc, 1994.

[5] L. Yanfei, "Development of low-level digital I/O experiments involving a high-level programming language," Computers in Education Journal, vol. 19, no. 1, pp. 77–86, 2009.

[6] A. E. Mulqueeny, "Teaching kids to code," COMPUTERWEEKLY.COM, no. March, pp. 23–27, 2014.

[7] E. Lahtinen and K. Ala-mutka, "A Study of the Difficulties of Novice Programmers," 2005, pp. 14–18.

[8] B. Y. Resnick, K. Brennan, and Y. Kafai, "scratch□: Programming for all," COMMUNICATIONS OF THE ACM, 2009.

[9] R. D. Pea, "Logo Programming and Problem Solving," 2007.

[10] R. K. Arvid Halma, "RoboMind Programming," 2005. [Online]. Available: http://www.robomind.net/en/index.html.

[11] M. Forehand, "Bloom ' s Taxonomy," no. 2002, Athens: The University of Georgia, 2010, pp. 1–9.

[12] D. R. Krathwohl, "A Revision of Bloom's Taxonomy: An Overview," Theory Into Practice, vol. 41, no. 4, pp. 212–218, Nov. 2002.

[13] E. Thompson, H. Grove, A. Luxton-reilly, J. L. Whalley, and P. Robbins, "Bloom ' s Taxonomy for CS Assessment," in Australian Computer Society, 2008, vol. 78, no. January.

[14] C. Kyriacou, Essential Teaching Skills, Third. London: Nelson Thornes, 2007.

[15] W. Harlen and M. James, "Assessment and Learning□: differences and relationships assessment Assessment and Learning□: differences and relationships between formative and summative assessment," Assessment in Education, no. February 2013, pp. 37–41, 1997.

[16] D. J. Nicol and D. Macfarlane□Dick, "Formative assessment and self□regulated learning: a model and seven principles of good feedback practice," Studies in Higher Education, vol. 31, no. 2, pp. 199–218, Apr. 2006.

[17] C. Coll, M. J. Rochera, R. M. Mayordomo, and M. Naranjo, "Continuous assessment and support for learning□: an experience in educational innovation with ICT support in higher education."

[18] R. H. Rutherfoord and J. K. Rutherfoord, "Exploring teaching methods for on-line course delivery-using universal instructional design," Proceedings of the 9th ACM SIGITE conference on Information technology education - SIGITE '08, p. 45, 2008.

[19] T. F. Hawk, "to Enhance Student Learning," vol. 5, no. 1, pp. 1–19, 2007.

[20] C. Watson, F. W. B. Li, and R. W. H. Lau, "A pedagogical interface for authoring adaptive e-learning courses," Proceedings of the second ACM international workshop on Multimedia technologies for distance leaning - MTDL '10, p. 13, 2010.

[21] S. Graf, S. R. Viola, and T. Leo, "In-Depth Analysis of the Felder-Silverman Learning Style Dimensions," vol. 40, no. 1, 2007.

[22] W. L. Leite, M. Svinicki, and Y. Shi, "Attempted Validation of the Scores of the VARK: Learning Styles Inventory With Multitrait-Multimethod Confirmatory Factor Analysis Models," Educational and Psychological Measurement, vol. 70, no. 2, pp. 323–339, Aug. 2009.

[23] Y. Eltigani, A. Mustafa, and S. M. Sharif, "An approach to Adaptive E-Learning Hypermedia System based on Learning Styles ( AEHS-LS ): Implementation and evaluation," vol. 3, no. January, pp. 15–28, 2011.

[24] J. Mylles, "Building teacher leadership in Hertfordshire(R)," Improving Schools, vol. 9, no. 1, pp. 69–76, Mar. 2006.

[25] D. Wiliam *, C. Lee, C. Harrison, and P. Black, "Teachers developing assessment for learning: impact on student achievement," Assessment in Education: Principles, Policy & Practice, vol. 11, no. 1, pp. 49–65, Mar. 2004.

[26] E. Hargreaves, "Assessment for learning? Thinking outside the (black) box," Cambridge Journal of Education, 2005.