Talk-to-the-Robot: Speech-Interactive Robot To Teach Children Computational Thinking

Nada Sharaf[1], Ghada Ahmed[2], Omnia Ashraf[2] and Walid Ahmed[2]
[1]*The German International University, Egypt*
[2]*The German University in Cairo, Egypt*

Keywords: Computational Thinking, Robotics, Programming for Children, Educational Robots.

Abstract: Nowadays, technology is fundamental in almost every aspect of our lives. Our daily life is endowed with many devices as mobiles, laptops, televisions that depend solely on technology. Therefore, Computational Thinking is growing faster into a form of an imperative literacy that needs to be learnt by young children as well as Programming in order for them to be more prepared for the future. From an educational point of view, it is very essential to improve and develop computational thinking and problem algorithmic-solving skills of young students. This study investigates how successfully small kids between 7-11 years old can comprehend and use basic programming concepts. Through this study, children have to learn three major programming concepts (sequential, conditional, iteration) by using a robot in a maze game that targets these concepts. The robot has a voice recognition feature that can easily be used by children to direct the robot out of the maze by using the three different programming approaches. The experiment has 36 participants, then by means of a "between-group experiment", the participants have been divided into two groups. One of which has learnt the concepts by using the robot to test the game (Experimental Group) while the other has learnt the concepts by using the traditional methods of teaching (Control Group). The results of testing the learning processes between the two groups have been compared and reported regarding the learning gain, engagement level and system usability scale. The experiment has proved that the group who uses the robot has achieved significantly better learning gain and better engagement than the group who has been taught by explaining the concepts on a paper. Accordingly, using an educational robot is considered to be an effective and operative method for teaching young children the basics of programming and computational thinking.

1 INTRODUCTION

Technology is definitely prevailing in almost every aspect of our life. Most of the technologies around children are made out of programmed chips and programmed software along with the hardware assembled together to form out these types of technological devices such as video games, tablets and mobile applications. Meanwhile, programming has become an indispensable part that is incorporated in almost every daily life activity. All the applications we use daily are made and developed by programmers, as a result it becomes very important to learn more about programming and to understand more about the technologies we use every day. Nowadays, computer programming is considered to be one of the most important aptitudes for learning. From an educational point of view, It is very important in improving and developing computational thinking and problem algorithmic-solving skills of young students(Fessakis et al., 2013).

Most kids interact with technologies every day in their life. They stay hours on their mobile phones using many applications or playing games without having any knowledge about how these applications work. Studies have shown that programming and designing games by children can have a deep impact on their way of thinking and their problem solving skills. Furthermore, computational thinking creates positive attitudes towards computers and electronics, increases creativity; academic knowledge and skills and finally makes kids interact more with computational practices (Denner et al., 2019). Learning computational thinking does not necessarily mean that all kids have to be programmers. Yet, understanding computational thinking by children does not only improve problem solving skills, it also creates an efficient way of thinking about real life problems and finding corresponding solutions. As a result, teaching children

Sharaf, N., Ahmed, G., Ashraf, O. and Ahmed, W.
Talk-to-the-Robot: Speech-Interactive Robot To Teach Children Computational Thinking.
DOI: 10.5220/0011852600003470
In *Proceedings of the 15th International Conference on Computer Supported Education (CSEDU 2023) - Volume 1*, pages 57-69
ISBN: 978-989-758-641-5; ISSN: 2184-5026

computational thinking and programming has become a priority.

One way to enhance children's computational thinking is by using some graphical applications that aim to teach children programming by playing a game, for example: Scracth, Blockly and others. Another way to teach children programming is through interacting with a robot programmed and developed to teach them the basic concepts of programming and how to think properly. Robots are simultaneously entrancing articles for the overall population and gadgets whose origination, understanding and programming include numerous fields. This exceptional blend makes them perfect devices for presenting science and innovation to youthful ages (Magnenat et al., 2012).

The main focus of this paper is to teach children computational thinking and programming using an interactive robot with certain features. The robot contains few sensors as well as different hardware parts. Children can easily communicate with the robot through voice commands since it has a voice recognition feature to make the robot actually move or do certain tasks as the main purpose of the robot is to teach children the computational way of thinking and enhance their problem solving skills.

The purpose of this research is to evaluate the impact of using an interactive robot to teach children computational thinking and basic programming concepts. The robot is used to teach children programming in an easier way than using a graphical programming application with a screen. As mentioned before, the robot contains few sensors along with some motors and hardware that make the robot interactive with a voice recognition feature. Children can use voice commands to make the robot move and do other functions. Moreover, robots not only offer a teaching environment but they also make learning much more interesting for kids. They engage with the robot trying to understand the different aspects of the daily technologies they use as well as understanding how the robot works with the different integrated software and hardware. Eventually, they know how to properly think and solve problems. In addition the robot motivates children to improve interactive skills, creativity and more. Furthermore, the main aim of the work is to compare between teaching children computational thinking by using an interactive robot and teaching children computational thinking by using the old traditional method which is basically explaining the basic concepts on a piece of paper. Two versions of the robot were constructed. One was designed to teach children programming through an interaction in English language and the other in Arabic language.

This paper is divided into six sections. Section one is an introductory section that discusses the importance of integrating computational thinking in the education of young children. Section two introduces the related work of using robots and games in education. Section three includes the methodology of the game and how the robot works. Section four contains the experimental design. Section five shows how the testing is done and reports the results. Section seven concludes the paper with directions to future work.

2 RELATED WORK

There are different practices carried out with a goal to teach computational thinking and other educational purposes. Educational robots are one of these approaches, they play a vital role in education, and their usage has increased throughout the last years. This chapter introduces the various work related to the field of teaching children the fundamentals of programming and other subjects which is done by different methods. It also reveals some work that implements the speech recognition feature.

2.1 Teaching Computational Thinking

Visual programming environments were used to enable children to build programs through simple interfaces (KALELIOĞLU and Gülbahar, 2014). Such platforms include Scratch (MIT Media Lab,) and Alice (Carnegie Mellon University,). According to Armoin et al. (2015), teachers reported that using Scratch increased the learning efficiency. Students also reported that they were encouraged to learn more about computer science.

"Program your robot" (Kazimoglu et al., 2012) is a serious game developed for teaching programming and computational thinking. The game aimed at integrating the game-play with the programming concepts and computational thinking skills. The game was evaluated using 25 students. The students enjoyed the game and reported that this type of games can improve their problem-solving skills.

Another example of an educational game for programming used tangible electronic building blocks instead of visual programming is presented in (Wyeth and Purchase, 2002). The Tangible electronic blocks are divided into sensors and sources for these sensors. The child can build a structured block that does a function by connecting the blocks together. The older children were able to debug their structure in case of a problem or undesired function. The results showed that young aged children did not realize the concepts of programming using the Electronic Blocks. Therefore, this way was not effective enough for the young children.

2.2 Robotic Approaches

This section introduces the approaches that use robots for teaching purposes.

2.2.1 IbnSina

IbnSina is the world's first Arabic language conversational android robot that has an aim to become an exciting educational and persuasive robot in the future (Riek et al., 2010). It is a robot that offers an Arabic conversation with people, it receives speech from the users and responds with a relevant response. Acapela speech recognition engine is the software integrated in the IbnSina robot in order to understand the speech and react accordingly. The Acapela uses an acoustic model that contains statistical representations of the sounds that make up each acoustic unit, and a language model that contains the probabilities of sequences of words. At the Language model layer of the speech system of the robot, an artificial grammar that restricts the recognition to a list of sentences that were modeled in grammar is implemented(AcapelaGroup, 2009). Moreover a collection of sentences were developed for IbnSina to say, these phrases were encoded as UTF-8 text files, each having a keyword to provide the appropriate response for the speech it recognized. The phrases in the corpus are converted to speech when they are retrieved using the Acapela text to speech engine. Figure 3.1 shows how the system of IbnSina works.

2.2.2 IROBI

IROBI is an educational and a home robot. It can teach English and tell the children nursery rhymes, entertain the family by singing and dancing, and provide home security by monitoring, detecting and photographing unwanted intruders(Han et al., 2008). The children can interact with it through voice commands and if the robot failed to recognize the voice the robot can be touched through the touch module that is incorporated in the robot. The touch interaction ameliorates children's concentration and academic achievement(Han et al., 2008).

2.2.3 Thymio

Thymio is an educational robot with a considerable amount of sensors and actuators to teach children how to program. The robot can be programmed by several programming languages like Scratch. It can teach the basics of computational thinking in forms of lessons: simple guiding commands, basic repetitive commands, repetitive commands combined with statements, basic conditions and variables, conditions and functions(Constantinou and Ioannou, 2020). Figure 3 shows the computational thinking exercises done by the Thymio robot programmed by Scratch.

2.2.4 Kodockly

Kodockly is an interactive programmable robot used to teach children computational thinking and programming through a game with several levels. It also teaches some hardware programming concepts. The robot has two versions Kodeocly 1.0 which is aimed for children between the ages 6 and 8, and Kodockly 2.0 which targets children between the age 8 and 11 years. The robots are programmed using TUI (Tangible User Interface) which are wooden blocks used by the children. Each wooden block has an electronic identification card that can be scanned by the scanner installed in the robots. The process of teaching the children programming is done through a game with different levels for Kodockly 1.0 and to teach programming and sensors based computational thinking, the child can make a program of whatever number of wooden blocks he wants from the available blocks of Kodockly 2.0(Mohamed et al., 2021).

2.2.5 Interaction with Robots

The work presented in (Rogalla et al., 2002) showed different ways to interact with robots including speech. However, the language used was German and the users were not children.

The work presented in (Wu et al., 2008) deploys interaction through gestures using Wii remote as well as voice. It targets teaching children as well. However, the aim is to teach them English and Mathematics. The robot is instructed by the teacher on how to respond using a visual language. The communication however needs a lot of modules and is not very customizable for other purposes.

Using social interaction along with robots and story telling was also employed with Kindergarten children (Kory and Breazeal, 2014; Vaucelle and Jehan, 2002; Hsieh et al., 2010). However, the aim is to improve the language skills of the children and the age group is smaller than our target age.

Robots were also used to enhance social interaction skills with children that have autism (Robins et al., 2005). The results showed enhanced skills.

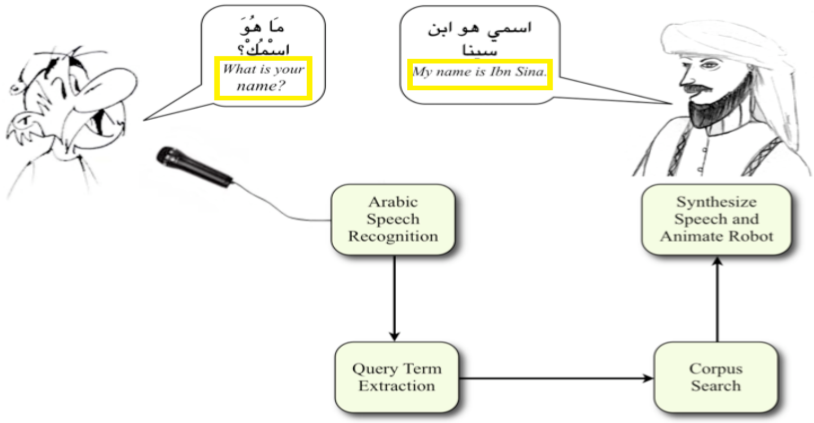

Figure 1: Overview of the system of IbnSina robot(AcapelaGroup, 2009).

Figure 2: Interaction between IROBI and a child(Han et al., 2008).

2.3 Problem

The problem in most of the ways used to teach children programming lack the interaction between the user and the robot or the graphical application.

3 DESIGN AND IMPLEMENTATION

In this study, two friendly-looking robots were designed and implemented. One robot recognises voice commands in English while the other recognises the commands in Arabic. Both robots were designed to teach children in the age range of 7 to 11 years old basics of programming. Children use the voice recognition feature to program the robot to move in a maze game made out of paper.

Figure 3: Thymio robot(Constantinou and Ioannou, 2020).

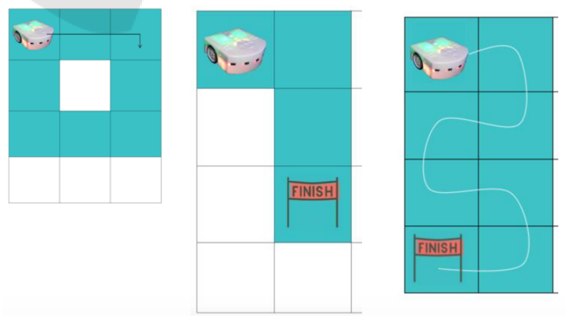

Figure 4: The computational thinking activities conducted by Thymio(Constantinou and Ioannou, 2020).

3.1 Robots Design

The robots were designed to have a pleasant look to encourage the kids to interact with it and have a fun experience as shown in figures 7, 8. The robots bod-

Figure 5: Kodockly 1.0 along with the maze(Mohamed et al., 2021).

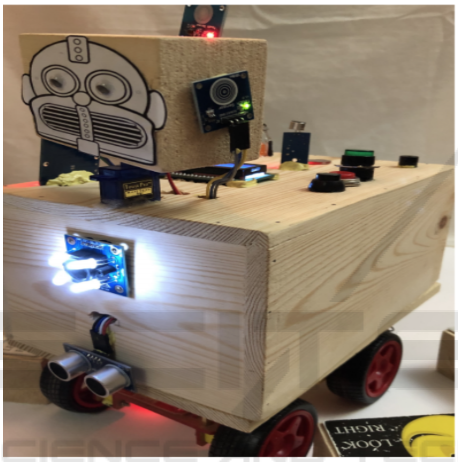

Figure 6: Front view of Kodockly 2.0(Mohamed et al., 2021).

ies are built of different sensors and electronic components that help in the process of the game.

The body of the robot is built by connecting pieces of card-boards together by means of hot glue. The robot itself does not have many sensors, it contains only one ultrasonic sensor to detect near-by objects in-case it is about to crash. The robot contains an arduino uno board which is connected to a nodeMCU (Wifi module) in order to receive commands using the wifi. The robot also includes an H-bridge connected to the 4 Dc motors in order to make the robot move, along with 3 batteries that give out 11.1. The H-bridge is connected to the arduino board via the output 5 voltages port in the H-bridge and the ground. The arduino board is also connected to a DF mini player which is linked to a speaker to give out feedback. Two power banks are used in this project as a source of power, one is connected to the nodeMCU while the other is connected to the arduino board to give out enough energy to make the speaker work. The main hardware

Figure 7: The Arabic Robot.

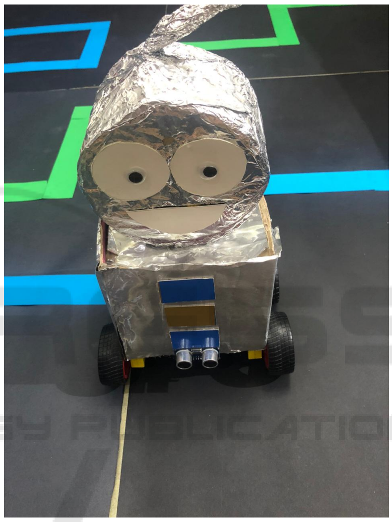

Figure 8: SYD: The english robot.

components used in the robots design are:

1. **Arduino**, Arduino is an open source platform that consists of hardware and software. It is easily used by everybody because of the fact that it is a simplified version of C++ (Arduino IDE) and the micro-controller hardware already introduced by the company(Badamasi, 2014).
2. **NodeMCU**, the NodeMCU is an easily programmable Wi-Fi arduino module (Barai et al., 2017). NodeMCU is used to send data coming from the sensors to the server so the user decides what he wants to do next. NodeMCU is used because of its cheap price and simplicity of communication because of its embedded Wi-Fi module(Edward et al., 2017).
3. **H-bridge**, the H-bridge is connected to the DC motors of the robot in order to make the robot move in the desired direction.

4. **TCS3200 Color Sensor**, This module detects the colors in RGB (red, green, blue) scale, it uses its white light emitter to light up the surface and the wavelengths of red, green and blue colors are measured using the three filters the sensor has (Red, Blue, Green) then a voltage equivalent to the identified color is generated through the light to voltage converter integrated in the sensor.
5. **LDR (Light-Dependent Resistor)**, An LDR (light-dependent resistor) is used to recognize if the robot is moving in a dark or bright place. It detects light levels, the resistance varies when light falls upon the module and it decreases as the light intensity increases so if the robot is in a dark place the resistance will be high.
6. **UltraSonic Sensor**, the Ultrasonic Sensor is used to detect near by objects. It is used to sense the objects at proximity from the robot in order to avoid hitting anything. When the module is powered, the transmitter on the sensor sends ultrasonic sound wave at a frequency above the range of human hearing and as soon as it encounters an object, it gets reflected back to the sensor where it is received by the Ultrasonic receiver.
7. **DC Motors and Wheels**, the robot consists of 4 DC motors that are connected to wheels and to a H-bridge.
8. **DF Mini Player**, In order to enhance the interaction experience with the robot, sound modules were added to the system. The DF mini player is used to play pre-recorded sounds by reading a memory card mp3 file.
9. **Speaker**, a speaker is connected to the DF mini player in order to play sounds and gives feedback to the user.

3.2 SYD: The English Robot

SYD is programmed to recognize and process programs in English language using voice or written commands. The keywords that can be used to trigger the robot's effect are shown in Table 1. The maze (shown In Figure 12) consists of 3 levels that become harder as the child moves on from a level to another. Each level allows the children to learn a new programming concept. The goal of the game is to get the robot out of the maze. The maze itself is static and the child has to get the robot out of the maze through the different programming approaches.

The robot is designed to receive voice commands from a web API. First the command has to be said using the ”start recognition” feature.The desired command will be sent to Firebase database. The Firebase database then sends the desired command to the NodeMCU, which in return sends the command to the arduino board and then the program starts executing.

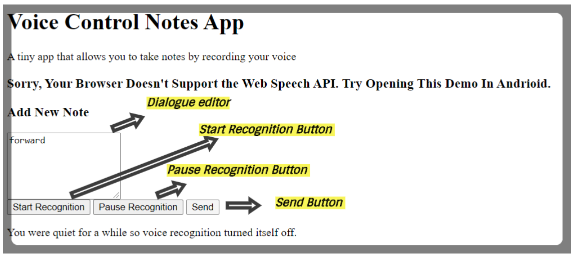

Figure 9: Web API for voice control.

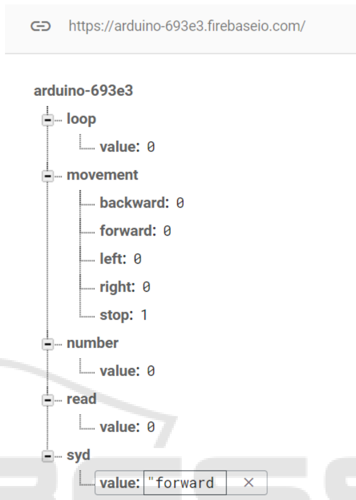

Figure 10: Firebase database.

The interaction with SYD robot is done in 2 direction way; meaning that the robot replies to the child command as well. To achieve that, a speaker is installed to provide some feedback for the child. The speaker is connected to DF Mini Player, which has a memory card with the saved dialogues that the robot

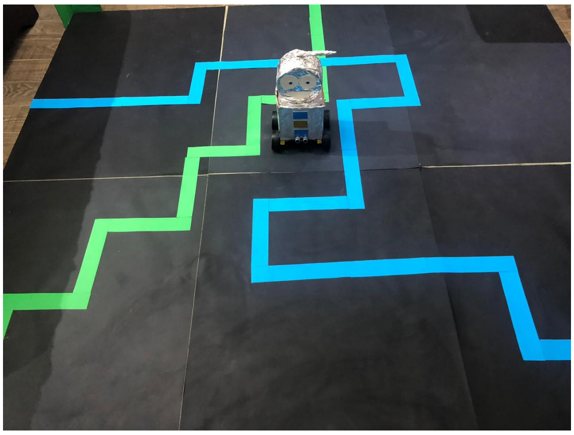

Figure 11: Maze game with the robot.

Table 1: Functions of every command.

Command Name	Function Of The Command
Hello	Identification of the robot's name
Forward	Moves the robot 1 step forward
Backward	Moves the robot 1 step backward
Right	Turns the robot to the right
Left	Turns the robot to the left
Start	Starts the loop feature in the robot
Always	Starts a forever loop
End	Ends the Start and Always commands
Forward if obstacle right	Moves forward forever if obstacle ahead moves right (always)
Forward if obstacle left	Moves forward forever if obstacle ahead moves left (always)
Backward if obstacle Left	Moves forward forever if obstacle ahead moves left (always)
Backward if obstacle right	Moves forward forever if obstacle ahead moves right (always)
Ok	Stops the robot in case of always (always)

says. Here is a list with some of the feedback examples:

1. If the child uses the ”hello” command the robot will identify itself by saying ”Hello my name is SYD, nice to meet you”.
2. if the command ”start” is used to perform a programming loop, the robot will ask the child how many times to repeat the desired movement by saying ”Please say the number of times”.
3. After the specified number is said by the child, the robot replies ”Please say the direction”.
4. After saying the direction, the robot instantly starts executing and when it finishes, the robot tells the child to end the loop by saying ”Please end the loop”.
5. If the child does not say ”end”, in that case the robot will not execute or respond to any other commands until the ”end” command is said. This is used to teach children that any program has to have a beginning and an ending before moving on to the next command.
6. if the child says the command ”always”, the robot will reply with ”Please say the condition and direction”, the child then has to pick up a condition and a direction by stating ”forward (or backward) if there is an obstacle move right (or left), to stop the robot the child has to say the ”ok” command.
7. In the end the child also has to say the ”end” command to perform another movement or command.

3.2.1 The Maze Game

- **Level 1 - Sequential Programming:**
  At this level the child is asked to use certain commands like ”FORWARD, BACKWARD, LEFT, RIGHT” to make the robot move in any desired direction. The child then has to make the robot reach a certain place using these commands. The child is asked to direct the robot into a certain place using only those simple commands. This allows the child to learn the sequential programming in a more fun and easy way. The commands used in this level can be repeated for example: ”forward forward forward left backward left left right backward” ... etc.
- **Level 2 - Loops:**
  In this level the child is asked to do the same sequential steps but this time s/he has to use the loops instead of repeating the same movement command. The child can use these commands ”FORWARD, BACKWARD, LEFT, RIGHT, START, END”, for example: ”start”, ”3”, ”forward right”, ”end”, the robot will then perform 3 forward right steps. The idea of every program has to have a start and an end is introduced through the ”start” and the ”end” command in this level to perform a loop.
- **Level 3 - Sequential vs. Loops**
  In the third level, the task assigned is to program the robot to find its way out of the maze using the least amount of commands. In this case the child can use these commands ”FORWARD, BACKWARD, LEFT, RIGHT, START, END”. The child has to choose between the regular movements to get the robot out of the maze or the loops idea to do the same task.
- **Level 4 - Conditional Programming:**
  In this level the child is introduced to conditional programming. The child can use these commands ”ALWAYS, FORWARD IF OBSTACLE RIGHT, FORWARD IF OBSTACLE LEFT, BACKWARD IF OBSTACLE RIGHT, BACKWARD IF OBSTACLE LEFT”. The idea of this level is to make the robot move with no stops until the robot finds an obstacle in front of it, then it will take a step backward then changes direction to the right or to the left depending on the desired direction and condition. A running example would be: ”always”, ”forward if obstacle right”. Moreover if the child wants to make the robot stop the child has to say the ”OK” command. Furthermore if the child wants to perform another command then he has to say the ”END” command.

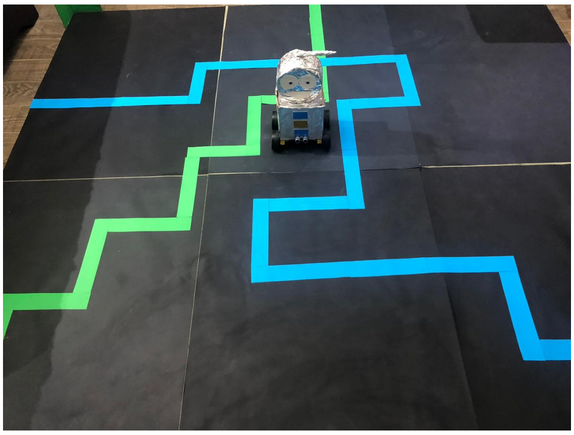

Figure 12: Maze game with the robot.

3.3 The Arabic Robot

The second robot is programmed by the children using Arabic voice commands as listed in figure 13. The voice command is transferred from the microphone to a python script created to transcribe the voice command to text. SpeechRecognition is the library used, it is a wrapper for many speech APIs. Google Web Speech API is the API implemented in the script. After the speech is transcribed to text, it goes through conditional statements and if it matches any of the commands available by the system, the command is sent to the Arduino that controls the robot through the NodeMCU. The NodeMCU receives data from the python script by implementing socket programming. Socket programming is a procedure of connecting two nodes on a network to communicate together, it maintains inter- process communication (IPC). The client socket is the socket that initiates the connection while the server socket is the one that listens on a specific port at an IP. Thus a socket connection is established between the NodeMCU and the python script. The NodeMCU hosts the server socket. A socket is created in the python script that connects to the port that the NodeMCU listens on and its IP address. The Arduino board receives the commands from the NodeMCU serially through the Tx and Rx pins and controls the movement of the robot and the electronic components accordingly.

3.3.1 The Maze Game

- **Level 1 - Sequential Programming:** Sequential programming is the concept to be gained from the first level. The children have to make the robot reach the end of the maze by following a certain path. The commands available in this level are: امام (forward), ورا (backward), يمين (right), شمال (left), وقف (stop). Any command can be repeated many times sequentially. This level starts by the robot stating the purpose of the game, the target of the level and the available commands to be used through the speaker module. Then, the child starts by giving a command to be executed one at a time by the robot until it gets out of the maze.

Voice Command	Functionality of the command
امام	The robot moves one step forward.
يمين	The robot turns to the right in its place.
شمال	The robot turns to the left in its place.
ورا	The robot moves one step backward.
وقف	The robot stops moving.
كرر	The robot repeats the action after this command a certain amount of times.
لو	Checks if the condition is true or false.
دائما	The robot keeps repeating the action that comes after this command forever.
في حيطة	Checks if there is a wall obstacle.
في مطب	Checks if there is a colored obstacle on the ground.
في نفق	Checks if there is a tunnel.
نور لمبة المطب	Turn the green led on.
نور لمبة النفق	Turn the red led on.

Figure 13: Arabic commands detected by the robot.

- **Level 2 - Loops:** The concept of loops is established in this level. The maze that needs to be solved is the same as in level one. The available commands are: امام (forward), ورا (backward), يمين (right), شمال (left), وقف (stop), كرر (repeat). A new command is accessible in this level which is كرر (repeat). This command allows the robot to repeat the command that comes after it the amount of times it was asked to repeat. This time the child has to program the robot to reach the end of the path using no more than five commands which will force him to use the new command. The path that will make the robot reach the end is a repeated pattern so the child can choose the right commands to repeat to solve the level. The level starts by the robot explaining the goal of the level, its available commands and the limit number of commands allowed to use through the speaker module then if the limit is reached and the maze is not solved, the robot states that the goal is not met, there are no more commands can be used and then asks the child to try again through the speaker module.
- **Level 3 - Conditional Programming:** In this level, the concept of conditions is introduced. The maze of this level includes three types

of obstacles: walls, ground colored obstacle, and a tunnel. Seven new commands are accessible in this level plus the commands of level one. The new commands are:

دائما (forever), لو (if), في حيطة (wall obstacle exists), في مطب (ground obctacle exists), في نفق (tunnel exists), نور لمبة المطب (turn ground obstacle LED), نور لمبة النفق (turn tunnel LED on).

Any command can be repeated more than one time. The دائما (forever) command allows the child to repeat the one command that comes after it forever, until reaching the desired position. In this level, more challenges are added to the maze to enforce the use of conditions. These obstacles are:

– The robot can not pass on a ground obstacle without turning the ground obstacle LED on.
– The robot can not pass under a tunnel without turning the tunnel LED on.
– If there is a wall, the robot cannot continue in its path unless it goes left or right

The walls and tunnel obstacles are dynamic; they can be placed anywhere on the maze. The robot declares issues to players by a buzzer followed by the robot stating the specific problem that needs to be solved through the speaker module in three cases:

1. If the robot is on a ground obstacle without the ground obstacle LED on.
2. If the robot is told to go forward and there is a wall at a certain proximity from it.
3. If the robot is in the tunnel without the tunnel LED on.

The ground obstacle is colored so that it can be detected by the robot using the color sensor when the frequency changes, the walls are detected by the ultrasonic sensor and the robot knows it is in the tunnel by the photo-resistor sensor when its readings change.

4 EXPERIMENTAL DESIGN

In this section, we are going to compare the effect of using an interactive robot to teach children computational thinking and different programming concepts to the traditional method of teaching. The purpose of this experiment is to examine if the use of an interactive robot in teaching programming concepts is more efficient than the traditional methods. This experiment aims to prove or reject the 3 null hypotheses. Generally, the 3 null hypotheses suggest that there are no differences between the efficiency of using the robot in the game and using the traditional methods of teaching programming to children.

1. The first hypothesis suggests that there is no difference in the learning gain between the two groups.
2. The second hypothesis states that the engagement level is not different in the two groups (Experimental and Control).
3. The third hypothesis claims that there is no difference in the system usability scale between the two approaches.

4.1 Focus Group

A focus group is a group that tests out the game before conducting the experimental group. The main purpose of making a focus group is to get feedback from children and their parents concerning improvements and/or modifications. A sample of 3 students within the target age range tried the English robot. They were asked to solve the first version of the pre-test and post-test used in calculating the learning gain. The learning gain test contained some cartoon mazes and the children are asked to write the steps to make the character reach a certain place. The children pointed out some important point that needed modification e.g. the question was not clear enough, some drawings were misleading. Accordingly, their feedback was taken into consideration and the tests were updated for the real experiments.

4.2 Experiment

A between-group design [1] has been used in this experiment. A between-group design is an experiment that has two or more groups of subjects each being tested by a different testing factor. It is basically dividing participants into two groups or more and testing each group with a different approach then compare the results.

Sample.

A sample of 36 children with a minimum age of 7 years old to 11 years old, has taken part in this experiment. All participants had no experience in writing a program (coding). The experiments were held to test the effect of both robots (the English and Arabic). Thus, the participants were divided into 2 groups; one

[1] https://en.wikipedia.org/wiki/Between-group_design

for testing each robot (English Group ,Arabic Group). The English group consisted of 20 children who were further divided into an experimental (12 children) and control (8 children) subgroups. The Arabic group consisted of 16 children who were further divided into an experimental (10 children) and control (6 children) subgroups. The experimental group has used the robot in the experiment while the control group has been tested using the paper traditional way. All students have been randomly selected for fair testing.

4.2.1 Procedure

Due to the current circumstances and the constraints created by the Ministry Of Education because of Covid-19 and the fact that there is a quarantine, internet video meetings have been scheduled with students individually. Thus, the experiment lasted longer than normal due to some technical issues. The experiment took about 2-3 hours to be completed for the experimental group. For the control group the experiment only lasted for 1 hour. First both groups take the learning gain pre-test and they have been asked to finish it within 20 minutes, if there is a delay in a question the students have been asked to pick the most comfortable answer for them. Then for the experimental group students have been asked to open the web API link on an android phone while opening the video conference on a laptop where they can see the robot and its corresponding actions. Basic movements and commands are explained and demonstrated to students, then they have started directly to engage with the robot and learning about the 3 programming concepts by playing the game. On the other hand, programming concepts have been explained to the control group using paper and basic examples. After both groups finish their experiment, they have been asked to take the learning gain post-test which is the same as the pre-test to be able to evaluate the progress they have made. Finally, both groups have been asked to take the engagement level test and the system usability scale to finish the experiment.

4.3 Evaluation

Three tests have been conducted in this experiment to compare between groups; learning gain, engagement level and system usability.

4.3.1 Learning Gain Test

The aim of this test is to assess the learning gain of children through measuring their improvement in computational knowledge, skill and development. The test is basically a rectangular maze that consisted of 25 blocks, it has a police car and a burglar. The test includes 3 levels that become harder as the child moves on to the next level, every level presents a different programming concept (Sequential, Iteration and Conditional). Movement commands are written on the maze to help children understand the available commands as some children have struggled to figure out the right or left rotations. The first 2 levels have the same questions but with different answers, as one of them tests the sequential type of programming while the other tests iteration programming. All questions of the test are multiple choice questions except for the last one, children are asked to type the right commands with no choices. All participants have taken the test before conducting the actual experiment whether it is trying out the robot or understanding the different programming concepts through the traditional way. Participants are then asked again to take the same test after finishing the experiment, to measure the amount of knowledge they have learnt from it. To measure the improvements children have, the pre-test and post-test are compared. Both the pre-test and the post-test have been the same to ensure that children have dealt with the same level of difficulty. An example of the questions is shown in Figure 14.

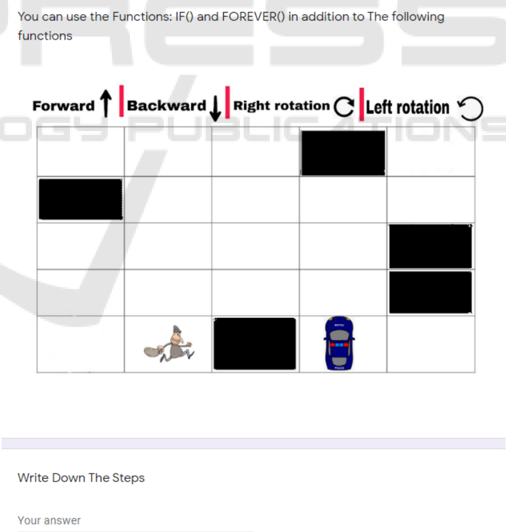

Figure 14: Learning gain test.

4.3.2 Engagement Level Test and System Usability Scale

After finishing the experiment along with the pre-test and post-test, participants have been asked to take the engagement level test and the system usability scale. The engagement level test is an 11 item questionnaire that determines the time that children can take to engage and interact in a certain activity [2]. The system usability scale provides a wider view of subjective assessment of usability, it is a simple, ten-item attitude Likert scale used to determine how easy a certain activity is [3]. Both of these tests have been given out to children at the end of the experiment and after taking the learning gain post-test. They have been used to analyse the overall experience and the feedback between the two groups (Experimental group and Control group). The child has to choose between Strongly agree, Agree, Disagree, Strongly disagree.

5 RESULTS AND DISCUSSION

The results of the learning gain test, engagement level test and system usability scale have been calculated and compared according to the learning approach of each group whether Experimental or Control. All the results of these tests have also been discussed and reported. Furthermore, an independent t-test has been used to analyse the data comparison between the tests of the two groups (Learning gain, Engagement level, System usability) by means of SPSS (Statistical Package for the Social Sciences) to show if there are significant differences between the learning gain, the system usability scale and the level of involvement between both the control group and the experimental group or not.

5.1 Learning Gain Test Results

The purpose of this test is to compare the knowledge gained by the experimental group which has interacted with the robot, to the knowledge gained by the control group which uses the traditional way of learning. The score of each participant is calculated by subtracting the score of the pre-test from the score of the post-test. The results of subtracting the learning gain tests (Pre-test and Post-test) show that the learning gain of the experimental group which has used the robot to learn the programming concepts, is significantly higher than the control group which has learned the programming concepts using the traditional way of explaining the concepts on a paper. Moreover the results of the independent t-test (shown in tables 2 and 3) stated that the learning gain of the experimental group is higher than the learning gain of the control group. The independent t-test results prove that the hypothesis stated in the experimental design part stating there is no difference in the learning gain between the experimental group (which used the robot) and the control group (which used the traditional way of teaching), is rejected.

Table 2: Mean and standard deviation results of the learning gain tests.

Group Name	N	Mean	Standard Deviation
Experimental A	12	2.91667	0.99621
Experimental B	10	3.70000	1.251666
Control A	8	1.37500	0.51755
Control B	6	1.33333	0.516398

Table 3: Independent t-test results of the learning gain.

Group	t	p	df
A	4.007	0.01	18
B	4.365	0.001	14

5.2 Engagement Level Test Results

The purpose of this test is to measure the level of engagement the participants have in the experiment while learning the different programming concepts. The results are then compared to see which group has a better engagement through an independent t-test as shown in tables 4 and 5. The experimental group's results (M = 3.43892 and SD = 0.304990) have proved to be significantly higher than the control group's (M = 2.02250 and SD = 0.080422) along with the t-test results (t = 12.737, $p \prec 0.05$, df = 18).

These results reject the null hypothesis stating there is no difference between the 2 groups concerning the level of engagement (using the robot or learning with the traditional teaching method).

Table 4: Mean and standard deviation results of the engagement level test.

Group Name	N	Mean	Standard Deviation
Experimental A	12	3.43892	0.304990
Experimental B	10	3.28140	0.539064
Control A	8	2.02250	0.080422
Control B	6	2.38250	0.384743

Table 5: Independent t-test results of the engagement level test.

Group	t	p	df
A	12.737	0.000	18
B	3.556	0.003	0.003

5.3 System Usability Scale Results

In this part of testing, the usability of the robot as well as the usability of the traditional method is eval-

[2]https://www.frontiersin.org/articles/10.3389/feduc.2018.00036/full

[3]https://en.wikipedia.org/wiki/System_usability_scale

uated. Participants have to answer 10 questions stating if the system is easy to use and learn or not. The results are then compared between the two different groups (Experimental and Control Groups) through an independent t-test as shown in tables 6 and 7. The Experimental group's results (M = 3.35000 and SD = 0.274690) have been significantly higher than the control group's results (M = 2.08750 and SD = 0.394380) along with the t-test results (t = 9.471, $p \prec 0.05$, df = 18).

These results reject the null hypothesis stating there is no difference between the 2 groups in terms of system usability scale (using the robot or using the traditional method of teaching).

Table 6: Mean and standard deviation results of the system usability scale.

Group Name	N	Mean	Standard Deviation
Experimental A	12	3.35000	0.274690
Experimental B	10	3.01000	0.372529
Control A	8	2.08750	0.394380
Control B	6	2.46667	0.344480

Table 7: Independent t-test results of the system usability scale.

Group	t	p	df
A	8.472	0.000	18
B	2.900	0.012	14

6 CONCLUSION

Throughout this work, we were trying to investigate if integrating robots in the educational process for teaching children coding principles has positive impact. For this purpose, we designed and assembled two interactive programmable robots through voice commands. The first robot is programmed through English commands, while the second one is programmed using Arabic commands. We tested the two robots with a group of 36 children through online meeting due to the Covid-19 and lockdown circumstances.

The results showed that integrating robots in the educational process has a great impact on children. The children's engagement with the robot in the learning process have given them an opportunity to enhance their knowledge more about computational thinking and programming which has facilitated the learning process and made the experience more fun for the children.

The main purpose of using an interactive robot in this work is to teach young children between 7-11 years old computational thinking and basic programming concepts. They have learnt three major programming concepts (sequential, conditional, iteration) by using a robot in a maze game that targets these concepts. The robot has a voice recognition feature that can easily be used by children to direct the robot out of the maze by using the three different programming approaches. This study proved that young children can understand complex programming concepts such as conditions and loops while mastering the sequential part easily. The experiment has tested 36 participants divided into two groups one with the English language and one with the Arabic language, and each group was divided into 2 sub-groups one of which has learnt the concepts by using the robot to test the game while the other has learnt the concepts by using the traditional methods by explaining the concepts on a paper. The results of the learning processes between the two groups have been compared regarding the learning gain, engagement level and system usability scale. To conclude, the group that has used the robot has significantly better learning gain and better engagement level than the group that has been taught by explaining the concepts on a paper.

The empirical results reported herein should be considered in the light of some limitations. The first limitation concerns the testing part. The pronunciation of words for some kids was not correct, as they speak English as a second language and some of them are too young to fully pronounce words the right way. Therefore, they have been asked to type the commands instead of saying them, which took them some time for typing correctly. A training period for correct pronunciation of the commands might be better for testing with the children before the real experiment.

However, in our opinion, although the English words might have been a challenge for some of the children, the results showed that both versions were educationally effective. Thus, the concern might have been that the English version would not be effective compared to the Arabic one which did not turn out to be the case. Thus, we believe that the system can also be used to teach the children some English vocabulary as a side effect. In the future, this should also be considered while testing the system to not only test the effectiveness of the system to teach computational concepts but to also check its effectiveness on the English language vocabulary.

In the future, maze game should be developed to be more challenging for children by adding more obstacles inside. In addition, more sensors can be added to the robot besides the ultrasonic one to expand the range of movements and challenges.

More programming concepts could also be added to the robot, as it only has one specific case of ”if” condition and no ”else” condition. More concepts to be added are nested loops and nested ifs.

REFERENCES

AcapelaGroup (2009). Acapela arabic speech recognizer and text-to-speech engine. *http://www.acapela-group.co*.

Badamasi, Y. A. (2014). The working principle of an arduino. In *2014 11th international conference on electronics, computer and computation (ICECCO)*, pages 1–4. IEEE.

Barai, S., Biswas, D., and Sau, B. (2017). Estimate distance measurement using nodemcu esp8266 based on rssi technique. In *2017 IEEE Conference on Antenna Measurements & Applications (CAMA)*, pages 170–173. IEEE.

Carnegie Mellon University. Alice. https://www.alice.org/. Last accessed 3 June 2018.

Constantinou, V. and Ioannou, A. (2020). Development of Computational Thinking Skills through Educational Robotics. *CEUR-WS.org*.

Denner, J., Campe, S., and Werner, L. (2019). Does computer game design and programming benefit children? a meta-synthesis of research. *ACM Transactions on Computing Education (TOCE)*, 19(3):1–35.

Edward, M., Karyono, K., and Meidia, H. (2017). Smart fridge design using nodemcu and home server based on raspberry pi 3. In *2017 4th International Conference on New Media Studies (CONMEDIA)*, pages 148–151. IEEE.

Fessakis, G., Gouli, E., and Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63:87–97.

Han, J., Jo, M., Jones, V., and Jo, J. H. (2008). Comparative Study on the Educational Use of Home Robots for Children. *Journal of Information Processing Systems*, Vol.4, No.4,.

Hsieh, Y., Su, M., Chen, S., Chen, G., and Lin, S. (2010). A robot-based learning companion for storytelling. In *Proceedings of the 18th international conference on computers in education*. Asia-Pacific Society for Computers in Education Putrajaya, Malaysia.

KALELIOĞLU, F. and Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners’ perspective. *Informatics in Education*, 13(1).

Kazimoglu, C., Kiernan, M., Bacon, L., and Mackinnon, L. (2012). A serious game for developing computational thinking and learning introductory computer programming. *Procedia-Social and Behavioural Sciences*, 47:1991–1999.

Kory, J. M. and Breazeal, C. (2014). Storytelling with robots: Learning companions for preschool children’s language development. In *The 23rd IEEE International Symposium on Robot and Human Interactive Communication, IEEE RO-MAN 2014, Edinburgh, UK, August 25-29, 2014*, pages 643–648. IEEE.

Magnenat, S., Riedo, F., Bonani, M., and Mondada, F. (2012). A programming workshop using the robot “thymio ii”: The effect on the understanding by children. In *2012 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO)*, pages 24–29. IEEE.

MIT Media Lab. Scratch. https://scratch.mit.edu/. Last accessed 1 June 2018.

Mohamed, K., Dorgham, Y., and Sharaf, N. (2021). Kodockly: Using a tangible robotic kit for teaching programming. In Csapó, B. and Uhomoibhi, J., editors, *Proceedings of the 13th International Conference on Computer Supported Education, CSEDU 2021, Online Streaming, April 23-25, 2021, Volume 1*, pages 137–147. SCITEPRESS.

Riek, L. D., Mavridis, N., Antali, S., Darmaki, N., Ahmed, Z., Al-Neyadi, M., and Alketheri, A. (2010). Ibn sina steps out: Exploring arabic attitudes toward humanoid robots. In *Proceedings of the 2nd international symposium on new frontiers in human–robot interaction, AISB, Leicester*, volume 1.

Robins, B., Dautenhahn, K., te Boekhorst, I. R. J. A., and Billard, A. (2005). Robotic assistants in therapy and education of children with autism: can a small humanoid robot help encourage social interaction skills? *Univers. Access Inf. Soc.*, 4(2):105–120.

Rogalla, O., Ehrenmann, M., Zollner, R., Becher, R., and Dillmann, R. (2002). Using gesture and speech control for commanding a robot assistant. In *Proceedings. 11th IEEE International Workshop on Robot and Human Interactive Communication*, pages 454–459.

Vaucelle, C. and Jehan, T. (2002). Dolltalk: a computational toy to enhance children’s creativity. In Terveen, L. G. and Wixon, D. R., editors, *Extended abstracts of the 2002 Conference on Human Factors in Computing Systems, CHI 2002, Minneapolis, Minnesota, USA, April 20-25, 2002*, pages 776–777. ACM.

Wu, E. H.-K., Wu, H. C.-Y., Chiang, Y.-K., Hsieh, Y.-C., Chiu, J.-C., and Peng, K.-R. (2008). A context aware interactive robot educational platform. In *2008 Second IEEE International Conference on Digital Game and Intelligent Toy Enhanced Learning*, pages 205–206.

Wyeth, P. and Purchase, H. C. (2002). Tangible programming elements for young children. In Terveen, L. G. and Wixon, D. R., editors, *Extended abstracts of the 2002 Conference on Human Factors in Computing Systems, CHI 2002, Minneapolis, Minnesota, USA, April 20-25, 2002*, pages 774–775. ACM.