ASHESI UNIVERSITY COLLEGE

INTRODUCING PROGRAMMING SKILLS IN CHILDREN THROUGH TOYS (ROBOT) TO ENHANCE THEIR COGNITIVE SKILLS AND CRITICAL THINKING

[APPLIED PROJECT]

B.Sc. [Computer Science]

Lynn Lutubah Mumia

2018

ASHESI UNIVERSITY COLLEGE

Introducing Programming Skills in Children through Toys (Robot) to Enhance Their Cognitive Skills and Critical Thinking

[APPLIED PROJECT]

[Applied Project] submitted to the Department of Computer Science, Ashesi University College in partial fulfilment of the requirements for the award of Bachelor of Science degree in [Computer Science]

Lynn Lutubah Mumia

June 2018

Declaration

I hereby declare that this [Applied Project] is the result of my own original work and that no part of it has been presented for another degree in this university or elsewhere.

Candidate's Signature:

……..…………………………………………………………………………………..

Candidate's Name:

...…..………...……………………………………………………………………

Date:

...…..…………………………………………………………………………………..

I hereby declare that preparation and presentation of this [Applied Project] were supervised in accordance with the guidelines on supervision of [Applied Project] laid down by Ashesi University College.

Supervisor's Signature:

……...…...…………………………………………………………………………

Supervisor's Name:

…...…....……………………………………………………………………………..

Date:

...……....……………………………………………………………………………….

Acknowledgements

My heartfelt appreciation goes to Jehovah for the strength and gift of good health throughout the process of this project and my education at Ashesi.

I would also like to pass my gratitude to my parents Mr and Mrs Mumia for the unwavering support and guidance they granted me throughout my school years at Ashesi. I am grateful to The MasterCard Foundation for their belief in Dr Patrick Awuah's dream and funding my education here at Ashesi University College. It is as a result of this world class education that my thinking was challenged, eyes opened and a grandiose dream of making a positive impact in the society triggered. Ashesi has served as the root to a vast sense of responsibility in me to detect and solve societal problems through innovative and diversified ways.

I am forever indebted to my supervisor, Dr Elena Rosca, for her constant guidance, support and being a shoulder to lean on in times of distress through this process which has yielded a satisfactory outcome. On the equal measure, my thanks go to Dr Nathan Amanquah for the constructive advice, feedback and support he provided throughout my dissertation process.

Finally, I would like to thank my closest friends; Ruth Karimi, Linda Bhebhe, Mercy Asante and Mohamed Hijazi for not giving up on me and continuously keeping me in check to help me achieve my goals. Additionally, I would like to thank them for sharpening my grammar and writing skills and bringing the best in me.

Abstract

As decade after decade passes, the world has proved extremely progressive especially in technology. As the world goes completely digital and most people become computer savvy, it is pivotal to have programming skills to avoid becoming redundant. In creating a world where programming skills is a culture, it is necessary to instil such skills right from the start with children. Therefore, teaching programming in schools is just as necessary as other core subjects. Nevertheless, in many schools in Africa, programming is a skill that many meet while in university if and only if the school in which they attend has a curriculum that forces them to do so otherwise it remains a gap in the education.

Programming is core in developing critical thinking, computational thinking, and problem-solving skills among children. Besides programming being taught, it is ideal that the method used to impart knowledge is fun, easier and the best in aiding understanding. This project proposes the use of the Safari (robot) developed using Arduino and Dual Tone Multiple Frequency (DTMF) to teach programming concepts to children using a phone as a controller to move the robot in four main directions; right, left, forward, backwards and repetitively. This will be useful both in schools and homes to teach young children programming concepts for its ease of use and enhancing fun.

Key Words: Critical thinking, computational thinking, programming, problem solving

Table of Contents

Declaration i

Acknowledgements ii

Abstract iii

Table of Contents iv

List of Figures vi

Chapter 1

Introduction and Literature Review 1

1.1 Research evidence / back up 2

1.2 Motivation 4

1.3 Literature Review and Related Works 5

Chapter 2

Requirements Analysis...10

2.1 Security Requirements 10

2.2 Functional Requirements 11

2.3 Non-Functional Requirements 12

2.4 User Requirements 13

Chapter 3

High Level Architecture...15

3.1 Bluetooth Control..15

3.2 DTMF wired control through phone..16

3.3 GSM control through phone...18

Chapter 4

Implementation 20

Chapter 5

Testing and Results 25

5.1 Unit Testing..25

5.2 Proposed Use cases 26

5.3 Testing and Results 27

Chapter 6

Recommendation and Conclusion 28

References..30

List of Figures

Figure 2.10. Chassis image..14

Figure 3.10. The basic architecture of the toy to teach programming. 15

Figure 3.11. This is the Bluetooth architecture use case...16

Figure 3.12. The DTMF architecture (The phone with keys represents the phone connected to the DTMF module which is pressed by a child. User is the child and DTMF and Arduino board are on the chassis)..17

Figure 3.13. This is the DTMF architecture use case..17

Figure 3.14. The GSM architecture (Phone is with the buttons is the phone used by the child. Connected to the chassis through cloud represented by the star in the diagram)....18

Figure 3.15. The GSM architecture use case..19

Figure 4.1. An Arduino board (Arduino Uno Rev3, 2018) 20

Figure 4.13. DTMF board (AliExpress, 2010) 21

Figure 4.12. DC Motor and the wheel (Plastic Tire Wheel With DC 3-6v Gear Motor For Arduino Smart Car, 2006)..21

Figure 4.13. Chassis assembling 22

Figure 4.14. Complete chassis with power bank and DTMF board..................................22

Figure 4.15. Front View of the chassis 22

Figure 4.16. Rear view of the chassis 23

Figure 4.1. Side view of the chassis..23

Chapter 1: Introduction and Literature review

As society is digital in almost every aspect from browsing the internet, driving, sending emails, remote control and home automations, it's important that the people are able to use these devices as well as be fluent in them (Quansah, 2014). With computing as a mundane part of life, it is essential that everyone is included in the drastic progressive wave of technology. However, for this inclusivity to be replicated in third world countries, skills in programming and computing must be made a part of the educational curriculum.

As part of the curriculum, acquisition of knowledge and skills in programming and computing by everyone will be a feasible consideration in achieving complete fluency. In effect, this smoothness in computing would result in better decision making when it comes to technologies to be used and their intended functionalities. Problem solving in the society will also be pioneered. Nevertheless, such expertise in technology use and manipulation can only be effected through programming skills. This valuable skill can be imparted through one major way, that is, education which is proficient in knowledge acquisition.

Firstly, programming is one of the skills necessary in manipulating and instructing computers to perform both simple and complex tasks. Computers can range from phones, tablets, PCs to desktops. Know-how in operating such to achieve a user's intent can also be referred to as programming. Evolution in technology aforementioned, emphasizes the urgency in acquiring programming skills. This is substantiated by an article written by Lydia which highlights the paradigm shift in businesses relying on computer code for their activities resulting to an opening of 7 million jobs in 2015, which is growing 12% faster than the market average according to a report from Burning Glass (Dishman, 2016). A keen look at this figure indicates the increasing demand in employees with programming skills which may further increase. Statistical evidence of this demand lays emphasis on introduction of programming among young children who will turn out to be the beneficiaries of such amassed opportunities.

Additionally, it has been observed over time that, the acquirement of programming skills by individuals has been fundamental to the development of problem solving, logical reasoning and critical thinking skills. As these three skills are core to our world at present, it is pivotal that they are presented and instilled into young children at an early age rather than when they are grown. Early development of such skills will enhance the inculcation of problem solving skills that are essential for solving societal problems innovatively. As a result, grasping of such skills by young ones will in turn ease dependence on governmental and foreign assistance in solving problems. This upcoming generation will be moved to solve societal problems inventively and ask for minimal assistance from governments and foreign aid which is healthier than sole dependence.

1.1 Research Evidence / Back Up

Instilling programming skills in young ones is crucial as proved by prior research. Based on sample research that will be discussed in relation to this paper, the extent of how critical programming is among young children is clearly brought out.

To begin with, in a study conducted by Kalelioglu and Gulbahar, the effects of teaching programming on problem solving skills among primary school students especially those of grade 5 is highlighted and emphasized. Subsequent to their study, an insightful conclusion is made; teaching programming skills indeed plays a role in enhancing problem solving and computational thinking skills since there was a significant increase in the problem solving ability among the sampled students used for this study (Kalelioglu & Gulbahar , 2014). For that reason, imparting both programming and computational skills from a tender age is significant in equipping them with skills which are handy for their future problem-solving ability. With the invaluable worth of these skills (programming and computation), this paper will be discussing one of the proposed ways in parlaying knowledge in programming, critical thinking and cognitive skills among children of a tender age.

Additionally, most educational institutions have found it compelling to introduce compulsory programming courses as part of their curriculum. These courses are to furnish their students with the problem solving and critical thinking skills which most of them lack. For instance, Ashesi University College is a model of such schools and has that kind of curriculum. In the first year, all students are required to take an introduction to computing and programming course. This course involves the teaching of a programming language with the inculcation of critical thinking and problem solving. As part of the course it is also required that students identify a societal problem and solve that problem using the programming skills just developed. Such model of schools highlight the importance of these skills by making it a priority to infuse them in their newly enrolled bunch of students.

Moreover, subsequent to introduction of programming courses in schools, and the rise in the benefits of programming skills, it is important to think about the commencement of teaching these skills at an early age. Since at the moment the focus on such skills (programming and critical thinking) is mostly in colleges just as demonstrated by a model of schools like Ashesi University College, it is the goal of this project to find a way of introducing such skills among young children. As it is significant to develop the problem solving and cognitive skills among children early, programming can be an easy way of doing so. Just as substantiated by a study conducted by Pea and Kurland, programming increases the mental power of students as it has been seen among pre-high school level children in USA (the study's sample space) where it is taught mainly for this purpose (Pea & Kurland, 1984).

1.2 Motivation

Motivated by an open source project called cubetto developed with a goal of teaching children programming at an early age, the main objective of this research is to localize this project to Africa. The cubetto is a robot that is used as a toy by children. It can be dismantled and arranged in a particular order determined by a colour change of the buttons on a board. The different colours determine the movement of the doll-like structure that comes along with it depending on the preference of the one using the toy. If the cards are arranged in a particular order for a combination of a particular colour, the doll will move left, right or take steps forwards (Woollaston, 2013). At some points the movements have to be done iteratively which symbolizes the looping concept in programming. Although this is a fun game, it still develops a child's programming skills by giving them an idea of what programming is through the movements done and commands they give the doll, which are programming concepts.

Similar to cubetto, the toy idea that would be localized will be a play board game with a couple of safari games (wild animals). The game will involve the arrangement and movement of animals around obstacles with a limited number of moves: left, right, front and back. This game will also be used as an evaluator. With the game, teachers and parents will be able to determine the children's thinking capability. This is because the toy will be incorporated with a timer that will determine how long a particular child takes to complete the challenge. The time concept will also be used to evaluate how patient a child can be since in programming a lot of patience is needed for the process of debugging to be successful. All these skills intended to be imparted will augment problem solving, computational and cognitive skills in children while at the same time allowing them to have fun. This method will be less cumbersome than the archetypical ways of teaching programming in class which sometimes have proved to be boring and unexciting among students. Also, safari will be appropriate for use for the target age group of the children.

1.3 Literature Review and Related Works

Related works in any research oriented project is vital. Not only is it informative on past research done on the topic but also evaluates, summarizes and correlates various ideas from different sources directly related to the current research. Likewise, the literature review and related works explored in this project gave a wider perspective on preceding implementation and identification of gaps in previous research. This was majorly informative on the design decisions and choice of materials for the proposed pedagogical method of teaching programming. Therefore, this section of the paper highlights prior research explored in this area which was carried out in this field of academia. Also, this section sheds more light on challenges and gains obtained from the studies which influenced the implementation plan of this project.

Firstly, an article written by Guizo, *These Robots will teach kids programming,* talks about a robot Yana and Bo which play hide and seek. This is a common and exciting game played mostly among children. These robots are attractive both in colour and appearance even though their sole purpose is to teach programming. Apart from the colour and shape, another key attractive feature of these robots is that they talk to the target children, mainly 5 years old which makes it interactive. The robots implement a Bluetooth 4.0 technology which uses an iPad or other tablets which enhance engaging and intuitive ways of learning programming (Guizo, 2013). Although this technology is aimed to teach coding in an easy and practical way, it uses expensive technology and hence it is costly to buy the robots. This means that it is only limited to those who can afford such as it works in cooperation with the iPad which is relatively expensive for an average household in Sub-Saharan Africa. This hinders a wider possibility of audience who are the main target.

Secondly, in a study conducted by Kalelioglu & Gulbahar, their objective was to find out how programming skills are divulged to students and in turn impacts their problem solving skills. The article highlighted that the motivation behind teaching of such skills among students was the vast lack of problem solving and computational skills which are needed in our era of information. In this article, it was explained that the method used to teach programming greatly affects the kind of affection students have towards it. Therefore, when one is choosing a method in which to convey these skills they should make sure it does not turn out to be a stumbling block in their learning process. In their study, skills like creative thinking, systematic reasoning and collaborative work were developed through Scratch the main tool of their study. Additionally, there was an integration of visualized programming learning which enhanced development of cognitive and affective skills in children. Although the method used passed on some skills, it had no impact on problem solving skills. The researchers however advised that use of high-order activities to be solved by programming would develop problem solving skills, yet no tangible suggestions of such activities were given. This could thus be an area of future exploration by other researchers.

Moreover, the criticality of the methods chosen in teaching programing cannot be accentuated enough. In a study conducted by Ibrahim Ouahbi, Fatiha Kaddari, Hassane Darhmaoui, Abdelrhani Elachqar and Soufiane Lahmine, creation of games in Scratch Programming Environment is focused to evaluate the resulting students' motivation towards programming. The findings of this study reveal that the use of Scratch (A free programming language and online community for creating interactive stories, games and animations) which is fun and easy to use, highly motivated students and empowered them. 63% of students who had an experience with Scratch considered continuity in their studies in programming (Ouahbi, Kaddari, Darhmaoui, Elachqar, & Lahmine, 2015). Nonetheless, majority of those who used a standard programming environment lost interest in programming. This was attributed to poor teaching methodologies, low interaction with students in class and a lack of interest mainly influenced by overwhelming and tedious theoretical concepts and techniques involved (Ouahbi, Kaddari, Darhmaoui, Elachqar, & Lahmine, 2015). Thus, use of fun methodologies in teaching could result in many students being drawn towards programming and the educational system in general.

Furthermore, Pea and Kurland, sought to prove that learning programming really develops cognitive skills and is not just a popular notion that it does develop cognitive skills when it reality it doesn't. In their research, they analysed the effects of programming on the learning abilities of a child as defined by cognitive science. As per this definition, they would identify any impact or improvement in the learning ability of a child. For instance, in the sense of computer programming, the ability of a child to easily navigate and explore new programming constructs show cognitive skills with focus to the specific knowledge domain developed (Pea & Kurland, 1984). So to measure in cognitive skills of the students, they used available cultural interpretive systems of both social and education interaction. Additionally, in programming, the idea of artificial intelligence and construction of such models as a way of understanding the human cognitive behaviour has led to better development of cognitive skills. Nevertheless, the paper does not explain how exactly the increase in cognitive skills is measured and one could wonder exactly how such a skill is developed. There is no tangible method to measure development of cognitive skills and hence makes it difficult to measure the exact impact of the inculcation of programming skills.

A multi-national, multi-institutional study of assessment of programming skills of first-year CS students, a study made by McCracken, Almstrum, Diaz ,Guzdial, Hagan, Kolikant and Wilusz, focusses on a way of assessing the programming skills of students, specifically those in first year in college. The partakers of the research gave methods they used in the assessment; take-home programming assignments, examinations and charrettes which trigger the students to work on their own and think critically (McCracken, et al., 2001). They also used specific methodologies in the problem solving quest. The methods are; abstracting of problems from their description, generating sub-problems, transforming sub-problems into sub-solutions, and recomposing sub-solutions into code, and then evaluate and loop through the solution (McCracken, et al., 2001). Critical review of these methodologies show that they could be adopted and improved upon so as to measure the progress in programming skills developed among children. Although the suggestions are good, according to the paper it did not yield good results as the student sample did worse than expectations and hence it could be risky to adopt this methodology unless improved upon or modified.

Lastly, programming by itself requires additional skills like patience when it comes to coding. As it is widely said, 'debugging is twice as hard as writing the code itself' (Kernighan, 2012). As a result, in teaching programming, such skills should be imparted into children since patience is needed in debugging and solving problems. This is demonstrated by an article written by O'Dell, The Debugging Mindset, *Debugging*. In this paper O'Dell describes debugging as problem solving since in a way one has to think through the errors occurred and try to solve them (O'Dell, 2017). However, the ability to solve such problems depends on the mindset of an individual. People with an incremental or growth mindset tend to solve problems easily compared to those with a fixed one (O'Dell, 2017). This mindset plays a pivotal role in one's ability to solve problems. As a result, while teaching programming, striving to develop a growth mindset among children would enhance their development of patience. With patience, debugging could be easily instilled in children although it was not described explicitly how that could be done.

Chapter 2: Requirements Analysis

For a successful design, building and deployment of a functional and usable system, the requirements of the software have to be given much attention. Similarly, the implementation of the suggested toy to teach programming to children is not much different from any other system. In gathering information for the requirements, **online reviews of surveys**, the use of **online standards for toys, literature reviews and available data** on toy products and **expert background knowledge for software developments** in cooperation **with interviews** from the intended users (parents, teachers and children) was used. The requirements discovered could be grouped into **security requirements**, **user requirements**, **functional** and **non-functional requirements**. These requirements will be discussed in detail in this section.

2.1 Security Requirements

Security is one of the major aspects that is to be considered in making toys especially for kids. The safety of the kids should be considered as well as the fulfilment that would come with the game. Some of the main requirements to be considered are:

2.1.1. The toy's power/electricity consumption

If the toy uses electricity, the working of the shock absorbers is key. This is to prevent damage or injuries to its users. The amount of the power in watts should not be too great such that it could result to electric shocks if directly connected to power while in use.

2.1.2. The Materials used

The materials used to make the toy should not be toxic. If metallic, they should not be poisonous if ingested by the child so as not to hinder growth and intellectual development (Toys & Chidren's Products, 2017). Also, the materials should not be flammable, hence the products should be made to enhance fire resistance to avoid injuries among children. The materials should not have sharp edges as well to avoid injury. In this project, some of the materials considered are wood, plastic and rubber. These are cheaply available and not injurious. The available parts of the toy are assembled into a chassis. None of the final chassis look has sharp edges and hence it is safe for children's use.

2.1.3. Privacy

The toy will be used to collect information about patience and thinking capability of the children and hence it should not be easily exposed. Data should be kept private for the major targets who are parents and teachers who will use it to help in the development of children.

2.2 Functional Requirements

Functional requirements are those that define the functions of the system and its components. Some of the functional requirements of the system are:

2.2.1. Teach Programming

The movement of the games is within the constraints of left, right, forward or backward. The movements could however be done iteratively to allow longer distance movements in one direction. With these four movement constraints, the games are to be manipulated with a combination of them until the obstacles are avoided and the target destination achieved.

2.2.2. Allow the Safaris (games) movement on the play board

The whole purpose of the game revolves around the movement of the games around the obstacles on the play board to reach the final destination. Therefore, the board's design should allow easy movement of the games on the basis of the directions allowed.

2.2.3. Hinder movement of barriers

The purpose of the barriers is to make it a bit challenging to move the games around the board. If the barriers can easily be moved about, the whole purpose of the barriers will not be achieved. Hence, the barriers should be immovable.

2.2.4. Time recording by the timer

The timer on the toy should be able to record the time used by the child to solve the specific challenge at hand. The timer should also record the child's name and the time used in its memory for later retrieval by the parent or teacher. This feature could also serve as a challenge for the player improving in speed to beat the previous times recorded.

2.3 Non-Functional Requirements

Non-functional requirements are not be explicitly stated by the user but have to be considered for an efficient system. Some of the non-functional requirements to be considered in the implementation of this system are:

2.3.1 Performance

The **performance** of the game should be efficient. The games/safaris should be fast in responding to the movement commands issued to them.

2.3.2 Availability

The time information should always be **available** to teachers and parents whenever it is needed.

2.3.3 Maintainability

Maintainability. The system should be easy to maintain in case of damages or improvements of the functionalities added to it. Thus the code needs to be readable and understandable by other programmers.

2.3.4 Adaptability

Adaptability. The system should be easily adaptable among different stakeholders. If the toy uses power, it should be adaptable in such a way that in less power prone areas it can be used in a conservative mode so as to save power. A solar panel can be adopted to supply power in such cases.

2.4 User Requirements

User requirements are those specified by the user according to what they would want the system to have. Some of the user requirements gathered by the interviews are as listed below:

2.4.1 Ease of use

Ease of use. The system should be easy to use and intuitive. It should not require any additional explanations and information for the kids to be able to explore and play the game.

2.4.2 Appearance

The appearance of the toy should be attractive. Since the main target is children, the only way to catch their attention to the game is by using attractive colours and interesting materials. For instance, the tyres are black with a yellow middle part. Yellow is a striking and noticeable colour and hence will be attractive to children. Moreover, the chassis' bottom is brown. This blend of colours forms a good pattern. See image below:

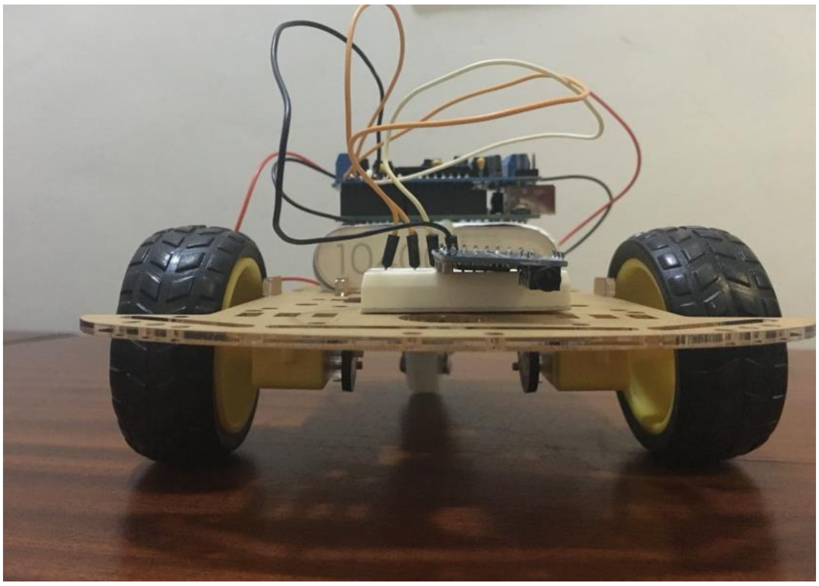

Figure 2.10. Chassis image

Chapter 3: High Level Architecture

The architecture of the system was determined through three iterations before the last two were finally decided on. Below is a detailed description of the three stages. The last two architectures on the list are the ones which were focused on for implementation.

3.1 Bluetooth Control

In this model of the architecture, the toy is divided into the play board and the toy which is referred to as Safari. The play board has different colour coded buttons which are used to control the robot. Safari is coded to move in four directions, which are left, right, forward and backwards. Some of these directions can however be repeated by multiple pressing of keys. As described above, the board has a timer which records the time the child takes to solve the movement of the safari around the task assigned to him or her.

The diagram below shows the pictorial view of the architecture:

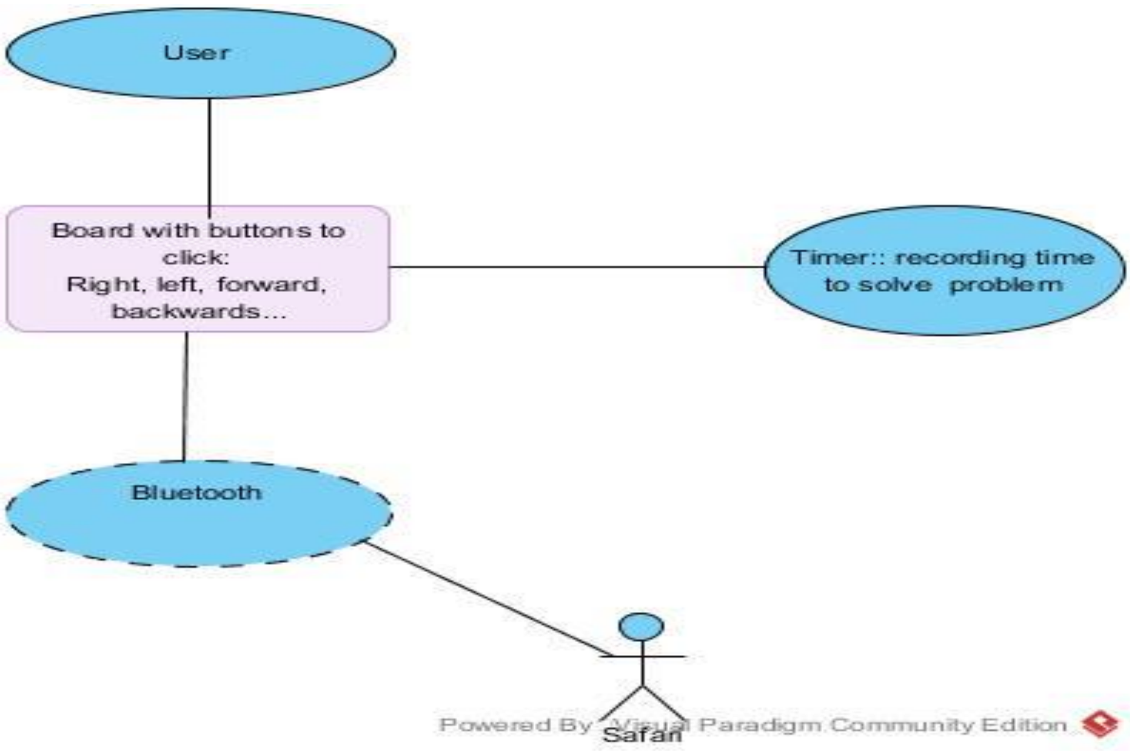

Figure 3.10. The basic architecture of the toy to teach programming.

Using this architecture, one use case for the toy was considered. The use case diagram is as below:

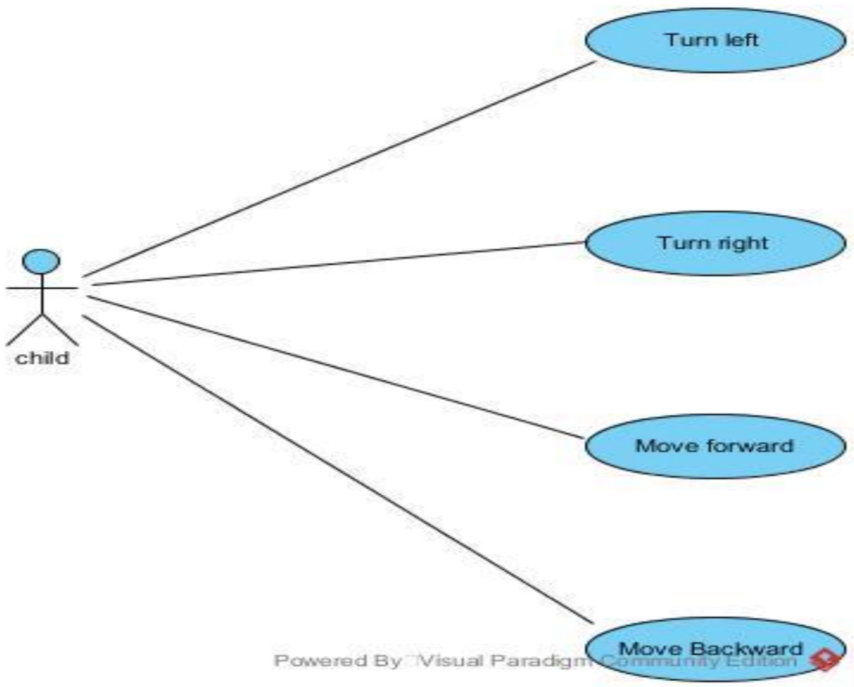

Figure 3.11. This is the Bluetooth architecture use case.

3.2 DTMF Wired Control through Phone

In this model of the architecture, the toy is a robot chassis which is controlled with a phone. The different keys of the phone are coded to represent the different directions of movement of the chassis. The keys represent left, right, forward, backward and repeat for each of the directions. If a non-coded key is pressed nothing happens to the chassis. The phone is connected to the chassis by a wire or cable. This is enabled by the DTMF board which has a port for connecting to the phone through the An Auxiliary Port (AUX). AUX is used for earphones and headsets. In this project, it is used to control the robot.

The diagram below shows the pictorial view of the architecture:

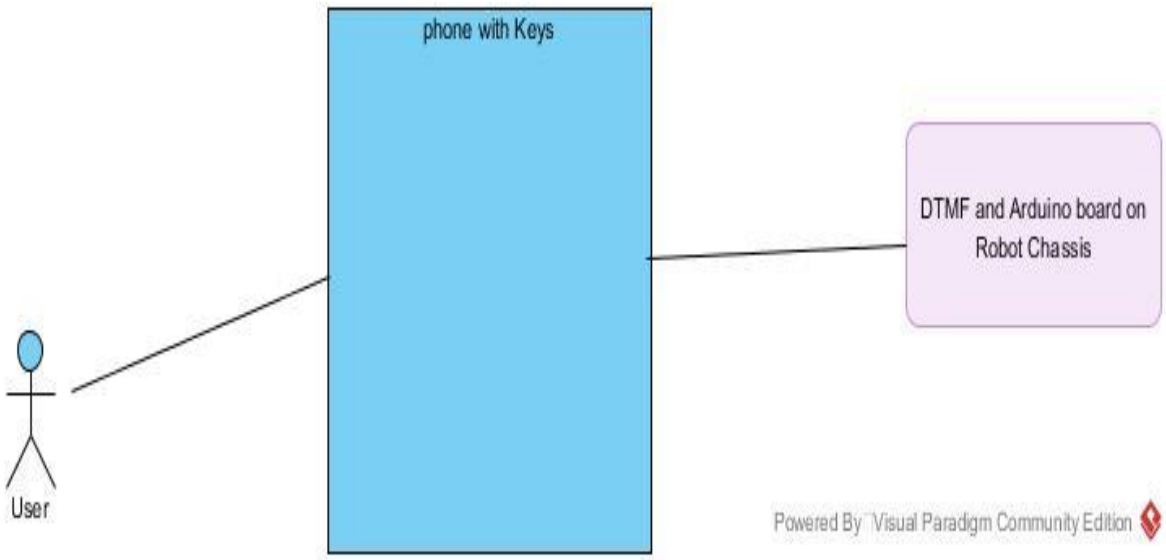

Figure 3.12. The DTMF architecture (The phone with keys represents the phone connected to the DTMF module which is pressed by a child. User is the child and DTMF and Arduino board are on the chassis)

Using this architecture, one use case for the toy was considered. The use case diagram is as below:

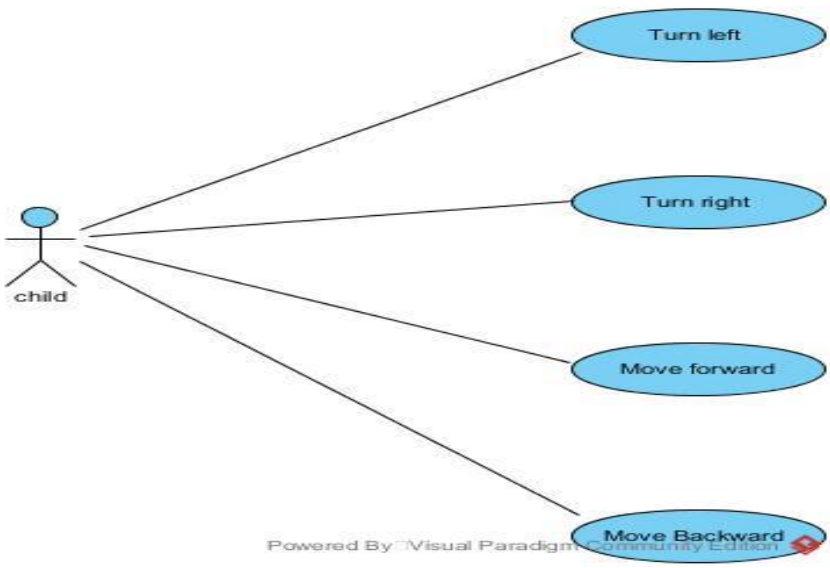

Figure 3.13. This is the DTMF architecture use case

3.3 GSM Control through Phone

In this model of the architecture, the toy is a robot chassis which is controlled with a phone. The different keys of the phone are coded to represent the different directions of movement of the chassis. The keys represent left, right, forward, backward and repeat for each of the directions. The directions of the buttons are clearly indicated on the phone being used to control the chassis. This makes it easier for the child to identify what to press compared to the previous architecture.

The phone is connected to the chassis as its controller through GSM which uses the internet for connection. This architecture is the best model but has challenges when it comes to areas with low internet connectivity. Since this does not require a cable to connect, it is limitless in the kind of distance the chassis can move.

The diagram below shows the pictorial view of the architecture:

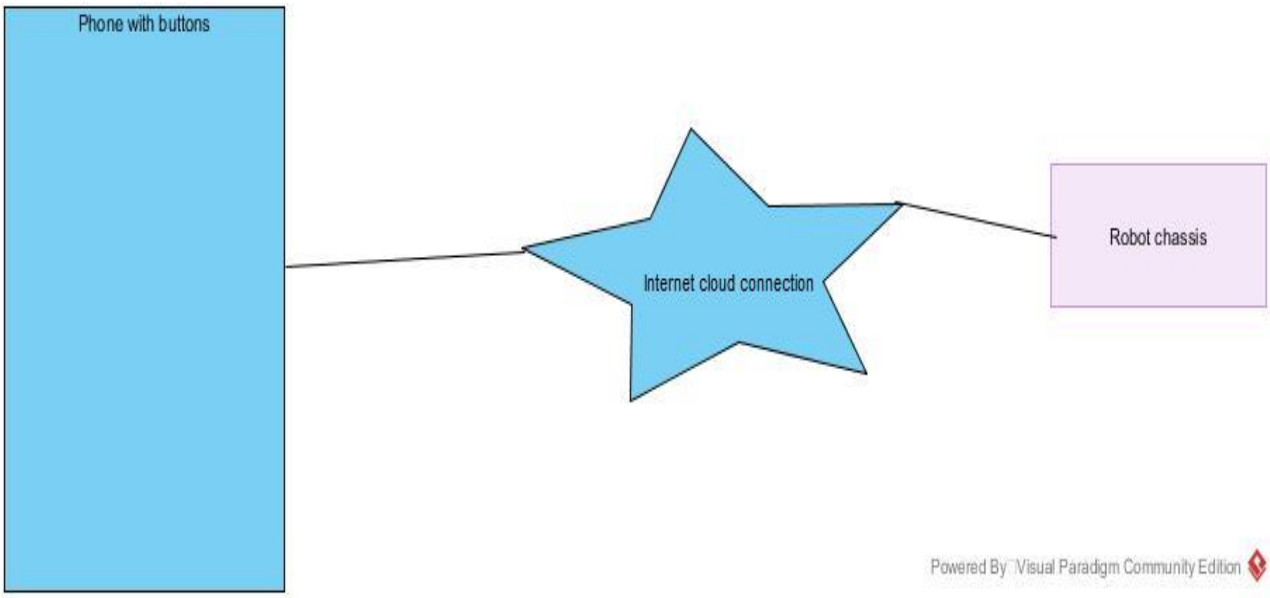

Figure 3.14. The GSM architecture (Phone is with the buttons is the phone used by the child. Connected to the chassis through cloud represented by the star in the diagram)

Using this architecture, one use case for the toy was considered. The use case diagram is as below:

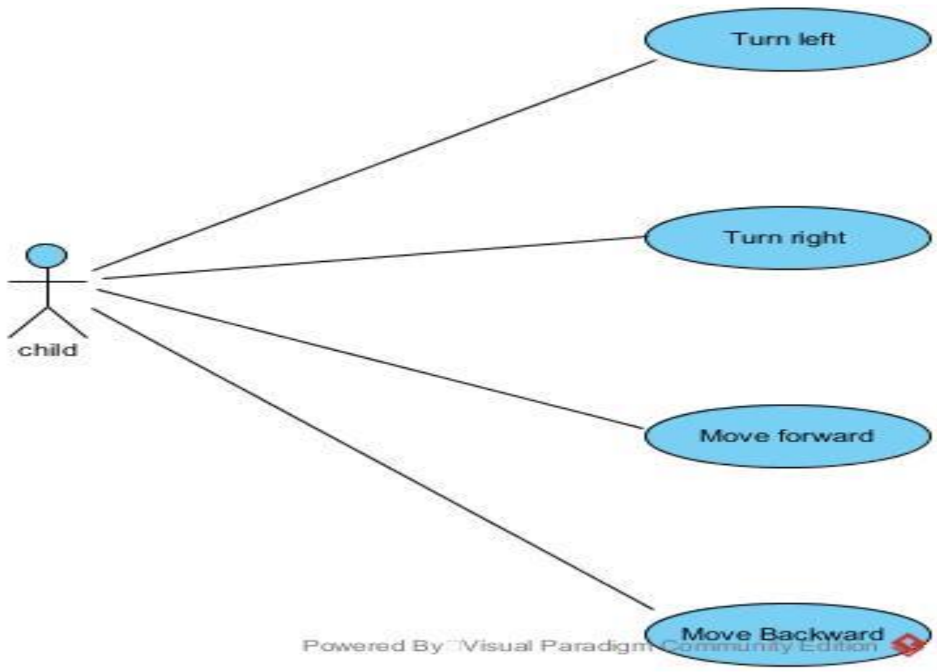

Figure 3.15. The GSM architecture use case

Chapter 4: Implementation

The fourth chapter discusses how the safari toy was developed, by use of which architecture, on which platforms and how it will be used to impart the necessary skills intended.

The robot chassis (Safari) was assembled using boards and DC motors responsible for movements of the chassis. The Arduino Uno board was attached to the chassis together with the DTMF board and the DC motors controller board. An Arduino Uno board is a micro controller board based on ATmega328P with 14 digital input and output pins, 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button (Arduino Uno Rev3, 2018). The use of an Arduino board is facilitated by an open-source electronic platform grounded by its easy to use hardware and software. Below is an image of an Arduino board.

Figure 4.1. An Arduino board (Arduino Uno Rev3, 2018)

Moreover, a Dual Tone Multiple Frequency (DTMF) board is a self-contained board which allows its users to control remotely via radio or audio producing source, the on or off state of the devices or relays (DTMF decoder products, 2004). Additionally, a DC motor is a motor which uses direct current and it is attached to a tyre. The motor is manipulated to move according to the coded directions which subsequently turn the tyres to the specified direction.

Figure 4.12. DC Motor and the wheel (Plastic Tire Wheel With DC 3-6v Gear Motor For Arduino Smart Car, 2006)

Figure 4.13. DTMF board (AliExpress, 2010)

The DC Motors are powered by a battery for locomotion. These dry cells are attached to the chassis and connected to the board using cables. The power from the cells is in turn transmitted to the motors connected to the board using a cable. In this project, the cells are rechargeable. The cells used are extracted from a power-bank. This is easily rechargeable and long lasting due to their large power levels. A power-bank was the best option to use in this project since the chassis needs power for its constant movement based on the activities given to the children. Its rechargeable ability aids in the reduction of cost which could otherwise be incurred in the periodical replacement of dry cells used in powering the chassis.

The images below show different stages in assembling the robot chassis until its complete form:

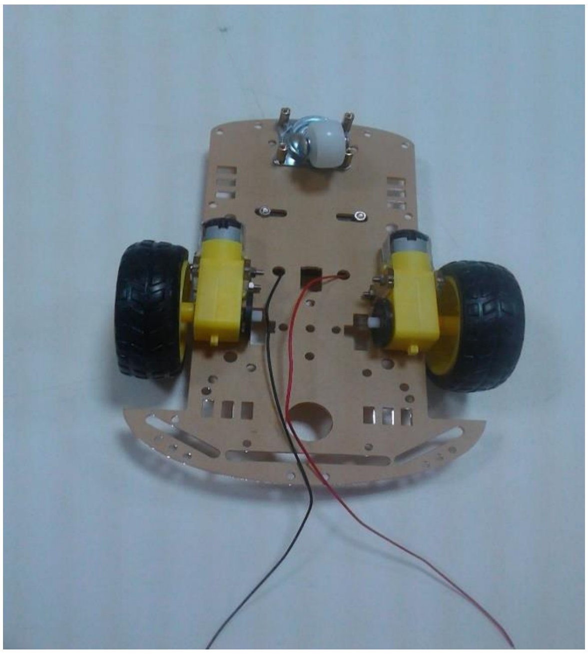

Figure 4.13. Chassis assembling

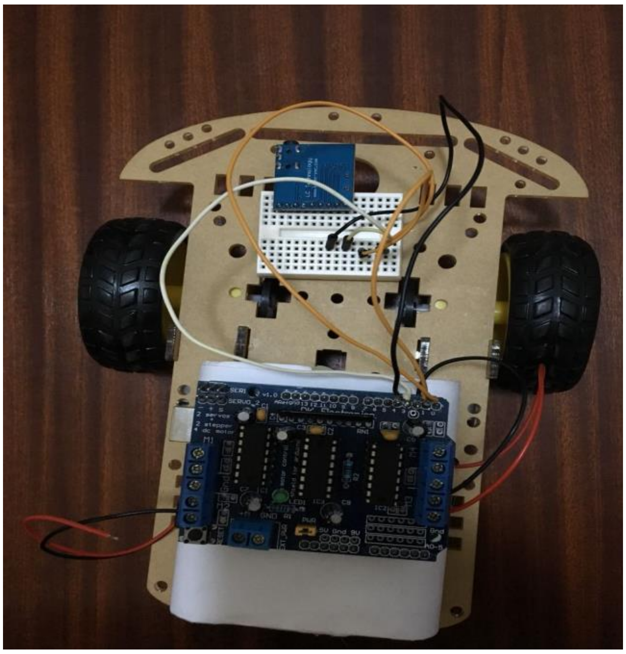

Figure 4.14. Complete chassis with power bank and DTMF board

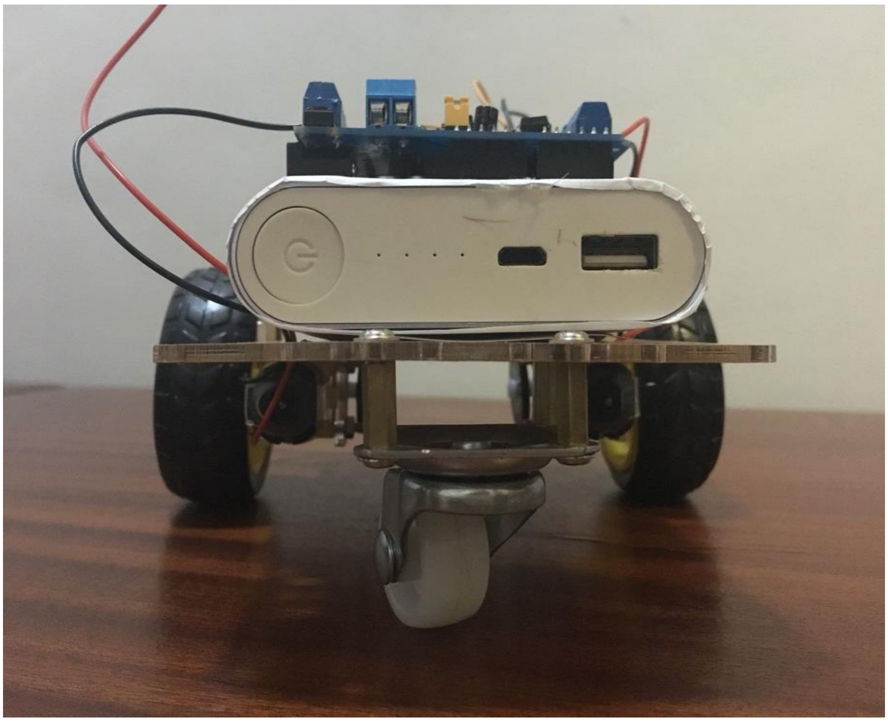

Figure 4.15. Front View of the chassis

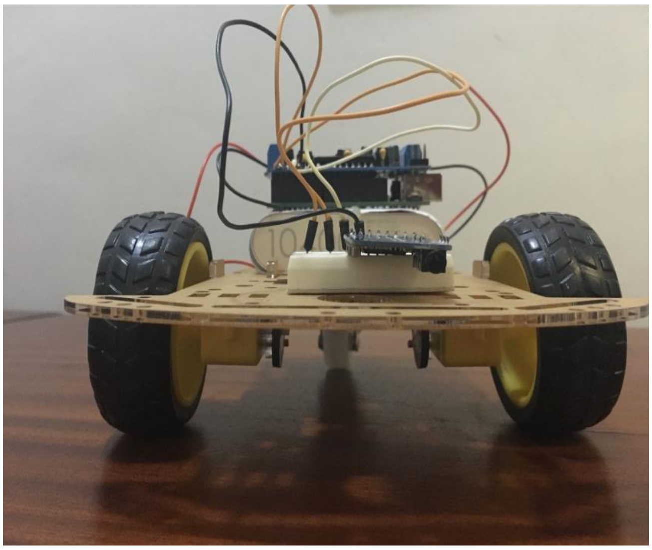

Figure 4.16. Rear view of the chassis

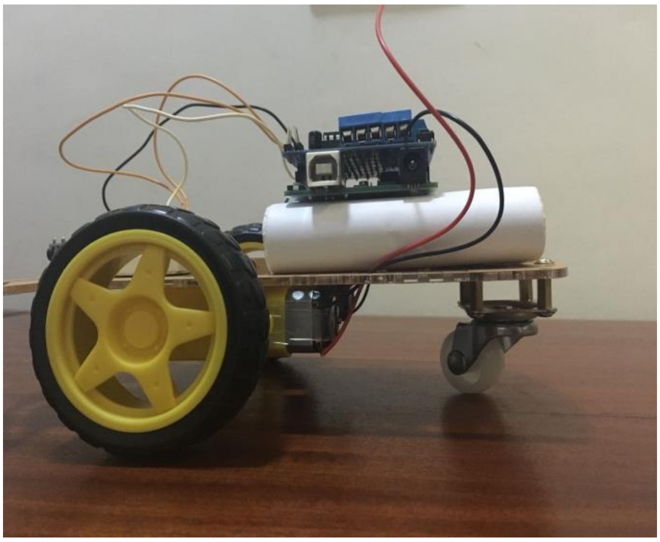

Figure 4.1. Side view of the chassis

The chassis movement is coded using the Arduino programming language for hardware. This language is complemented with C++ which makes it easy to code for an individual who is familiar with such programming languages. Through the open source platform, several libraries and their documentation are readily available and hence information can be accessed at any needed time. The Arduino code is loaded to the Arduino Uno board for deployment. The code is written using the Arduino Integrated Development Environment (IDE). This IDE also allows users to add new libraries they need for specific projects and to load it into the Arduino board and hence testing is done as building is in progress.

The movements of the chassis were coded using the Motor Shield Library. This library is responsible for the setting up of the speed of the Motors and assigning which specific direction they should move in. Also, the code to control the keys and buttons on the phone is loaded through the IDE. Any type of phone can be used as long as it has an AUX port for audio related purposes as described earlier in the architecture section or it can connect to the internet that is for the GSM architecture.

Safari works in a way that the child is given the phone with the meanings of the keys on the phone. Furthermore, a challenge is given to them, for example navigating a maze. The task the child has is to figure out which combination of keys will enable them solve the challenge at hand. A challenge can also be in the form of a specific path with obstacles which has to be passed through without collision with the obstacles. In a scenario where the chassis senses an obstacle, the child is notified of no path in the direction in which the safari is moving. This is to allow them to choose a different direction in which to move the chassis.

Chapter 5: Testing and Results

This chapter explains the process of testing safari based on two main use cases and the proposed subsequent steps. Even though three different architectures were developed fully for use during implementation, inadequate hardware tools limited its implementation to the DTMF module architecture only. The testing intended involving participants from chosen schools in Ghana, but due to poor response of the contacted schools this approach was abandoned. Therefore, testing involved unit testing of the functionalities of the robot and the proposed use cases for testing meant for the participants.

5.1 Unit Testing

The robot chassis had two main functionalities, movement to the left, right, forward and backward, and mobile control of these movements.

5.1.1 Movements

The completion of the code to model the movements of safari called for testing. With an already powered Arduino, the cable was connected to the laptop and the code loaded to the board. Afterwards, the robot was set in a spacious room where there were no obstacles to distract its movement. Since the change in direction of the movement was fixed to change after a specific time-frame, the robot was allowed to navigate freely. After approximately twenty minutes of watching the movements, it was satisfactory that the robot moved as intended. The first stage of testing each unit of the direction of movement was completed successfully and therefore the toy was set for the next stage of coding and testing.

5.1.2 Mobile control of movements

The prior code was modified to include the mobile keys for controlling the robot. Completion of the code meant testing. Therefore, using the phone connected to the DTMF on the chassis using an AUX cable, a specific key meant for a particular movement was pressed and the safari set in an open room. Safari was observed to be moving in the specified direction based on the pressed key triggers in all the cases inclusive of the repetitive ones. The working of the robot was a go ahead that the chassis worked based on the DTMF architecture.

5.2 Proposed Use Cases

5.2.1 Navigating a maze task

Using a maze drawn on the floor in a confined room, a child is given the phone, told the meaning of the different keys of the phone and then told to move the robot around the maze until they make it out. In this test, the child has to first figure out a path to use to the end of the maze before navigating the robot. Finding the path involves coming up with a step by step instructions (algorithm) and finding an efficient way of moving the robot without colliding with obstacles. This process involves critical thinking and computational thinking for the game to be completed successfully. In this form, the major programming concepts are set and imparted to the child.

5.2.2 Navigating a specified drawn path with obstacles

Similar to navigating a maze, a specific path with obstacles is drawn on the ground and the child given safari and the controller phone, told the meaning of the different keys of the phone and then told to move the robot along the path until the end. The process of figuring out which set of instructions to give to the robot so as to pass through the path successfully involves computational thinking and critical thinking. In some cases, it may involve repetitive tasks which teach looping in programming. This is important for repetitive tasks even in solving real world problems.

5.3 Testing and Results

Initially, testing was to be done among 10 Ghanaian International schools. The sample schools were contacted but only one person from the 10 schools contacted responded. On follow up discussions, the respondent failed to communicate resulting in a gap in the continuity of the conversation to the testing point. Limited by the time constraint of the testing results, this could not be achieved. Due to no sample of testing group, the testing of safari using the participants was not successful. The system therefore remains dependent on its workability based on the unit testing and component testing of the functionalities.

Hence, any propositions made with regards to its usability is solely based on unit and component testing. Any other conclusions made are based on intuition and projections of the user's behaviour which in this case is the child. Additionally, review of other related tests on similar products influence the conclusions and recommendations that will be made. Lack of implementation of other architectures of Safari also resulted to the limited options of testing. This was due to limited hardware and time constraint to consider all the three architectures.

Chapter 6: Conclusion and Recommendation

Basic and easy games can be very impactful if they are made practical and interesting to children. This will help children in picking up programming easily and serve as a rock-bound foundation for them. The practicality of the lessons makes it easier to identify and notice the instantaneous effects of their work. This direct feedback results in appreciation of the capabilities of programming.

The challenges of the use cases described above in this paper is a first-step for children into their programming journey. Controlling safari to navigate the maze or to follow the designated path is a problem solving task that kids can relate to or identify with. Although the game is challenging, it is also entertaining. The simplicity of safari's direction to just four basic mundane activities, left, right, forward and backward will make it easy for the children to use.

Besides waiting on the testing results of Safari which has been proposed to teach programming, it is also pivotal to think about improvements that could be made on it so as to be beneficial for children at higher levels of education or with an advanced age group. Likewise, it is advisable to determine what age group is most appropriate for use of programming languages in teaching children. Furthermore, future developments can consider using the Bluetooth or GSM architecture in implementation for use with Smartphones to increase the scope of movement of the toy. The GSM architecture will also be advantageous in that one can control Safari without necessarily being in close distance to it. Safari can also be improved upon to allow teaching of other subject concepts inclusive of programming. This will be an easy access for learning to children who lack the necessary tools and facilities for education.

Essentially, controlling Safari to solve challenges like navigating a maze using simple and easy to understand commands, can provide a solid foundation for teaching young children programming concepts at an elementary level.

References

(2010). Retrieved April 9, 2018, from AliExpress: https://www.aliexpress.com/item/1Pcs-MT8870-DTMF-Audio-Decoder-Speech-Decoding-Board-Voice-Module-For-Arduino-With-Pins-LED-Indicators/32806254666.html

Arduino Uno Rev3. (2018, March 26). Retrieved from Arduino: https://store.arduino.cc/usa/arduino-uno-rev3

Dishman, L. (2016, June 14). *Why Coding Is Still The Most Important Job Skill Of The Future*. Retrieved April 17, 2018, from FastCompany: https://www.fastcompany.com/3060883/why-coding-is-the-job-skill-of-the-future-for-everyone

DTMF decoder products. (2004). Retrieved March 26, 2018, from Intuitive Circuits, LLC: http://www.icircuits.com/prod_dtmf8.html

Guizo, E. (2013, October 30). *These Robots will Teach kids Programming skills.* Retrieved from IEE Spectrum: https://spectrum.ieee.org/automaton/robotics/home-robots/play-i-bo-yana-robots

Kalelioglu, F., & Gulbahar , Y. (2014). The effects of Teaching Programming via Scratch on Problem Solving Skills: A Discussion from Learners' Perspective . *Information in technology, Vol 13. 1*, 33-50.

Kernighan, B. (2012, December 11). *Kernighan's lever*. Retrieved from linusakesson.net: http://www.linusakesson.net/programming/kernighans-lever/

McCracken, M., Almstrum, V., Diaz , D., Guzdial, M., Hagan, D., Kolikant, B.-D. Y., . . . Wilusz, T. (2001, December 4). A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin, 33*(4), 125-180. doi:10.1145/572139.572181

O'Dell, D. H. (2017, March 22). The Debugging Mindset. *Debugging, 15*(1). Retrieved from http://queue.acm.org/detail.cfm?id=3068754

Ouahbi, I., Kaddari, F., Darhmaoui, H., Elachqar, A., & Lahmine, S. (2015). Learning Basic Programming Concepts By Creating Games With. *Procedia- Social and Behavioural Sciences, 191*, 1479-1482. Retrieved April 16, 2018, from https://www.sciencedirect.com/science/article/pii/S1877042815024842

Pea , R. D., & Kurland, M. D. (1984). On the Cognitive Effects of learnign Computer Programming . *New Ideas Psychol Vol 2 No. 3*, 137-168.

Plastic Tire Wheel With DC 3-6v Gear Motor For Arduino Smart Car. (2006). Retrieved April 9, 2018, from Bang good.com: https://www.banggood.com/Plastic-Tire-Wheel-With-DC-3-6v-Gear-Motor-For-Arduino-Smart-Car-p-997113.html?cur_warehouse=CN

Quansah, M. F. (2014). Developing a mobile based application for feature phones to teach children in rural area how to code. Accra, Eastern Region, Ghana. Retrieved March 9, 2018, from https://air.ashesi.edu.gh/bitstream/handle/20.500.11988/240/Michael_Quansah_AppliedProject_2014.pdf?sequence=1&isAllowed=y

Toys & Chidren's Products. (2017). Retrieved from C &K Testing: http://en.cirs-ck.com/services/toys?gclid=EAIaIQobChMIrJa-qZec1wIVQZR-Ch0DHQxVEAAYASAAEgIZYPD_BwE

Woollaston, V. (2013, December 10). *The gadget that teaches FOUR-year-olds how to program a computer - but costs £135 and parents need to build it themselves.* Retrieved from mail online: http://www.dailymail.co.uk/sciencetech/article-2520909/The-Primo-Cubetto-teaches-FOUR-year-olds-program-computer.html