CHRISTIAN SERVICE UNIVERSITY COLLEGE

FACULTY OF HEALTH AND APPLIED SCIENCES

DEPARTMENT OF COMPUTER SCIENCE

USING SCRATCH TO TEACH PUPILS COMPUTER PROGRAMMING.

BY

AMPOMAH YAW

FERDINAND ACKAH

JEPHTER KWAME OPOKU

PATRICK OWUSU ANSAH

DENNIS NANA BOAKYE AGYEMANG DUAH

A THESIS SUBMITTED TO THE DEPARTMENT OF COMPUTER SCIENCE/INFORMATION TECHNOLOGY, CHRISTIAN SERVICE UNIVERSITY COLLEGE IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF BSC. COMPUTER SCIENCE FACULTY OF HEALTH AND APPLIED SCIENCES

DECLARATION

We hereby declare that this submission is our own work towards the BSc and that, to the best of our knowledge, it contains no material previously published by another person nor material which has been accepted for the award of any other degree of the University, except where due acknowledgment has been made in the text.

Name	**Index Number**	**Signature**	**Date**
YAW AMPOMAH	10003374	.................../	...../...../......
FERDINAND ACKAH	10003169	..................../	...../..../........
JEPHTER KWAME OPOKU	10003047	..................../	..../...../........
PATRICK OWUSU ANSAH	10002227	..../	..../...../.........
DENNIS NANA BOAKYE AGYEMANG DUAH	10004384	....................../	..../...../..........

Certified by:

MRS. AFIA BRUCE Signed: ………………….

(Supervisor) Date: …………………......

Certified by:

MRS. LINDA AMOAKO BANING Signed: ………….............

(Head of Department) Date: ……………………...

SUPERVISOR’S DECLARATION

I hereby declare that the preparation and presentation of the dissertation was supervised in accordance with the dissertation policy of the Christian Service University College.

MRS. AFIA BRUCE Signed: ………………….

(Supervisor) Date: …………………......

MRS. LINDA AMOAKO BANING Signed: ………….........

(Head of Department) Date: ……………………….

DEDICATION

We dedicate this work to our Supervisor, Mrs. Afia Bruce, the defense Panelists and lastly our parents, colleagues and friends who helped us throughout our first degree and the reseach work.

ACKNOWLEDGEMENT

It is a great pleasure to thank the many people that supported us during our studie, especially the Almighty God who has played a major role from the beginning to the end of our first degree academic life. We appreciate the one and only God for the good health and wellbeing that he made available to complete this book.

Unreserved thanks goes to our project supervisor Mrs.Afia Bruce for her constant and precious time anytime we consulted her meek guidance throughout the rsearch work.We appreciate the the time spent on discussing and accurately reviewing all the reseaech work we did. She consistedntly allowed this paper to be our own work and steered us in the right direction whenever she thought we needed it. We would also like to thank the experts who were involved in the validation survey for this research project: Mr. Atiso, Mr. Adinkrah, and Mr. Charles. Without their passionate participation and input, the validation of this project could not have been successfully conducted.

We want to thank the Computer Science department, in particular, the lecturers; we treat all of you as special persons. Our regards also extends to our fellow course mates for their feedback, cooperation and of course, their friendship.

We also thank our family for the unceasing encouragement, support, and attention. We are so grateful to our parents who supported us throughout this venture.

We also place on record, our sense of gratitude to one and all, who directly or indirectly, have lent their hand in this venture.

ABSTRACT

The researchers' main focus was to find a solution to problems identified. We found and suggested that computer programming should be taught from the primary or elementary school level to aid computational thinking, logical reasoning, and problem solving skills in order to prepare them cognitively and inculcating in them the desire to pursue a carrier in computer science programme or programming course. It imparts students with some basic knowledge before they get to the tertiary institution. This basic programming concept when handled proprerly, can reduce the difficulty rate and the drop out level in the computer science department at the university. Introducing programming or coding to pupils at the primary school level is a good idea and is becoming universal but the method or approach in delivering computer programming is quite difficult for the pupils to grasp. Therefore, there would be a need to suggest a simple learnable visual block-based programming environment. The researchers supported the use of Scratch programming environment as one of the best methods known to introduce programming to pupils. This research focuses on an important matter: how to teach pupils basic concepts of computer programming using one of the commonly best-used programming environment and a new approach.The introduction of this approach and environment is easy and also effective in improving computational, logical reasoning, and problem solving skills. The research shows that pupils were able to learn to use the scratch environment and also created projects for simple storytelling and animation on their own.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background

A collected instruction given to a computer to implement, build and execute certain tasks through a process sequentially is called Computer Programming.

The instructions are accepted to be computer programs to help the computer function smoothly. The instructions or source code are either written in human language form, then translated to machine language-binary form (0’S and 1’S) before the computer can carry out a particular task. The instructions are line of codes that should be free of bugs for smooth execution.

Throughout all educational levels, computer programming is seen as important proficiency in creative thinking and solving problems. Therefore, it has been considered valuable.

The world today is rapidly increasing in technology and most of our daily activities are based on technology. The Ghana Education Service found that there is the need to introduce Information Communication Technology (I CT) to the basic and high schools.

Information Communication Technology was defined as electronic devices for collecting, processing, storing, retrieving and communicating information by Gunton (1993), it has been understood by most of the pupils by definition; practically, majority of students in Ghana today have little understanding in computer known as "computer literacy" alone. They know how to work on MS words, use spreadsheets and others but cannot develop one of this software that can be used by others. Computer programming commonly known as ‘’coding’’ is vastly gaining a valuable presence in primary schools outside this country, to be precise, outside the African continent. Stressing on how important coding is to the growth and development of a nation and its economy, the Government of Ghana in collaboration with the Ghana Education Service has agreed to make programming part of school's curriculum to be taught to pupils starting from Kindergarten one (1) and upwards effective this 2019/2020 academic year. We belief this new initiative can help reduce the difficulty faced by computer science students when learning programming at the tertiary level.

Based on the experience of the authors, predecessors, and colleague, programming has become a major challenge for most of the computer science students and has put fear and panic when they hear of coding in their courses. With this, adding basic programming to basic schools' curriculum will ensure that pupils at the Basic level practically will do something with the computer. In this view students will become conversant with the use of a computer and feel at ease doing practical things on the computer. Since there is a rapid growth of technology each year, it may be a stepping stone for the future.

We now ask, how can basic programming be made effective in the Ghanaian basic schools with Information Communication Technology inclusive and in which way can it be promoted?

The study involves detecting the cause(s) of the problem. Computer Science is aimed at providing theoretical training as well as providing practical skills to students at the completion of the program.

Our findings target the solution of making teaching and learning basic programming effective and a core and integral part of the Ghana Education Service curriculum for pupils at the elementary level.

1.2 Problem Statement

- Learning programming is considered hard for students in the tertiary institutions.
- Programming courses in universities usually have highest dropout rates and even some students change their computer science program to offer another that they find to be easy or little bit easy. Also programming courses are often regarded as most difficult.
- Programming involves writing codes to solve problems but knowing how to code has become a major issue for the Ghanaian tertiary students.

1.3 AIM

This study is to enhance student's capacity of coding so that Ghana as a nation would be competitive in this new age of rapid technological advancement around the globe.

To achieve this, we make a full analysis of the benefits of introducing computer programming as a subject of study to the basic school curricular of the Ghana Educational Service system.

1.4 Objectives Of This Study

- To impart into children creative and logical reasoning ability, encourage autonomous learning and social interaction, and to develop their problem-solving skills.
- Early introduction of pupils to programming concepts so that they can be confident enough to make computer science as a career without fear of difficulty.

- To teach coding to pupils through educational games and storytelling to enable pupils grasp programming concepts quickly and easily at a younger age since children can learn easily through playing games and having fun.
- The approaches to better sustainability of the introduction of the programming concepts at the upper primary level.

1.5 Specific Objective

The objective is how to implement basic programming in the B S C P (Basic School Computerization Policy) and the grounds for policy and decision makers as well as Civil Society Organizations to advocacy and addressing the gaps that may exist.

1.6 Project Organization

This finding is into five chapters.

The first chapter ‘’introduction’’ provides an overview of the research work conducted, states the problems to be addressed and summarizes the objectives and then the justification of this project.

❖ The succeeding section deals with

2. Literature review

❖ The third section deals with

3. Methodology

❖ . The penultimate section deals with

4. Results and Discussion

❖ The ultimate section deals with

5. Conclusion and Recommendations.

1.7 Research Questions

- What is the existing system?
- What impact does it have on students?
- What are the problems of the existing system?
- How will the new system complement existing system to enhance students' capacity in solving computational problems and thinking logically?

1.8 Justification

Information Communication Technology is purposely training people to use computer for example teaching students to use Microsoft application platform such as spreadsheet, word, notepad and others.

This study teaches the basic level what computer science, information technology and digital literacy is about. Also, teaching them coding, programming on their own and not how to work a computer, but how it works and the way you want it to function.

The study will give the students creative ideas and encourage them to become excellent software developers.

It helps them to express and make logical assumption at the Basic level when there is meaning to the background of what is happening.

"If you teach computing and do it right, you can help children develop their learning in literacy and numeracy," Bill Mitchell an educationist cited studying algorithms can help their understanding of sentence structure when some children used the Scratch programming language to make animations for their creative writing and suggested that''.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

In US and UK computer science education has expanded tremendously in the basic and secondary level. Today, pupils learning computer science at a younger age will have many benefits. The reason being that:

Since the world is advancing into technology, most of them will acquire job or there will be an increase of job creation in the future. It seems to be promising job opportunities in some nearby future.

At their or this stage, it will expose them to different ideas and picking topics that will make much sense when they advance to adulthood.

Computational literacy is a platform where people can express and change their ideas to simple and creative instructions to solve everyday problem.

There were arguments about coding and the stage at which it should be taught. Out of this Duncan et al (2014) discussed several issues that must subsidize the decision which involved educators who have sufficient knowledge, proficiency in the discipline and that the tool to work with must be in place. At the basic level, students are below the developing stage but can fathom certain elementary programming ideas.

Storytelling is one of the ideas and it will be used as a foundation to help improve the main ideas of how to understand, studying how to narrate their intellectual growth and to support the idea of elementary programming at their stage.

To conclude, we reflected on three correlated matters:

- Finding the means to integrate or embed coding into primary school pupils' curricular.
- The way and means to fuse playful objects into coding to relate a story.
- Setting ultimate coding space for storytelling to help students at the basic level.

2.2 Children's' Ability to Understand

Every student has a degree of mental reasoning as maturity takes place. Piaget & Inhelder (1969) developed such a theory called cognitive development.

The beginning of a child's growth start with sensor motor stage then continues to the pre-operational intellectual and actual functioning stages and at the age of 11, graduate to the official functioning intellectual stage.

Lister (2011), recorded Piaget phases to programming skills and stated "full programming ability is not present until the final stage of development", the official functioning stage. Lister applied the stages to the Neo- pagetian theory that talks about the individuals going through the stages of development several intervals irrespective of their age as they come across fresh challenges. It is possible for kids of age 11 and above to develop or learn how to code or program but not in the case of those below that age due to the process they have to go though. Rader , Brand, and Lewis (1997) during their test such as picture matching subroutines and others while using Cocoa Language added that Kids between 2 to 3 and 4 to 5 must be allowed in programming. At the kindergarten and lower primary level, they realized that most of them could not finish any self-activities but at the upper primary level, they implicitly executed their wok except for properties.

(Morgado, 2005) contemplated on some systems such as Tortis which was created in 1974 – 1976 by Radia Perlman with the concept of the Logo project.

Others were the Micro World JR and Cocoa which also came as a result of the Logo and they lack some characteristics to assist storytelling. To dive into Storytelling, its importance will be first dealt with.

2.3 Importance of Storytelling

Years ago as a child storytelling was what convinced us. Anything that was put into story form we would pay much attention to that without hesitation. Willingham (2004) also have proven that "People pay more attention to stories than other text and are better able to remember information in the form of a story" which is an integral part and it serves as an entertainment which one is able to remember each time; "Children enjoy stories from a very early age, and storytelling as a form of instruction, has many benefits". It serves as a powerful tool which is now been used in coding or programming and other project researches. Kelleher and Pausch (2007) used "storytelling to motivate girls to learn coding in the Alice environment". Howland and Good (2015) also "explored the use of storytelling and natural language in programming in their work with Never winter Nights and the FLIP language". Some research works have been focused on little children and have used storytelling as a driving force for computer programming.

2.4 Storytelling Programming Environments for Kids

(Flannery et al 2013; Resnik et al 2009) respectively in a close look up of "Scratch Jr." adapted the "blocks-based Scratch programming environment" to the kindergartens which solely dealt with story creation. It was designed to offer a setting with a "low floor and a high ceiling" which would have involved pupil between 2 and 5 years "but still support creative solutions as more complex languages do".

Programming in "Scratch Jr.", as in "Scratch", "involves connecting blocks together to form sequential commands". "Blocks' labels are icons, to enable pre-literate users to interact with the system. Most commands involve the manipulation of one of several on-screen 2D sprites, and nearly all have immediately visible effects."

"These sprites represent the characters children use to tell their stories. Stories programs can also be shared between students," "and Scratch Jr." offers "classroom support for teachers to facilitate multiple students using the language simultaneously."

Kahn (1996) came up with "ToonTalk" which was an earlier example of a system designed for kids. "ToonTalk provides a constraint-based system and presents itself like an open-ended Lego sandbox." Kids "program agents by adding blocks, all represented by icons instead of text, to a work space and programming by demonstration." more as a game where users can manage a town and create their own games to share with friends.

"Magic Forest" is another icon-based programming environment that allows users to edit or create new games by placing "stones" (commands) inside of "scrolls" "that are attached to objects (Andrade, 2007)." "The scrolls dictate the object's behavior." Like "ToonTalk," "Magic Forest presents itself as a game making and playing environment more than a storytelling platform." "Storytelling is an integral Part of game-making, however, and tools to build games can also be used to craft stories." "The role of animated graphics in storytelling environments is important both as a means of semantic feedback to children and as a source of humor and richness." "This could be seen even within the relatively primitive environment of Spinnaker Software's Story Machine (InfoWorld 1983)." "With more modern graphical affordances such as those described by Pahud (2016), one can expect Children's motivation to be stronger." "Storytelling has been used by these systems primarily to contextualize and motivate programming actions. From a young child's perspective, storytelling and programming may be quite similar tasks."

"There are separate theories on childhood narrative development, however, and thus there may be something to be gained from using these theories to re-examine how we present programming tasks to children."

2.5 Cognitive Narrative Development" and how it can be related to Programming

Blackwell defines programming in broad terms in "What is Programming?'' (2014). He did not deal with the meaning alone but, his explanation speaks much on "varied and modern forms of programming", such as end-user programming, while still remaining broadly applicable. "He describes programming as composed of five types of activities: requirements, specification, design, coding, and debugging". "Blackwell's definition of programming does not specifically include one type of activity particularly common among young programmers: recoding, also referred to as reuse or remixing. Dasgupta et al define remixing as the ''reworking and combining of existing artifacts'' (2016).

CHAPTER THREE

METHODOLOGY

3.1 Introduction

The introduction of information technology devices like tablets and smart phones have been of common and popular use among school children in the twenty - first century, and in recent years, children get to have some basic knowledge of computer-literacy as early as possible but not have the slightest of idea on the set of instructions that enables the computer to run. Also teenagers are realizing the benefits and privileges programming is offering them. Parents and teachers with some awareness of the capabilities of programming are doing their very best to give their children the needed training appreciate and explore these new age technologies. In recent times, many Ghanaian educationists share the same school of thought that teaching programming to children at an early stage of their life will improve technical skills and reasoning of the youth in the country and as a results create an opportunity for jobs in the country. Software developments is gaining more importance therefore there is the need to equip and train children to grasp the opportunity that learning coding will provide for them. Through enormous research, we discovered that a visually interactive programming tool that are well structured and can relate stories is the best way to introducing children to learn coding. We discovered from a research we personally conducted with some group of students at the upper primary level, and found out that using scratch to teach pupils how to code was quite easier to absorb by the pupils better than other programming languages like: java, C++,C# and some others. This work was voluntarily conducted by our group through a data collection technique known as observation.

During a face to face interview with students from some other tertiary institutions in Ghana, data gathered showed that some students expressed difficulty in learning programming which is commonly called coding during their university education many students as a results of poor performance and inability to cope with the demands of learning coding defer and eventually divert to other discipline. Other nations around the globe especially Africa share the same concerns faced by students who opt for computer science and other relater courses at the tertiary level of education. To this effect, many countries in Europe. Asia and America have commenced the teaching of coding to children at the basic level and has furthermore added coding to their educational curriculum.

3.2 Methods Of Research

This research work referred to some methods of research; qualitative and action method though there exist other which was not referred to in this work. Qualitative research includes processes such as observations and interviews, in which the information is observed, written or spoken, while action research involves learning by doing. The action research for this study includes, making pupils create Scratch projects and using it to solve some basic problems or learning some lessons, and the other research method, is qualitative method. It used observation technique to observe the Scratch projects created and hands-on practice done by the pupils, evaluating projects done by pupils to measure level of understanding.

3.2.1 Interviews

There exist various types of interviews that may be used in research; structured, semi-structured, and unstructured interviews. We chose both unstructured and semi-structured interview. The reason for this is that we needed to get views, opinions from both interviewees to comparatively determine the success or failure of the Scratch programming tool as a tool to teach or introduce programming to pupils. ‘The interview is a flexible tool for data collection, enabling multi - sensory channels to be used: verbal, non – verbal, spoken and heard’’. The order of interview can still be controlled while still giving space for spontaneity’’ (Cohen et al, 2007 p.349). We had a clear list of issues and questions to be addressed and answered. A list of questions was sent to them through their e-mails, WhatsApp and others by SMS.

Using semi-structured, we had discussions through mobile phones and electronic mail, with some programming lecturers, graduates, and undergraduates, in computer science soliciting for their knowledge on the use of scratch to teach pupils.

3.2.2 Teaching Experience

Research have shown that the implementation of scratch as a teaching aid to teaching pupils code is effective and efficient as it helps pupils to learn how to code, which can aid children in computational and logical thinking to solve problems. Exploring and deploying the method listed below effectively using a scratch programming tool or environment is essential to deliver the basic concepts of programming to pupils. Therefore, tutors should have prior knowledge of these before teaching the pupils.

We think Scratch programming class should be within 40-45 minutes per day. Areas like those listed below should be considered.

- Fundamentals of computing concept.
- Consider Program structuring.
- Explore various solutions for programming efficiency.

Embedded in this third point, the instructor can set study activities for pupils that should be carried out by the pupils.

Firstly, teach pupils how to make different things occur simultaneously. Here, they should be guided to do work like converting ideas into Scratch Script; make or use Sprites do different things they can think of, except it is not allowed on the programming environment.

The second activity is that Sprites (same or different) should be made to have interaction with other Sprites. Also, engage two Sprites to converse with one another and then introduce what is known as broadcast block.

The last activity should focus on the program flow style such as loops, if conditional statements. The pupils should be taught how loops and repeats, conditional statements like the If () Then blocks are made.

This project aims at teaching coding to pupils at the primary level in Ghana, therefore instructors that will handle the coding course should be learned, well trained and giving the opportunity to embark on developmental programs.

3.2.3 Questionnaires

The authors were not fully convinced as to the use of questionnaires for some reasons; the first reason is that there is too little number of suitable contacts who could have benefited from the study by completing questionnaires.

The other reason is that there was an increased possibility of having low response rate.

3.3 Theoretical Method

3.3.1 Tools for Coding

There are many tools that are readily available for teaching pupils how to code. Some of them include:

Toolkit, Girlbot or Boybot, Robomind, Hopscorch, Scratch, Alice, Tynker, Etoys etc. Also, there are some designed games that help teach kids basic methods for solving problems. Such as, Lightbot Jr which is specifically designed for ages 4-8 and player can choose Girlbot or Boybot with the goal of solving puzzles that use game mechanics.

Based on review on other literature works, we found out that scratch programming tool is one of the best tools to teach children basic programming. It has been known that, scratch programming tool is an open source tool that is available free of charge. This particular tool was developed by Mitchel Resnick of the Massachusetts Institute of Technology(MIT) in the year 2005. Scratch programming tool has been revised and simplified to the new version known as scratch Jr in the year 2014 which can be used to teach pupils below age of seven. Our focus tool will be scratch throughout this chapter.

3.4 The Software Development Lifecycle.

3.4.1 Software/Project Requirements and Specification.

This phase defines requirements or ask question like what is the software required to do after development. It means to decide on the intended result of executing the program. For example, is intended purpose geared towards developing a war game where the actor or object so to say battle with opposition along the way, or a narration of a story about my cat.

When illustrating a story or game, similar steps can be followed listing or labeling. For this phase, the pupils are clear on what or who is involved in the story and what the story will be about basically. The requirements phase rarely includes what is known as the “how” and therefore are very necessary.

The consideration of a higher-level component to requirement should be of concern. Here, do consider your audience. To know what an audience actually want demands a degree of empathy (be in the same shoe with your audience) that children is not capable till when they become adults. “A study of fourth graders using the LaPaya blocks-based environment found that they rarely created programs for anyone other than themselves” (Hansen et al, 2015).

3.4.2 Project Specification

This aspect focus on customers and developers defining the program to be implemented and the constraints on its function. It can also involve the identification of when software will be executed, and allowing for variation in diverse situations. This level, decides the component of the software’s input and output, as well as the general link up between input and output. In this level the most important aspect is the context. In the narrative levels, actions start to appear in the listing and

Connecting phases and cause-and-effect relationships appear in the sequencing stage. Here, children, been the developer takes into consideration the kind of project to be developed, is it a game, story, or animation. Which of the software development technique should be used in developing the project; is it waterfall, integration, or reusable. They spell out the inputs and what output will be derived from the input, that is, the study of the relationship between the input and the desired output.

3.4.3 Software/Project Development

This phase focus on the design and the actual programming or coding of the defined and desired software.

3.4.4 Design

At this design stage the programmer selects from a collection of technical features that may be supportive to the desired behavior. This phase of the software development life cycle ask the question like ‘'how’’. Design can be like making a sketch of how the project will be like after the development, writing out algorithms and algorithm sequence (that is, a logical arrangement) of the block of codes that will be used for the project. There is also the need for a design strategy that is effective and this requires that one should be familiar with multiple strategies already, with the ability to evaluate the most effective strategy. From the narrative point of view, this requires one to be aware of the relationship between actions (sequencing) and to have a clear notion of the purpose of the story (narrating).

3.4.5 Coding

This is the phase where the software is been developed. To the pupils, it is where the games, storytelling, and animations projects are developed based on the requirements and specification. Coding can either be of a wide range of abstract reasoning capabilities or can be just straightforward as entering abstract control commands as well as data. The actions (sequences) involved with coding can map to any phase of the process of narrative development. This phase is where the Sprite or Sprites are chosen including other features like backdrop, other tools like the pen tool, brush, and the various block of codes needed to develop the project for the children.

We examined the attitudes and habits of the pupils whilst using Scratch program and found that they can interchange different block of codes on same Sprite to make the same Sprite or Sprites behave differently. We discovered that children had the capability of carrying out coding tasks at a low level of abstraction without doing much planning.

3.4.6 Software/Project Validation & Maintenance

This is where the software or project is checked to ensure that it is what the customer requires.

The user of a software project requires that he or she gets value for what was paid for by him or her, and that the software meets all the requirements that the user specified. This phase involves debugging where anticipation and accounting for deviation from the intended behavior must be considered. After a prototype software or project is delivered to the customer who actually is the user, any bug or error identified should be appropriately fixed. Bugs or errors may appear in one's program but bearing a clear sense of the user's requirements debugging should be carried out to ensure that the required result is delivered to the user. While there may exist errors in a program, the pupils should be taught to strongly bear in mind the intended or expected behavior. Children learning computer programming with any visual block- based programming should be taught what a bug is, its causes and how or steps to debug those bugs found in their programs or stories.

3.4.7 Reuse

In storytelling, retelling has been an elemental part and therefore reuse during narrative development is likely to show up in numerous forms.

Ignorant or childlike form of reuse can be found in the projects developed by children during the initial narration and programming stages. Pupils do not often develop new or fresh characters at these stages: listing and labeling, instead they use some entities of their personal lives even from other stories that have been told them. Children, in subsequent narrative development activities or stages model themes or approaches that have been seen by them elsewhere sometimes ago in their lifetime.

3.5 Scratch

In Scratch environment, dragging blocks into a coding area and then snapping them together creates code. Developers of this programming tool noted that “We have always been intrigued and inspired by the way children play and build with Lego bricks. Given a box full of them, they immediately start tinkering, snapping a few bricks, and the emerging structure then gives them new ideas”.

The Image below is the basic interface for scratch version 1.2.1 programming tool.

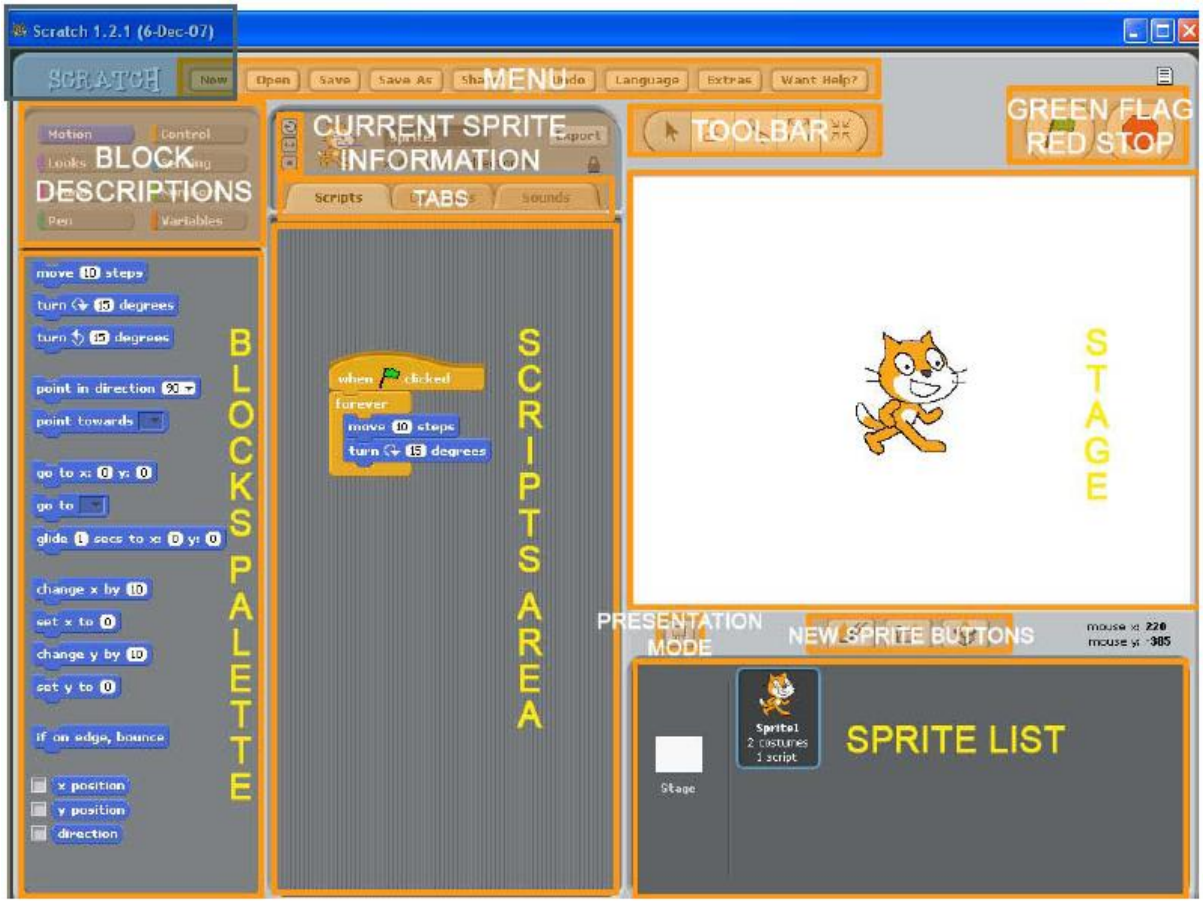


Figure 3.0 Scratch basic interface (version 1.2.1)

The image below is also for Scratch version 2

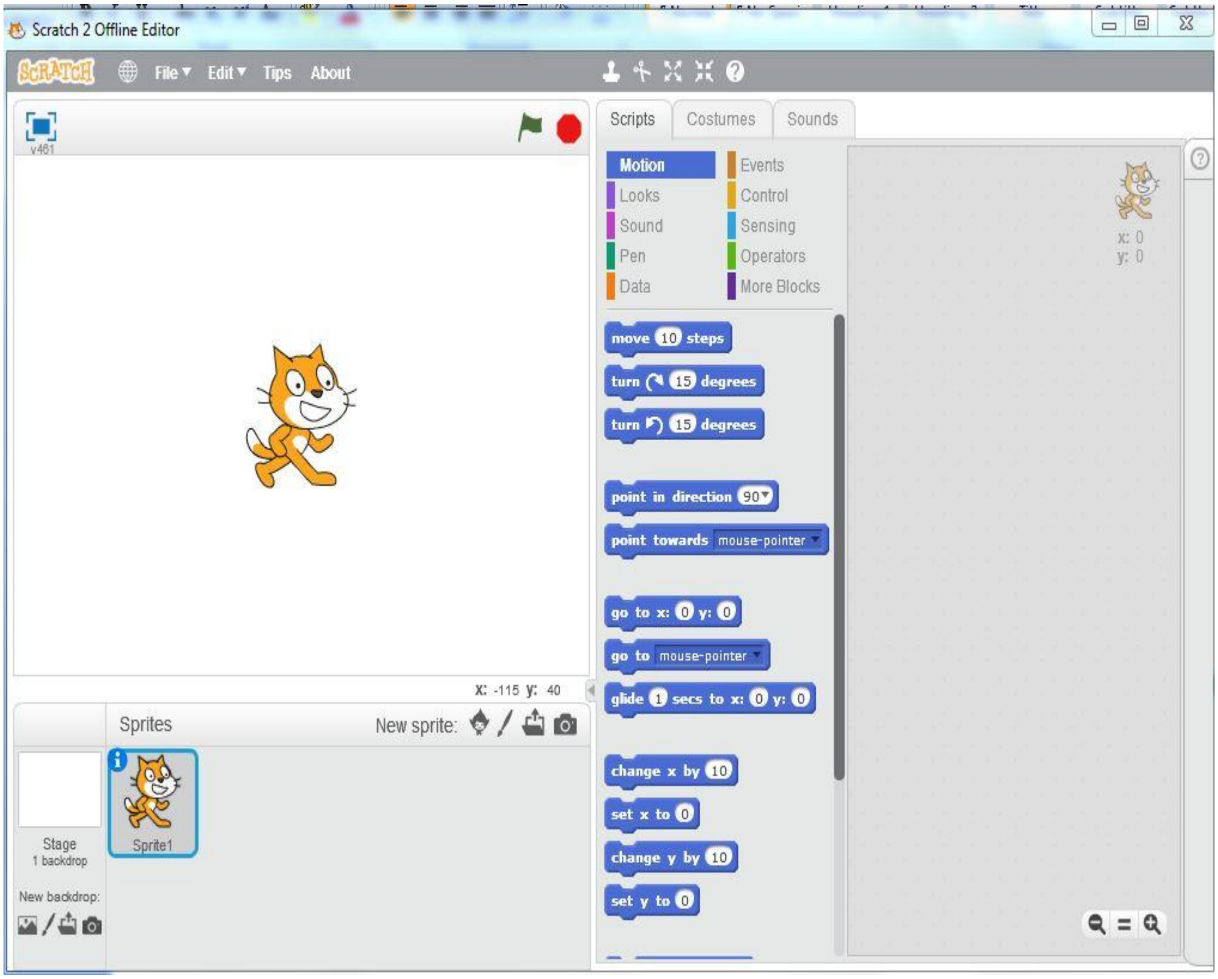


Figure 3.1. Scratch basic interface (version 2.0)

To start Scratch go to Start menu, click All Programs then select and double-click on Scratch or double-click on the Scratch from your computer's desktop.

3.5.1 Menu:

This contains where you create a 'New' project, 'Open' or 'Save' an existing project, 'Save (a new project) as' whatever name you choose, 'Undo' a previous action, and obtain 'Help'.

3.5.2 Block Descriptions:

The block description area lists the eight categories of blocks and has two other special blocks (Data and more blocks) each block category has a color different from other blocks categories so when you see a block of a specific color, you can quickly determine the category it belongs.

3.5.3 Blocks Palette:

This area shows all of the blocks available to you for use in your programming. Note that the blocks palette will change depending upon the current block category. When you select a new block category, the blocks palette will change to reflect the new options available.

3.5.4 Current Sprite Information:

Here you will find the name and picture of the current sprite together with its x-y position, direction, and rotation style.

3.5.5 Tabs:

Allows you to have a view of the current sprite's **"scripts**, **costumes**, and **sounds"** and make amendments. "Scripts tab" reveal scripts that exist at the moment to create scripts relating to the one that exit at the moment. With the "costumes tab" you are able to edit, create or copy a costume from scratch or from a file. "Costume" is the visual image of the sprite on-screen. "Sprites can have multiple costumes and use scripts to change between them." "Sounds tab displays the current sprite's sounds".

3.5.6 Scripts Area:

Relating to ongoing scripts, we develop (create) and look at (view) the scripts.

3.5.7 Stage:

“Stage is where all of the action takes place. The stage is 480 units wide by 360 units tall and the center of the stage is at x-y coordinate (0, 0). This means the lower left is at (-240, -180), the upper left is at (-240, 180), the upper right is at (240, 180), and the lower right at (240, -180).”

3.5.8 Toolbar:

“A number of tools exist for your use. The arrow is the default selection and it allows you to pick up and move sprites and blocks of code around. There are also options for you to duplicate and delete items as well as grow and shrink your sprite.”

3.5.9 Green Flag / Red Circle:

“The “green flag” is what you click to start or run the main program and the red sign to end it.”

3.6 New Sprite Buttons:

“Using these buttons, you can paint a new sprite, choose a new sprite from a file, or get a surprise (random) sprite.”

3.6.1 Sprite List:

“On the left, you will see a thumbnail for the stage. Clicking on this thumbnail changes the ‘Current Sprite Information’ area to reflect the properties of the stage. Stages can still have scripts and sounds. However, to change the appearance of the stage, you would select a different ‘Background’ as opposed to ‘Costume’. On the right, you will see thumbnails of all of the sprites in the project together with the sprite’s name, amount of costumes, and amount of scripts. You can easily change the current sprite by clicking on a different one.

When you do this, the ‘Current Sprite Information’ area together with the ‘Scripts area’ are updated too.”

3.6.2 Presentation Mode:

This provides a full-screen view of the stage.

3.6.3 Learning How To Use Scratch

We are of the belief that the best way to learn Scratch and other programming languages is through experimentation (hands-on practice) .Do not worry; Scratch is an interactive environment, easy to use and enjoyable.

3.6.4 Writing Simple Scripts

To create a script, we simply drag a block from the blocks palette onto the Scripts area. To run it, we can double-click it and observe what happens on the stage.

At the moment, our current sprite is Sprite1 (the cat). By default, he is located in the center of the screen. You can drag him anywhere on the screen that you wish at any time.

3.6.5 Basic Movement:

Let’s make him move 10 steps forward by selecting from the Blocks Palette and dragging it onto the Scripts Area. When you double-click the block, you should observe the cat move 10 steps to the right. You can double click the block as many times as you wish. The cat will continue to move.

3.6.6 Editing A Text Field:

You can edit the white text field portion of the block by clicking on the ‘10’ and changing it to another number like ‘--8’. Double click it and see what happens. Now change it to ‘98’ and observe the difference.

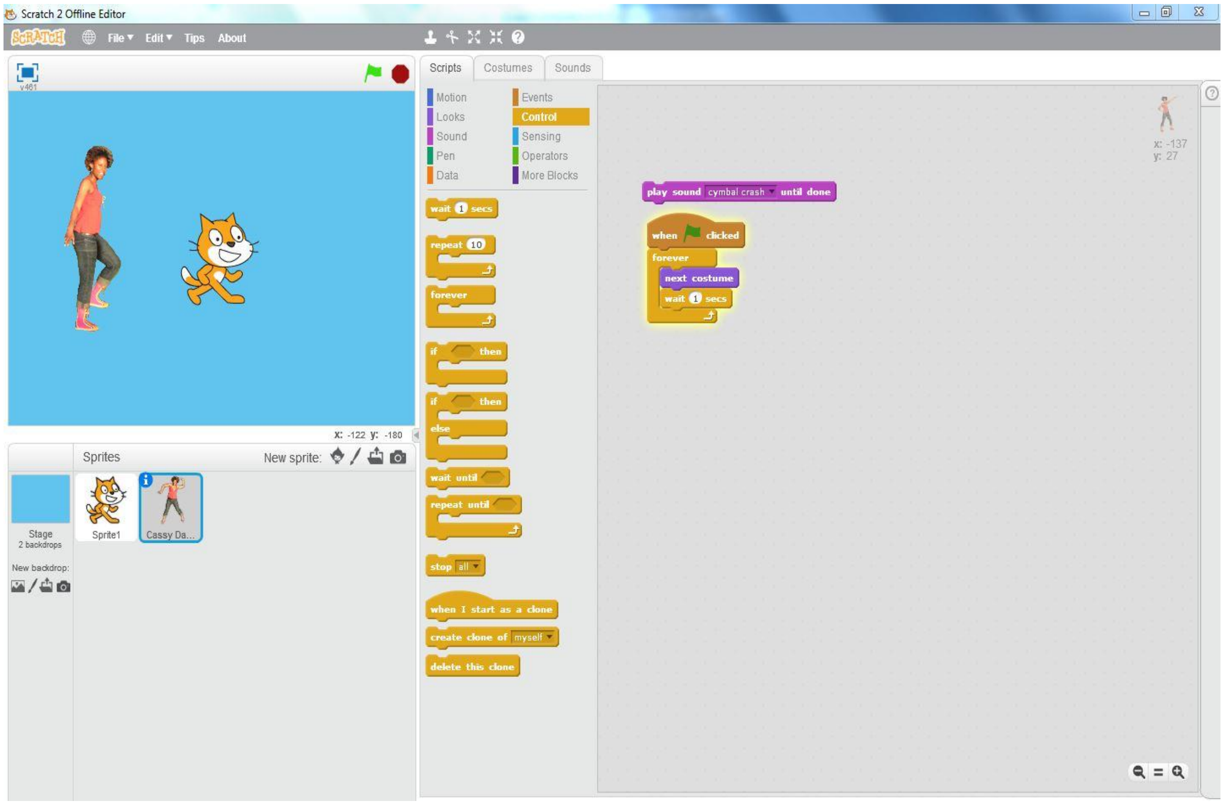
3.6.7 Help:

To find out what a block of code does, simply right-click on the block and select ‘Help’ from the pop-up menu.

3.6.8 Expanding A Script

To expand the script, simply snap a second block to the first. Scripts are executed from the top to the bottom so you need to add the block accordingly. If you want the new block to execute first, add it on top of the existing block. Otherwise, add it below. As you drag a block into the Scripts Area, a white line will indicate where you can properly join the new block with the existing script structure. When you are ready, you can double click anywhere on the new script to execute it.

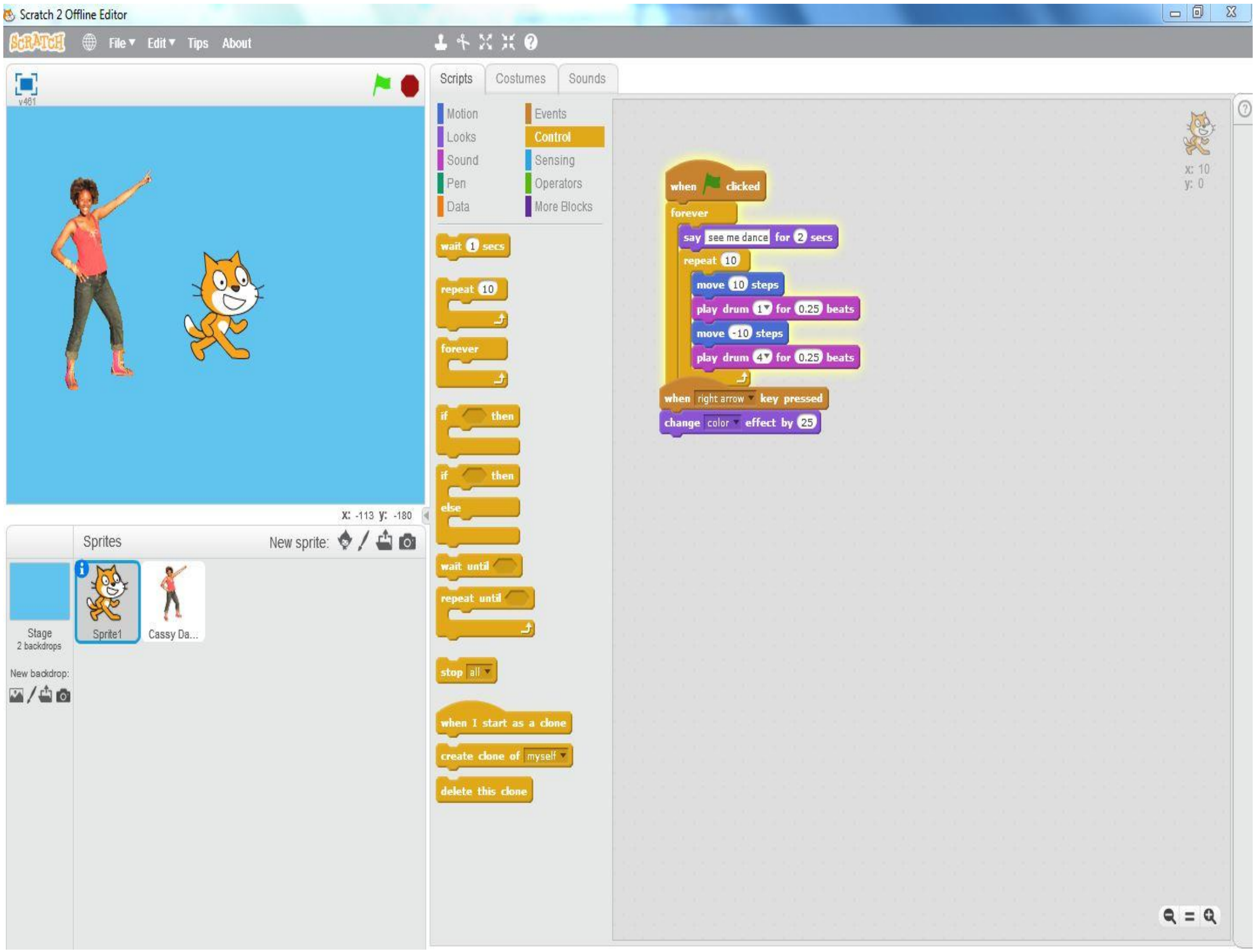
A Dance Show With The Use Of Scratch



```
play sound cymbal crash until done
```

```
when clicked
forever
    next costume
    wait 1 secs
```

Figure 3.2 A dance show with the use of Scratch



```
when [flag] clicked
forever
  say [see me dance] for (2) secs
  repeat (10)
    move (10) steps
    play drum (1▾) for (0.25) beats
    move (-10) steps
    play drum (4▾) for (0.25) beats
when [right arrow ▾] key pressed
change [color ▾] effect by (25)
```

Figure 3.3. A dance show with the use of Scratch

The various blocks from different blocks categories were used for both Sprites. Another Sprite (Cassy Dance) was selected apart from Sprite 1. It also has its own set of codes.

Learning Programming with Scratch (version)

Scratch is a programming language that lets you create your own interactive stories, animations, games, music, and art.

This guide shows you how to make a project in Scratch

To start a new project, go to

Scratch and click Create.

If you have a Scratch account, sign in so your project saves.

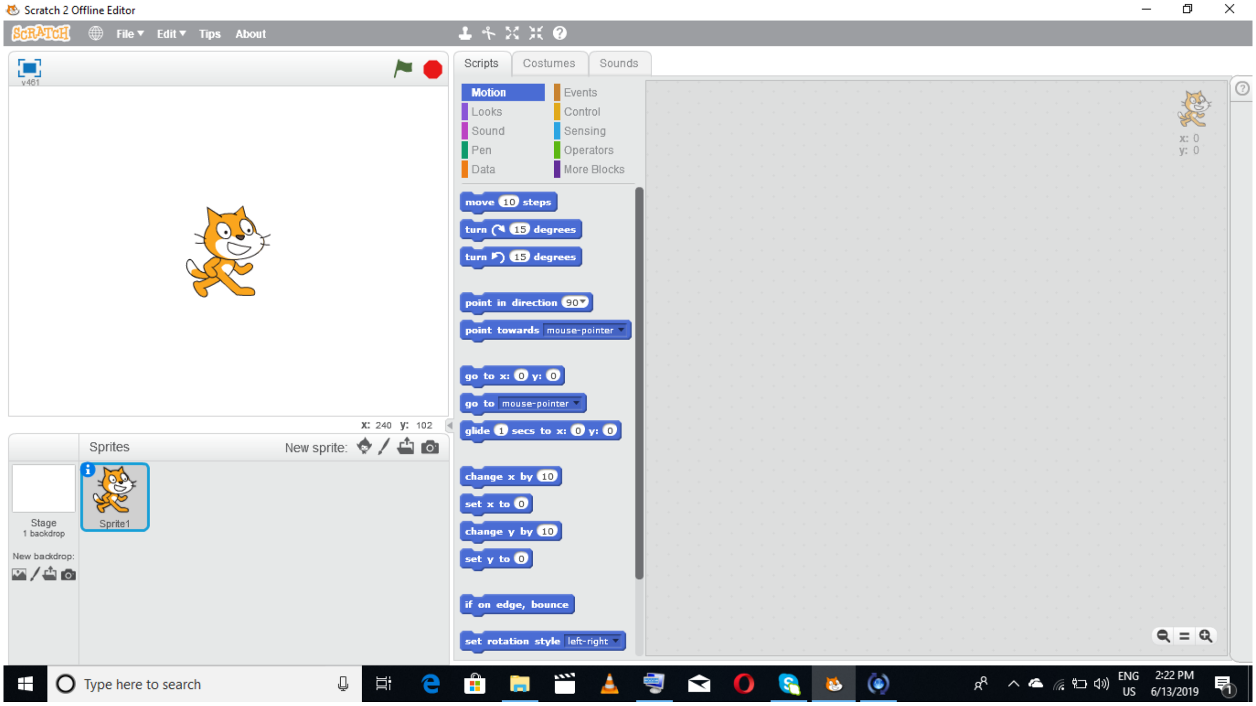


Figure 3.4 To start moving

Step 1: **Start Moving**

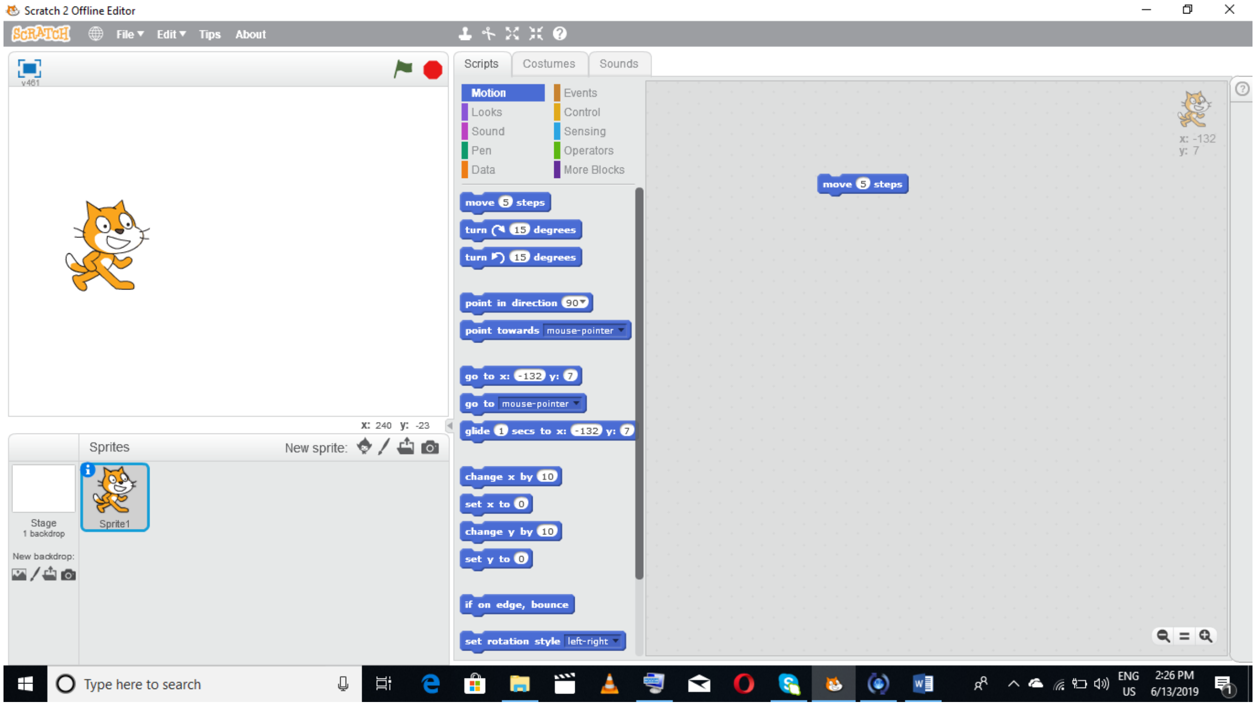


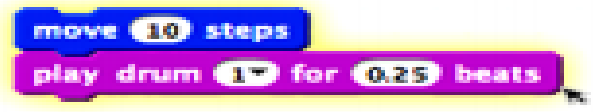
Figure 3.5 Select move

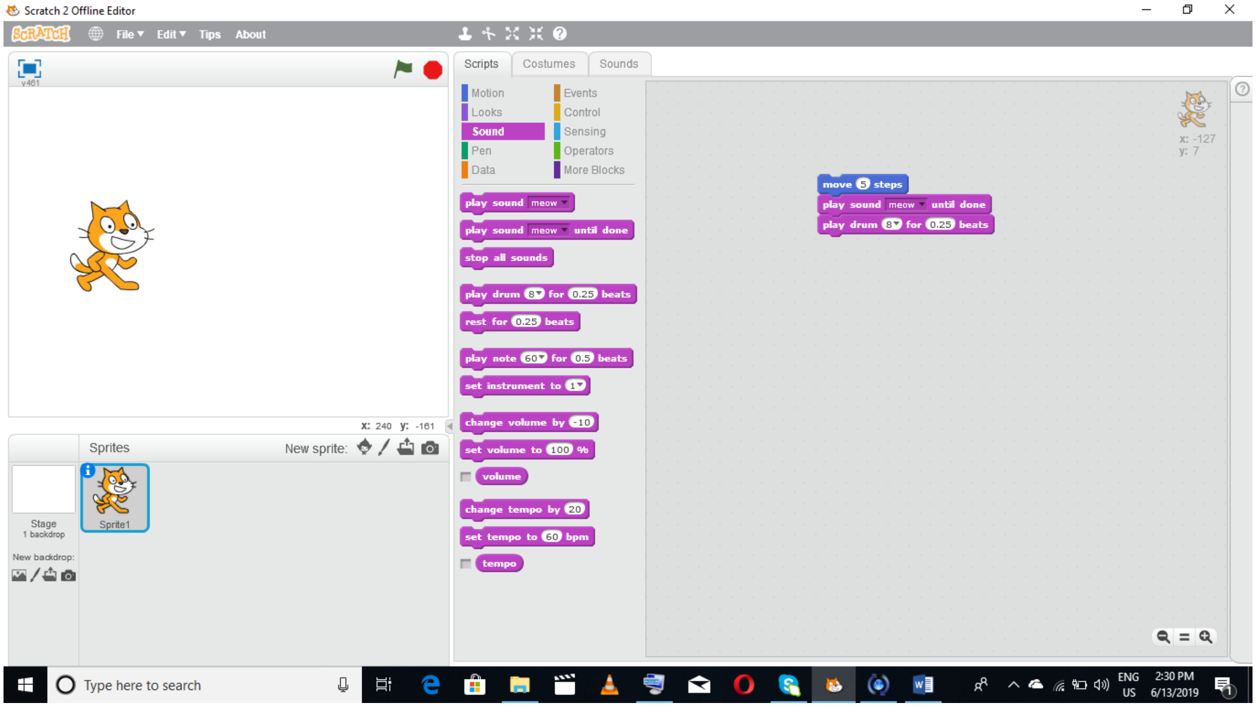
Drag a **MOVE** block from the motion category of the Script Tab into the Scripts area.

Click on the block to make the cat move.

Step 2: **Add a Sound**

Drag out a PLAY DRUM and snap it onto the MOVE block.


```
move 10 steps
play drum 1 for 0.25 beats
```



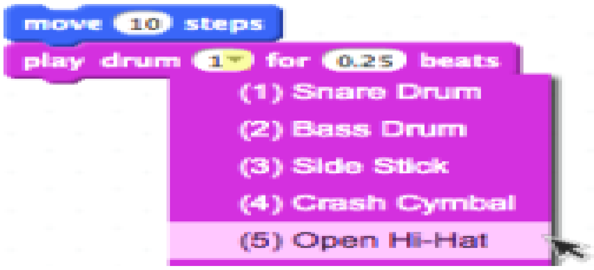
```
move 5 steps
play sound meow until done
play drum 8 for 0.25 beats
```

Figure 3.6 Play drum.

You can add more sound by dragging out another PLAY DRUM and snap it to the first one (PLAY DRUM)

Click and listen.

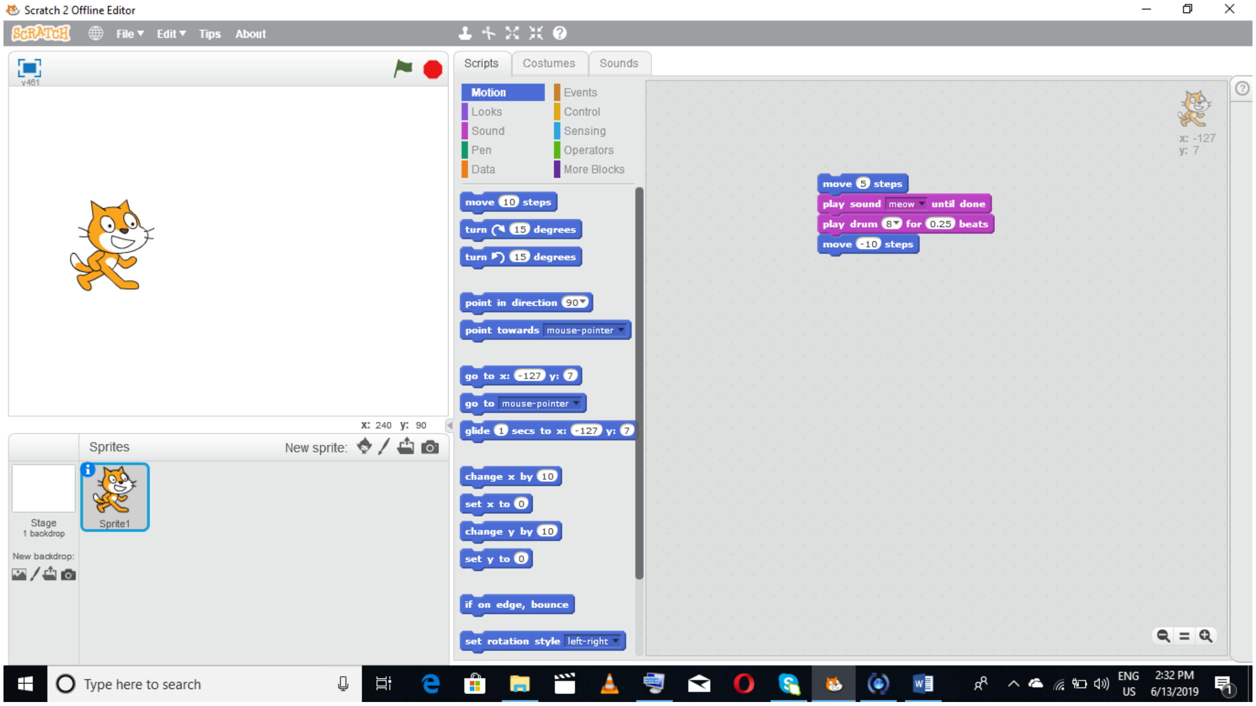
If you can’t hear it, ensure the sound on your computer is put on.



```
move 10 steps
play drum 1 for 0.25 beats
```

You can then choose different drums from the drop-down menu.

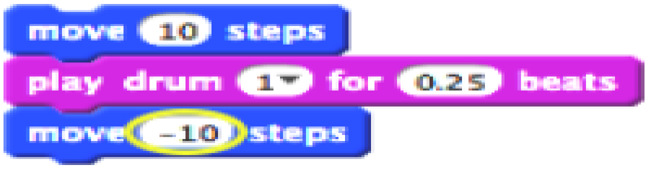
Step 3:



```
move 5 steps
play sound meow until done
play drum 8 for 0.25 beats
move -10 steps
```

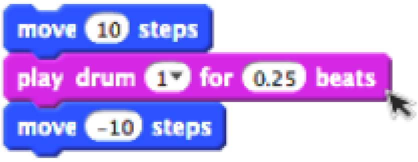
Figure 3.7 Another Move and Play drum.

Start a Dance



```
move 10 steps
play drum 1 for 0.25 beats
move -10 steps
```

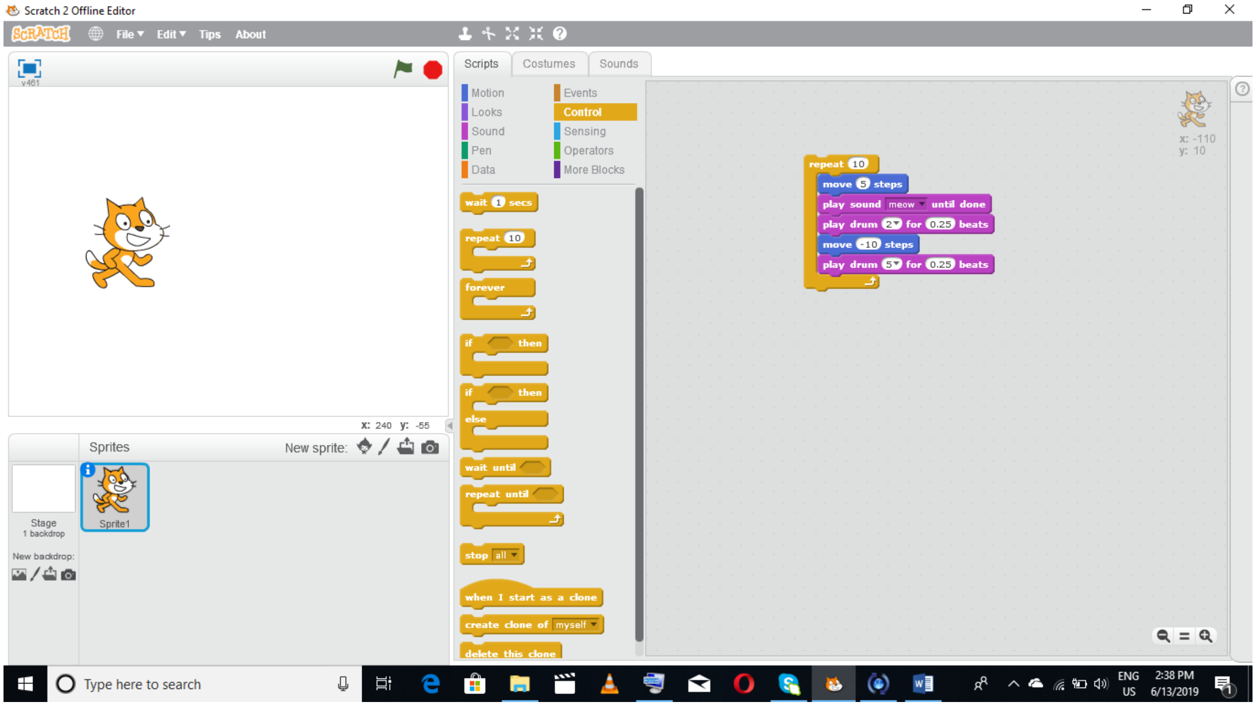
Add another MOVE block and click inside the block and type in a negative or minus sign.



```
move 10 steps
play drum 1▾ for 0.25 beats
move -10 steps
```

You can click on any of the blocks to run the stack.

Step 4



```
repeat 10
    move 5 steps
    play sound meow ▾ until done
    play drum 2▾ for 0.25 beats
    move -10 steps
    play drum 5▾ for 0.25 beats
```

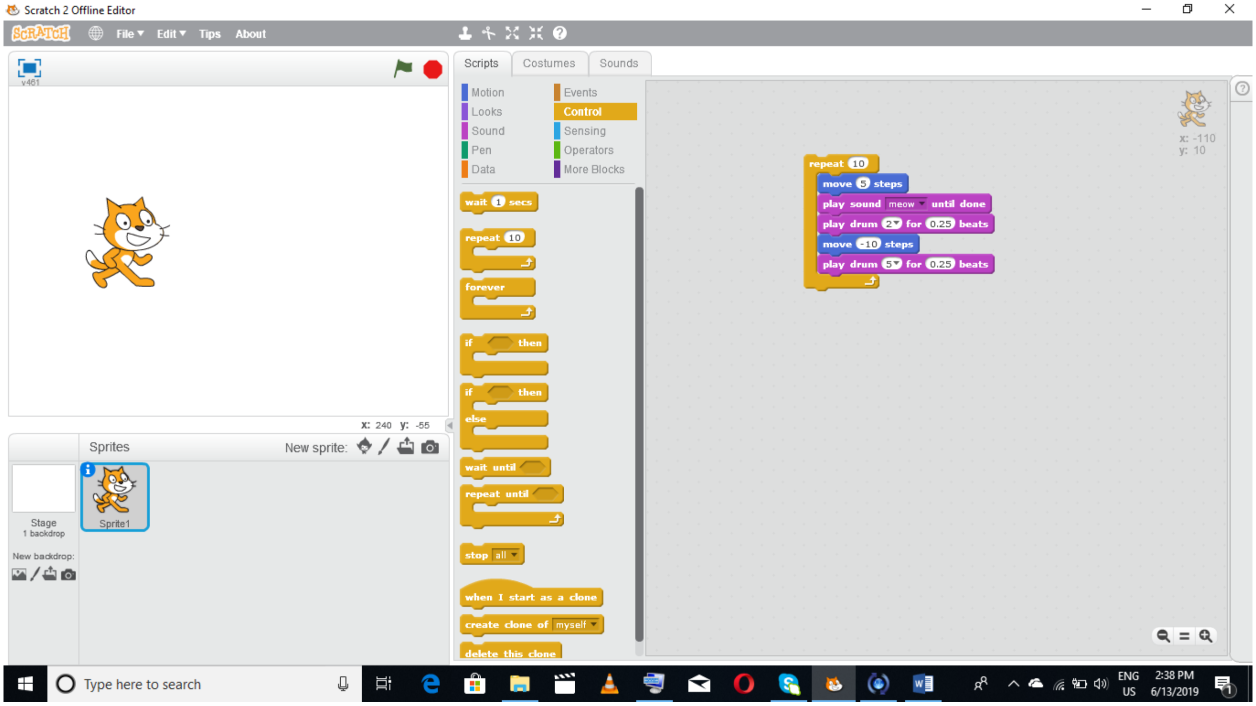
Figure 3.8 Another Play drum

You can also add another PLAY DRUM block, and choose a drum from the menu. Click to run.

<u>Step 5;</u> Again **and again**

Drag out a REPEAT block and drop it on top of the stack of the scripts area. You can make the mouth of the REPEAT to wrap around the other blocks. Just drag the repeat block to fit around the first block to the last block

To drag a stack, pick it up from the top block.



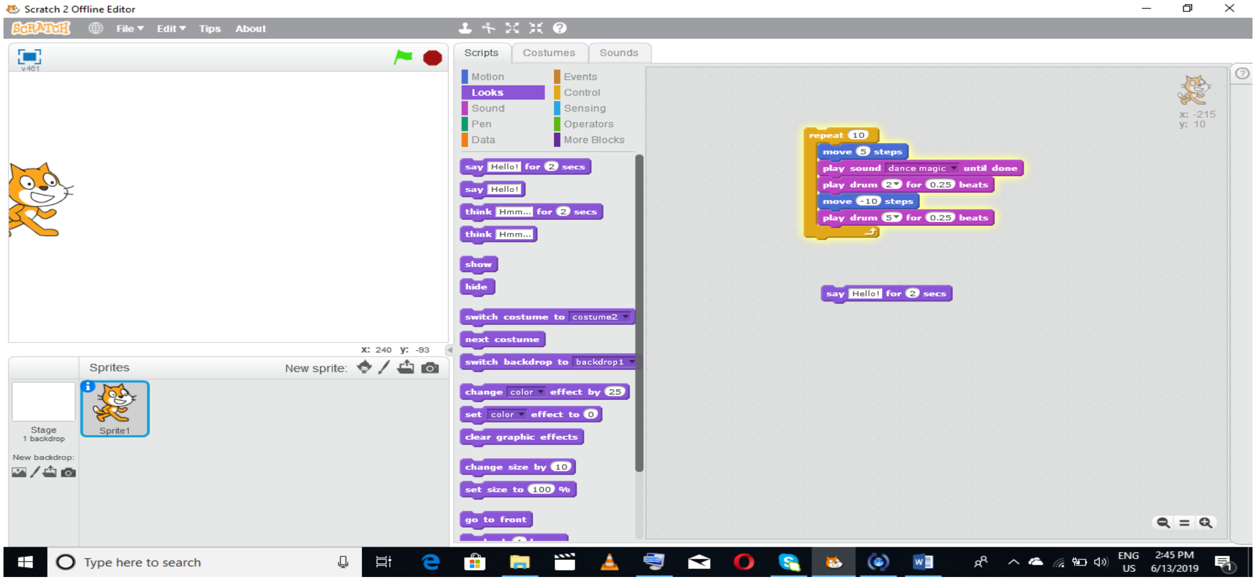
```
repeat 10
    move 5 steps
    play sound meow until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats
```

Figure 3.9 Select a repeat block

Click to run.

You can change how many times it repeats. You can click on any block to run the stack of blocks.

Step 6: **Say Something**



```
repeat 10
    move 5 steps
    play sound dance magic until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

say Hello! for 2 secs
```

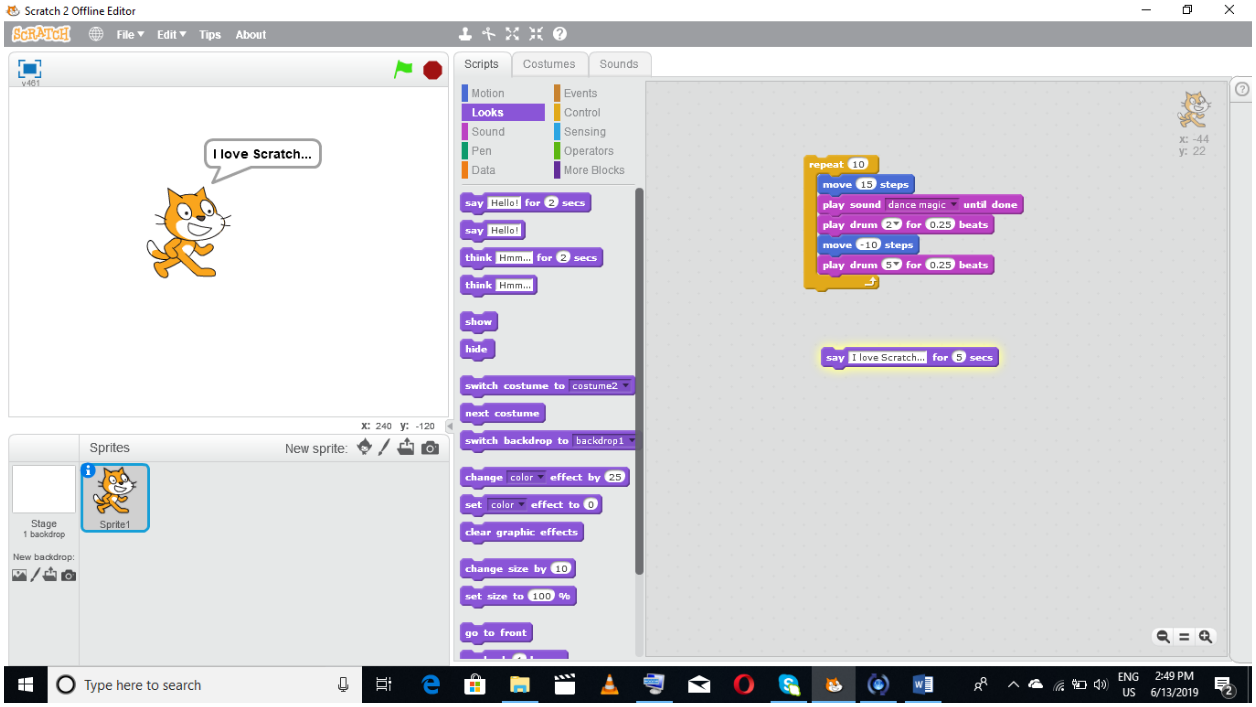
Figure 3.10 Say something

Click on LOOKS blocks category and drag out a SAY block.

Click inside the SAY block and enter by typing a different word or words to change the words.

Then snap the SAY block on the top.

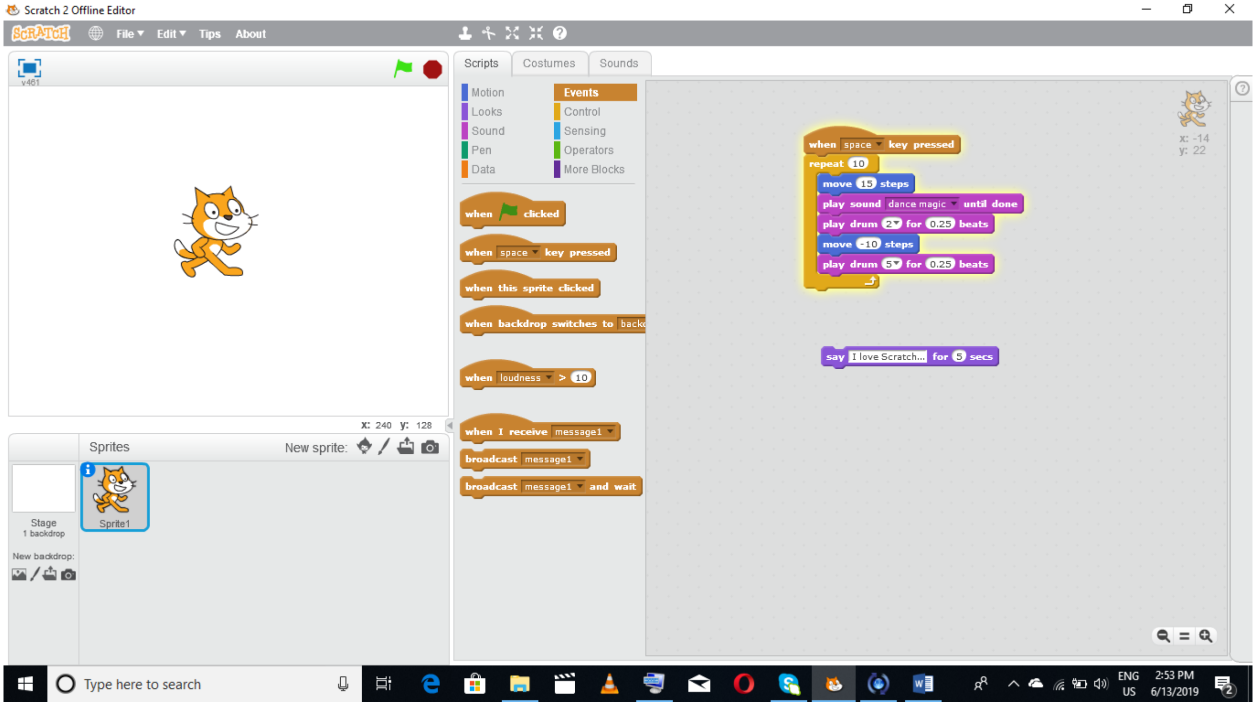
The Sprite says ‘I LOVE SCRATCH’



```
repeat 10
    move 15 steps
    play sound dance magic until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

say I love Scratch... for 5 secs
```

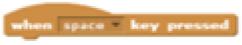
Figure 3.11 Say something

Step 7:



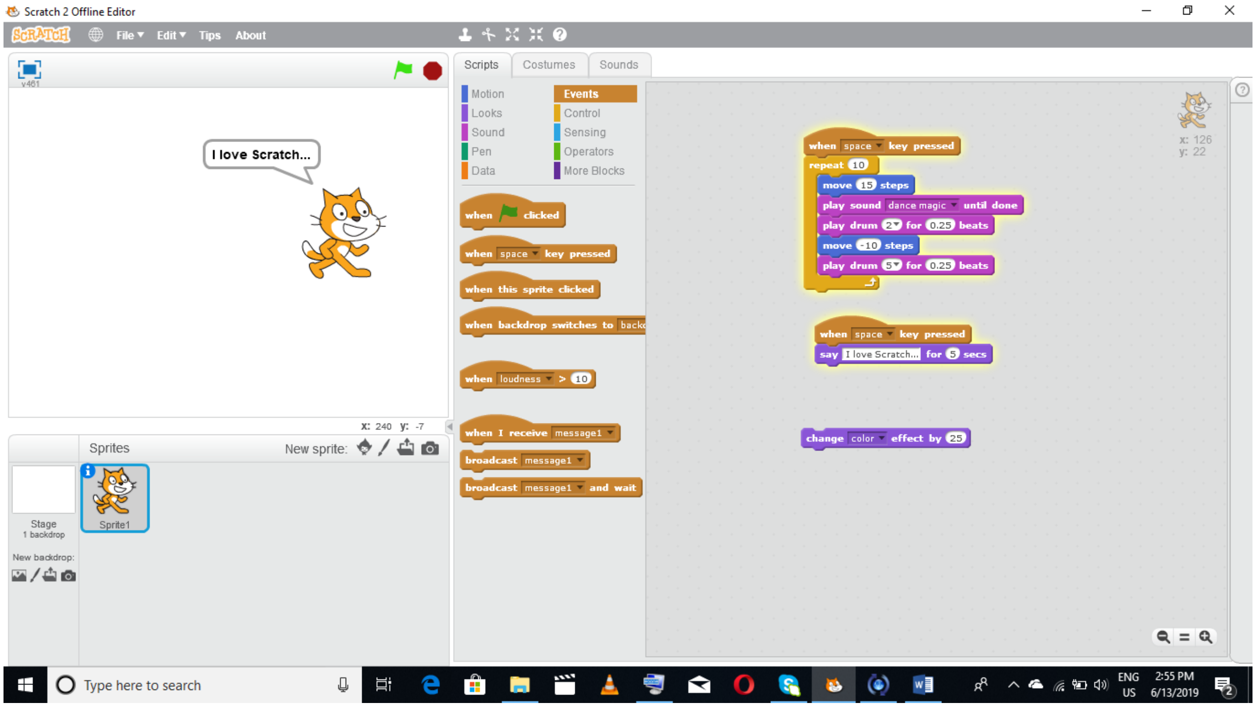
```
when space key pressed
repeat 10
    move 15 steps
    play sound dance magic until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

say I love Scratch... for 5 secs
```

Figure 3.12 Add an event block.

Drag out a  block from EVENT block and snap it on top.

<u>Step 8:</u> Color Changing

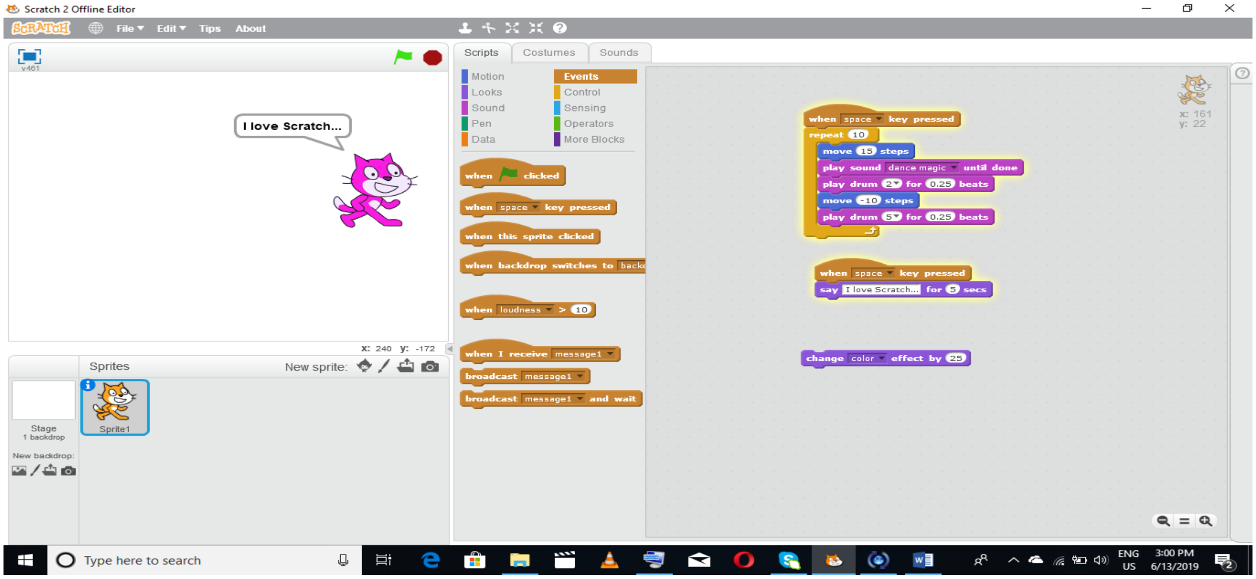
Drag out a CHANGE COLOR block.



```
when space key pressed
repeat 10
    move 15 steps
    play sound dance magic until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

when space key pressed
say I love Scratch... for 5 secs

change color effect by 25
```

Figure 3.13 Changing color

The result after applying the Change Color block. The Sprite changes color to different color whenever you click on the block.



```
when space key pressed
repeat 10
    move 15 steps
    play sound dance magic until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

when space key pressed
say I love Scratch... for 5 secs

change color effect by 25
```

Figure 3.14 Changing color

Step 9: Select a backdrop

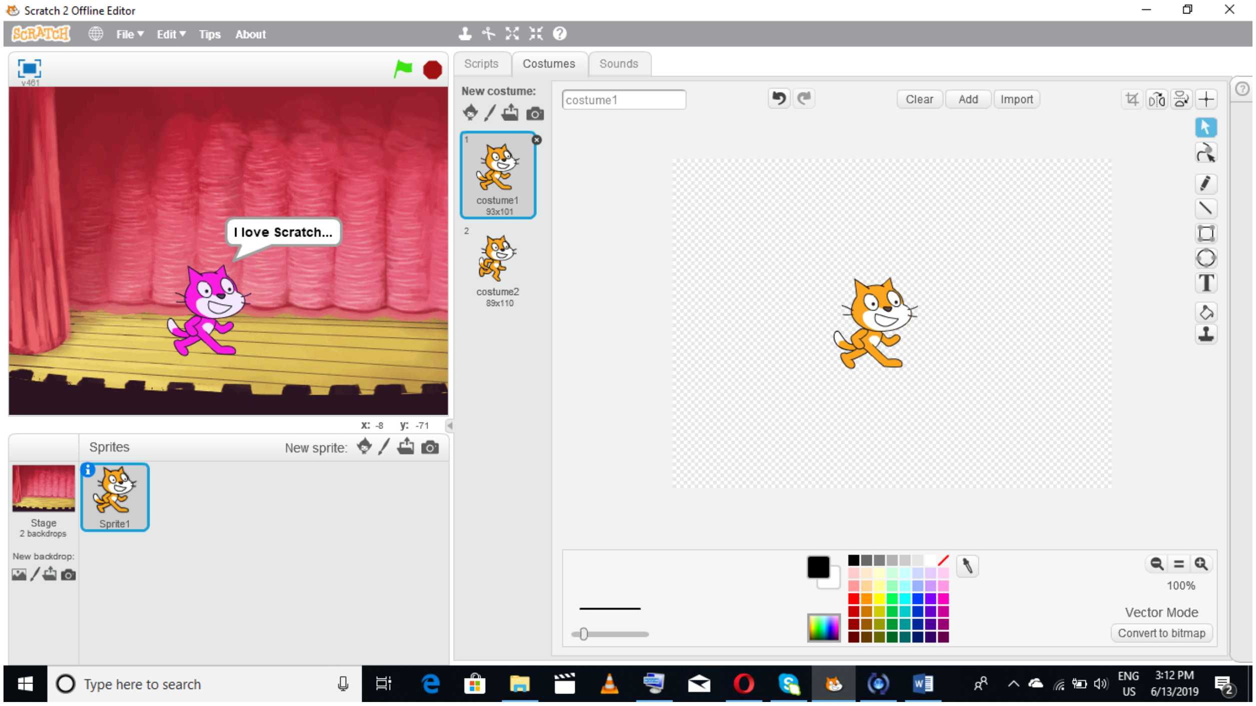
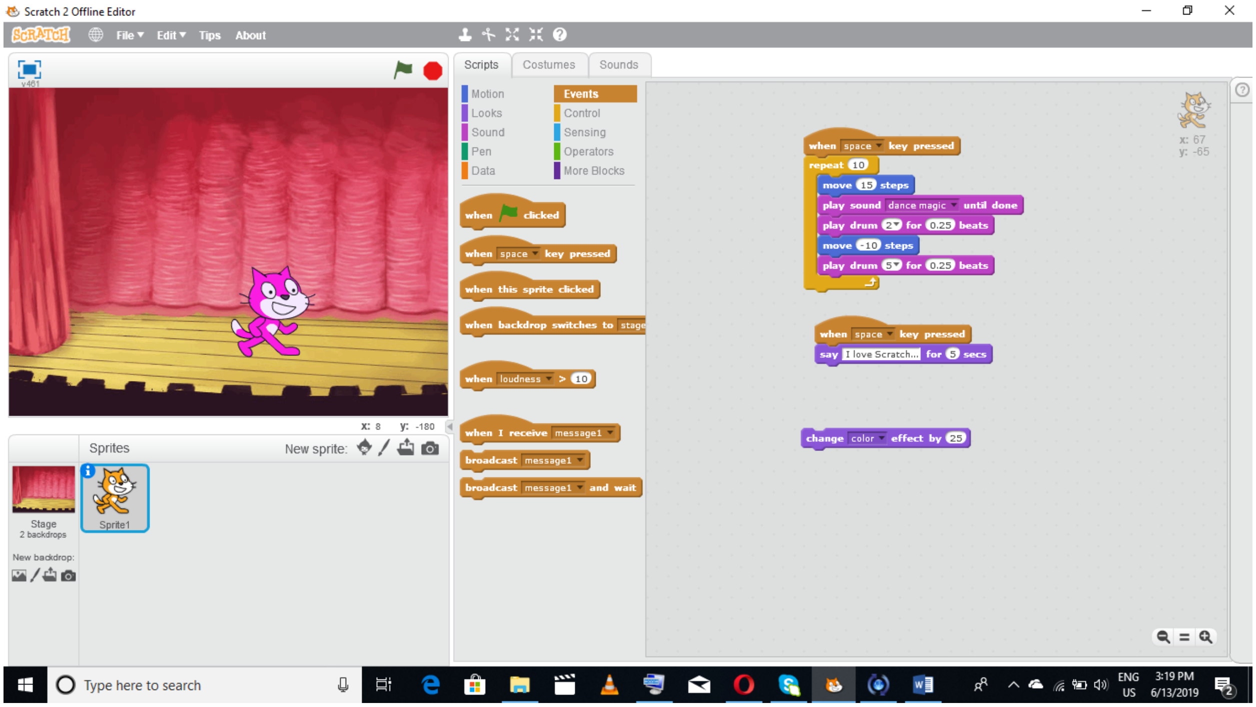


Figure 3.15 Select Backdrop

Choose a backdrop from the library (e.g. Stage 1). Then click OK.

The new backdrop then appears on the stage.



```
when space key pressed
repeat 10
    move 15 steps
    play sound dance magic until done
    play drum 2 for 0.25 beats
    move -10 steps
    play drum 5 for 0.25 beats

when space key pressed
say I love Scratch... for 5 secs

change color effect by 25
```

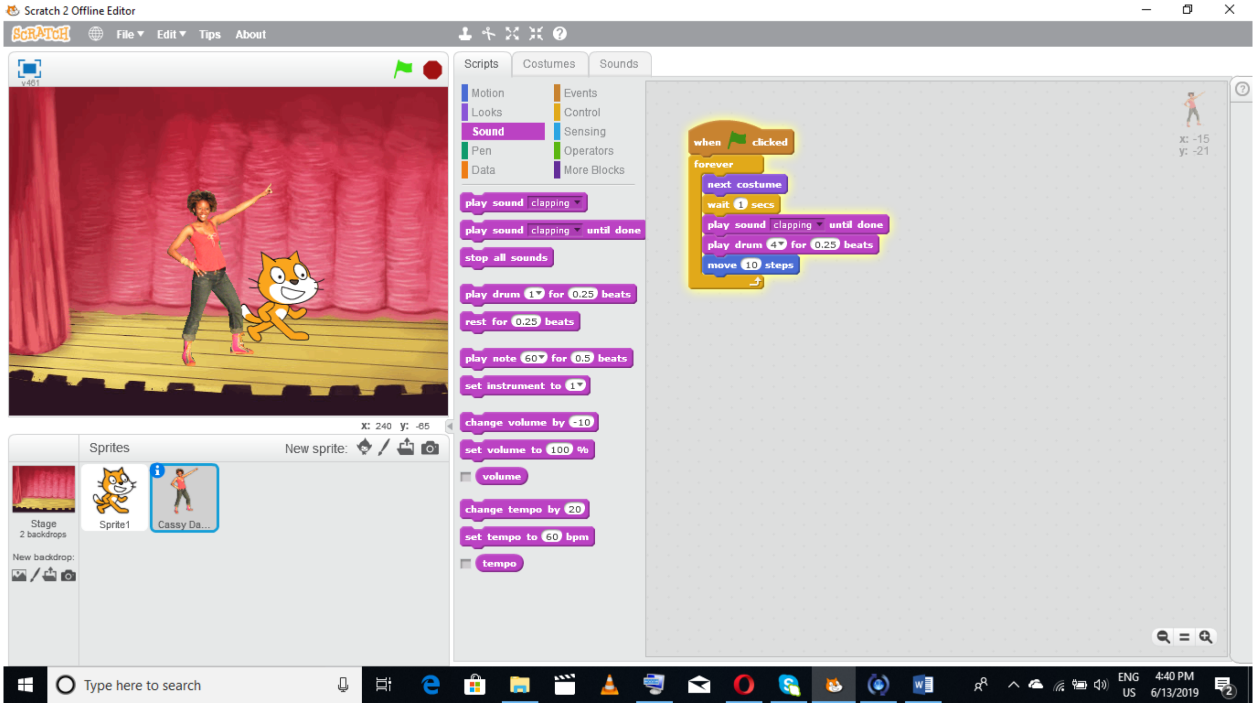
Figure 3.16 Selected Backdrop

Step 10: Add a Sprite (Object) An object is called a *sprite* in Scratch. To add a new sprite, you have to click on one of these buttons below.

SELECT NEW SPRITE BUTTON:

Select from the library for a new sprite.

For you to add this sprite, click then click Human and select “Cassy Dance.



```
when [flag] clicked
forever
    next costume
    wait 1 secs
    play sound clapping until done
    play drum 4 for 0.25 beats
    move 10 steps
```

Figure 3.17 Add new Sprite

Step 11:

Tell or show the sprite what it should do. Add sound. To add sound, click on the sounds tab. You should then select a sound

Go back and click on the Scripts tab, and drag in Play Sound a block. Select your desired sound from the drop-down menu.

3.7 Changing Costumes.

Any of the sprites can have more than one costume. In order to do this, click on the Costumes tab, and then click on a different costume for the selected sprite. Then go back to the scripts tab, click the Looks block and drag next costume to the Scripts area. Make sure it is snapped together with other scripts that are wrapped by the forever block. .You can learn from the above picture. After running the blocks of codes, the two Sprites will be dancing. Do save your completed Scratch projects.

3.8 Ways To Teach Coding To Children

As teaching children to code is a special and unique approach, the below ways are used to have an efficient and effective results.

(1)An hour class per day.

(2)Promote individual and group work by assigning children projects to work on daily.

(3)Engage children with variety of designs each meeting.

(4) Make multiple Sprites interact with each other.

(5)Help kids design programs with the use of , loops, and conditional statements.

3.9 Conclusion

Children desire and love for coding will increase as they are taught in a way that will elevate their thinking and push them to give their all to studying. This is only achieved when they are well involve in the lesion plan and can express their selves without fear or panic.Scratch programming, when taught well will build children learning capacity and logical thinking.

CHAPTER FOUR

RESULTS AND DISCUSSION

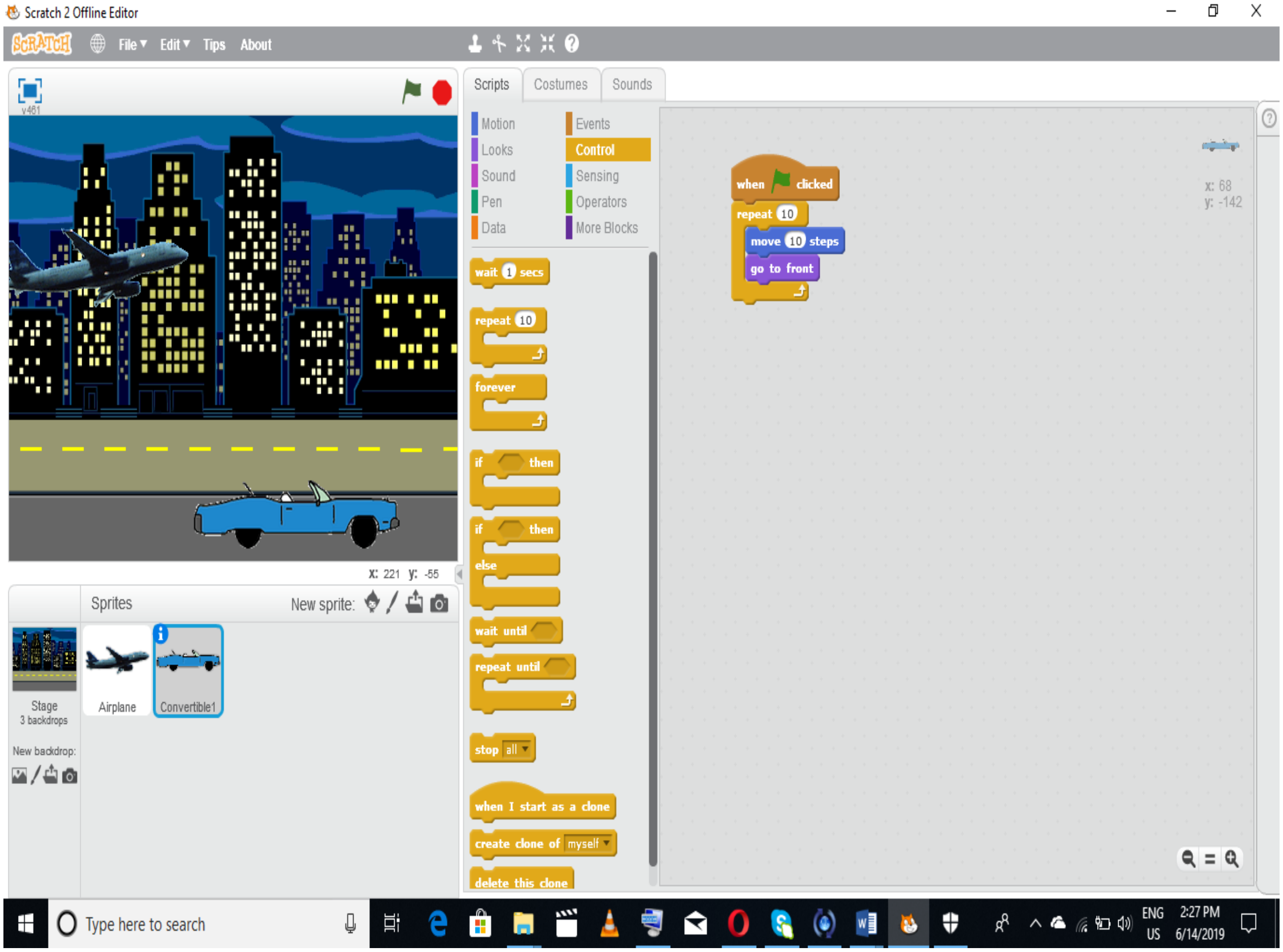
4.1 Overview

This chapter will elucidate at the information collected in the research about Scratch and how it can be used with primary students to develop programmable thinking skills. This examination will involve children interactions with the scratch environment and how they were able to use scratch program to create their own stories...

4.2 Creating With Scratch

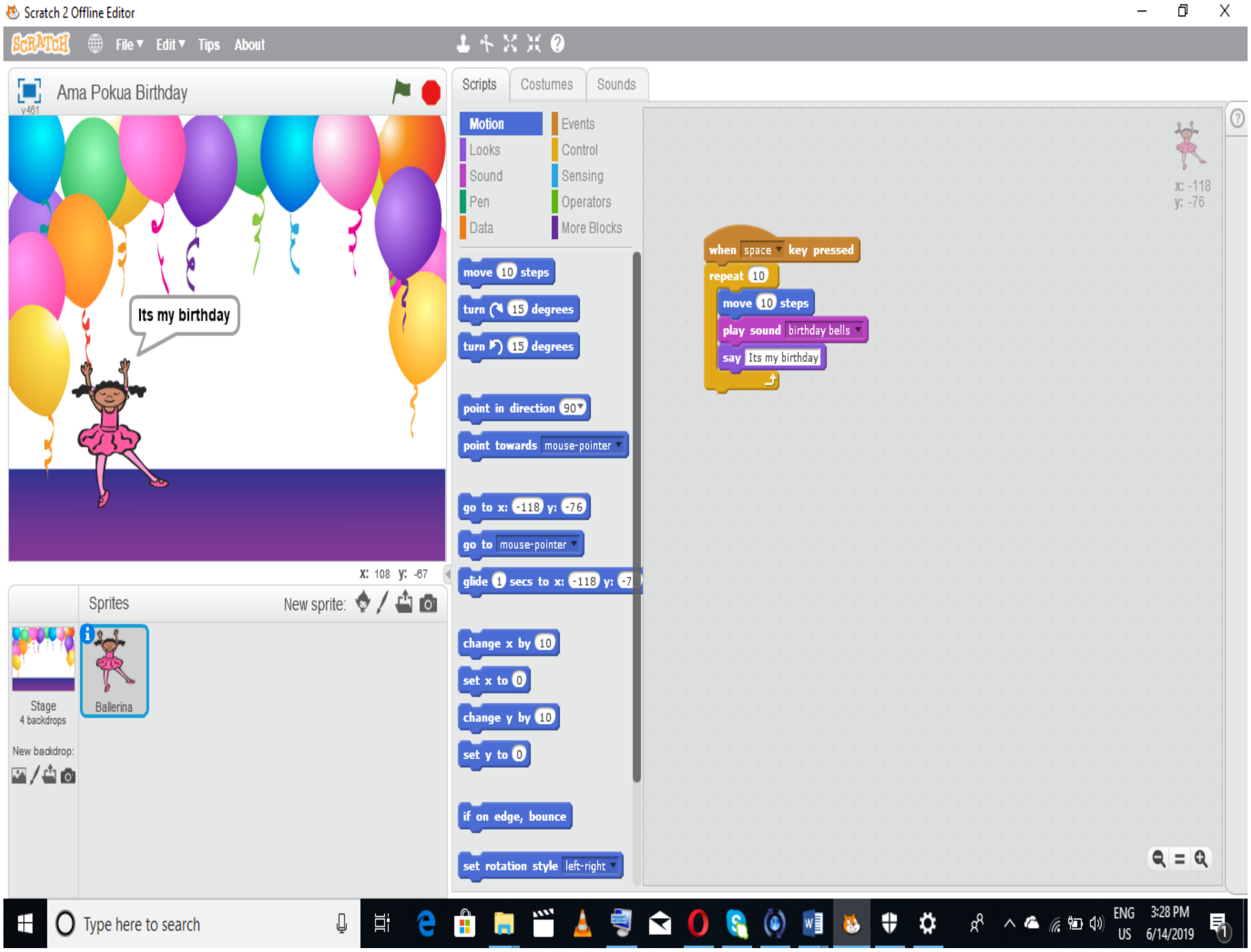
Scratch is an interactive programming environment that provide users especially children the platform to develop great programs which enhances their theoretical, logical and computational thinking ability to create their own stories and illustrations, animations, and games.

After teaching the kids how to use scratch program, they made series of programs following laid down guidelines and procedures of which was supervised by the teachers and coordinators. Below are typical designs created and programmed by some of the children we taught.



```
when 🏁 clicked
repeat 10
    move 10 steps
    go to front
```

Figure 4.0 Stephen’s private jet and car program



```
when space key pressed
repeat 10
    move 10 steps
    play sound birthday bells
    say Its my birthday
```

Figure 4.1 Ama Pokua’s birthday program.

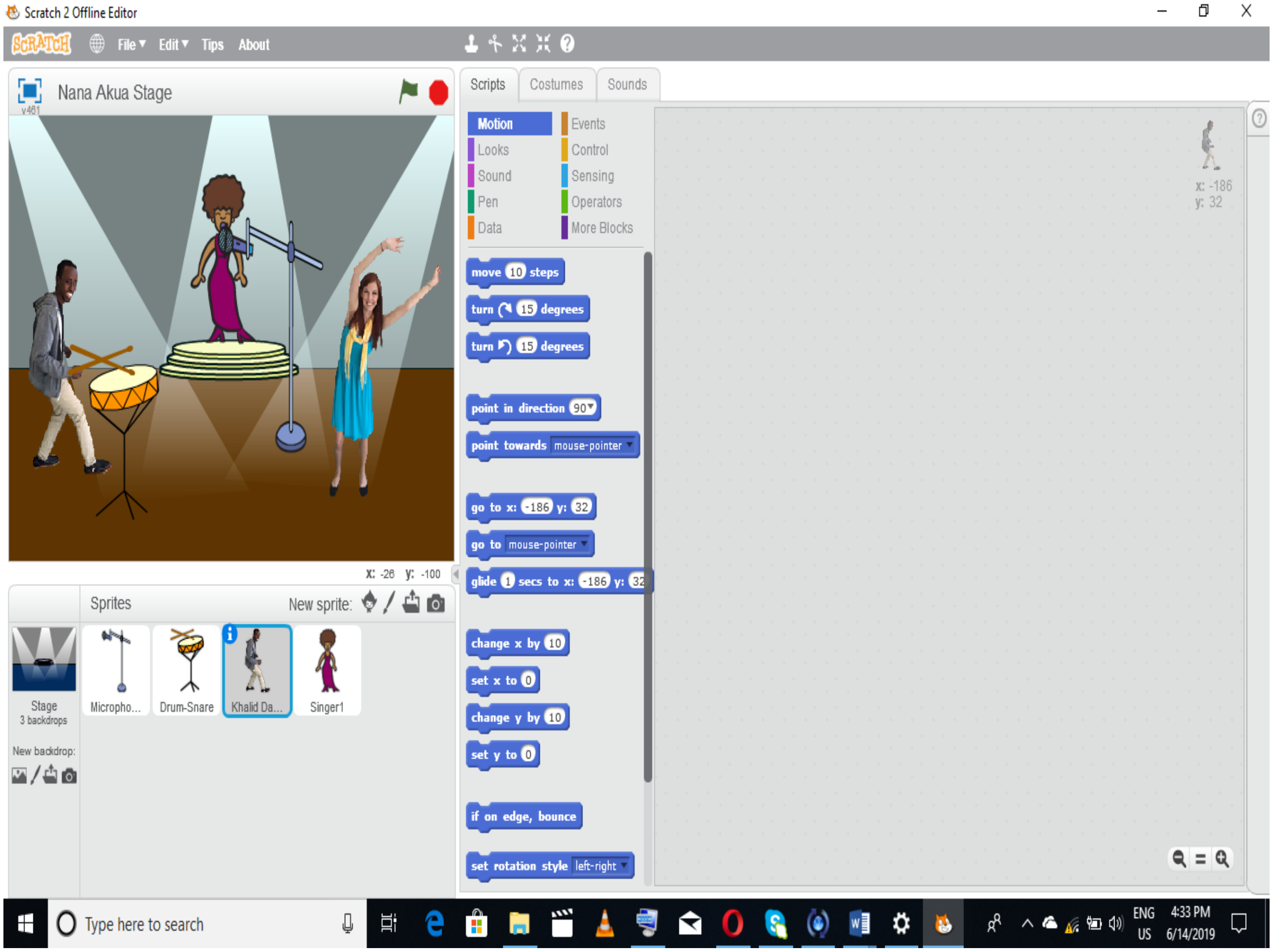


Figure 4.2 Nana Akua Rock star stage design

4.3 Observations

After providing the children with the opportunity of working and making series of designs with scratch program, many of the children were able to program and create simple stories as shown above. They had a lot of confidence and were very optimistic as to the outcome of what they were creating. They interacted well with the scratch environment. Whereas others were afraid to make mistake and therefore was very difficult for them to really express themselves. We observed that, the children as they were getting used to the scratch programming environment, they became more confident and really explored the scratch environment to create very interesting projects.

4.4 Discussion

Many people share a notion that programming is very difficult. Teaching computer programming to children must be taught with programming language that is interactive, among which are difficulties encountered in comprehending programming ideas.

In order to avoid the barriers and encourage children to appreciate and to develop interest in computer coding, the programming language should be very interactive and should make computer programming very learnable.

An effective programming language that is user friendly and interactive to users is the scratch coding environment for children.

According to results of our findings, Scratch programming language had an important impact on students that thought programming was very difficult. The effect of the scratch programming language was evident base on the results.

CHAPTER FIVE

SUMMARY, CONCLUSION AND RECOMMENDATIONS

5.1 Summary

Many students in programming classes might have difficulties in obtaining the necessary skills due to the complexity and difficulty in computer programming. For the purpose of this study, the programming concepts were taught and the results showed that the

Scratch platform seemed to have a significant overall impact on the student’s perception and attitudes towards programming. The members of this project group are convinced that Scratch programming language will help students think creatively, reason systematically and work collaboratively.

5.2 Conclusion

These findings provides in-depth understanding on children desire in ‘’coding’’ with an interactive programming language called Scratch;

- User friendly that is interactive to users.
- Uses visuals to create programs
- Improve children’s ability and understanding on algorithms.
- Loops and functions are easy to apply and understand.
- Improves children ability to be creative and helps children to express themselves using drag and drop method.
- Allows children to develop simple games and storytelling programs, providing potentials to build even semi-complex and complex games.
- Scratch programming aids learning animations and games development programs.

5.3 Recommendation

We are highly optimistic that when children are taught coding with the use of Scratch programming environment it will improve their thinking and understanding abilities and inspire them to pursue computer programming related courses at the tertiary educational level If this project on teaching children coding is implemented throughout the Ghana Education Service curricular, there will be a greater number of software developers and Ghana would be a global competitor in the area of Information Technology. .

REFERENCES.

Andrade, M. (2007) Floresta Magica 2, product manual, Cnotinfor, Coimbra, Portugal. Referenced from the on-line version at http : // arca . imagina . pt / manuais / Floresta % 20 Magica%202.pdf, retrieved on July 30th, 2016.

Blackwell, F. (2014) What is programming? Psychology of Programming Interest Group, 204-218.

Cohen, L., Manion, L., & Morrison, K. (2007). Research methods in education, New York, NY; Routledge.

Dasgupta, S., Hale, W., Monroy –Hernandez, A., & Hill, B. M. (2016) Remixing as a pathway to computational thinking. In Proceedings of CSCW '16. San Francisco, CA, USA, 1438-1449.

Duncan, C., Bell, T., and Tanimoto, S. (2014). Should your 8-year-old learn coding? In Proceedings of the 9th Workshop in Primary and Secondary Computing Education (WiPSCE 2014), 60-69.

Flannery, L. P., Silverman, B., Kazakoff, E. R., Bers, M. U., Bonta, P., and Resnick, M. (2013).

Gunton, T. (1993). Dictionary of information technology. Penguin, London.

Hansen, K., Dwyer, H. A., Hill, C., Iveland, A., Martinez, T., Harlow, D. and Franklin, D. (2015).

Howland, K., and Good, J. (2015) Learning to communicate computationally with Flip: bimodal programming language for game creation. Computers & Education, 80, 224-240.

InfoWorld (1983). Software Reviews: Spinnaker. InfoWorld Vol. 6, No. 33, 56-58.

Kahn, K. (1996) Toon TalkTM – An animated programming environment for children. The Journal of Visual Languages and Computing, April 16, 7(2), 1-23.

Kelleher, C., and Pausch, R. (2007) Using storytelling to motivate programming. Communications of the ACM, 50(7) p. 58-64.

Lister, R. (2011) Concrete and other neo-Piagetian forms of reasoning in the novice programmer.

Morgado. L. (2005) Framework for computer programming in preschool and kindergarten. Ph.D. dissertation, University of Trás-os-Montes and Alto Douro, Portugal.

Pahud, M. (2016) Insights from exploration of engaging technologies to teach reading and writing:

Piaget, J., & Inhelder, B. (1969) The Psychology of the Child. Basic Books.

Rader, C., Brand, C., and Lewis, C. (1997) Degrees of comprehension: Children's understanding of a visual programming environment. Proceedings of the 1997 ACM SIGCHI Conference on Human factors in computing systems, 351-358.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009) Scratch: Programming for all. Communications of the ACM, 52:11 (November 2009), 60-67.

Willingham, D. T. (2004). Ask the cognitive scientist: The privileged status of story. *American Educator*, 28(2), 43 – 45; 51 – 55.

Wing, J. M. (2006) Computational thinking. Communications of the ACM, 49:3, 33-35.