

IMPROVING THE ENROLMENT RATE INTO COMPUTER SCIENCE AT COLLEGES AND UNIVERSITIES THROUGH THE INTEGRATION OF SCRATCH AND ALICE SOFTWARE AT BASIC EDUCATION LEVEL

¹Bashir Idris & ²Ibrahim Gambo Auta

¹*Federal College of Education (Tech), Gusau Waziri Umaru Federal Polytechnic Birnin Kebbi*

²*Department of Computer Science, Waziri Umar Federal Polytechnic, Birnin Kebbi*
Email: Edrisbashir@gmail.com, ockanya@gmail.com

ABSTRACT

It is gradually becoming an imperative for most people to have programming skills to aid them accomplish their daily jobs ranging from medicine to basic science, business and finance. Despite of this rapid growing need in computer programming profession the number of student enrolment into computer science program drastically reduced every year. This happened as a result of student having no prior knowledge of any programming language. Having no skill entails having less interest to take up the course as a carrier subject of interest. Scratch and Alice are part of the recent solution to these obstacles. This paper investigate how Scratch and Alice contributed in shaping the programming skills for kids using none-code or graphics programming in the form of drag-and-drop manner there by making the kid intrinsically and cognitively programming oriented.

INTRODUCTION

With the ultimate cognizance made by the broader need of participation into computing, the number of students enrolment into computer science programme drastically decrease with every successive year. Research indicated statistically that, the rate of the decrease among female is more staggering (Lopez, 2007).

Complaints were received constantly from the instructors dealing directly with subject related to programming as well as those people organizing activities related programming contest on the drawback to partake in computer related fields. It was also noticed that, the rate at which students enrol willingly into IT and robotics programme is overwhelmed compared to programming (Idlbi, 2009). Many students ran away from university or change from one department to the other as a result of programming forbear or difficulty. Other obstacles are cultural factors that influence decisions to pursue computer science especially in girls which are very difficult to correct, but the process of providing simple ways for learning how to program (eg. Scratch and Alice) can be a motivating gateway to the study of computer programming (Kelleher, Pausch, & Kiesler, 2007).

Many students chose computer science at university most often due to their curiosity to study the leading technology, which they ultimately believe will lead them to a brighter future. But what suddenly broke their illusion was programming which is mostly viewed as an abstract concept (Choi, Cairncross, & Kalganova, 2001). Also stress that, Students with less motivational ability to learning or low self-esteem could easily be discouraged and may eventually be disengaged from the course if cannot improve to some certain level of their satisfaction after a while (Idris, 2013).

These series of obstacles that lead to so many deterrents in the field of computing has been a concerned issue across so many researches to come up with a way that kids could be brought up with low programming barrier. This leads to the creation of so many multimedia programs that was used by many instructors or teachers to teach kids programming through simple drag and drop format.

This paper investigate how Scratch and Alice changes the kids programming experience to be more interesting, more engaging

and more friendly to the kids which however, may become a pathway to producing programming set-minded youngsters. Scratch was developed by Lifelong Kindergarten group at MIT's Media Lab. It has rich programming environment that lets users create their own interactive stories, animation, games, music and interactive art (Malan & Leitner, 2007). Alice on the other hand was developed in 2002 by Professor Randy Pausch and his team at Carnegie Mellon University as an introductory learning tool with a simple and a friendly drag-and-drop interface (Lopez, 2007)

Introducing Scratch

In collaboration with Yasmin Kafai's group at UCLA, Scratch was design and developed by Lifelong Kindergarten group at MIT's Media Laboratory. although scratch was not the only programming that aimed at novice, it was specifically design to targeted children at age 10-18 from economically advantaged and culturally diverse communities (J. Maloney et al., 2004). As cited in (J. H. Maloney, Peppler, Kafai, Resnick, & Rusk, 2008) there are a number of other programming that are primarily develop for developing youth skills and confidence to become future programmers. But scratch invariably differ in a number of ways, one of which was limiting the code related programming aspect or the syntactical feeling into drag and drop format. This approach systematically eliminate syntax error handling by users (which normally is the major problem of programming), allowing children to dwell directly into solving their problems by manipulating images, animations, movies and sounds.

Moreover, Scratch is design with core features that includes among others 1) graphical blocks that are easily mouse-dragged onto the script pane in a syntactically-correct way 2) programmable manipulation of rich media which allow all multimedia elements to be manipulated 3) multiple language

support which alternatively support dynamic switching from one language to the other (Idlbi, 2009)

Scratch Design and Features

The scratch interface is such a very user friendly that comprises four interacting phases. These include palette, scripts, stage and the sprites. The programming is done by simply dragging the block of command from the palette into the script pane while the action triggered by the command appeared on the stage. These multiple commands are assembled on the scripting pane there by creating a "stack" of blocks. Double clicking on each stack triggered the run command to execute (Maloney et al., 2008). Below is an example of a scratch scenario.

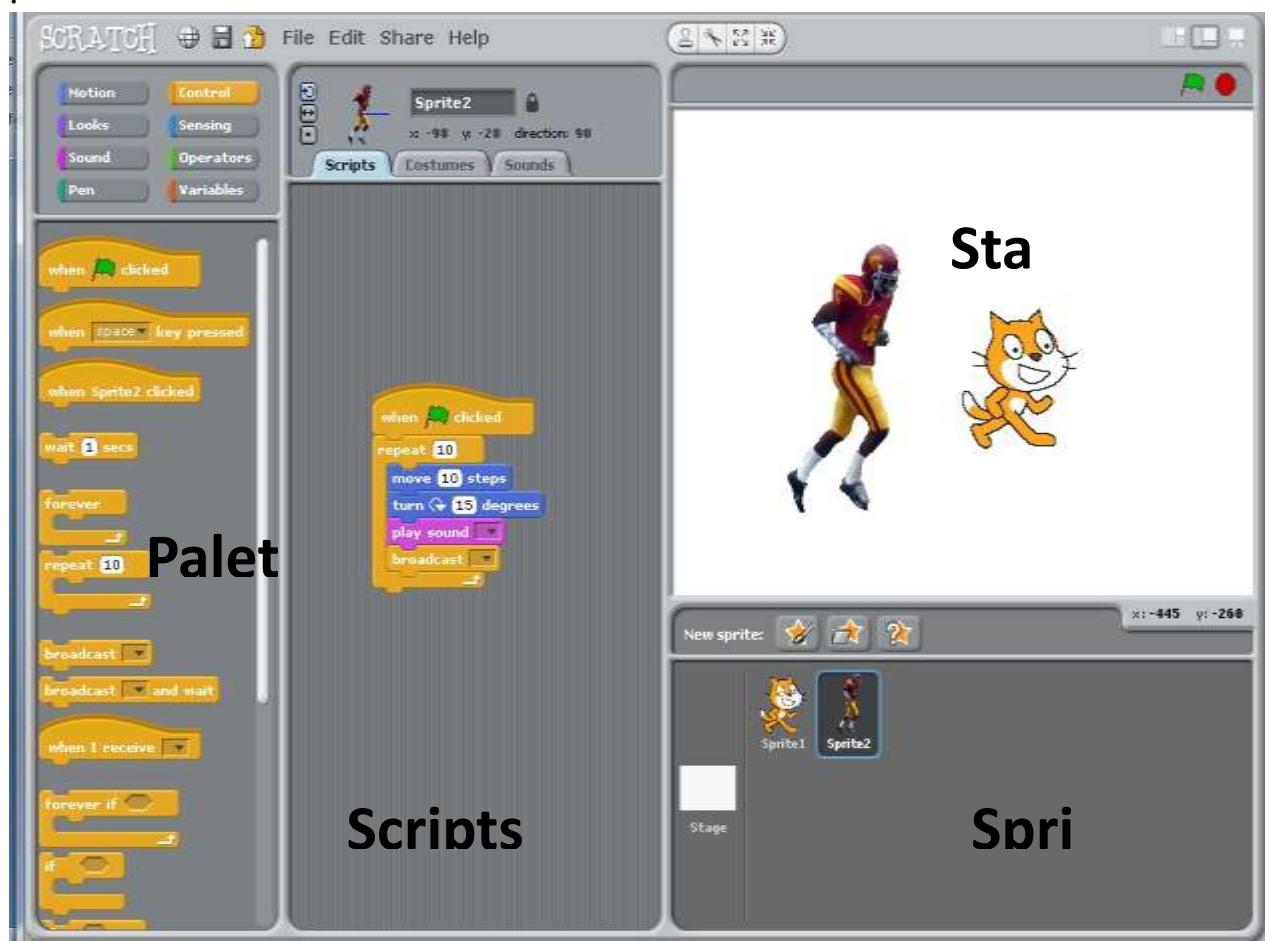


Figure 1 Screenshot of Scratch Interface

The scratch interface is divided into four main areas (palette, scripts, stage and sprites). Figure 1 illustrated these four main areas. It is based on the building block metaphor where learners build scripts by snapping together graphical block muck like pieces in a jigsaw puzzle (Maloney et al., 2004).

Like all other programming languages scratch has a number of control structures of If's ,loops and event handlers which are graphically represented in scratch by things like "when clicked, when key-pressed, forever if, repeat-until" and communication between the block of stacks with different control structures is done via the broadcast command. One broadcast can trigger multiple sprites to run.



Figure 2 Screenshot of an interactive scratch project by users (J. Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010)

Introducing Alice:

Alice was developed in 2002 by Professor Randy Pausch and his team, the stage 3 Research group at Carnegie Mellon University as an introductory learning tool with a simple and a friendly drag-and-drop interface. It is a 3-D Interactive Programming

Environment that is aimed at making programming for novice easier. Alice is more or less an advancement of scratch; it is more fun and interesting because of the 3-D graphical application. Alice is primarily a scripting and prototyping for 3-D object behaviour that demonstrate a virtual world that include images such as animals, vehicles etc.(Lopez, 2007)

One of the challenging things that always frustrate programmers is having program for a whole day or night but ended up not working due to a syntax error. Debugging can possibly take another time same as that used for the coding. Alice was design to incorporate object oriented techniques. Moreover it do not require the user to make any syntax debugging throughout the programming. During script execution the objects responds to user input through keyboard and mouse. Each object is animated accordingly over a specified duration in speed and time(Cooper, Dann, & Pausch, 2000).

Alice Design and features

Alice like scratch has five main areas which include; Object tree, detail area, world window, editor area and event area (see fig. 3).

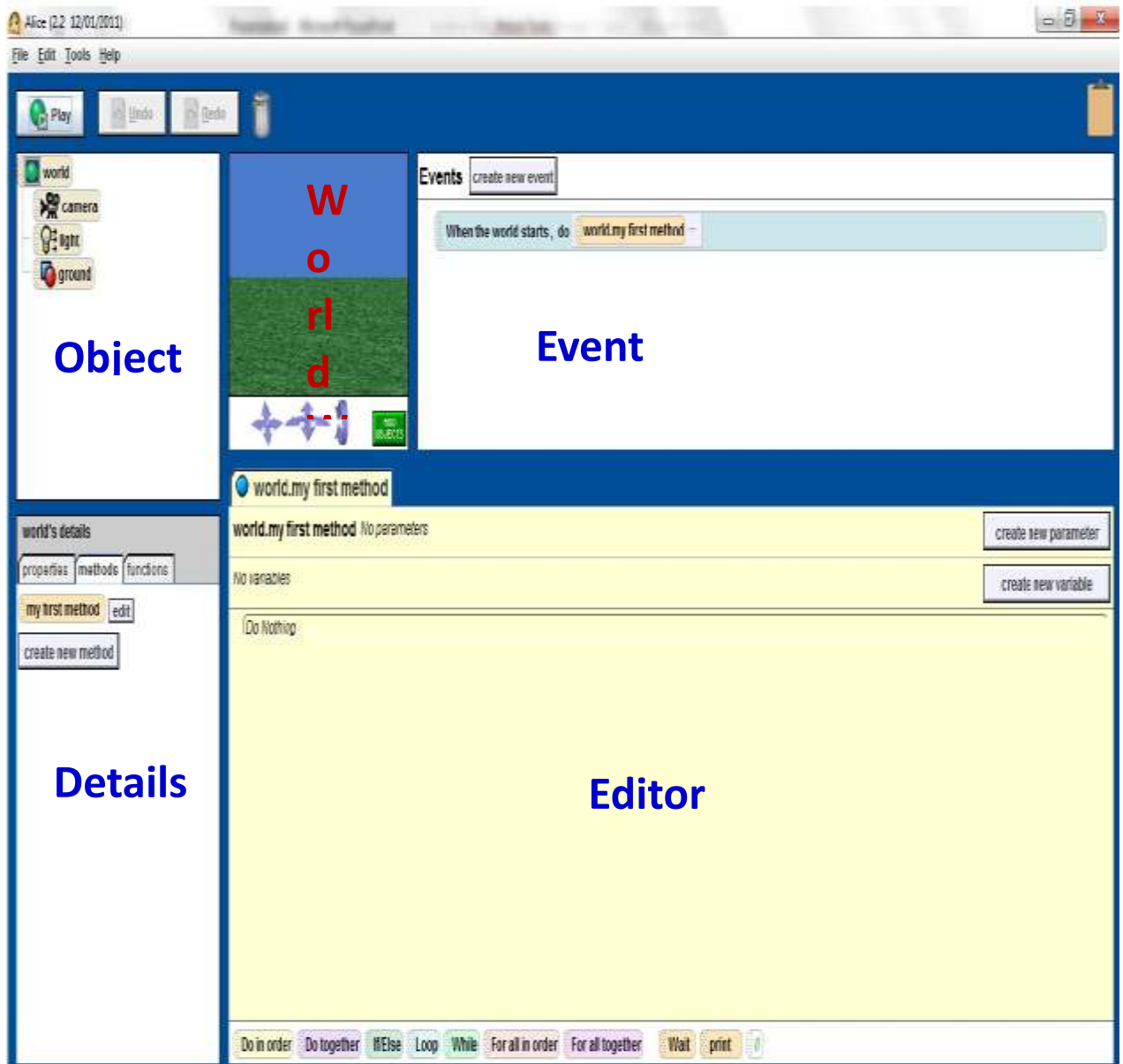


Figure 3 screenshot of Alice Interface

Alice as a 3-D application provides several built-in action commands which are subdivided into two categories (Cooper et al., 2000). The first part instructs objects to perform a motion within the world while the second part provides changes to the physical nature of an object. Users add objects from an object gallery by clicking the add object button which automatically linked the user to the Alice library that contains hundreds of low-

polygon models whose high fidelity comes from carefully hand-painted texture map (Conway, Audia, Burnette, Cosgrove, & Christiansen, 2000).

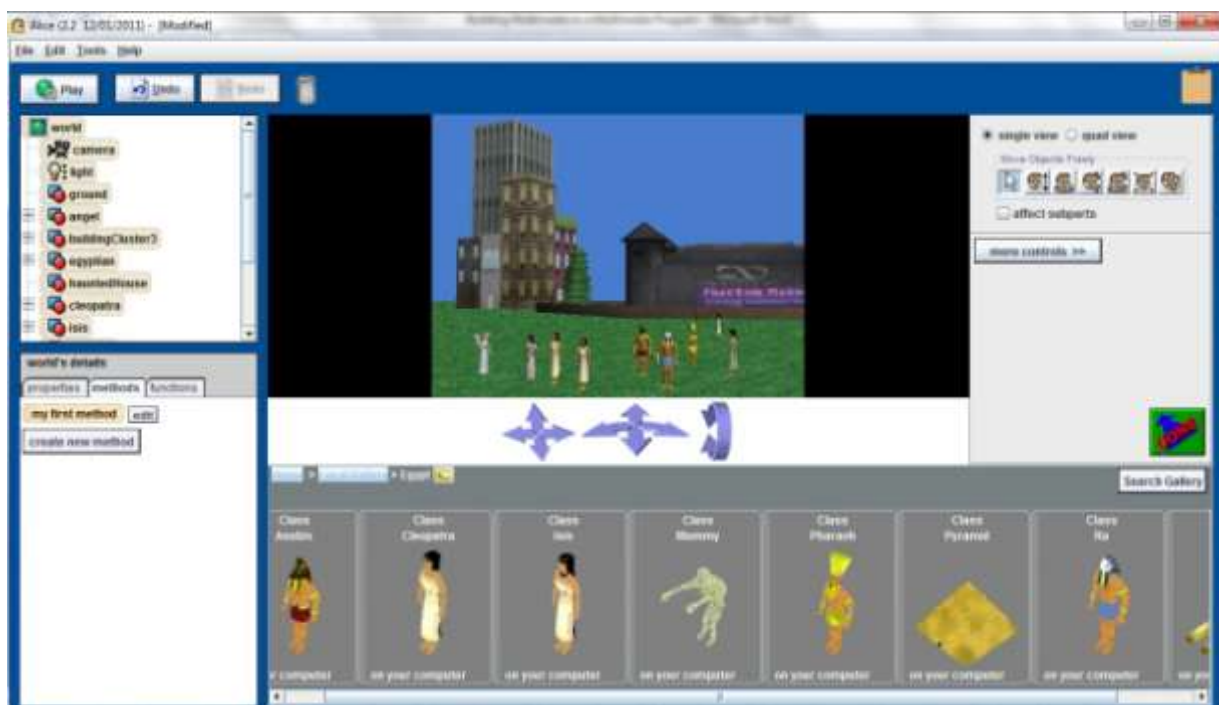


Figure 4 Screenshot of Alice Add object pane with objects added on the world

The Rational Behind Drag-and-Drop Object Oriented Programming:

As a matter of fact, the main reason why programmers strive more effort to elicit youths interest into programming is to bridge the gap in the computer field backsliding. Computing and Information Technology are among the fastest growing professions in the world but programming as one of its crucial components is making so many people to decline in taking it as their own career course in colleges and Universities. In 2007 a study by Lopez reported that:

"The demand for computer scientists is growing much more rapidly than schools are producing qualified candidates to fill the growing industrial need. Interest in computer science as a

major is declining among incoming students nationwide; the percentage of college freshmen that listed computer science as their most likely major dropped 70 percent from 2000 to 2004”

That necessitated the introduction of such programming languages as Scratch and Alice to enable kids and younger adults develop their mental capability to become creative thinkers and problem solvers before going to higher academic level.

CONCLUSION

Research indicated that, with the ultimate cognizance made by the broader need of participation into computing, the number of students enrolment into computer science programme drastically decrease with every successive year. This obstacle lead to the creation of so many different multimedia programs that help influenced youth to learn how to program without the syntax forbear.

Alice provides a 3-D animated programming environment that supports one approach for teaching problem solving in the particular way peculiar to the normal programming. Students were able to see what they do know what went wrong in their programs and easily debug and correct them by a simple drag-and-drop.

A Scratch programming on the other hand strives to help users build intuitions about computer programming as they try to create projects that engage their interests. On the user interface layout, the user manipulates various commands between the palette and central scripting area. User creates blocks of stack each of which is control by a broadcast. A loosely-coupled inter-object communication is maintained in the system which allows

sprites to be exchanged without breaking dependencies, fostering collaboration and code sharing.

By so doing, many researchers has empirically proved that, the syntax bugs in programming which became the soul problem in decreasing the number of enrolment into computer science especially those course that has direct relevance to programming are concealed behind the scene, thereby giving the direct relevance of problem solving skills to users, which gradually change their mental image of the way they look at programming (as a difficult task) and eventually dwell into it, as well may make it as choice of interest as their major life career.

RECOMMENDATIONS

The researchers finally recommend that:

1. Programming courses that will make use of such simple programming techniques should be introduced into primary and secondary school levels and be made compulsory for all learners
2. Students whose marks are higher in programming courses should be given priority for admission into computer science courses.
3. Government should introduce a scholarship scheme for computer programmers with a good remuneration.

REFERENCES

- Choi, S., Cairncross, S., & Kalganova, T. (2001). Use interactive multimedia to improve your programming course.
- Conway, M., Audia, S., Burnette, T., Cosgrove, D., & Christiansen, K. (2000). *Alice: lessons learned from building a 3D system for novices*.
- Cooper, S., Dann, W., & Pausch, R. (2000). *Alice: a 3-D tool for introductory programming concepts*.

- Idlbi, A. (2009). Taking kids into programming (contests) with Scratch. *Olympiads in Informatics*, 3, 17-25.
- Idris, B. (2013). *Online Games For Digital Game-Based Learning (Dgbl):An Emperical Invetigation on Students Perception in Accepting DGBL as Learning Tool*. Master, Multimedia University.
- Kelleher, C., Pausch, R., & Kiesler, S. (2007). *Storytelling alice motivates middle school girls to learn computer programming*.
- Lopez, M. R. (2007). *Use of ALICE programming language to aid reading comprehension and to introduce computer science skills in elementary school students*. CRA-DMP Intern, Summer 2007. Computer Science Duke University.
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. *ACM SIGCSE Bulletin*, 39(1), 223-227.
- Maloney, J., Burd, L., Kafai, Y., Rusk, N., Silverman, B., & Resnick, M. (2004). *Scratch: a sneak preview [education]*.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16.
- Maloney, J. H., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: urban youth learning programming with scratch. *ACM SIGCSE Bulletin*, 40(1), 367-371.

Reference to this paper should be made as follows: Bashir Idris & Ibrahim Gambo Auta
(2018) Improving the Enrolment Rate into Computer Science at Colleges and Universities through the Integration of Scratch and Alice Software at Basic Education Level. *J. of Education and Leadership Development Vol. 10, No. 1, Pp. 49-60*
