The Impact of using 3D Interactive Animation Tool in Teaching Computer Programming at the Senior High School Level

Najim Ussiph, PhD.  
Kwame Nkrumah University of Science and Technology  
Kumasi, Ghana

Hamidu Kwame Seidu  
Akim State University College  
Akim Oda, Ghana

ABSTRACT
A quasi experiment with interview was adopted to study the aptness of using 3D animations as an instructional method to introduce programming concepts to students at the Senior High School level. This research work was conducted with 100 students of Akroso Senior High School in the Birim central municipality of the eastern region of Ghana who were generally programming novice. Programming concepts considered included programming environments, loops, functions sequential and conditional execution of programs. A paired t-test carried out on the results of the performance test presents a p-value of 0.008 indicative of a numerically significant difference between the mean marks of participants during the experiments that used 3D animation method as against the experiments that used the text base method. Results from the interview showed that the instructional method used had impact on the performance of the learners. The use of 3D animation method presented programming concepts in a form that the learners can understand, motivates them to pursue programming related courses at a higher level and also impacts positively on their performance.

Keywords
Computer programming, Game based learning, Human computer interaction Program visualization, Virtual learning environment, 3D Animation.

1. INTRODUCTION
In the year 2008, Ministry of Education (MoE) and the Ghana Education Service (GES) introduced elective Information and Communication Technology (ICT) course into the Senior High School (SHS) curriculum. The course was intended to give students the needed opportunity to “acquire knowledge and skills to operate effectively at more than a basic level of competence which will enable them pursue further ICT course at the tertiary level or enter the job market”[32]. The course is broken into sections with each of them being broken further down into units. Topics under two or three units are taken for a term. One of the sections to be taken by students at this SHS level is introduction to computer programming which is a fundamental component of every information technology curriculum.

The objective of computer programming is to find a sequence of commands that will mechanize the computer in performing specific task or solving a given problem which is usually facilitated by a programming language such as Java, Basic, Python, C, C++, C# etc. According to the general objectives of the syllabus, students are to acquire the basic skills needed to begin the development of computer programs. It is considered as one of the most difficult and detested component by many beginner computer science students who, having faced difficulties in a programming component, often fallout from the course altogether. There has been numerous research work [26] [41] [1] that attempted to make bare the causes of such difficulties in learning programming. Unfortunately, most of these researches have failed to address the teaching of the course at the SHS level and instructors still teach programming as it was done so many decades back resulting in students getting frustrated when they first learn about programming. The elective ICT course afforded an opportunity to lay the needed foundation of the aspects of programming that are required to undertake a major program in either computer science or information technology at the tertiary level.

Rigid Syntax and dealing with syntax errors is also a major problem to students. Some students may have full control over the syntax and semantics of the concepts but the text based output generated by the program devoid graphics or multimedia which students of this generation want to interact with, resulting in students mounting the perception that programming is boring and non-interactive. The use of static traditional teaching methods is often passive; they do not engage the computing student in active learning of programming within a real world context. The tight spot was how to present these concepts in such a way that they can be remembered and used by the students for the programming process.

It is against this background that the we find the study into the problem worth considering to help suggest alternate teaching approach of introducing students at the SHS level to concepts of computer programming to eliminate this phobia for programming to achieve the objective of which the Ministry of Education (MoE) and Ghana Education Service (GES) introduced the elective ICT course into the Ghanaian senior high school curriculum.

The objective of this research is to encourage the best pedagogical method for effective teaching and learning of programming through the use of 3 dimensional (3D) interactive animations using visualization and animation tools. This is to minimize the difficulties in learning programming concepts and also the methods of programming by providing visual and interactive learning environments which presents a visual appeal of 3D animations for current generation of visually-oriented students.
1.1 Research Objectives
The objectives of this research were:

1. To present the abstract concepts of programming in a form that can be easily remembered and used by novice students.
2. To investigate the effectiveness of using interactive 3D animation as an alternative introductory programming approach at the beginner level.
3. To motivate students to learn computer programming by creating 3D animated products to visualize programming concepts.

This paper has been organized as follows: section 2 presents discussions on previous studies conducted in areas of the subject under discussion. Section 3 presents the design whose results are presented in section 4. The results are then discussed in section 5, and finally conclusions are presented in section 6 and recommendations made in section 7.

2. LITERATURE REVIEW
This section discusses previous studies under the following sub-headings: Pedagogy of Programming, effects of human computer interaction on learning programming, causes of difficulties in learning programming, using program visualization as instructional aide, programming Learning Support systems for creating animations, programming in 3D Virtual Learning Environments (3D - VLEs) and Game Based Learning (GBL).

2.1 Pedagogy of Programming
This section focuses on discussions relating to how programming has been taught over the years and their impact on learning outcomes. This study concentrated on three major approaches which include the use of the traditional text base lecture method, web based delivery and the use of program visualization tools.

2.1.1 Traditional Text Base Lecture Method
The unrelenting recognition of the lecture-based teaching method in the midst of programming instructors has been accredited to its ease of use. [39] conducted a survey to establish the efficacy of various methods of teaching students at the graduate level. Majority of the respondents reported lecture method as the paramount teaching method citing the reason of being able to provide all knowledge related to a topic. Critics of this approach argue for it to be replaced with an interactive teaching method that afford students the opportunity to meaningfully interact with academic subjects. Many educators [45] [27] in the computing environment shares this view contending that lecture methods are ill-suited for nurturing highly classified cognitive skills including programming concepts and also do not excel in developing the skills of application which is the bedrock of learning programming.

2.1.2 Web-Based Teaching of Programming
Web-based teaching of programming has turn out to be an admired and essential issue in recent years and the rapid increase of various applications reveals its importance. [5] conducted an experiment to examine an interactive and concerted virtual teaching environment that has been created by supporting Moodle Learning Management System (LMS) with collaborative learning tool GREWPtool. The focus of this investigational study has been to ascertain the achievement rate of students when using the collaborative tool in teaching programming languages over the internet. They reported a superior achievement rate when an LMS system is used together with a higher collaborative tool during the teaching of programming languages in a Web-based environment. [18] recommended a web-based programming assisted system (WPAS) which offered learners the leverage to code and debug online arguing that the web based system supports learners to develop their cognition abilities in web-based programming.

2.1.3 Teaching Programming Via Visualization Tools
In their study, [26] reports on the use of visualization tools on learning programming concepts; an investigation conducted using university students in their first programming course to study the effectiveness of visualization tools in learning programming. Whereas the control group used only traditional text based material during the session, the treatment group, used the same material with a visualization system as an additional tool. They concluded that program visualization systems augment the learning of novice programming students significantly.

It must therefore be noted that, a choice of any approach to teaching especially to novices raises a number of methodological issues that the instructor must consider. This includes the concept to be taught, time constraint, number of students to be taught and availability of suitable teaching and learning aids.

2.2 Effect of Human Computer Interaction On Learning Programming
Human Computer Interaction is an area of study concerned with techniques that use both interaction styles and devices and the procedures that programmers use in creating a user interfaces as well as tools in designing, building, and evaluating such user interfaces. Every virtual environment (VE) application thrive on a well-designed three dimensional user interface. [23] offer a general idea of the modern developments in 3-D user interfaces and interaction. They presented some key interacting techniques and interfaces created purposely for 3-D systems. [23] discussed navigation, selection/manipulation interaction techniques as well as a more general interaction tasks system control referring to it as an interaction involving user and the system which is not part of the virtual environment. [19] investigated utilization of two common mouse interaction styles by children; drag-and-drop and point-and-click, to establish whether the preference of interaction techniques affects the performance of children in interactive learning environments. The interaction techniques were evaluated to establish whether either method was better to the other with respect to speed, error rate, or user favorite. His conclusion reports that the point-and-click interaction technique was faster, resulted in lesser errors generated in using it; and it was chosen by users over the drag-and-drop interaction technique. There are publications showing promising merits of dedicated 3D input devices over generally known desktop environment. For instance, [25] in their work sought to compare the interface of keyboard and mouse to SpaceTraveller and Globefish which are bi-manual environment which makes use of the 3D input devices in a spatial orientation. The tasks to be performed requires navigation in egocentric and exocentric viewpoint. Their conclusions divulged that both interface design performed likewise with reverence to time used in completion of task; nevertheless, the bi-manual techniques produced a considerably less inaccuracies.
3.2 Causes of Difficulties in Learning Programming

Teachers’ teaching approach usually fail to support the learning styles of all their students. A problem, [22] opine that it is a critical duty for the teacher to make sure that the students acquire the most appropriate learning method for the subject under consideration. In an experiment conducted by [45], students were given a learning method preference test and then their chosen learning methods were compared to their performance score and the practical programming element of the introductory programming. They reported a considerable difference in performance between the groups of students and concluded that that some learning styles of students are more suitable to learning programming than others.

Lack of generic problem solving skills by students has also been cited. [28] studied the relationship between students who have a flimsy grasp of both fundamental programming principles and the aptitude to thoroughly carry out routine programming tasks by examining students in two separate experiments. First, students were examined on their ability to guess the outcome of executing a program code. In the second experiment, students were examined on their skill, when given the desired function of a near complete program code, to choose the correct conclusion of the code from a small set of options. The study concluded that many students showed weakness at these tasks, especially the latter task, indicating that such students have a weak skill that are a requirement for problem solving.

In a related work, [12] in the Superior Institute of Engineering of the Polytechnic of Coimbra was essentially to identify the main breaches in mathematics and to analyse their influence on programming capacity. The research reports that students had complications in quite a few areas of mathematics. The research reported that the lack of programming skills was directly caused by deep deficiency in mathematical knowledge. Their assertion that having a strong mathematical background is very important for success in programming is confirmed the work [3] which evidenced some connection between programming skills and mathematics experience.

2.4 Using Program Visualization as Instructional Aides

[37] defined Program visualization (PV) as “a research area that studies ways of visually assisting learners in understanding behavior of programs”. The major objective of visualizing programs is to augment understanding of students in different areas of program execution. [35] identified two main approaches to visualization in support of teaching programming: algorithm animation which aims at illustrating how an algorithm works and program visualization, which is programming language dependent and focuses on animating source code execution. A number of research literature have also introduced visualization as a suitable means for teaching introductory programming.

In recent years, there have been the development of numerous PV tools which has aided the teaching and learning of programming. However, these tools differ in characteristics and potentials. [11] presented a meticulous breakdown of these tools and grouped them into four categories as follows: (1) tools that include a simple and reduced development environment; (2) example-based environments; (3) tools based on visualization and animation; and (4) simulation environments. A similar work by [16] executed an algorithm development and visualization model named “What You See Is What You Code” for programming beginners, which presented programming lessons using a crucial model. A thorough scrutiny of its learnability demonstrates that their model helps programming beginners to swiftly discover errors and debug them to build up error free program code.

On their part, [9] explores the use of PV tools as a contemporary scientific approach towards achieving the teaching objectives successfully. Their work submits and compares the findings of teaching two consecutive programming courses using the traditional text based approach versus the PV tools’ approach. The results of their work reports a major progress of students taught using the PV. Their study confirms the result of other researches that PV tools are very important in the teaching of programming. In order to minimize the impact of cognitive burden on users of such learning tool, [26] suggest that they should be properly familiarized with it in advance. In their study conducted on the effects of cognitive load in using a program visualization tool, reports that the students with prior knowledge of the tool learned appreciably better. They therefore concluded that to get the most benefit of a visualization tool, the students should be advised to use it effectively and that the prior knowledge of users should be considered when examining the effectiveness of such tools.

The findings reported by [44] in his quasi experiment showed that visualization of programming concepts has serious repercussions on how programming teaching tools are designed and developed indicating that the impact of such dynamic program tools have on the teaching and learning of programming. His findings fell in line with that of previous researches including [37], [33], [14] on the effects of PV methods on the learning performance of programming students. Even though the conclusion of [44] indicated that “Example-Based Dynamic Program Visualization Environment (EDPVE) is effective for students at the tertiary level, he suggested additional study to assess its effectiveness on the learning outcomes of students at the pre-tertiary level whether it is effective for high school students.

2.5 Programming Learning Support Systems for Creating Animation.

In the early 1980’s, most researches in teaching programming resulted in the development of and use of several programming systems. The user friendly programming system LOGO, which was created for children by [34] uses an egocentric coordinate system and turtle graphics to teach children the mathematical geometry concepts. Nevertheless, Papert identified variables and debugging as the two concepts that caused troubles for beginning programmers using LOGO. In recent years, numerous applications have been created to make programming instruction a little easier. For the purpose of this study, three of such advanced applications will be discussed - Greenfoot, Scratch and Alice.

2.5.1 Greenfoot

Greenfoot was designed by a Programming Education Tools Group which is an affiliate of the Computing Education Research Group at the School of Computing, University of Kent in Canterbury, UK. It is an Integrated Development Environment (IDE) aimed at assisting students of age 14 years and above to become skilled at programming concepts using java language. In their research to evaluate the effectiveness of using Greenfoot to stimulate interest in computing, [8] used this material to teach computing related courses. Their
assessment reports a major increase in knowledge acquisition in computer related courses and they concluded that it motivated students to take up computer science related courses in the future.

2.5.2 Alice
Alice is developed by Carnegie Melon University in Australia to help expose students to programming concepts in object-oriented programming. This 3D programming environment guides learners to study computer programming allowing them to produce animation based stories or video games which can be shared with others. [40] present results of an experiment that was conducted within Cairo University in Egypt to measure the effectiveness of teaching programming concepts using Alice. They argued that using Alice programming tool to teach college students is best placed before students enter the university as a summer preparation course or as part of the introduction to computer science course with a course project to maximize the benefit of Alice when students have no previous exposure to computer programming. However, critics of this application, raises some pedagogical issues. [36] reports on their experiences incorporating Alice in the classroom. They indicated that even though Alice has been revealed as increasing self-belief and understanding of students at the tertiary level, their experiences at Tufts University in Medford, USA, have established some instructive drawbacks to the method. They argue that Alice creates misconceptions in the minds of students which can be detrimental when these same students change over to more professional languages such as C++ or Java.

2.5.3 Scratch
Scratch is a desktop and online multimedia programming environment meant to create animations and provide the platform for more professional programming. [30] report on the use of the Scratch environment by students aged between 8 to 18 years over a period of 18 months. Their analyses of data gathered through their experiment reported successful implementation of scratch indicating that the codes designed using Scratch blocks make simpler the technicalities of writing programs by avoiding the violation of programming rules. In a similar development, [31] examine whether Scratch can be used to teach computer related courses. Even though there were some shortcomings regarding initialization and variables, the results of their work demonstrated that in a broad-spectrum learner could implement programming concepts using scratch.

2.6 Programming in 3-Dimensional Virtual Learning Environments (3D VLES)
There have been a number of researches that acknowledges the use of virtual worlds at the tertiary level. [7] investigated the feasibility of novice students learning computer programming better inside virtual world than through traditional text based methods. Results of their study supports the argument that the use of virtual worlds results in effective learning of programming. [46] agreed, pointing out that virtual world is currently the most mature and popular platform that has been used for this purpose. However, there has not been a clear guideline for its practice. Some researches in this field has indicated that the mere use of visualization system to teach programming does not result in successful teaching and learning [16], since it may result in a situation where students over concentrate on the visualized element and not appreciating what they communicate. A position, [42] share arguing that visualizations has come to stay and its positive impact will only be felt when it actively engages the student rather than make them spectators in the learning process. Enquiries of researches to date advocate that the case for how these technologies support learning remains unclear, with the learning outcomes achieved in projects in this area being discussed in very generic terms [21]. As indicated by [6], more work is needed to bring the virtual world or games development and before education communities closer together and researchers must seek which combinations of pedagogic strategies and tools best target the desired objectives.

2.7 Game-Based Learning Approach
In view, [15] presents a form of reasoning that a text base lecture approach to teaching and learning "programming is like a very nutritional soup, good for you but lacks taste. Game scenarios provide the spice for our soup that is not only good for the students but makes the soup more delicious and attracts students to try it again and again". Educational games that serve academic purposes are pertinent for being used in computer programming education because of the interactive graphics they provide which encourage students to learn computer programming.

[10] partition instructional games into three which became popularly known as the three generations of educational computer games model. In the first generation mainly involves drill and practice games using a behaviorist approach to condition the player, while the second generation games enhance the motivation of the learner by incorporating player attention, focus, curiosity and fantasy. They require the player to develop deeper understanding in accordance with the cognitivist approach. The third generation of instructional games are based on a constructivist approach, meaning the player creates knowledge through experiences and interaction with social communities.

[20] opine that by having to experience the learning process via game plays, knowledge can be constructed rather quickly, as the learners will be actively exploring and trying out activities within the game. Hence, information can be passed using various ways in comparison to the traditional textbook learning which can be boring and very passive.

[38] experimented game designed for novices to learn data structure concepts. The result showed that the use of games had influence on the students and they agreed that the game is interesting and has helped them to unveil the stack concept in data structure when learning programming. They conclude by pointing out that, learners are usually immersed in computer games due to the visual environment, where they are encouraged to make decision and choices, as well as reflecting on the decisions made.

An investigation conducted by [4] on the effect of game development in introductory programming course on students' performance reports that students who had taken the computer game development course became more successful. These findings support the position held by [43], who puts out that games improved students' performance in relation to students' course performance in previous years.

[29] present an alternative method for teaching programming using an online debugging game, Gidget. They argue that debugging games approach avoids the problem where learners need a large amount of prior debugging knowledge before they can begin creating their own programs adding that adopting a debugging-first method encourages learners to learn without an instructor. [47] in their study were interested in looking at how children can be taught computer programming concepts
through game construction. Game construction tools such as Scratch, Kodu and Gamemaker are available for children as young as 7 to create their own games. Their work evaluated the Scratch software (game programming tool) and its effectiveness for teaching children about programming. On completion of the project, there were some improvements in learning with Scratch. The result of the study was not generalized because it was situational and did not include learners from different age categories to make a more general conclusion.

2.8 Conceptual Framework

From the various literature reviewed, the method of instruction used has influence on the performance, interest and motivation of students in pursuing computer programming courses. The comparisons between these teaching methods determines the performance and motivation of students in programming courses.

3. RESEARCH DESIGN

This section presents the analysis and findings of the research design used in the various practical activities that were undertaken in an effort to achieve the aims and objectives of this research work.

3.3 Implementation of Research Design

This section presents a summary of the various practical activities that were undertaken in an effort to achieve the aims and objectives of this research work.

3.3.1 Procedure

Table one (1) below shows a breakdown of the framework indicating a laboratory session over a period of eight (8) weeks.

<table>
<thead>
<tr>
<th>LABORATORY SESSIONS</th>
<th>TOPICS TAUGHT DURING EXPERIMENTS ONE (TEXT METHOD) AND TWO (3D ANIMATION METHOD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Introduction to environment/interface</td>
</tr>
<tr>
<td>2</td>
<td>Programming commands</td>
</tr>
<tr>
<td>3</td>
<td>Sequential execution of programs</td>
</tr>
<tr>
<td>4</td>
<td>Conditional execution of programs</td>
</tr>
<tr>
<td>5</td>
<td>Looping</td>
</tr>
<tr>
<td>6</td>
<td>Functions</td>
</tr>
<tr>
<td>7</td>
<td><strong>PERFORMANCE TESTS</strong></td>
</tr>
</tbody>
</table>

Table 1: Programming concepts taught during the quasi experiments.

3.3.2 Experimental Design and Implementation

A quasi experiment was used in this study to assess the impact of using 3D animations on participants’ comprehension of programming concepts. There were two (2) experiments which was sequentially executed involving three participant groups. Experiment one (1) involved the use of text base method using Microsoft Basic followed by experiment two (2) which involved the use of the 3D animation method using Alice programming environment. Each participant group was taken through the laboratory sessions of the two (2) experiments. In the course of the experiments, each participant group received instruction one hour twice a week for 8 weeks. After each experiment, participants' understanding of the concepts discussed was assessed through a performance test. The results of the performance test were subjected to t-test analysis and presented in the next section.

4. DATA ANALYSIS AND FINDINGS

This section presents the analysis and findings of the research based on the data collected during the two (2) experiments. Data collected included the time spent by participants in performing laboratory activities, scores of participants in the
performance test, responses of participants to the interview questions after the experiments and observations made by the researcher. These data collected from both experiments were compared to assess the impact of the two instructional methods used in the experiments.

4.1 Comparison of findings to assess the impact of the instructional methods

The assessment of the two (2) instructional methods focused on the main parameters that have direct relationship with the research questions - student performance, ease of use/learnability of method, and student motivation. The following subsections presents and analyzed the data with respect to these parameters.

4.1.1 Student Performance

This sub-section presents t-test analysis on the performance test conducted at the end of the experiments and also responses of participants as to whether the method of instruction had any influence on their performance.

4.1.1.1 Paired-Samples T-Test for Performance test

The performance test administered at the end of the experiments was subjected to a paired-sample t-test. The outcome of the paired t-test as presented by Minitab application output is displayed in Table 2 below.

<table>
<thead>
<tr>
<th></th>
<th>N</th>
<th>Mean</th>
<th>StDev</th>
<th>SE Mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Post-test</td>
<td>100</td>
<td>6.530</td>
<td>1.494</td>
<td>0.149</td>
</tr>
<tr>
<td>Pre-test</td>
<td>100</td>
<td>6.010</td>
<td>1.096</td>
<td>0.110</td>
</tr>
<tr>
<td>Difference</td>
<td>100</td>
<td>0.520</td>
<td>1.920</td>
<td>0.192</td>
</tr>
</tbody>
</table>

95% CI for mean difference: (0.139, 0.901)
T-Test of mean difference = 0 (vs ≠ 0); T-Value = 2.71
P-Value = 0.008
*Pre-Test (Text Base method)
*Post-test (3D animation method)

Table 2: Outcome of the paired t-test

4.1.1.2 Interpreting Result of the Paired T-test

The result provides essential descriptive statistics for the two related groups that were compared including the sample size (N), mean, standard deviation (StDev) and standard error of the mean (SE Mean) and the actual results from the paired t-test. The mean column shows slightly higher performance of participants was higher (approx. 65%) when the 3D method was used compared with a performance of approximately 60% when the text base method was used. The mean disparity between the two was found to be approximately 5.2% (shown in the difference row).

In addition, in the event of comparing the methods across the difference row, it was observed that the standard deviation was 1.920 (the StDev column) with a standard error of mean of 0.192 marks (the SE Mean column). Moreover, the 95% CI for mean difference row shows a 95% confidence interval (95% CI) for the mean difference 0.139 to 0.901. In the final row of Minitab output, an obtained t-value of 2.71 is presented with the statistical significance (2-tailed p-value) of paired t-test (p-value) which is 0.008. With the p-value being less than 0.05 (i.e., p < .05), the conclusion can therefore be drawn that there is a statistically significant difference between the two methods used. In a similar manner, the disparity between mean performance marks after the text base method and after the 3D method is not equal to zero. Though Minitab application does not include the degrees of freedom, it is simply the sample size (the N column) minus 1 (i.e., N-1). Therefore, the degrees of freedom are 100 -1 which is 99.

4.1.2 Student Motivation

This sub-section presents data analysis on the performance of participants on time spent during laboratory activities and their scores on the performance test. It also presents responses of participants on how the methods used during the experiment has motivated them to pursue programming courses at the higher institution of learning.

Laboratory activities were performed at the closing stages of each laboratory session using the text based method and the 3D animation method to determine the level of students’ understanding of programming concepts. During the laboratory sessions, participants were given the same duration of 15 minutes to perform a programming task in text based environment and the 3D environment where interactive animations were created. The overall time spent by participants on each session was equally timed and recorded using a stop watch. Average time spent by participants is presented in figure below:

Figure 2: Average time spent by participants in laboratory activities in the two experiments.

During introduction to the interface of the programming environment used in the 3D method, participants spent an average time of 10 minutes representing 67% of the total time given for the activity as against 6 minutes (40%) during text base method. The participants spent more time due to the fact that the Alice programming tool used in the 3D animation method has more features that the participants were to familiarize themselves with compared to the Microsoft Basic environment used in the text based method. The story was quite different when it came to the activities involving the use of programming commands where the participants spent 8 minutes (53%) of their time during the 3D animation method as against 12 minutes (80%) of the text base method. In a similar development, the participants spent 11 minutes (73%) during text based method as against 14 minutes (93%) used in the 3D animation method on laboratory activities on sequential execution of programs. This disparity was attributed to the interaction styles used in the programming environment as it was faster using the mouse during the 3D animation method to just drag and drop codes into do in order control statements as compared to the text based which made use mainly of the keyboard which was a bit slow.
In performing activities in parallel execution, 8 minutes (53%) was used by the participants during the 3D animation method as against 11 minutes (73%) by the text base method. During 3D animation method on looping structures, the participants spent approximately 10 minutes (67%) and 11 minutes (73%) on decision structure and repetition structure respectively. On the other hand, during text base method, participants spent 13 minutes (87%) of their time on both decision and repetition structures of looping a programming code. It was quite logically difficult during the text base method to understand these two concepts of looping as they saw it as an abstract concept whose real world direct usage is hard to come by.

4.1.2.1 Assessing whether methods used motivated participants to pursue programming at the tertiary level.

Figure 3: Comparing Participants' response as to whether the methods motivated them.

To establish whether the use of the methods provided any motivation for participants to pursue programming at the tertiary level, 42(42%) indicated YES with 58(58%) indicating NO for the text base method as against 53(53%) who responded in the affirmative and 47 (47%) for the 3D animation method indicating it has not provided any significant motivation to pursue programming related courses at the tertiary level.

4.1.3 Assessing methods in terms of ease of understanding programming concepts.

This sub-section presents participants' responses to interview questions in relation to ease of use and learnability of the instructional methods used.

When asked to assess the text based method and the 3D animation method in terms of ease of understanding programming concepts, 26 participants assessed the 3D animation method as very easy with no (zero) participant assessing text base method as such. 49 participants assessed the 3D method as easy as against 19 for the text base method. A total of 25 participants stayed neutral of this response with 10 assessing the 3D method as difficult as against 30 participants for the text base method. There were 8 participants who described the 3D animation method as very difficult as against 33 indicating same for text base method.

In general, per the assessment made by participants, the 3D animation method presented programming concepts in a form that are easier for students to understand.

4.1.3.1 Overall Judgment of the text base method and the 3D animation method.

Figure 5: Participants' overall judgment of the two methods used.

In the quest to find out from participants as to their overall judgment of the text base method and the 3D animation method, no (zero) participant rated the text base excellent, while there were 14 participants who rated the 3D animation as excellent. 6 participants rated the text base very good as against 43 participants for the 3D animation method. 48 participants rated the text base method as poor as against 3 participants for the 3D method. While 25 participants rated the text base method as very poor, there was none (zero) who rated the 3D animation method very poor. When pushed further to give reasons for their ratings, most participants cited the interaction styles (command line) and the interaction device (keyboard) used as difficult and slow in performing tasks. They seem to prefer the interaction style of drag and drop in the 3D animation method using the mouse as the interaction device.

4.2 Observations made by the researcher

An observation checklist was used in the process of observing behaviours and attitudes of students. This helped the researcher in observing behaviours which were of relevance to the study but cannot be expressed verbally, for instance body language and facial expression. In this section, the researcher presents a summary of some noticeable observations made during the two experiments.

1. Participants preferred drag and drop interaction style to the use of the keyboard to interact with programming environments.
2. The repetition and recursive concepts presented some stumbling block for most participants in both experiments.
3. Participants were more enthused of the outputs generated by their programs in the 3D environment and got highly motivated by them to learn programming.

5. FINDINGS

This section discusses the various findings made in this research work in relation to the research questions under the study.

5.1 Development and Findings Relating to Research Question 1.

The effect size produced by the t-test performed on the performance test is a major component in the findings of the study helping in measuring the effectiveness of the 3D animation method as an alternative introductory programming approach. The data below presents the measure of effect size of the 3D interactive animation method.

<p>| Descriptional Statistics: Pre-test, Post-test |</p>
<table>
<thead>
<tr>
<th>N</th>
<th>N</th>
<th>Mean</th>
<th>SE Mean</th>
<th>SD</th>
<th>Minimum</th>
<th>Q1</th>
<th>Median</th>
<th>Q3</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pre-test</td>
<td>100</td>
<td>6.010</td>
<td>0.110</td>
<td>1.096</td>
<td>4.000</td>
<td>5.000</td>
<td>6.000</td>
<td>7.000</td>
<td>9.000</td>
</tr>
<tr>
<td>Post-test</td>
<td>100</td>
<td>6.530</td>
<td>0.149</td>
<td>1.494</td>
<td>3.000</td>
<td>5.000</td>
<td>6.000</td>
<td>8.000</td>
<td>10.000</td>
</tr>
</tbody>
</table>

The p-value presented by the t-test shows a significant difference in dependent variable (performance test) of the methods used. It however does not reveal the magnitude (actual size) of the difference between the two methods. The effect size of 0.52 is the standardized difference between the means of the text base method (pre-test, 6.010) and the 3D animation method (post-test, 6.530) under the mean column of the descriptive statistics above. With a Cohen’s d of 0.5 (medium effect), there is a significant effectiveness in using interactive 3D animation pedagogy to introduce programming concepts to novice compared to the use of the text-based teaching method.

5.1.1 Implication of Paired-Samples T-test for Performance test

Based on the Minitab output presented above, the researcher reported the result of the study as follows:

A paired samples t-test which was run on a sample of 100 novice programmers to determine whether there was statistically significant difference in the level of understanding programming concepts based on instructional method. In other words, the paired t-test was to determine whether there was a statistically significant mean difference in the understanding of programming concepts using text base method and 3D animation method. Participants’ marks in the performance test improved after the 3D animation method (6.530 ± 1.494 marks) than the text base method (6.010 ± 1.096 marks); a statistically significant mean increase 0.520 (95% CI, 0.139 to 0.901) marks, t(99) = 2.71, p < .008.

Both methods performed likewise in relation to performance of laboratory activities; nonetheless, the 3D animation method resulted in relatively less syntax errors and in cases where they arouse, they were easier debugging them. Animations presented a visual output of avatars and in instances where they behave unexpectedly, it clearly signifies bug in the code of the animation that manipulates it actions, making it easier to identify and fix such bugs. This is similar to a position put out by [2] who reports that apart from generating enthusiasm, animations assist students in debugging their program codes because of the imagery components they present.

When asked to assess the text based method and the 3D animation method in terms of ease of understanding programming concepts, participants assessed it as a means that presented programming concepts in a form that are easier for students to understand. Teaching method adopted by the teacher has implication on the perception developed by the learners of programming. Some teaching methods are more suitable to the learning styles of students in learning programming. Comparing the performance of students based on their preference of learning styles, [45] reports that some learning styles of students are more suitable to learning programming than others. A position [22] holds arguing that the critical onus lies on the instructor to make sure that the learners take up the most suitable learning approach for the subject under consideration.

In the quest to find out from participants as to their overall judgment of the text base method and the 3D animation method, their response indicated that the 3D animation method was comparatively better in terms of ease of understanding programming concepts. When reasons were to be assigned for their response, most participants cited the interaction styles (command line) and the interaction device (keyboard) used in the text base method as difficult and slow in performing tasks which was evident in the duration used during laboratory activities. For instance, sequential execution of programs, the participants spent 11 minutes during text based method as against 14 minutes during the 3D method. This disparity was attributed to the interaction styles used in the programming environment as it was faster using the mouse during the 3D animation method to just drag and drop codes into do in order control statements as compared to the text based which made use mainly of the keyboard which was a bit slow. This outcome confirms the work of [19] which concludes that the preference for the mouse as an interactive device (or style) in interactive learning environment affects the performance of the students. It must therefore be noted that the choice of interactive device and style adopted in any programming instructional tool has repercussions on both the learners and the teaching method.

5.2 Development and Findings Relating to Research Question 2

Though the two methods of instruction produced good result in terms of performance score, less time was used during the 3D animation method to perform relatively the same task. The average time spent by the participants in the experiment using the 3D animation method was better as compared to the time used by the same participants during the experiment involving the text base method. In addition to this, the result of the paired t-test analysis on the performance test showed that teaching programming concepts using interactive 3D animations had statistically significantly higher comprehension (6.530 ± 1.494) compared to teaching programming concepts using text based method (6.010 ± 1.096), t(99) = 2.71, p = 0.008. Statistically, as the p-value, is less than 0.05 (i.e., p < .05), it can therefore be deduced that there is difference in mean performance between the two experiments. Other researches has resulted in similar conclusion with [13] positing that active teaching method led to better student performance while the lecture text base led to the lowest overall scores of any of the teaching methods available to the teacher.
In addition, responses provided by participant during the interview point out that the use of 3D animation method impacted positively on their scores in the performance test. In assessing the method in terms of ease of understanding programming concepts, responds from participants indicated difficulties in the text method whereas the 3D animation method was assessed as a method that presents programming concepts in form that enable novice to understand easily. A position [26] presented a solid evidence to arguing that program visualization systems improves the learning performance of novice programming students significantly. During the experiments, the text output presented by the text based method could be directly responsible for the responses provided by the participants. They had difficulties relating such outputs to real solid world replica making the concepts not to be easily assimilated. This validates the arguments put out by some notable researchers in the computing environment. [45] and [27] put out that the text base lecture of teaching is ill-suited for nurturing cognitive skills and does not excel in developing the skills of application which is the foundation of learning programming. The text base method which is currently used by instructors to introduce programming concepts at the SHS level does not match up with the learning styles of the present generation of learners who are deeply immersed interacting with 3D environments. This suggests that the teaching approach of using interactive 3D animation proves to enable students to perform better in programming tasks. It therefore validates the position for the conclusion to be drawn that, using 3D interactive animation to teach programming concepts to novice impacts positively on the performance of students. This finding was consistent with a number of similar previous experimental researches including [14] [37] [33] on the use of program visualization (PV) tools that improves the students learning performances.

5.3 Development and Findings Relating to Research Question 3

To ascertain whether the use of text base method provided any motivation for participants to pursue programming at the tertiary level, 58 (58%) responded NO. This was identified in the literature review as one major cause of difficulties in learning computer programming. A position supported by [30] who opined that this problem can be partly be blamed on the fact that programming has been taught using programming languages that are complicated to use and with activities that are not connected to students’ enthusiasm. In seeking to know whether the use of 3D method provided any motivation for participants to pursue programming related course at the tertiary level, 53 (53%) indicated that it has provided significant motivation to pursue programming related courses at the tertiary level. This affirms findings made by [8] who reports a major increase in knowledge acquisition when animations are used to teach programming and also indicated that it motivated students to take up computer science related courses in the future. [40] also shared the same opinion arguing that the use of such systems is best placed before students enter the university as part of an introduction to programming.

In addition, observations made by the researcher via the use the observation check list exposes the use of the text base method of teaching introductory programming to be partially blamed for the misconceptions that hover around the minds of novices. The ages of students at this level of the Ghanaian education ladder ranges between 13 to 19 and as such, using unappealing text based methods to teach abstracts concepts as those in programming easily puts them off and result in the development of mean posture towards the course. The researcher observed frustrations and lack of interest on the faces of participants during the experiment on the text base method as participants struggle to comprehend concepts especially those involving iteration and recursion. These concepts were presented in a simpler form in the experiment that used the 3D animation method where participants had to drag and drop iterative and recursive codes and get it run correctly. Programming concepts such as iteration and recursion when introduced to novice in a 3D environment presents these concepts in a real world replica removing the mystery they hold about programming. Also, engaging students in laboratory activities during programming lessons create a challenging classroom atmosphere for the quest to make their program work and generate the needed enthusiasm for the course. The response provided by the participants on their judgment of the 3D method concludes that its usage has provided the needed motivation to pursue programming related course at the tertiary level. This view was also shared by other research outcomes [45] [27] who indicated that programming is better learnt with in-class activities.

The interface in the text base method created some interactive limitations. A barrier, [24] recommended a cautious design of the interface of such programming systems to go with its semantics which was not implemented in the interface design used in the text base interface. In assessing the interface used in the 3D animation method, majority of them responded that it was user friendly. The sum up percentage of 73% of respondents who indicated that the interface was user friendly could be attributed to the 3D nature of the interface which the participants interacted with similar to that of the real world. On the basis of the above, it is imperative at this point to conclude that the 3D animation method generated the needed motivation for students to learn computer programming.

5.4 Contribution of the study to the body of knowledge

The interventions to deal with difficulties in learning programming come in various forms comprising changes to the course content, the pedagogy and the use of programming learning support systems including programming in 3D virtual learning environments (3D VLEs). Although there have been numerous research publications surrounding 3D aspects of these programming environments, a critical scrutiny of them revealed that their use in the teaching of programming has produced conclusions that are highly unambiguous with many generic weaknesses. It was therefore worth considering to compare the use of the traditional text base method to the use of the 3D animation method. The use of 3D animation method presents programming concepts in a form that the learners can understand, motivates them to pursue programming related courses at a higher level and also impacts positively on their performance. This study therefore presents the arguments for the suitability of using 3D animations as an instructional method of programming filling in the gap that existed.

6. CONCLUSIONS

Findings made with respect to data collected during the implementation of the research design led to the following conclusions:

1. That the use of interactive 3D animations to teach students at the SHS level is an effective introductory approach at the SHS level. However, the use of the tool might present an additional learning task for students on how to navigate and use the programming environment.
2. That the use of 3D animated products to visualize programming concepts motivated students to learn programming and enriched learning experience.
3. That the use of interactive 3D animations approach to teaching programming at the SHS level impacted positively on student performance.

7. RECOMMENDATIONS

There are quite a few areas for further development and application of the work undertaken in this study. These are enumerated as follows:

Alice programming environment should be adopted by the Ministry of Education and the Ghana Education Service (GES) as a teaching tool for introductory programming course especially in explaining the concepts of iteration and decision making statements. Caution should however be taken by the instructor in the use of the method to avoid creating false impression in the minds of students which can be damaging when these students are to switch over to more professional language whose programming environment are not 3D based and has no avatar to manipulate. In order to minimize the cognitive burden of using such animation environment as a learning tool, there is the need for the instructor to introduce to students the features of the tool ahead of time.

In order to assess a more general impact of this approach in introductory programming course, it is recommended that the geographical scope of the sample be expanded to include SHS schools in more districts and other regions of Ghana.

Assessment has been made between major programming environments in introducing novice to programming, there is the need to assess the impact of using programming visualization tools as a supplement to the traditional text base method. In addition, survey should be considered as an additional data collection tool to broaden the scope of the data collected.

8. REFERENCES

Sciences Laboratory, National Institute of Education, Nanyang Technological University.


