The Effect of Improvised Computer-Based Software Package on Secondary School Students’ Achievement and Attitude Towards Mathematics

Etukudo Udobia Elijah
Department of Mathematics Education, Federal College of Education (Technical), Omoku, Rivers State, Nigeria

Egba Anwaitu Freder
Department of Computer and Robotics, Federal College of Education (Technical), Omoku, Rivers State, Nigeria

Abstract

The study was conducted to determine the effect of improvised computer-based software on the achievement and attitude of senior secondary school students towards mathematics. Fifteen secondary school mathematics teachers were trained to produce a computer-based software package and use the package to teach ‘graphical solution to quadratic equations’ to senior secondary class one (SSC1) students. A total sample of 1487 SSC1 students was used in the study: 745 students in the 15 secondary schools purposively selected on the rationale of their closeness to a computer center, used as the experimental group, and 742 students from the 15 schools randomly selected for use as the control group. Pre-test, post-test, and quasi-experimental research designs were used to conduct the study. Fifteen intact classes were simultaneously used for treatment in the experimental and control groups. The experimental groups were taught with computers using the computer-based software package on teaching graphical solutions to quadratic equations produced by their teacher, while the control groups were taught using the conventional strategy. The result reveals a significant difference in academic achievement between students taught with improvised computer-based software packages and those taught using the conventional strategy. The mean achievement score of 73% and standard deviation of 9% was obtained from the experimental group, while the control group had a mean achievement of 57% and standard deviation of 5%. Similarly, the experimental group had an attitude test score of 82% and a standard deviation of 6%, while the control group had a mean attitude test score of 59% and a standard deviation of 11%. This gave the calculated value of the t-test as t = 42.3 for achievement and t = 50.1 for attitude, showing a significant difference in achievement and attitude between the experimental and control groups. Thus, an improvised computer-based software package is more effective in teaching mathematics than the conventional strategy. Hence, improvising computer-based software for teaching and learning is highly recommended.

Keywords: Mathematics, improvisation, computer-based software package, achievement, attitude

Introduction

It has become obvious that using computers in teaching and learning has contributed immensely to alleviating the problem of understanding and mastery. Computers have played a huge role in making learning easy. Teachers who have been able to lay their hands on adequate and suitable software for teaching and learning always find it easy to communicate the instructional content to learners so they can easily master it. The joy of a good teacher is in having desirable results. This is made possible with the use of the computer. Rouse (2022) emphasized that computer-based learning has many benefits, such as helping users to learn at their own pace, being very interactive, learning at one desired time, having that which is globally accepted, and having the ability to accommodate traditional methods. Rouse (2022) stated that knowledge-based training and assessments, simulation, and drills are done prudently using computers. This makes using computers relevant to teaching and learning (Ingram, 1985). The computer software package was developed and used in teaching the English Language, and it was discovered that the product was easy to duplicate, contain elaborate learning material, pictures, and displace design, which made the students more enthusiastic and interested, while the exercises were more fascinating (Rohmah, 2019). Using computers in teaching has helped to remove the complexity in instructional delivery (Garba et al., 2010; van Merriënboer & Martens, 2002).

The most severe obstacles to implementing computer-assisted instruction in school systems include the non-availability of facilities and the lack of affordable and relevant software packages. The relevant packages to use and carry out instructional delivery are scarce. Several models have been presented to be used to develop computer-based instructional packages. Rosenthal (1976) presented a model containing segments and phases of instruction, which included administration and coordination, development of a project team, definition of project scope and objectives, development of a program for system design, and conducting orientation meetings. Evaluating this model before implementing computer-based instruction makes it look herculean. Chen and Shen (1989) noted that waterfall models are the key to making software adaptable to
users and satisfying their needs. Similarly, Pappas (2014) instructed that the production of computer-based software packages has been the bends in computer-based instructional delivery. Ruliah et al. (2019) elucidated the advantages of computer-assisted instruction. They proposed using a conceptual model comprising the introduction, presentation of information, question and responses, feedback, question and responses, feedback and responses, judging and responses, remedial, case exercise, and closing.

The software has been developed with various languages for teaching and learning. However, the process seems complex and clumsy, making it difficult to spread and effectively use; hence, some packages have a short life. Software developed by Benitez (2016), Garba et al. (2010), and Michael and Ignewari (2022) has exciting results but was produced with peculiarity for specific instruction and cannot be generally applied or used in wider perspectives. These lapses make it obvious that there should be a need to produce software that can be applied in all areas when the need arises. It has become pertinent that teachers should know how to produce software for their instructional usage. It cannot be anticipated that instructional software can be sourced from the market for every topic to be taught.

This study is carried out to train mathematics teachers in producing instruction software using a reality model. The reality model provides direction that the teachers can always follow to produce software as much as they choose to and for every topic they need to use. In addition, secondary school teachers were taught to improve the computer software used to teach graphical solutions to simultaneous equations and compare them with the conventional strategy.

Statement of the Problem

Considering that computer-based software has a huge positive impact on teaching and learning as well as general instructional delivery when applied, it becomes necessary that the product be available for teachers and learners at any time there is a need to apply. This cannot be the case, except that the teachers can produce the software for their use in teaching and learning mathematics. Teaching mathematics for learners to achieve maximally and induce an acceptable attitude towards the subject has always been the bend in education. Many strategies have been devised to help stimulate learners' attitudes towards the subject and boost their achievement, among which are improvised strategies for effective teaching and learning of the subject. The reason question and answer me and learners' attitude towards the subject has always been the bend in education. Many strategies have been devised to help stimulate learners' attitudes towards the subject and boost their achievement, among which are improvised strategies for effective teaching and learning of the subject. The reason

Hypotheses

1. There is no significant difference in the mean achievement score of students taught mathematics using improvised computer-based software packages and those taught using conventional strategy.
2. There is no significant difference in the mean attitude test scores of students taught mathematics with improvised computer-based software packages and those taught using conventional strategy.

Methodology

Design

The study is a quasi-experiment that was carried out with thirty (30) senior secondary school mathematics teachers. Fifteen (15) were in experimental and control groups, respectively.

Population of Study

The study population includes all mathematics teachers in public and private secondary schools in Rivers State of Nigeria. A total of 2,155 mathematics teachers made up the population.

Sample and Sampling Technique

A total of one thousand four hundred and eighty-seven (1,487) senior secondary class one (SSC1) students were used for the study. They were seven hundred and forty-five (745) for the experimental group and seven hundred and forty-two (742) for the control group. A purposive random sampling procedure was used to select the participants in the experimental groups due to proximity to standard and well-equipped computer laboratories, while random sampling was used to select the schools for the control group.

Instrumentation

The instruments used in the study include
1. Training manual
2. Program packages
3. Participant involvement checklist
4. Training checklist
5. Participant's trial test items
6. Participant assessment checklist
7. Mathematics achievement test
8. Mathematics attitude inventory test
**Training Manual**

This is a set of rules for the training. It states categorically the time each aspect of the training takes place. The introduction takes about ten (10) minutes. Familiarization with the computer and mastery of the keyboards twenty (20) minutes. Introduction to programming thirty (30) minutes. Interaction with the programmers and the technicians is thirty (30) minutes, and practical exercise is thirty (30) minutes. This formed the two-hour training for the first day.

The second day of the training was an introduction to BASIC programming. The participants were taught how to write BASIC programming language. The question and answer session was on the different technicalities of writing BASIC programming and producing software with it. The practical session was also included for forty-five minutes. The session ends with the correction of trainees’ programs and assignments.

The third day continued with BASIC programming. The theoretical session was one hour, while the practical session was one hour, which ended with correcting trainees’ programs and assignments.

Day four was the rehearsal of BASIC programming and package production with BASIC. The fifth day, day five, was an introduction to Python programming. It was one hour of theoretical and practical work was for one hour. The session ended with an assignment on Python programming language.

The sixth day, day six, was practical programming with Python, which took one hour and forty-five minutes, except for the fifteen minutes of introduction, which ended with assignments.

The seventh day, day seven, was animation programming. One hour was the introduction and theoretical work on animation, and another hour was the practical programming and production of two-dimensional and three-dimensional objects.

The eight days were spent on mixed program package writing. Programming packages with BASIC and Python, while the ninth day was the introduction to mathematics program packages production and sample packages.

The tenth day was practical package production using topics from the scheme of work.

The eleventh day was trial package validation, while the twelfth day was implementation of the trial package. The packages were short instructions produced by the teachers.

The thirteen and fourteen days were for revision.

The third week was the practical implementation of the packages, marking pre- and post-tests of students taught with the improvised packages, collecting results on the participant’s achievement, and closing the training workshop.

**Program Packages**

The training instructions were packaged in BASIC and Python. They were delivered through the local area network (LAN) to the different computers the trainees were using, with screen and multimedia facilities used for explanations, illustrations, and interactions. The packages were validated correctly. Other programming packages in mathematics instructions were also made available and used as examples for discussion.

**Trainees’ Involvement and Commitment checklist**

The checklist contained the rating of the activities of the trainees during the lesson. It was presented in the table below for each of the participants. Technicians and research assistants gave the rating.

**Training Checklist**

This general checklist was used to monitor the rate at which the training was progressing and serving each aspect that was required.

**Table 1**

*Participants Checklist*

<table>
<thead>
<tr>
<th>Items</th>
<th>1-10</th>
<th>11-20</th>
<th>21-30</th>
<th>31-40</th>
<th>41-50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punctuality</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Attentiveness</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Contributions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Asking questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Answered questions</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to use a computer</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to write codes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to interpret codes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Logical production of ideas</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality program package on a chosen topic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note.* The above was the individual checklist for each of the participants during learning.

**Table 2**

*Training Checklist*

<table>
<thead>
<tr>
<th>Items</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proper introduction</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proper use of package</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevance to trainees need</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Carrying trainees along</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Helping the trainees to master concepts</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Enabling the trainees to meet objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trainees’ ability to produce packages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Trainees’ package meeting the objectives</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adequacy of trainees’ packages</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learners’ learning outcome improvement with trained package</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Participant Trial Test Items**

This comprised the pre-test and post-test items the trainees used to implement their packages. They were set on how their packages were produced and administered before and after their experiments.

**Participant Achievement Checklist**

This is where items were scored and evaluated by the participants.
Table 3
Participant Achievement Checklist

<table>
<thead>
<tr>
<th>Items</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Involvement in the training</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to write the program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production of a workable package</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Relevant of the package to the object to the topics</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acceptance of the package by learners</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Achievement of objectives through the package</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proper usage of the package</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evaluation of the objective via the package</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>General assessment of the trainee product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Learners assessment of trainees' product</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Mathematics Achievement Test

The Mathematics achievement test comprised fifty (50) multiple choice items given to the subjects in both the control and experimental groups as pre-test and post-test. The test had a parallel form reliability coefficient of 0.87. The items selected were those that satisfied the discriminatory index of 0.45 to 0.55. The items were distributed on the basis of 20 on cognitive, 15 on psychomotor, and 15 on affective domains. In totality, ten items were on memory, 10 on comprehension, 10 on analysis, ten on application, 5 on synthesis and 5 on evaluation. The subjects had to draw graphs to answer some of the items in the affective and psychomotor domains. The items were selected from 80 items used for the pilot study.

Mathematics Attitude Inventory Test

This comprised a 20-item Likert scale test, which was structured to determine the effect of strategy on attitude towards mathematics. It had a split-half reliability coefficient of .78. It was given to the subjects at the end of the treatment.

Research Procedure

The research was carried out by purposively sampling fifteen secondary school teachers who were used to teach the experimental group and participate in a training program for fifty mathematics teachers for computer-based instructional package production. The training was carried out in the Federal College of Education (Technical) computer laboratory with the help of two programmers, four technicians, and five other research assistants. The trainees were taught how to produce computer-based instructional packages in mathematics.

After the teaching, they produced their packages and tested them. Their packages were in different topics in mathematics taught at the level which they were teaching. They used the packages to evaluate their learner’s learning outcomes to determine the success of the training. They were evaluated using the checklist. The data were collected using the checklists.

After the general training, the fifteen secondary school teachers produced computer-based instructional packages on graphical solutions to quadratic equations and used them for this study.

Training Model

The efficiency skewed four prompt models encompass four key agents that drive the activities: Instruction, programmer, teachers, and learners, which were used for the study.

The teacher and the programmer first examined the instructional contents, and then the nature of the software package was decided by flowcharting the components and processes. It was from here that the programming language or languages were chosen. After that, the presented instructional content and subject matter were keyed into the package. Flexibility, adaptability, suitability (relevance), acceptability (meeting the learners’ needs), adequacy, and durability were considered very important elements of the functional ability of the packages. The packages were exposed to learners’ appraisal in order to seek their satisfaction in trial with a smaller group, a smaller group of the contemporaries for which the package is produced. It is at this point the package can be considered efficient. This gives rise to the “efficiency skewed four prompt model” for producing computer-based packages for teaching and learning mathematics.

The training was based on the “efficiency skewed four prompt model” termed REALITY MODEL given below.

Figure 1
Efficiency Skewed Four Prompt Model

Data Analysis Procedure

The hypotheses were tested using a t-test to determine the difference between the experimental and control groups in achievement and attitude. All the hypotheses were tested at .05 level of significance.

Results

Hypothesis 1: There is no significant difference in the mean achievement score of students taught mathematics using improvised computer-based software packages and those taught using conventional strategy. The data analysis for testing hypothesis 1 is given in Table 4 below.
The result revealed a significant difference between experimental and control group student achievement. The experimental group, with a mean achievement score ($M = 73, SD = 9$), as against the control group with a mean achievement score ($M = 57, SD = 5$), had higher achievement, which gave rise to $t$-test of 42.3 which is higher than the critical value of 1.96. As a result, hypothesis one is rejected. There is a significant difference in the achievement of the experimental and control groups.

Hypothesis 2: There is no significant difference in the mean attitude test scores of students taught with improvised computer-based software packages and those taught using conventional strategy. The data analysis for testing hypothesis 2 is given in Table 5 below.

Table 5
Mean, Standard Deviation, and $t$-test of Attitude Scores of Experimental and Control Groups

<table>
<thead>
<tr>
<th>Groups</th>
<th>$n$</th>
<th>$M$</th>
<th>$SD$</th>
<th>$t$-test</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Experimental</td>
<td>745</td>
<td>82</td>
<td>6</td>
<td>50.1</td>
<td>Reject</td>
</tr>
<tr>
<td>Control</td>
<td>742</td>
<td>59</td>
<td>11</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note. $p = .05$, $t_{critical} = 1.96$. $N = 1487$.

The result revealed a significant difference between the attitude scores of students in experimental and control groups. The experimental group with a mean attitude score of ($M = 82, SD = 6$), against the control groups with mean attitude scores ($M = 59, SD = 11$), had higher attitude scores, which gave rise to $t$-test of 50.1 which is higher than the critical value of 1.96. As a result, hypothesis two is rejected. There is a significant difference in the attitude scores of the experimental and control groups.

Summary

The research was carried out to determine the effect of improved computer-based instructional software packages on achievement and attitude of students towards mathematics, using 1487 students, divided into 745 students in the control group and 742 students in the experimental group. The result showed that students taught with an improvised computer-based software package had higher mean achievement and attitude test scores than those taught using the conventional strategy. Therefore, teachers should be trained to produce computer-based software packages for teaching mathematics.

Conclusion

This research focused on the effect of improvised computer-based software packages for teaching and learning mathematics on students’ learning outcomes, achievement, and attitude toward mathematics. The result showed significant differences in the achievement and attitude towards mathematics of students taught with improvised computer-based instructional software in mathematics and those taught using conventional strategy. Those taught using improvised computer-based software packages for teaching and learning mathematics had higher mean achievement and attitude test scores. Hence, the improvisation of computer-based software packages for teaching and learning mathematics should be encouraged.

Recommendations

Based on the findings of this study, it is recommended that:

1. Teachers should be trained to produce computer-based software packages for teaching and learning.
2. The training should be done by:
   2.1. Teachers themselves involve computer programmers to help them
   2.2. School authorities sponsoring the teachers for the training
3. The use of teacher-made computer software packages should be encouraged to reduce the scarcity of software packages for effective teaching and learning.

References


Received: 09 May 2024
Revised: 01 June 2024
Accepted: 12 June 2024