Development of a Compact, Interactive Online Video Lesson for Learn Adlib College Lower Six Students Using Adobe Animate and Adobe Premiere 2022

Mavu Newton
Zimbabwe Open University
Bulawayo, Zimbabwe

Abstract: This research is a development and experimental research premised on the 4D model. A compact video lesson was developed for Lower Six students at LearnAdlib College in Bulawayo, Zimbabwe. An introductory video was developed in Adobe Premiere 2022 using presets and background sounds obtained from the internet. Main content of the lesson was recorded using Zoom. Adobe Animate was then used to develop the user interface and to integrate all parts of the lesson. Interactivity was enabled through the use of ActionScript programming language in Adobe Animate 2022. Buttons were developed to enable the student to navigate the lesson from the introduction till the end. The main aim of the lesson was to introduce Visual Basic programming through both theory and a practical activity. Videos to explain the basic concepts and to demonstrate the practical activity were developed and deployed to student machines so that they run the lesson in the browser. A Lower Six class of 28 students was split into two, the experimental class and the control class. Ten (10) multiple choice questions and one practical exercise were used to test student understanding for both the experimental group and the control group. Three Computer Science teachers evaluated the compact lesson's suitability to the level of learning of the students. They also commented on validity of lesson content in relation to the requirements of the Zimbabwean Advanced Level Computer Science Syllabus 6023. The outcome of the experimental research showed that students who used the online compact lesson managed to answer questions and to produce scores as high as those of the control group. The compact online lesson was therefore as effective as the physical lesson. It had the merit that students could replay some of the areas when doing the practical, each replaying as many times as he or she wants until he or she gets the steps correctly. Thus in an interactive video lesson the student can control the pace of learning while in a face to face lesson the teacher will have to determine pace. That was very convenient for the students and that made them enjoy the lesson. However, they could not ask questions on the go in times when they would have needed to: the advantage which was enjoyed by those in the control group. Students in the face to face class can ask the teacher some questions any time during the course of the lesson.

Keywords: computer programming, 4D model, interactive online video lesson, Adobe Animate, Adobe Premiere Pro

1. INTRODUCTION

Computer programming is a very important component both for high school Computer Science students in Zimbabwe and for the real world of computing in general [5]. Programming as a process of software development involves a lot of problem solving skills. Students also need to understand how the computer works for them to be able to design solutions that will be automated by the use of the computer [6]. High school Computer Science students in Zimbabwean schools will need to understand both the theory and practice of programming for them to be ready for the examinations and the world of work after school. However, there is shortage of skilled Computer Science teachers in the country and students at times spend a lot of time looking for online alternatives to learning especially during the holidays or weekends [4]. Also, currently there is still need to cover learning gaps that were created by learning stoppage during the COVID-19 era [3].

One of the ways to close learning gaps is to enable independent learning on the part of the student during the time they will not be in the usual physical class [3]. This is where online learning comes in and hence the need the develop multimedia lessons that are interactive to motivate the student to go an extra mile. Thus the student needs to do both face to face classes and online classes. This is called blended learning and is a means of making the student cover up for time lost because of any problem [3].

Adobe Premiere Pro 2022 is a video authoring tool (software) developed by a software company called Adobe Systems [1]. Adobe Animate 2022 is an animation development tool (software) that was also developed by Adobe Systems [2]. Thus both these tools are products of the same company and when using them one can easily export files from one tool into another. In Adobe Animate 2022, one can do programming using the programming language called ActionScript. It is also possible to develop applications that can run in the browser using HTML and CSS [2]. Thus HTML and CSS is used to develop the user interface for applications (in this case the lesson) that runs in the browser. To control and navigate the video, ActionScript programming language comes into play.

2. THE RESEARCH METHOD

This research is a research a mix of development and experimental research which begins with the 4D model for designing instructional media and then goes further into testing of the video lesson in teaching the experimental class. Performance of the experimental class is going to be compared with the performance of the control class.

The instructional media to be developed is a compact video lesson to introduce Visual Basic programming to Lower Six students. It will introduce the IDE (Integrated Development Environment) called Visual Studio Community from Microsoft. Students will watch a theory video lesson on variables, constants and data types in Visual Basic programming language. They will then watch a demonstration

on how to create a project in Visual Studio and then to declare, initialize, manipulate and print results onto the screen. A theory and a practical exercise will then be done and results will be tabled. Finally, both the Computer Science students and their teachers will complete some questionnaires on product usability and effectiveness.

3. RESULTS AND DISCUSSION

3.1 Define

The aim of this research is to develop a compact and interactive online video lesson for Lower Six Computer Science students at LearnAdlib College. The objectives of the lesson to developed were as follows:

By the end of the lesson students should be able to:

-define what a variable is and to explain at least three associated variable data types found in Visual Basic programming language.

-create a new Visual Basic project in Visual Studio IDE.

-declare and initialize variables using Visual Basic programming language.

-manipulate variables and print output on the console window.

During the definition stage related articles were read to gain understanding of the nature of the problem at hand. The researcher also visited LearnAdlib College to do preliminary interviews with the teachers and the students. The following documents were downloaded from Zimbabwe's Ministry of Primary and Secondary Education website (http://mopse.co.zw/).

1. Education policy 5.0
2. Advanced Level Computer Science Syllabus
3. Sample examination papers

The main reason for reading these documents was to check on the competencies and skills required by the ministry so that the media was to be developed aligning to such requirements. Introduction to Programming was chosen as a suitable topic to teach the students considering what they had covered with their teachers. That introduction was to focus mainly on variables and data types in Visual Basic programming language.

3.2 Designing

User interface was designed using HTML and CSS in Adobe Animate canvas. A simple lesson home page was developed which only had important information and the college logo. Other lesson stages were developed using ActionScript buttons and video components. A student would click a button to go to any lesson stage. The illustration of the lesson home page is shown below.

Figure 1. Lesson home page

On the home page above, lesson stages are seen as buttons that would allow the student to navigate to any part of the lesson. Each button is linked to a video component which holds the suitable video for that stage of the lesson. When the dutton is clicked, the correct video gets onto the timeline for the student to watch that part of the lesson. So the lesson stages are

- Lesson introduction
- Lesson objectives
- Basic terms
- Lesson content
- Lesson activities
- Lesson exercise
- Conclusion

Examples of video components with videos for lesson introduction and lesson objectives are as follows.
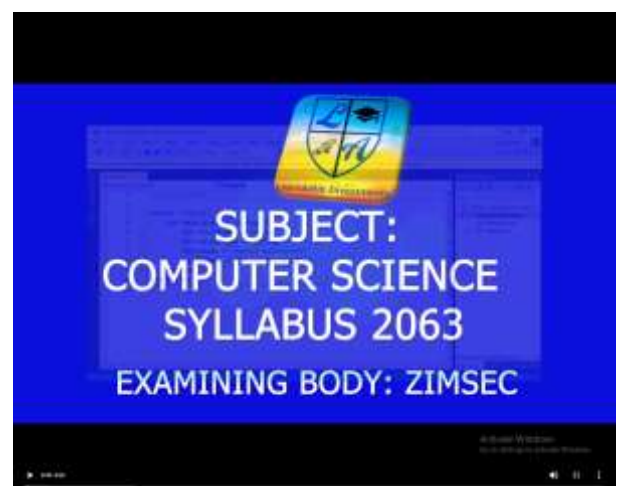

Figure 2. Video for the lesson introduction
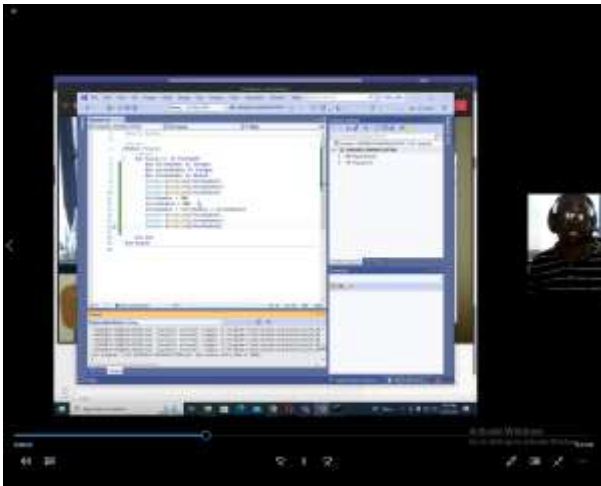
Figure 3. Video for the lesson objectives

Figure 3. Video for the lesson practical programming

3.3 Developing functionality and interactivity

As can be seen on two sample videos above, navigation from one stage of the lesson has been laid from top to bottom, but the student can click at any button at random and will be taken to the lesson stage clicked. Sample ActionScript code for the navigation buttons was as follows:

```
/* Stop at This Frame

The  timeline will stop/pause at the frame where you insert this code.

Can also be used to stop/pause the timeline of movieclips.

*/

this.stop();

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.
```

```
2.Frame numbers in EaselJS start at 0 instead of 1

*/

this.btnIntro.addEventListener("click", fl_ClickToGoToAndStopAtFrame.bind(this));

function fl_ClickToGoToAndStopAtFrame()

{

        this.gotoAndStop(5);

}

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.

2.Frame numbers in EaselJS start at 0 instead of 1*/

this.btnObjs.addEventListener("click", fl_ClickToGoToAndStopAtFrame_2.bind(this));

function fl_ClickToGoToAndStopAtFrame_2()

{

        this.gotoAndStop(10);

}

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.

2.Frame numbers in EaselJS start at 0 instead of 1

*/

this.btnBasics.addEventListener("click", fl_ClickToGoToAndStopAtFrame_3.bind(this));

function fl_ClickToGoToAndStopAtFrame_3()

{

        this.gotoAndStop(15);

}

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.

2.Frame numbers in EaselJS start at 0 instead of 1

*/

this.btnContent.addEventListener("click", fl_ClickToGoToAndStopAtFrame_4.bind(this));

function fl_ClickToGoToAndStopAtFrame_4()

{
```

```
        this.gotoAndStop(20);
}
```

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.

2.Frame numbers in EaselJS start at 0 instead of 1

*/

```
this.btnActivities.addEventListener("click", fl_ClickToGoToAndStopAtFrame_5.bind(this));

function fl_ClickToGoToAndStopAtFrame_5()
{
        this.gotoAndStop(25);
}
```

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.

2.Frame numbers in EaselJS start at 0 instead of 1

*/

```
this.btnExercise.addEventListener("click", fl_ClickToGoToAndStopAtFrame_6.bind(this));

function fl_ClickToGoToAndStopAtFrame_6()
{
        this.gotoAndStop(30);
}
```

/* Click to Go to Frame and Stop

Clicking on the specified symbol instance moves the playhead to the specified frame in the timeline and stops the movie.

Can be used on the main timeline or on movie clip timelines.

Instructions:

1. Replace the number 5 in the code below with the frame number you would like the playhead to move to when the symbol instance is clicked.

2.Frame numbers in EaselJS start at 0 instead of 1

*/

```
this.btnExit.addEventListener("click", fl_ClickToGoToAndStopAtFrame_7.bind(this));

function fl_ClickToGoToAndStopAtFrame_7()
{
        this.gotoAndStop(34);
}
```

The code shows that if a student clicks a certain button, ActionScript will automatically jump to a defined point in the timeline. When it arrives at a defined point in the timeline, it

will load a video at that point and wait for the user to click the play button. The user will also be availed with other control buttons at that point. Thus he can pause the video, increase or decrease the volume, increase or decrease playback speed, view on full screen mode or in picture by picture mode or even download the video for offline play.
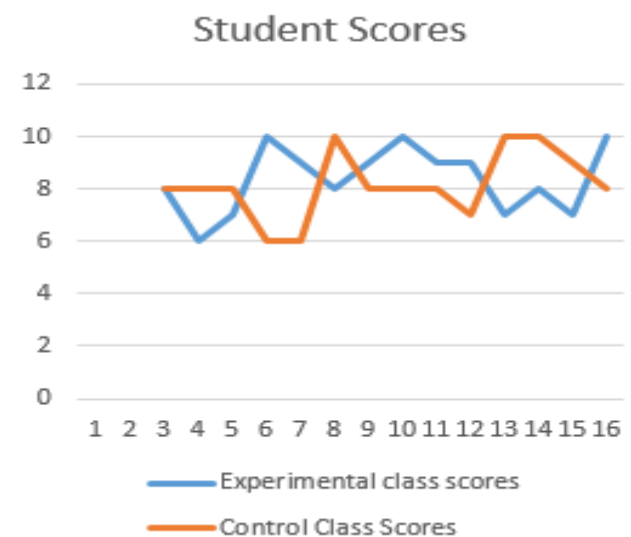
3.4 Student Scores for the Theory Exercise out of 10

From the table below, it can be noted that marks ranged from 6 to 10 out of 10 for all the two classes, the experimental class and the control class. Average mark for the experimental class was slightly greater than that of the control class. Average for experimental class was 8.3 out of 10 while that of the control class was 8.1 out of 10.

Table 1. Student scores for the theory exercise out of 10

Experimental class	8	6	7	10	9	8	9	10	9	9	7	8	7	10
Control class	8	8	8	6	6	10	8	8	8	7	10	10	9	8

The student scores in the above table have been plotted on the graph below.

**Figure 3. Student scores for the theory exercise out of 10**

3.5 Student Scores for the Practical Exercise out of 20

From data in the table below, the difference between scores of the experimental class and those of the control class was also small. The average mark for experimental class was 17.8 while that of the control class was 17.4 out of 20.

Table 2. Student scores for the practical exercise out of 20

Experimental class	16	18	19	18	17	20	20	19	17	15	16	17	18	19
Control class	18	19	20	17	16	10	15	17	19	16	20	18	16	17

Student scores in the above table have been plotted on the graph below.

Student Scores on Practical Exercise
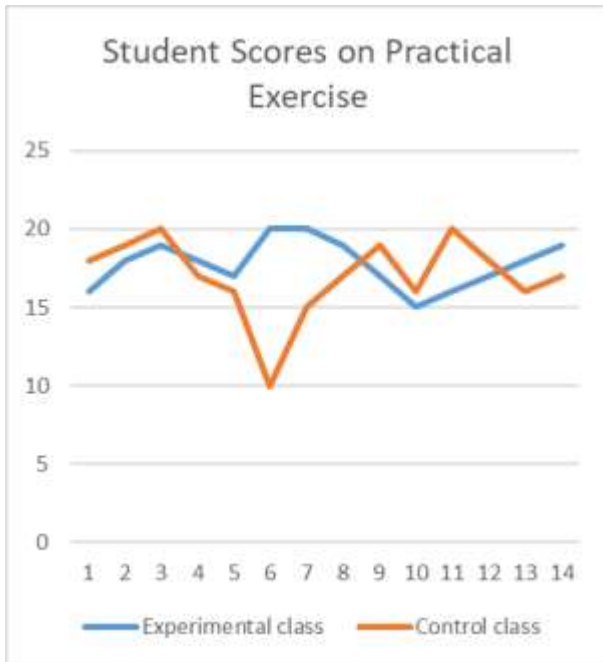
3.6 Media Evaluation by Students

Students were generally thrilled the video lesson. They cited that they had control on the lesson because they could navigate to any stage of the lesson as they learned independently. They could also revisit some of the stages if they did not understand. Or at times they could decrease playback speed so that they follow the practical session to the end.

Their problem, however, was that they could not ask questions in a video lesson the way they would do if they were in a real lesson. They had to remain with unanswered questions and would wait with such questions until they meet the teacher some other time. That was unlike those who were in the control class who would ask questions as and when they so wish.

3.7 Media Evaluation by Computer Science Teachers

All the three Computer Science teachers felt that the media was quite suitable for the age level and it managed to achieve the objectives for which it was developed. Two of them however, noted that there is need for videos to be perfected more so that they draw more attention from the students.

One of them suggested that there is need for thorough preparation when developing an online video lesson than when preparing for a live lesson because in a video lesson no one will be available to answer student questions if some of the information becomes unclear to the student watching the lesson. That means a video lesson should try as much as possible to exhaust details associated with a concept it is trying to convey.

4. CONCLUSION

The development of an online video lesson using Adobe Animate and Adobe Premiere Pro 2022 and the testing of the video lesson was done and completed. Performance of the students in the experimental class was slightly above that of the students in the control class. This proved that students can do well if such material is developed and provided for them so that they do blended learning instead of waiting for physical face to face lessons with a teacher. Online video lessons will make them learn outside the classroom. However, there is need for video lessons to be prepared more thoroughly as they will be used by the in the absence of the teacher and hence such lessons should not leave students with many unanswered questions.

5. ACKNOWLEDGMENTS

6. REFERENCES

[1] Adobe Premiere Reference Book accessed at https://help.adobe.com/archive/en/premiere-pro/cs6/premiere_pro_reference.pdf

[2] Adobe Animate Reference book accessed at https://helpx.adobe.com/pdf/animate_reference.pdf

[3] Hove, B., & Dube, B. (2021). COVID-19 and the entrenchment of a virtual elite private school: Rethinking education policies in Zimbabwe. Journal of Culture and Values in Education, 4(2), 84-94. https://doi.org/10.46303/jcve.2021

[4] Elsje Scott The use of e Learning Systems for pedagogy What Zimbabwean Educators say Conference Paper · December 2016

[5] Zimbabwe Advanced Level Computer Science Syllabus accessed at http://mopse.co.zw/sites/default/files/public/syllabus/Computer%20Science.pdf

[6] David Watson and Helen Williams 2019 Cambridge International AS & A Level Computer Science Hodder Education, London