International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 10 Issue: 02 | Feb 2023 www.irjet.net p-ISSN: 2395-0072

A Review of Virtual Programming Laboratory: Design Issues

Chibuike Ezeaocha Madubuike[1], Augustine Chidiebere Onuora[2], Ezekiel Uchechi Ezeorah[3]

[1, 2, 3] *Department of Computer Science, Akanu Ibiam Federal Polytechnic Unwana, Ebonyi State, Nigeria.*

Abstract - *With the wave of COVID-19 lockdown, most schools have moved their learning support to virtual platforms, while most of these learning systems are there to provide theoretical materials, video lectures, evaluation system and other services to support the classroom work, the practical classes have not enjoyed these level of transition either because limited knowledge in this areas when compared with the classroom or insistence of most school authorities to transition to these platforms. In this study, different implementations of virtual programming laboratory by different schools were x-rayed with a view to understand where each of the proposed system can be best applied. Schools who are yet to provide virtual programming platform can do so to increase the learning curve of their students as well expose them to costly high-end devices from the comfort of their low end mobile devices.*

Key Words: Online Coding, Programming laboratory, Virtual Programming Laboratory (VPL), Cloud-based laboratory, Programming languages

1. INTRODUCTION

This The current traction drawn by IT companies in the global market and the salary derivation of programmers even in the most dwindling economies of the world have pushed a lot of youths around the world and mostly in Africa to venture into programming career. A lot of student from all over the world are annually enrolled into the computing or information technology courses in various institution of higher learning. [1]

The challenge is that so many schools in Nigeria teach programming basically as a white board practice in the classroom with few occasions of hands-on training in the laboratories burdened with few working computers; inversely proportional to the admitted students combined with inconsistent power supply.

Though most of the graduates have excused themselves from skill development in the programming community even graduates of computer science, some of them have continued to leverage in the versatility of mobile devices to position themselves for choice opportunities in the programming community.

Similarly, some institutions have declined their admission rate to only accommodate the number of working computers in programming labs others have continued to proffer alternatives to physical and face to face on-hands training using virtual platform.

In this study, am exploration of some the platforms used by different institutions to meet their demands in teaching programming courses is presented as well as their limitations and strength. The outcome of this study is to help institutions make inform decisions on appropriate coding laboratory that meets their peculiar budget and expected curriculum were necessary.

2. An Overview

2.1 Virtual Programming Laboratory: Terminology and Definitions

Virtual Programming Laboratory (VPL) is a programming laboratory that allow students to perform their programming assignment in remote location outside the specified practical rooms and at no specific lab hours.

The term virtual lab can be traced to the study by [2] in which CS1 and CS2 course were taught in Villanova University at public sites without a reserved room in the school for on-hands training and a specified laboratory hours. This was to overcome the problem of proper classroom facility at that period.

[3] recognized virtual labs as an e-learning activity whose outcome can be both useful at applying knowledge and evaluating knowledge. And it can be offered to both synchronous and asynchronous platform either allowing both the teacher and the learner to be online at a given time or at different times.

Other terms for Virtual Programming Laboratory are online coding, remote coding, coding without computers, mobile touch device coding, virtual programming environments

Most virtual laboratories comprised of two major components namely: the laboratory lessons and the code editor. The teacher prepares and publishes the laboratory lessons for the students to study through on their own. After studying the lessons, the students are expected to practice what they have done by writing the appropriate codes that will demonstrate that a particular programming skill has been acquired.

The aim of the virtual laboratory is to provide a laboratory experience for distance (or remote) students who

may not have the opportunity to practice what they have learnt theoretically or to provide a low cost and efficient virtual platform that will allow schools to do practical on high cost hardware or robots.

In this study we reviewed different implementations of virtual programming laboratory by different schools with a view to exposed the circumstance behind their implementations, their strength and the limitations.

2.2 Virtual Programming Laboratory (VPL) - Design

Several works have been done on the subject of virtual laboratory system. Some of these works include the one by [4] where an in-house virtual lab app called LEFT1 was use to teach PHP server side scripting to students. LEFT1 comprised of two major parts, the administration and application management component and the student workspace component. With LEFT1 which is a web-based application, the students need not to install Apache Web Server and MySQL on the computers since it is installed in the server.

Furthermore, [5] designed and implemented new system of programming parson problem using mobile touch devices. Their new system which falls between rigid scaffolding and real coding exercise tries to eliminate the problem of restricted thinking and limited creativity imposed by the drag and drop system of scaffolding. Using real coding exercise, the students were exposed to the syntax and semantics of programming.

[6] also developed an in-house web application that allows students to perform laboratory exercise on Javascript, HTML and PHP language over the Internet from the comfort of their choice locations. Their system compiles program on a node.js server; where also the submitted program is stored and marked.

[7] proposed the use of VPL in teaching algorithm and programming (APROG). This will involve the integration of VPL into the existing MOODLE of the school. Using the VPL, the scoring and plagiarism test of submitted codes was automated. Though more time were required to set up the VPL environment.

[8] used a game called PROSOLVE to improve the programming skills of students. PROSOLVE combined both gamification and scaffolding techniques. With gamification technique, students efforts to learn programming were rewarded with bonus, points and marks; the scaffolding allowed pseudo-codes to be arranged appropriately by the students. The result of their study proved that including gamification in learning will increase the students enthusiasm to learn; however, the not much attention was given to learning the programming syntax or semantics.

[9] developed an online coding environment, CodingHere, for learning programming. It has some security features, which operates a Linux deamon, to prevent the visitor from running malicious on the server. It could also detect errors and coding patterns in the source code. The environment was built using Go language and C language. It integrates Docker container and some Linux services.

[10] used Google Collaboratory cloud services both to prepare the students on computer programming for the unified state exam; and to teach computer programming course to high school students. Using the interactive notebooks of the Google collaboratory cloud services (GCCS), the teachers can both teach and provide an on hands practical for topics covered in Python language. The result of the study proved GCCS to be effective.

[11] wanted to overcome the challenge that new programmers face learning programming especially children between 8 -16 years, this led them to develop scratch. The scratch environment which is built with squeak language allows new programmers to write programs usually games by simply arranging blocks of codes correctly similar to the LEGO bricks played by most children. Different control structures have different shapes to prove where they can be plugged. This is the same for different operator and arguments of a function.

The scratch project is both an open source project with lots of existing projects on their website; and a community-driven project where members of the community upload and share their project to either get voted for or commented on.

In other to promote computer programming in rural communities in Japan where there are little or no teacher with programming proficiency, [12] developed a rogue-like mathematical game to teach pupils and teachers in elementary school scratch programming. Using the design concepts which covers digital technology, program education and artificial intelligence, the teachers and the pupils where to learn programming by imitating or reproducing an existing game on scratch website. By programming the positions and movements of the objects in the game, the students were to integrate their mathematical knowledge in other to save a princess from a rogue dragon.

The scratch project is both an open source project with lots of existing projects on their website; and a community-driven project where members of the community upload and share their project to either get voted for or commented on. These two concepts have helped to drive the programming motivation amongst young aged children in United States and Europe. However, scratch does not have a mobile version; and is yet to program physical devices [11].

[13] introduced Sololearn as part of their learning tools to their students Zhytomyr Polytechaic State University, Ukraine, in other to over both the challenges of the COVID-19 lockdown and the Russia/Ukraine war. Using Sololearn, the engineering students were taught fundamentals of

programming with emphasis in c language. Sololearn has a mobile version meaning that students don't really need laptop to use it. It has both free and paid PRO version. Sololearn provided both theoretical lessons on several languages with test questions which could either be objective, subjective or scaffold (block-based). With Sololearn code editor, the students practiced what the learnt right from their mobile devices. As a community-based project, Sololearn allowed members to be rewarded and recognized by sharing projects, answering questions, and sending questions. However, just like scratch, Sololearn owns the codes submitted by its users and does not provide privacy of these codes to avoid code plagiarism.

Github is the most popular cloud service for hosting and managing open source software. With Github developers can collaborate on code, manage version, perform bug tracking and integrate additional services. Also Github provides a code editor to run programs online. Due to its ability to track changes in a code repository, most teachers are using Github to track the contributions of individual students in a group programming assignment [14].

Github is currently used by over 500,000 students and 5,000 teachers. [15] conducted a study to identify more features for its increased adoption in the programming community to include its popularity, the high demand of Github skills in the market, the ability to integrate third (3rd) party integrated development environment (IDE) like Eclipse and VSCode, and the possibility of sharing professional portfolio on Github when applying for jobs.

Though Github cannot be classified a full-fledged learning management system (LMS) because of its inability to check for plagiarism, manage student grade book or even track the learning coverage of the student, using the wikis, static sites, the code editor and the version control [15] used Github to teach Java programming, object oriented programming and design, and freshman course on programming. Students were given group projects and also asked to correct other group projects with bugs. In other to protect public view, submissions were made using private repositories. Though [15] did to evaluate if the exercise improved the students performance, the result of their study shows that the students are more likely to continue using Github as most of them were seen using it to run their practical in subsequent semesters.

Robotic programming is used by most nations as the entry point of computer for students in college due to its ability to engage the students more, behaviorally and emotionally, when compared to other forms of programming. However the overhead cost of an average classroom robot kits poses a challenge for poorly funded schools. [16] used iPad software, ARtonomous, to provide a low cost virtual alternative to these high cost learning kits. In addition, reinforcement learning (RL) was introduce to the students to raise the curiosity of the students in studying artificial intelligence, an area of computer science in high demand of skill employees.

Using ARtonomous, the students were to create both a virtual robot, schoolbus; and the track on which the robot will run on. The robot will be tested and trained on the drawn tracks; after which the students were to write codes in swift that will enable the schoolbus robot to convey students along a predefined track. At each point of interaction, the iPad app was always interacting with the remote server for both training of the models and execution of the codes.

In a similar study, [17] used two block-based programming language, scratch and LEGO robots, to program physical robots, and virtual game sprite, respectively. The students were allowed to create projects with both languages and the engagement level of the students was tested. The result of the evaluation proved that both languages can be the used for engaging the students and that the virtual platform proved to be easier for the students.

Most of the authors proposed and implemented theirs using the open source software module, VPL. These allowed institutions which were already using MOODLE to integrate the virtual lab into an environment they were already used to.

3. METHODOLOGY

Several works have been done on the subject of virtual laboratory (or online coding) with the aim of providing virtual situation of a physical programming laboratory. In this study we used both Google Scholar and semantic Scholar to identify thirteen (13) different proposed virtual programming (or online coding) systems dated between 2008 and 2023 from different journals. However, for the sake of simplicity ten (10) of the systems which have a known name were selected for tabular presentation.

The aim of the review is to understand the different implementations of the proposed system with respect to their aim of study, supported language, platform supported, type of course material supported, the type of evaluation provided by these systems and the service on which they are hosted.

3.1 Findings

The table below highlights the various proposed virtual laboratory system used by the different authors reviewed:

Reviewed papers									
S/N	**Year**	**Author(s)**	**Aim of study**	**Proposed System**	**Language support**	**Platform support**	**Course materials**	**Type of evaluation**	**Hosting Service**
1	2008	Bucus, Dragulescu and Termaucu	To allow students test and implement web based apps written in PHP	LEFTI	PHP	web	N/A	Project based	Internet
2	2017	Cardoso, Barroso and Castro	To teach fundamentals of programming	Incorporation of VPL plugin into MOODLE	Support languages whose translator can be installed in the jail server.	web	Teacher-defined	Project based	Wifi
3.	2017	Syahpuri and Nahal/	To teach web programming languages	VP-Lab	PHP, and JavaScript -hard coding	web	N/A	Project based	wifi
4	2018	Angulo and Aktunc	To Java programming to users	Github	C, C++, Go, Java, JavaScript, PHP, Python, Ruby, Scala and TypeScript. Also for languages that support package manager	Desktop, mobile and web	-teacher defined	Project based with version control	Internet
5	2019	Mathew, Malik and Tawafak	To teach problem solving skill	PROSOLVE GAME	Pseudocode learning using brick game	web	N/A	Project based	Wifi
6	2022	Markelov and Zavyalova	To prepare college students for unity exams	Google Colaboratory (Google Colab)	Over 40 languages including Python, R, and Java.	Web, desktop and mobile	Teacher defined	Project based with version ontrol	Internet
7.	2022	Wang	To help provide programming platforms for poorly funded schools in Japan	Rogue game using scratch	Scratch	web	Platform defined	Project based	Internet
8	2022	Wang, J. Liang, J. & Chang	To create an online judgement system with additional features to aid students in programming	CodingHere	Python, C and C++	web	Teacher-generated	Random generated test case	Internet
9	2022	Dietz, et al	To overcome the overhead cost of robot kits and to buils their enthusiasm in artificial intelligence	ARtonomous	Swift	mobile	N/A	-project based	-wifi
10	2023	Vakaliuk, et al	To support students during the lockdown and the Ukraine/Russia war	MOOC sololearn`	JavaScript, CSS, HTML,\C, SQL, PHP, Ruby, Swift 4, C#, jQuery, Java, C++, Kotlin, R, Go	Mobile	Platform approved	-multichoice -sub theory -test case -project based	Internet

4. Discussion

From table1 above, it can be observed that ten (10) different solutions has been identified; this include: LEFTI, integration of virtual programming laboratory (VPL) plugin with modular object-oriented digital learning environment (MOODLE), Github, Massive open online courses (MOOCS) Sololearn, Google colaboratory (colab), ARtonomous, CodingHere, Game using Scratch, PROSOLVE game, and VP-Lab.

Apart from the PROSOLVE and the scratch game which uses code blocks, the rest were introduced to allow the new programmer learn programming by hard coding the program syntax.

When it comes to evaluation which is a critical component of learning, systems like Github, Google Colab, ARtonomous, and LEFTI adopts asynchronous evaluation using project based evaluation such that the students will be evaluated manually by the teacher after submission of project that been made. In addition, Github and Google Colab uses version control to determine where a code has been changed. With this the teacher can give group assignment and monitor the contributions of each member of the group.

However, CodingHere, Sololearn, and VPL plugin automatic evaluation of students work using test case exercises. The questions are set and the test cases are submitted to the judgment systems to automate the evaluation process. This will help the student to get instant feedback as well as to appreciate his/her effort in the exercise. Sololearn even took it further by introducing multi-choice questions, block-based questions and subjective questions. With this the students' knowledge of the course is wholly tested.

The choice of platform is a major determinant when it comes to programming because most of the programmers are young people who would always be with their phones. So getting something that works with the phone is very crucial. Software can be developed to support three (3) main platforms: mobile, web, and desktop. Six of the solutions (CodingHere, VPL plugin, PROSOLVE, LEFTI, Scratch game and VP-Lab) reviewed support only web; ARtonomous support mobile (ipad); while three others (Github, Google Colab amd Sololearn) support all three platforms (mobile, web and desktop).

There are two categories of language support identified in these system: the pre-installed language solutions (VP-Lab, PROSOLVE, LEFTI, Sololearn, Scratch game, CodingHere and ARtonomous); and the user-installed solutions (VPL plugin, Google Colab and Github). The pre-installed language support solutions is easy to use because it does not require the user to set the language environment; however, has a fixed set of programming language that it can support. The user-installed solutions comes with some initial setup difficulties because the user have to setup the coding environment on the cloud (or Internet); however, it allows as many language as the user wish to setup. PHP is the most supported language; while Swift and Scratch are the least.

LEFTI, PROSOLVE, VP-Lab and ARtonomous does not have provision for course material. Sololearn is the only solution in the ten that comes with predefined course materials; others uses the materials provided by the teacher.

5. CONCLUSIONS

The need for a virtual programming laboratory can be seen in its attempt to reduce the burden of buying costly hardware materials, the allow the students to continue doing programming exercises outside the physical laboratory, to allow the students get automatic feedback on his/her coding exercises; to allow the teacher to remotely track the contributions of a each student in a group assignment; or to eradicate the need for a student to set up the development environment from his local device. The need to also introduce the concept of redundant component when developing the virtual laboratory can be helpful especially when there are network failures for whatever reason. [18].

In this study, several journal articles on virtual programming solutions were review out of which ten was selected.

ACKNOWLEDGEMENT (Optional)

The authors wish to acknowledge Tertiary Education Trust Fund (TETFUND) for their grant toward the actualization of this research.

REFERENCES

[1] J. C. Alejandrino, "IT student's selection and admission analysis using naive Bayes and C4.5 algorithm." International Journal of Advanced Trends in Computer Science and Engineering, vol 9, no. 1, 2020, pp. 759-765, doi:10.30534/ijatcse/2020/108912020.

[2] D. T. Joyce, "A virtual lab to accompany CS1 and CS2", Proceedings of 21st SIGCSE in technical symposium on computer science education, pp. 40-43, (1990).

[3] S. Kirange & D. Sawai, "A comparative study of e-learning platforms and associated online activities", the online journal of disease, education and e-learning, Vol 9(2), pp. 194-199, 2021.

[4] M. C. Bucus, B. Dragulescu & A. Termaucu, "Developing virtual lab at "politehnica" university of

Timisoara", Interactive conference in computer aided learning, 2008.

[5] P. Ihantola, J. Helminen & V. Karavirta, "How to study programming on mobile touch devices: interactive python code exercises", Proceedings of the 13th Koli calling international conference on computing educational research, pp. 51- 58, 2013.

[6] R. Syahpuri, & S. R. Nahal, "VP-Lab: a virtual way to stay connected with programming laboratory.", (2017).

[7] Cardoso, Barroso & Castro, "Virtual programming labs in computer programming learning process, prepare a case study", proceedings of EDULEARN17 conference, 3-5 July 2017, Barcelona Spain.

[8] R. Mathew, S. I. Malik, & R. M. Tawafak, "Teaching problem solving skills using an educational game in a computer programming course, informatics in education", Vol 18(2), pp.359-373. 2019.

[9] J. Wang, J. Liang, & C. Chang, "The CodingHere platform for programming course", Information engineering express, international institute of applied informatics, vol 8(2), pp.1- 14, 2022.

[10] V. K Markelov & O. A Zavyalova, "Tools of the Google Collaboratory Cloud Environment for programming training and the preparation for the unified state exam informatics", 2022.

[11] M.Resnick, J. Maloney, A. M. Hernandez, Rusk, Eastmond, E. Silverman, & Y. Kafai, (2009). Scratch: programming for everyone, communications of the ACM, 54(11), 60-67.

[12] T. S. Wang, "The ability of scratch teaching in programming language- taking rogue like mathematics games as an example.", Technium, vol 4(10), pp. 217-238. 2022.

[13] T. A Vakaliuk, O. V. Chyzhmotria, O. H. Chyzhmotria, Didkivska & V. V. Kontsedailo, "The use of massive open online courses in teaching the fundamentals of programming to software engineers, educational technology quarterly", pp.1-16, 2023.

[14] O. G. Glazunova, O. V. Parhumenko, V. I. Korolchuk, & T. V. Voloshyna, "The effectiveness of Github cloud services for implementing a programming training project: student's point of view", 2021.

[15] M. A. Angulo and O. Aktunc, "Using Github as a teaching tool programming courses." Proceedings of the 2018 ASEE Gulf-Southwest Section Annual conference the university of Texas at Austin, April 4-6, 2019, pp. 1-4.

[16] G. Dietz, J. K Chen, J. Beason, M. Tarrow, A. Hillard, & R. B. Shapiro, "ARtonomous: introducing middle school students to reinforcement learning through virtual robots", IDC'22 interaction design and children, June 27-30, 2022, Braga, Portugal.

[17] C., Qin, Y. Liu, & Zhang, "Scratch versus LEGO robots: which engages undergraduate more in programming education?", Journal of computer assisted learning, 2023.

[18] A. C. Onuora, P. Ana, U. N. Nwanhele and O. J. Idemudia. "Improving Software Quality by Developing Redundant Components." International Research Journal of Engineering and Technology (IRJET). Vol 7, pp. 151-155, 2020.