# A novel method for establishing virtual laboratory for scientific and engineering curriculums to be applied in developing countries

Agieb, Ramy Said; Amer, Ayman; Ababneh, Jafar; Solyman, Ahmed; Hafez, Mohamed; Fawzy, Mohamed

*Published in:*
2023 2nd International Engineering Conference on Electrical, Energy, and Artificial Intelligence (EICEEAI)

*DOI:*
10.1109/EICEEAI60672.2023.10590602

*Publication date:*
2024

*Document Version*
Author accepted manuscript

# A Novel Method for Establishing Virtual Laboratory for Scientific and Engineering Curriculums to be Applied in Developing Countries

Ramy Said Agieb
*Department of Electrical Engineering*
*Faculty of Engineering, MTI University*
Cairo, Egypt
drragieb@eng.mti.edu.eg

Ayman Amer
Faculty of Engineering
Zarqa University, Zarqa, Jordan
College of Engineering, University of Business and Technology, Jeddah, 21448, Saudi Arabia
Aamer@zu.edu.jo

Jafar Ababneh
The Faculty of Information Technology, *Cyber security,*
*Zarqa University,* Zarqa, Jordan
jababneh@zu.edu.jo

Ahmed Solyman
*School of Computing, Engineering and Built Environment*
*Glasgow Caledonian University*
Glasgow, UK
ahmed.solyman@gcu.ac.uk

Mohamed Hafez
*Faculty of Engineering FEQS*
*INTI-IU-University*
*Nilai, Malaysia*
mohdahmed.hafez@newinti.edu.my

Mohamed Fawzy
*Department of Electrical Engineering*
*Faculty of Engineering, MTI University*
Cairo, Egypt
mohamed.fawzy@eng.mti.edu.eg

***Abstract*—Modern students must adapt to dynamic circumstances in the ever-evolving educational environment. Creative and streamlined ways are necessary to acquire and comprehend academic knowledge properly. This work aims to fulfill this requirement by presenting an offline virtual laboratory (VL) developed explicitly for practical subjects. The selection of an offline solution is especially relevant for implementation in developing nations, where internet connectivity is frequently unreliable or inaccessible in several areas. The Virtual Learning (VL) system being suggested utilizes the capabilities of the C# programming language within the Unity3D cross-platform development environment. The main objective of this study is to streamline the process of obtaining and comprehending scholarly knowledge, ensuring its availability to a diverse student population, irrespective of age or educational history. This methodology guarantees the laboratory's applicability across multiple platforms and compatibility with smart devices, enhancing its versatility and broad accessibility. The study resulted in the creation of a dynamic virtual laboratory that has the potential to be seamlessly incorporated into educational curricula, including various disciplines, including physics, mechanics, and engineering. This project aims to improve the educational experience for students in developing nations and promote their involvement in both practical and theoretical elements of their studies by providing a comprehensive offline solution that is widely compatible.**

***Keywords—Virtual labs; e-Learning; C#; Unity3D; Smart Devices.***

## I. Introduction

Amidst the current global issues, such as the ramifications of diseases like COVID-19 and persistent conflicts in different areas, the globe is contending with a convergence of crises[1]. Concurrently, there is a notable surge in the population of developing nations, leading to pressure on existing resources and infrastructure. Furthermore, the present state of the global climate is characterized by decreased spending allocated towards education, posing challenges for numerous pupils regarding their ability to attain entrance to conventional educational institutions. The emergence of various challenges has led to the development and deployment of innovative solutions, such as Virtual Laboratories (VL), to address this complex issue [2].

In addition to these difficulties, the swift and continuously progressing technological developments in recent times have necessitated the utilization of technology in education [3]- [5]. Virtual laboratories are crucial in facilitating technology integration within education. Interactive learning is a pedagogical strategy that effectively incorporates social networking and urban computing into the design and delivery of courses, facilitating a seamless educational experience. This approach prioritizes active student involvement throughout the educational process, promoting an interactive and dynamic learning experience rather than a passive one[6].

There are various and multidimensional advantages associated with the utilization of Virtual Laboratories. The subsequent sections emphasize several advantages that underscore the crucial role of virtual learning (VL) in addressing the problems presented by the current global context and future educational requirements .[7]

Virtual Laboratories (VLs) play a pivotal role in addressing various challenges and providing a myriad of benefits in the realm of education:

1. VLs alleviate financial constraints and effectively manage the growing number of classroom students.
2. They promote equitable comprehension of educational content, bridging the gap in regions with limited access to well-trained and well-equipped teachers.
3. VLs empower learners to engage in various experiments that may be challenging or unsafe to conduct in traditional laboratories.
4. They optimize time and cost by eliminating the need for physical presence in laboratories during specific hours and reducing travel between locations.
5. Learners and educators gain access to cutting-edge technology through VL platforms.

6. VLs enable students to perform practical experiments that complement their theoretical coursework, enhancing their understanding of the subject matter.
7. The interactive and engaging nature of VLs makes learning enjoyable and captivating during experimentation[8].
8. Learners can repeat experiments multiple times to reinforce their understanding and improve their skills.
9. VLs eliminate physical exposure to hazardous substances and dangerous devices, ensuring the safety of both students and instructors.
10. They offer the flexibility to modify experiment parameters and procedures without worrying about adverse consequences.

In summary, Virtual Laboratories provide a versatile and safe environment for learning, offering numerous advantages that enhance the educational experience and mitigate various challenges[9].

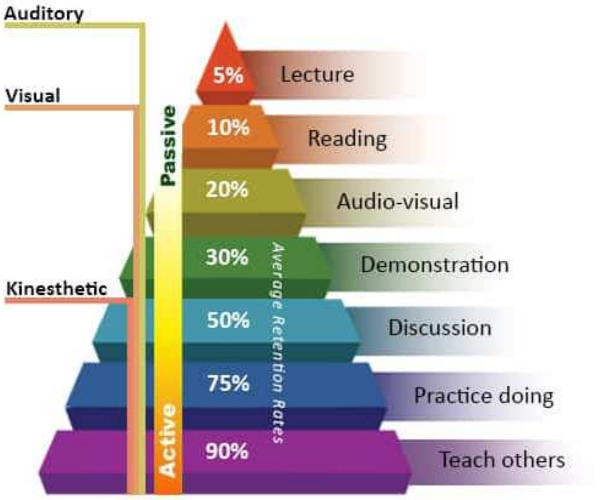

Fig. 1. Learning pyramids for retention and recall of information

This paper is structured as follows: Section 2 overviews related works. Section 3 outlines the selection of the appropriate platform. Section 4 delves into the discussion of the proposed model. Section 5 presents the results, while Section 6 encapsulates the research's conclusion.

## II. Related Works

Numerous VLs have been established and manifest in various forms, most accessible online. The NEWTON project is an exemplary European endeavor in this realm, which primarily caters to European adolescents seeking access to online science laboratories. Under this initiative, students can initiate their laboratories and partake in collaborative learning. Additionally, LiLa represents another VL project, offering 3D assets for facilitating online collaboration among learners [10]. VccSA, another VL project, incorporates practical simulations and exercises[24], allowing educators and students to create accounts and embark on the learning journey [11].

Furthermore, the Random project operates as a web portal, providing various web-based educational materials in mathematics [12], [13]. Open-source physics stands out as an extensive resource, offering hundreds of open-source virtual laboratories and associated curricular content spanning the domains of physics, computing, and computer modeling [14], [15]. Nmsu Virtual Labs contributes to the landscape by presenting fundamental laboratory procedures and practical analytical assignments, encompassing tasks such as Corn Mould testing, Gram Staining, and Water Activity control in food [16].

The Technical University of Madrid has developed a practical, experiential learning program for students in various fields, including topography, physics, electronics, and chemistry, known as GridLabUPM. This Virtual Laboratory (VL) was created utilizing the open-source software OpenSim [17]. Numerous subject-specific collections of VLs exist, including ChemCollective [18]. This resource focuses on chemistry education and incorporates virtual laboratories, scenario-driven learning exercises, lectures, and conceptual assessments. Educators can incorporate these materials into their curricula, offering an alternative to traditional textbook assignments and engaging students in in-class activities, individually or collaboratively. Notably, these virtual labs encompass online chemistry lab simulations, enabling students to select from a wide range of standard aqueous reagents and manipulate them as if they were in a physical laboratory without the added expenses or potential hazards associated with actual equipment[19].

One of the primary objectives of VL initiatives is to encourage inquiry-based learning, a feature already present in many VLs. For instance, students gain practical experience in control engineering fundamentals through experiments conducted within the TriLab project [20], covering key aspects such as the essential components of a feedback loop and the intricacies of Proportional-Integral-Derivative (PID) tuning. Additionally, VLs utilize LabVIEW to seamlessly integrate hands-on, virtual, and remote laboratories within a unified software package.

## III. Related Works Selection of The Appropriate Platform

VLs are developed utilizing platforms that are supported by distinct programming languages. The selection of these components is a crucial step that necessitates carefully evaluating numerous complex aspects to guarantee that virtual laboratories (VLs) are operational and efficient in fulfilling their educational objectives. The selection process encompasses various essential stages, encompassing the decision about the programming language, identifying relevant tools, and evaluating the optimal platform for the given educational setting. In the following sections, the elements that impact and direct the selection process of programming languages, tools, and platforms will be explored. These aspects eventually shape the development of virtual learning environments (VLs) to achieve their educational goals [21].

### A. Selection of the programming language

Given the plethora of programming languages available, each possessing its unique capabilities, it became imperative to establish the criteria that would inform the selection process. The following factors have been identified for this purpose:

*1. Imperative software (IS)*

Imperative software (IS) comprises instructions that direct smart devices to execute tasks, akin to the imperative mood in human languages used to convey commands. The objective of imperative programming is to delineate the operational logic of a program. Implicit programming, on the other hand, is a programming paradigm that utilizes statements to modify a program's state.

*2. Structure software (SS)*

Structured software (SS) can store various data types together, particularly valuable in experiments involving multiple parameters [40]. Additionally, structured language is known for its ease of maintenance.

*3. Automatic Memory Management (AMM)*

Automatic Memory Management (AMM) refers to a programming language's capability to release memory blocks that are no longer in use, which plays a crucial role in optimizing memory usage and ensuring that a program runs efficiently. This feature allows the programming language to automatically identify and deallocate memory that is no longer needed, preventing memory leaks and enhancing the overall performance of applications. This aspect is essential in avoiding memory-related issues and maintaining the stability and reliability of software[22].

*4. Object-oriented programming (OOP)*

Object-oriented programming (OOP) is about constructing structures that include data and functions, while procedural programming is about writing procedures or tasks that execute operations on the data. OOP is a more efficient and straightforward method of programming. The services have a simple framework thanks to OOP. OOP makes it easy to manage, alter, and debug. OOP allows developers to build fully reusable software with fewer codes and less time.

*5. Functional programming (FP)*

Functional programming (FP) is a fundamental paradigm utilized in software engineering to design software systems. The defining feature of this approach is the representation of function descriptions as hierarchical structures composed of expressions that perform assignments of values to other values. The FP diverges from the imperative paradigm in which a series of assertions alters the operational state of the program. Functional programming (FP) emphasizes the development of a declarative programming paradigm, which prioritizes the interconnections between values and their corresponding transformations. This philosophical approach results in software frequently being more amenable to logical analysis, testing, and upkeep. Functional programming (FP) is founded on computation as the assessment of mathematical functions, with the objective of circumventing state alteration and mutable data. This approach leads to code that is characterized by enhanced predictability and reliability.

*6. Procedural programming (PP)*

Procedural programming (PP) is a programming paradigm based on the fundamental concept of procedure calls. This programming paradigm is an advancement from imperative programming, which focuses on utilizing procedures, commonly called routines or subroutines, to organize and structure code. Procedures are a collection of computing instructions that must be carried out to achieve a particular objective. Within the runtime of a program, it is possible for any process to be called at any given moment, either by other procedures or even by itself, so establishing a hierarchical arrangement. This methodology facilitates the modularization of code, dividing it into smaller, more easily controllable components. Each process can fulfill a distinct function, enhancing the comprehensibility and manageability of the code. Procedural programming emphasizes the systematic execution of instructions sequentially, making it well-suited for addressing problems that can be decomposed into a series of clearly defined tasks. It is imperative to acknowledge that procedural and object-oriented programming (OOP) are fundamentally different. In OOP, data and the corresponding methods that manipulate that data are enclosed into objects. In contrast, procedural programming emphasizes the functions and their interactions with data, frequently employing a global scope to facilitate data accessibility.

*7. Reflective programming (RP)*

Reflective programming (RP) is a methodology that authorizes procedures or functions based on types or structures that can be created later. These procedures are only instantiated or run when particular kinds or structures are supplied as parameters. Within RP (short for "programming paradigms"), the primary emphasis lies in establishing abstract or generic procedures not inherently linked to particular data types or implementations. However, these methods have been specifically built to possess adaptability and the ability to function with diverse data kinds or structures that adhere to specific requirements. This method facilitates more flexibility and adaptability in the realm of software design. This approach proves to be highly advantageous when confronted with intricate systems in which the precise characteristics of the data or structures to be processed are not ascertainable during the initial stage of procedure definition. Reactive programming (RP) is characterized by its inclination towards an abstract and elevated approach to programming, prioritizing examining process interactions and behaviors over their specific implementations. One significant benefit of employing reflective programming techniques is the potential to augment code reusability and boost the adaptability of software systems in response to evolving needs. This feature is frequently utilized in programming languages or dynamically typed environments, allowing for determining types during runtime. Reflective programming offers a methodology for crafting methods that possess the flexibility to be expanded and altered, hence accommodating the incorporation of novel data kinds or structures as necessitated.

*8. Event-Driven Programming (EDP)*

Event-driven programming (EDP) is a software design approach where events determine the program's execution. These events can encompass various actions, such as user interactions like mouse clicks, keyboard inputs, sensor readings, touchscreen interactions, messages received from other programs, and interactions with ports and devices[23].

Table 1 compares the most widely recognized programming languages and assesses whether they meet the specified criteria. According to the data in Table 1, it is evident that C# satisfies all the necessary parameters for developing Virtual Laboratories (VLs). Therefore, the choice of the programming platform is significantly influenced by using C#.

TABLE I THE CHARACTERISTICS OF THE MOST FAMOUS PROGRAMMING LANGUAGES

| Language | IS | SS | AMM | OOP | FP | PP | RP | EDP |
|---|---|---|---|---|---|---|---|---|

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| C++ | √ | √ | X | √ | √ | √ | X | X |
| C# | √ | √ | √ | √ | √ | √ | √ | √ |
| Java | √ | X | √ | √ | √ | √ | √ | √ |
| Python | √ | X | √ | √ | √ | √ | √ | √ |
| Dart | √ | X | √ | √ | √ | √ | X | √ |
| MATLAB | √ | X | X | √ | X | X | X | X |

### *B. Selection of the suitable Tool*

Platforms can be broadly categorized into two main types: Native and Cross-platform. Native platforms are primarily employed to develop applications compatible with a single operating system (OS), while Cross-platform solutions are utilized to create applications that can run on multiple OSs. For a detailed comparison between Native and Cross-platform platforms, please refer to Table 2 , [24]

TABLE 2 THE CHARACTERISTICS OF THE MOST FAMOUS PROGRAMMING LANGUAGES

| Issues | Native | Cross-Platform |
|---|---|---|
| Architecture | Changed code for different OSs | One code for multiple OSs |
| Cost | High cost of progress | Relatively low cost of progress |
| Code Reusability | Almost no code works for a single platform | Single code can be used on multiple platforms |
| Hardware Access | Complete hardware accessibility | Limited access to all device |
| UI/UX | Consistent platform-specific UI/UX | Unified UI/UX for all platforms with limited consistency |
| Performance | Seamless, fully native performance | High-performance, native-like |
| Target Audience | Limited to a specific platform | Reach a vast number of developers |
| Time to Market | Extra time is required as different code needs to be written. | Significantly reduced time to market due to a single codebase |
| Team Size | Extensive (different resources for different platforms) | Small (1 for all platforms) |

According to Table 2, cross-platforms are more suitable for initializing VLs. Many applications depend on cross-platform to save time and money. Table 3 maps the famous type of Cross-platform and the programming language used by each one. The programming language plays a vital role in selecting the platform used in VL; it must satisfy some conditions, as mentioned in the previous section. Based on sections A and B, the most suitable platform for producing offline VLs is Unity3D.

TABLE3 CROSS PLATFORMS AND THEIR PROGRAMMING LANGUAGES

| Programming Language | Platform |
|---|---|
| Java | React Native, Cordova, Ionic, NativeScript, Appcelerator |
| JavaScript | Kivy, BeeWare |
| C# | Unity3D |
| C++ | Qt |
| Ruby | RubyMotion |
| Dart | Flutter |
| Java | B4A |
| Basic | Codename One |

### *C. Unity3D*

Unity3D stands out as a potent cross-platform development engine that provides developers with the tools to create interactive 3D content and develop games. This platform allows you to seamlessly integrate art and assets to construct scenes and environments, incorporating features such as special effects, physics, animations, lighting, and more, as outlined in [25]

- The additional script editors, MonoDevelop (available for Mac and Windows) and Visual Studio (for Windows) can script without requiring additional resources.
- The Graphical User Interface (GUI) system facilitates easy design and testing.
- The 3D terrain editor enhances applications and labs' visual appeal and descriptive quality.

Unity3D comprises essential components, namely the Toolbar, Scene View, Hierarchy, Project Window, and Game View, each with distinct roles:

- Toolbar: Equipped with various tools for modifying scenes and game windows.
- Scene View: Presents a complete three-dimensional representation of the active scene, allowing the addition, modification, and removal of GameObjects.
- Hierarchy: Displays a list of all Game Objects within the active scene.
- Project Window: Useful in complex games, this component searches for specific game elements as required, combing through the project's assets directory for Models, Scripts, Prefabs, and Textures.
- Game View: Unity permits real-time gameplay observation and simultaneous modifications to the game during play.

When designing VLs within Unity3D, there are several crucial practices to consider, such as [22]:

- Optimize Component References: Keep component references consistently cached to streamline script usage.
- Efficient Memory Allocation: Instead of instantiating new objects on the fly, consider constructing and utilizing object pools. This approach minimizes memory fragmentation and simplifies the task of the garbage collector.
- Layer Management and Collision Matrix: Create a new column and row for each additional layer to determine layer interactions in the collision matrix.
- Effective Raycasting: Utilize ray casts to project a ray in a specific direction and length, receiving notifications upon hitting objects.
- Appropriate Physics Engine Selection: Choose a physics engine that aligns with your game type, whether 2D or 3D.
- Rigidbody Implementation: Incorporate rigid bodies when establishing physical interactions between in-game elements.
- Fixed Timestep Adjustment: The fixed timestep setting impacts the fixed update () and physics update rates.

## IV. THE PROPOSED MODEL

It is essential to assemble the following elements To develop the theoretical aspect: a Lesson Plan, Learning

Objectives, Templates, Content, Media, Activities, and Assessments. The Lesson Plan entails delineating the lesson's subject matter and facilitating the instructional process, functioning as a fundamental framework for other elements. The templates, content, and media elements are derived from the lesson plan and utilized in the practical phase of virtual learning development.

### A. *The Theoretical Part*

Indeed, developing the theoretical part necessitates the preparation of several key components: Lesson Plan, Learning Objectives, Templates, Content, Media, Activities, and Assessments. The Lesson Plan involves documenting the lesson's content and the responsibilities that guide the teaching process. This Lesson Plan is the foundation from which the other components are derived. Templates, Content, and Media are extracted from the Lesson Plan and utilized in the practical part of VL production.

### B. *The Technical Part*

The practical component of the design entails dividing the VL application or program into distinct stages, as depicted in Figure 2.

The initial phase, the Interactive Graphical User Interface (IGUI) or front end, is depicted as block number one in Figure 2. This stage has a range of interfaces, including buttons, displays, text fields, timers, navigation buttons, and other components. The IGUI should consider several important factors, such as the allocation of unique colors to individual sections, the maintenance of suitable spacing between parts, the determination of the master controller's placement, and the input of data inside the designated thumb region.

The second phase, referred to as the Experiment Controller Script Interface (ECSI) or back end, as seen by block number two in Figure 2, is responsible for overseeing the sequence of steps in the experiment and the overall action-reaction process that occurs during the experiment. ECSI carries out its operational activities utilizing blocks numbered three, four, five, six, seven, and eight. The selection of experiment components by ECSI is tailored to the individual requirements of each experiment, as not all studies necessitate the same components. Block 3 covers the navigation component, which facilitates the smooth transition between various experiments and the initiation, cessation, and reloading of the entire experience. Block 4 encompasses the scripts that manipulate assets utilizing movement and rotation, facilitating targeted alterations in their respective positions. Block 5 executes text-related operations, encompassing activities such as inputting numerical data, managing variables, and presenting output. Block 6 is utilized in situations where the use of tools such as timers, counters, rulers, or weights is required. The sound and animation functions located in Block 7 are utilized to communicate particular actions or document outcomes in specified regions. Block 8 comprises a set of inquiries and corresponding responses employed to evaluate students' performance, hence facilitating the identification of specific areas where pupils may require further development. The score tracking is also implemented within Block 7 to monitor students' development. Additionally, a reset feature is incorporated to enable the score to be reset to zero as required.

## V. APPLY THE PROPOSED MODEL

Virtual Laboratories (VLs) should be meticulously crafted to encapsulate and illustrate diverse scientific topics. This should be achieved by commencing with a straightforward statement and presenting the information lucidly and succinctly. Figure 3 illustrates an interpretation of Newton's first law of motion. In this experimental setup, the participant engages with two distinct cases positioned at the upper and lower sections by the actuation of a force button. Both instances pertain to the presence of a moving object. Nevertheless, Force 2 surpasses Force 1 in the lower scenario, leading to the object reaching its destination quicker than in the upper scenario. The animation and an auditory alert are activated when an impediment is encountered, as depicted in Figure 3.

To offer a comprehensive comprehension, an exemplification of an extensive experiment is presented in Figures 4 and 5. Figure 4 presents the theoretical component of the experiment, whereas Figure 5 depicts the actual component. In the present study, participants can quantify the elongation of a spring by affixing various masses to a mechanical device known as a crane. The measurement of the spring's extension is documented concerning the applied weight, as depicted in Figure 5. These examples demonstrate the ability of virtual laboratories (VLs) to encompass both theoretical principles and actual implementations, thus enriching the educational experience of students.

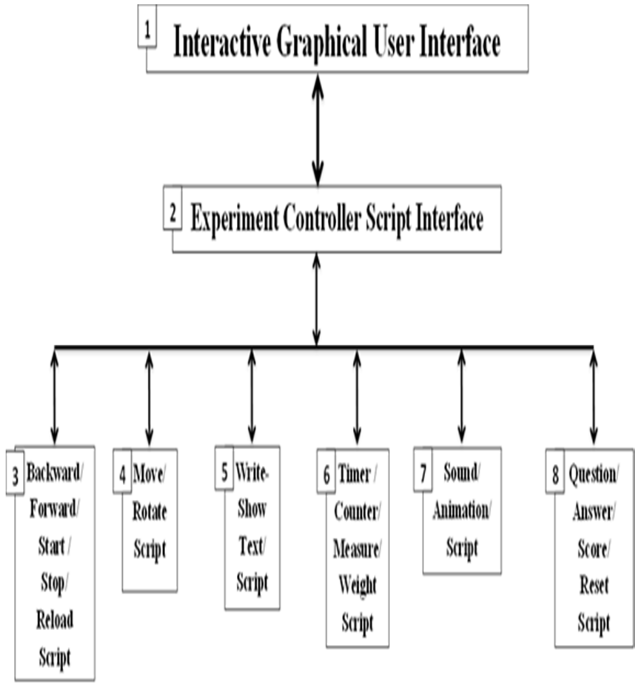


Fig. 2. The proposed practical model

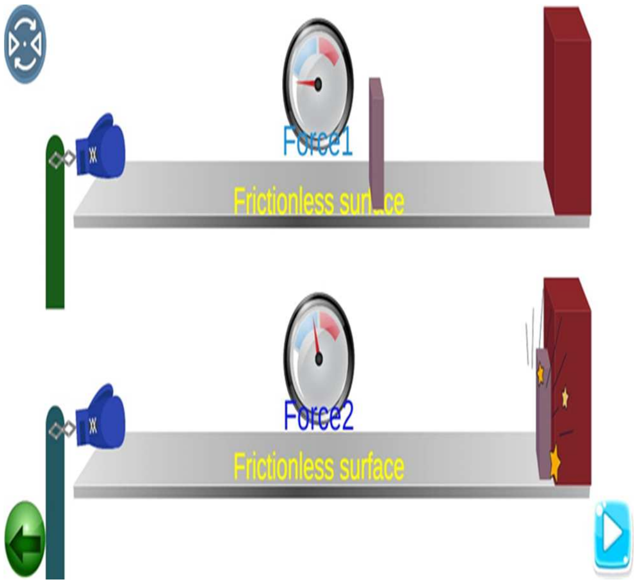

Fig. 3. Interpreter of Newton's first law of motion

Number footnotes separately in superscripts. Place the actual footnote at the bottom of the column in which it was cited. Do not put footnotes in the abstract or reference list. Use letters for table footnotes.

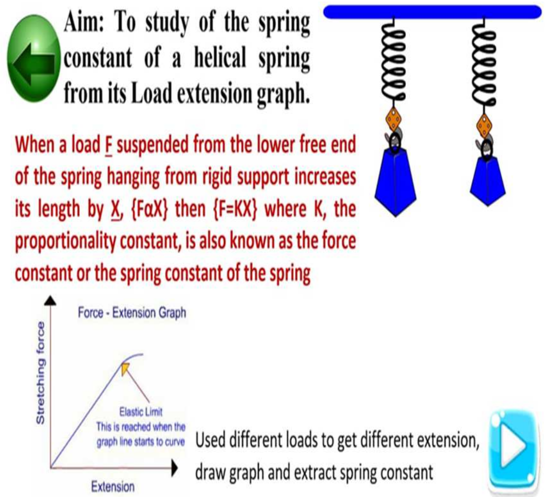

Fig. 4. The theoretical part of the experiment

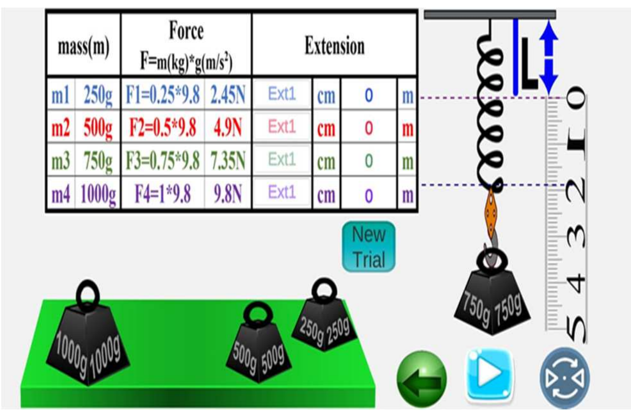

Fig. 5. Practical part of the experiment

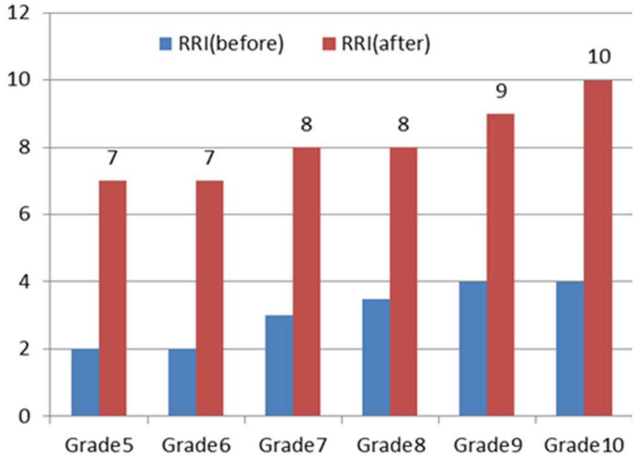

Fig. 6. Average PRI before and after using VL.

## VI. Results

To assess the efficacy of the planned Virtual Laboratory (VL), six subjects were selected from the grade ten physics curriculum within the government educational framework. The themes mentioned above were utilized as exemplars for developing virtual learning modules. The Virtual Learning (VL) system was evaluated using a sample of students enrolled at the Integrated Distinct Governmental Language (IDGL) School in Menoufia Governorate, Egypt. The assessment was conducted during a programming workshop hosted by the Misr El-Kheir Foundation in June 2021. The kids chosen for the study encompassed a range of grade levels, specifically from fifth to tenth grade.

The pupils' Relative Readiness Index (RRI) was measured before and after utilizing the Virtual Learning (VL) platform. As illustrated in Figure 6, the findings demonstrate a noticeable enhancement in RRI values after using the VL. This outcome was expected to occur to a certain degree, given that the virtual learning (VL) platform was specifically developed to improve the educational process. Nevertheless, the extent of the improvement, particularly in lower grade levels relative to the target grade level, was noteworthy. The achievement can be ascribed to the student's capacity to comprehend and actively participate in the educational material using a tool that aligns with their experiences and preferred learning methods.

## VII. conclusion

In conclusion, the education field is facing unprecedented challenges in effectively teaching and communicating curriculum information to the present generation of students. This is primarily due to the rapid and continuous technological developments that have shaped today's students' learning preferences and capabilities. Virtual laboratories (VLs) have emerged as one of the most effective solutions to address this problem, particularly the offline type, which is more suitable in developing countries with limited internet access. This paper has presented a specific method for selecting the appropriate platform to create VLs, focusing on Unity3D, a well-known cross-platform tool commonly used for developing games and applications for a wide range of operating systems. The proposed VL was successfully created utilizing Unity3D.

The response from students during the experimental stage has demonstrated the significance and relevance of VLs in current and future educational settings. The proposed VL was adequate for the academic year for which the curriculum was intended and showed effectiveness in younger grades. This outcome highlights the potential of VLs to extend beyond traditional grade boundaries, providing a versatile tool for enhancing students' access to and understanding of educational information. VLs can significantly augment the volume of information that students can study and comprehend, making them a valuable asset in modern education.

## References


[1] Alison Reida, Jonathan Cook (2020). "Developing management effectiveness: The nexus between teaching and coaching," The International Journal of Management Education Volume 18, Issue 1

[2] Kanesan Muthusamy. (2005). “Virtual Laboratories in Engineering Education," Asian Journal of Distance Education, vol 3, no 2, pp 55 – 58

[3] M. Firman Ramadhan. (2017). “Using virtual labs to enhance students' thinking abilities, skills, and scientific attitudes," International Conference on Educational Research and Innovation (ICERI 2017)

[4] Huda Mohammad Babateen. (2011). "The role of Virtual Laboratories in Science Education," 5th International Conference on Distance Learning and Education

[5] Galya Shivacheva. (2017). “Designing a Virtual Laboratory for Teaching Programming," International Conference on Computer Systems and Technologies - CompSysTech

[6] Prieto-Blazquez. (2009). “A Virtual Laboratory Structure for Developing Programming Labs,” International Journal of Emerging Technologies in Learning, Vol 4

[7] Galya Shivacheva. (2017). “Designing a Virtual Laboratory for Teaching Programming," the 18th International Conference on Computer Systems and Technologies, June

[8] Manisha Bajpai. (2013). " Developing Concepts in Physics through Virtual Lab Experiment: An Effectiveness Study," An International Journal of Educational Technology, Techno LEARN: 3 (1): 43-50. June

[9] LiLa. (2011). "Library of Labs" Available at: www.lila-project.org/Project

[10] VccSSe. (2009). "Virtual Community Collaborating Space for Science Education" Available at: www.vccsse.ssai.valahia.ro/

[11] Random. (2019). "Virtual Laboratories in probability and statistics". Available at: http://www.math.uah.edu/stat/

[12] Open Source Physics. (2019). "National Science Foundation and Davidson College, North Carolina, US". Available at: http://www.opensourcephysics.org/

[13] USDA CSREES. (2021) (Cooperative Research and Extension Services) and USDA National Institute of Food and Agriculture. Available at: http://virtuallabs.nmsu.edu/)

[14] D. Fernández-Avilés, D. Dotor, D. Contreras, J.C. Salazar. (2016).”Virtual labs: a new tool in the Education. 13th International Conference on Remote Engineering and Virtual Instrumentation (REV),”UNED, Madrid, Spain, February

[15] OpenSim. (2015). "https://simtk.org/docman/?group_id=91”2"

[16] Chem Collective. (2016). Available at: http://chemcollective.org/.

[17] The TriLab project. (2017). "Southborough University, UK". www.lboro.ac.uk/departments/chemical/undergraduate/teaching/trilab/

[18] Juha Sorva. (2013). "A Review of Generic Program Visualization Systems for Introductory Programming Education," ACM Transactions on Computing Education, Vol. 13, No. 4, Article 15, Publication date: November

[19] K. Aldrawiesh. (2009). "Selecting the best object-oriented programming language for developing Distributed Computing Systems," International Conference on Computer Engineering and Systems

[20] Aleksi Lukkarinen. (2021). " Event-driven Programming in Programming Education: A Mapping Review," ACM Transactions on Computing EducationVolume 21Issue Article No.: 1pp 1–31, 1March

[21] Celal Murat Kandemir. (2021). "Pedagogy of teaching introductory text-based programming in terms of computational thinking concepts and practices," Special Issue: Computational thinking: Enhancing STEAM and engineering education, from theory to practice, wiley

[22] D. Zhang, et al. (2021). "A Comparison of Native and Cross-Platform Frameworks for Mobile Applications," IEEE computer Volume: 54, Issue: 3, pp 18 – 27, March

[23] Andreas Biorn Hansen. (2020). "An empirical investigation of performance overhead in cross-platform mobile development frameworks," springer, Empirical Software Engineering 25:2997–3040

[24] Xinhui Xie. (2018). "Design and Making of Sudoku Game Based on Unity3D," springer, Advances in Graphic Communication, Printing and Packaging pp 410-415

[25] Jingming Xie. (2017). "Research on key technologies base Unity3D game engine," 7th International Conference on Computer Science & Education (ICCSE)