VRduino Learning: Introduction in Arduino with Virtual Reality

Georgiana-Alexandra Bodea, Victor Ioan Bâcu
Technical University of Cluj-Napoca, Computer Science Department

Cluj-Napoca, Romania
bodea.gh.georgiana@student.utcluj.ro, victor.bacu@cs.utcluj.ro

ABSTRACT
Traditional methods of learning Arduino often encounter challenges such as component damage, limited interactivity, and lack of real-world simulation. To address these issues, several 2D/3D applications have been developed, aiming to enhance traditional learning approaches by safeguarding components from damage, thereby reducing risks and costs associated with practical experimentation. However, despite these applications being cost-effective, they offer limited hands-on experience and do not mitigate potential disruptions.

In this paper, we introduce an application aimed at enriching Arduino learning through immersive experiences. This application allows users to actively engage in learning and provides teachers with tools to create customized lessons and projects using the available components.

Author Keywords
Virtual Reality; eLearning; Arduino

DOI: 10.37789/ icusi.2024.20

INTRODUCTION

Virtual reality has gained significant interest nowadays, particularly due to its potential for continuous growth and its ability to create an infinite number of fictional realities. Because of its immersive nature, it has been very well received by the population, which is becoming more familiar and comfortable with it with each passing day.

Another important and relevant topic for today's society is the educational domain. It represents the pillar of society, and often it is said that it refines and shapes the population. Its significance is evident throughout history and particularly now, as people seek new ways to enhance the learning experience as effectively as possible.

It is ever-changing and constantly seeking new solutions. With the growth of virtual reality technology, many people have started considering various ways to integrate this new technology into the teaching process. Initially, it may have been met with skepticism because it was new, and students were not yet accustomed to it. However, it soon became a significant potential replacement to traditional teaching methods. The pandemic period also forced students and teachers to be more open to using technology, accelerating the transition from traditional to modern teaching methods.

The goal of the application is to increase engagement, create a less distracting environment, help the students to be more independent and help teachers to make their lessons more interactive. This is achieved by making the Arduino learning process more accessible, eliminating the need for physical components, and allowing users to focus more on learning without the stress of potentially damaging components. This application was developed to help students feel more at ease when first working with components that could break if connected incorrectly, while also providing a realistic learning environment. The current implementation is straightforward, with plans for future enhancements and improvements.

So, in this implementation, teachers can create new and personalized lessons and projects. They could provide the students with a better learning experience by giving them different possible approaches for learning the material. Additionally, it aids in scaling because the same lesson or project can be used by multiple students who have access to it.

The customization feature of the application represents a strong point because it doesn't rely solely on existing content, but it also enables limitless creation of lessons and projects that can then be studied and problem-solving. Through this, the application can become better and better with each new lesson and project added.

RELATED WORK

Virtual reality or VR is a term used by the popular media to describe an imaginary world which can only exist in computer or our own mind. [1] [The Definition of Virtual Reality]. It is also described as an oxymoron because it describes something that doesn't exist but it's real, something imaginary that represents reality.

A better definition is given by the website merriam-webster.com [2] as "an artificial environment which is experienced through sensory stimuli (as sights and sounds) provided by a computer and in which one's actions partially determine what happens in the environment". This refers to the idea that VR represents a computer-generated world

which can be interacted with as if the environment is real. In [3] it is specified that VR is a useful tool when it comes to complex issues or when a deeper and closer visualization is needed for a better understanding. In another article [4] Taylor Freeman, the author of it, mentions the main powers of VR are the ability to appeal and the ability to spark curiosity and interest from everyone. It also presented some benefits like:

- the power of immersion,
- accelerated learning and increased confidence,
- reduced disruption and enhanced focus.

Some disadvantages how are presented in another paper [5] could be:

- VR needs more space,
- people can start to suffer from Virtual reality sickness.

The main difference between this application and other available solutions is the fact that it is a 3D application, as opposed to Wowki [6], and it is a VR application, as opposed to Crumb [7]. The approach from the proposed application allows the player to have more " hands-on" practice with the subject matter. Having more "hands-on" experience can provide the student with more confidence in the subject and allow him/her to handle the real-life applications of it way better.

Since this application simulates real-life Arduino projects, it's important to note that students are more likely to experiment and make mistakes. The application provides a safe, fail-proof environment where errors do not lead to real-life issues such as broken or damaged components.

CONCEPTUAL ARCHITECTURE

In Figure 1 it is presented the top-level components and how they interact between them. When a user performs commands, they are received by the input handler, which directs them to specific areas of the Unity application and generates a corresponding response. This response is then interpreted by the device API and presented to the user either audibly via the speaker or visually through various elements within scenes. Each scene within the application represents a dynamic 3D environment which has different panels, fields and objects. Additionally, each scene includes a back button to facilitate user navigation back to the previous page or interface, ensuring seamless usability within the application.

The most important functionalities for this application are:

- navigate through the scenes: this will be done through some menus (some menus can be accessed just by those who have teacher role),
- create/update lessons related to Arduino, which is allowed just by the teachers,
- create/update Arduino project which needs to be solved by the students. This operation is allowed to be done just by the teachers,
- learn a lesson and solve a project,
- load/persist projects/lessons from/to a JSON file,
- move through the scene, using the joysticks.

In Figure 2 can be seen the general scenes and how they interact between them.

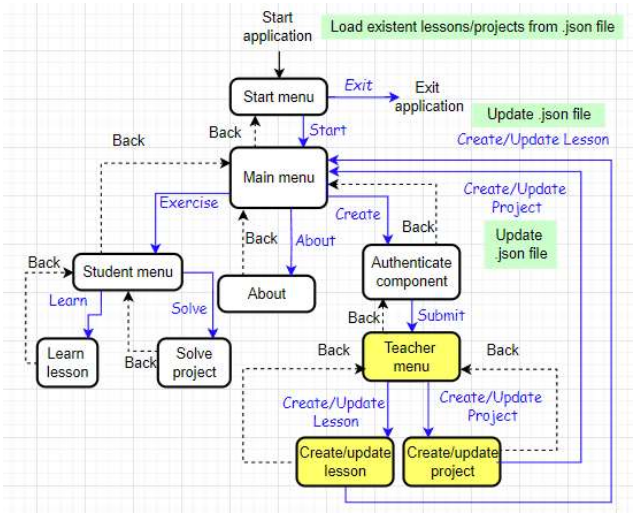

Figure 2. Components hierarchy

LESSON PRESENTATION

A lesson consists of multiple sections containing various information and photos related to Arduino. When students access a lesson, they are presented with comprehensive content designed to guide their understanding of Arduino concepts, principles, and practical applications. The sections typically cover topics such as basic theories, step- by-step instructions for projects, explanations of components and their functions.

An example is presented in Figure 5 where is a lesson called “Introduction in Arduino” where the student can read the overall description of the lesson and can see the photos with their descriptions. Students have the flexibility to scroll

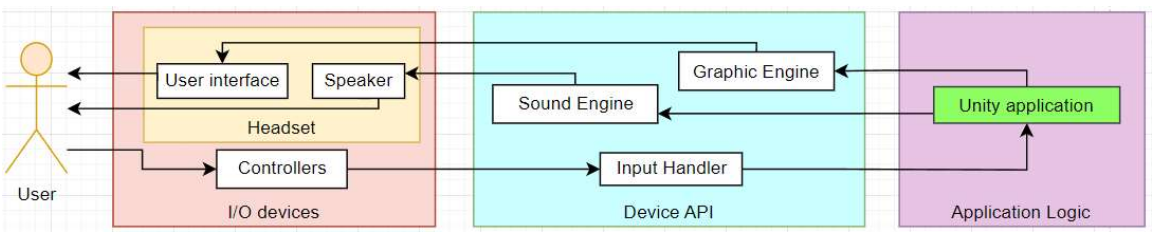

Figure 1. General application scheme

through both panels to access all the information presented in the lesson.

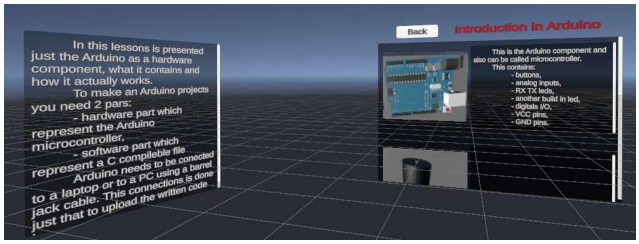

Figure 5. Example of lesson to learn

PROJECT PRESENTATION

A project consists of multiple steps which must be solved. This solving part represents the most interactive part of the application. Each step has in its description some components which must be connected between them. The connection is made automatically the moment a user clicks on a check button and pins/components are correctly selected based on the step description. If the step is solved correctly, the linking will be done, and the next step will appear (if the actual one is not the final step). If the step is solved correctly and the current step was the final one, a message with congratulations will appear which informs the user that the project was completed. If the step is not solved correctly, a panel with an error message appears, prompting the user to retry selecting the correct components and pins. In addition, when the user hovers over some pins the pin's name will appear just like a label. Pins and components involved in solving the project are highlighted in blue when hovered over, and selecting a component changes its color to yellow. Once all the selections are done, he/she can click on a button dedicated to check if the selected components are correct.

An example of a scene where a student needs to solve a project is presented in Figure 9. There it can be seen an example of a project called "Led Project". In the scene, the first step is shown as completed, with a green male-female cable linking the LED to the Arduino. The step description is displayed on the right side of the scene alongside the check step button.

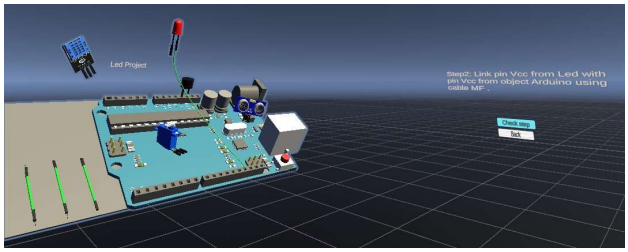

Figure 9. Example of project to be solved

CREATE/UPDATE LESSON

A lesson is used by the teacher to present to the students the basic concepts about Arduino functionalities and components in a more interactive way. This lesson can be created and updated solely by teachers. Each lesson is structured into two essential sections to facilitate effective learning. The first part is the overall description section which provides comprehensive textual information about Arduino. It is designed to convey foundational knowledge and explanations that do not necessarily rely on visual aids. The second part consist of the images section with descriptions which allows teachers to enhance the lesson with visual aids. This part improves the learning experience by offering students visual references that complement the textual information.

Figure 3 and Figure 4 illustrate the process of creating a lesson scene. In Figure 3 are the above explained parts: the left input field where the lesson description is written and the right panel which has the images along with their descriptions. There is also a drop down from where the user can choose what image to add to the lesson. The images from the drop down appear in Figure 4. By clicking the add selected image button the selected image in the drop down is added to its above white area.

In this application an image represents a texture, and a texture needs to be associated with an object and because of this an image cannot be imported like in a word file. This leads to the creation of those white zones which are called raw images, where the image will be saved, but unfortunately those zones are limited. If a teacher wishes to add more photos to a lesson, they must contact the developer to potentially increase the number of available sections. The input field adjacent to each image allows the teacher to add text relevant to the selected image, and the "X" button is used to delete the adjacent selected image.

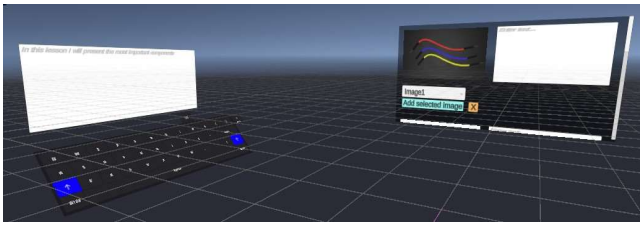

Figure 3. Some of the components from create lesson scene

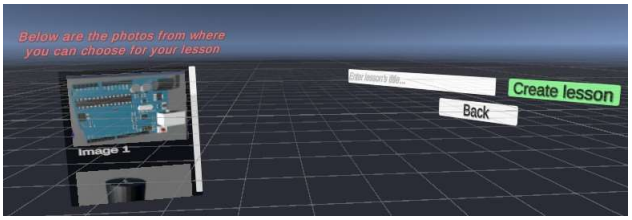

Figure 4. The other components from create lesson scene

In Figure 4, the left panel displays the images available for adding to the lesson. This approach was chosen because we couldn't find a free standalone plugin that allows users to upload photos directly from their machine. This panel is only visible during the lesson creation or update process and does not appear in the lesson itself. On the right side is the input field

where the teacher must enter the lesson's title. The lesson title is mandatory otherwise a panel with an error will appear. Clicking the "Create Lesson" button initiates the lesson creation process, assuming the title is valid. If a lesson with the same title already exists, a message panel informs the teacher of the duplication and prompts them to try a different title. Once a valid title is entered and confirmed, the lesson is successfully created, and the teacher is redirected to the main menu.

When a teacher updates a lesson, the scene will appear like the creation process. The main difference is that existing data will be displayed, including the lesson title shown in Figure 3. In Figure 4, the input field for the lesson title and the "Create Lesson" button will no longer be present; instead, an "Update Lesson" button will appear to finalize the modifications.

CREATE/UPDATE PROJECT

A project is a practical part designed by teachers to allow students to apply their Arduino knowledge practically. These projects can be created and updated just by the teachers. A project is composed of more steps, where each step has a general format like "Link pin --- from object --- with pin – from object --- using cable --- ", "---" being replace with a desired pin or Arduino component. This structure was created as a general one to simplify the project solving process, but also for creating/updating project process.

- Arduino [8]
- Led [9]
- Ultrasonic sensor [10]
- Servomotor [11]
- Buzzer, humidity sensor [12]
- Male-female, male-male, female-female cables

Some of the components were used directly how they were downloaded, but some of them had to be a little modified using Blender [13].

Figure 8 Panel for finish project creation

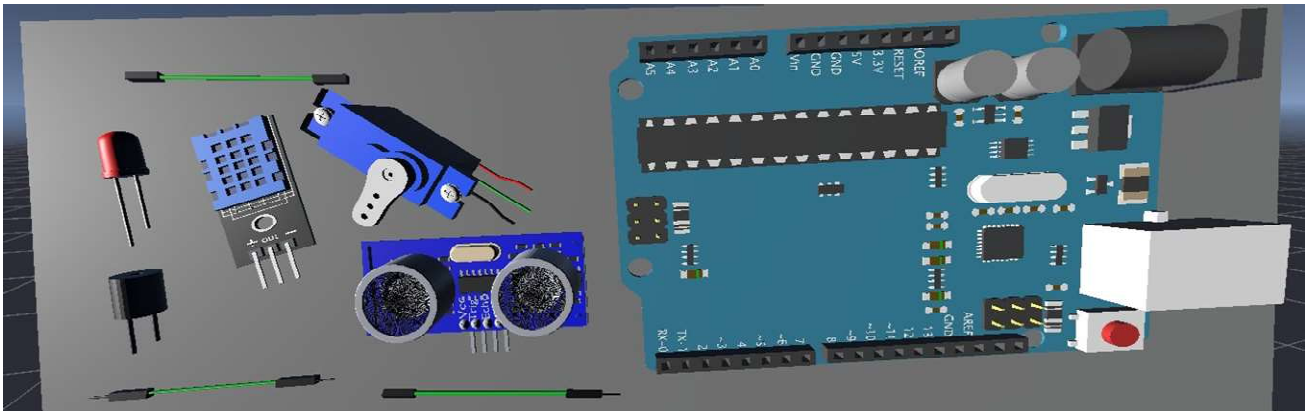

Figure 6. The components from create project

An example of how the scene which shows how a project is created can be seen in Figure 6, Figure 7 and Figure 8.

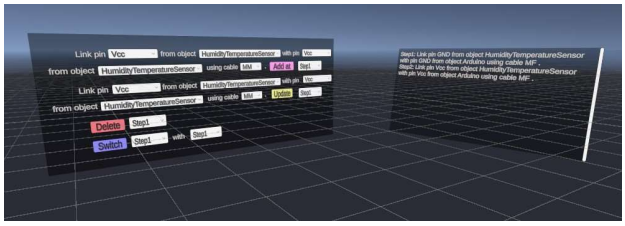

Figure 7. Steps for project creation

In Figure 6 are presented the existent components from where the teacher can choose to create the project. The components presented here are the only posibilities. These components are:

Figure 7 displays a panel where the user selects the order of steps and the desired components for each step. The "Add" button inserts the selected step into the project at the chosen position. The "Update" button modifies the currently selected step with new values, which is disabled if no steps are present. The "Delete" button removes the selected step, also disabled when no steps exist. Additionally, the "Switch" button exchanges the positions of two steps, disabled when no steps are present. Any performed operations will be seen in the panel on the right. For instance, in the image provided, two steps have already been added to the lesson.

Figure 8 presents the last panel which helps the user to finalise the project. The teacher needs to enter the project title in the input field and then to click on “Create Project” button. The title is mandatory and if a project with that title already exists a panel with an error message will appear.The user just needs to click on “Try again” button and to enter a new tititle. If title is correct the project will be created and the user will be redirected to the main menu.

When updating a project, the process is like creation, with the distinction that in Figure 7, the project name will also be displayed alongside its existing steps. In Figure 8, instead of the input field for the project title and the "Create Project" button, an "Update Project" button will be available to finalize the modifications.

OTHER USER INTERFACE DETAILS

Figure 9 displays the start menu which allows the user to go to the main menu by pressing the “Start” button. By clicking on the “Exit” button, the user will be sent out of the application.

Figure 9. Start Menu

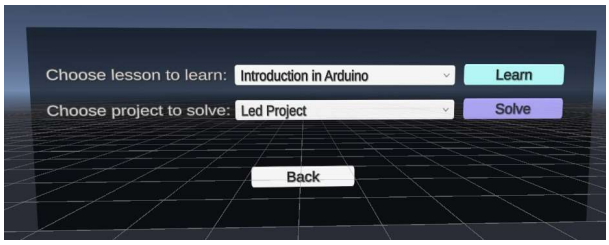

Figure 10. Main Menu

Figure 10 displays the main menu, providing users with several options. Clicking the "About" button directs users to a page where they can briefly learn about how the application operates. The "Exercise" button allows users to access the student menu for ongoing activities, while the "Create" button redirects them to the authentication scene. This authentication is necessary as only teachers are permitted to create or update projects and lessons.

In Figure 11 is the menu from where a student can choose to learn a lesson or to solve a project.

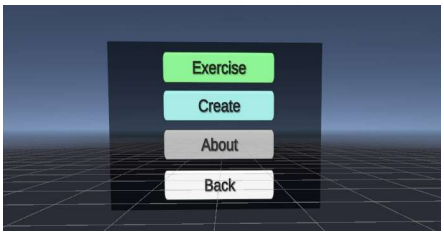

Figure 11. Student menu

In Figure 12 can be seen the scene where the application verifies if a user is a teacher or not. This check ensures that only teachers can access the teacher menu. If the user enters the correct password, it will be redirected to the teacher menu, otherwise a panel with an error message will appear. The user can close it and try again to enter the password.

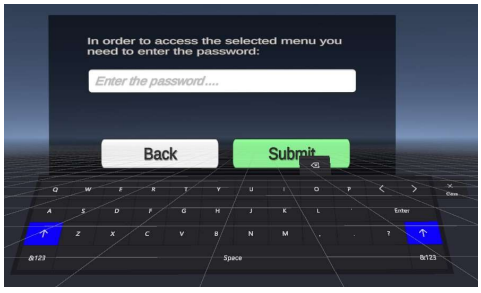

Figure 12. Validate user scene

Figure 13 illustrates the teacher menu, accessible exclusively by teachers. Clicking the "Create Lesson" button directs the user to the create lesson scene. Clicking the "Update Lesson" button navigates the user to update the selected lesson from the dropdown menu. If no lesson exists, this button will be disabled. Similarly, clicking the "Create Project" button sends the user to the create project scene. Clicking the "Update Project" button directs the user to update the selected project from the dropdown menu. If no project exists, this button will also be disabled.

Figure 13. Teacher Menu

Every panel with an error message will look like the panel from Figure 14, the only thing which differs from is the error message.

Figure 14. Error message panel

SCRIPTS

When a lesson/project is updated or created, those changes are written in a JSON file. We chose this option to save the data because it was the easiest and most convenient method. Using a database would have required hosting it on a server and connecting our local application to that server. In the chosen method, we simply needed to declare the fields we wanted to save as SerializeField and ensure the classes were Serializable. Additionally, a small amount of code needed to be added to a script. The data is mandatory to be saved somehow such that to not lose the information when the user exits the application. Based on this JSON file each time the application is started for the first time, all saved data will be restored to the application.

Furthermore, several scripts written in C# were added to this application. The most important functionalities of the scrips are to:

- have the structure of a lesson/project such that to be able to save/update them in the JSON file,
- validate if a new lesson/project title doesn't already exist,
- check if the entered password to access the teacher menu is correct,
- create new lessons/projects and to add them to the final project,
- to convert dtos to entities and vice versa,
- to change color of some objects when hover/click over/on them,
- to populate a dropdown,
- to save data to JSON and to load data from JSON,
- handle the lesson/project updates.

EVALUATION AND TESTING

Different type of testing was done and is ongoing for this application:

- testing done by us using try and error,
- testing done through other people without guidance.

In the second method, we engaged 5 students to create lessons and projects, and 5 teenagers and young adults to test these materials. Following the feedback obtained, we observed a significant positive impact due to this new immersive approach. All participants acknowledged that this application represents a substantial improvement for both teachers and students. However, those involved in creating lessons and projects noted that the process is somewhat challenging because typing with rays can be difficult, and the limited components and photos are not ideal. Nevertheless, they agreed that the application is highly beneficial for introducing Arduino concepts.

In the future our objective is to continually gather feedback to understand user reactions when using this application. We aim to encourage users to speak out loud their thoughts during the testing process. We try to choose people who have interacted with VR and people who haven't, male, female, within different age ranges. We will measure usability using the SUS questionnaire [14] to identify any challenges users may encounter with the application.

CONCLUSION

The implementation part for this application was a challenging one, but we successfully managed to find a solution for each proposed thing. The most problematic aspect was enabling teachers to select any photo they wanted to add to a lesson. Due to the unavailability of a free plugin for this functionality, we customized this feature by incorporating a list of default images from which users can choose. This customization enhances the user experience by providing a user-friendly interface that is easy to navigate.

REFERENCES

1. J. Jerald, The VR Book: Human-Centered Design for Virtual Reality. Association for Computing Machinery and Morgan & Claypool, 2015
2. "Definition of VIRTUAL REALITY — merriam- webster.com," https://www.merriam-webster.com/dictionary/virtual%20reality
3. D. Hamilton, J. McKechnie, E. Edgerton, and C. Wilson,"Immersive virtual reality as a pedagogical tool in education: a systematic literature review of quantitative learning outcomes and experimental design," Journal of Computers in Education, vol. 8, 07 2020
4. Taylor Freema, "How can vr help with education?" How Can VR Help With Education? - Axon Park, 2023
5. Z. Xu, "Analysis of the advantages and disadvantages of virtual reality technology in gaming," Lecture Notes in Education Psychology and Public Media, vol. 14, pp. 157–161, 10 2023
6. "Wokwi - Online ESP32, STM32, Arduino Simulator — wokwi.com," https://wokwi.com/
7. "CRUMB | Circuit Simulator — crumbsim.com," https://www.crumbsim.com/
8. 3DExport, "Free 3d model: Arduino uno," https://3dexport.com/free-3dmodel-arduino-uno-353631.htm
9. Sketchfab, "Led (light emitting diode)," https://sketchfab.com/3d-models/led-light-emitting-diode-0c5855559ee5465aa8044a0fdbac39fb
10. GrabCAD, "Hc-sr04 ultrasonic sensor," https://grabcad.com/library/hc-sr04-ultrasonic-sensor-10/details?folder_id=12996195
11. mik.a. (@mik1190), "Servomotor (explode animation), https://sketchfab.com/3d-models/servomotor-explode- animation-0fad21024bbc461db314a21db96b1395

12. Sketchfab, “Komponen elektronika microcontrollers,” https://sketchfab.com/3d-models/komponen-elektronika-microcontrollers-bb4c7f974cef44f8b1db21afb0cce78d#download
13. Blender Foundation, “Blender,” https://www.blender.org/
14. QuestionPro, “System usability scale (sus),” https://www.questionpro.com/blog/system-usability-scale/