NORTH SOUTH UNIVERSITY

Body Parts Demonstration in Virtual Reality for Doctors and Medical Students

A DISSERTATION
SUBMITTED TO THE DEPARTMENT OF
ELECTRICAL AND COMPUTER ENGINEERING
OF NORTH SOUTH UNIVERSITY
IN THE PARTIAL FULFILMENT OT THE REQUIREMENTS
FOR THE DEGREE OF
BACHELOR OF SCIENCE IN
COMPUTER SCIENCE AND ENGINEERING

CSE 498R, SPRING 2022
DIRECTED RESEARCH PROJECT

Declaration

It is hereby acknowledged that:

- No illegitimate procedure has been practiced during the preparation of this document.
- This document does not contain any previously published material without proper citation.
- This document represents our own accomplishment while being Undergraduate Students in the North South University.

Sincerely,

Mo'in Uddin Huq

Moin Uddin Huq
1620005042

Sidratul Muntaha Taj
1731905042

Md. Rifat

Md. Rifat
17310275042

Approval

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

DR. MOHAMMAD MONIRUJJAMAN KHAN
Assistant Professor
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

I certify that I have read this dissertation and that, in my opinion, it is fully adequate in scope and quality as a dissertation.

DR. MOHAMMAD REZAUL BARI
Associate Professor & Chair
Department of Electrical and Computer Engineering
North South University
Dhaka, Bangladesh

Abstract

We are making a system in Unreal Engine 4.26.2 that will display the human body, brain, eye, lungs, and digestive tract system in VR using OpenXR, an open-source add-on supported by Unreal Engine. Learning and teaching with VR is one of the technologies at the forefront of having the potential to improve learning and teaching. In recent years, virtual reality technologies have advanced significantly, opening up many new learning opportunities. Outside of the Classroom, virtual reality enables a more tailor-made learning experience. This paper describes the development and discussion of a virtual reality application for teaching medical students about human body parts in virtual reality. This application will allow medical students to closely observe and understand different human organs in a 3D view. We are doing this for the benefit of medical students and doctors. And to prove that VR technology can be used for advanced and low-resource practical learning.

Contents

List of Tables

List of Illustrations

Acknowledgements

The authors of this study would like to express their gratitude to the Electrical and Computer Engineering Department of North South University in Dhaka, Bangladesh for allowing us to conduct this cutting- edge research.

List of Acronyms

SDK: Software Development Kit

JDK: Java Development Kit

JRE: Java Runtime Environment

VR: Virtual Reality

NDK: Native Development Kit

VS: Visual Studio

HMD: Head-mounted display

Glossary

There were none.

1 Introduction

In recent years, VR technology has advanced hugely, as now very low-powered devices can run VR applications smoothly. As the era changes, the teaching method is moving further towards digital methods. Using a 3D environment instead of a 2D paper image is far more helpful towards the understanding of a medical student. We can also see the 3D object in greater detail and in true 3D with VR technology. Also, the head movement of the platform further helps with the viewing angles of the subject. As we all know, the most effective way to learn human anatomy is through visualization. Real body parts are the best way to learn, but it's not always possible. However, 3D representations can be close to real-life objects. Also, with the increase in screen resolutions, GPU power in small handheld devices is growing rapidly and now at a point where it can be used to comfortably push two 540*480 3D video feeds simultaneously. Thus, now students can watch very high-definition anatomical structures in VR on low-end Android devices. Due to constraints in resources [1], it has recently been very difficult to educate medical students in human anatomy. Recent studies indicate that game-based VR environments can be a very effective teaching medium [2], [3]. The main goal of this project is to make an app for medical students and doctors to reference while in their studies and/or practice sessions. This app will show detailed objects of the human body, such as the brain, eye, lungs, and digestive tract system. All these objects will be shown in VR on various platforms. Users can freely move around the anatomical body parts and see the enlarged but accurate-to-scale versions. The resolution and fidelity of the project are high enough to be used as a simulation.

The way people connect with and interact with educational materials is changing due to technological advancements. In recent years, virtual reality technologies have advanced significantly, opening up many new learning opportunities. Because of its potential to create a virtual environment [3], VR technology in education and training has gotten much attention. It adds value to the process and immerses users in a pleasant and productive learning environment. The 3D model is usually static, but there is a mode where users can view animations of joints and bones in motion [1]. Most notably, as with real-life physical models of the human body, VR anatomy focuses on the visual appearances, spatial relationships [2], and dynamic movements of individual body parts and the systems they comprise. The goal for users is to identify and understand the location, function, and terminology of anatomical structures in the human body. The most effective way to understand human anatomy is through visualization. While real organs are the best source to learn from, it is impossible to observe one whenever an individual student needs to. Medical students learn human anatomy during their first year of study, but when they reach the preclinical and clinical phases, their anatomy knowledge is used. As a result, gross anatomy is more crucial for clinical students to master during their first year. The students experienced difficulties with 3D conceptualization and visualization structures [1]. Teaching human anatomy to medical students has become a complex problem due to rising student numbers and constrained resources [2]. According to recent studies, game-based VR environments have a more significant learning impact than virtual worlds or VR simulations alone [3]. They are using virtual reality applications that will allow individual users of different platforms to closely observe and understand 3D models of different anatomical systems and organs within virtual reality.

Moreover, they discuss a virtual reality smartphone app. The VR mobile application is for the teaching purpose of human body parts to medical students. Using VR applications, students will better understand human anatomy when learning from a 3D view rather than a 2D image [3].

When wearing even a low-end VR product, the fidelity is high enough to be a convincing, realistic simulation of a medical learning environment. It attempts to replicate a medical school classroom that would contain a highly detailed physical model of the human body. Learners can freely remove each layer to see the various relationships between muscles, nerves, and organs, zooming in to the microscopic level if needed. Students can understand the whole anatomy much better before practicing it in the real world [4]. In [5–12], authors have presented their work in the open literature on education. Some of them have worked on the application of virtual reality in different fields. This paper presents the application of virtual reality in teaching medical students.

2 METHOD AND MATERIALS

This virtual reality app is multi-platform. It can be used with Windows, Linux, and Android. However, the key focus is on Android tablets and smartphones. So, henceforth, this report will focus on android-based methods. An android-based device is used for the purposes of displaying the virtual 3D environment. The device generates the 3D world in two views, and we use VR goggles to watch and experience the 3D VR environment in a stereoscopic view. This allows the user to move around the environment and get close or get far and get a good view of the part they are looking for. Our system creates a realistic and believable environment. Also, we have implemented head tracking features so the user can move their head freely and the in-game world will move with the motion of their head. First, we make all the levels that are necessary for the game. As we are going to have 4 items, there must be 5 levels in total: four levels for the four items, and one main menu level. Then we make an enclosed level with zero outside light. As no outside light is coming in, it helps make the VR experience less nauseating [4]. The next step is to populate the levels with all the 3D VR-ready assets. As the Unreal Engine is very popular, works with the fbx file format, and is also supported by Blender, we have chosen to use fbx files for this project. We have obtained usable fbx files from various sources. Some of the files had missing textures due to how we used blender to fix them. Also, the brain model was in.obj format, for which we again used Blender to re-texture and export to.fbx. After that, we implemented the player character as we began our project with blank settings and no starter content, in which no player character asset was given to use. We have made another project with VR support (which does not support mobile devices) and imported the movement code and virtual joysticks from that project to our main project. After that, we have implemented various widgets and blueprints, which are discussed below in further detail. And lastly, we have created the VR support needed to be implemented. We have discussed the process and difficulties we faced regarding this step at the end of this section.

This Table shows the hardware requirements.

Table 1 List of hardware requirements

No.	Name
1	Android Mobile Phone
2	VR Headset

This table shows the software requirements [13-23]

Table 2. List of software requirements

No.	Name
1	Android SDK 31
2	Android NDK 21.0003
3	Android JRE 18
4	Android JDK 8
5	Unreal Engine 4.26.2
6	VS Code
7	Blender
8	Quixel Bridge
9	OpenXR
10	OpenXR Headtracking plugin
11	OpenHMD
12	FBX Files
13	Android Studio 4.0
14	Netcore 3.1
15	Netcore 5.0

The details of the app including the codes, HUD, items, and so on are depicted through the figures as follows.

2.1 Game Menu

This Figure shows game menu code.

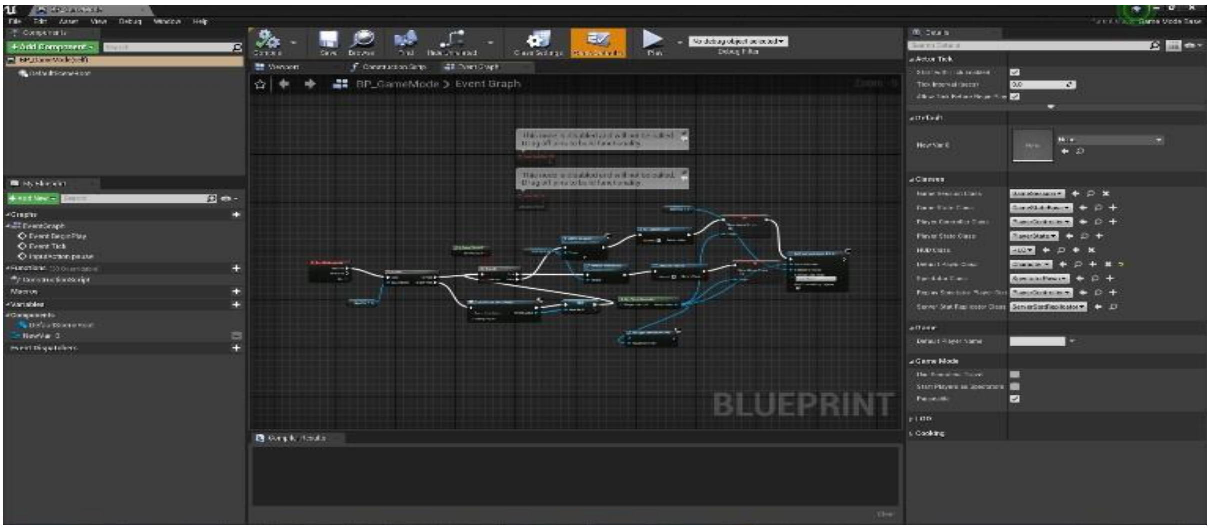

Figure 1. Game Mode Code

Above are the game mode codes. In this blueprint we have all the level logic and parameters the game runs on. We reference this blueprint to use the function cast to game mode. Many variables are stored in this blueprint. Also, here we set what character blueprint we use, which physics model collision settings, lighting settings, etc.

2.2 3D assets

This Figure shows list of 3d models /fbx Files.

Figure 2. fbx files list

This figure shows the list of all the FBX files we have used. As discussed, previously we mainly sourced our fbx files from free licensing and open-sourced platforms. Here is the full list of them. We further used fbx files so that we could import all the models, sometimes in pieces and also without skeletal mesh. which helped reduce our game's overall size.

2.3 HUD

This Figure shows HUD widget.

Figure 3. HUD Wiget

Above is the HUD the player sees. There are also the two virtual joysticks implemented in the player character object. Other than that, there is a pause button. We use a transparent background as this widget is directly rendered above the player character viewport. We accomplish this by setting the parameter of the function z-order to 0. So, it always renders on top.

2.4 Items Widget

This Figure shows Body Part Selector Widget

Figure 4. Body part / Item selector Widget

Widget design for selecting which level/body part/item to go into. This menu can be accessed from the main menu or the pause menu. Also, there is a button to go directly back to the main menu. Here we use the on-click functionsThen we make it a variable and use the function. open a level and use the name of the level as a parameter to open selected levels.

2.5 Pause Button

This Figure shows Pause Button code

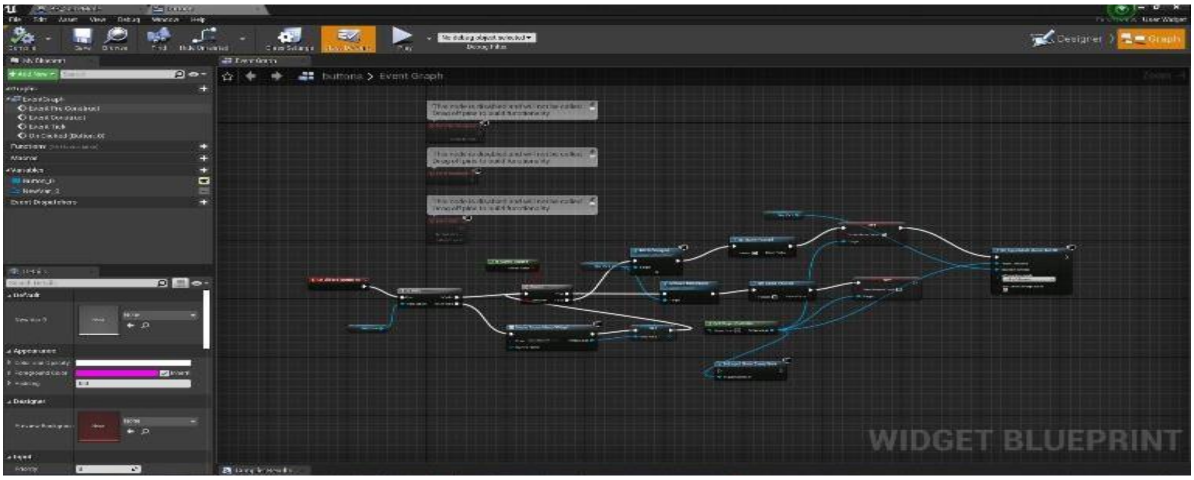

Figure 5. Pause Button Code

This is the code for the pause button. On button click, it checks if the game is in a paused state or not. If it's running, it pauses the game. Otherwise, if paused, it un-pauses the game. We use the isgamepuased function to get the output, then promote the output to a variable, then we use the variable in conjunction with a flop flop, then use the pause game function if not paused, or if paused, simply ignore the game.

2.6 Body parts selector menu

This Figure shows Body Parts Selector Menu

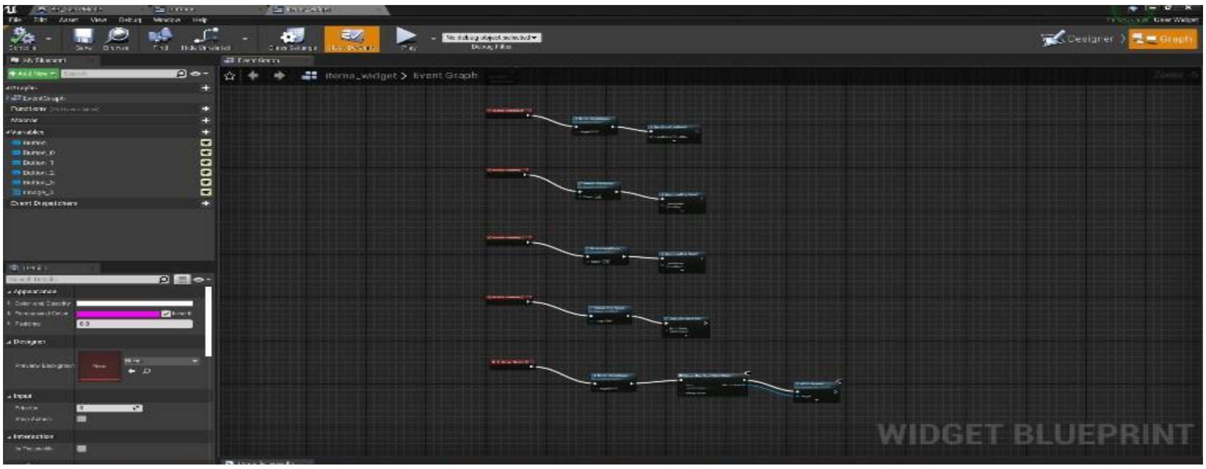

Figure 6. Body part selector menu

Code for all the buttons we have seen in figure 5. Each button contains an open level function with the level name as its parameter. We again use the open level function with the on-click function and then use the level name as a parameter. And finally, use the remove from viewport function to remove the widget from display.

2.7 Main Menu Widget

This Figure shows Main Menu Widget.

Figure 7. Main Menu Widget

Widget and background design for the main menu This is the first thing a player sees after launching the application. Also, we made three buttons with transparent backgrounds, then used anchors to set the dpi scaling. Figure For Main Menu Code.

2.8 MAIN MENU CODE

This Figure shows main menu code.

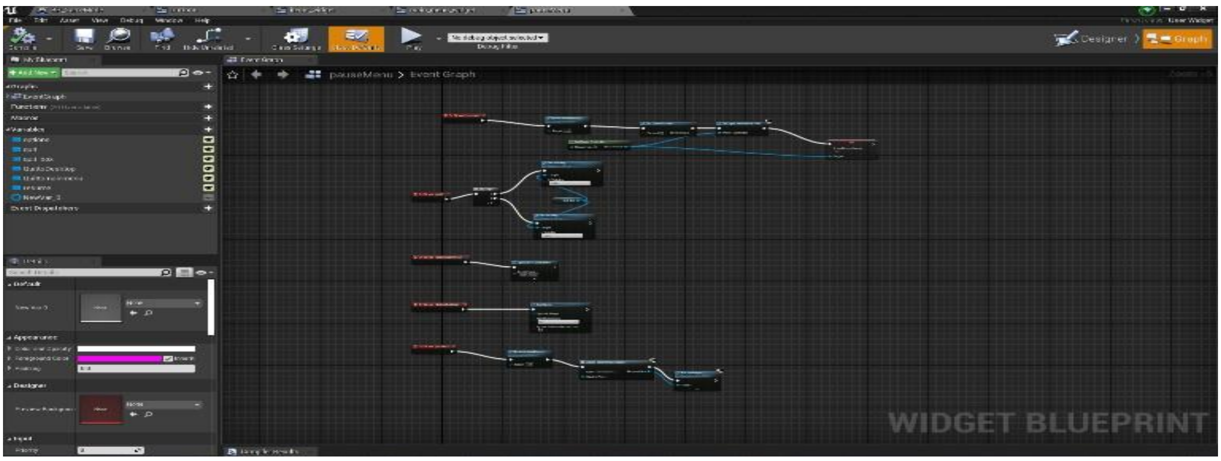

Figure 8. Main menu code

Code for all the buttons in the main menu. This works similarly to Figure 6. Figure For Pause Menu Code

2.9 Pause menu widget

This Figure shows Pause Menu Widget

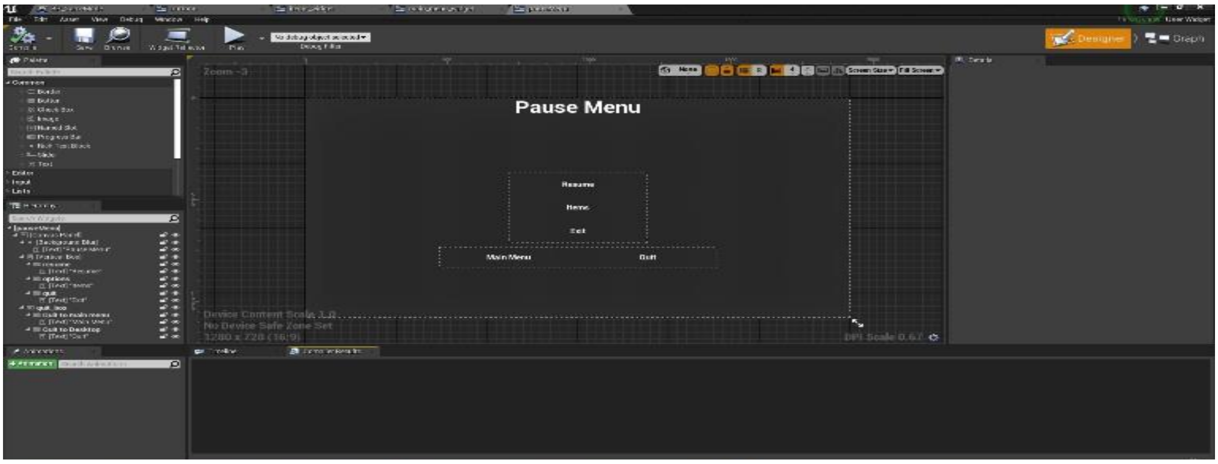

Figure 9. Pause menu Widget

This is the design for the pause menu in the game. This comes up when the user presses P on Windows or Linux or presses the button present in the HUD. We use a vertical list here. Then add the button to the list, and then use another horizontal list to add the quit buttons inside the quit button. Then, using a flip flop, determine whether it clicked and expand the buttons. We also use a translucent background so that the player can see what is going on in the background and to indicate which level is currently on.

2.10 Pause menu code

This Figure shows Pause Menu Code.

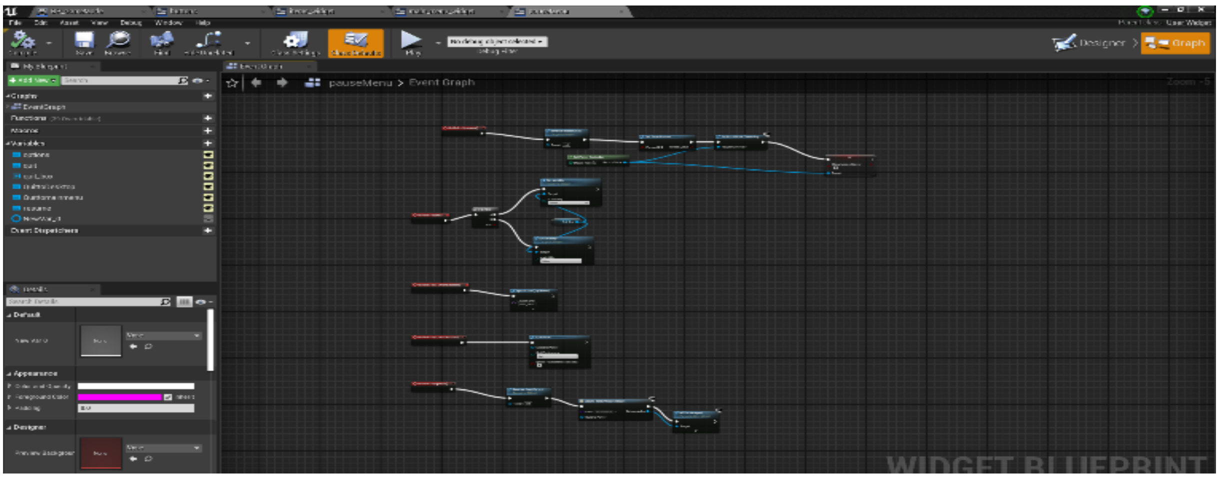

Figure 10. Pause menu Code

Code for all the buttons present in figure 9. Here, as we only go to widgets, we don’t use open level. We first create a widget on click and then use the widget name as a parameter. Before that, we need to remove the current widget from the view, so we use the remove from parent function. Then we call those two functions and finally add them to the viewport to view the function. Figure For Main Menu Level Code.

2.11 Main menu level code

This Figure shows Main Menu Level Code

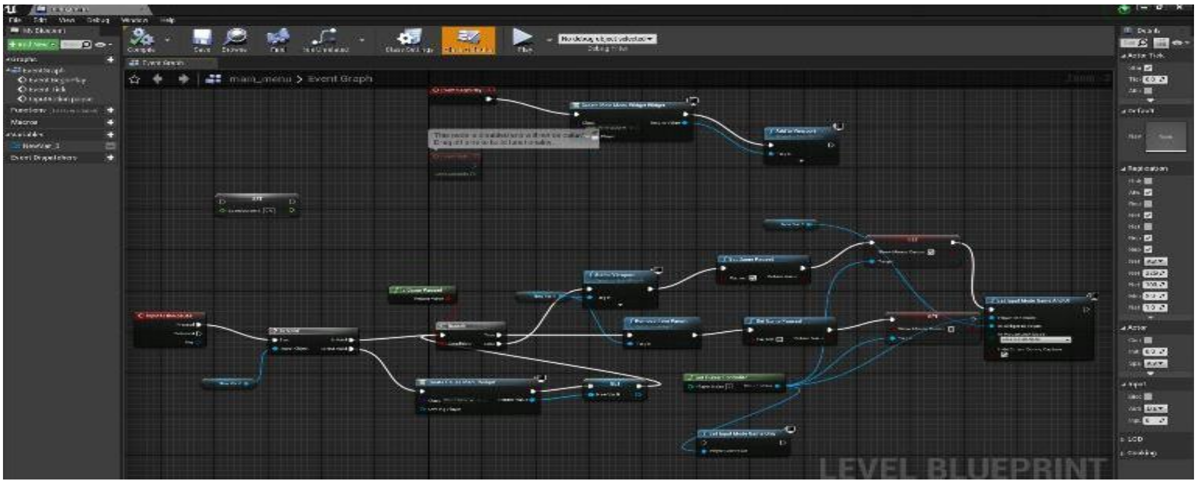

Figure 11. Main Menu Level Code

This is the main level code. From this level, the main menu widget is launched. Also, here we lock the cursor to the widget whenever a widget is called on screen. Also, the same pause menu code is also present here. Here we also reference the player character for movement. And also, here we call our HUD and add it to the viewport.

2.12 Character and movement code

This Figure shows Character Code and Movement.

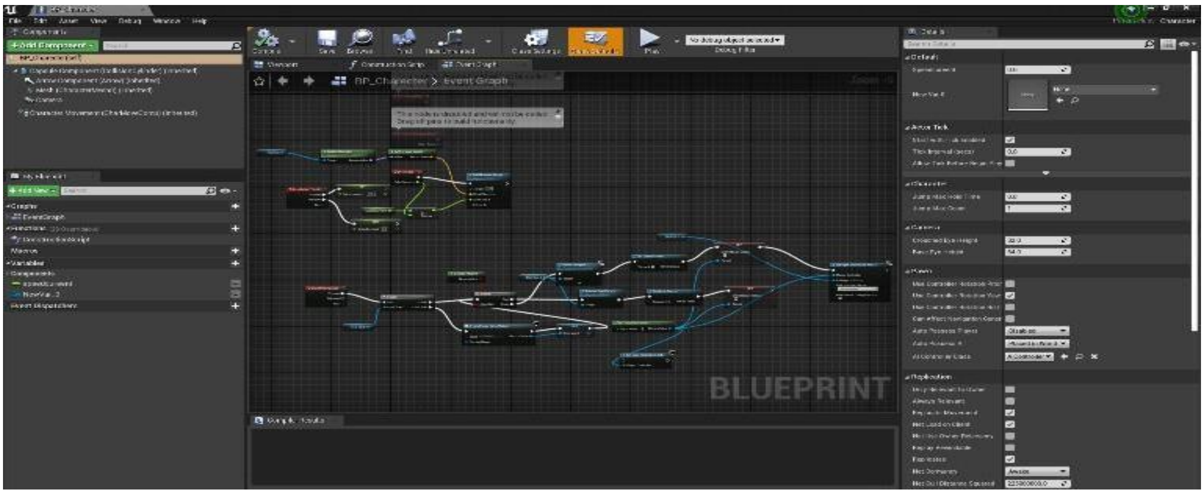

Figure 12. Character code and movement

These contain all the key bindings, camera movement, virtual joystick, and keyboard movement code.

2.13 OpenXR implementation

This Figure shows OpenXR Implementation.

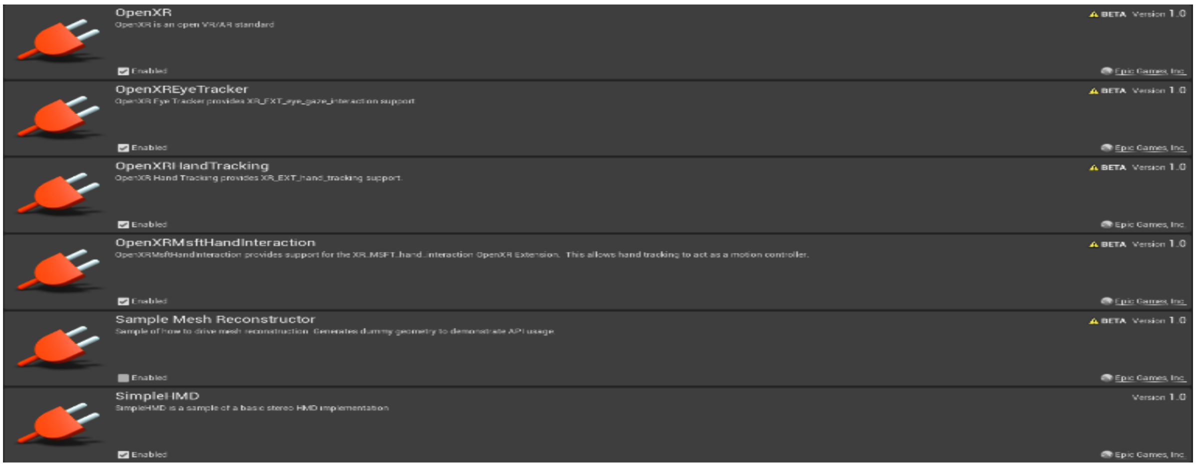

Figure 13. OpenXR Plugins

A list of plugins imported to the Unreal Engine for VR support. We have used OpenXR, OpenHMD, OpexXR headtracking and other pre-requisite plugins.

We have selected all the settings for the packaging. We have only cooked the contents of the game. We have excluded all the necessary bits by selecting "cook only selected items" and then manually selecting them. Here we choose which android versions we target. We targeted a minimum of Android 8 and a maximum of Android 12. Also, the maximum aspect ratio is selected at 2:1 and orientation are set at landscape by default. We have mostly tuned other parameters, such as the name for the OBB, package names, and last but not least, which VR output we would like. For most compatibility, we have chosen the Cardboard VR template here. We have selected android sdk v. 31, jdk v18, jre v. 8, and buildtools v21.

There is also background work that has been done, such as lighting implementation, lightscape building, getting the Unreal Engine ready for VR and miscellaneous tasks. The Unreal Engine and Google dropped support for Android VR in 2019, so the latest versions don’t support the Android VR SDK. At first, we looked into downgrading to a previous version of the Unreal Engine, specifically to the last supported version with Android VR SDK support, but that version is dependent on NVidia code for its android sdk, jdk, and ndk. We still tried to implement it by downloading specific versions of SDK, JDK, and NDK from Google archives, but without epics proprietary software and code, it’s not possible to build an apk and it's resulting in only supporting Google pixel devices, i.e., Google Daydream supported android VR devices. Currently, Epic and Google officially support the Oculus Quest 1 and Quest 2 Meta. Since NVidia code work has been discontinued, their servers are not running anymore, so we can't get our sdks from that source. For our solution, we have downloaded the Unreal Engine 4.26.2 source code from GitHub, provided to us by Epic, and then we have implemented an open-source alternative, Open XR, for our VR support. We are using Vulcan instead of OpenGL 3.1 as our graphics API. which is more optimized for arm-based chips anyway, resulting in a slight but noticeable performance boost. After downloading openXR sdk from their official repository, we added the necessary files and code given to us by open XR and recompiled using visual studio (vs code). We have also replaced epic proprietary pipelines with openXRs. After that, we used the official android sdk v. 31, jdk v18, jre v. 8 and buildtools v.21 to compile an apk for android. As open XR is an open-source platform, it supports Windows, Linux, oculus, Windows HoloLens, and Android VR. It can be both played with or without VR (only supported on PC).

3 Results

The outcomes of the suggested system are in this section. A VR Android application has been developed. Human body parts are with the mobile application. Some blood cells have been added to unity 3 D, as shown in Figure 4. The figures below depict human lungs in a 3D VR view of a mobile application. Students can see the brain, lungs, intestinal tract, eye, and whole human body. VR technologies contain distinct elements such as display, optics, eye tracking, motion tracking, and head tracking. This developed system has already been tested, and it has been found excellent in terms of performance. We do not want to limit the use of this VR system only for educational purposes. We plan to add more features for other medical purposes, like surgery and others. *Virtual reality* is a new technology that can be utilized in hospitals and clinics for rehabilitation and training. Virtual guidance and other virtual goals in the medical area are some of the applications of this technology. This product will be sustainable for many medical hospitals in Bangladesh and medical students. However, we plan to upload the developed finished product of our VR mobile application to the Google Play store so that users in other countries can benefit from this system. Medical hospitals here in Bangladesh will directly benefit from this. A considerable number of students will use the system. This idea is innovative here in our country, and we have a few researchers who have started work in this field. These are the results after all the procedures and implementation shown in the previous section. All the screenshots are from the game running on a Redmi Note 9s. The current operating system on the device is Android 12. The system gives up a constant 60fps, and it's locked at that framerate to avoid jittering or sudden frame drops. Below is the screenshot of the in-game results.

This Figure shows VR Device Used for devolopment of our system.

Figure 14. VR Box 2

We used this Google Cardboard like device known as VR BOX 2. We open the back and we find a slider to pull so we then we put our android device inside via pulling the clip. Then we slide the device back in. After that we can adjust two lenses inside by the controls located in the upper side of the device.

This Figure shows brain in VR.

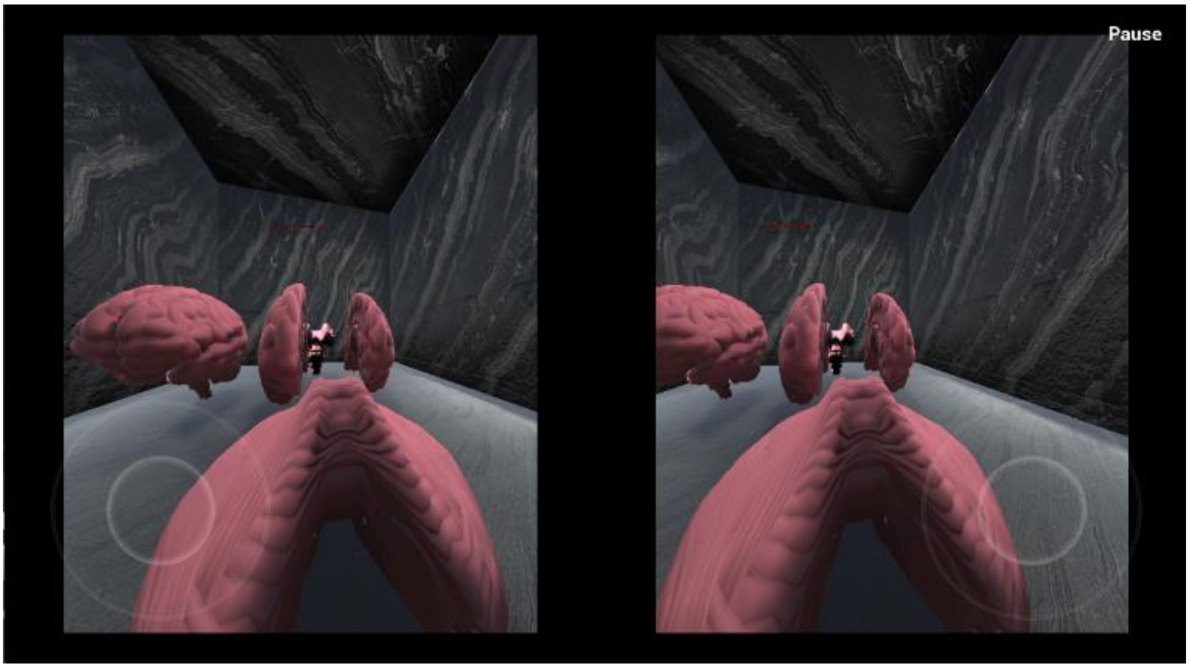

Figure 15. View of Brain

This is the in-game view of the brain in VR. The whole brain and also brain divided into three parts in present in this section.

This Figure shows lungs and intestinal tract in VR.

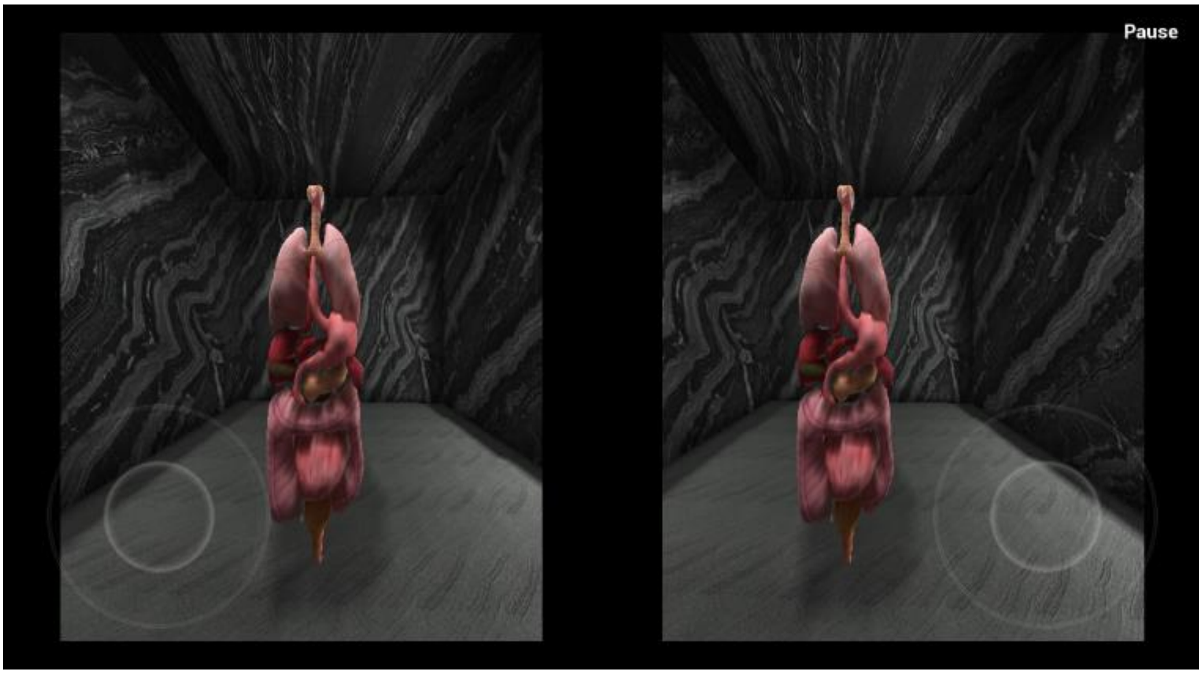

Figure 16. View of lungs and intestinal tract

This is the in-game view of the lungs and intestinal tract of the human body. We can see this section contains all the anatomy from lungs to digestive system.

This Figure shows human body in VR.

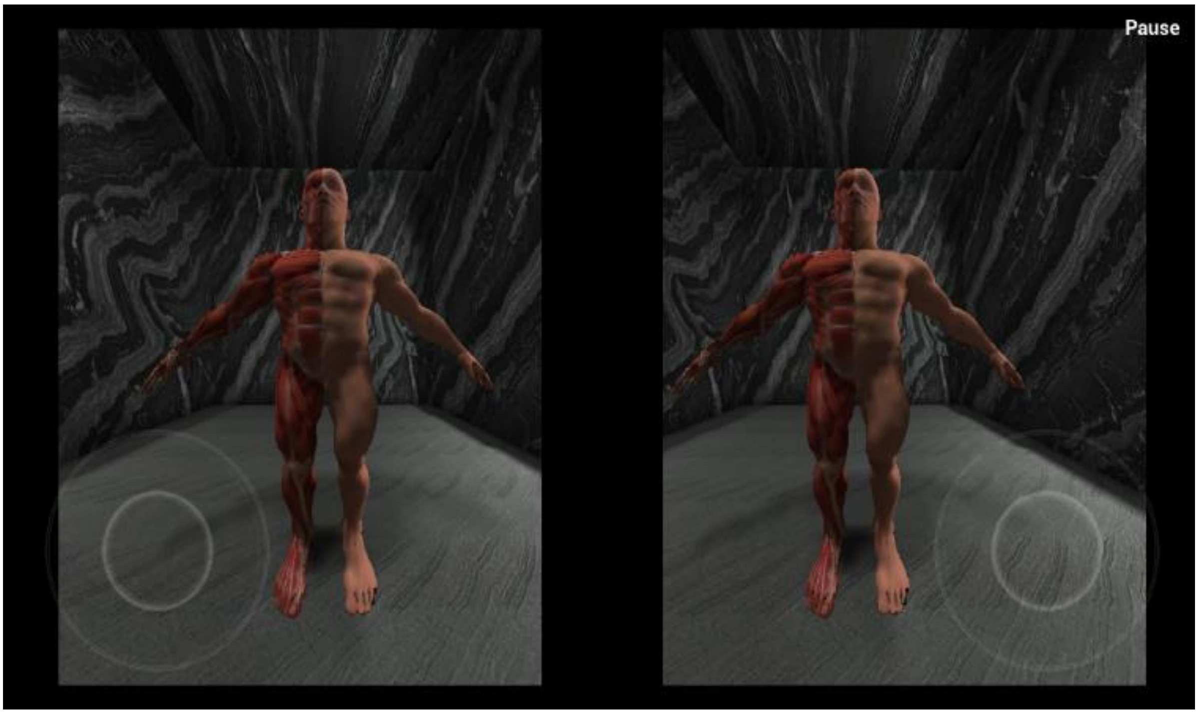

Figure 17. View of Body

This is the in-game view of the whole body. It is split in to two sections. Main focus of this section is to show the muscles of the human body.

This Figure shows eyes in VR.

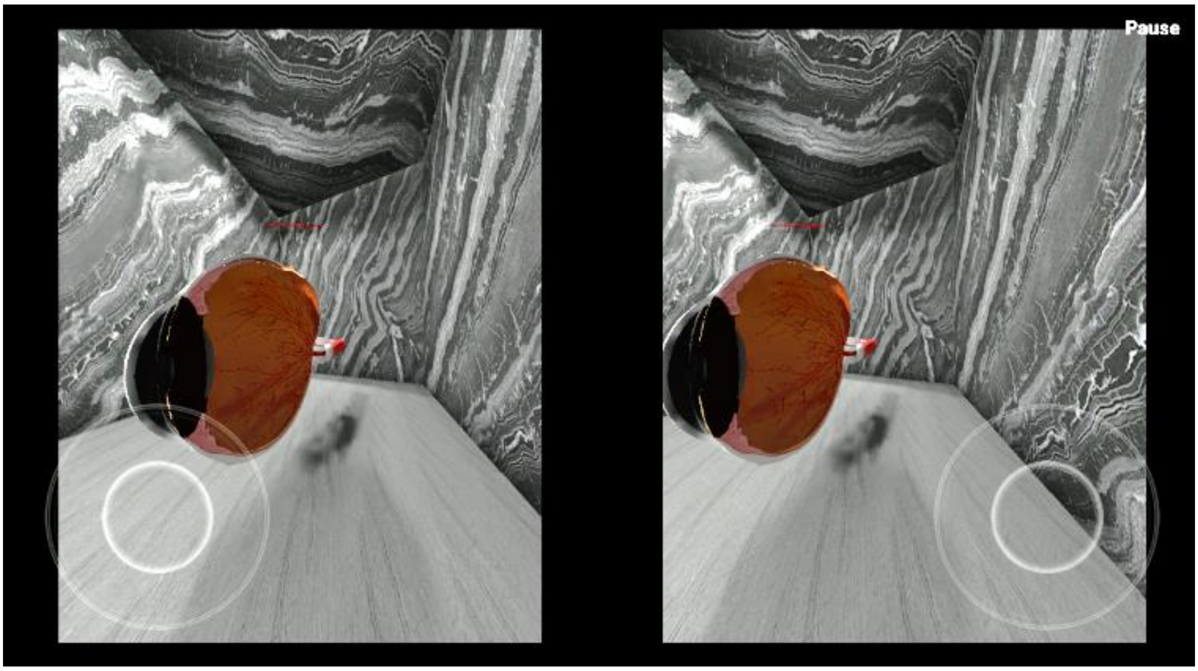

Figure 18. View of Eye

This is the in-game view of the human eye. In this section we can see the cross section of a human eye. With all its individual parts.

4 Conclusion

In conclusion, this application is for teaching medical students about human body parts in 3D. The required hardware and software tools have been discussed. The use of the VR system for medical study purposes was discussed. The results of the proposed system have been shown and analyzed. The whole system is working correctly. Medical students will benefit from the virtual reality application for their studies and learning purposes. VR technology is a cost-efficient learning strategy to continually practice several simulated clinical scenarios in healthcare and is an interactive and practical experiential learning tool for medical students. Teachers will be able to demonstrate human anatomy as necessary to students without leaving things to the imagination. In the future, we will add more organs of the human body to the system. We have developed a VR app for medical students and doctors, which will greatly benefit them. All the necessary hardware and software requirements are discussed above and shown in the figures. The application is functioning properly and is complete.

5 References

[1] Izard, S G, et al. “Virtual Reality Educational Tool for Human Anatomy.” Current Neurology and Neuroscience Reports., U.S. National Library of Medicine, May 2017.

[2] B. Rothbaum, "Effectiveness of Computer- Generated (Virtual Reality) Graded Exposure in The Treatment of Acrophobia", Am. J. Psychiatry, Pp. 626-628, Apr. 1995.

[3] M. M. Khan, M. N. Hossain Bhuiyan and T. A. Tania, "Research and Development of Virtual Reality Application for Teaching Medical Students," *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2021, pp. 01-04, doi: 10.1109/ICCCNT51525.2021.9580004

[4] Chang, E., Kim, H. T., & Yoo, B. (2020). Virtual reality sickness: a review of causes and measurements. *International Journal of Human– Computer Interaction*, *36*(17), 1658-

[5] M. Kabir, S. T. Nisa and M. M. Khan, "A Virtual Reality (VR) Based Interactive and Educative Experience of Hajj and Umrah for the People of Bangladesh," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0170- 0173, doi: 10.1109/CCWC51732.2021.9375915.

[6] N. Jahan, T. Ghani, M. Rasheduzzaman, Y. Marzan, S. H. Ridoy and M. M. Khan, "Design and Feasibility Analysis of NSUGT A Machine Learning-Based Mobile Application for Education," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0926-0929, doi: 10.1109/CCWC51732.2021.9376040

[7] Hridoy, T. Rahman, A. I. Rashique and M. M. Khan, "First Enriched Legal Database in Bangladesh with Efficient Search Optimization and Data Visualization for Law Students and Lawyers," 2021 IEEE World AI IoT Congress (AIIoT), 2021, pp. 0468-0473, doi: 10.1109/AIIoT52608.2021.9454220.

[8] H. Tamim, F. Sultana, N. Tasneem, Y. Marzan and M. M. Khan, "Class Insight: A Student Monitoring System with Real-time Updates using Face Detection and Eye Tracking," 2021 IEEE World AI IoT Congress (AIIoT), 2021, pp. 0213-0220, doi: 10.1109/AIIoT52608.2021.9454176.

[9] H. Khan, F. Soroni, S. J. Sadek Mahmood, N. Mannan and M. M. Khan, "Education System for Bangladesh Using Augmented Reality, Virtual Reality and Artificial Intelligence," 2021 IEEE World AI IoT Congress (AIIoT), 2021, pp. 0137-0142, doi: 10.1109/AIIoT52608.2021.9454247.

[10] M. M. Khan, N. Tasneem and Y. Marzan, "‘Fastest Finger First - Educational

Quiz Buzzer' Using Arduino and Seven Segment Display for Easier Detection of Participants," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 1093-1098, doi: 10.1109/CCWC51732.2021.9376139.

[11] Tasfiqul Ghani, Nusrat Jahan, Mohammad Monirujjaman Khan, S M Tahsinur Rahman and Sabik Tawsif Anjum Islam, "Development and Analysis of a Machine Learning Based Software for Assisting Online Classes During COVID-19," Journal of Software Engineering and Applications. Vol. 14 No. 3, pp. 83-94, DOI: 10.4236/jsea.2021.143006 ,2021. (Google Scholar)., 2021

[12] Mohammad Monirujjaman Khan, SM Tahsinur Rahman and Sabik Tawsif Anjum Islam, "Online Education System in Bangladesh During COVID-19 Pandemic," Creative Education (CE). Vol. 12, No. 2, PP. 441-452, 2021, DOI: 10.4236/ce.2021.122031, (Google Scholar), 2021

[13] Unreal Engine, Available on: https://unrealengine.com/

[14] Android Studio 4.0: https://developer.android.com/studio/...

[15] NetCore 3.1, Available on: https://dotnet.microsoft.com/en-us/do...

[16] NetCore 5.0, Available on: https://dotnet.microsoft.com/en-us/do...

[17] JSE8 Available on: https://www.oracle.com/java/technolog...

[18] JDK18,Available on: https://www.oracle.com/java/technolog...

[19] OpenXR, Available on: https://www.khronos.org/openxr/

[20] Visual Studio , Available on: https://visualstudio.microsoft.com/downloads/

[21] Quixel Bridge, Available on: https://quixel.com/bridge

[22] OpenHDM, Available on: http://www.openhmd.net/

[23] Blender, Available on: https://www.blender.org/