A Project Report on

Campus Walk-through Using VR

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Engineering

in

Information Technology

by

Mandar Kumbhar(17104017)
Sahil Naik(17104054)
Rutvik Lathiya(17104050)

Under the Guidance of

Prof. Kaushiki Upadhyaya
Prof. Nahid Shaikh

Department of Information Technology
NBA Accredited
A.P. Shah Institute of Technology
G.B.Road,Kasarvadavli, Thane(W), Mumbai-400615
UNIVERSITY OF MUMBAI

Academic Year 2020-2021

Approval Sheet

This Project Report entitled ***"Campus Walk-through Using VR"*** Submitted by ***"Mandar Kumbhar"(17104017), "Sahil Naik"(17104054), "Rutvik Lathiya"(17104050)*** is approved for the partial fulfillment of the requirenment for the award of the degree of ***Bachelor of Engineering*** in ***Information Technology*** from ***University of Mumbai***.

Prof. Nahid Shaikh
Co-Guide

Prof. Kaushiki Upadhyaya
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Place:A.P.Shah Institute of Technology, Thane
Date:

CERTIFICATE

This is to certify that the project entitled ***"Campus Walk-through Using VR"*** submitted by ***"Mandar Kumbhar" (17104017),"Sahil Naik" (17104054),"Rutvik Lathiya" (17104050)*** for the partial fulfillment of the requirement for award of a degree ***Bachelor of Engineering*** in ***Information Technology***,to the University of Mumbai,is a bonafide work carried out during academic year 2020-2021.

Prof. Nahid Shaikh
Co-Guide

Prof. Kaushiki Upadhyaya
Guide

Prof. Kiran Deshpande
Head Department of Information Technology

Dr. Uttam D.Kolekar
Principal

External Examiner(s)

1.

2.

Place:A.P.Shah Institute of Technology, Thane
Date:

Acknowledgement

We have great pleasure in presenting the report on **Campus Walk-through Using VR.** We take this opportunity to express our sincere thanks towards our guide **Prof. Kaushiki Upadhyaya** & Co-Guide **Prof. Nahid Shaikh** Department of IT, APSIT thane for providing the technical guidelines and suggestions regarding line of work. We would like to express our gratitude towards his constant encouragement, support and guidance through the development of project.

We thank **Prof. Kiran B. Deshpande** Head of Department,IT, APSIT for his encouragement during progress meeting and providing guidelines to write this report.

We thank **Prof. Vishal S. Badgujar** BE project co-ordinator, Department of IT, APSIT for being encouraging throughout the course and for guidance.

We also thank the entire staff of APSIT for their invaluable help rendered during the course of this work. We wish to express our deep gratitude towards all our colleagues of APSIT for their encouragement.

Mandar Kumbhar
17104017

Sahil Naik
17104054

Rutvik Lathiya
17104050

Declaration

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that We have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

(Signature)

Mandar Kumbhar 17104017
Sahil Naik 17104054
Rutvik Lathiya 17104050

Date:

Abstract

Virtual campus acts as an interactive walk-through application that is based on Virtual Reality. This paper explains all phases of building the virtual campus starting from requirement gathering to building the final application, which includes other phases such as modeling, texture mapping, and scripting. This Virtual Reality application can be used by the institute to introduce students, newcomers, to the campus with a Virtual Model of the college and make them familiar with the facilities available. It will provide a real-life walk-through of the campus while being stationary with VR ready devices. It uses Blender for preparing 3d models, unity 3d to prepare the structural model, and adding user interaction to an application through scripts and Google VR plugins.

Contents

List of Figures

List of Tables

List of Abbreviations

VR:	Virtual Reality
BMP:	Bitmap
JPEG:	Joint Photographic Experts Group
PNG:	Portable Graphics Format

Chapter 1

Introduction

VR can immerse us in a virtual world of our own making through computer generated graphics. Through VR we can experience any type of environment or imagination which is not possible to visualize in the real world. VR can be considered as a combination of various technologies like multi-sensor, multimedia, computer graphics, artificial intelligence, etc. This virtual campus is a simulation of the real college campus which will provide an actual feel of visiting the campus in real life. In this, we are going to display the three-dimensional model of our college to the user in virtual reality view. For making 3d models we have used blender and unity 3d tools. To make virtual roaming more efficient and realistic we have added interaction mechanisms that include teleporting, displaying information on pointing to a particular object, entry and exit mechanism to enter and exit a particular room. For this interaction mechanism, we have used Google VR plugins and some scripts to handle the interaction. This application can be installed on user's smartphones and with the help of google cardboard they can have a virtual tour of our campus or they can also use an Oculus instrument or any other VR device.

1.1 Objectives

- To provide an interactive walk-through of college.

- To provide users with an actual feel and experience as there are visiting the campus in real life.

- Displaying correct information about facilities and providing a navigation system to the user for efficient virtual roaming.

- Using teleporting technique for virtual roaming other than controllers to avoid motion sickness.

1.2 Problem Definition

- To create a Virtual Reality application for the institute to introduce students newcomers to the institute with a Virtual Model of the college and make them familiar with the facilities available.

- To provide a real life walk-through of the campus while being stationary with VR ready devices.

1.3 Scope

If a student wishes to visit college to see the facilities during Pandemic, they would be unable to do so. They won't be able to experience anything on their own as a result of this. All of this will be possible with our system because the student will be able to take a virtual tour of the college.

1.4 Technology stack

Software Requirements:

- Unity 3D

- Blender

  - Texture Mapping- UV Mapping

  - Rendering Engine- Eevee and Cycles

Hardware Requirements:

- VR Headset

- VR Controllers

Chapter 2

Literature Review

- Gilson Giraldi et al. (2016), This paper has represented the basics of virtual reality and describes its application, how to interact with 3D computer-generated worlds. Also discussed virtual reality applied to scientific, visualization, medicine, and engineering and main perspectives of virtual reality and presents various virtual reality devices like Head mounted Displays, VR glasses, Crystal Eyes, Data glove, Cyber puck, explain various VR system like Cave immerse Desk, Infinity wall, Collaboration VR system.

- Xue-qin Chang et al. (2016), This paper describes web 3d technology as a system based on virtual reality, which can be implemented by the method of system engineering design. This system is a simulation of real-life campus of college where students can attend class virtually, do activities and make friends. The teacher can take lectures, can review the work of students and also can take an online test.

- Jorge Martin Gutierrez et al. (2016), This Paper explains the various applications of virtual reality. It has explained how virtual reality can be implemented in various sectors like tourism, medicine, industry, video games, or education. As in educational institutions, virtual technology can benefit to teach content that is impossible to visualize in a physical classroom. It has also explained how these virtual technologies will break the boundaries of visualizing and implementation of formal education

- Chris Christou (2015), This paper describes the key features of VR that allows multisensory interaction within the visualized space and explain the various technologies that can be used to visualize and provide interaction in the Virtual world.

Chapter 3

Project Design

3.1 Design Flow

This Flow includes the phases of our project.

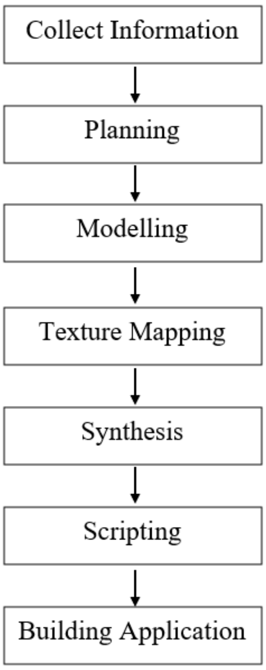

Figure 3.1: Design(Flow Of Modules)

- Collect Information
  - In this phase, we are going to gather all the required information about the college campus of which we are going to build the virtual walk-through. This information will include all the surrounding elements or objects of campus and also the interior structure of the college. For example, the number of floors per building, dimension of each floor, dimension of the classroom, location of cafeteria and labs, etc. This information will help build the accurate 3D model of the college.

- Planning
  - In this phase, we will do planning based on the collected information. This includes which software, render engine, or texture mapping tools can be preferred and also on what kind of VR devices we are planning to deploy. According to this the performance and budget will be calculated and also decide about the sequence in which the task should be performed to get the best possible output.
- Modelling
  - Based on collected information of the various 3D object of the college campus we will start making the required 3D models. We will use blender and unity 3D for modeling. Small models like chairs, computers, AC, benches, projectors, etc. will be developed in Blender and structural components like classroom structure, floor, and building structure will be directly built in unity 3D.
- Texture Mapping
  - After our 3D models are ready, we will process this model for texture mapping as it will provide a realistic look to our virtual model. This texture mapping is done by including Materials, Shaders, Textures, and UVs. You can apply Materials to make your floor look like it's made of tile, wood, stone, or anything else that is required. Using ProBuilder we can apply a Material to the entire Mesh, or just on selected faces. This allows us to add more realistic-looking surfaces during walkthrough and visualization. For example, we will use tiles textures on the floor, brick texture for walls, and stone texture so on. Some Materials use Textures, which are bitmap images (for example, Unity can import BMP, JPEG, PNG, and most standard 2-dimensional image formats). To achieve a more realistic result these images are projected on the surfaces of the Mesh by unity.
- Synthesis
  - In this phase, we will combine all separate objects or models to make a scene of a virtual campus. As we have developed some models in Blender and some structure in unity 3d we will combine this by importing all model files in unity 3d and adjust their transformation according to information gathered to have a perfect setup of a scene similar to our college.
- Scripting
  - The behavior of game objects is controlled by the components that are attached to them. By using scripts, we can trigger an event that can modify component properties of the game object over time and respond to user input in any way we like. By Scripts, we can make the walk-through more interactive to users. The basic walk-through includes the head movement, roaming or traveling around the surrounding, interacting with the surrounding objects, showing appropriate directions (Navigation System), etc. The Google VR plugins have some predefined scripts for some basic interactions.

- Building Application
  - After modeling and scripting, our final virtual model is ready, now it's time to build the application. As unity is cross-platform we can build our application for any platform of our choice, most of the VR device contains android and ios operating system. As we have made the application fully based on Google VR so the user can install this application on their smartphones and with the help of google cardboard, they can have a virtual tour of the campus or they can also use Oculus instrument or any other VR devices.

3.2 Use Case Diagram

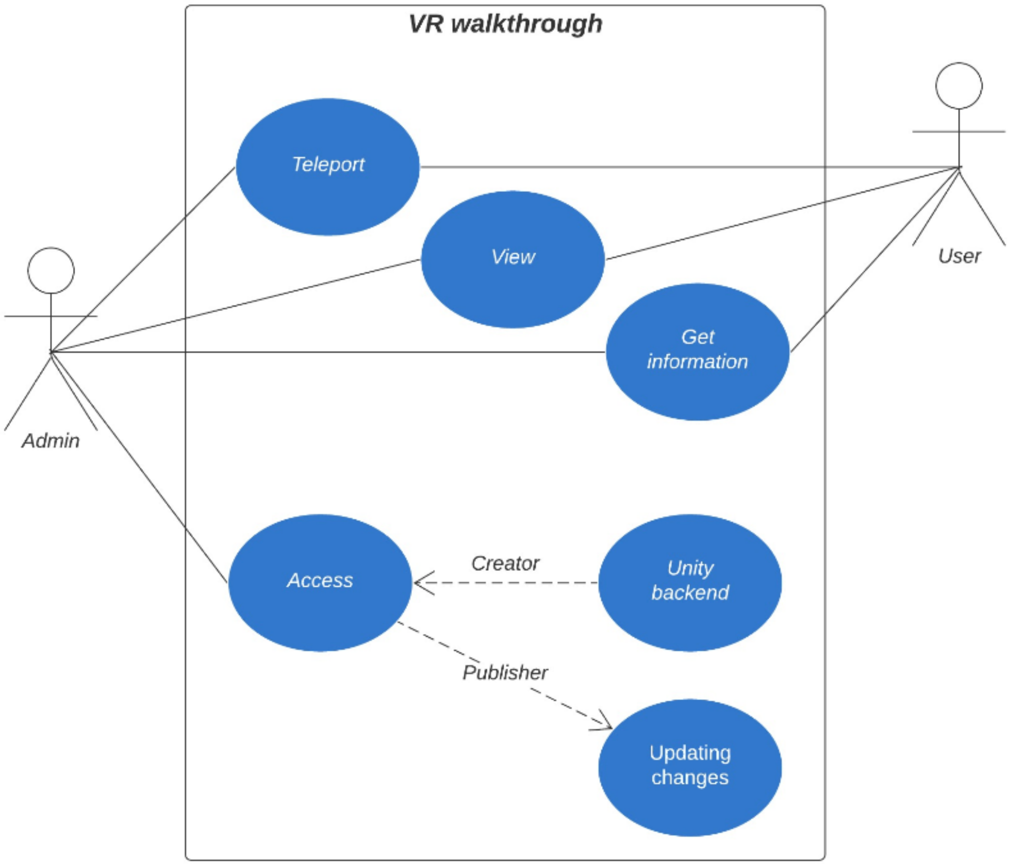

Figure 3.2: Use Case Diagram

Admin and User are the actors in the above diagram, and User can Teleport, View, and Get Information, while Admin can access the Unity back-end and update changes.

3.3 Activity Diagram

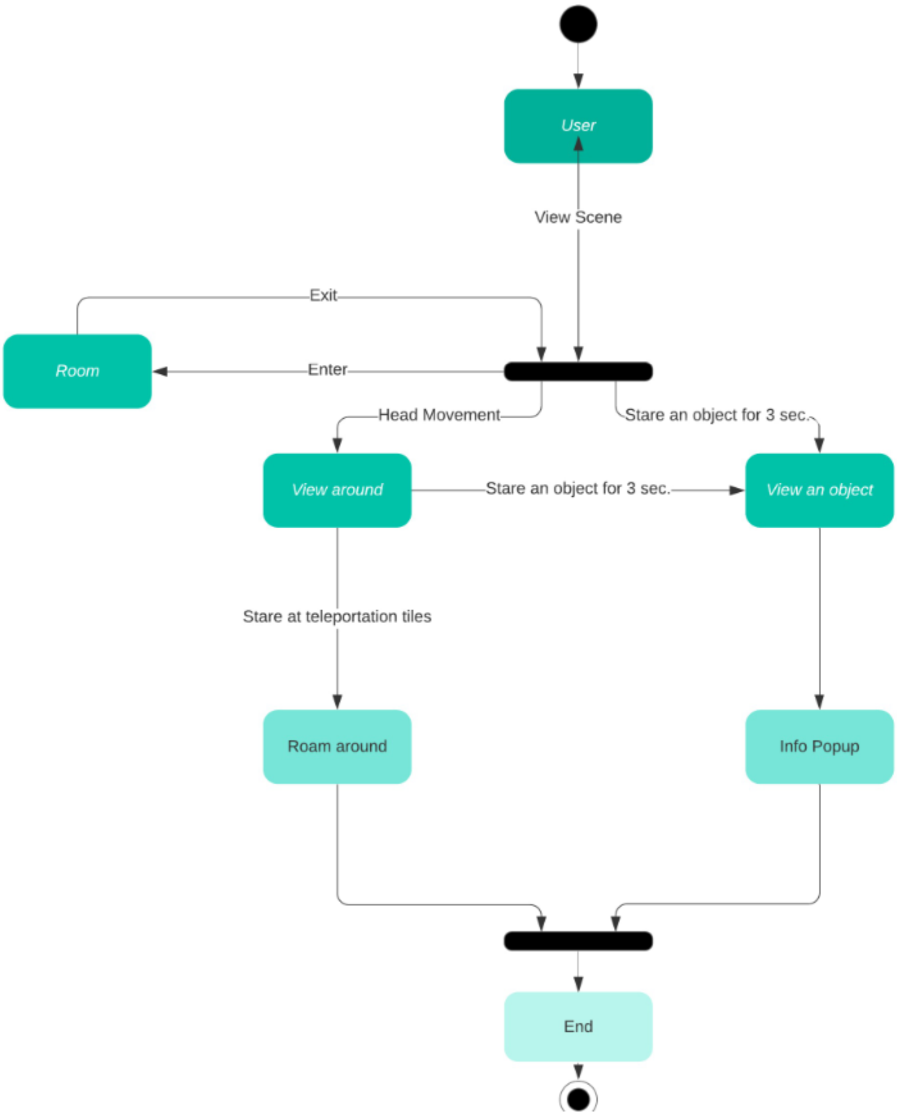

Figure 3.3: Activity diagram

3.4 Class Diagram

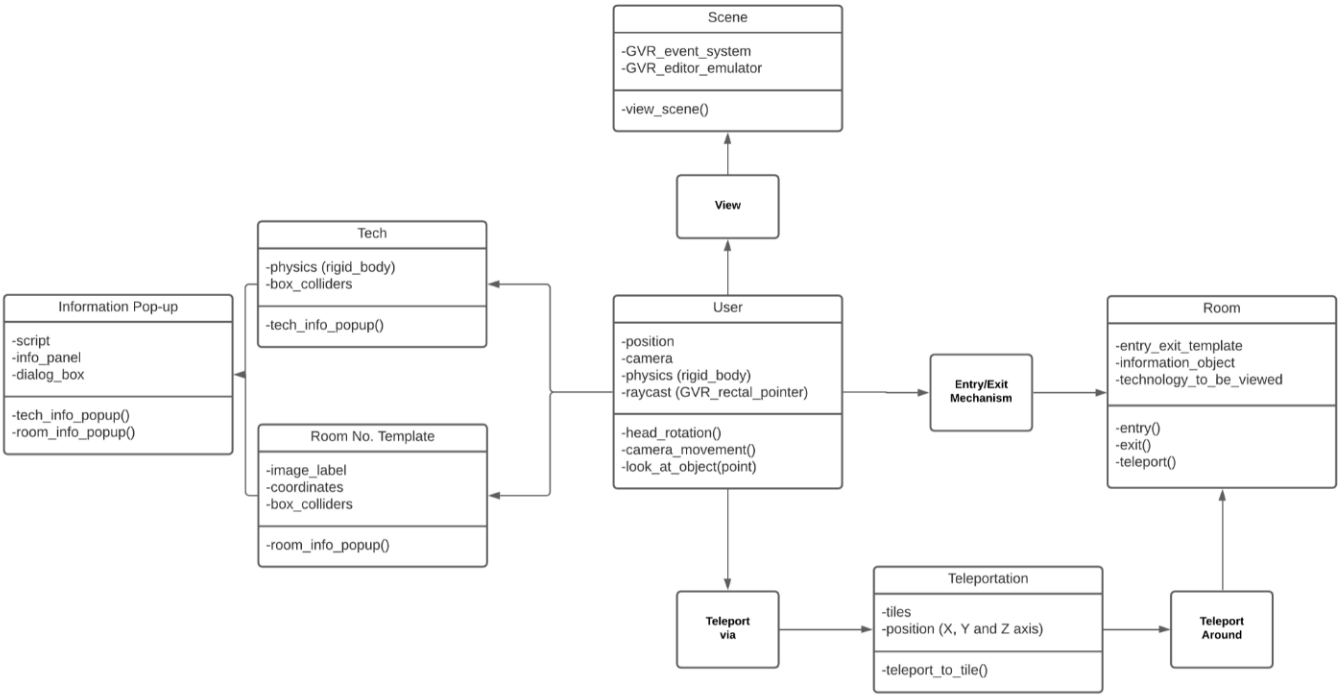

Figure 3.4: Class Diagram

The class diagram for this project is shown above, and it contains different scripts, their attributes, and their methods which directly relate to the main class User.

Chapter 4

Project Implementation

- Head Movement: This is the most important interaction in the walkthrough as the user should be able to have a full 360-degree view of the surrounding. For this we can use the GvrEditorEmulator prefab from Google VR plugins, it includes the GvrEditorEmulator script component that is used for controlling head movement and controller movement in the unity 3d. When it is enabled in your scene in play mode in the unity 3d, it allows you to control head movement by moving head-on VR head mount and moving the mouse while holding the ALT or CTRL keys on the Unity editor window, and emulator controller movement using the Controller Emulator companion app.

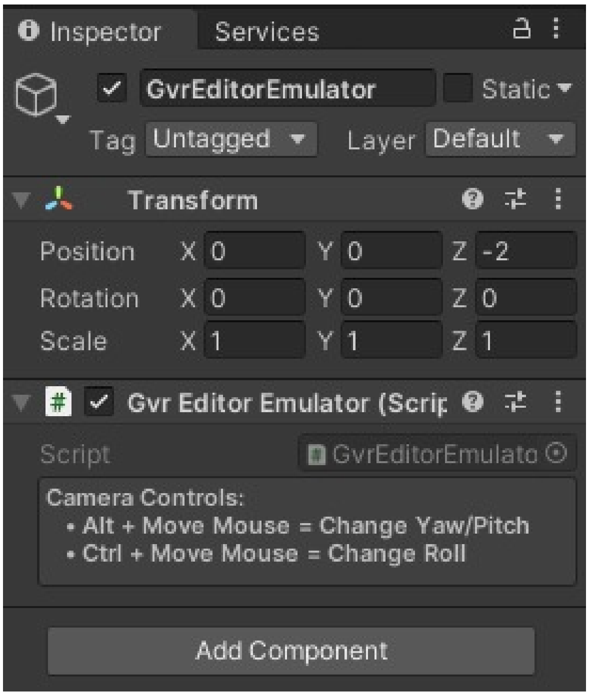

Figure 4.1: Head Movement

- Teleporting: This can be achieved either by using XR plugins if using VR controllers or by Google VR plugins if using google cardboard VR. In this project we will use Google VR plugins, to teleport we have added the tiles on which if the users see for 2 to 3 sec it will be teleported to the location of that tile. For this, we have use GvrEventSystem which is Container for the EventSystem and GvrPointerInputModule components, and GvrReticlePointer which is used to renders the gaze reticle. It can be disabled or removed through the unity editor window manually or by writing scripts. In this project, we have made it as a child of the main camera, so that the pointer and reticle follow the user's gaze. For using the Google VR input system, the Gvr-ControllerMain prefab is required in the scene. As when the user's gaze intersects with an interactive object it automatically adjusts reticle size in the scene. Following script checks the gvrStatus from GvrReticlePointer and call the Teleport method.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Teleport : MonoBehaviour
{
    public GameObject player;

    public void TeleportPlayer()
    {
        player.transform.position = new Vector3(transform.position.x, transform.position.y + 16.5f, transform.position.z);
    }

}
```

Figure 4.2: Teleport Script

- Interacting with the surrounding objects: This includes displaying information of the object on pointing to that object, for example in a lab on pointing the pointer on the computer system or other components the specification information will be displayed. For this, we can add a collider to that object and check for GvrReticlePointer to trigger the collider on collision call the following info popup script to enable the information panel or disable the information panel if not triggered.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class infopopup : MonoBehaviour
{
    public GameObject Panel;

    public void OpenPanel()
    {
        if (Panel != null)
        {
            Panel.SetActive(true);
        }
    }

    public void ClosePanel()
    {
        if (Panel != null)
        {
            Panel.SetActive(false);
        }
    }
}
```

Figure 4.3: InfoPopup Script

- For the entry-exit mechanism (to enter or exit the room) we can use the same logic of transformation that we used in teleporting. Following script checks the gvrStatus from GvrReticlePointer and call entry and exit method.

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

public class Entryexit : MonoBehaviour
{
    // Start is called before the first frame update
    public GameObject player;

    public void EnterPlayer()
    {
        player.transform.position = new Vector3(transform.position.x + 16.5f, transform.position.y - 2.5f, transform.position.z);
    }

    public void ExitPlayer()
    {
        player.transform.position = new Vector3(transform.position.x - 16.5f, transform.position.y - 2.5f, transform.position.z);
    }
}
```

Figure 4.4: EntryExit Script

- VR Gaze

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.UI;
public class VRgaze : MonoBehaviour
{
    public Image imgGaze;
    public float totalTime = 2;
    bool gvrStatus;
    float gvrTimer;

    public int distanceOfRay = 10;
    private RaycastHit _hit;

    void Start()
    {

    }

    // Update is called once per frame
    void Update()
    {
        if (gvrStatus)
        {
            gvrTimer += Time.deltaTime;
            imgGaze.fillAmount = gvrTimer / totalTime;
        }

        Ray ray = Camera.main.ViewportPointToRay(new Vector3(0.5f, 0.5f, 0f));
```

Figure 4.5: VrGaze

```
        if (Physics.Raycast(ray, out _hit, distanceOfRay))
        {
            if (imgGaze.fillAmount == 1 && _hit.transform.CompareTag("Teleport"))
            {
                _hit.transform.gameObject.GetComponent<Teleport>().TeleportPlayer();
            }
            //if (imgGaze.fillAmount == 1 && _hit.transform.CompareTag("Tv"))
            //{
            // _hit.transform.gameObject.GetComponent<Video>().StartVideo();
            //}
            if (imgGaze.fillAmount == 1 && _hit.transform.CompareTag("Entry"))
            {
                _hit.transform.gameObject.GetComponent<Entryexit>().EnterPlayer();
            }
            if (imgGaze.fillAmount == 1 && _hit.transform.CompareTag("Exit"))
            {
                _hit.transform.gameObject.GetComponent<Entryexit>().ExitPlayer();
            }
            if (imgGaze.fillAmount == 1 && _hit.transform.CompareTag("EntryR"))
            {
                _hit.transform.gameObject.GetComponent<EntryexitR>().EnterPlayer();
            }
            if (imgGaze.fillAmount == 1 && _hit.transform.CompareTag("ExitR"))
            {
                _hit.transform.gameObject.GetComponent<EntryexitR>().ExitPlayer();
            }
        }

    }
    public void GVROn()
    {
        gvrStatus = true;
    }

    public void GVROff()
    {
        gvrStatus = false;
        gvrTimer = 0;
        imgGaze.fillAmount = 0;
    }
}
```

Figure 4.6: VrGaze_2

Chapter 5

Testing

Test No.	Test Name	Expected Result	Actual Result
1.	Head movement	To look X-Y-Z Direction	Successful
2.	Head rotation	To move 360 degrees	Successful

Table 5.1: Test Case 1: Basic Functionalities

Test No.	Test Name	Expected Result	Actual Result
1.	App Scalability	Assets to be stored on cloud	Assets on collab cloud
2.	Load asset	Assets to be loaded for cloud	Assets on local machine
3.	Teleport	From Tile to Tile	Successful
4.	Tech info popup	In form of dialog box	Successful
5.	Room no info	In form of dialog box	Successful
6.	Entry/ Exit Room	By using door mechanism	By using teleportation

Table 5.2: Test Case 2: Main App Functionalities

Test No.	Test Name	Expected Result	Actual Result
1.	Detect tech object	Detect on Raycast	Successful
2.	Detect room no plate	Detect on Raycast	Successful
3.	Detect teleporting tiles	Detect on Raycast	Successful

Table 5.3: Test Case 3: Detection Functionalities

Test No.	Test Name	Expected Result	Actual Result
1.	Texture	To be loaded properly	Successful
2.	Info display canvas	In form of dialog box	Successful
3.	Gaze Pointer	Point of detection	Detecting object successfully

Table 5.4: Test Case 4: UI Functionalities

Chapter 6

Result

- Scene of room 317.

Figure 6.1: Room 317

- Implemented the teleporting tiles to teleport around the scene to provide a real life walk-through.

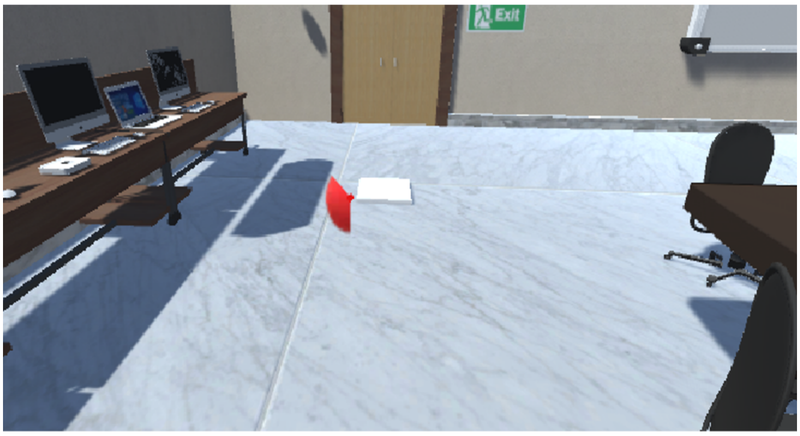

Figure 6.2: Teleport

- Implemented the information panel popping up on seeing on a specific object in a scene.

Figure 6.3: InfoPopup

- Implemented the entry and exit mechanism to enter and exit a room in scene.

Figure 6.4: EntryExit

• Room Information

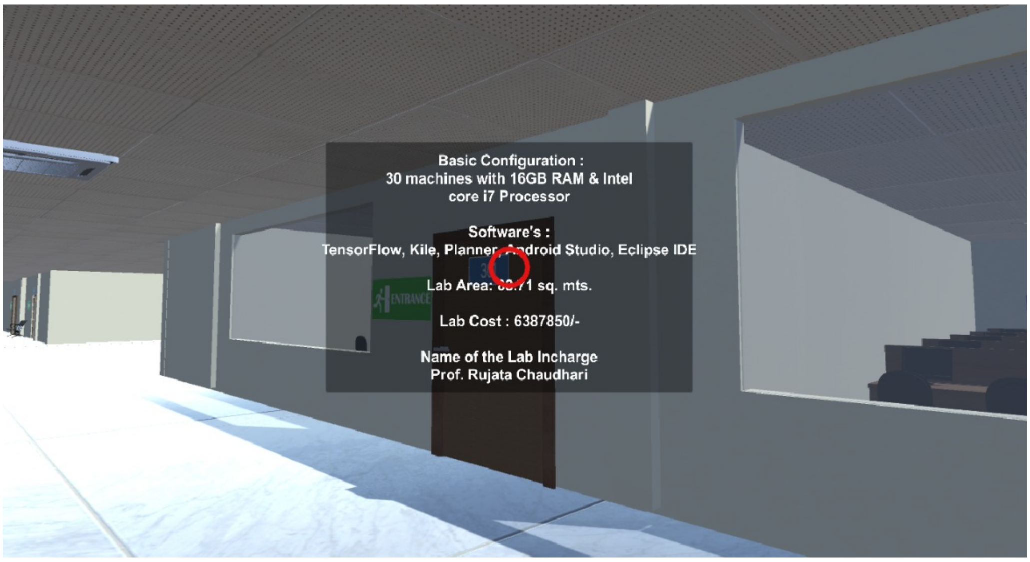

Figure 6.5: Room Info

Chapter 7

7.1 Conclusion

By just sitting at one place in reality the user can have experience of virtual roaming through this VR application. This application has provided interactive walk-through through various interaction mechanisms which include teleporting technique for virtual roaming other than controllers to avoid motion sickness, displaying information on pointing to a particular object, entry and exit mechanism to enter and exit a particular room. Also learned modeling technologies like Blender and unity 3d and about various prefabs and scripts of Google VR plugins. Through this application, users can get information about the facilities provided by college and experience college campus as they are visiting it in real life.

7.2 Future Scope

In addition to this, we can also add more features to our application like security in terms of authentication and we can also add an animation of tour guide which will guide us in our whole tour. An artificial intelligence system could also be implemented so if the user has some questions regarding college, it could be answered by this system.

7.3 Benefits for Environment and Society

- This Application can be used anywhere and anytime without the need of commuting to the actual place , thus saving energy and time.
- Also this can be beneficial for people willing to take admission in the college and getting to know the amenities during pandemic.
- A clear idea of the infrastructure can be obtained through virtual reality visualisation which would be helpful for infrastructure reference in the future.

Bibliography

[1] Ziguang Sun, Qin Wang and Zengfang Zhang," Interactive walk-through of the virtual campus based on VRML," Computer-Aided Industrial Design and Conceptual Design, 2008. CAID/CD 2008. 9th International Conference on, Kunming, 2008, pp. 456-458.

[2] J. Jie, K. Yang and S. Haihui," Study on the Virtual Natural Landscape Walkthrough by Using Unity 3D," Computational and Information Sciences (ICCIS), 2013 Fifth International Conference on, Shiyang, 2013, pp. 1-4.

[3] Xue-qin Chang, Dao-hua ZHANG and Xin-xin JIN. Application of Virtual Reality Technology in Distance Learning. International Journal of Emerging Technologies in Learning (iJET), [S.l.], v. 11, n. 11, p. pp. 76-79, Nov. 2016. ISSN 1863-0383.

[4] Gilson Giraldi, R. S. (n.d.), "Introduction to virtual reality https://doi.org/10.1109/VR.2003.119117.

[5] Martin-Gutierrez, J., Mora, C. E., Anorbe-Diaz, B., and GonzalezMarrero, A. (2017)," Virtual technologies trends in education", Eurasia Journal of Mathematics, Science and Technology Education, 13(2), 469–486.https://doi.org/10.12973/ eurasia.2017.00626 ISSN No 1305- 8223.

Appendices

Steps For Installation of Required Softwares.

Appendix-A: Blender Download and Installation

1. Navigate to the Blender website.

2. You can click on this link to go directly to the Blender download page. https://www.blender.org

3.. Run the installer. It will usually be located in your download folder.

4.. Start the installation process. Click Next Button.

5. Agree to the Terms and Conditions. When prompted, click I Agree in the installation wizard.

6. . Choose a destination folder. The default location is the Program Files folder of Drive C: in your computer. You may change the location of the application by clicking Browse.... Then click Install.

7. Wait for Blender to install. The status bar indicates the progress of the installation. Depending on your computer processor and speed, it will probably take less than 5 minutes.

8. Finish the installation. Click Finish when the installation is completed. Blender is now installed in your computer. You can start exploring the application when it starts automatically.

Appendix-B: : Unity 3D Setup environment for Windows

1. Download and install the Unity Editor from the link: https://unity3d.com/get-unity/download/archi

2. Clicking on the Download (for Windows) button, will show a drop down list of options - Unity Editor (64-bit) - Unity Installer - Unity Editor (32-bit)

3.. Select Unity Editor(64-bit) package.

4.The installer uses a Download Assistant and has detailed instructions that you need to follow. Unity Download Assistant is a light weight, small sized executable (.exe) program, that will let you select the components of the Unity Editor, which you want to download and install.

5. Select the editor component to install and then click the Next button.

6. In the next step, if you're not sure which components you want to install, you can leave the default selections, click Next to continue, and follow the installer's instructions. Some of the check boxes are:
- Microsoft Visual Studio tools for Unity (is required).
- Windows Build Support (if you are planning to make Windows phone based application as well).
- Android Build Support (if you are planning to make Android based application using Unity 3D).
Rest, leave default selected check-boxes.

7.Click on Next Button after Selecting the desired check boxes. The Unity will get installed in the System.

8. Finish the installation. Click Finish when the installation is completed. Blender is now installed in your computer. You can start exploring the application when it starts automatically.

Publication

We have submitted our paper in three conferences that is Springer ICTIS 2021, IEEE IC-SCCC 2021, and Springer ICACDS 2021.