Vidyavardhini's College of Engineering & Technology Department of Computer Science and Engineering (Data Science)

Name:	Nimish Vartak
Roll No & Branch:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	07
Title:	To Create an immersive environment (living room/ battlefield/ tennis court) with only static game objects
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: To Create an immersive environment (living room/ battlefield/ tennis court) with only staticgame objects

Theory:

Augmented Reality (AR) and ARCore

Augmented Reality (AR) is an interactive experience that combines the real world with computer-generated elements, enhancing the user's perception of reality. ARCore, developedby Google, is a powerful framework for building AR applications on Android devices.

Plane Detection in ARCore

Plane detection is a fundamental feature of ARCore. It allows the application to understand the physical environment by identifying flat surfaces such as floors, walls, and tabletops. ARCore accomplishes this by analyzing the visual features and patterns in the camera feed.

Feature Points
ARCore relies on identifying feature points in the environment. These are distinct visual cues like corners, edges, and other unique patterns. By tracking these points over time, ARCore constructs a 3D map of the environment.

Surface Reconstruction
Once feature points are detected, ARCore extrapolates surfaces from the points. These surfaces represent planes in the real world. This information is invaluable for anchoring virtual objects in the correct spatial context.

Challenges in Plane Detection
Surfaces with limited texture or uniform colors may pose challenges for accurate detection. In such cases, ARCore may have difficulty identifying feature points, potentially affecting the quality of plane detection.

Depth API vs. Plane Detection
While ARCore's Depth API provides a more detailed understanding of the environment, this lab focuses on Plane Detection for simplicity. Depth API offers enhanced spatial awareness, making it suitable for advanced AR applications.

Procedure:

- Unity Project Setup
  Open Unity or download the base project.

- Add Plane Detection
  Attach AR Plane Manager to AR Session Origin.

- Create Picture Frame
  Download 3D model and image, import to Asset folder.

- Configure Picture Frame
  Create "PictureFrame" GameObject with position and scale.Set up 3D model as child with correct rotation.

- Apply Material
  Create "PictureFrameMaterial" and assign color.Apply to 3D model.

- Add Image
  Adjust image settings (Texture Type, Pixels Per Unit).
  Create child GameObject under "PictureFrame" and set scale.

- Tag Picture Frame
  Add "Spawnable" tag to "PictureFrame" GameObject.

- Spawn Object

  Create "SpawnableManager" script and attach.Assign AR Session Origin and prefab.

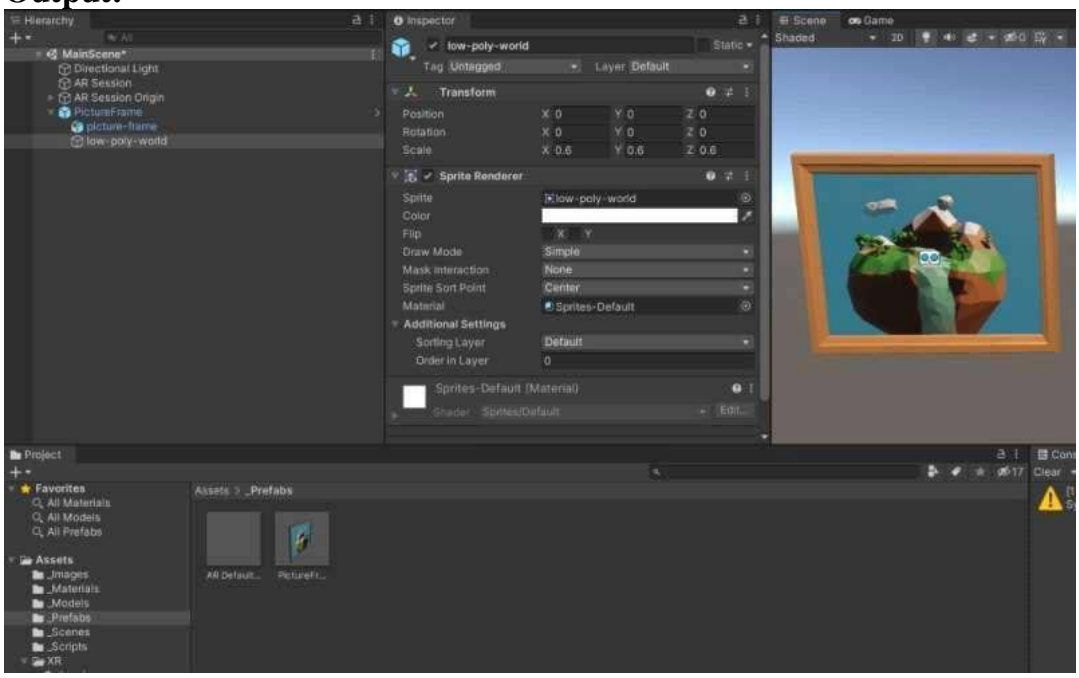
- UI Setup

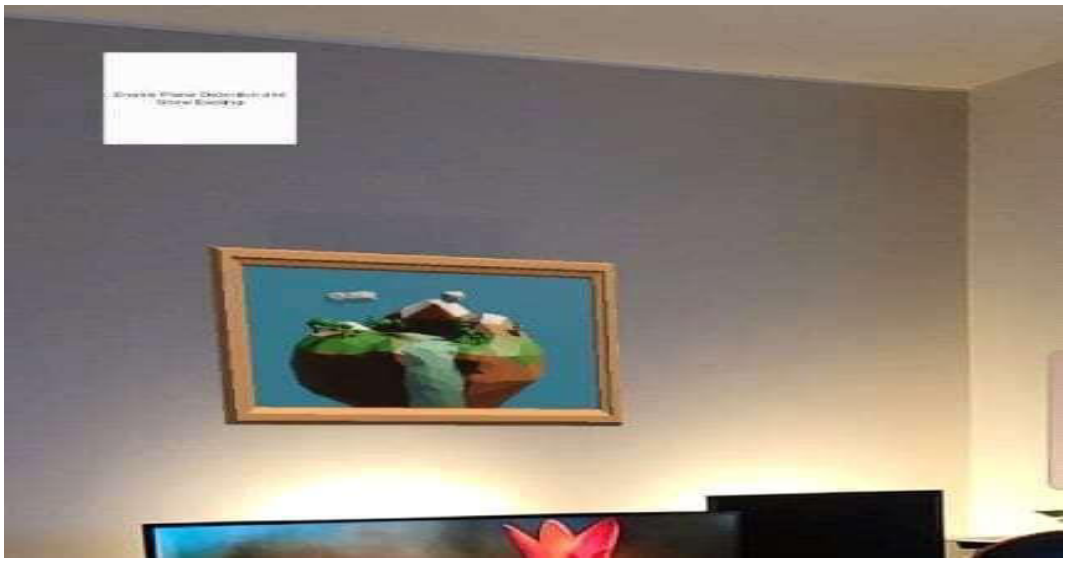
  Create "HidePlanesButton" and position.
  Anchor elements and add "PlaneDetectionController" script.

- Implement UI Interaction

  Add event to button for "TogglePlaneDetection" method.

Output:





Conclusion:

This experiment showcased the implementation of ARCore's plane detection in Unity. The application successfully identifies vertical surfaces and overlays virtual objects. By utilizing feature points and surface reconstruction, accurate spatial awareness is achieved. This projectserves as a stepping stone for creating more intricate and interactive AR experiences. It provides a solid foundation for future developments in augmented reality applications.

Vidyavardhini's College of Engineering & Technology Department of Computer Science and Engineering (Data Science)

Name:	Nimish Vartak
Roll No:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	6
Title:	Develop a simple UI(User interface) menu with images, canvas, sprites and button
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: To develop a simple UI(User interface) menu with images, canvas, sprites and button.

Theory:

Unity's XR Interaction Toolkit:
Unity's XR Interaction Toolkit is a comprehensive package designed to streamline the development of VR applications. It offers a set of tools and assets that facilitate the integration of VR functionalities across a variety of VR headsets. This toolkit simplifies complex tasks such as handling input from VR controllers, managing interactions, and creating user interfaces tailored for immersive experiences.

XR Plugin Management:
XR Plugin Management is a crucial component for ensuring compatibility with specific VR platforms. By selecting the appropriate VR plugin (such as Oculus XR Plugin),

developers establish a connection between Unity and the target VR hardware. This step is essential for enabling seamless communication between the Unity environment and the VR headset.

XR Origin and World Space:
The XR Origin serves as the central point in the virtual environment, defining the coordinate system for the entire scene. By establishing this origin, objects within the virtual space can be accurately positioned, oriented, and scaled relative to the user's perspective. Setting the Tracking Origin Mode to 'Floor' ensures that the virtual world aligns with the physical floor, enhancing realism and user comfort.

Hand Controller Configuration:
Hand controllers are fundamental to the VR experience, acting as the user's primary means of interaction. In this experiment, the XR Interaction Toolkit is utilized to customize the behavior of these controllers. By adding components like XR Direct Interactor and Sphere Collider, developers define how the controllers interact with objects in the virtual environment. This step lays the foundation for precise and intuitive user interactions.

GameManager Script:
The GameManager script serves as the backbone of the VR application, managing game logic and interactions. Through this script, developers establish the rules governing how the application responds to user inputs. In this experiment, the GameManager script orchestrates actions such as scoring points upon successful interactions, providing a dynamic and engaging user experience.

User Interface (UI) Design:
Creating an intuitive and visually appealing UI is crucial for a successful VR application. Within Unity's UI system, elements like buttons and images are arranged within a Canvas object. This canvas serves as a container for UI components, allowing developers to design a user-friendly interface that seamlessly integrates with the VR environment.

Interacting with UI Elements in VR:

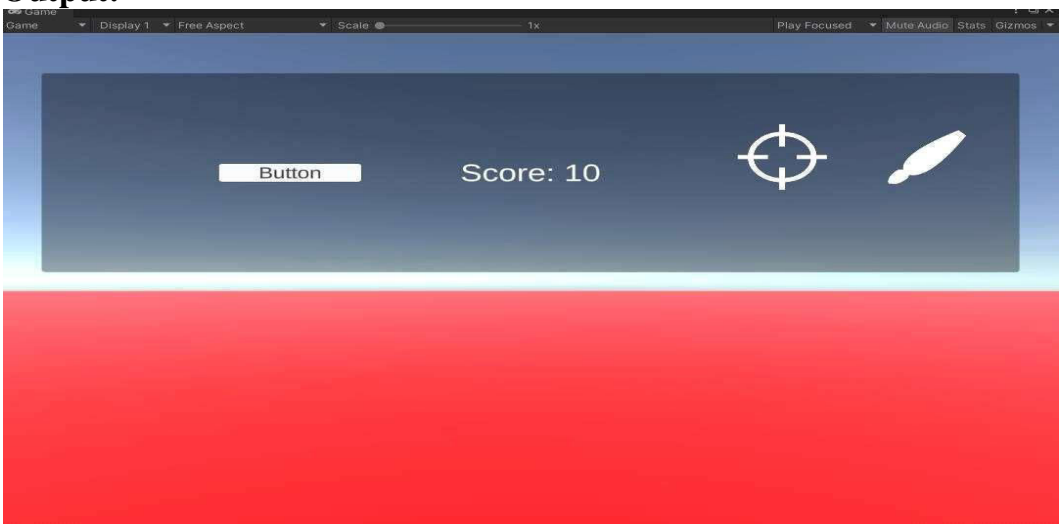
Interacting with UI elements in a VR environment introduces unique challenges. The Tracked Graphic Raycaster is added to the Canvas object, enabling the detection of user interactions within the virtual space. By integrating the GameManager script, UI elements can trigger specific actions, providing users with a natural and immersive means of navigation and interaction.

Procedure:

- Install XR Interaction Toolkit via Package Manager.
- Configure XR Plugin Management for your VR platform.
- Create an XR Origin for tracking and positioning.
- Customize hand controllers with XR Direct Interactor and Sphere Collider.
- Write a GameManager script for game logic.

- Create UI elements like Canvas, Button, and Text.
- Position and style UI elements.
- Enable VR interaction with UI through ComponentEvent Trigger.
- Implement a World Space Canvas for VR visibility.
- Add Ray Interactors for UI interaction.
- Play the game and use VR controllers to interact with UI elements while observingscore updates on the Canvas.

Output:



Conclusion:

Creating a simple UI menu in Unity with images, canvas, sprites, and buttons

demonstrates the ability to design user-friendly interfaces for games or applications. Unity's UI system provides a user-friendly environment for designing and implementing interactive menus, enhancing the overall user experience. This project showcases Unity's versatility for creating engaging and visually appealing user interfaces.

Vidyavardhini's College of Engineering & Technology Department of Computer Science and Engineering (Data Science)

Name:	Nimish Vartak
Roll No & Branch:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	05
Title:	Develop a scene in Unity that includes a sphere and plane . Apply Rigid body component, material and Box collider to the game Objects.Write aC# program to grab and throw the sphere using vr controller
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: Develop a scene in Unity that includes a sphere and plane . Apply Rigid body component, material and Box collider to the game Objects.

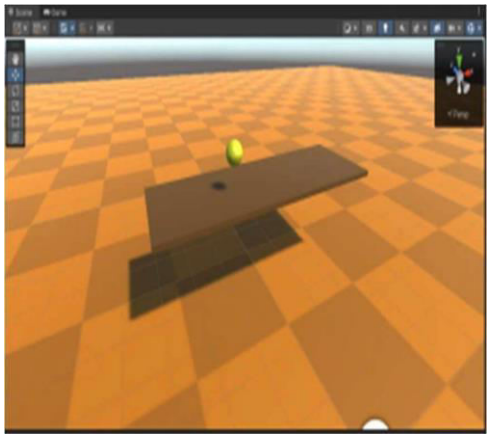
Theory: Virtual Reality (VR) is a cutting-edge technology that immerses users in digital environments, making them feel as if they are physically present in those spaces. Within Unity,a popular game development platform, it is entirely feasible to apply essential components to game objects, such as the Rigid body for simulating physical behaviours, Box Colliders for defining collision shapes, and Materials for controlling the objects' visual properties. Moreover, in Unity, developers can craft a dynamic VR experience that enables users to pick up and throw a sphere with a VR controller. This is accomplished by incorporating a VR software development kit (SDK), configuring a VR camera rig to replicate the user's perspective, and implementing custom scripts to facilitate the interaction.

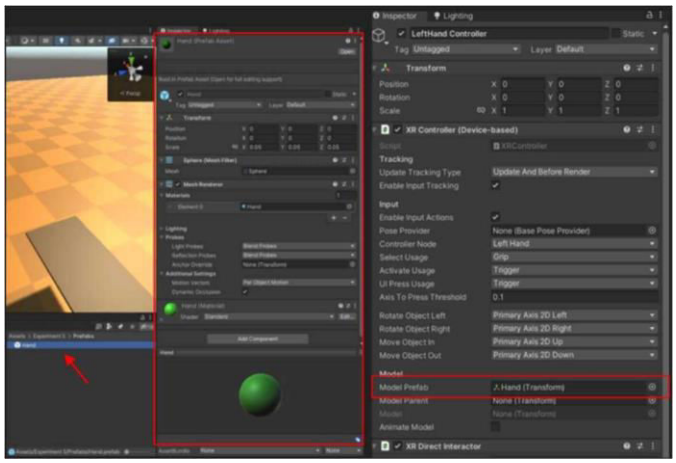
The XRGrabInteractable script, often used for this purpose, tracks the controller's input and applies forces to the Rigid body component of the sphere, allowing users to manipulate and launch it within the VR environment, fostering an engaging and immersive VR interaction.

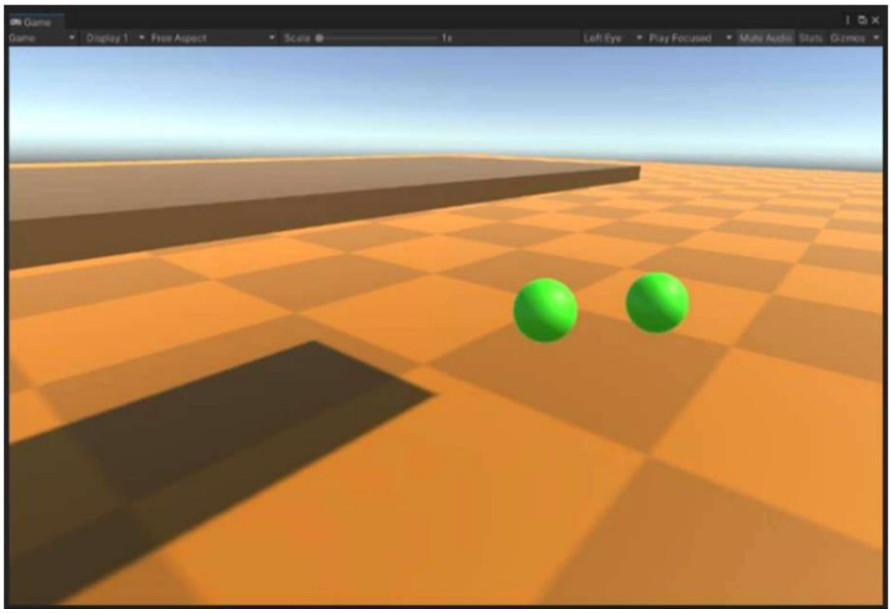
Procedure:

- Create a new Unity 3D project.
- Import VR SDK for your hardware.
- Import assets (sphere, plane, materials).
- Create sphere and plane objects.
- Add Rigid body components to both.
- Apply materials to objects.
- Add Box Collider to the sphere.
- Implement VR controller interaction using provided SDK components.
- Write a script for picking up and throwing the sphere.
- Attach the script to the VR controller object.
- Test the scene in your VR environment.

Result:







Conclusion:

Developing a Unity scene with a sphere and plane, applying the Rigid Body component and Box Collider, and customizing materials enhances the physics and visual aspects of the objects. This project highlights how Unity empowers developers to create interactive and realistic 3D environments, making it a valuable tool for game development, simulations, and various interactive applications.

Name:	Nimish Vartak
Roll No & Branch:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	04
Title:	To develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the colour, material and texture of each Game object separately in the scene. Write a C# program in visualstudio to change the colour and material/texture of the game objects dynamically on button click
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim :-

To develop a scene in Unity that includes a cube, plane and sphere. Create a new material and texture separately for three Game objects. Change the colour, material and texture of each Game object separately in the scene. Write a C# program in visual studio to change the colour and material/texture of the game objects dynamically on button click

Theory:-

Unity is a versatile game development platform that enables the creation of interactive 3D and 2D experiences. In this project, we establish a Unity scene featuring a cube, plane, and sphere while emphasizing crucial concepts like materials, textures, C# scripting, object manipulation, and user interaction. Materials define object appearance and texture application, with each object possessing its material. C# scripts enable dynamic property

changes, accessed through GameObjects, and handled via button clicks in Unity's UI system. Proper resource management and Play mode testing are integral parts of this project, serving as a foundation for developing engaging Unity applications and games.

Procedure:-

Step 1: Create an object

Select the GameObject menu in the menu bar. The GameObject Menu has several objects to create a game. Select the 3D object and pick the cube option.

The cube object will be displayed in the Scene View.

Increase the object size using the Scaling tool. Pull the arrow for the x-axis, y-axis, and z-axis. The object size can be increased.

Step 2: Create materials

Select the Assets menu in the menu bar. Select "Create" and pick the "Material" option. The material is displayed in the assets. Now change the material name to red. Select the Material (red) and click the color box. The color box is displayed on the left side. Choose the red color. The color is applied to your material (red).

Drag the red and drop it in the cube. The object can be changed into a red

color.Step 3: Create shaders and textures

Right-click on the assets, Select Import the asset.
Already I have brick wall images. If you want to apply for any other images, you can apply them. Now I have selected a brick wall image. Click to the Import button.

The brick wall image is added to assets. Select the image drag and drop the image into the object. The object can be changed fully to a brick wall. Now the new material added into the assets.

Now delete the brick wall image and double click on the new material. Delete the brick wall material. Select the material, press F2, and rename the new material to Textures.

Now, double click on the textures, open the brick wall image, and drag and drop the image into Textures.

Give the name for the Brick wall.

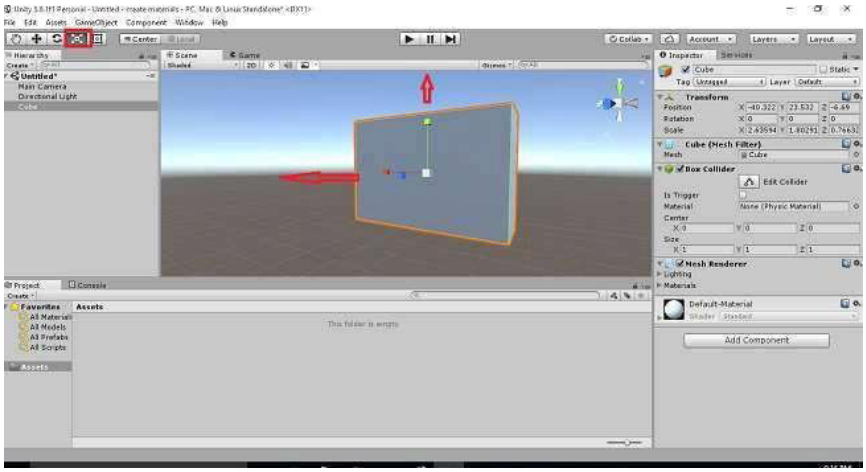
Select the Brick wall and go to the standard list box. Choose the Legacy shaders and pick the Diffuse option. When you select diffuse, the window is displayed like this.

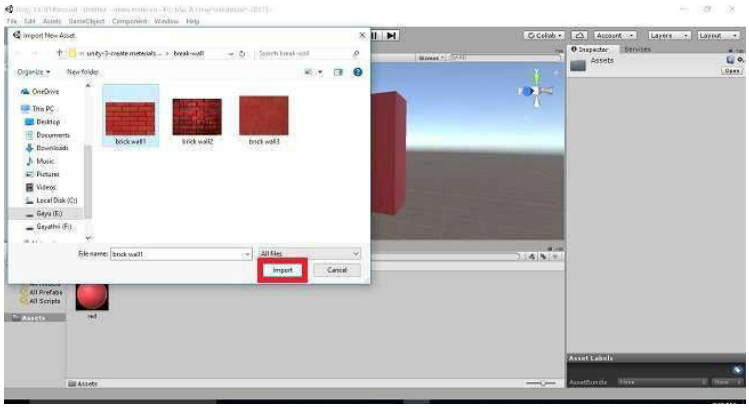
Click on the select option on your right side. Select the Texture window open on your leftside. Double click on the Brick wall. The brick wall material can be changed.
Pick the red marked tool. Press the alt key and scroll the mouse. You can see all sides in your object with a brick wall fully applied.

Step 4

Now, select the red material and go to a standard list box. Choose the Legacy shaders and picka specular option.
The object can be displayed in more shine.

Result:-





Conclusion:-

Creating a scene in Unity with a cube, plane, and sphere, and assigning separate materials and textures to each object, demonstrates versatile asset customization. Implementing a C# scriptin Visual Studio to dynamically change the color, material, and texture on button click enhances interactivity and showcases Unity's scripting capabilities. This project underscores Unity's flexibility in creating dynamic, user-controlled 3D environments for a wide range of applications, from gaming to simulations and more.

Name:	Nimish Vartak
Roll No & Branch:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	03
Title:	To develop a scene in Unity that includes: i. a cube, plane andsphere, apply transformations on the 3 game objects. ii. add a video and audio source.
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim :-

To develop a scene in Unity that includes:

- a cube, plane and sphere, apply transformations on the 3 game objects.
- ii. add a video and audio source.

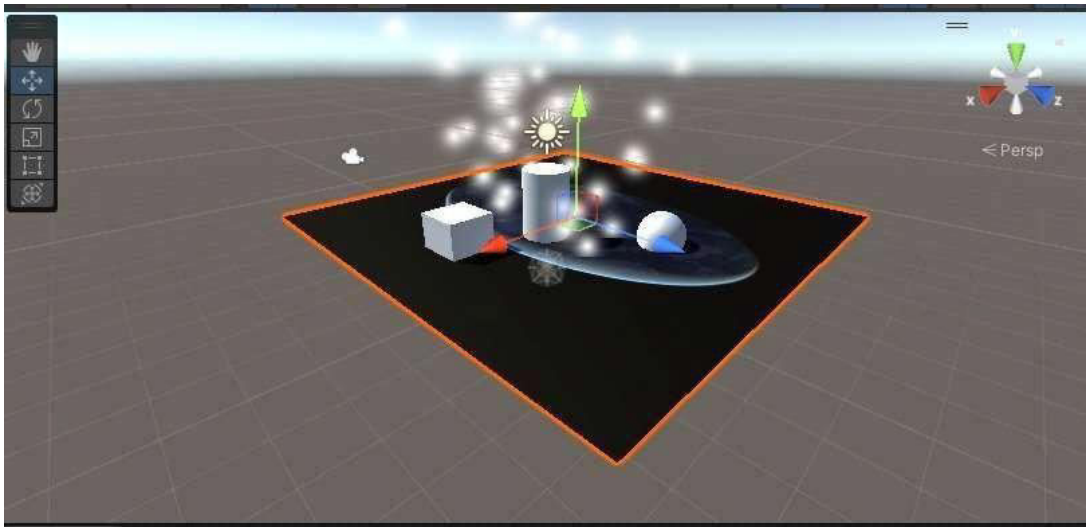
Theory:-

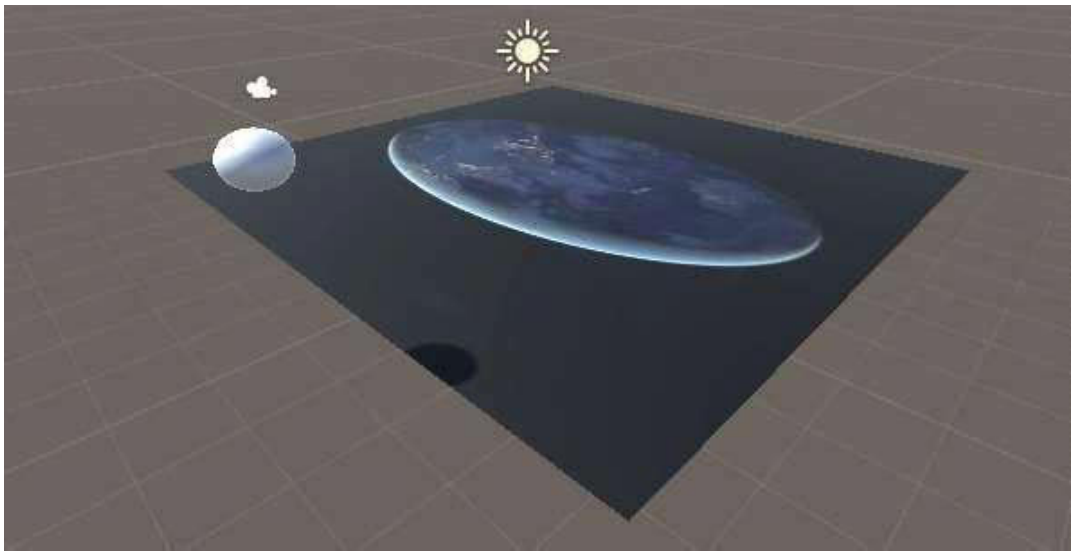
In Unity, you can create a dynamic scene by adding various game objects and components. Inthis context, we aim to create a scene that involves a cube, plane, and sphere, and apply transformations to these objects. Transformations, including translation, rotation, and scaling,alter the position, orientation, and size of game objects, respectively. This manipulation of transformations is fundamental for positioning and animating objects within the Unity environment. Additionally, we will add a video and audio source to enhance the scene's interactivity. A video source allows for the playback of video content within the scene, enriching the visual experience. Meanwhile, an audio source provides the capability to integrate sound and music, further engaging users in the immersive environment

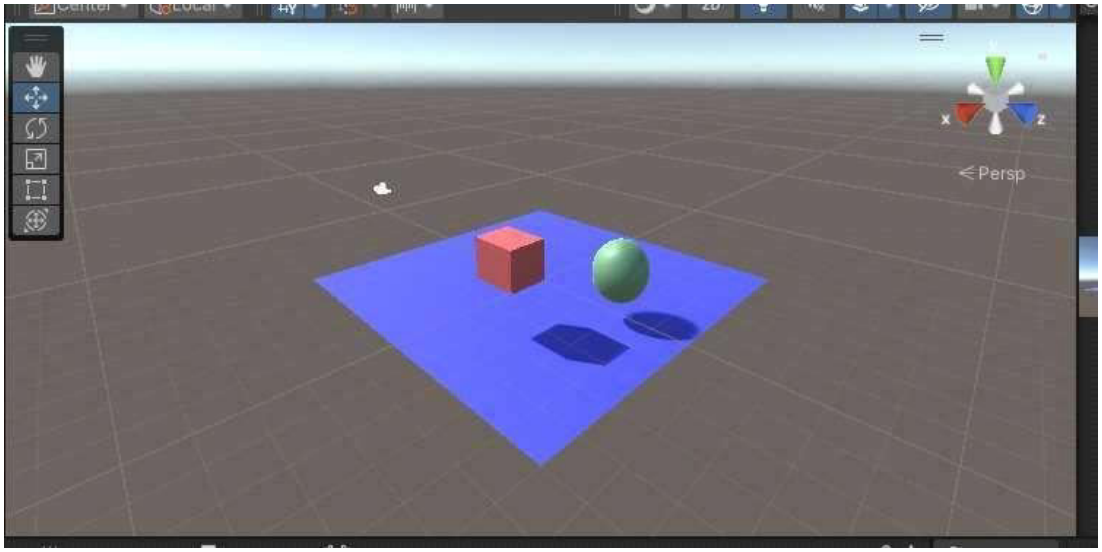
Procedure:-

- **Create Project**: Start a new 3D Unity project and ensure you have the required Unityversion and video/audio packages installed.
- **Create & Position Objects**: In the Hierarchy, create game objects (Cube, Plane,Sphere) and adjust their properties.
- **Import Assets**: Import video (MP4 or WebM) and audio files into the project's"Assets" folder.
- **Create Materials**: Generate materials for game objects by right-clicking in theProject window, then assign these materials in the Inspector.
- **Add Video & Audio Components**: For video, create a Video Player component and assign the video clip. For audio, add an Audio Source component and assign the audioclip.
- **Configure Playback**: Write scripts if needed to control video and audio playback.
- **Testing**: Save the scene and press Play to verify video/audio playback, objecttransformations, and material settings.

Result:-







Conclusion:-

Developing a scene in Unity with a cube, plane, and sphere while applying transformations showcases fundamental 3D object manipulation. Adding a video and audio source enriches the scene's multimedia elements, enabling a wide range of interactive and immersive experiences. Unity's versatility makes it a powerful tool for creating dynamic and multimedia-rich environments for various applications, from games to simulations and beyond.

Name:	Nimish Vartak
Roll No & Branch:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	02
Title:	Demonstration of the working of HTC Vive, Google Cardboard, Google Daydream and Samsung gear VR.
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim :-

Demonstration of the working of HTC Vive, Google Cardboard, Google Daydream and Samsung gearVR.

Theory:-

In this VR headset demonstration for the lab, we explore four distinct virtual reality technologies: the HTC Vive, Google Cardboard, Google Daydream, and Samsung Gear VR. Each of these headsets offers unique features and experiences. The HTC Vive is a high-end VR system that uses external sensors for room-scale tracking, providing an immersive experience with handheld controllers. Google Cardboard is an entry-level option that transforms a smartphone into a basic VR headset, making use of the phone's sensors for head tracking. Google Daydream builds upon the Cardboard concept, offering a more comfortable design and motion controller for intuitive interactions. Samsung Gear VR is tailored for Samsung Galaxy smartphones, offering a wide range of VR content through the Oculus store.During the demonstration, participants will experience these headsets, gain insights into their functionalities, and explore the diverse VR environments they can offer. This hands-on experience aims to familiarize students with a spectrum of VR technologies, from high-end setups to more accessible and smartphone-driven solutions.

HTC Vive:

VIVE, sometimes referred to as HTC Vive, is a virtual reality brand of HTC Corporation. It consists of hardware like its titular virtual reality headsets and accessories, virtual reality software and services, and initiatives that promote applications of virtual reality in sectors like business, arts, and video gaming.

The brand's first virtual reality headset, simply called HTC Vive, was introduced as part of a collaboration with Valve Corporation, implementing its SteamVR hardware and software ecosystem. It was unveiled during HTC's Mobile World Congress keynote in March 2015. Development kits were sent out in August and September 2015, and the first consumer version of the device was released in April 2016.It has since been succeeded by newer models with upgraded specifications. HTC has also released accessories that integrate with the Vive and SteamVR, including sensors for motion capture and facial capture.

Google Cardboard

Cardboard VR is compatible with a wide range of contemporary smartphones. Google Cardboard's biggest advantage is its low cost, broad hardware support, and portability. As a bonus, it is wireless. Using the phone's gyroscopes, the VR applications can track the user in 360 degrees of rotation. While modern phones are very powerful, they are not as powerful as desktop PCs. But the user is untethered and the systems are lightweight.

Google Daydream

Rather than plastic, the Daydream is built from a fabric-like material and is bundled with a Wii-like motion controller with a trackpad and buttons. It does have superior optics compared to a Cardboard but is not as nice as the higher end VR systems. Just as with the Gear VR, it works only with a very specific list of phones. Users could insert their phones into a compatible Daydream VR headset, which often came with a comfortable fabric design and a motion controller for interaction.

Daydream offered a variety of VR applications and games through the Google Play Store, making it accessible for both entertainment and educational purposes.

Samsung Gear VR

The Samsung Gear VR is a virtual reality headset developed by Samsung Electronics, in collaboration with Oculus VR, and manufactured by Samsung. The headset was released on November 27, 2015.

When in use, a compatible Samsung Galaxy device acts as the headset's display and processor, while the Gear VR unit itself acts as the controller, which contains the field of view, as well asa custom inertial measurement unit, or IMU, for rotational tracking, which connects to the smartphone via USB-C or micro-USB. The Gear VR headset also includes a touchpad and backbutton on the side, as well as a proximity sensor to detect when the headset is on.

The Gear VR was first announced on September 3, 2014.To allow developers to create content for the Gear VR and to allow VR and technology enthusiasts to get early access to the technology, Samsung had released two innovator editions of the Gear VR before the

consumer version.

Conclusion:-

HTC Vive offers immersive VR experiences with room-scale tracking, while Google Cardboard provides a basic, low-cost VR experience using a smartphone. Google Daydream offers a more advanced mobile VR experience, and Samsung Gear VR provides a comfortable and interactive VR experience for Samsung device users, each catering to different user needs and preferences. Overall, these VR platforms offer varying levels of immersion, accessibility, and compatibility, making them suitable for a wide range of virtual reality applications.

Vidyavardhini's College of Engineering & Technology Department of Computer Science and Engineering (Data Science)

Name:	Nimish Vartak
Roll No & Branch:	CSE-75
Class/Sem:	BE/VII
Experiment No.:	01
Title:	Installation of Unity and Visual Studio, setting up Unity for VRdevelopment, understanding documentation of the same.
Date of Performance:	
Date of Submission:	
Marks:	
Sign of Faculty:	

Aim: Installation of Unity and Visual Studio

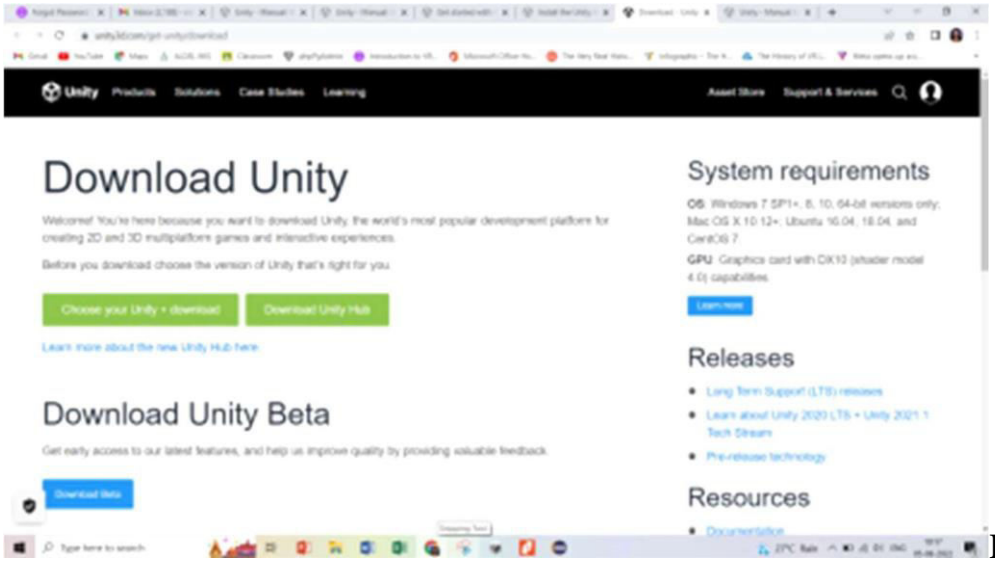
Theory:

Virtual Reality (VR) is a transformative technology that leverages computer-generated environments to immerse users in simulated realities, typically facilitated through specialized headsets. It has gained widespread use due to its ability to transport users to entirely different worlds, whether for entertainment in gaming, immersive storytelling, or the creation of lifelike training simulations for industries like aviation, medicine, and engineering. In education, VR enables dynamic and interactive learning experiences, from exploring historical sites to understanding complex scientific concepts. Additionally, it is employed in therapeutic settings, offering treatment options for conditions like PTSD or phobias. As VR technology advances, its applications continue to expand, revolutionizing how we engage with information, experiences, and virtual spaces.

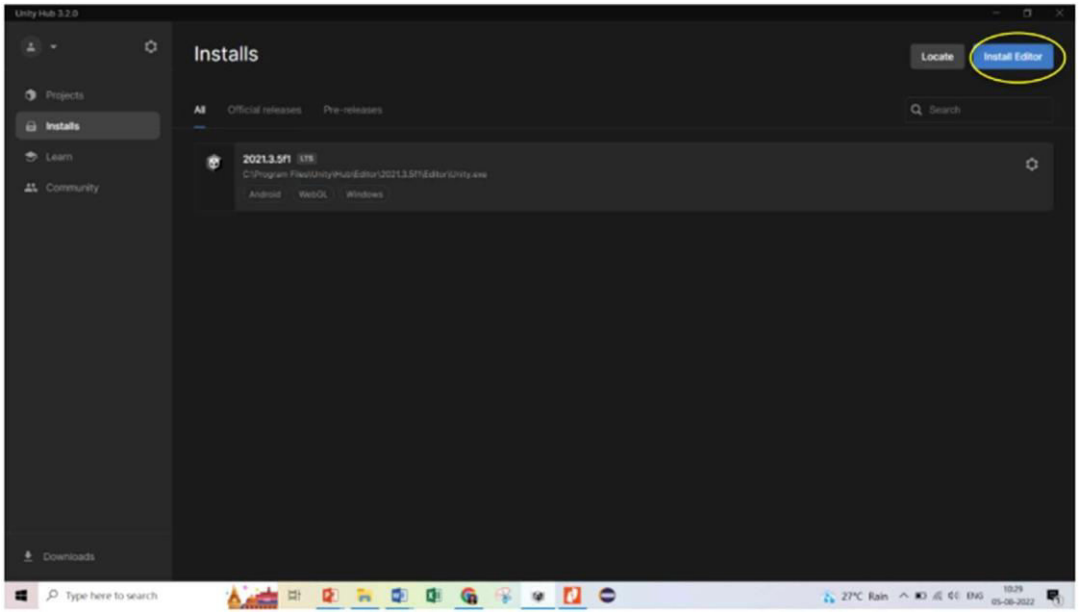
Procedure:

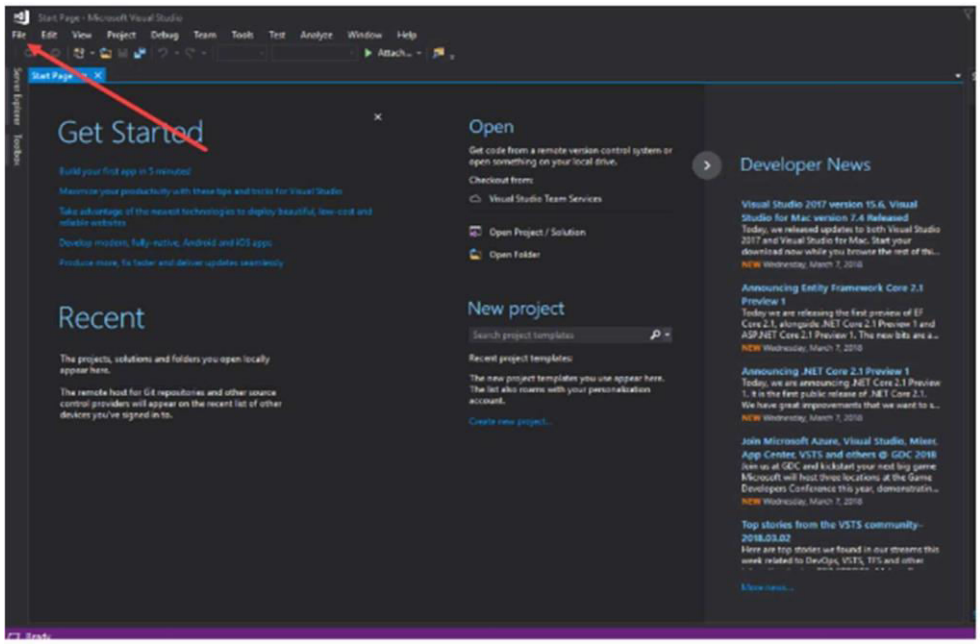
- Install Unity and Visual Studio.

- Choose your VR platform and install the respective SDK.
- Create a new Unity 3D project and enable VR support.
- Import the VR SDK and assets.
- Set up a VR camera rig (e.g., OVRPlayerController for Oculus).
- Design your VR environment with 3D models and terrain.
- Refer to official documentation for your VR platform and Unity for guidance.
- Learn how to use the SDK's features, such as hand tracking and controllers.
- Explore scripting for VR interactions (grabbing, throwing) in Unity.
- Join VR developer communities and seek tutorials or courses for additional learning andsupport.



Result:





Conclusion:

Installing Unity Hub and choosing a Unity version with necessary modules, along withinstalling Visual Studio with Unity tools, is essential for setting up a robust development environment to begin game development using the Unity engine.