



Vidyavardhini's College of Engineering & Technology  
Department of Artificial Intelligence and Data Science

---

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



**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:**



## Vidyavardhini's College of Engineering & Technology

### Department of Artificial Intelligence and Data Science

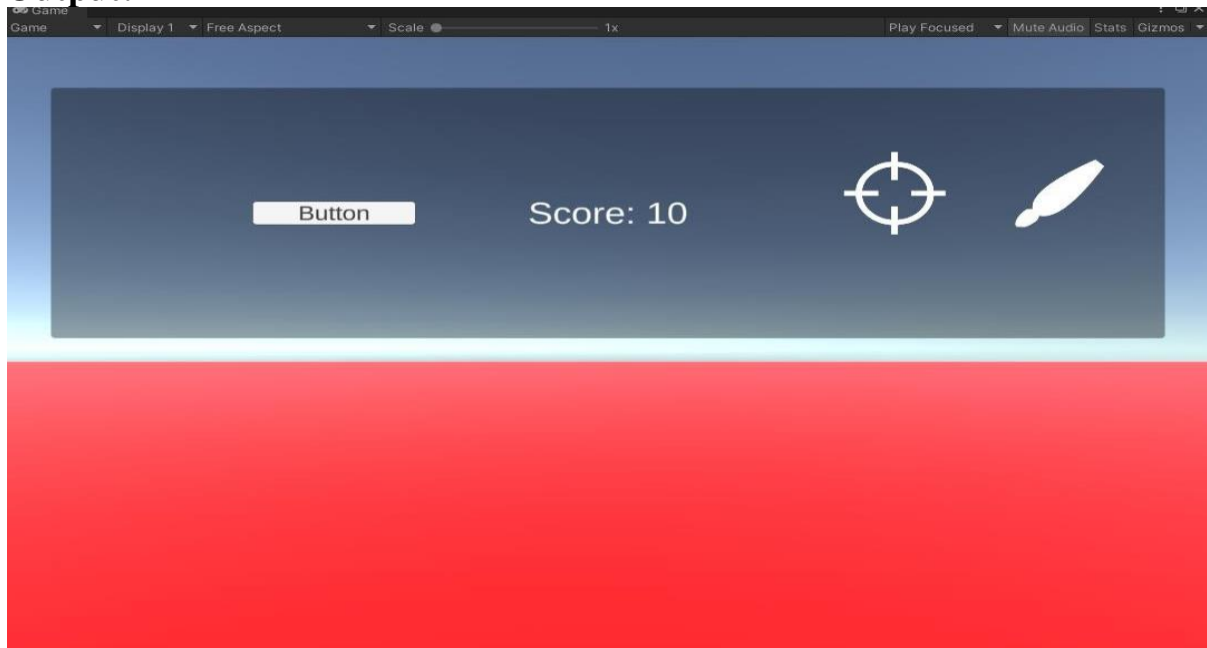
---

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:

1. Install XR Interaction Toolkit via Package Manager.
2. Configure XR Plugin Management for your VR platform.
3. Create an XR Origin for tracking and positioning.
4. Customize hand controllers with XR Direct Interactor and Sphere Collider.
5. Write a GameManager script for game logic.
6. Create UI elements like Canvas, Button, and Text.
7. Position and style UI elements.
8. Enable VR interaction with UI through ComponentEvent Trigger.
9. Implement a World Space Canvas for VR visibility.
10. Add Ray Interactors for UI interaction.
11. Play the game and use VR controllers to interact with UI elements while observing score updates on the Canvas.

### Output:





## Conclusion:

In conclusion, the development of this UI menu was an enlightening and hands-on experience that allowed me to delve into the intricacies of user interface design. Incorporating images, canvas, sprites, and buttons not only enhanced the visual appeal but also enriched the overall user experience.

The utilization of a canvas provided a versatile platform for dynamic graphic rendering, while the inclusion of sprites facilitated seamless integration of interactive elements. Buttons, as fundamental components, not only served as navigational tools but also enhanced the user's engagement by responding to their interactions.

Working through the technical aspects, I gained a profound understanding of the underlying mechanics behind each element. Image integration was a crucial aspect, offering a visually pleasing dimension to the UI. The careful consideration of image placement and alignment played a pivotal role in achieving a harmonious and aesthetically pleasing design.

Debugging and troubleshooting challenges emerged as valuable learning opportunities, reinforcing my problem-solving skills. The intricate interplay of these UI components highlighted the significance of meticulous coding and attention to detail in crafting a polished and functional interface.

In essence, this practical undertaking provided a hands-on exploration of UI design, offering insights into the technical nuances that contribute to an effective and visually appealing user interface. The amalgamation of images, canvas, sprites, and buttons not only expanded my technical skill set but also deepened my appreciation for the art and science of user-centric design.