Session 6

Grabbing Objects Using Controllers

Welcome to the session, **Grabbing Objects using Controllers**.

This session provides an introduction to grabbing objects using controllers and also illustrates how to set up Object Grab in VR Scene.

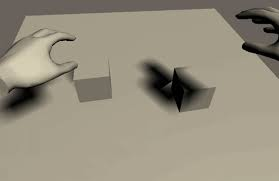
In this session, students will learn to:

* Explain local avatar prefab in Oculus Interaction SDK
* Describe how to make hands grab the objects
* Explain left and right grabbable objects

6.1 What Is Object Grab in VR Scene?

In Virtual Reality (VR), object grab is the ability to interact with virtual objects as if they were real physical objects in the real world. This means that users can pick up, move, and manipulate objects using their hands or controllers, just as they would do in real life.

To implement an object grab in Unity, developers can use a combination of physics and scripting. The physics engine in Unity can be used to detect collisions between the user’s hand or controller and the virtual object, while scripts can be used to apply forces and constraints to the object to simulate the physics of grabbing and moving it as shown in Figure 6.1.

****

**Figure 6.1: Illustration of Grabbing an Object in VR Scene**

One way to implement object grab in Unity is to use the Unity input system, which provides a unified input API that can be used for both traditional input devices (such as keyboards and gamepads) as well as VR controllers. By using the input system, developers can create scripts that detect when the user presses a button or trigger on their VR controller to initiate object grab.

Once the user initiates object grab, the script can use the physics engine to apply a force to the object to simulate theuser's grip. The script can also apply constraints to the object to limit its movement and rotation, ensuring that it behaves as expected when the user moves it around.

To make object grab more intuitive and realistic, developers can also add haptic feedback to the VR controller, which provides users with tactile feedback when they interact with virtual objects. This feedback can be used to simulate the sensation of grabbing and holding objects, making the VR experience more immersive and engaging.

Overall, object grab is an important feature in VR as it allows users to interact with virtual objects in a way that feels natural and intuitive. By using physics and scripting in Unity, developers can create VR applications that provide a more immersive and realistic experience for users.

6.2 Why to Make an Object Grabbable?

Making an object grabbable in a VR application can provide several benefits to the user experience. The reasons why object grab may be important in a VR application are as follows:

1. **Immersion**: By allowing users to interact with objects in a realistic way, object grab can increase the sense of immersion in a VR environment. When users can pick up and move objects using their hands or controllers, it can create a more engaging and interactive experience.
2. **Natural Interaction**: Object grab can provide a more natural and intuitive way for users to interact with virtual objects. Instead of using a traditional input device, such as a mouse or keyboard, users can manipulate objects in the same way they would in the real world.
3. **Problem Solving**: In many VR applications, the ability to grab objects can be important for problem-solving and completing tasks. For example, a puzzle game may require the user to move objects around in order to solve a puzzle, or a virtual training application may require users to manipulate objects to learn a new skill.
4. **Accessibility**: Object grab can also make VR applications more accessible to users with physical disabilities or limitations. By allowing users to manipulate objects using their hands or controllers, VR applications can be more inclusive and provide a more equitable experience for all users.

6.3 Benefits of Making VR Hand Interactable in a VR Application

Making VR hands interactable in a VR application can provide several benefits to the user experience. The reasons why hand interactivity may be important in a VR application are as follows:

**Enhanced Gameplay**: Hand interactivity can also enhance the gameplay of VR applications by enabling new gameplay mechanics and interactions. For example, in a game, users may require to use their hands to grab objects, shoot a gun, or perform other actions. By making hands interactable, developers can create new and engaging interactions that enhance the overall gameplay experience.

6.4 LocalAvatar Prefab Oculus Interaction SDK

The LocalAvatar prefab in the Oculus Integration SDK is a pre-made GameObject that represents the user’s virtual self in the VR scene. It includes a set of standard components and scripts that are designed to work with the Oculus touch controllers and provide a basic set of interactions, such as hand presence and basic grabbing functionality.

The LocalAvatar prefab is used to represent the player’s own virtual body and hands within the VR environment and allows for natural and intuitive interactions with objects in the scene.

The prefab includes a set of components such as the OVRPlayerController, which provides basic movement controls and collision detection, and the OVRHandPrefab, which represents the player's hands and includes scripts for hand tracking, haptic feedback, and basic grabbing functionality.

In addition to the standard components and scripts, the LocalAvatar prefab can be customized and extended with additional scripts and components to add more complex interactions and functionality, such as advanced grabbing and throwing mechanics, object manipulation, and more.

6.5 The GrabAvatarLeft and GrabAvatarRight Prefabs

The **GrabAvatarLeft and GrabAvatarRight** prefabs in the Oculus Integration SDK for Unity are pre-made GameObjects that represent the user’s hands in the VR scene and provide advancedgrabbing functionality.

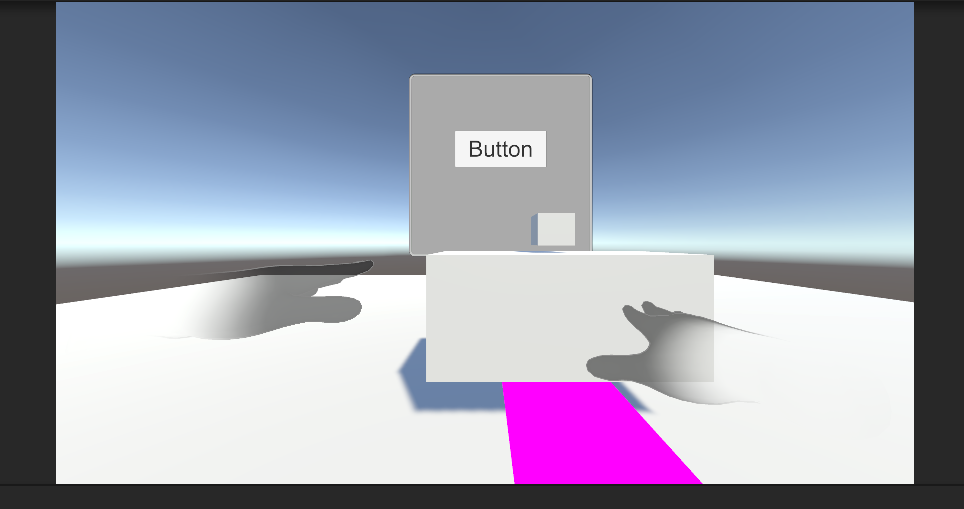


Figure 6.2 : GrabAvatarLeft and GrabAvatarRight

These prefabs are an extension of the LocalAvatar prefab and include additional scripts and components that enable more advanced grabbing and object manipulation.

They are specifically designed to work with the Oculus touch controllers and provide a range of features, including:

* **Grabbing and releasing** objects with different grips and force thresholds.
* **Haptic feedback** when grabbing and releasing objects.
* **Throwing** objects with realistic physics and velocity.
* **Two-handed interactions**, allowing for the manipulation of larger objects or multiple objects at once.

TheGrabAvatarLeft and GrabAvatarRightprefabs work in conjunction with the OVRGrabbable script, which is added to any object with which interaction is required in the scene. This script allows the object to be grabbed and released by the player’s hands using the advanced grabbing functionality provided by the GrabAvatar prefabs.

Overall, the GrabAvatarLeft and GrabAvatarRight prefabs provide a powerful set of tools for developers to create immersive and intuitive interactions in VR scenes using the Oculus Integration SDK.

The OVRGrabbable prefab in the Oculus Integration SDK for Unity is a pre-made game object that can be used to make any object in the VR scene grabbableby the player’s hands.

6.6 OVRGrabbable Script

When the OVRGrabbable script is added to an object in the scene, it allows the player to interact with that object by grabbing it with their hands using the Oculus touch controllers. The script provides a range of features that enable realistic and intuitive grabbing and object manipulation, including:

* The ability to specify different grips and force thresholds for grabbing and releasing the object.
* Haptic feedback when the object is grabbed or released.
* The ability to specify whether the object can be grabbed with one or two hands.
* Support for object scaling and rotation when it is grabbed.

The OVRGrabbable script works in conjunction with the GrabAvatarLeft and GrabAvatarRight prefabs, which represent the player’s hands in the VR scene and provide the advanced grabbing functionality. When the player’s hand collides with an object that has the OVRGrabbable script attached, the GrabAvatar prefabs detect the collision and enable the player to grab and manipulate the object.

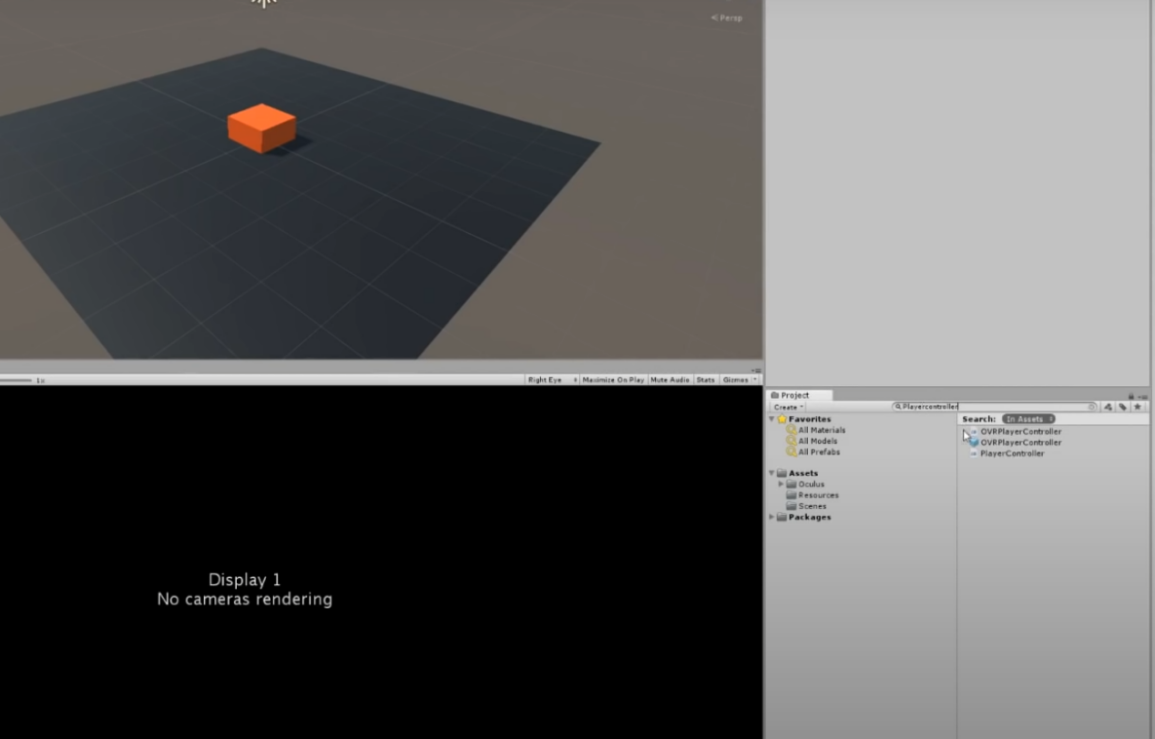
Overall, the OVRGrabbable prefab provides a powerful tool for developers to create interactive and immersive VR experiences using the Oculus Integration SDK for Unity.

6.7 Steps to Make an Object Grabbable in VR Scene

Till now, a User Interface (UI) was added in the scene along with hands as controllers. In this scene, let us add a Cube object on the table.

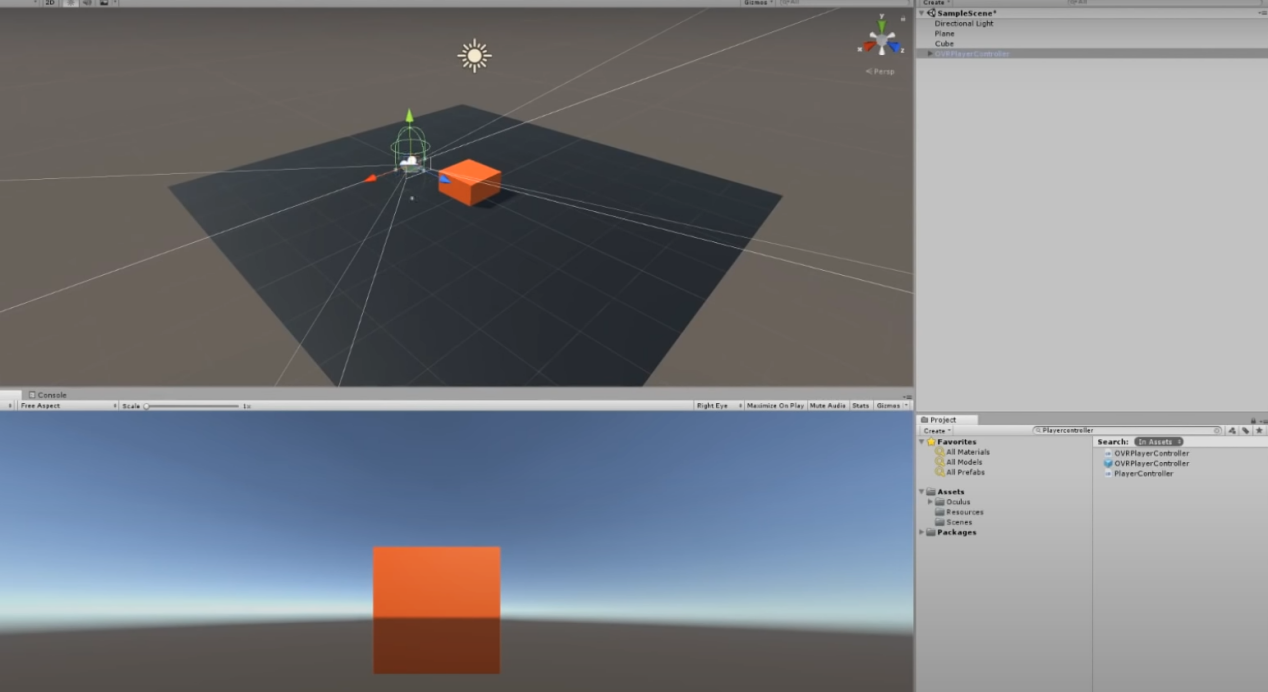
Following are the steps to make object Grabble in VR scene:

**Step 1**:In the Project panel, search for **OVRPlayerController** as shown in Figure 6.2.



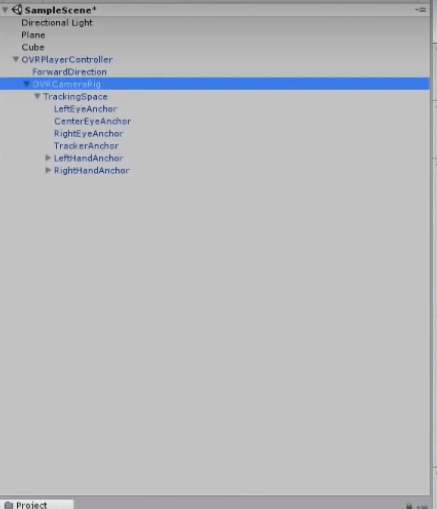
**Figure 6.3: Search for OVRPlayerController**

**Step 2**:Drag **OVRPlayerController** in the Scene as shown in Figure 6.3.

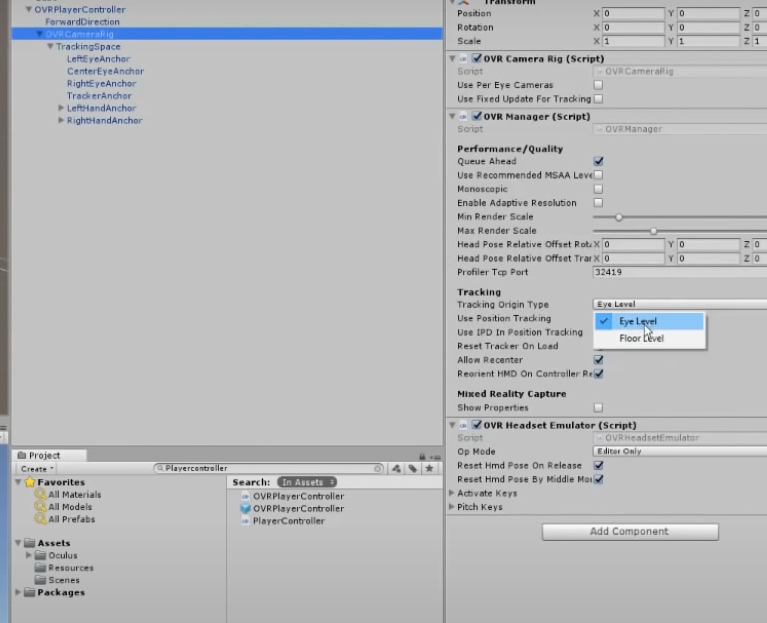


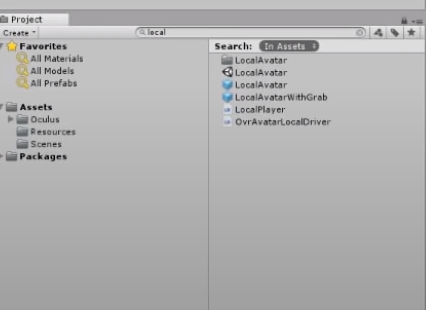
**Figure 6.4: Drag OVRPlayerCon**

**Step 3**: In the Hierarchy Panel, search **OVRCameraRig** under OVRPlayerRig as shown in Figure 6.4.

**Figure 6.5: OVRCameraRig**

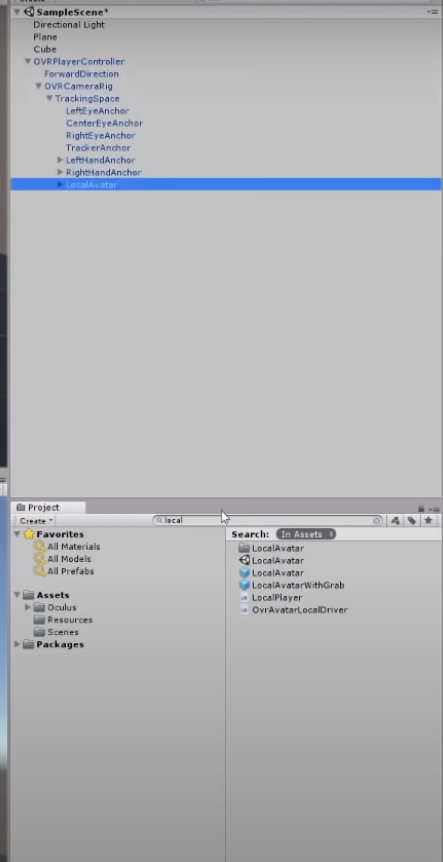
**Step 4**: In OVRCameraRig, set thevalue of **Tracking Origin Type** fieldto **Floor Level** as shown in Figure 6.5.

**Figure 6.6: Changing Position Tracking to Floor Type**

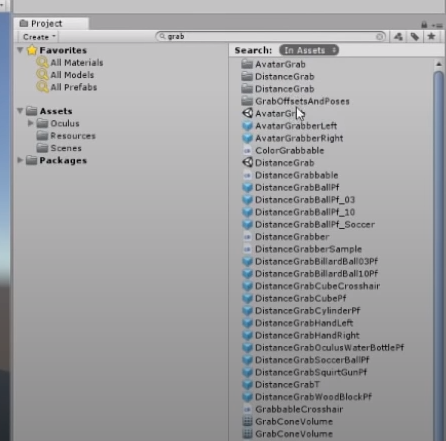
**Step 5**: Search **LocalAvatar** in the Search Panel as shown in Figure 6.6.

**Figure 6.7: Search for LocalAvatar**

**Step 6**: Drag LocalAvatar to CameraRig Component. **LocalAvatar** is dragged as shown in Figure 6.7.

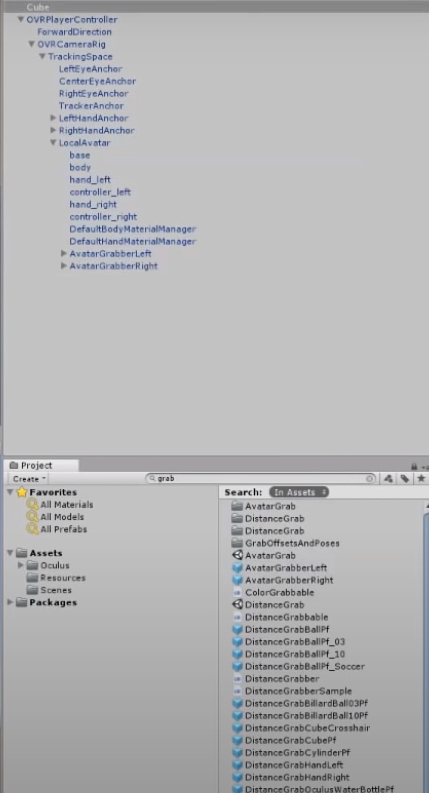
**Figure 6.8: Drag the LocalAvatar to CameraRig Component**

**Step 7**:Search **GrabAvatarLeft and GrabAvatarRight** in the Search Panel as shown in Figure 6.8.

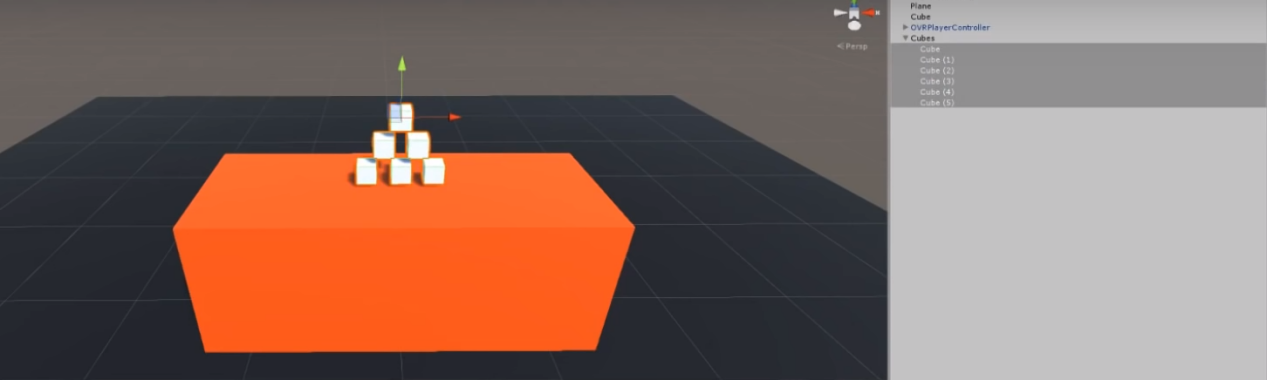


**Figure 6.9: Search GrabAvatarLeft and GrabAvatarRight Prefab**

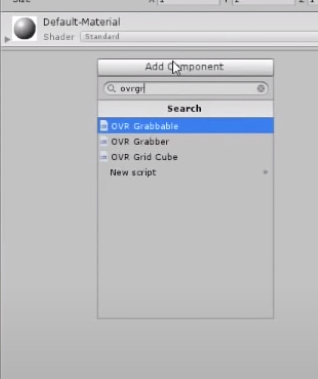
**Step 8**: Drag **AvatarGrabberLeft** and **AvatarGrabberRight** to Hierarchy Panel, that is under **OVRCameraRig → LocalAvatar** as shown in Figure 6.9.

**Figure 6.10: Drag AvatarGrabberLeft and AvatarGrabberRight Under LocalAvatar**

**Step 9**:Add Cubes in the scene. Right-click the Hierarchy Panel, select **3D Object → Cube** as shown in Figure 6.10.

**Figure 6.11: Adding Cubes to the Scene**

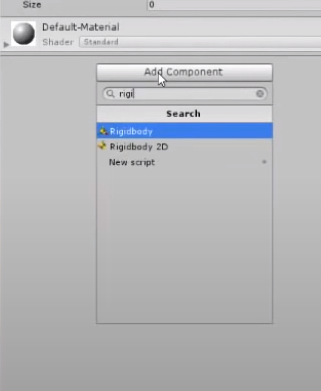
**Step 10**:Select Cube and go to **Edit Panel → Add Component** and search the script named **OVRGrabbable.cs** as shown in Figure 6.11.

****

**Figure 6.12: OVRGrabble Script**

**Step 11**: Select the Cube, go to **Edit Panel → Add Component** and search the Rigidbody component.

**RigidBody** provides the physics capability to the GameObject as shown in Figure 6.12.

**Figure 6.13: RigidBody Added to Object**

**Step 12**: The scene is ready to test now, as shown in Figure 6.13.

**Figure 6.14: Final Output**

Now, the object(Cube) is grabble in the VR Scene.

6.8 Summary

* LocalAvatar is a pre-made game object in the Oculus Integration SDK for Unity that represents the player’s avatar in the VR scene, providing features such as hand tracking, microphone input, and support for multiplayer interactions.
* GrabAvatarLeft and GrabAvatarRight are pre-build game objects in the Oculus Integration SDK for Unity that provide advanced grabbing functionality for the player’s hands in the VR scene, working with the OVRGrabbable script.
* The Object Grab component in Unity enables players to grab and manipulate objects in VR Scenes and can be customized to define how objects are interacted with.
* OVRGrabbable is a prefab that enables players to grab and manipulate objects in the VR scene with customizable grips and haptic feedback, working in conjunction with the GrabAvatar prefabs in the Oculus Integration SDK for Unity.
* LocalAvatar is a game object in Oculus Integration SDK for Unity that represents the player's avatar in VR, providing hand tracking, microphone input.

6.9 Check Your Progress



1. Which of the following represents the player’s avatar in the VR Scene providing features such as hand-tracking, microphone and so on?

1. Local Avatar
2. Avatar Prefab
3. Object Grabbable
4. Local Action
5. Which component in Unity enables players to grab and manipulate objects in VR Scenes?
6. Object Grab Component
7. Object Grabbable
8. Object Script
9. GrabAvatarLeft and GrabAvatarRight
10. Which component in Unity allows GameObject to be affected by physics?
11. RigidBody
12. Object Grabbable
13. GrabAvatar
14. GrabObject
15. Where can the LocalAvatar component be added?
16. Hierarchy → OVRCameraRig
17. Hierarchy OVRPlayerController
18. Hierarchy OVRPlayer
19. Hierarchy → CameraRig
20. Why is it important to add an object-grab feature to an object?
21. Natural Interaction
22. Immersion
23. Accessibility
24. All of these

6.10 Answers

|  |  |
| --- | --- |
| 1 | a |
| 2 | a |
| 3 | a |
| 4 | a |
| 5 | d |

6.11 Try It Yourself

* Build working application using Local Avatar and LocalAvatarLeft and LocalAvatarRight and test it.
* Make an object grabbable in VR Scene.