Installation Requirements:

Unity: https://store.unity.com/download?ref=personal
Steam: https://store.steampowered.com/about/
SteamVR: https://store.steampowered.com/app/250820/SteamVR/
Windows MR for SteamVR:
https://store.steampowered.com/app/719950/Windows_Mixed_Reality_for_SteamVR/
SteamVR Plugin - Unity Asset Store:
https://assetstore.unity.com/packages/tools/integration/steamvr-plugin-32647

Steps:

Connecting Controllers (If Necessary):

1. Open Bluetooth settings from Windows Search bar
2. Select “Add Bluetooth or other device” and choose Bluetooth device
3. Power on each controller by holding down the Windows button
4. Open the battery compartment on each controller and hold down the button at the base of the controller until the lights start to flash
5. Wait for the controllers to appear on the computers device window and select each from the menu to pair

Creating a VR Scene:

1. Create a new Unity project
    a. Use 3D template
    b. Use whatever project name you like
2. Open Asset Store tab in editor window and import SteamVR plugin (use search bar and download if not already done)
3. Generate the actions.json file for SteamVR interaction system
    a. Select the window tab at the top of the editor
    b. Select SteamVR input
    c. You may be prompted to generate an actions.json file, select yes
    d. When the SteamVR input window opens scroll to the bottom and select “Save and Generate”
    e. Once the window is done compiling you can close it
4. Place a plane in the scene
5. Place the player prefab on the plane
    a. SteamVR->InteractionSystem->Core->Prefabs->Player

Basic Interactions:

6. Place a cube in front of the player prefab
    a. This is basically a table
    b. Resize and move to be approximately in front of the player prefab
7. Place a sphere (or any other primitive object) above the cube

    a. Resize to be about 1 / 8 the size of the cube
8. Add the “Interactable” script to the object
    a. SteamVR->InteractionSystem->Core->Scripts->Interactable

This might be a good time to put on the headset for a second?

9. Create a new script (name it whatever you like) and open it
10. Add “namespace Valve.VR.InteractionSystem” to encompass the class
11. Create attachment flags
    a. [EnumFlags]
    b. public Hand.AttachmentFlags attachmentFlags = Hand.AttachmentFlags.ParentToHand | Hand.AttachmentFlags.DetachFromOtherHand;
12. Add hover function
    a. private void HandHoverUpdate(Hand hand)
13. Check for grab within hover function
    a. GrabTypes startingGrabType = hand.GetGrabStarting();
    b. if (startingGrabType != GrabTypes.None)
    c. Checks if user is starting to perform some type of grab (different buttons)
14. Attach object to hand if grab starting
    a. hand.AttachObject(gameObject, startingGrabType, attachmentFlags);
15. Check if grab is ending when attached to hand
    a. private void HandAttachedUpdate(Hand hand)
    b. if (hand.IsGrabEnding(gameObject))
    c. hand.DetachObject(gameObject);
    d. By default restores original parent object but can be disabled

Add visual feedback

16. Add public variable for additional material
    a. public Material grabMat;
17. Store original material
    a. private Material originalMat;
    b. void Start()
    c. originalMat = GetComponent<Renderer>().material;
18. Change to new material when attached to hand
    a. private void OnAttachedToHand(Hand hand)
    b. GetComponent<Renderer>().material = grabMat;
19. Change back to original material when detached
    a. private void OnDetachedFromHand(Hand hand)
    b. GetComponent<Renderer>().material = originalMat;
20. Add script to interactable object
21. Assign new material for grab indicator
    a. Create new material and change albedo to something obvious
    b. Drag into empty material spot on script

Give everyone a moment to try on headset again and interact with ball

Look through throwable script for more detailed example

Adding teleportation

1. Add Teleporting prefab to scene
    a. SteamVR->InteractionSystem->Teleport->Prefabs->Teleporting
    b. Handles bulk of teleportation mechanics with lots of modifiable settings
2. Create new plane slightly above ground
3. Convert new plane into teleportation area
    a. Add TeleportArea script to ground
    b. SteamVR->InteractionSystem->Teleport->Scripts->TeleportArea

Try the scene one more time with teleportation mechanics to move

Open exploration time, check out the InteractionSystem sample to see many of the possibilities with SteamVR

SteamVR->InteractionSystem->Samples->Interactions_Examples

Implement 2 VR based interactions or mechanics either in the InteractionSystem sample scene or in a new Unity scene. Note: Don't worry about the visuals of the objects, feel free to use basic primitives for actual objects but do consider including some amount of feedback if necessary.

Ideas:

- Make a basic tee ball setup (hit a physics based ball with some form of bat)
- Create a punching bag that returns to starting position after being hit
- Make a lantern that can be picked up and then changed somehow (such as changing the color of one part) through subsequent controller actions while holding it
- Create a potato cannon (either an object that can be picked up that subsequently launches other objects or a button that causes a cannon to fire)
- Add another form of player movement besides teleporting (such as moving forward in the direction the player is facing when using a certain controller action)
- Create a simple remote controller to manipulate an object in the environment (do not use the existing remote controller scripts from the sample scene but look through them for guidance)
- Make a single axis constrained mechanism (can only be dragged around a 2D plane) such as in a sliding puzzle
- Create an object that can be held with both hands simultaneously (such as being in the average of the positions of the two hands)
- Create a boomerang type of object that returns to the point where thrown from after some amount of time (doesn't have to be physically accurate to a real boomerang)

- Make a simple 3D paintbrush to leave some kind of 3D trail when a certain controller action is used (consider only intermittently generating the trail in order to reduce computational load)