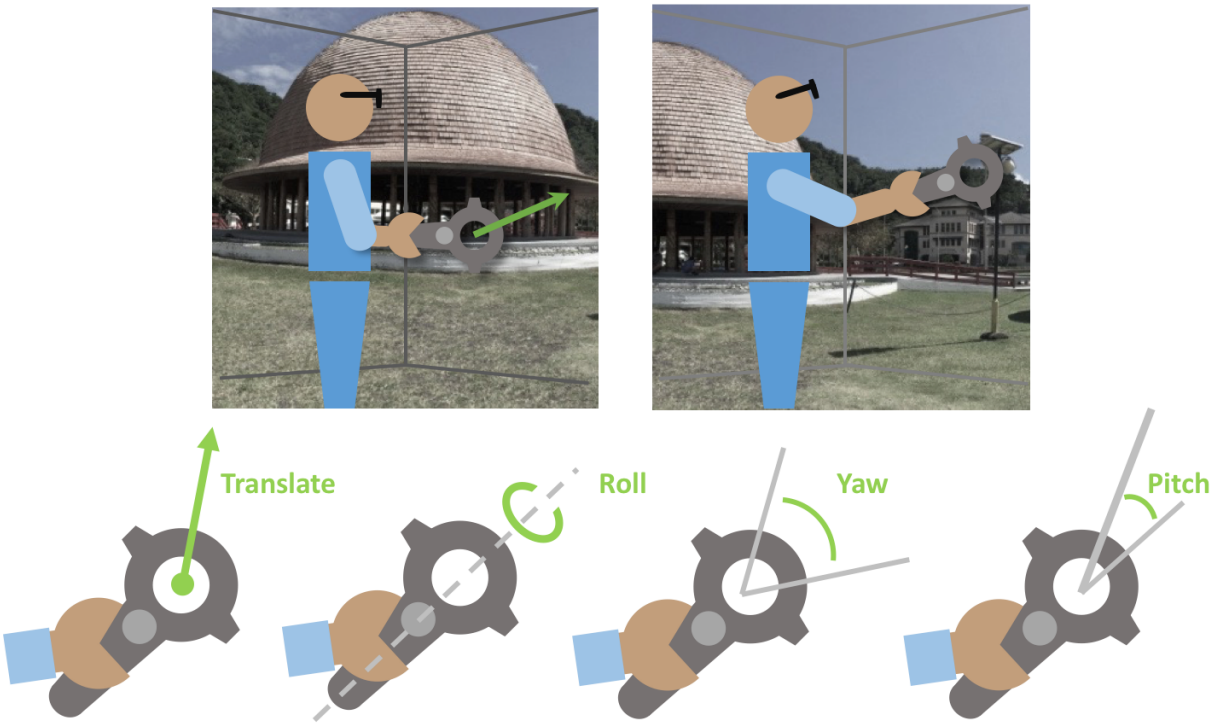
The Omni-Navigator

Version 2024/10/14

In the Omni-navigator VR travel scheme[1] the user imagines that he/she is holding a paper airplane or bird in one's hand, and how he/she positions and orients the bird determines how the user will travel through the space. To use this travel scheme, the user begins by holding the VR interaction controller (wand) at an initial origin/starting position and orientation. Then by pressing a designated button and moving the wand from that origin, the distance, direction and orientation from the initial origin determines the direction, rate of translation (movement in location) and rotation about all 3 axes (see picture below). For example, to move forward, the user presses a button and moves the wand forward. The distance from the starting position of the wand determines the speed of movement. To move upward, the user presses the button and moves upward. The distance from the starting position again determines the speed of movement. To pitch, yaw, or roll, the user presses the button and tilts the wand in one or more of the 3 axes. The greater the angle of tilt the more rapid the rotation.



The Omni-navigator control scheme.

This travel scheme has the advantage that with only the use of one button and the position and orientation of the wand, the user can achieve full 6-degree of freedom movement in addition to rate of movement. Once mastered users typically find the degree of control unparalleled by any

[1] Excerpted from Chapter 20 of the Book: Virtual Reality Development Gems

other VR travel scheme. Of course, if desired, some degrees of movement can also be constrained. For example, if one wanted to adapt this scheme to a first-person-shoot-style of movement along a horizontal plane, one simply has to exclude vertical (Y axis) movement, pitching and rolling.

Using the Omni-navigator

In most VR environments there is the notion of a “container” in which the user’s head and interaction controllers are held. So, when travel is performed it is this container that is moved through the larger virtual space, and then from within the container the user may walk around the limited physical space constrained by the VR hardware – it’s like riding in a rectangular, invisible platform or vessel. This container is sometimes called a “Camera Rig”, “VR Rig”, “VR Hub” etc.. In Unity, the container would be a gameobject Inside the camera rig there are usually game objects that refer to the user’s head, as well as his/her hands.

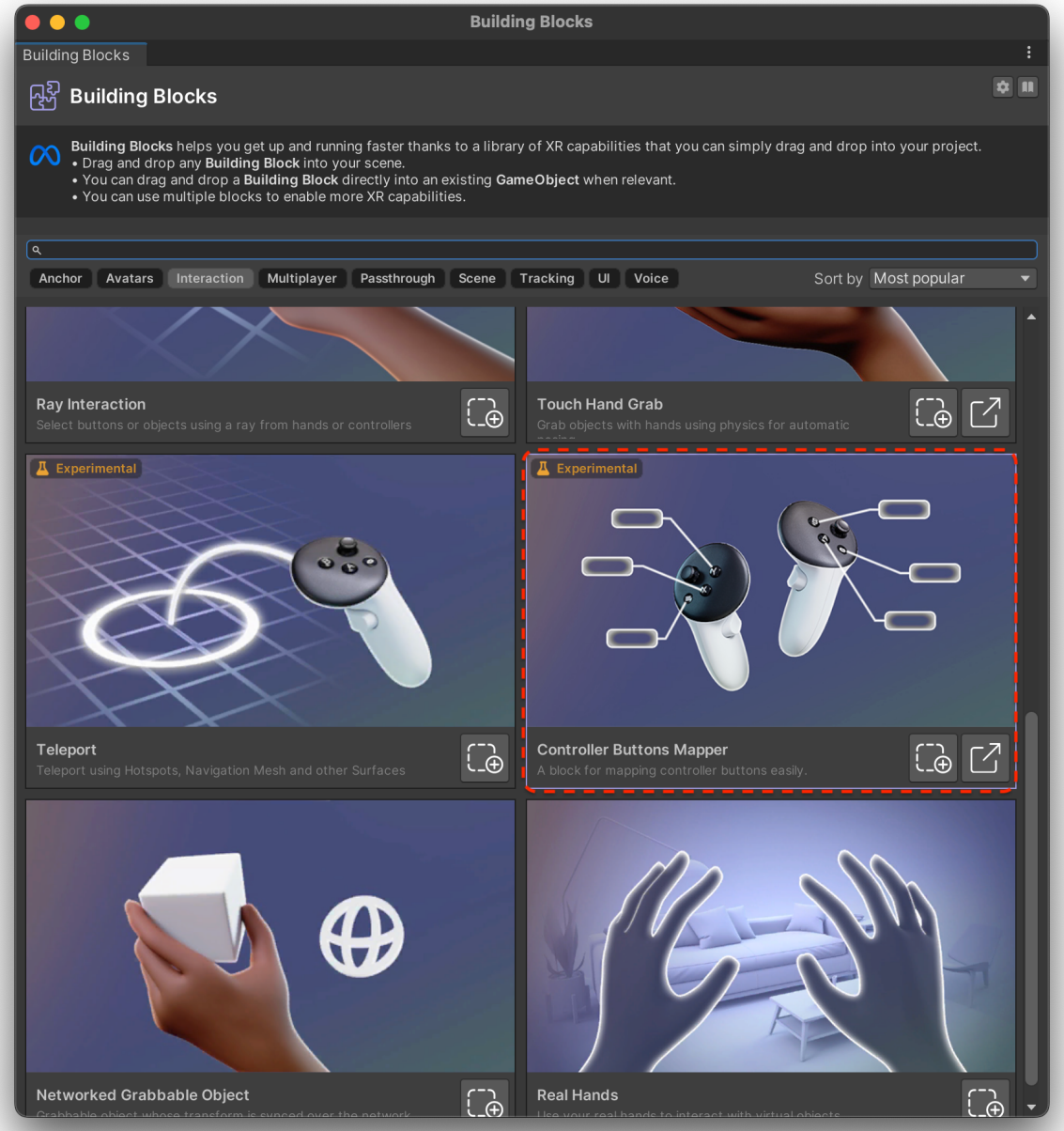
The general steps for using the Omni-navigator are below. If you wish to apply these steps to Meta Bulding Blocks go to Building Blocks Instructions in the next section.

1. Create an empty game object and add the Omni-navigator script (component) to the game object.
2. In the Omni-navigator script component:
    a. assign the Camera Rig to the Main Hub parameter.
    b. assign the game object in the Camera Rig that refers to the user’s head to the Head parameter.
    c. assign the game object in the Camera Rig that refers to the user’s hand (either left or right hand) to the Wand parameter.
    d. adjust other parameters as necessary.
    e. If you wish to enable your movement to be blocked by stationary walls, or if you wish to be able to walk up ramps, set Enable Collision to true, and add a Rigid body and a Capsule Collider to the Camera Rig. Then for the rigid body set Freeze Rotation x y and z to all true. For the capsule collider set Center to (0, 0.89, 0;) set radius to 0.05; set Height to 1.75; set Direction to y-axis. These set the collider to represent your body trying to physically move through space. 1.75 is roughly 5'10".

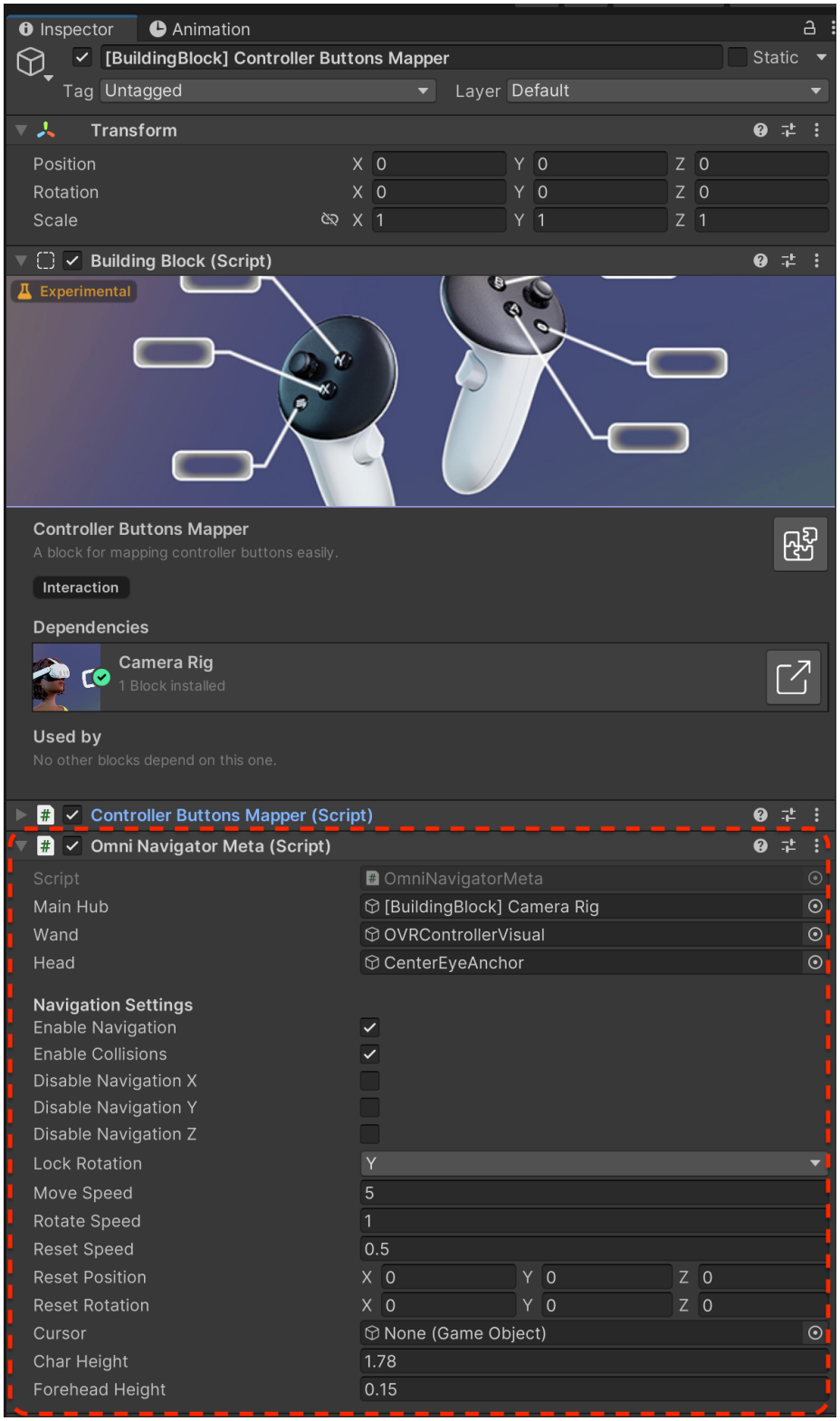
To trigger the navigation, you need to write code to call the script’s NavGo function when a button is pressed. Then when your button is released, call NavStop. You need to also do the same to designate the Reset Key that calls Reset Go when the user clicks on your Reset Key,

Meta Building Blocks Instructions

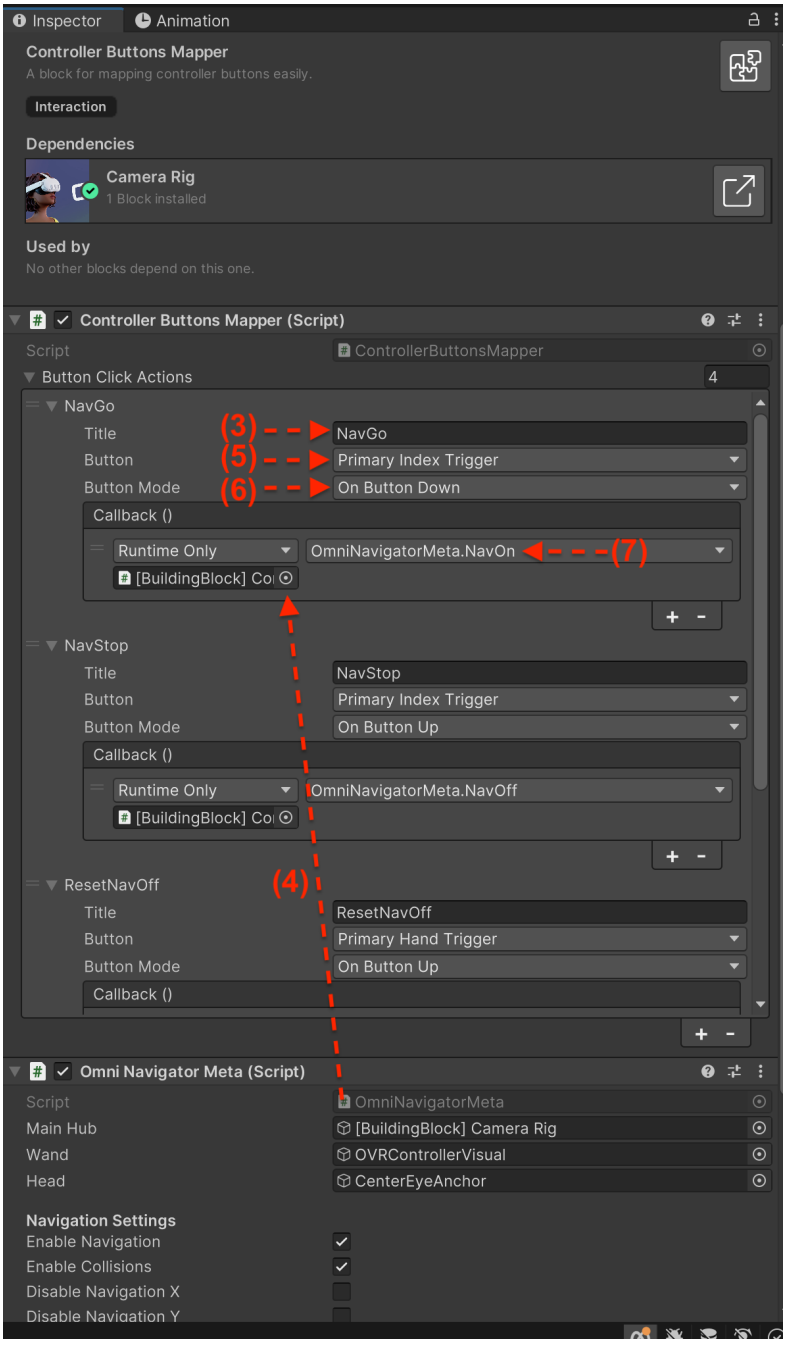
1. Instead of instructions above, add a Controller Buttons Mapper to your scene from the Meta Building Blocks panel.



2. Add the Omni-Navigator script to the Controller Buttons Mapper.



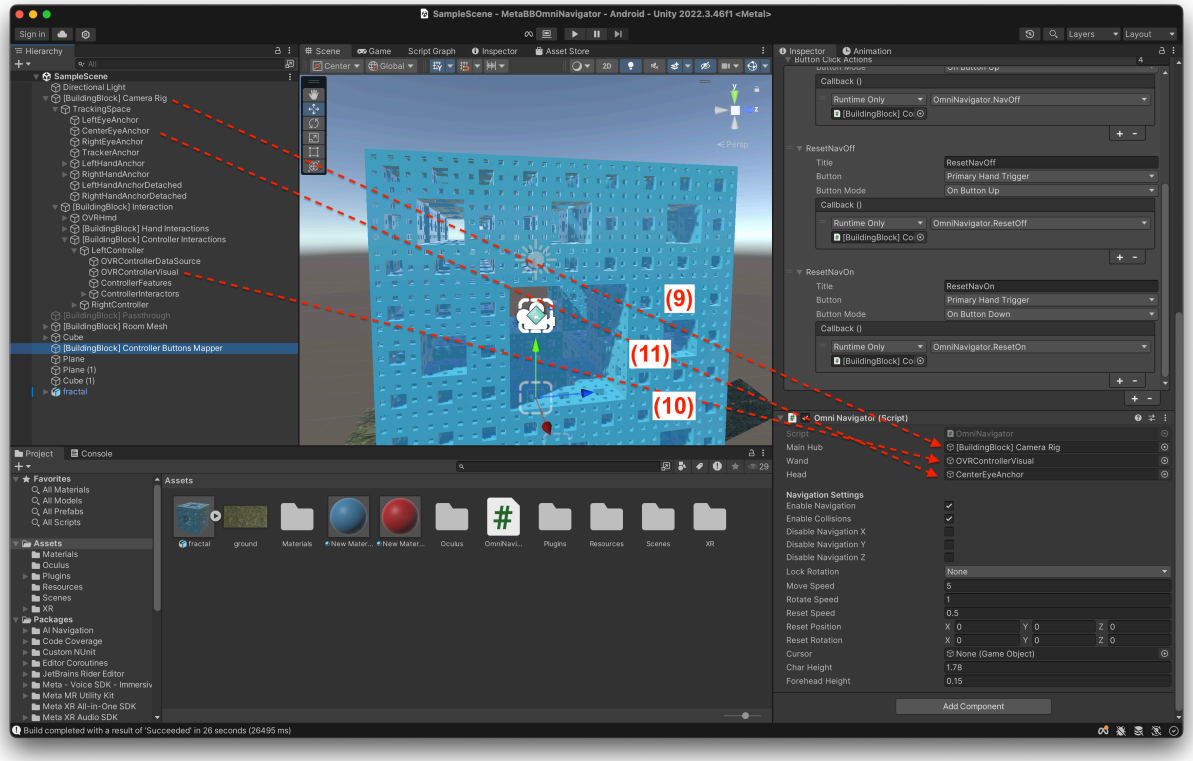
3. In the Controller Buttons Mapper component, add 4 button click actions to the controller buttons mapper, named: navGo, navStop, resetGo, resetStop.



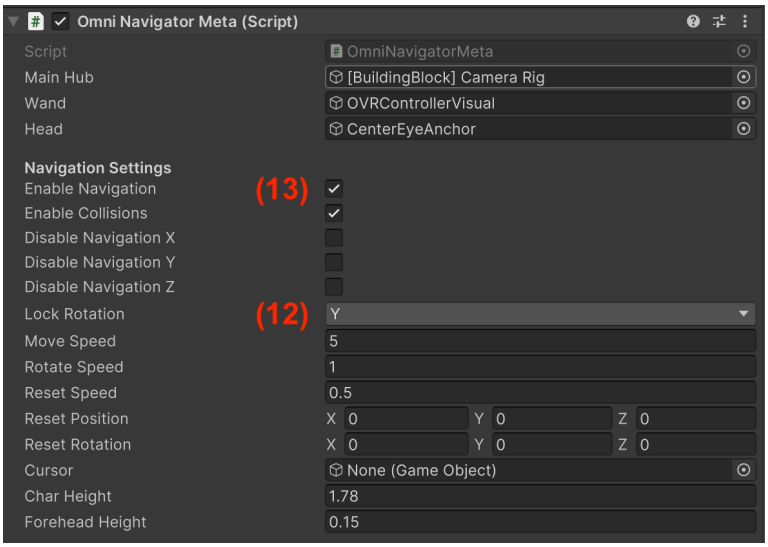
4. For navGo click and drag the OmniNavigator component to the navGo callback.
5. Select the input button to trigger the navigation (e.g. Primary Index Trigger).
6. Set button mode to On Button Down.
7. Select the function to call, in this case: OmniNavigator.NavOn.
8. Do this for all the other actions (navStop, resetGo, resetStop).

- for resetGo: input button would be Primary Hand Trigger, button mode is On Button Down, function is resetGo (this makes it so that if you hold the grip button, the navigator will smoothly return you to your home starting position and orientation.
- for navStop: input button would be Primary Index Trigger, button mode is On Button Up, function is navStop (signals navigation to stop when you release the trigger button)
- for resetStop: input button would be Primary Hand Trigger, button mode is On Button Up, function is resetStop (signals return-home behavior to stop when you release the grip button).

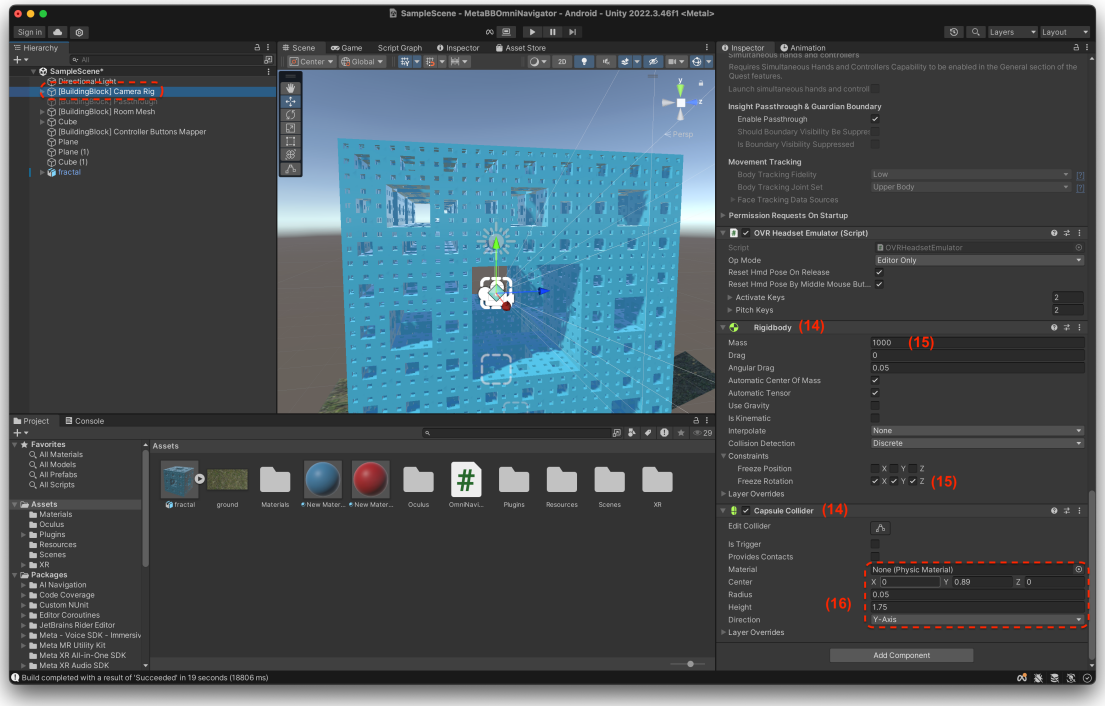
9. In the OmniNavigator component, click and drag [Camera Rig to MainHub.



10. Click and drag Camera Rig> Interaction>ControllerInteractions>LeftController> OVRControllerVisual to Wand. (this assigns your left controller as the navigation wand)
11. Click and drag CameraRig>TrackingSpace>CenterEyeAnchor to Head (this lets the navigator detect your head position, so that you can duck under objects).
12. Lock rotation in OmniNavigator to None for fully unconstrained rotation – recommend you leave it as None (later you can try the classical 1st person shooter type of constrained rotation, by setting this to Y)
13. If you wish to enable your movement to be blocked by stationary walls, or if you wish to be able to walk up ramps, set Enable Collision to true



14. In the Camera Rig, add a Rigid body and a Capsule collider
15. For the rigid body set Mass to 1000, Freeze Rotation x y and z to all true, and set Use Gravity to false (later when you try the classic 1st-person-shooter set Use Gravity to true.)
16. For the capsule collider set Center to (0, 0.89, 0); set radius to 0.05; set Height to 1.75; set Direction to y-axis These set the collider to represent your body trying to physically move through space. 1.75 is roughly 5'10".



After you've gotten this to work, you may wish to try the classic 1st-person shooter style navigation where you are walking along the ground and climbing up ramps. To do this, make the following adjustments:

- In the Camera Rig's Rigid Body component, set Use Gravity to true.
- In the Omni-Navigator component, Set Lock Rotation to Y.

Explanation of the other Omni-Navigator Settings

- **Disable Navigation** (in X, Y, Z): Disable movement along the selected axes.
- **Lock Rotation**: Lock rotation around an axis (either X, Y, Z or none). Lock rotation so that it only occurs along one of the axes.
- **Move Speed**: Movement speed in (grid-units/s).
- **Rotate Speed**: Rotation speed in (degrees/s).
- **Reset Position** (Vector3): Position considered the home position for the environment.
- **Reset Rotation** (Vector3): Specify the home rotation angle.
- **Cursor** (GameObject): Specify a 3D model asset to make visible when in navigation mode (in this example the bird is assigned as the cursor).
- **Char Height**: Anticipated height of the user (set to 1.78m by default).
- **Forehead Height**: Height of the user's forehead (set to 0.15m by default).