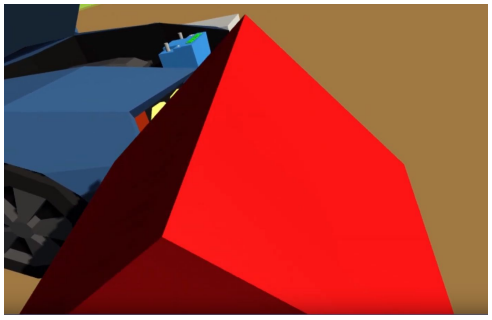
4.2 Add Interactable Cube

Activities:

Step 1: Find and Add the Prefab

Step 2: Apply the Red Material and Test



Length:	15 minutes
Overview:	Add an existing prefab object that you can pick up in VR
Project Outcome:	An interactable cube that has the red material applied to it.
Learning Objectives:	The students will be able to: - Navigate through nested objects to find the prefab - Apply a material to an object - Test the ability to pick up the cube
Prerequisite:	Previous tutorial steps (Sections 1-3)
ISTE Student Standards	6c “communicate complex ideas using a variety of digital objects such as simulations””

Video	https://vimeo.com/349083477

OR

Step 1: Find and Add the Prefab

1. Navigate inside the Assets folder > VRTK > Prefabs > Interactions > Interactables folder
2. Drag the "Interactable.Primary_Grab.Secondary_Swap"

- **Explain** “Primary Grab” means the first controller that attempts to pick it up will pick it up. “Secondary Swap” means once it is picked up, the second controller

prefab object into the scene. It looks like a plain cube.

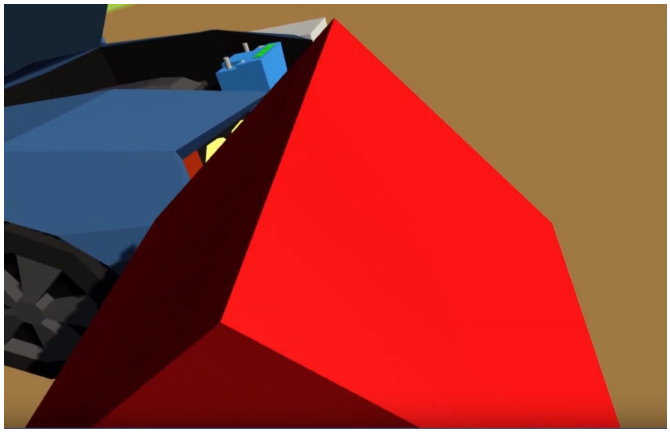
3. Scale the cube down to 0.5, 0.5, 0.5

can grab it away from the first controller, swapping it.

Step 2: Apply the Red Material and Test

1. From the materials folder, drag the Red material onto the cube, turning it Red.
2. Play test to make sure you can pick up the new Red cube.

Warning: When testing with a VR headset, your “grab” may be the side “squeeze grip” button.



Explain A “Prefab” is an object that is pre-fabricated, or assembled ahead of time and can include multiple objects, children objects, and settings. Best used for elements that will be re-used throughout a project.

End of Lesson

Recap

Key concepts:

- Prefabs allow multiple objects, scripts, and settings to be saved in a reusable asset.

Key skills:

- Navigate through nested folders
- Apply a material

Key terms:

- **Prefab: (From [Unity documentation](#))** Unity's Prefab system allows you to create, configure, and store a GameObject complete with all its components, property values, and child GameObjects as a reusable Asset.