VRBasics

By Zac Zidik

Initial Set Up

1. Place an instance of the VRBasics game object from the VRBasics -> Prefabs folder into your scene
2. Add a layer to your project named Ignore Collisions
3. Under Edit -> Project Settings -> Physics use the Layer Collision Matrix to disable all collisions between the Ignore Collisions layer and all other layers
4. Select the instance of VRBasics in your scene and make sure the Ignore Collisions variable to set to the index of the Ignore Collisions layer

(Combine with SteamVR)

5. Download the SteamVR plug-in for Unity
6. Place an instance of the [CameraRig] game object from the SteamVR -> Prefabs folder into your scene
7. Add the VRBasics_Controller script to both the Controller (left) and Controller (right) child game objects of the [CamerRig] instance
8. From the VRBasics -> Prefabs -> Widgets folder, add a Toucher game object as a child of both Controller (left) and Controller (right)

Examples Set Up

1. Open one of the scenes from the VRBasics -> Examples -> Scenes folder
2. Follow the Initial Set Up instructions

(tankBelt Scene)

3. Add an instance of the tankBeltRigged game object from the VRBasic -> Examples -> Prefabs folder as a child object of the [CameraRig] game object in the scene

(gaze Scene)

3. Add the VRBasics_Gaze script to the Camera (eye) game object of the [Camera Rig] and set the Gaze Camera variable to the Camera (eye) in the Inspector
4. Add an instance of the Reticle game object from the VRBasics -> Examples -> Prefabs folder as a child object of the Camera (eye), and set the Reticle variable of your VRBasics_Gaze script to the Reticle in the Inspector

(speech Scene)

3. Move the StreamingAssets folder from the VRBasics folder into the root Assets folder (this is where the grammar file is stored)

WIDGETS

In the VRBasics -> Prefabs -> Widgets folder you will find four game objects. These objects can be used as a base for physics based interactions in VR worlds.

CONNECTOR

VRBasics_Connector

Connectors are game objects that have trigger colliders. Their purpose is to connect their parent object to another connector's parent object. For a connection to be made, the connectors touching must be of opposite type, one male and one female. They must also share a coupleID number.

Male connectors will present several options for how a connection can be broken.

1. Never – the connection, once made, cannot be broken
2. Force – if the proper amount of force is applied to the joint of the connection the connection will break
3. Grab – if either parent object in the connection is grabbed the connection will break

Examples: Plug and Outlet, Lid and Jar, Lego Blocks

How to use:

1. **Place a connector prefab widget as a child of each of the objects you would like to connect**
2. **Position and scale the trigger box collider of the connector**
3. **Make sure to make one connector male and one female and that they share a coupleID**
4. **Run your scene, when the connectors touch, a joint will be formed between the parent objects**

(See the riggedBlocks in the demoStation scene for examples of connectors)

LEVER

VRBasics_Lever and VRBasics_Hinge

Levers provide a stable mount for a child hinge to rotate around a singular axis. They also provide visual indication of how the lever will move in the Scene view by drawing a gizmo that displays the length of the lever as well as the direction and limits of rotation.

Examples: Doors, Knobs, Dials, Switches, Needle of a Gauge

How to use:

1. **Place a lever prefab widget in your scene**
2. **Position and rotate the lever**
3. **Adjust the lever length property to roughly the size of the lever**
4. **Set the rotational limits of the child hinge object**
5. **Adjust the spring and damper settings of the hinge**

(See the various levers in the cabnet of the demoStation in the demoStation scene for examples of levers)

RAIL

VRBasics_Rail and VRBasics_Slider

Rails provide a stable mount for a child slider to move back and forth along a single axis. They also provide visual indication of how the slider will move in the Scene view by drawing a gizmo that displays the length and direction of the rail as well as the position of the slider along the rail.

Examples: Sliding Toggles, Buttons, Drawers

How to use:

1. **Place a rail prefab widget in your scene**
2. **Position and rotate the rail**
3. **Adjust the rail length property to roughly the size of the slide**
4. **Adjust the position of the anchor of the rail**
5. **Adjust the starting position of the slider**
6. **Adjust the spring and damper settings of the slider**

(See the various rails in the cabnet of the demoStation in the demoStation scene for examples of rails)

TOUCHER

VRBasics_TouchAndPushManager and VRBasics_GrabManager

The Toucher object is used to allow spatially tracked controllers to interact with virtual objects. The Toucher object contains a trigger collider used to indicate when a controller collides with a VRBasics_Touchable object.

The Pusher, a child object of the Toucher, contains a collider that is slightly smaller than the trigger collider of the Toucher. Once an object is touched, if it is set to pushable, the Pusher collider will move the object.

The Toucher is also used for collisions with VRBasics_Grabbable objects which are an extension of the VRBasics_Touchable class.

Examples: Finger, Hand

How to use:

1. **Place a Toucher prefab widget as a child of a spatially track controller**
2. **Position Toucher and scale the colliders if desired**

(To make an object touchable add the VRBasics_Touchable script to your game object. To make an object grabbable attach the VRBasics_Grabbable script.)