About

XRKJ (XR Kinematic Joystick) is a 3D virtual joystick made to be used with Unity's XR Interaction Toolkit (Updated to 60000.0.25f1 [originally 2022.3.31f1]).

XRKJ Works in any orientation and maintains rotations appropriate to a joystick.

Any of the meshes can be replaced with models of your choice.

XRKJ can also be used as a simple knob.

The XRJK Controller (Script) uses a scriptable float object and/or a delegate to communicate to other scripts.

Known Issues

NOTE: Development testing only used Meta Quest 3.

- When moving the joystick around at maximum its allowable magnitude, there appears to be some visual stuttering. This was introduced after implementing newer editions of Unity XR Toolkit scripts (many of the original components used are now depreciated).

- The Knob feature seems to be affected by the Y-rotation of the player.

Quick & Safe Demo Setup.

Note: Except for movement and some other non-essential features, XRKJ should work with no additional set up when following the below steps.

1) Create a new project using VR or MR Core.
    a. Use MR Core if you want to use AR features (unless you already know how to set up an AR Session).
2) Import the XRKJ package.
3) Open the XRKJ Demo Scene.
    a. Assets > OHGAR > XRKJ Demo Scene
4) Add the Unity VR or MR prefab player to the XRKJ Demo Scene.
    a. VR Core: XR Origin Hands (XR Rig).
        i. Controllers & Hands.
    b. VR Core: XR Origin (XR Rig).
        i. Controllers only.
    c. MR Core: MRInteractionSetup
        i. AR already set up
        ii. Delete the Goal Manager and Object Spawner to clear console errors.
5) Plug your headset in and make sure it is recognized by Unity in Build Settings.
    a. The Meta Quest Link app needs to be running on your computer, while using Unity.

6) Click play in the Unity Editor and put your headset on.
    a. AR functionality won't work when testing in the editor on Meta Quest.
    b. You will get console when testing in the editor. This is normal and ok.
7) To test XR/MR, create a build using Unity's MR Core (MRInteractionSetup).
    a. Clear the Scene List, then add XRKJ Demo Scene to the scene list.
    b. Set build settings to the appropriate platform.

XRKJ Controller (Script)

By default, this script is located on the root game object of the XRKJ prefab.

This script is dependent on UnityEngine.XR.Interaction.Toolkit, by default it must interact with an XR Grab Interactable component.

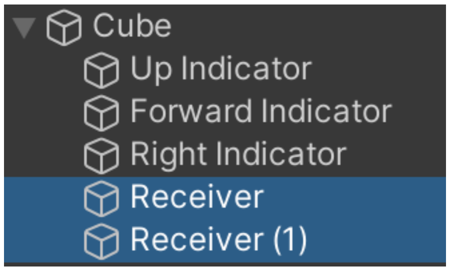
Automation

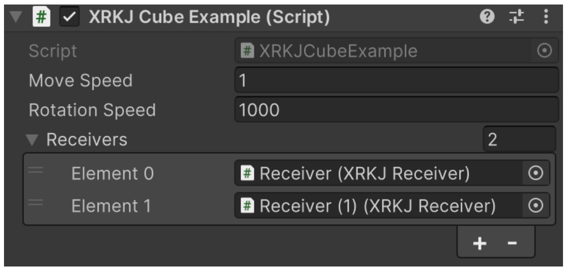
NOTE: Only rigidbody constraints and the Grab Interactable's track position and track rotation are affected by automation.

- Auto Settings
    - If true, rigid body and grab interactable settings will be changed OnStart via SetJoystick & SetKnob methods (Determined by Is Knob in Knob Parameters).
    - Set this to false if you want to use custom grab interactable settings.
- Activate On Start
    - Intended for debugging.
    - Not recommended for use on a final product.
    - If true, Update code will begin OnStart()
        - This is useful for controlling the joystick with a mouse instead of XR.
    - If false, the dev will need to set up a way to call Activate() and/or Deactivate() from XRKJController (Script).
        - See Grab Interactable for more guidance.

Output

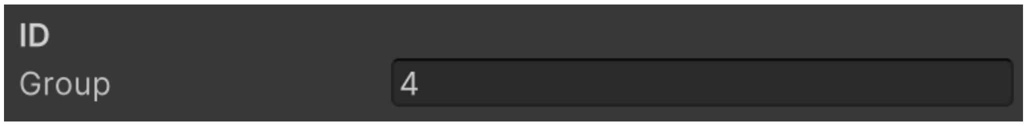
- Print Output
    - Will Debug.Log the group, output, and Dead Zone status of the joystick.
    - Useful for debugging and can be used in conjunction with the XRKJ Receiver (Script)'s Print Input Boolean (output and input should match).
- Group
    - By default, this is only applicable when using the XRKJ Receiver (Script). However, it can be useful in custom scripts as well.
    - Use this to pair joysticks to specific receivers.
    - Receivers can be used on a parent or child game object.
    - If you want to use more than one joystick to control a single object and want to use the XRKJ Receiver, you should use more than one receiver and set them to different groups (see below).


Cube
Up Indicator
Forward Indicator
Right Indicator
Receiver
Receiver (1)

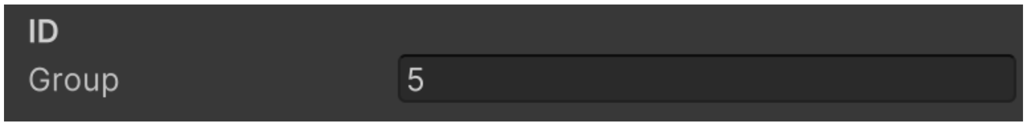

XRKJ Cube Example (Script)

Script	XRKJCubeExample
Move Speed	1
Rotation Speed	1000
Receivers	2
Element 0	Receiver (XRKJ Receiver)
Element 1	Receiver (1) (XRKJ Receiver)

Receiver


ID
Group 4

Receiver (1)


ID
Group 5

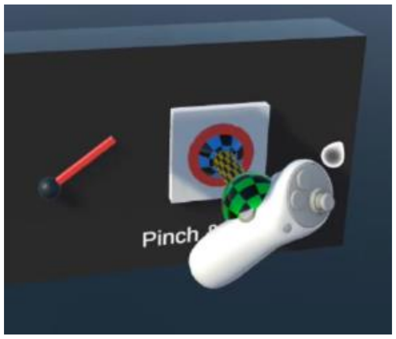
- Use Delegate
  - Set this to true if you want to use the built-in Output delegate (int group, Vector3 output).
- Output Scriptable Object
  - This field can be used on its own, or in conjunction with the Output delegate.
  - XRKJOutputSO (Vector 3) is the default scriptable object included.

Joystick Parameters

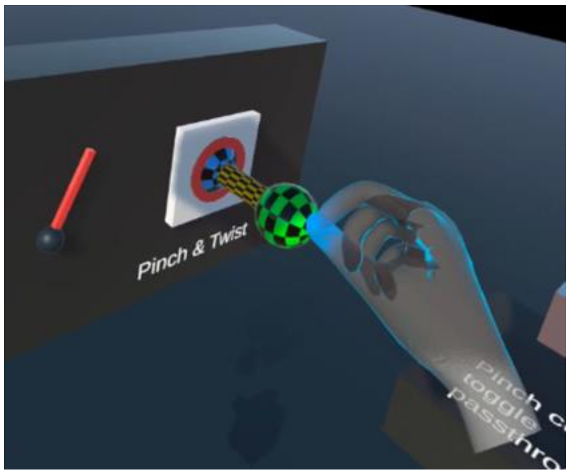
- Sensitivity
  - Use this to balance the joystick's rotation speed with anticipated player hand movement.
  - Does not work if Is Knob is true.
  - Locking the XR hand/controller to the joystick is not included and would need to be implemented by the developer.
- Max Pivot Radius
  - Determines how far the joystick can rotate.
  - Uses Vector3.ClampMagnitude()
- Use Dead Zone
  - If true, the Joystick's output will read 0,0,0 while the Dead Zone Radius.
- Dead Zone Radius
  - Only applicable if Use Dead Zone is true.
  - A magnitude from the center the joystick needs to reach before an output other than 0,0,0 is given.

Knob Parameters

When using controllers in knob mode, twist the knob from the center of the controller with the controller perpendicular to the joystick's y axis (most accurate). This could be changed, but no method for doing so is provided in this package.



When using hands in knob mode, by default, pinch and twist with wrist (less accurate than controllers).



- Is Knob
    - If Auto Settings is true
        - All rigid body constraints will be frozen except Y Rotation.
        - The grab interactable component will switch from track position to track rotation.
    - If Auto Settings is false
        - By default, knob functionality will not work.
        - The Developer will need to manually change the rigid body and grab interactable settings to accommodate the expected knob behavior.

References

See Grab Interactable and Pivot.

Unity Events

- On Activate()
    - If Activate On Start is true.
        - Invoked OnStart().
    - If Activate On Start is false.
        - Invoked when the grab interactable recognizes a grab.

    - See Grab Interactable for more information.
- On Deactivate()
    - If using XR
        - Invoked when the grab interactable recognizes a release.
    - If debugging with mouse
        - Has no default way of being Invoked.

Grab Interactable

The Grab Interactable is the first child game object of XRKJ Joystick Prefab. It contains a rigidbody, XR Grab Interactable (Unity XR Interaction Toolkit), and an XR General Grab Transformer (Unity XR Interaction Toolkit).

This packages default settings are what I found to work best; however, I am certain they can be improved upon. Most settings in these components can be altered without affecting the joystick. Only rigidbody constraints and the Grab Interactable's track position / track rotation are affected by automation. Additionally, the rigidbody must remain Kinematic.

NOTE: The Grab Interactable game object's parent is cleared when grabbed (how the XR Interaction system works), then immediately re-parented to maintain local position and rotation calculations (executed in XRKJ Controller's Activate() method) This re-parenting is fundamental to the function of the joystick.

- XR Grab Interactable
    - By default, the Select Entered & Select Exited events on the XR Grab Interactable are used to call XRKJ Controller's Activate() and Deactivate() methods.
        - This is very important yet easy to miss.
    - Uses Attach Point for the Attach Transform field.
    - Uses the Handle's collider in the Colliders field.

Pivot

By default, the pivot has a sphere mesh. However, it only needs to exist as a transform. Similarly, the Handle and Grip do not need meshes.

It is important that the Handle remains as a child to the pivot, so that the visuals move appropriately.

Handle & Grip

The Handle and/or Grip needs to have a collider. This is used in the XR Grab Interactable's Colliders field for controller/hand detection.

- Meshes for these objects can be freely changed.

Base

This is a cosmetic feature and can be removed, replaced, or have its mesh/collider changed.

How to…

NOTE: Look at the scripts attached to controlled objects in the examples from the demo scene.

Using XRKJ Receiver

0. Set up a player using Unity's XR Interaction Toolkit.
1. Add the XRKJ Joystick Prefab to the scene.
    a. Set the Group integer to any value.
2. Try grabbing the joystick.
    a. It should be grabbable and move without any set up.
3. Add an object (with a mesh) of your choice to the scene.
    a. Give this object the XRKJ Receiver component.
    b. Set the Group integer to the same Group value as the joystick.
4. Create a new script for the object's behavior and add it to the object.
    a. Reference XRKJReceiver.Input to get the Output from the controller.
    b. See the example scripts in Assets > OHGAR > XRKJ Scripts
        i. All examples except XRKJPixelExample use the Reciever component.
    c. Do not add the object's behavior code to the XRKJ Receiver component as it will lose its reusability.
5. Try using the joystick again and see if your code works.

Using Output Scriptable Object

0. Set up a player using Unity's XR Interaction Toolkit.
1. Add the XRKJ Joystick Prefab to the scene.
    a. Place the XRKJOutputSO in the Output Scriptable Object field.
        i. You can create additional XRKJOutputSO by right clicking the desired folder > create > XRKJOutput > rename
        ii. You can also create your own Scriptable Object, but it must use a Vector3 variable called Value (unless you want to change code in XRKJController).
2. Try grabbing the joystick.
    a. It should be grabbable and move without any set up.
3. Add an object (with a mesh) of your choice to the scene.
4. Create a new script for the object's behavior and add it to the object.
    a. Reference the XRKJOutputSO to get the Output from the controller.
    b. See the XRKJPixelExample in Assets > OHGAR > XRKJ Scripts
5. Try using the joystick again and see if your code works.

Debugging, Common Issues, and Tips

Activation & Deactivation

- Check that the Grab Interactable component is calling XRKJ Controller Activate() and/or Deactivate() in its Unity Events section.
- If using an active/inactive state on your object, check that it is being accessed in the XRKJ Controller Unity Events section, or some other way.

Groups

- Check for matching group IDs between controller and receiver
- Check that there are no overriding group IDs
    - Each controller must use a unique receiver with a matching Group ID.

Movement

- If you want to move in two dimensions use (x, y, 0f), (x, 0f, z) ...
    - A Vector3 is needed since that is what the joystick outputs. Alternatively, a conversion from V3 to V2 is usable.
- If the object is moving in unexpected directions
    - Try using negative values.
        - (x, y, -z), (-x, y, z) ...
    - Try rearranging axis values.
        - (z, x, y), (y, x, z) ...
- When controlling objects with physics, use the input in FixedUpdate().
    - Make any necessary adjustments to the input in Update().

Optimization

- Check that Activate On Start is false (unless otherwise necessary).
- Use active/inactive states to avoid unnecessary Update() calculations.

Prefab

- Make a duplicate of the prefab before altering it. This way you have a reference to the default settings.
- The prefab can be a child of another game object.
- Make sure to scale the prefab evenly and while it is not a child of another game object.

Scriptable Object

- Check that you are setting the value accordingly when starting/ending or activating/deactivating the game/controller.
- Check that the corresponding Scriptable Object is the desired controller/object
    - It is helpful to use unique names when naming a new Scriptable Object.

From OHGAR

Thank you for downloading XRKJ!

This is my first public Unity asset and I’m grateful for your support. If this package manages to gain a reasonable number of downloads, I will update/maintain a paid version of the asset.