

MixedRealityToolkit-Unity/README.md at v2.0.0 · microsoft/MixedRealityToolkit-Unity · GitHub

 github.com/microsoft/MixedRealityToolkit-Unity/blob/v2.0.0/README.md

microsoft



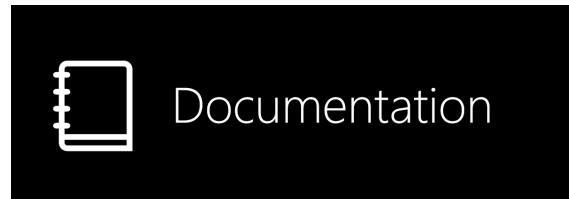
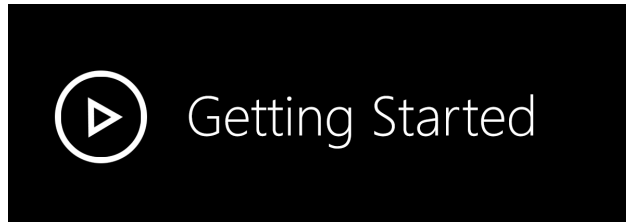
What is the Mixed Reality Toolkit

MRTK-Unity is a Microsoft driven project that provides a set of components and features to accelerate cross-platform MR app development in Unity. Here are some things MRTK does:

- Provides the **basic building blocks for Unity development on HoloLens, Windows Mixed Reality, and OpenVR.**
- Enables **rapid prototyping** via in-editor simulation that allows you to see changes immediately.
- Designed as an **extensible framework** that provides developers ability to swap out core components.

- **Supports a wide range of platforms**, including
 -
 -
 - Windows Mixed Reality headsets
 - OpenVR headsets (HTC Vive / Oculus Rift)

Getting Started with MRTK



[Getting Started Guide and Documentation](#)

[API Reference](#)

Build Status

Branch	CI Status	Docs Status
mrtk_development	 Azure Pipelines never built	 Azure Pipelines set up now

Required Software



[Windows SDK 18362+](#)

To build apps with MRTK v2, you need the Windows 10 May 2019 Update SDK. To run apps for immersive headsets, you need the Windows 10 Fall Creators Update.



[Unity 2018.4.x](#)

The Unity 3D engine provides support for building mixed reality projects in Windows 10



[Visual Studio 2019](#)

Visual Studio is used for code editing, deploying and building UWP app packages



[Simulator \(optional\)](#)

The Emulators allow you test your app without the device in a simulated environment

Feature Areas



[Input System](#)



[Hand Tracking \(HoloLens 2\)](#)



[Eye Tracking \(HoloLens 2\)](#)



[Profiles](#)



[Gaze + Gesture \(HoloLens\)](#)



[UI Controls](#)



[Solvers](#)



[Multi-Scene Manager](#)



[Spatial Awareness](#)



[Diagnostic Tool](#)



[MRTK Standard Shader](#)



[Speech & Dictation](#)



[Boundary System](#)

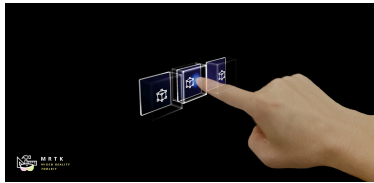


[In-Editor Simulation](#)



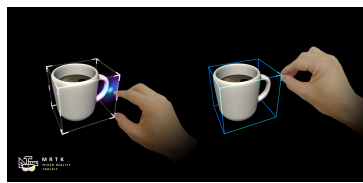
[Experimental Features](#)

UI and Interaction Building blocks



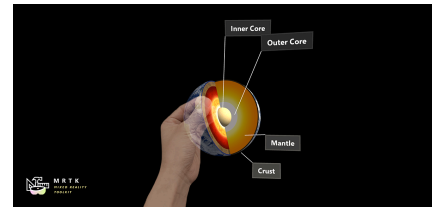
Button

A button control which supports various input methods including HoloLens 2's articulated hand



Bounding Box

Standard UI for manipulating objects in 3D space



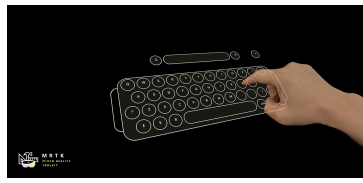
Manipulation Handler

Script for manipulating objects with one or two hands



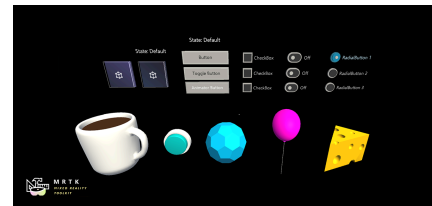
Slate

2D style plane which supports scrolling with articulated hand input



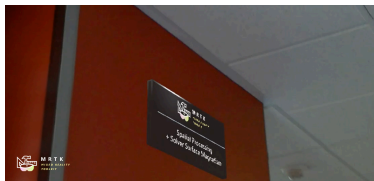
System Keyboard

Example script of using the system keyboard in Unity



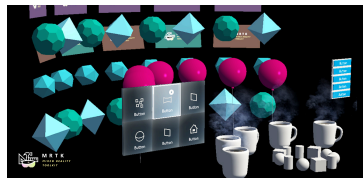
Interactable

A script for making objects interactable with visual states and theme support



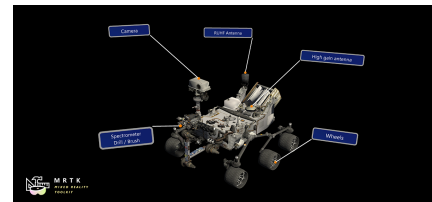
Solver

Various object positioning behaviors such as tag-along, body-lock, constant view size and surface magnetism



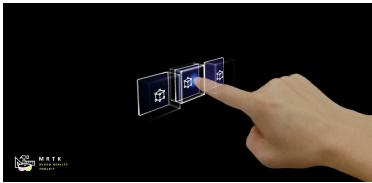
Object Collection

Script for lay out an array of objects in a three-dimensional shape

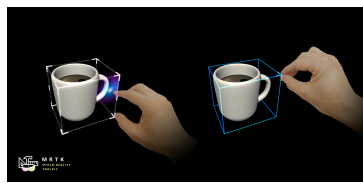


Tooltip

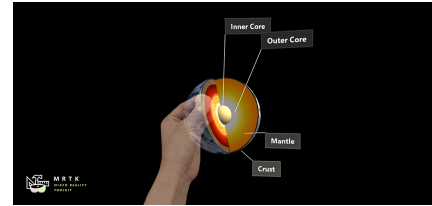
Annotation UI with flexible anchor/pivot system which can be used for labeling motion controllers and object.



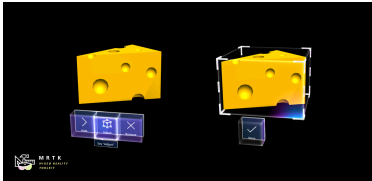
[Button](#)



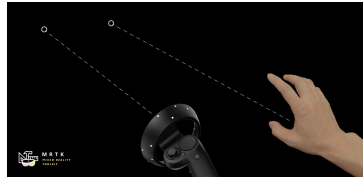
[Bounding Box](#)



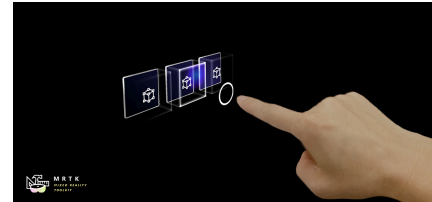
[Manipulation Handler](#)



[App Bar](#)



[Pointers](#)

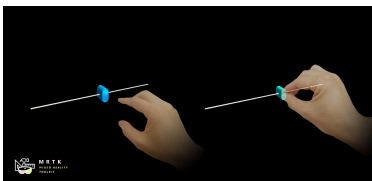


[Fingertip Visualization](#)

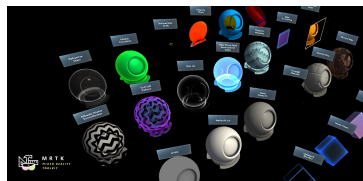
UI for Bounding Box's manual activation

Learn about various types of pointers

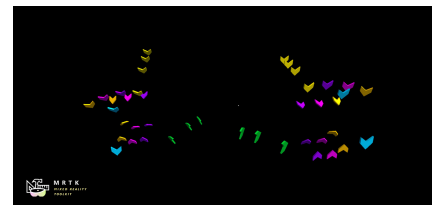
Visual affordance on the fingertip which improves the confidence for the direct interaction



[Slider](#)



[MRTK Standard Shader](#)

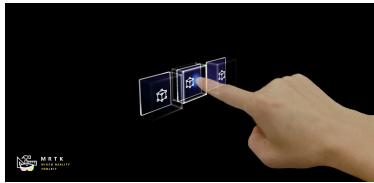


[Hand Joint Chaser](#)

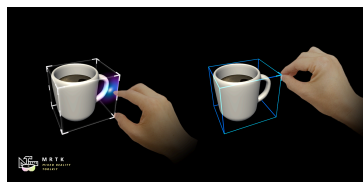
Slider UI for adjusting values supporting direct hand tracking interaction

MRTK's Standard shader supports various Fluent design elements with performance

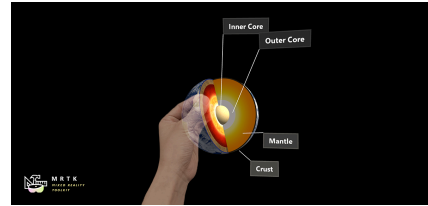
Demonstrates how to use Solver to attach objects to the hand joints



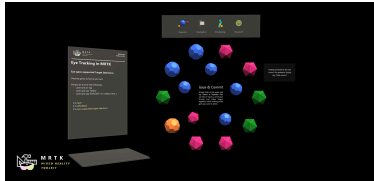
[Button](#)



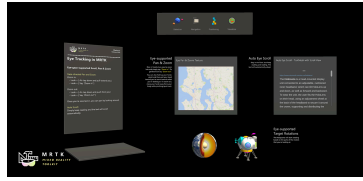
[Bounding Box](#)



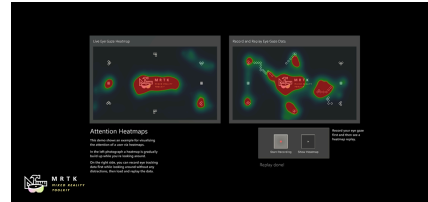
[Manipulation Handler](#)



[Eye Tracking: Target Selection](#)



[Eye Tracking: Navigation](#)



[Eye Tracking: Heat Map](#)

Combine eyes, voice and hand input to quickly and effortlessly select holograms across your scene

Learn how to auto scroll text or fluently zoom into focused content based on what you are looking at

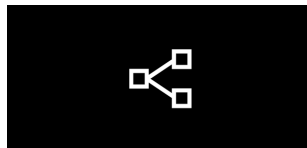
Examples for logging, loading and visualizing what users have been looking at in your app

Tools



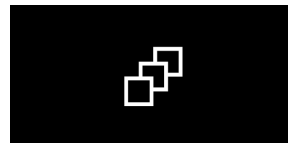
[Optimize Window](#)

Automate configuration of Mixed Reality projects for performance optimizations



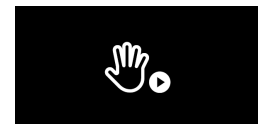
[Dependency Window](#)

Analyze dependencies between assets and identify unused assets



[Build Window](#)

Configure and execute end-to-end build process for Mixed Reality applications



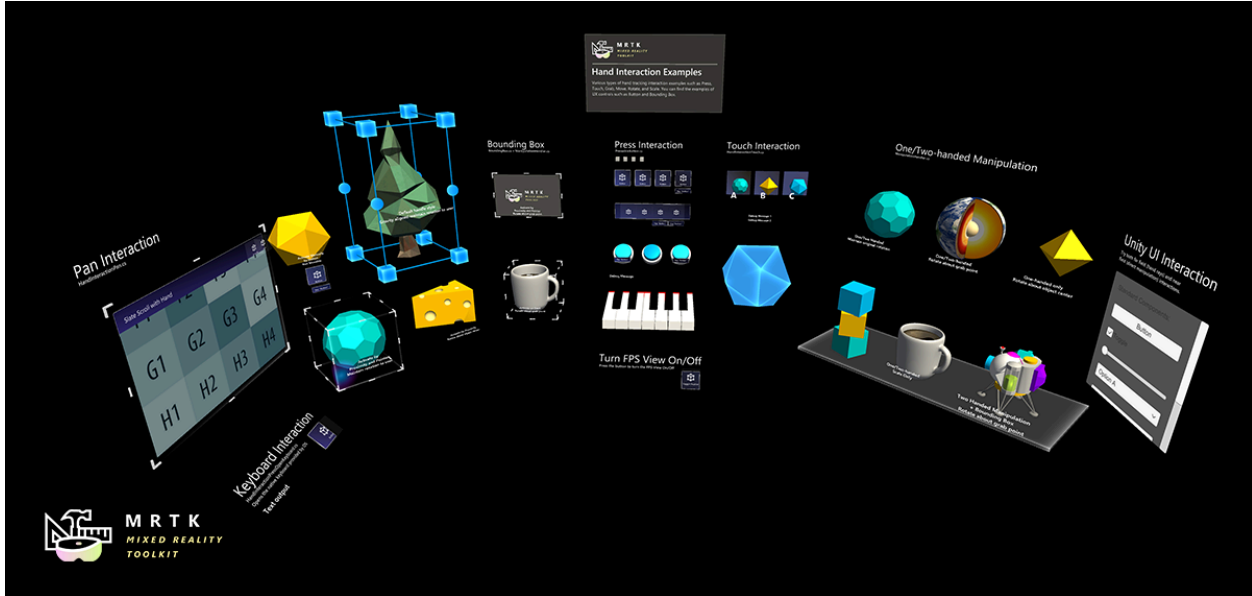
[Input recording](#)

Record and playback head movement and hand tracking data ins editor

Example Scenes

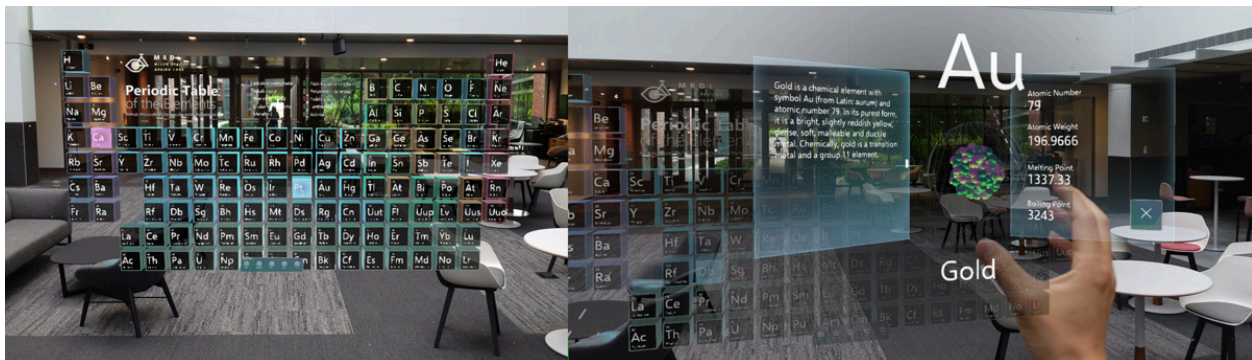
Explore MRTK's various types of interactions and UI controls in [this example scene](#).

You can find other example scenes under [Assets/MixedRealityToolkit.Examples/Demos](#) folder.



Sample App

[Periodic Table of the Elements](#) is an open-source sample app which demonstrates how to use MRTK's input system and building blocks to create an app experience for HoloLens and Immersive headsets. Read the porting story: [Bringing the Periodic Table of the Elements app to HoloLens 2 with MRTK v2](#)



Engage with the Community

- Join the conversation around MRTK on [Slack](#).

- Ask questions about using MRTK on [Stack Overflow](#) using the **MRTK** tag.
- Search for [known issues](#) or file a [new issue](#) if you find something broken in MRTK code.
- Join our weekly community shiproom to hear directly from the feature team. (link coming soon)
- Deep dive into project plan and learn how you can contribute to MRTK in our [wiki](#).
- For issues related to Windows Mixed Reality that aren't directly related to the MRTK, check out the [Windows Mixed Reality Developer Forum](#).

This project has adopted the . For more information see the [Code of Conduct FAQ](#) or contact with any additional questions or comments.

Useful resources on the Mixed Reality Dev Center



[Discover](#)

Learn to build mixed reality experiences for HoloLens and immersive headsets (VR).



[Design](#)

Get design guides. Build user interface. Learn interactions and input.



[Develop](#)

Get development guides. Learn the technology. Understand the science.



[Distribute](#)

Get your app ready for others and consider creating a 3D launcher.

Useful resources on Azure



Spatial Anchors

Spatial Anchors is a cross-platform service that allows you to create Mixed Reality experiences using objects that persist their location across devices over time.



Speech Services

Discover and integrate Azure powered speech capabilities like speech to text, speaker recognition or speech translation into your application.



Vision Services

Identify and analyze your image or video content using Vision Services like computer vision, face detection, emotion recognition or video indexer.

Learn more about the MRTK Project

You can find our planning material on [our wiki](#) under Project Management Section. You can always see the items the team is actively working on in the Iteration Plan issue.

How to Contribute

View the [How To Contribute](#) wiki page for the most up to date instructions on contributing to the Mixed Reality Toolkit!

For details on the different branches used in the Mixed Reality Toolkit repositories, check this [Branch Guide here](#).