AR Tests

Libraries for AR

| ARCore |
| --- |
| Proprietary |
| Software development kit created by google. It runs on Android devices and it has support for IOS devices.  Programing language: kotlin/java |
| Supported devices: <https://developers.google.com/ar/devices> |

| ARKit |
| --- |
| Proprietary |
| Apple SDK to create AR applications.  Programing language: (Swift ?? ) |
| Supported devices: https://developer.apple.com/library/archive/documentation/DeviceInformation/Reference/iOSDeviceCompatibility/DeviceCompatibilityMatrix/DeviceCompatibilityMatrix.html |

| Unity Mars |
| --- |
| Proprietary |
| Unity Mixed and Augmented Reality Studio (**MARS**), which provides developers with additional functionality for rules-based generation of augmented reality (**AR**) applications. It sits on top of the **Unity AR foundation**. It supports IOS (**ARKIT**) and Android (**ARCORE**).  MARS uses **Unity XR ToolKit** to support user interactions. This library has been changed in the last releases making old unity applications to migrate their button-event mapping to the new system. |
| Installation:   * Download unity hub and install unity 2020.3 * Make sure you add the support for desktop, android and ios * Download unity mars package. * Create a unit project * Add the unity mars package asset to the project. * Make sure the dependencies to unity mars have the same number of the Unity AR foundation. i.e:   Unity Mars 4.1.9 - Unity AR foundation 4.1.9 - ARKit 4.1.9 |
| Demos:   * Basic Placement * Semantic segmentation * Conditional Placement |
| The documentation is updated on every release (**1.4.0**). However, going beyond the basic examples (depth calculations) is not too intuitive. The support in the Unity forums is fast and reliable. They will contact you personally by email if necessary. |
| Personal Opinion: It’s easy to install and use. It seems to have plenty of functionalities, but I couldn't find documentation on how to use elements from the environment to build a demo. Placing and the simulation view are the strongest point on Unity Mars |

| ARDK |
| --- |
| OpenSource |
| The Lightship AR Developer Kit brings real-time mapping technology together with shared multiplayer experiences, semantics and depth for more realistic AR experiences. ARDK is made for developing AR experiences for both Android and iOS mobile platforms – and integrates directly within Unity. |
| Programing language: C# |
| Installation:   * Create an account on: <https://lightship.dev/signin?continue=/account/dashboard> * Create a token key * Download the ARDK package for Unity and add it to the project. * In the Resource script add the key you created previously.   You have to sign in to have access to the tutorials.  *The documentation suggests developing on Unity + Mac devices. The windows package generates linker errors.* |
| Demos:   * Basic Placement * Semantic segmentation * Depth Buffer |
| Documentation is not constantly updated, they keep old videos and references that dont apply to new releases. However, it is open source and filling the gaps between versions is not difficult. The code can be debugged using a regular c# debugger. The AR manager system is easier to understand and manipulate, making it easy to implement new features. |

| IATK |
| --- |
| ATK: Immersive Analytics Toolkit is a Unity project to help you build high quality, interactive and scalable data visualisations in Immersive Environments (Virtual/Augmented Reality). Use the Visualisation script to create data visualisations interactively in the editor, press play and view and interact with your data in V/AR. Write simple code to use the IATK core graphics components to make your own interactive visualisations programitcally. |
| *It has limitations on the Unity versions supported. The code generates problems in Unity > 2019. Many of the tutorials are in Unity 2017 which has an outdated XR Input Library non compatible with Unity Mars.*  It depends on [maptool bos for unity](https://docs.mapbox.com/unity/maps/guides/) and [VRTK 3](https://vrtoolkit.readme.io/) |
| Installation:   * Download [maptool bos for unity](https://docs.mapbox.com/unity/maps/guides/) * Clone the IATK repo: <https://github.com/MaximeCordeil/IATK.git> * Clone the VRTK repo: <https://github.com/ExtendRealityLtd/VRTK.git>   Switch the git repo to tag 3.3.0   * Install Unity 2019 and open the example project in the VRTK repo. * Add the IATK package form the repo * Add the maptoolbox package in the project. |

| RagRug |
| --- |
| OpenSource |
| Main repo: https://github.com/philfleck/ragrug |
| It is a server-client application using node-red js to capture real time data from devices, MQTT Protocol to communicate the device with the client, and IATK as a UI software to display the data using AR/VR/XR. Because of this, the client side would be a Unity application where the user can manipulate data using Unity events with [VRTK 3](https://vrtoolkit.readme.io/).  There is documentation on how to install dependencies and start working with the base code. |
| Couldn't do a demo on this software |