"How to VR for Dummies" Part 1

As a developer, I need to gather information for a "how to VR for dummies"

- Look into features inside the Unreal engine
    - Pipeline integration
        - Datasmith
            - includes support for importing part and assembly data from native SolidWorks and CATIA files—including basic color extraction, hierarchy, and metadata—and preparing it for real-time use in Unreal Engine.
            - “Save hours, if not days importing data into Unreal Engine”
        - FBX, USD, and Alembic support
            - Connect to media production pipelines, with support for industry standards like FBX, USD, and Alembic. First-class USD support enables users to collaborate better with team members and work in parallel.
    - Animation
        - Live Link data streaming
            - The Live Link plugin enables you to connect a stream of real-time data from an external source to Unreal Engine. You can stream character animation, cameras, lights, and other data from DCC tools such as Maya or Motionbuilder, or from motion capture or performance capture systems, including Apple’s ARKit face tracking, so you can capture facial performances from an iPhone. Live Link is designed to be extensible through Unreal Plugins, enabling third parties to add support for new sources.
    - Integrated Media Support
        - Unreal Audio Engine
            - Enhance your project’s audio with a rich audio feature set that includes real-time synthesis, dynamic DSP effects, physical audio propagation modeling, OSC support, layered Sound Concurrency, a Spectral Analyzer for Submixes, and the ability to bake Spectral Analysis curves and envelopes. Recent additions include convolution reverb processing and sound field rendering.

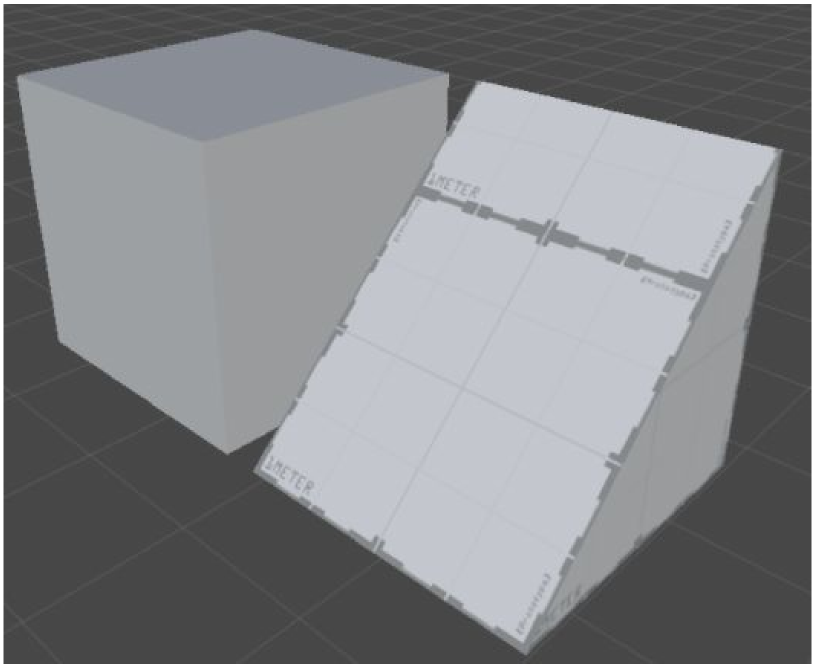
Research different software to guide the “how to”

- Tutorial on how to develop VR
    - **C#**

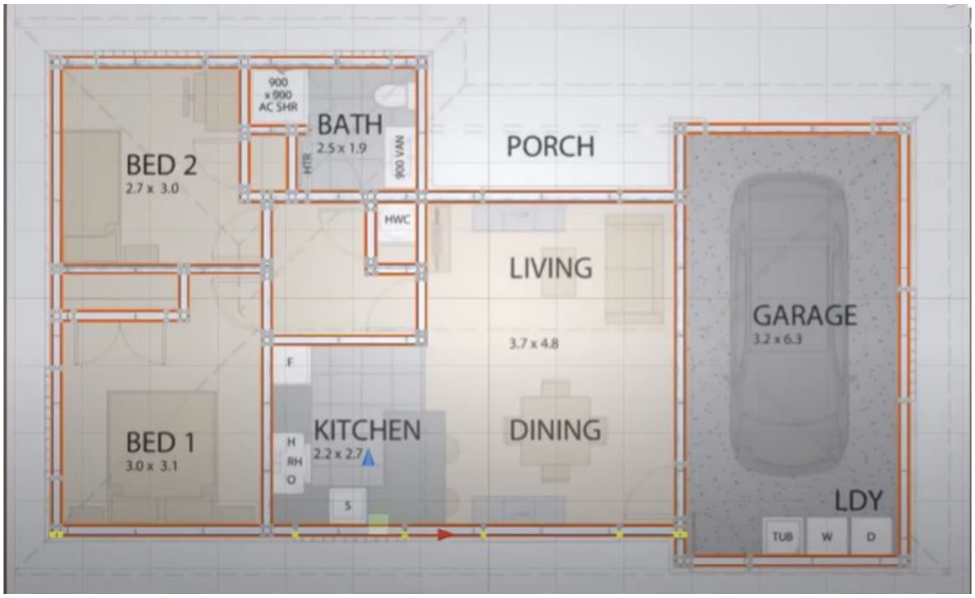
    - It works across both platforms (PC and Mac) and most of the VR applications are coded in Unity. Using the most popular language for the most popular engine is a ‘virtual’ no-brainer. It also has an extensive network of enlisted designers—should you need development help.
  - C++
    - If you’re going to develop something on a smaller scale, you may go with C#, but if you’re going big-time and need more…you’ll need Unreal and C++.
  - Java
    - Like C#, this object-oriented language is also useful for cross platform applications (PC and Mac).The new 3D API lets VR applications generate the needed 3D images in VR and helps track body movements well.
  - JavaScript
    - Since it’s the language understood by most browsers, JavaScript works well for the WebVR platform involved in creation of web based Virtual Reality applications. It works with developers to readily create VR apps for the web (and for download).
  - **Python**
    - Perhaps the easiest of the programming languages to learn, Python is a good language to learn for someone just beginning the field of VR development. The language itself is pretty straightforward to understand; it also allows a user to develop their ideas with fewer code lines. This means that it’s also faster as well as easier. Developers can create VR projects quickly due to several interesting packages associated with Python.
  - Blueprints
    - Visual Scripting system in Unreal Engine is a complete gameplay scripting system based on the concept of using a node-based interface to create gameplay elements from within Unreal Editor. In addition, Blueprint-specific markup available in Unreal Engine's C++ implementation enables programmers to create baseline systems that can be extended by designers.

"How to VR for Dummies" Part 2

As a developer, I can add on to the "VR for dummies" on the measurements inside of Unity:

- Research the interchangeability of coordinates inside Unity
    - Probuilder features
        - I can duplicate objects and place them in different locations to interchange the coordinates of the objects.
        - I can mirror an entire scene to replicate the stations to other tooling stations.
        - I can insert edge loops onto objects to distort them more easily.
    - ProGrids
        - I can snap multiple vertices of an object together to modify the objects to make the coordinates interchangeable.
        - It is especially handy for modular or tile-based environments, but it also significantly speeds up workflow and increases quality for all types of work.
        - The world grid never changes position or orientation. This means you always know exactly where items are, how far they are from each other, and how far you are moving them.
- Map the units in Unity to real life scenarios
    - 

    - Take into account that unity works on a metric system rather than the imperial system

- 

- Convert feet into meters when doing metric conversions