NEED 3.3Volt power source

Download visual studio 2019 in c++

Header files go in the include folder

Lib folder holds open vr lib file

For new projects: project file -> properties -> configuration properties -> vc++ directories -> include your include folder and same for library directories

3 driver classes:

* Controller driver: talks to device
  + Multiple devices = multiple of these classes
  + Activate method
  + Tells openvr about your device; set properties
    - Sends an unique id from driver
    - Where you should store properties of your driver
  + Vr driver manifest:
    - Where you specify name of driver
  + Controller type:
    - I.e. left hand, right hand, stylus, treadmill
  + Scalar components:
    - Where you tell open vr what kind of input
    - I.e. joystick and trackpad
  + Absolute = things like joysticks or trackpads
  + Boolean = yes or no like a button
  + Serial: add a serial number for your controller
  + Driver poise: represents position of your devices in space
  + Deactivate: when openvr is closing down you can dispose of whatever
  + Enter standby: low power mode
* Device provider: manages device
  + Creates instances of classes
  + Loops through devices
  + Init: when driver is being initialized
    - Init the driver contacts
  + Should block standby mode: want device to block steam from going into standby
* Device factory: returns an instance of provider
  + Used to get provider back and calls it every frame

Input file tells how applications are bound to input states

Hmd - device is a hmd and wants the binding ui to be in hmd mode when configuring it for input

Single\_device: device should be on a page by itself (x-box or gun style controllers)

Controller\_handed: the device used as a pair (one in each hand) can be specific to left or right

**“Imu” -** Internal Measurement Unit:

Gyroscope

Accelerometers

Magnetometers

Connecting the driver to openVR:

Unity takes data from open vr (from controller) and puts it into the game, steam vr runs game,

Drivers translates controller data to open vr as an object

Unity finds control object and imports it into game software

Data from controllers →openVR →unity →game(steamVR)

Lighthouse -> sensors on controller → microcontroller → driver applications → openVR driver API -> logged in steamVR

Sensor angles + imu data gets translated through the driver\_lighthouse →steamVR and openVR Driver

**Json id - input profile**

* *Manufacturer* - String
* *Model No.* - String
* *Device Class* - String
* *Device*