# Issues Document and FAQ

# Oculus Rift and Kinect Issues

## Movement:

Movement issues have been experienced because the only way to move forward while using the Kinect and oculus is through some form of gesture. The idea is to push your hands away from your body to stop moving forward and to pull towards your body to start moving. The Kinect does not pick up the movement that precisely so you can push which the Kinect would sometimes read as pulling and vice versa.

To turn the users body the plan was to use other gestures such as the swipe left and swipe right which would simulate the user actually turning their body but the Kinect sensor for this type of movement is very crude and rarely ever picks up the movement.

The conclusion is the gesturing for the Kinect is very inefficient and makes it hard to perform any actions in regards of spinning the user’s body as well as moving forward and backwards.

The solution that we proposed was mapping the Kinect body to the oculus movement so that when you turned your head your body would turn with you but to do this the rotational values must be removed from the Kinect sensor which enables the mapping of the sensor to a user’s body. Unity does not allow us to change a single variable from multiple sources to avoid collisions in the code so this method cannot work because we require the Kinect and the oculus to be working together not the Kinect miss-reading the user’s body and glitching or failing to read the user all together.

## Next Proposed solution

The next idea to arise while I was writing up this document was to make a box which recorded its past movement and reset if it loses track of the user that will act like the gesture controller but within the 2D space of the camera view. This means if we map the object to the player’s hands and both hands were to move to the right together then the user would turn that way.

## Resting the Hardware

A solution to the Kinect losing the mapping is to reset the Kinect’s visual piece. To perform this, the observer will need to cover the entire lens with his/her hand and then remove it once the patient is back into the correct position.

# Leap Motion Issues

“Currently, grasping objects with hands is still quite jittery. Future SDK or Leap hardware updates should improve on this hopefully.”

USB cable has been known to cause trouble

Leap has to be connected before running executable

Device has to be clean

When building .exe in Unity, have to change “Architecture” to “x86” from x86\_64” in “Build Settings”.

“Leap Service” regularly stops working which also usually crashes Unity. The service and unity both have to be restarted. Sometimes the service takes a few restarts to get working again.

* Applications using Image API must link with updated DLLs.
* The latest version of the Oculus firmware limits bandwidth for the DK2’s built-in USB port, which harms the controller’s performance. For now, we \* recommend using the free cable extender bundled with the mount to connect the controller directly to the computer.
* Tracking performance may degrade when closely facing large reflective surfaces like white walls. (This is a broader issue with V2 tracking that we’re constantly improving on.)
* Tracking degrades when arms are held directly away from the body (i.e. elbows straight).
* Some poses are not currently tracked reliably in head-mounted display (HMD) mode (e.g. pinch/rotate, 3- and 4-finger poses).
* If your application was built linked with the 2.1.0 or 2.1.1 SDK, you must update your DLLs
* avast! Antivirus can interfere with service installation (disable shields to work around)
* WebSockets TLS isn’t supported for Firefox and Linux Chrome
* Linux Chrome version 36 requires setting “Allow insecure WebSocket from https origin” under chrome://flags
* Image API not available in Objective-C
* Fist poses may be less stable (such as curling fingers and rotation)
* Tracking may not work as well with bracelets, rings, watches, sleeves, etc.
* Occasionally, a hand can briefly initialize as the wrong hand (left vs. right)
* Occasionally, a tap gesture will register when a small circle gesture is made
* CPU usage and latency is not yet optimized for skeletal tracking
* Upside-down hands initialize as right-side up (inappropriately)
* Tracking quality is lower when making a fist or with one finger extended
* The WebSocket protocol and JavaScript client library do not support setting gesture parameters.
* On Windows 7, the Chrome browser application sometimes fails to respond to emulated touch points. To fix this; click inside of the program with a mouse.
* The sphereRadius and sphereCenter functions of the Hand class are unstable when the hand is fully open.
* Linux does not support background apps or onFocus callbacks

# OVR issues

Problem: OVR not orienting. OVR is a child of First Person Controller, but when the orientation of FPC was changed the orientation of the OVR would not change. The orientation would change when a normal camera was used.

Fix: “Follow Orientation” variable was not set to any object, fixed it by adding “First Person Controller” object.

# Unity Issues

Problem: Oculus not working in Unity, coming up with error: <RI.Hid> Device is not supported: Tracker DK (0003, 0x0005). Oculus working with Configuration Utility demo scene, and HDMI is working, but tracking is not being supported by Unity and not registering the device. Thought Unity Pro trial had run out but it hadn’t. Thought firmware for Oculus needed to be upgraded to DK2 but not compatible using Oculus version 1.

Note: This problem was continuous, and for multiple team members.

Fix: Reinstalled Unity, version 4.5.2.

# Driver Issues

The driver issues that I have been having were when I was running all of the devices on a windows 8 environment.

I installed every single device drivers on my computer which caused collision errors with all the drivers. The synaptic pointing device which is the default laptop touch pad device had some small errors with the oculus rift which were resolved by turning off the touch pad.

The devices such as the leap motion and razor hydra and Kinect would have similar drivers which on the occasion would cause some errors.

My computer would blue screen due to driver boot issues and would eventually resolve itself after about 5ish blue screens on startup. In the end I reformatted my computer and did a clean install of windows 7.

Everything works wonderful with windows 7 and I have no errors.

The Oculus does not work in unity all the time but will work with the executable version of the program.