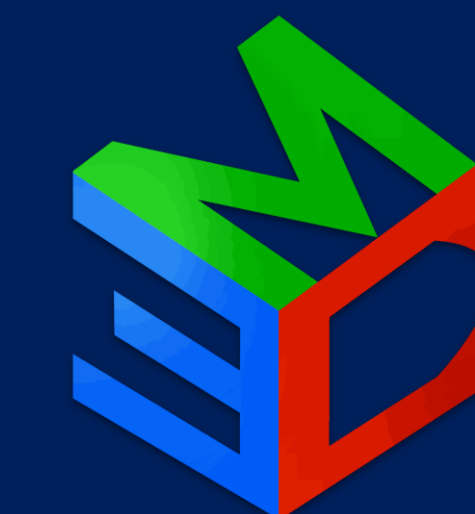




## 3D Mouse

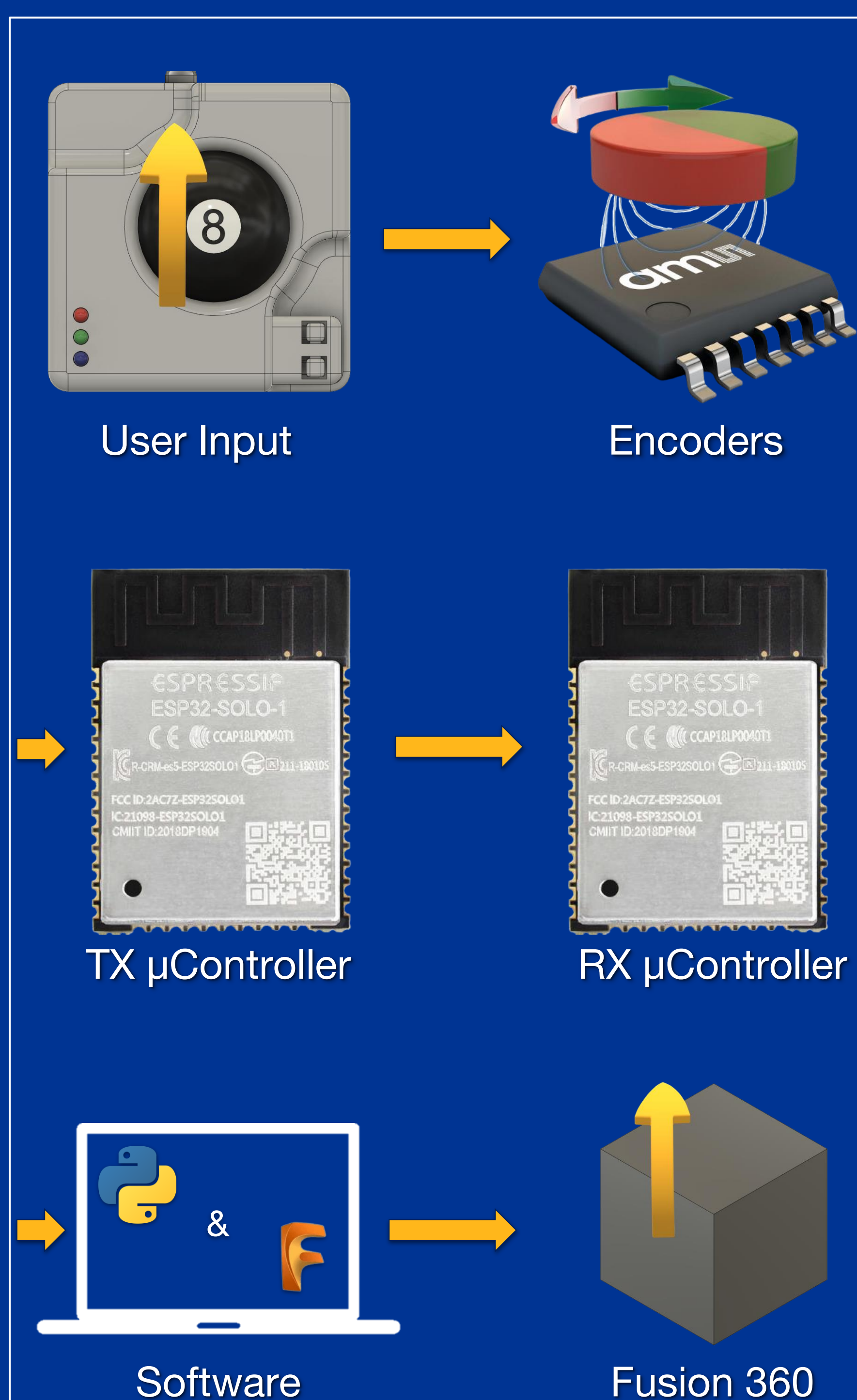
Dylan Butler, Michael Barkand, Kareem Omar, Jared Carl  
Electrical and Computer Engineering



### Background

Interacting with 3D modeling software and other computer-aided design (CAD) packages with a standard computer mouse can be cumbersome. We created a device with capabilities beyond a regular (2D) mouse while improving upon the efficiency when navigating around a 3D model.

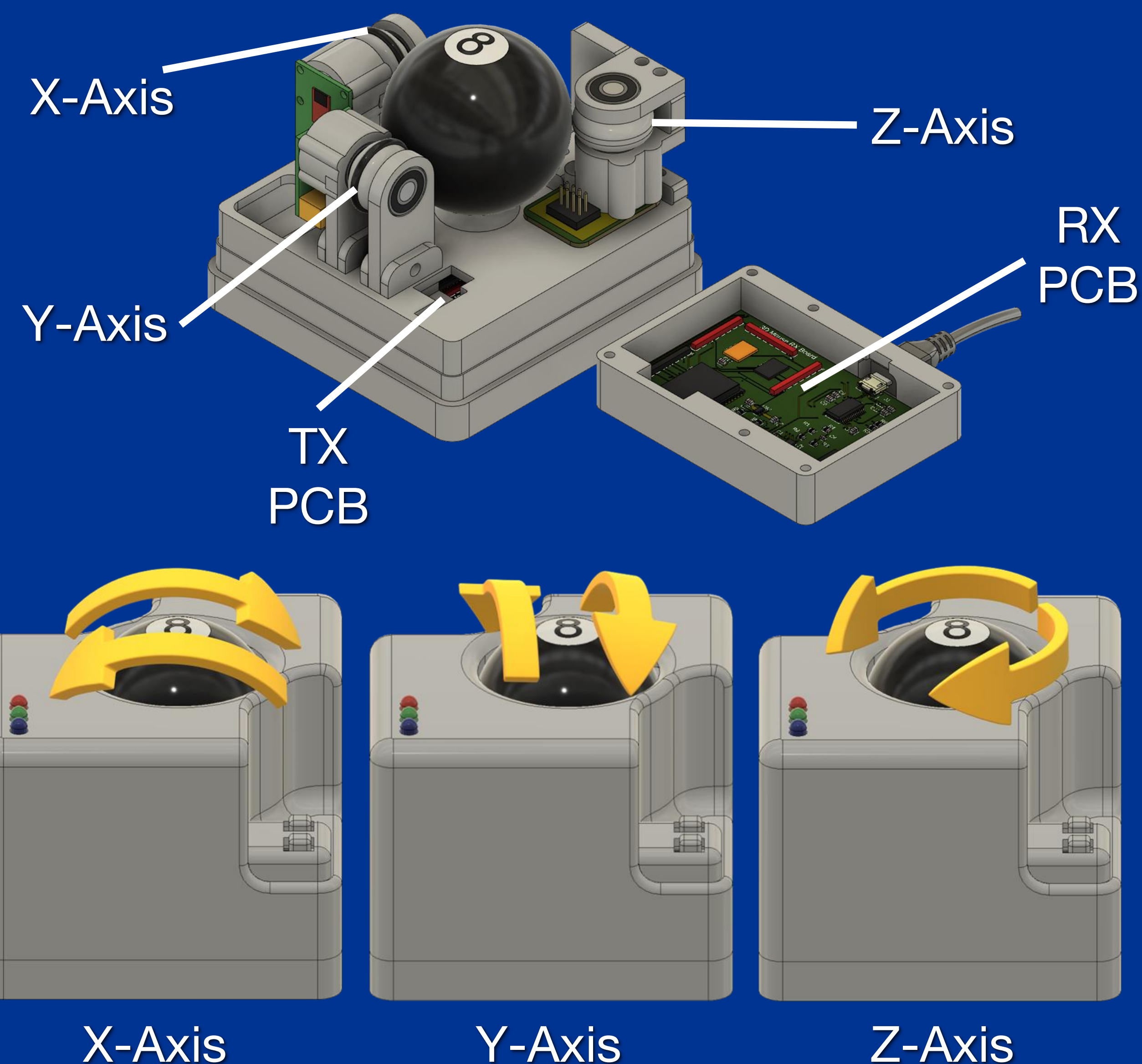
### How It Works



### Details

- Plug and Play operation
  - No driver installation needed
- Wireless communication between device and receiver
- User friendly GUI to change sensitivity settings

### Hardware Design



### Specifications

#### 3D Mouse Specs

- Average Receiver Latency - **82.1 ms**
- Battery Capacity - **2000 mAh**
- Peak Current Draw - **160 mA**
- Average Battery Life - **12.5 hrs**
- Wireless Range - Tested up to **250 ft**

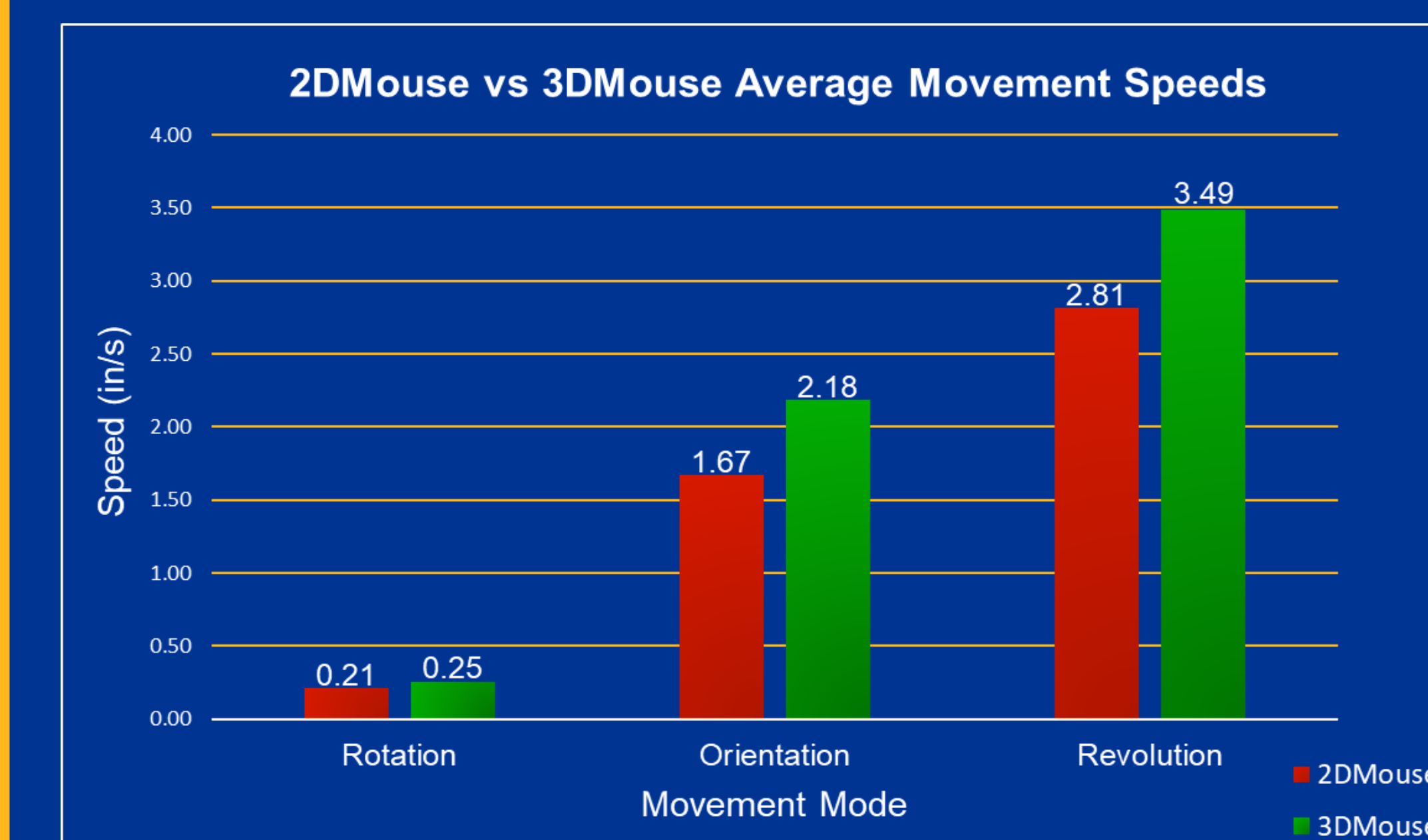
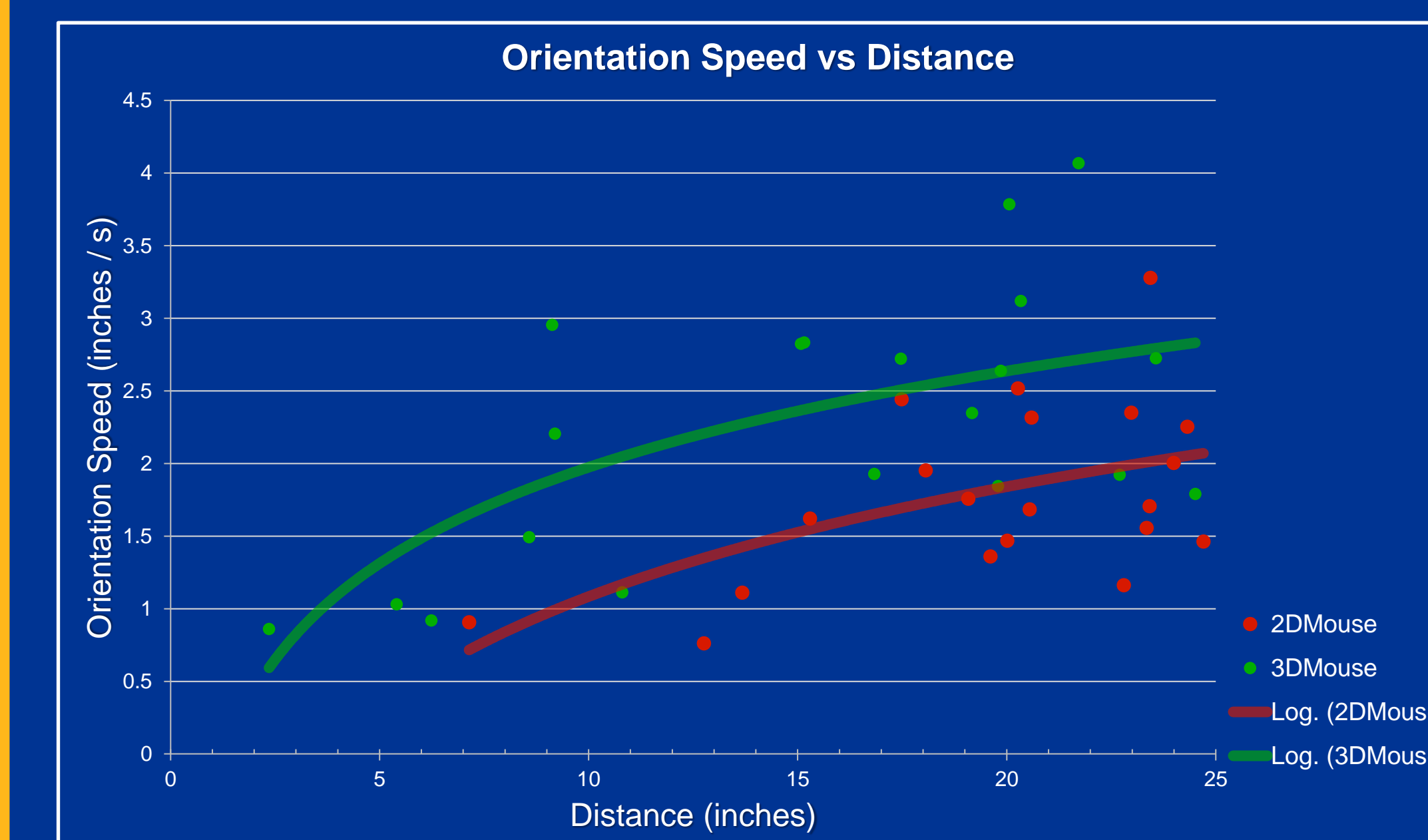
### Testing Results

#### Qualitative Analysis

Surveyed 28 respondents on three metrics:

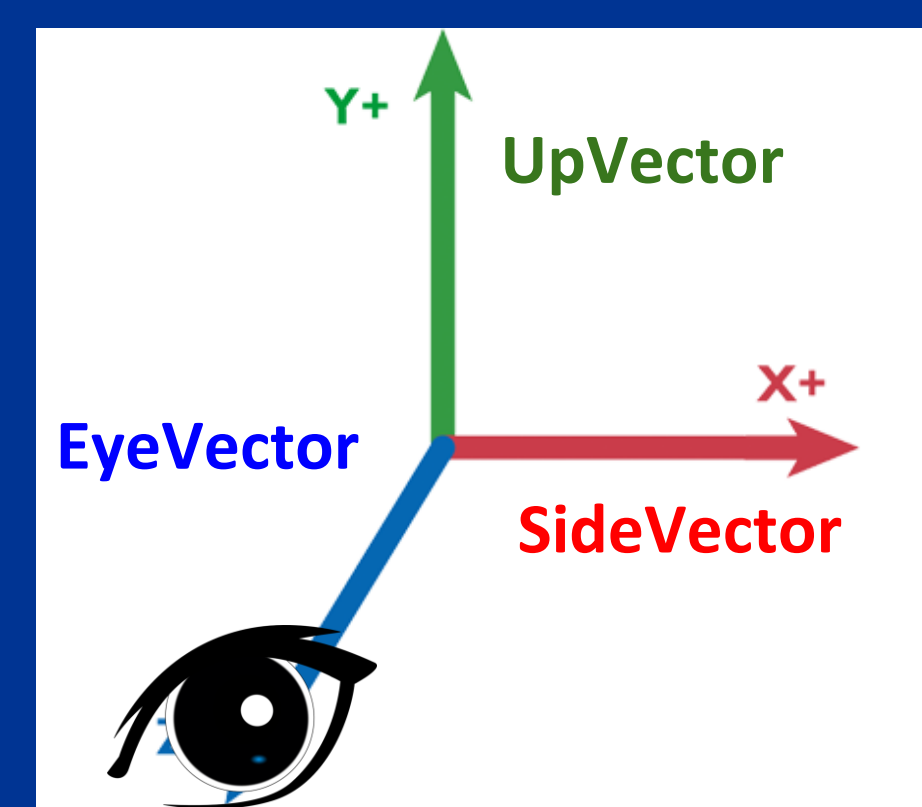
- How likely would they be to use the mouse daily?
  - (1-5) Average: **4.1 (Likely)**
- How comfortable was the mouse to use?
  - (1-10) Average: **8.3 (Relatively Comfortable)**
- How responsive was the mouse during use?
  - (1-10) Average: **8.6 (Relatively Responsive)**

#### Quantitative Analysis



### Software Algorithm

#### Camera Vectors



Three Vectors describe camera position:

- **UpVector**
  - A vector object in the Autodesk Fusion 360 API, always points straight up from the Camera Target
- **EyeVector**
  - Manually created from two point objects in the API, the Camera Eye and Camera Target
- **SideVector**
  - Manually created from the cross product of the UpVector and EyeVector

#### Camera Movement

Moving the Camera in a single direction (x, y, or z) involves re-creating a pair of the three Camera Vectors

- $\Theta$  is the change in direction input from the hardware
- $v1' = |v1'| / |v1| * v1$        $v2' = |v2'| / |v2| * v2$
- $v1' = \cos(\Theta) * v1$        $v2' = \sin(\Theta) * v2$
- $v3 = v1' + v2'$
- Re-create the second Camera Vector by calculating the cross-product of  $v3$  and the third Camera Vector

