**Virtual Computer Mouse using mmWave Radar (71) Demo Plan**

**Chirp Configuration/Data Collection (Greyson Heath)**: My subsystem includes the two radar devices mmwave + IWR6843 (2x) connected to a single computer. I will demonstrate both radar devices streaming their respective data to the computer (in a terminal). Using the python code, the mouse positioning radar will output a varied number of array’s (depending on the points recorded in a single frame) with the following usable information: [X, Y, Z, doppler]. The Gesture radar outputs a fixed array of size 10 corresponding to points in a single frame and their weighted doppler, range, and more for the Gesture Recognition subsystem to use. The radar devices will only record data points up to a certain distance (~1.5 feet), and block out anything past that distance. Additionally, it will only record data points within a specified angle (azimuth and elevation) from the device.

**Gesture Positioning (Daniel Lu)**: My code should analyze the x, y, and z coordinates sent by the radar and output the velocity of the moving gesture. From Grayson's demo, around 2-15 points represent the moving gesture, and the number of points recognized by the radar is not fixed over time. The validation plan of my demo code can be constructed with two test cases. The first case is error coordinates, where the radar recognizes the human body in front of the radar with a few points. The second case is an error velocity, where some points recognized by the radar, representing the gesture, have error velocities. These error velocities should not be sent to the Input Mapping part, because it will cause a shock move of the mouse.

These cases are confirmed with Nick on Tuesday's sponsor meeting.

**Gesture Recognition (Oscar Chavez Araiza)**: For the demo I plan to show the gesture recognition work for real-time values for all 3 gestures (NO\_GESTURE, SHINE, PUSH). I will also show data corresponding to accuracy when tested on live-data, Maximum recognition range, and maximum angle of recognition (Azimuth and Elevation.)

**Input Mapping (Zane Meikle)**: For the demo I plan to run a series of test files to prove the function of my subsection. The first test file will move the mouse to each corner of the screen then left click and right click. The next test files will have errors in the data that will be handled by my code. These errors will include left click and right click inputs that aren't 0, 1, or 2. As well, as test if the X Y vectors are invalid such as if they are a string instead of a number.