**TITLE**

**Conditionals**

**LAB # 05**

**SECTION # 08**

**FULL NAME**

**Adam Jennissen**

**SUBMISSION DATE:**

**10/06/2022**

**DATE**

**10/01/2022**

# Problem

For this lab I had to determine which direction the DualShock 4 controller was facing, and output that direction every time it changed, and the controller was not moving. I also had to use at least 3 different functions, and make the triangle button end the program.

# Analysis

For this problem, I had to take input of the controller’s gyroscope, as well as its moving average in each direction. I also had to take in the triangle buttons value. I had to use all of this information to determine if the controller was moving or not, and which direction it was facing.

# Design

When thinking about how to make this work, I originally thought that I would need a way to figure out if the controller is moving, and a function to determine which direction the controller was facing. In order to do this, I needed a function to figure out if the value was close to the point, so that there is some tolerance. I made one function to determine the orientation, one to determine the magnitude, and one to determine if the value is close to the point. For my tolerance values, I picked .125 for the magnitude, and .2 for the orientation. I picked these after testing around a bit and finding values that I thought felt good. I decided that my orientation function should return a string with the orientation, which allowed me to compare it with a string variable to determine if it was a new orientation or not. To do this I had to use some of my prior knowledge about pointers, because in C you cannot return an array or string from a function without the use of pointers. Due to my use of pointers, I had a small issue with my function returning NULL sometimes, but I was able to fix that with an if statement to make sure it is not equal to NULL. When making the close to function, I just made it check if the value was between the point + tolerance, and the point – tolerance. I also realized that the close to function had to be called a lot to check if the orientation was close to what 1 or -1 and if the magnitude was close to 0.

# Testing

When testing the program, I had to make sure that it output the correct orientation and did not output the same direction multiple times in a row. My first test of my program, I only had it outputting the orientation, it did not check if it was a new orientation. All of my orientations were mixed up, so I had to switch around variable to make it output the correct directions. I then had to test if my program would only output when the controller moved to a new orientation. I also made sure to test that my program would end when pushing the triangle button.

# Comments

This lab taught me a lot about using functions and loops. I learned that in order to have a function output an array or string, you need to use pointers. I also figured out how to make a function only output when the value changes. I also got a better understanding of how the -g and -a flags for the DualShock 4 controller work.