

Unit tests

1. For the first unit test, I would check whether the drone task is accurately calculating the drone's velocity. In order to do this, I would hardcode a set of values to represent angular velocities read from the GYRO. I would input these values into the drone task and see if the velocity which is calculated is what I would expect it to be. This cut would be made on the drone velocity value which would be sent from the drone task to the LCD output task. This cut will isolate whether the drone velocity calculation is correct. The stimulus will be the drone semaphore being posted and then acquired by the drone task.
2. A second unit test could be made at the output of the LCD output task. This value will be the next position of the drone. Assuming that I have hardcoded the GYRO values as described in the previous unit test, I will expect the LCD output task to use a predetermined velocity (hardcoded) in order to calculate the next position of the drone. This would also require hardcoding the original position of the drone. This cut will determine whether the LCD output task accurately calculates an appropriate shift in the drone for a specific angle of the GYRO. The stimulus will be the LCD semaphore being posted by the GYRO task.

Functional Tests:

1. X movement of ball

I will begin by holding the board flat. I will then hold it lengthwise and tilt it left to right. The ball should move left and right accordingly.

2. Y movement of ball

I will begin by holding the board flat. I will then hold it lengthwise and tilt it forwards and backwards. The ball should move forwards and backwards accordingly.

3. Ball stays stable when the board is set down

I will run the program and set the board flat on a table. I will expect to see the ball stay in about the same position that it started in.

4. Ball does not leave boundaries of hard edged board

I will run the program on default settings (hardEdged = true) and turn the board so that the ball approaches the edges of the board. I will turn the board towards both the vertical and horizontal edges of the grid in order to make sure that the ball does not leave this grid.

5. Button pressed

In order to test that my program successfully is able to detect button presses, I will begin the program and press the button and see the ball changes color as a result.

6. Button released

In order to test that my program successfully detects button releases, I will begin the program and press the button and see the ball changes color as a result. I will then release the button and see the ball change to the original color.

7. Waypoints

I will run the program with the default settings and see that there are outlines of circles showing where the waypoints are. I should see one circle is a different color to show that it is the next waypoint I am supposed to pass through. After I pass through this waypoint, it should turn a

third color to show that I have already passed through it and I should see the next waypoint light up.

8. Holes

I will test the holes by running the program in default mode and tilting the board until my ball reaches a hole. I should see the game end.