

## Unit Testing Plan

Test	Description	Cutting Point	Status
<b>Unit Test #1:</b> Angle Change	Test that the functions to increment and decrement angle setpoint modify variable successfully.	Angle task functions (after sem post)	Pass
<b>Unit Test #2:</b> Throttle Change	Test that the functions to increment and decrement throttle setpoint modify variable successfully.	Isolates throttle task functions (cuts out physics)	Pass
<b>Unit Test #3:</b> Rocket Translation	Test that vertices of the rocket polygon update to correct location after translation (coordinate movement).	Isolates moving rocket	Pass
<b>Unit Test #4:</b> Rocket Rotation	Test that vertices of the rocket polygon update to correct location after rotations.	Isolates rotation change of displayed rocket	Pass
<b>Unit Test #5:</b> LCD Display	Test that display is working and all include dependencies working for graphics.	LCD and inclusions	Pass
<b>Unit Test #6:</b> Config Input	Confirm that the configuration parameters are read and stored correctly.	Configuration input	Pass
<b>Unit Test #7:</b> PWM Frequency	Confirm that a timer is set up correctly for the given frequency to make a PWM.	PWM frequency	Pass
<b>Unit Test #8:</b> Physics to Rocket	Test that the physics task modifies the rocket struct as expected.	Isolates physics task and mutex to rocket	Pass
<b>Unit Test #9:</b> Physics Thrust	Assert that the physics task calculates thrust correctly.	Isolates thrust calculation	Pass
<b>Unit Test #10:</b> Physics Blackout	Confirm that blackout is sensed correctly for the given configuration.	Isolates blackout determination	Pass

### Functional Testing Plan

Test	Description	Status
<b>Functional Test #1:</b> Game Start	Confirm the game starts and LCD displays graphics.	Pass
<b>Functional Test #2:</b> Button 0	Confirm pressing Button 0 causes the rocket to rotate counter clockwise.	Pass
<b>Functional Test #3:</b> Button 1	Confirm pressing Button 1 causes the rocket to rotate clockwise.	Pass
<b>Functional Test #4:</b> Slider Throttle	Test that the position on the slider changes the fuel burn rate (also consider position slider not pressed).	Pass
<b>Functional Test #5:</b> LED0	Test that LED0 shows current thrust as a % of the maximum via pulse width modulated lighting.	Pass
<b>Functional Test #6:</b> LED 1 Normal	Test that LED1 shows current acceleration as a % of the maximum.	Pass
<b>Functional Test #6:</b> Win	Confirm the game can be won (may take several tries).	Pass
<b>Functional Test #7:</b> Loss	Confirm the game can be lost.	Pass
<b>Functional Test #8:</b> Restart	Check that after a win or a loss the game can be played again.	Pass
<b>Functional Test #9:</b> Blackout	Assert causing too much acceleration leads to blackout and LED1 blinks with 50% duty cycle.	Pass
<b>Functional Test #10:</b> Configurable	Confirm that the game data is configurable.	Pass

## **Summary**

### **Config Input:**

This test makes sure that the configuration data structure is accessed properly. It does this by doing read and checking the outputs are within the expected ranges. The original plan for this was to have it read a configuration from an external file, but it was decided that the file accessing and parsing was beyond the scope of this project so this unit test is fairly simple.

All functional tests are now passing. The game is fully playable at this time.

### **Previously Passed:**

LCD Display-Week 2  
Angle Change-Week3  
Throttle Change-Week3  
Rocket Translation-Week3  
Rocket Rotation-Week3  
Physics to Rocket-Week4  
Physics Thrust-Week4  
PWM Frequency-Week5  
Physics Blackout-Week5