

## 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	Fail
<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	Fail
<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	Fail

### Functional Testing Plan

Test	Description	Status
<b>Functional Test #1:</b> Game Start	Confirm the game starts and LCD displays welcome screen.	Not Run
<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).	Not Run
<b>Functional Test #5:</b> LED0	Test that LED0 shows current thrust as a % of the maximum via pulse width modulated lighting.	Not Run
<b>Functional Test #6:</b> LED 1 Normal	Test that LED1 shows current acceleration as a % of the maximum.	Not Run
<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.	Not Run
<b>Functional Test #10:</b> Configurable	Confirm that the game data is configurable.	Not Run

## **Summary**

### Physics to Rocket:

The functions to access the rocket data structure work as expected, therefore the physics code is able to get and set values in the rocket struct. The display task also accesses the rocket struct to print it to the lcd.

### Physics Thrust:

The thrust function returns expected values given sample inputs. Function based on updated formula provided by professor Haines where force from thrust equals propulsion velocity times the change in mass.

Multiple functional tests are now passing as initial gameplay features have been implemented.

## **Previously Passed:**

LCD Display-Week 2

Angle Change-Week3

Throttle Change-Week3

Rocket Translation-Week3

Rocket Rotation-Week3