Fourth week: Functiona Test Implementation, more firmware implementation.

https://github.com/carsonsyberg/ecen_3753_final_project

Week 4: Augment your test plan to also have a section on functional tests, and populate it with at least 10 functional tests. Summarize results. ("NotRun" will make sense for many functional tests that are not possible until you get further)

Test 1:

- Upon reset, does the shield platform remain stationary when the capacitive sensor is not touched?
    - Sometimes this test is successful, but other times the shield platform starts accelerating on its own without input from the capacitive touch sensor.

Test 2:

- Does the shield platform accelerate to the right when the right of the capacitive sensor is touched? Does it accelerate left when the left capacitive sensor is touched?
    - Sometimes this test is successful, but other times the shield platform is accelerating on its own without input from the capacitive touch sensor, and placing a finger on the opposite side of the acceleration direction only sets the input force on the shield to 0 instead of overriding it completely.

Test 3:

- Does the holtzmann mass, upon reaching the shield, bounce off of it and arrive at a lower height than it started its descent from?
    - Upon reaching the shield, the holtzmann mass does bounce off and reverse y velocity and arrives at a lower height than where it began.

Test 4:

- Upon pressing button 0, is the holtzmann mass reset to the top of the screen when there is more than one holtzmann mass left and more than zero lasers remaining?
    - Upon pressing button 0, the holtzmann mass is reset to the top of the screen.

Test 5:

- Upon pressing button 1 in a certain specified time interval before the holtzmann mass shield collision and holding it long enough for the bounce to occur, the

holtzmann mass bounces off the shield and arrives at a higher height than it started its descent from.
    - Upon pressing 1 in the right amount of time before the collision, the holtzmann mass does bounce off and arrives at a higher height than where it began.

Test 6:

- After pressing button 0 as many times as the maximum number of lasers, ensure that pressing button 0 again does not change the position of the holtzmann mass or the number of holtzmann masses remaining.
    - After pressing button 0 as many times as the maximum number of lasers, pressing button 0 again no longer has an effect on the holtzmann masses.

Test 7:

- Upon reset, immediately press button 1 and hold until after the bounce moment. Ensure that the holtzmann mass bounces and arrives at a lower height than it started its descent from.
    - Pressing button 1 too early before the bounce moment does not cause the holtzmann mass to bounce to a higher location than where it began, but instead gives it the same bounce as if button 1 had never been pressed.

Test 8:

- Upon reset with holtzmann mass initial x velocity set to 0, allow the holtzmann mass to bounce on the stationary platform with the given set minimum y speed. After five bounces, verify that the holtzmann mass no longer bounces and rather goes through the platform.
    - The holtzmann mass bounces five times off the stationary platform center screen and on the sixth collision it goes through the platform and continues falling.

Test 9:

- Upon reset with holtzmann mass initial x velocity set to 1, verify that the holtzmann mass reverses x velocity upon hitting the wall.
    - The holtzmann mass always reverses x velocity upon hitting the wall.

Test 10:

- After accelerating the platform left with the capacitive sensor, verify that the platform bounces off the wall and reverses x velocity.
    - The platform always bounces off the wall and reverses x velocity.

Accurate summary statement of your functionality deliverables and usability so far.

This week I wrote and debugged the shield enhancement functionality, as well as created 10 functional tests for my test plan.

Summary effort & estimate numbers.

I have completed **76.6%** of my currently-scoped, estimated work (21.18 actually spent /27.67 total estimate) in **114.52%** of the initially-estimated time. (31.68 estimated for the items I have completed, of 27.67 hour estimate). For the work that has been completed, I took **1.61x** (3.5/2.17) as much time as I estimated.

List of in-scope work items (NOT just _this_ week's), **indicating complete or not-yet-complete, along with your estimates of how long you think they will take in total for each** (i.e. no partial-item "credit").

Week	Task	Estimated Time (mins)	Actual Time (mins)	Completed	Description
1	Reading project description	60	75		
1	Creating task diagram	90	90		
1	Deciding on cutting points	15	10		
1	Writing functionality deliverables and usability so far	5	5		
1	Creating summary effort and estimate numbers	10	30		
1	List in-scope work items	45	60		
1	Update risk register	10	10		
2	Create github repository	5	1		Create repository for project.
2	Create project in Simplicity	5	5		Copy and paste an old

	studio				project to build this final project off of.
2	Create data structures and tasks necessary for system	60	45		Create all variables and write all code to initialize OS and tasks.
2	Develop physics model for shield platform	60	100		Research kinematics and find way to model changes in velocity, position, and acceleration of sheld platform.
2	Develop physics model for holtzmann masses	80	90		Research kinematics and find way to model changes in velocity, position, and acceleration of holtzmann masses.
2	Create unit testing plan.	60	90		Develop testing plan for all parts of system possible to test.
3	Write LCDUpdate Task	75	60		Plan out sequence that happens periodically to display canyon walls, base platform, shield platform, and holtzmann masses. Implement commented steps in actual code.
3	Test and debug LCDUpdate Task	60	80		Run tests on displaying all game elements in different controllable positions.
3	Write PhysicsUpdate Task	180	90		Plan out sequence that happens periodically to update the physics on all objects. Implement commented steps in actual code.
3	Test and debug PhysicsUpdate Task	120	120		Run tests on displaying moving game elements following physics models.
3	Write unit testing code.	90	180		Write automated unit testing code for any individually testable parts of the system.
4	Write GPIO even and odd interrupt handlers	60	30		Plan out sequences necessary upon Btn0 and Btn1 presses and write out

					in comment form. Implement commented steps in actual code.
4	Test and debug GPIO interrupts and handlers for functionality	30	10		Run the program with only GPIO interrupts and ensure upon button presses that the ISRs are entered and work as designed.
4	Write ShieldForce Task	60	30		Plan out, in comments, the sequence that happens periodically to sample the capacitive touch slider, update the Shield data, and post flags to the Led1Output Task. Implement commented steps in actual code.
4	Test and debug ShieldForce Task	60	60		Use written tests to check functionality of ShieldForce task.
4	Write functional testing code	45	45		
5	Write ShieldEnhance Task	60			Plan out sequence that follows Btn1 press and updates Shield data in comments. Implement commented steps in actual code.
5	Test and debug ShieldEnhance Task	60			Use written tests to check functionality of ShieldEnhance task.
5	Write LaserUpdate Task	60			Plan out sequence that follows Btn0 press and updates Game data and Holtzmann Mass data in comment form. Implement commented steps in actual code.
5	Test and debug LaserUpdate Task	60			Use written tests to check functionality of LaserUpdate task.
6	Write LED0 and LED1 Tasks	75			Plan out sequences that happen upon flags being set that control the on/off state of the leds. Implement

					commented steps in actual code.

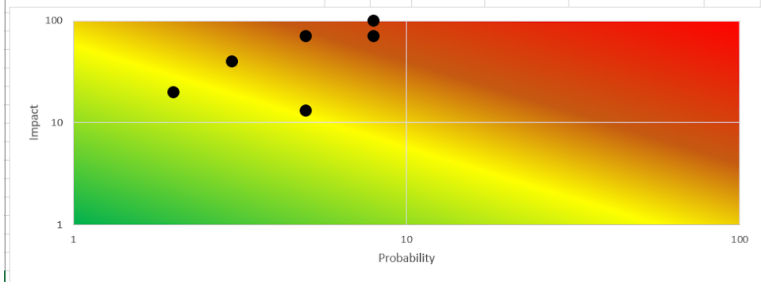
Completed this week:

- Shield enhancement functionality and Functional Tests

This week I spent a lot of time working on getting the shield enhancement functionality to work. I had to debug a good amount and rework some of my physics updates to work with the shield enhancements. I also wrote the interrupt handler for button 1 and the shield enhance task. Lastly, I wrote and tested ten functional tests.

Update your risk register R.O.A.M. your risks each week.

Item	P	I	Risk (P*	Recognize	Mitigated/ Resolve	ROAI	How
I catch CoViD	3	40	120	3-Mar-22	Mitigated	M	learned instructor has extension possible
My gecko is not working	2	20	40	3-Mar-22		O	Test it. If mine not working, use my roommates.
Laptop stops working	5	70	350	3-Mar-22	Mitigated	M	regularly backup code to github. Use campus computers from then on.
Forgot everything about kinematics	8	70	560	3-Mar-22	Resolved	R	research kinematics and find examples of similar physics problems to our game.
Shield platform sometimes gets stuck on walls.	8	100	800	18-Mar-22		O	Continue testing and debugging physics engine to determine cause of sticking.
Modularizing physics task led to errors in behavior	5	13	65	1-Apr-22	Mitigated	M	Can use old code from before modularization that worked well, or continue debugging.
			0				
			0				
			0				



-
-