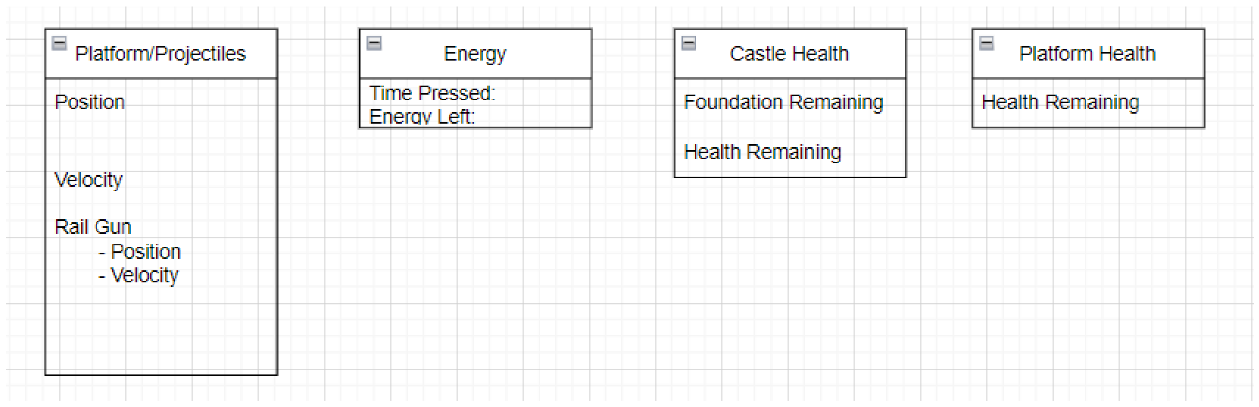
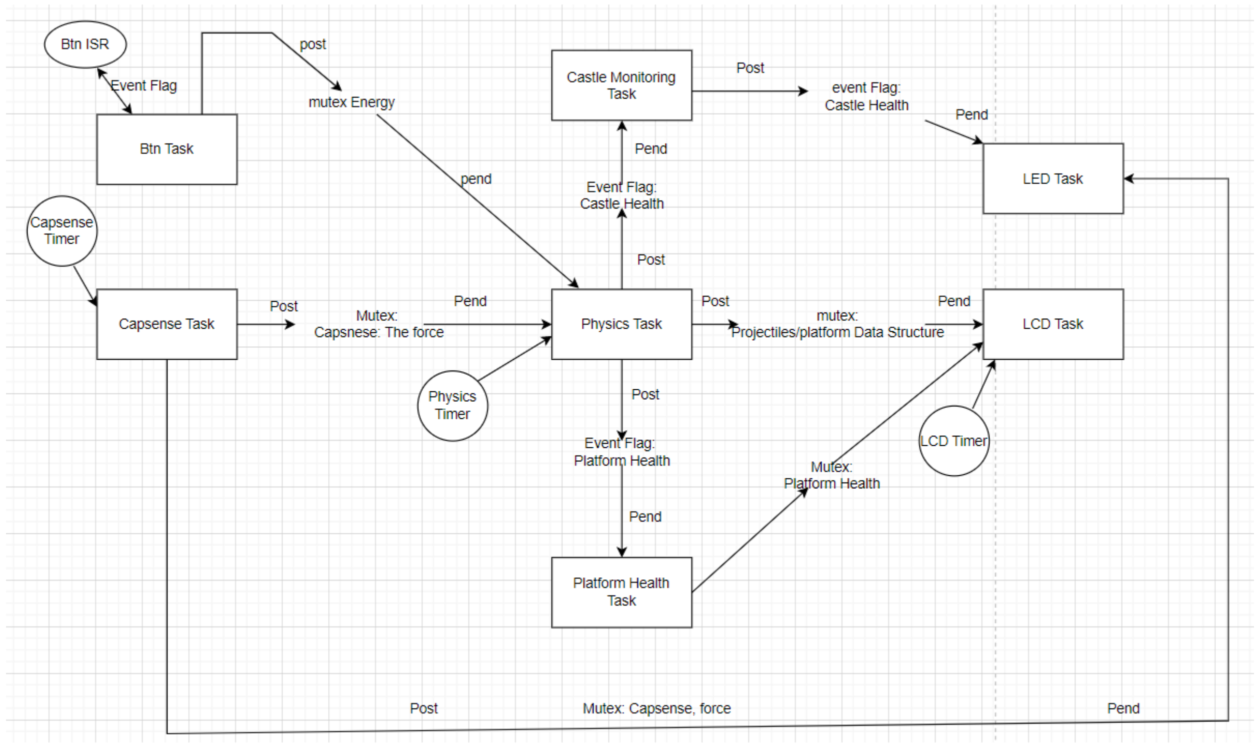




Unit Tests:

Good cutting points for my design are between my physics task and all the other tasks. With these cutting points I will be able to Unit test my button Task, my Capsense Task, my Physics task, my castle Monitoring and Led Task in conjunction, and my LCD Task.

Unit Test 1: Physics Task. I plan to isolate the physics task from all the other tasks so that I am able to test that everything within the task is working correctly. I will test out the force to velocity conversion that I will have to do when the touch slider is touched. I will do this by using a random number generator to generate random forces, within the range that they can be made from touching the slider and, and making sure that platform is in the correct position and has the correct velocity. Another functional test I will perform is to make sure the parabolic position of the projectiles is accurate based on how long the buttons are pressed. In order to accomplish this I will manually calculate where the projectiles should land based on pushing the button in increments of 1 second, and then use the incremental times in my physics task and make sure that the correct trajectories are indeed calculated.

Unit Test 2: Btn Task and ISR. I will utilize another cutting point between my button task and the Physics Task. I will create two tests, one to test that the task can detect when the button is pressed and released and another to establish that the task can determine how long the button was pressed before it is released.

Unit Test 3: LCD Task. I will utilize one of my cutting points at the LCD Task in order to isolate the LCD functionality and test that I am able to display all the things I need to have a functional game. I will test three things: the slider can be displayed within its full range of movement, The projectiles can be displayed in their full range of motion (practically the whole screen), and that the castle can be displayed properly.

Functional Tests:

Physics Task:

- Force applied to platform: I will test if the physics task can accurately move the position of the platform based on varying forces applied to it. I will test three different cases as well as edge cases. I will apply no force (edge case), maximum force (10 seconds on either far edge of the slider), 2 seconds on the far edge of the slider, and 4 seconds on one of the middle sections of the slider.

  Pass/Fail: Would pass the edge case of no force, would most likely fail the other cases
- Railgun while platform is still: I will test if my trajectories are correct for the projectiles when my platform is at rest. I will use varying values for the amount of energy for the projectiles and make sure that the trajectories and final destinations calculated will be correct

  Pass/Fail: Would fail this test since I have not written the code to fully implement it
- RailGun while platform is moving: I will test if the trajectories and final locations for the projectiles are correct with varying values for the amount of energy the projectiles have as well as varying values for the initial velocity the projectiles have based on the platform's velocity.

  Pass/Fail: Would fail this test since I have not yet implemented the code to model this behavior.
- Satchel's being thrown: Test that I can accurately predict where the satchel's will land with a random number generator making the trajectories random.

  Pass/Fail: Will fail because I have not fully implemented this code yet
- Collision with the wall: I will test that my physics task can accurately tell when the platform hits the wall with critical speed. I will accomplish this by inputting varying velocities of the platform and placing the platform at the far edge of the playable area by

the wall and see if the platform health data structure is changed when the platform is at critical velocity upon collision.

Pass/Fail: This test will fail because the code has not been fully implemented to have this functioning

- Shield Test: I will test of the platform is shielded from projectiles correctly when the shield is activated

  Pass/Fail: This test will fail because I am not done with the physics task.
- Health of the platform: I will test if the platform will be "destroyed" after it is hit with a satchel charge. I will test varying trajectories of the satchel charges with the platform moving at a fixed velocity back and forth and make sure that when the trajectory of the satchel lines up with the platform it is destroyed.

  Pass/Fail This test will fail because I have not implemented the physics task yet.
- Castle Health: I will test if the castle is hit by a projectile, and if it is hit if it can accurately monitor if it is hit again before everyone is evacuated.

  Pass/Fail: This test would fail because my physics task is not complete yet
- Castle Health Test 2: I will test if the game will end if the castle is hit twice before the allotted time has run out for the people to evacuate.

  Pass/Fail: This test will fail because the physics task is not fully implemented yet.
- Platform no Velocity test: I will test if the physics task can tell when equal force is being applied to both sides of the platform(both sides of the slider are being touched).

  Pass/Fail: This test will fail because the physics task is not fully implemented yet.

LCD Task:

- Castle: I will test that I can display the correct image for the castle

  Pass/Fail: This test will fail because I havent started the LCD task
- Satchel: I will test that I can display an image of a satchel or something to represent the satchels

  Pass/Fail: This test will fail because I havent started the LCD task
- Area of the Satchels: I will test that I can display the satchels anywhere that it is possible to have a satchel be.

  Pass/Fail: This test will fail because I havent started the LCD task
- Projectiles: I will test that I can display the projectiles or something to represent the projectiles.

  Pass/Fail: This test will fail because I havent started the LCD task
- Area of the Projectiles: I will test that i can display the projectiles in all the areas that it is possible to shoot a projectile

  Pass/Fail: This test will fail because I havent started the LCD task
- Platform Image: I will test that I can display an image that accurately represents where the platform is.

  Pass/Fail: This test will fail because I havent started the LCD task
- Platform Movement: I will test that I can display the platform in all the possible locations that the platform can be.

  Pass/Fail: This test will fail because I havent started the LCD task

- Rail Gun Display: I will test that I can display an accurate image of the rail gun that depicts where the projectiles are shot from.
  Pass/Fail: This test will fail because I havent started the LCD task
- End Game: I will test that I can display the message when you either win or lose.
  Pass/Fail: This test will fail because I havent started the LCD task
- (OUT OF SCOPE) Test that I can display the health of the platform and the castle in some form. This is out of scope, but I think it would add a lot to the game if I have time to implement it.
  Pass/Fail: This test will fail because I havent started the LCD task

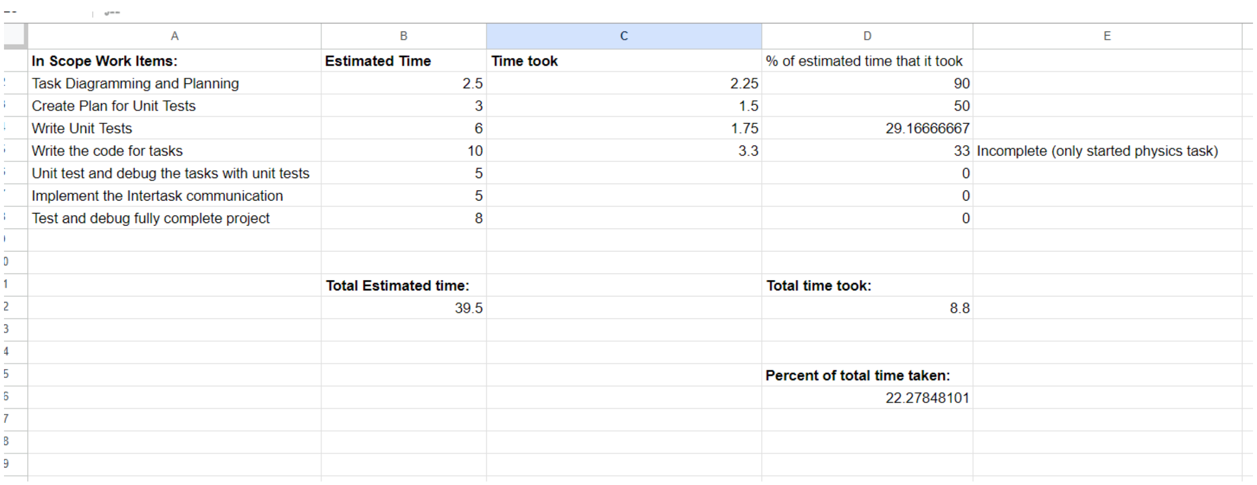
Functionality and Usability:

Summary of Effort:

The total time I have outlined for the in scope work for this project is 39.5 hours. The work for this week took me a total of 5 hours. This accounts for 12.65% (5 hours of estimated work completed out of 39.5 hours of total work time) of the estimated total inscope work in 24.6% of the budgeted time (9.75 hours of time spent out of 39.5 hours of total work time). For the work this week I took 120% of the original time I had outlined for the inscope work, so I was more efficient than I originally anticipated.

In Scope Work Items:

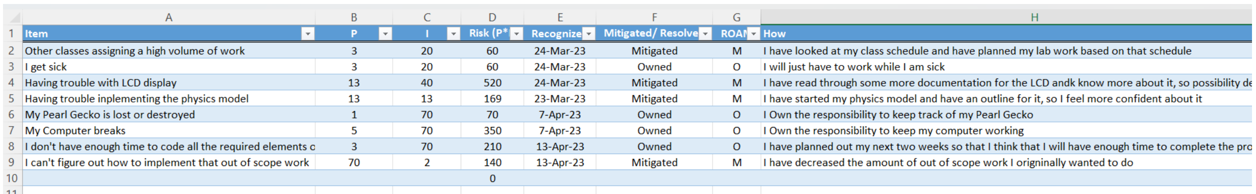
- Task Diagramming and planning: 2.5 hours (**Completed: 2.25 hours)**
- Create Plan for Unit Tests: 3 hours (**Complete: 2.5 hours**)
- Write the Unit Tests: 6 hours (**Complete: 1.75 Hours**)
- Write the code for the tasks: 10 hours (**Started/Incomplete - Physics task started, 3.3 Hours so far**)
- Unit test and debug the tasks based on cutting points: 5 hours (**Incomplete**)
- Implement the intertask communication methods: 5 hours (**Incomplete**)
- Test and debug the fully complete code: 8 hours (**Incomplete**)
  Total: 39.5 hours



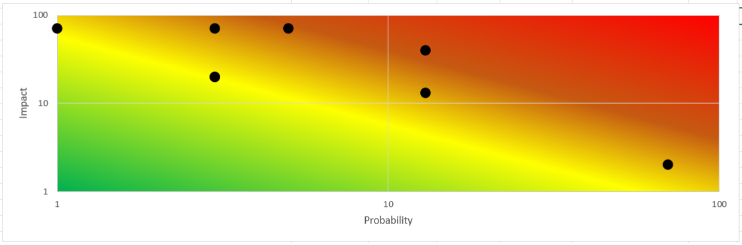
In Scope Work Items:	Estimated Time	Time took	% of estimated time that it took	
Task Diagramming and Planning	2.5	2.25	90	
Create Plan for Unit Tests	3	1.5	50	
Write Unit Tests	6	1.75	29.16666667	
Write the code for tasks	10	3.3	33	Incomplete (only started physics task)
Unit test and debug the tasks with unit tests	5		0	
Implement the Intertask communication	5		0	
Test and debug fully complete project	8		0	
	Total Estimated time:		**Total time took:**	
	39.5		8.8	
			Percent of total time taken:	
			22.27848101	

- **Functional Tests and Physics Task**

  This week I continued working on my physics task (still incomplete) and wrote 20 functional tests (10 for 2 different Unit tests). I continued to implement firmware this week, I would estimate I am 60% done with my physics task and have implemented all the task initialization elements, such as stack allocation, and the task create function calls.



Item	P	I	Risk (P*	Recognize	Mitigated/ Resolve	ROAI	How
Other classes assigning a high volume of work	3	20	60	24-Mar-23	Mitigated	M	I have looked at my class schedule and have planned my lab work based on that schedule
I get sick	3	20	60	24-Mar-23	Owned	O	I will just have to work while I am sick
Having trouble with LCD display	13	40	520	24-Mar-23	Mitigated	M	I have read through some more documentation for the LCD andk know more about it, so possibility de
Having trouble inplementing the physics model	13	13	169	23-Mar-23	Mitigated	M	I have started my physics model and have an outline for it, so I feel more confident about it
My Pearl Gecko is lost or destroyed	1	70	70	7-Apr-23	Owned	O	I Own the responsibility to keep track of my Pearl Gecko
My Computer breaks	5	70	350	7-Apr-23	Owned	O	I Own the responsibility to keep my computer working
I don't have enough time to code all the required elements c	3	70	210	13-Apr-23	Owned	O	I have planned out my next two weeks so that I think that I will have enough time to complete the pro
I can't figure out how to implement that out of scope work	70	2	140	13-Apr-23	Mitigated	M	I have decreased the amount of out of scope work I origninally wanted to do
			0				


Impact
Probability